Symulacje stochastyczne i metody Monte Carlo

Wojciech Niemiro [1]

[1]Wydział Wydział Matematyki, Informatyki i Mechaniki, Uniwersytet Warszawski oraz Wydział Matematyki i Informatyki, Uniwersytet Mikołaja Kopernika w Toruniu. wniem@mimuw.edu.pl, wniem@mat.umk.pl

Spis treści

Część I

Generowanie zmiennych i procesów losowych

Rozdział 1

Generowanie zmiennych losowych

1.1 „Liczby losowe”

U podstaw symulacji stochastycznych leży generowanie „liczb pseudo-losowych”, naśladujących zachowanie zmiennych losowych o rozkładzie jednostajnym na przedziale $[0, 1]$. Jak to się robi, co to jest „pseudo-losowość”, czym się różni od „prawdziwej losowości”? Dyskusja na ten temat przekracza ramy tych wykładów. Z punktu widzenia użytkownika, „liczby pseudo-losowe” można dość bezpiecznie traktować jako „losowe”. Dobre generatory liczb losowych są łatwo dostępne. Przyjmę pragmatyczny punkt widzenia i zacznę od następującego założenia.

1.1.1 Założenie. *Mamy do dyspozycji potencjalnie nieskończony ciąg niezależnych zmiennych losowych* $U_1, \dots U_n \dots$ *o jednakowym rozkładzie* $\mathrm{U}(0, 1)$.

W języku algorytmicznym: przyjmujemy, że każdorazowe wykonanie instrukcji zapisanej w pseudokodzie

```
Gen U ~ U(0,1)
```

wygeneruje kolejną (nową) zmienną U_n. Innymi słowy, zostanie wykonane nowe, niezależne doświadczenie polegające na wylosowaniu przypadkowo wybranej liczby z przedziału $[0, 1]$.

1.1.2 Przykład. Efektem wykonania pseudokodu

```
for i = 1 to 12
  begin
  Gen U ~ U(0,1)
  write U
end
```

są, powiedzmy, liczby

0.32240106	0.38971803	0.35222521	0.22550039	0.04162166	0.0539661
0.13976025	0.16943910	0.69482111	0.28812341	0.58138865	0.9955146

Nawiasem mówiąc, rzeczywisty kod w R, który wyprodukował nasze 12 liczb losowych był taki:

```
U <- runif(12); U
```

△

W Części I zajmujemy się pytaniem, jak „wyprodukować" zmienne losowe o różnych rozkładach, wykorzystując zmienne $U_1, U_2, \ldots$. Najpierw pokażę zabawny przykład, a w następnym podrozdziale przedstawię kilka poważnych metod.

1.1.3 Przykład (Przybliżona generacja rozkładu normalnego)**.** Zmienna losowa

$$X = \sum_{i=1}^{12} U_i - 6$$

ma *w przybliżeniu* standardowy rozkład normalny $\mathrm{N}(0,1)$. Wynika to z Centralnego Twierdzenia Granicznego (jeśli uznamy, że liczba 12 jest dostatecznie bliska ∞; zauważmy, że $\mathbb{E}X = 0$ i $\mathrm{Var}X = 1$). Można zbadać (na przykład symulacyjnie!) jak dobre jest przybliżenie. Dość trudno odróżnić próbkę $X_1, \ldots, X_n$ wyprodukowaną przez powyższy algorytm od próbki pochodzącej *dokładnie* z rozkładu $\mathrm{N}(0,1)$ (chyba, że n jest ogromne).

Oczywiście, w czasach szybkich komputerów przedstawiona tu przybliżona metoda zdecydowanie *nie jest polecana*! Istnieją efektywne, dokładne algorymy generujące próbki z rozkładu normalnego. Parę z nich przedstawię w dalszej części tych wykładów. △

1.1.4 Uwaga. W tym miejscu należy się dygresja. Przyjmujemy natępującą umowę. Algorytm uważamy za *dokładny*, jeśli zmienna losowa na wyjściu ma dokładnie rozkład docelowy przy założeniu, że operacje arytmetyczne są wykonywane bezbłędnie i liczby losowe na wejściu są niezależnymi zmiennymi losowymi o rozkładzie jednostajnym. (Jest to, oczywiście, pewna idealizacja.)

Wszystkie algorytmy przedstawione w Części I są dokładne. Wyjątkiem jest Przykład 1.1.3.

1.2 Odwracanie dystrybuanty

Metoda opiera się na prostym fakcie. Jeżeli F jest ciągłą i ściśle rosnącą dystrybuantą, $U \sim \mathrm{U}(0,1)$ i $X = F^{-1}(U)$, to $X = F^{-1}(U) \sim F$.

1.2.1 Przykład (Rozkład Wykładniczy)**.** To jest wyjątkowo łatwy do generowania rozkład – wystarczy taki algorytm:

`Gen` U`;` $X := -\frac{1}{\lambda}\log U$

Na wyjściu, $X \sim \mathrm{Ex}(\lambda)$. Żeby się o tym przekonać, wystarczy obliczyć dystrybuantę tej zmiennej losowej: $\mathbb{P}(X \leqslant x) = \mathbb{P}(-\frac{1}{\lambda}\log U \leqslant x) = \mathbb{P}(U \geqslant \mathrm{e}^{-\lambda x}) = 1 - \mathrm{e}^{-\lambda x}$. Jest to najprostszy przykład ogólnej metody „odwracania dystrybuanty". △

Następująca definicja funkcji „pseudo-odwrotnej" (zwanej też *funkcją kwantylową*) pozwala pozbyć się kłopotliwych założeń o odwracalności dystrybuanty.

1.2.2 Definicja. *Jeżeli $F : \mathbb{R} \to [0,1]$ jest dowolną dystrybuantą, to funkcję $F^- :]0,1[\to \mathbb{R}$ określamy wzorem:*

$$F^-(u) = \inf\{x : F(x) \geqslant u\}.$$

1.2.3 Stwierdzenie. *Nierówność $F^-(u) \leqslant x$ jest równoważna $u \leqslant F(x)$, dla dowolnych $u \in]0,1[$ i $x \in \mathbb{R}$.*

Dowód. Z prawostronnej ciągłości dystrybuanty F wynika, że kres dolny w Definicji 1.2.2 jest *osiągany*, czyli

$$F(F^-(u)) \geqslant u.$$

Z drugiej strony,

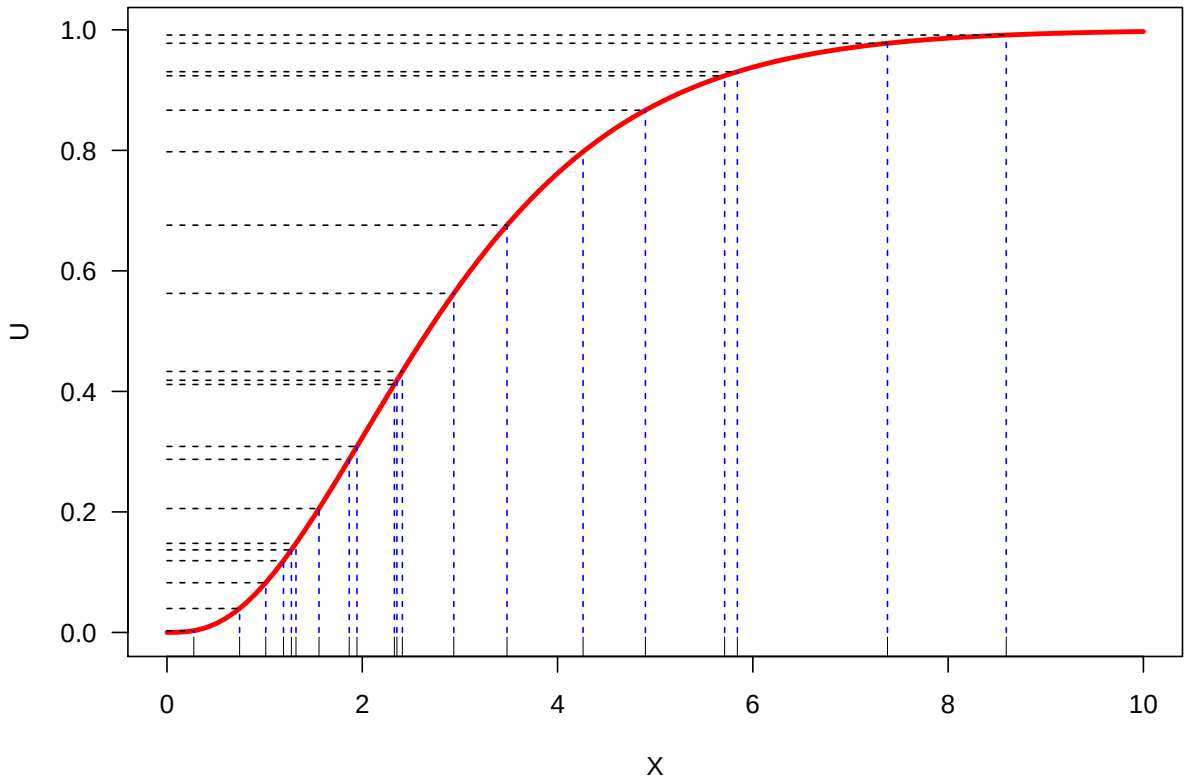
$$F^-(F(x)) = \min\{y : F(y) \geqslant F(x)\} \leqslant x,$$

po prostu dlatego, że $x \in \{y : F(y) \geqslant F(x)\}$. Teza stwierdzenia natychmiast wynika z dwóch nierówności powyżej. □

1.2.4 Wniosek (Ogólna metoda odwrócenia dystrybuanty)**.** *Jeżeli $U \sim \mathrm{U}(0,1)$ i $X = F^-(U)$, to $\mathbb{P}(X \leqslant x) = F(x)$. W skrócie, $X \sim F$.*

Na Rysunku 1.1 widać 20 punktów $U_1, \ldots, U_{20}$, które wylosowałem z rozkładu $\mathrm{U}(0,1)$ (na osi pionowej) i odpowiadające im punkty $X_i = F^{-1}(U_i)$ (na osi poziomej).

Zauważmy, że ta metoda działa również dla rozkładów dyskretnych i sprowadza się wtedy do metody „oczywistej".



Rysunek 1.1: Zmienne losowe U_i i $X_i = F^-(U_i)$.

1.2.5 Przykład (Rozkłady dyskretne)**.** Załóżmy, że $\mathbb{P}(X = i) = p_i$ dla $i = 1, 2, \ldots$ i $\sum p_i = 1$. Niech $s_0 = 0$, $s_k = \sum_{i=1}^{k} p_i$. Jeżeli F jest dystrybuantą zmiennej losowej X, to

$$F^-(u) = i \quad \text{wtedy i tylko wtedy gdy} \quad s_{i-1} < u \leqslant s_i.$$

△

Odwracanie dystrybuanty ma ogromne znaczenie teoretyczne, bo jest całkowicie ogólną metodą generowania dowolnych zmiennych losowych jednowymiarowych. Może się to wydać dziwne, ale w praktyce ta metoda jest używana stosunkowo rzadko, z dwóch względów:

- Obliczanie F^- bywa trudne i nieefektywne.
- Stosowalność metody ogranicza się do zmiennych losowych jednowymiarowych.

Podam dwa przykłady, w których faktycznie stosuje się metodę odwracanie dystrybuanty.

1.2.6 Przykład (Rozkład Weibulla)**.** Z definicji, $X \sim \text{Weibull}(\beta)$, jeśli

$$F(x) = 1 - \exp(-x^\beta)$$

dla $x \geqslant 0$. Odwrócenie dystrybuanty i generacja X są łatwe:

$$X = (-\ln U)^{1/\beta}, \quad U \sim \text{U}(0,1).$$

△

1.2.7 Przykład (Rozkład Cauchy'ego)**.** Gęstość i dystrybuanta zmiennej $X \sim \text{Cauchy}(0,1)$ są następujące:

$$p(x) = \frac{1}{\pi}\frac{1}{1+x^2}, \qquad F(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(x).$$

Można tę zmienną generować korzystając z wzoru:

$$X = \tan\left(\pi\left(U - \frac{1}{2}\right)\right), \quad U \sim \text{U}(0,1).$$

△

1.2.8 Uwaga (Przestrzeń probabilityczna). W tym skrypcie pokazuję, jak generować różne zmienne losowe, używając ciągu $U_1, \ldots, U_n, \ldots \sim_{\text{i.i.d.}} \text{U}(0,1)$ (zobacz Założenie 1.1). W tym sensie, wszystkie rozpatrywane zmienne losowe są określone na przestrzeni probabilistycznej $(\Omega = [0,1]^\infty, \mathcal{F} = \mathcal{B}^\infty, \mathbb{P} = \text{Leb}^\infty)$, gdzie $\mathcal{B}$ jest σ-ciałem podzbiorów borelowskich, zaś Leb jest miarą Lebesgue'a na przedziale $[0,1]$. W istocie, Zadanie 1.3 pokazuje, że jeśli zmienna losowa X jest funkcją ciągu $U_1, \ldots, U_n, \ldots$, to jest postaci $X = \phi(U)$ dla odpowiednio dobranej funkcji ϕ. Możemy zatem przyjąć za przestrzeń probabilistyczną $(\Omega = [0,1], \mathcal{B}, \text{Leb})$. Można pokazać (wykracza to poza ramy naszych rozważań), że *każdy* rozkład prawdopodobieństwa na $\mathbb{R}^d$ (a nawet więcej, każdy rozkład na przestrzeni polskiej) jest rozkładem prawdopodobieństwa zmiennej postaci $X = \phi(U)$. Wniosek 1.2.4 stanowi prosty dowód tego faktu w szczególnym przypadku zmiennych losowych jednowymiarowych.

1.3 Metoda przekształceń

Odwracanie dystrybuanty jest szczególnym przypadkiem metody przekształceń. Załóżmy, że umiemy losować zmienną losową X. Zmienna losowa Y, która ma postać $Y = h(X)$, a więc jest pewną funkcją zmiennej X, w naturalny sposób „dziedziczy" rozkład prawdopodobieństwa zgodnie ze schematem $\mathbb{P}(Y \in \cdot) = \mathbb{P}(h(X) \in \cdot) = \mathbb{P}(X \in h^{-1}(\cdot))$ („wykropkowany" argument jest zbiorem; zmienne X i Y nie muszą być jednowymiarowe). Odpowiednio dobierając funkcję h możemy „przetwarzać" jedne rozkłady prawdopodobieństwa na inne. Następujące twierdzenie jest podstawowym narzędziem obliczania gęstości przekształconych zmiennych losowych.

1.3.1 Twierdzenie. *Załóżmy, że X jest d-wymiarową zmienną losową o wartościach w otwartym zbiorze $A \subseteq \mathbb{R}^d$. Jeżeli zmienna X ma gęstość p_X względem d-wymiarowej miary Lebesgue'a i $h : A \to B \subseteq \mathbb{R}^d$ jest dyfeomorfizmem, to zmienna $Y = h(X)$ ma gęstość daną wzorem*

$$p_Y(y) = p_X(h^{-1}(y)) \left|\det \mathrm{D}h^{-1}(y)\right|,$$

gdzie D *oznacza macierz pochodnych cząstkowych ($d \times d$).*

Zwróćmy uwagę, że w tym miejscu mówimy o gęstościach względem miary Lebesgue'a i twierdzenie ogranicza się do przekształceń „zachowujących wymiar". Jeśli h jest przekształceniem z $\mathbb{R}^d$ do $\mathbb{R}^k$, gdzie $k < d$, to można wprowadzić $d - k$ „pomocniczych zmiennych" i „rozszerzyć" h do przekształcenia z $\mathbb{R}^d$ do $\mathbb{R}^d$ – a potem „wycałkować niepotrzebne zmienne". Zobacz na przykład dowód Twierdzenia 1.6.7.

Prawie wszystkie algorytmy generowania zmiennych losowych zawierają metodę przekształceń jako część składową. W tym miejscu ograniczę się do podania jednego przykładu, w którym metoda przekształceń występuje w „czystej postaci". Przedstawiony poniżej algorytm jest współczesną, dokładną metodą generowania zmiennych o rozkładzie normalnym. Ciekawe, że łatwiej jest generować zmienne losowe normalne „parami".

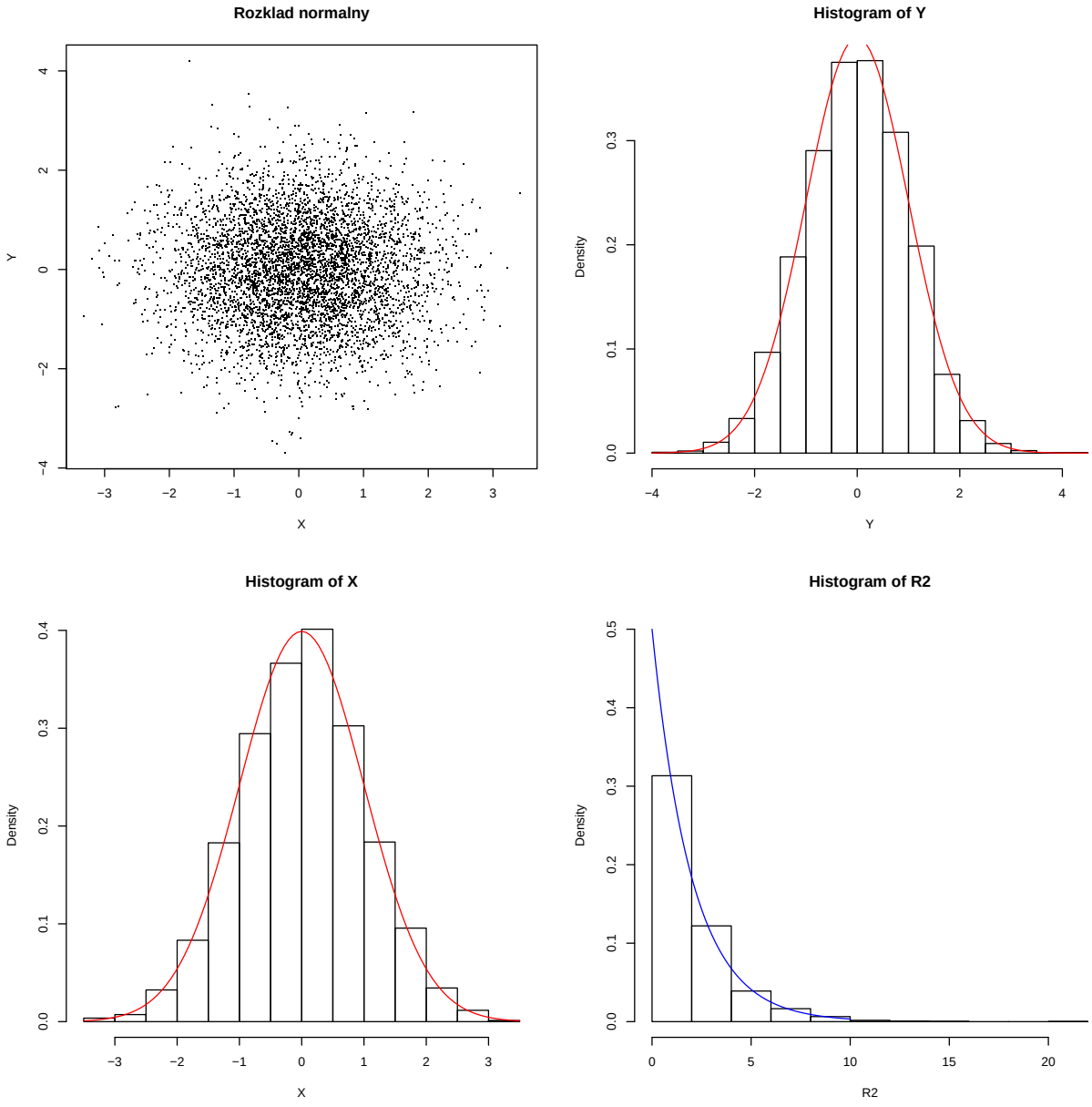
1.3.2 Przykład (Algorytm Boxa-Müllera)**.** Algorytm opiera się na przejściu do współrzędnych biegunowych w $\mathbb{R}^2$.

`Gen` U_1; $\Theta := 2\pi U_1$,
`Gen` U_2; $R := \sqrt{-2\log U_2}$,
`Gen` $X := R\cos\Theta$; $Y := R\sin\Theta$

Na wyjściu *obie* zmienne X i Y mają rozkład $\mathrm{N}(0,1)$ i w dodatku są niezależne. Uzasadnienie poprawności algorytmu Boxa-Müllera opiera się na dwu faktach: zmienna $R^2 = X^2 + Y^2$ ma rozkład $\chi^2(2) = \mathrm{Ex}(1/2)$, zaś kąt Θ między osią i promieniem wodzącym punktu (X, Y) ma rozkład $\mathrm{U}(0, 2\pi)$. △

Doświadczenie, polegające na wygenerowaniu zmiennych losowych X i Y powtórzyłem 10000 razy. Na Rysunku 1.2 widać 10000 wylosowanych w ten sam sposób i niezależnie punktów (X, Y), histogramy i gęstości brzegowe X i Y (każda ze współrzędnych ma rozkład $\mathrm{N}(0,1)$) oraz histogram i gęstość $R^2 = X^2 + Y^2$ (R^2 ma rozkład wykładniczy $\mathrm{Ex}(1/2)$).

Histogram jest „empirycznym" (może w obecnym kontekście należałoby powiedzieć „symulacyjnym") odpowiednikiem gęstości: spośród wylosowanych wyników zliczane są punkty należące do poszczególnych przedziałów.



Rysunek 1.2: Dwuwymiarowy rozkład normalny i rozkłady brzegowe.

1.4 Metoda eliminacji

To jest najważniejsza, najczęściej stosowana i najbardziej uniwersalna metoda. Zacznę od raczej oczywistego faktu, który jest w istocie probabilistycznym sformułowaniem definicji prawdopodobieństwa warunkowego.

1.4.1 Stwierdzenie. *Przypuśćmy, że $Z = Z_1, \ldots, Z_n, \ldots$ jest ciągiem niezależnych zmiennych losowych o jednakowym rozkładzie, o wartościach w przestrzeni $\mathcal{Z}$. Niech $C \subseteq \mathcal{Z}$ będzie takim zbiorem, że $\mathbb{P}(Z \in C) > 0$. Niech*

$$N = \min\{n : Z_n \in C\}.$$

Zmienne losowe N i Z_N są niezależne, przy tym

$$\mathbb{P}(Z_N \in B) = \mathbb{P}(Z \in B | Z \in C) \quad \textit{dla dowolnego} \quad B \subseteq \mathcal{Z},$$

zaś

$$\mathbb{P}(N = n) = (1-\theta)^{n-1}\theta, \quad (n = 1, 2, \ldots), \quad \textit{gdzie} \quad \theta = \mathbb{P}(Z \in C).$$

Dowód. Wystarczy zauważyć, że

$$\begin{aligned}\mathbb{P}(X_N \in B, N = n) &= \mathbb{P}(Z_1 \notin C, \ldots, Z_{n-1} \notin C, Z_n \in C \cap B) \\ &= \mathbb{P}(Z_1 \notin C) \cdots \mathbb{P}(Z_{n-1} \notin C)\mathbb{P}(Z_n \in C \cap B) \\ &= (1-\theta)^{n-1}\mathbb{P}(Z \in C \cap B) = (1-\theta)^{n-1}\theta \cdot \mathbb{P}(Z \in B | Z \in C).\end{aligned}$$

□

W tym Stwierdzeniu $\mathcal{Z}$ może być dowolną przestrzenią mierzalną, zaś C i B – dowolnymi zbiorami mierzalnymi. Stwierdzenie mówi po prostu, że prawdopodobieństwo warunkowe odpowiada doświadczeniu losowemu *powtarzanemu aż do momentu spełnienia warunku*, przy czym rezultaty poprzednich doświadczeń się ignoruje (stąd nazwa: eliminacja).

Uwaga (Losowa liczba akceptacji). Zasadniczy algorytm eliminacji opisany powyżej polega na powtarzaniu generacji tak długo, aż zostanie spełnione kryterium akceptacji. *Liczba prób* N jest losowa i ma rozkład geometryczny $\mathrm{Geo}(\theta)$, a rezultatem jest jedna zmienna losowa o zadanym rozkładzie. Specyfika języka R narzuca inny sposób przeprowadzania eliminacji. Działamy na wektorach, a więc od razu produkujemy n niezależnych zmiennych $Z_1, \ldots, Z_n$, następnie poddajemy je wszystkie procedurze eliminacji. Przez sito eliminacji (spełnienie warunku $Z_i \in C$) przechodzi K spośród nich. Liczba zaakceptowanych zmiennych K ma oczywiście rozkład dwumianowy $\mathrm{Bin}(n, \theta)$ z $\theta = \mathbb{P}(Z \in C)$. Otrzymujemy więc *losową liczbę zmiennych* o rozkładzie docelowym. Dla przykładu, generowanie zmiennych (X, Y) o rozkładzie jednostajnym na kole jednostkowym $\{x^2 + y^2 \leqslant 1\}$ może w praktyce wyglądać tak:

```
> n <- 1000
> X <- runif(n,min=-1,max=1)  \# generowanie z rozkładu U(-1,1)
> Y <- runif(n,min=-1,max=1)
> Accept <- X^2+Y^2<1         \# wektor logiczny
> X <- X[Accept]
> Y <- Y[Accept]
```

Otrzymujemy pewną losową liczbę (około 785) punktów (X, Y) w kole jednostkowym.

Metoda eliminacji dla gęstości

Zakładamy, że umiemy generować zmienne losowe o gęstości q, a chcielibyśmy otrzymać zmienną o gęstości p. Mówimy, że p jest gęstością *docelową* a q *instrumentalną*.

Ważną zaletą przedstawionego poniżej algorytmu jest to, że nie trzeba znać „stałych normujących" obu gęstości p i q. Wystarczy, że potrafimy obliczać wartości funkcji *proporcjonalnych* do gęstości, $\tilde{p} \propto p$ i $\tilde{q} \propto q$ (to znaczy $p(x) = \tilde{p}(x)/c_p$ i $q(x) = \tilde{q}(x)/c_q$ dla pewnych stałych c_p i c_q). Zakładamy, że $\tilde{p} \leqslant \tilde{q}$. Następujący algorytm produkuje zmienną X o gęstości p.

`repeat`
  `Gen` $X \sim q$;
  `Gen` $U \sim \mathrm{U}(0,1)$
`until` $U \leqslant \dfrac{\tilde{p}(X)}{\tilde{q}(X)}$;
`return` X

Dowód poprawności algorytmu. Niech $\mathcal{X}$ będzie przestrzenią wartości zmiennej losowej X. Zastosujemy Stwierdzenie 1.4.1 do zmiennej losowej $Z = (X, U)$ na przestrzeni $\mathcal{Z} = \mathcal{X} \times [0,1]$. Na mocy tego stwierdzenia wystarczy pokazać, że

$$\mathbb{P}\left(X \in B \middle| U \leqslant \frac{\tilde{p}(X)}{\tilde{q}(X)}\right) = \frac{1}{c_p}\int_B \tilde{p}(x)\mathrm{d}x. \tag{1.4.2}$$

Z niezależności zmiennych losowych X i U wynika, że para (X, U) ma łączną gęstość równą $q(y) \cdot 1$. Zatem

$$\mathbb{P}\left(X \in B, U \leqslant \frac{\tilde{p}(X)}{\tilde{q}(X)}\right) = \int_B \int_0^{\tilde{p}(x)/\tilde{q}(x)} \mathrm{d}u q(x)\mathrm{d}x = \int_B \frac{\tilde{p}(x)}{\tilde{q}(x)} q(x)\mathrm{d}x = \frac{1}{c_q}\int_B \tilde{p}(x)\mathrm{d}x.$$

Zastępując B przez $\mathcal{X}$ otrzymujemy

$$\mathbb{P}\left(U \leqslant \frac{\tilde{p}(X)}{\tilde{q}(X)}\right) = \frac{1}{c_q}\int_{\mathcal{X}} \tilde{p}(x)\mathrm{d}x = \frac{c_p}{c_q}.$$

Wzór (1.4.2) dostaniemy dzieląc stronami dwie ostatnie równości. □

Uwaga. W istocie $\mathcal{X}$ może być ogólną przestrzenią z miarą μ. Wystarczy przyjąć umowę, że symbol $\int \cdots \mathrm{d}x$ jest skrótem dla $\int \cdots \mu(\mathrm{d}x)$.

Uwaga. Efektywność algorytmu zależy od dobrania gęstości q w taki sposób, aby funkcja $\tilde{q}$ majoryzowała funkcję $\tilde{p}$ ale nie była dużo od niej większa. Istotnie, liczba prób N do zaakceptowania X ma rozkład geometryczny z prawdopodobieństwem sukcesu $\int \tilde{p}/\int \tilde{q}$, zgodnie ze Stwierdzeniem 1.4.1, zatem $\mathbb{E}N = \int \tilde{q}/\int \tilde{p}$. Ten iloraz powinien być możliwie bliski jedynki, co jest możliwe jeśli „kształt funkcji $\tilde{q}$ jest podobny do $\tilde{p}$".

Zwykle metodę eliminacji stosuje się w połączeniu z odpowiednio dobranymi przekształceniami. Doskonałą ilustracją jest rodzina algorytmów przedstawiona w następnym podrozdziale.

Iloraz zmiennych równomiernych

Szczególnie często stosowany jest specjalny przypadek metody eliminacji, znany jako algorytm „ilorazu zmiennych równomiernych" (*Ratio of Uniforms*). Zaczniemy od prostego przykładu, który wyjaśni tę nazwę.

1.4.3 Przykład (Rozkład Cauchy'ego)**.** Metodą eliminacji z prostokąta $[0,1]\times[-1,1]$ otrzymujemy zmienną losową (U,V) o rozkładzie jednostajnym na półkolu $\{u \geqslant 0, u^2+v^2 \leqslant 1\}$. Kąt Φ pomiędzy osią poziomą i punktem (U,V) ma, oczywiście, rozkład $\mathrm{U}(-\pi,\pi)$.

`repeat`
  `Gen` $U \sim \mathrm{U}(0,1)$`;`
  `Gen` $V \sim \mathrm{U}(-1,1)$
`until` $U^2+V^2<1$
$X := V/U$

Na wyjściu $X \sim$ Cauchy, bo

$$\mathbb{P}(X \leqslant x) = \mathbb{P}(V \leqslant xU) = \frac{1}{\pi}\Phi + \frac{1}{2} = \frac{1}{\pi}\arctan(x) + \frac{1}{2}.$$

△

Ogólny algorytm metody „ilorazu równomiernych" oparty jest na następującym fakcie.

1.4.4 Stwierdzenie. *Załóżmy o funkcji* $h:\mathbb{R}\to\mathbb{R}$*, że*

$$h(x) \geqslant 0, \int h(x)dx < \infty.$$

Niech zbiór $C_h \subset \mathbb{R}^2$ *będzie określony następująco:*

$$C_h = \left\{(u,v) : 0 \leqslant u \leqslant \sqrt{h\left(\frac{v}{u}\right)}\right\}.$$

Miara Lebesgue'a tego zbioru (pole figury) jest skończone, $|C_h|<\infty$*, a zatem można mówić o rozkładzie jednostajnym* $\mathrm{U}(C_h)$*.*

Jeżeli $(U,V) \sim \mathrm{U}(C_h)$ *i* $X = V/U$*, to* X *ma gęstość proporcjonalną do funkcji* h *(*$X \sim h/\int h$*).*

Dowód. „Pole figury” C_h jest równe

$$|C_h| = \iint_{C_h} \mathrm{d}u\mathrm{d}v = \iint_{0\leqslant u\leqslant\sqrt{h(x)}} u\mathrm{d}u\mathrm{d}x$$
$$= \int_{-\infty}^{\infty} \int_{0}^{\sqrt{h(x)}} \mathrm{d}u\mathrm{d}x = \frac{1}{2}\int_{-\infty}^{\infty} h(x)\mathrm{d}x < \infty$$

Dokonaliśmy tu zamiany zmiennych:

$$(u, v) \mapsto \left(u, x = \frac{v}{u}\right).$$

Jakobian przekształcenia odwrotnego $(u, x) \mapsto (u, v = ux)$ jest równy

$$\frac{\partial(u, v)}{\partial(u, x)} = \det \begin{pmatrix} 1 & 0 \\ x & u \end{pmatrix} = u.$$

W podobny sposób, na mocy znanego wzoru na gęstość przekształconych zmiennych losowych, Twierdzenie 1.3.1, obliczamy łączną gęstość (U, X):

$$p_{U,X}(u, x) = up_{U,V}(u, ux) \quad \text{na zbiorze } \{0 \leqslant u \leqslant \sqrt{h(x)}\}.$$

Stąd dostajemy gęstość brzegową X:

$$\int_0^{\sqrt{h(x)}} \frac{u\,du}{|C_h|} = \frac{h(x)}{2|C_h|}.$$

□

Żeby wylosować $(U, V) \sim \mathrm{U}(C_h)$ stosuje się zazwyczaj eliminację z prostokąta. Użyteczne jest następujące oszacowanie boków tego prostokąta.

1.4.5 Stwierdzenie. *Jeśli funkcje $h(x)$ i $x^2h(x)$ są ograniczone, wtedy*

$$C_h \subseteq [0, a] \times [b_-, b_+],$$

gdzie

$$a = \sqrt{\sup_x h(x)},$$

$$b_+ = \sqrt{\sup_{x\geqslant 0}[x^2h(x)]}, \qquad b_- = -\sqrt{\sup_{x\leqslant 0}[x^2h(x)]}.$$

Dowód. Jeśli $(u,v) \in C_h$ to oczywiście $0 \leqslant u \leqslant \sqrt{h(v/u)} \leqslant \sqrt{\sup_x h(x)}$. Załóżmy dodatkowo, że $v \geqslant 0$ i przejdźmy do zmiennych (v,x), gdzie $x = v/u$. Nierówność $u \leqslant \sqrt{h(v/u)}$ jest równoważna $v^2 \leqslant x^2 h(x)$. Ponieważ $x \geqslant 0$, więc dostajemy $v^2 \leqslant b_+^2$. Dla $v \leqslant 0$ mamy analogicznie $v^2 \leqslant b_-^2$. □

Ogólny algorytm RU jest następujący:

`repeat`
  `Gen` U_1, U_2`;`
  $U := aU_1$`;` $V := b_- + (b_+ - b_-)U_2$
`until` $(U,V) \in C_h$`;`
$X := \dfrac{V}{U}$

1.4.6 Przykład (Rozkład normalny)**.** Niech $h(x) = \exp\left(-\frac{1}{2}x^2\right)$. Wtedy

$$C_h = \left\{(u,v) : 0 \leqslant u \leqslant \exp\left(-\frac{1}{4}\frac{v^2}{u^2}\right)\right\} = \left\{\frac{v^2}{u^2} \leqslant -4\ln u\right\}.$$

Zauważmy, że $a = 1$ i $b_+ = -b_- = \sqrt{2e^{-1}}$. Otrzymujemy następujący algorytm:

`repeat`
  `Gen` U_1, U_2
  $U := U_1$`;` $V := \sqrt{2\mathrm{e}^{-1}}(2U_2 - 1)$`;`
  $X := \dfrac{V}{U}$
`until` $X^2 \leqslant -4\ln U$

△

Gęstości przedstawione szeregami

Ciekawe, że można skonstruować dokładne algorytmy eliminacji bez konieczności dokładnego obliczania docelowej gęstości. Podstawowy pomysł jest następujący. Niech p będzie funkcją proporcjonalną do gęstości docelowej, zaś q funkcją proporcjonalną do gęstości instrumentalnej (jak wiemy, nie są potrzebne stałe normujące) i $p \leqslant q$. Załóżmy, że mamy dwa ciągi funkcji, przybliżające p z dołu i z góry:

$$\underline{p_n} \leqslant p \leqslant \overline{p_n}, \qquad \underline{p_n} \to p,\ \overline{p_n} \to p \quad (n \to \infty).$$

Jeśli umiemy ewaluować funkcje $\underline{p_n}$ i $\overline{p_n}$ to możemy uruchomić algorytm eliminacji. Warunek akceptacji, $Uq(Y) \leqslant p(Y)$, jest sprawdzany w następujący sposób:

- Jeśli dla pewnego n mamy $Uq(Y) \leqslant \underline{p_n}(Y)$ to akcepujemy Y.
- Jeśli dla pewnego n mamy $Uq(Y) > \overline{p_n}(Y)$ to eliminujemy Y.
- Jeśli $\underline{p_n}(Y) < Uq(Y) \leqslant \overline{p_n}(Y)$ to zwiększamy n.

Zbieżność przybliżeń dolnych i górnych do funkcji p gwarantuje, że prędzej czy później podejmiemy decyzję (stwierdzimy, czy warunek $Uq(Y) \leqslant p(Y)$ jest spełniony, czy nie). Zauważmy, że wytarczy tutaj zbieżność punktowa ciągów funkcji.

Szczególnym przypadkiem jest „*metoda szeregów zbieżnych*". Załóżmy, że $p(x) = \sum_{i=1}^{\infty} a_i(x)$, przy czym reszty tego szeregu umiemy oszacować z góry przez znane funkcje, $\left|\sum_{i=n+1}^{\infty} a_i(x)\right| \leqslant r_{n+1}(x)$. Za dolne/górne oszacowania gęstości p mozemy przyjąć $\underline{p_n} = \sum_{i=1}^{\infty} a_n \pm r_{n+1}$.

Inny szczególny przypadek to „*metoda szeregów naprzemiennych*". Załóżmy, że $p(x) = \sum_{i=1}^{\infty}(-1)^{i+1}a_i(x)$, gdzie $a_i(x) \searrow 0$. Wtedy parzyste sumy częściowe szeregu są mniejsze od $p(x)$, zaś nieparzyste są większe od $p(x)$. Mamy więc naturalne oszacowania dolne/górne.

1.4.7 Przykład. Rozkład Kołmogorowa-Smirnowa ma gęstość

$$p(x) = 8\sum_{n=1}^{\infty}(-1)^{n+1}n^2 x\mathrm{e}^{-2n^2x^2}, \quad (x \geqslant 0).$$

△

1.5 Zadania i uzupełnienia

Zadania teoretyczne

Najpierw zauważmy, że „liczba losowa" jest w istocie tym samym co nieskończony ciąg rzutów monetą.

1.1 Zadanie. Jeśli $\varepsilon_1, \ldots, \varepsilon_n, \ldots \sim_{\text{i.i.d.}} \mathrm{Ber}(1/2)$, to znaczy $\mathbb{P}(\varepsilon_n = 1) = \mathbb{P}(\varepsilon_n = 0) = 1/2$, to zmienna losowa $U = \sum_{n=1}^{\infty} 2^{-n}\varepsilon_n$ ma rozkład jednostajny $\mathrm{U}(0,1)$.

1.2 Zadanie. Odwrotnie, jeśli zmienna losowa U ma rozkład jednostajny $\mathrm{U}(0,1)$ to kolejne cyfry $\varepsilon_1, \ldots, \varepsilon_n, \ldots$ rozwinięcia dwójkowego $U = \sum_{n=1}^{\infty} 2^{-n}\varepsilon_n$ są niezależne i każda ma rozkład $\mathrm{Ber}(1/2)$.

Jedna (idealna) „liczba losowa" jest w istocie tyle samo warta, co nieskończony ciąg niezależnych „liczb losowych".

1.3 Zadanie. Jeśli zmienna losowa U ma rozkład jednostajny $\mathrm{U}(0,1)$ to można zdefiniować ciąg funkcji $\phi_n : [0,1] \to [0,1]$ tak, że zmienne losowe $U_n = \phi_n(U)$ tworzą nieskończony ciąg $U_1, , \ldots, U_n, \ldots$ niezależnych zmiennych losowych o jednakowym rozkładzie $\mathrm{U}(0,1)$.

Wskazówka: Wykorzystaj Zadania 1.2 i 1.1.

Uwaga: Oczywiście, w realnych symulacjach nie należy „rozmanażać" pojedynczej liczby losowej, która ma skończoną, przybliżoną reprezentację zmiennoprzecinkową. Trzeba zachować zdrowy rozsądek i zadbać o to, żeby teoretycznie „dokładne" algorytmy zadowalająco działały w praktyce.

1.4 Zadanie. Piękny algorytm generowania z rozkładu normalnego, wynaleziony przez Marsaglię, jest następujący:

`repeat`

  `Gen` $V_1, V_2 \sim U(-1,1)$

$R^2 := V_1^2 + V_2^2$

`until` $R^2 < 1$

$R := \cdots ? \cdots$

$X := RV_1\,;\ Y := RV_2$

`return` (X, Y)

Uzupełnij brakującą linijkę "$R := \cdots ? \cdots$". Na wyjściu powinniśmy otrzymać dwie niezależne zmienne o rozkładzie $\mathrm{N}(0,1)$.

1.5 Zadanie. Rozważ następujący prosty przykład algorytmu eliminacji:

`repeat`

  `Gen` $X \sim U(0,1)$

  `Gen` $U \sim U(0,1)$

`until` $U < X$

`return` X

Podaj rozkład prawdopodobieństwa zmiennej X na wyjściu. Zbadaj co się stanie, jeśli przestawimy kolejność instrukcji w taki sposób:

`Gen` $U \sim U(0,1)$

`repeat`

  `Gen` $X \sim U(0,1)$

`until` $U < X$

`return` X

Podaj rozkład prawdopodobieństwa zmiennej X na wyjściu.

Ćwiczenia komputerowe

1.1 Ćwiczenie. Wypróbuj działanie generatora „liczb losowych" w R. Funkcja nazywa się `runif`:

```
> n <- 1000
> U <- runif(n)  \#  generowanie z rozkładu jednostajnego na [0,1]
> hist(U,prob=TRUE,col="gray")
```

Jak należy interpretować histogram, produkowany przez funkcję `hist`? Zwróć uwagę na znaczenie parametru `prob=TRUE`. Napisz `?hist`, żeby wezwać pomoc.

1.2 Ćwiczenie. Wygeneruj próbkę (ciąg niezależnych zmiennych losowych) z rozkładu normalnego $N(0,1)$. Funkcja nazywa się `rnorm`. Narysuj histogram. Nałóż na histogram wykres gęstości. Sugeruję użycie funkcji `curve`:

```
> n <- 100
> X <- rnorm(n)  \#  generowanie ze standardowego  rozkładu normalnego
> hist(X,prob=TRUE,col="gray")
> curve(dnorm(x),col="blue",add=TRUE)
```

Zwróć uwagę na zasadniczą różnicę pomiędzy argumentem `X` funkcji `hist` i argumentem `x` funkcji `curve`!

Zrób wykres dystrybuanty empirycznej (*empirical cumulant distribution function*) Przypomnij sobie definicję. Funkcja w R nazywa się `ecdf`. Nałóż na to wykres „prawdziwej", teoretycznej dystrybuanty. Na przykład tak:

```
> ...
> plot(ecdf(X))
> curve(pnorm(x),col="blue",add=TRUE)
```

Zorientuj się, jakie są dodatkowe parametry funkcji `rnorm`, jakie są dostępne inne funkcje związane z rozkładem normalnym, jakie inne rozkłady prawdopodobieństwa są „wbudowane" w R (na przykład zajrzyj do „Introduction to R").

1.3 Ćwiczenie (Kontynuacja)**.** Sprawdź, czy wylosowana w poprzednim ćwiczeniu próbka jest rzeczywiście zgodna z rozkładem $N(0,1)$. Użyj testu Kołmogorowa-Smirnowa:

```
> ...
> ks.test(X,pnorm)
```

Przypomnij sobie, jak jest obliczana statystyka testowa (`D`) i p-wartość (`p-value`).

Wygeneruj $m = 10000$ próbek o liczności $n = 100$ każda i zastosuj test Kolmogorowa-Smirnowa do każdej z nich. Zapamiętaj p-wartości i przeanalizuj je. Jaki jest rozkład prawdopodobieństwa p-wartości, jeśli hipoteza zerowa jest prawdziwa?

1.4 Ćwiczenie. Napisz funkcję realizującą przybliżoną generację z rozkładu $\mathrm{N}(0,1)$, opisaną w Przykładzie 1.1.3. Sugestia: wystarczy napisać

```
> rnormCTG <- function(n) { replicate(n, sum(runif(12))-6) }
```

Powtórz zabawę sugerowaną w poprzednim ćwiczeniu, używając tym razem funkcji `rnormCTG` zamiast `rnorm`. Czy porafisz wykryć niedoskonałośc przybliżonej metody?

Następujące ćwiczenie ilustruje podstawowe pojęcia statystyki Bayesowskiej i wyjaśnia istotę tak zwanej metody ABC (*Approximate Bayesian Computation*). W modelu Bayesowskim zakłada się, że obserwowana zmienna losowa X pochodzi z rozkładu o warunkowej gęstości $f(\cdot|\theta)$. Parametr θ traktuje się jako realizację zmiennej losowej ϑ o rozkładzie *a priori*, który ma gęstość $\pi(\cdot)$. Oblicza się gęstość warunkową $\pi(\theta|X = x)$, czyli rozkład *a postriori*. Analitycznie, rozkład a posteriori jest dany znanym wzorem Bayesa $\pi(\theta|X = x) \propto f(x|\theta)\pi(\theta)$. ABC jest bezpośrednim zastosowaniem metody eliminacji (Stwierdzenie 1.4.1) do próbkowania z rozkładu *a posteriori*. Generuje się parę (ϑ, X) w dwóch krokach: $\vartheta \sim \pi(\cdot)$ a następnie $X \sim f(\cdot|\vartheta)$. Jeśli $X = x$, to $\vartheta \sim \pi(\cdot|X = x)$, w przeciwnym razie parę się eliminuje.

1.5 Ćwiczenie (ABC w modelu Bin/Beta)**.** Rozważ następujący 2-etapowy schemat losowania: najpierw wylosuj zmienną losową $\vartheta \sim \mathrm{U}(0,1)$ a następnie $X \sim \mathrm{Bin}(n, \vartheta)$. Wybierz np. $n = 9$.

- Zbadaj eksperymentalnie rozkład brzegowy X. Oblicz ten rozkład analitycznie.
- Zbadaj eksperymentalnie rozkład *a posteriori* zmiennej ϑ przy $X = 3$, metodą ABC. Oblicz ten rozkład analitycznie. Porównaj.
- Wypróbuj losowanie 2-etapowe w odwotnym porządku: wygeneruj X z rozkładu brzegowego, a następnie ϑ z rozkładu *a posteriori*. Sprawdź, że otrzymana zmienna ϑ ma brzegowy rozkład *a priori*.

Następne ćwiczenia ilustrują różne typy zbieżności zmiennych losowych. Zwróćmy uwagę na zasadniczą różnicę między zbieżnością „mocną” i „słabą”. Zbieżność mocna (inaczej: prawie na pewno, z prawdopodobieństwem 1) jest własnością *trajektorii*, czyli wylosowanego ciągu zmiennych losowych. Jeśli $X_n \to_{\text{p.n.}} X$ to rysując wykres ciągu $X_1, X_2, \ldots, X_n, \ldots$ można zobaczyć, że trajektorie dążą do wartości granicznych. Zbieżność słaba jest w istocie zbieżnością *rozkładów prawdopodobieństwa*. Żeby „zobaczyć” rozkład prawdopodobieństwa, trzeba na ogół powtórzyć całe doświadczenie symulacyjne wiele razy (powiedzmy, m razy) i analizować rozkład empiryczny (narysować histogram, obliczyć momenty itp.). Jeśli badamy rozkład graniczny, pojedyncze doświadczenie polega na wylosowaniu zmiennej X_n dla „dostatecznie dużego n”.

1.6 Ćwiczenie (Mocne Prawo Wielkich Liczb)**.** Wygeneruj dużą próbkę $X_1, \ldots, X_n, \ldots$ z jakiegoś rozkładu o skończonej wartości oczekiwanej $\mu = \mathbb{E}X_1$. Narysuj wykres S_n/n w funkcji n, gdzie $S_n = X_1 + \cdots + X_n$ (możesz użyć funkcji `cumsum`). Powtórz doświadczenie kilka razy i naszkicuj kilka trajektorii na tym samym rysunku. MPWL jest klasycznym przykładem zbieżności p.n. ciągu zmiennych losowych do liczby ($S_n/n \to_{\text{p.n.}} \mu$).

1.7 Ćwiczenie (Urna Polya)**.** Rozważmy ciąg zmiennych losowych X_n o wartościach w zbiorze $\{0, 1\}$ zdefiniowany rekurencyjnie:

$$\mathbb{P}(X_1 = 1) = \frac{\alpha}{\alpha+\beta}, \quad \mathbb{P}(X_1 = 0) = \frac{\beta}{\alpha+\beta}$$
$$\mathbb{P}(X_{n+1} = 1 | X_1, \ldots, X_n) = \frac{\alpha + S_n}{\alpha+\beta+n}, \quad \mathbb{P}(X_{n+1} = 0 | X_1, \ldots, X_n) = \frac{\beta + n - S_n}{\alpha+\beta+n},$$

gdzie

$$S_n = X_1 + \cdots + X_n.$$

Zinterpretuj tę regułę w terminach ciągu „losowań kul z urny".

- Przeprowadź symulację (dość długiego) ciągu $X_1, X_2, \ldots$. Narysuj wykres S_n/n w funkcji n. Powtórz doświadczenie kilka razy i naszkicuj kilka trajektorii na tym samym rysunku (powiedzmy, dla $\alpha = \beta = 1$).

- Udowodnij, że ciąg

  $$M_n = \frac{\alpha + S_n}{\alpha+\beta+n}$$

  jest martyngałem. Wywnioskuj stąd zbieżność $M_n \to_{\text{p.n.}} X$ i $S_n/n \to_{\text{p.n.}} X$ (zbieżność do zmiennej losowej X). Porównaj z zachowaniem trajektorii na rysunku.

- Zbadaj doświadczalnie rozkład prawdopodobieństwa granicznej zmiennej losowej X (powtórz doświadczenie m razy, zapamiętując wartości S_n/n dla „dużego" n).

- Sprawdź, że $X \sim \text{Beta}(\alpha, \beta)$. Zrób kilka doświadczeń dla różnych wartości parametrów α, β.

1.8 Ćwiczenie (Prawo Arcusa Sinusa)**.** Niech $X_1, \ldots, X_n$ będą niezależnymi zmiennymi losowymi z wartością oczekiwaną $\mathbb{E}X_i = 0$. Niech $S_i = X_1 + \cdots + X_i$ dla $i = 1, \ldots, n$). Określmy

$$T_n = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(S_i > 0).$$

Zinterpretuj T_n w terminach „ciągu gier" z wypłatami X_i. Zrób wykres trajektorii $(0, S_1, S_2, \ldots, S_n)$ i oblicz odpowiednią wartość T_n. Zbadaj doświadczalnie rozkład prawdopodobieństwa zmiennej losowej T_n dla dużego n. Sprawdź doświadczalnie, że dla $n \to \infty$, $T_n \to \text{Beta}(1/2, 1/2)$ (nazwa twierdzenia pochodzi od dystrybuanty tego rozkładu beta).

1.6 Rozkłady wielowymiarowe

Ogólne metody generowania zmiennych losowych są z powodzeniem stosowane również do zmiennych losowych wielowymiarowych. W szczególności, dotyczy to metody *eliminacji* i *przekształceń*. Wyjątek stanowi „najbardziej ogólna" metoda *odwracania dystrybuanty*, która nie ma naturalnego odpowiednika dla wymiaru większego niż 1.

Metoda rozkładów warunkowych

Jest to właściwie jedyna metoda „w zasadniczy sposób wielowymiarowa". Opiera się na przedstawieniu gęstości łącznej zmiennych losowych $X_1, \ldots, X_d$ jako iloczynu gęstości brzegowej i gęstości warunkowych (wzór łańcuchowy):

$$p(x_1, x_2, \ldots, x_d) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \cdots p(x_d|x_1, \ldots, x_{d-1}).$$

Wynika stąd następujący algorytm:

`Gen` $X_1 \sim p(\cdot)$
`for` $i := 2$ `to` d `do`
  `Gen` $X_i \sim p(\cdot|X_1, \ldots, X_{i-1})$

1.6.1 Przykład (Wielowymiarowy rozkład normalny)**.** Ograniczymy się do pary zmiennych losowych X_1, X_2 pochodzących z rozkładu dwuwymiarowego $\mathrm{N}(0, 0, \sigma_1^2, \sigma_2^2, \rho)$ o gęstości

$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\Big(\frac{x_1^2}{\sigma_1^2} - 2\rho\frac{x_1x_2}{\sigma_1\sigma_2} + \frac{x_2^2}{\sigma_2^2}\Big)\right].$$

Jak wiadomo (można to sprawdzić elementarnym rachunkiem),

$$p(x_1) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left[-\frac{x_1^2}{2\sigma_1^2}\right],$$

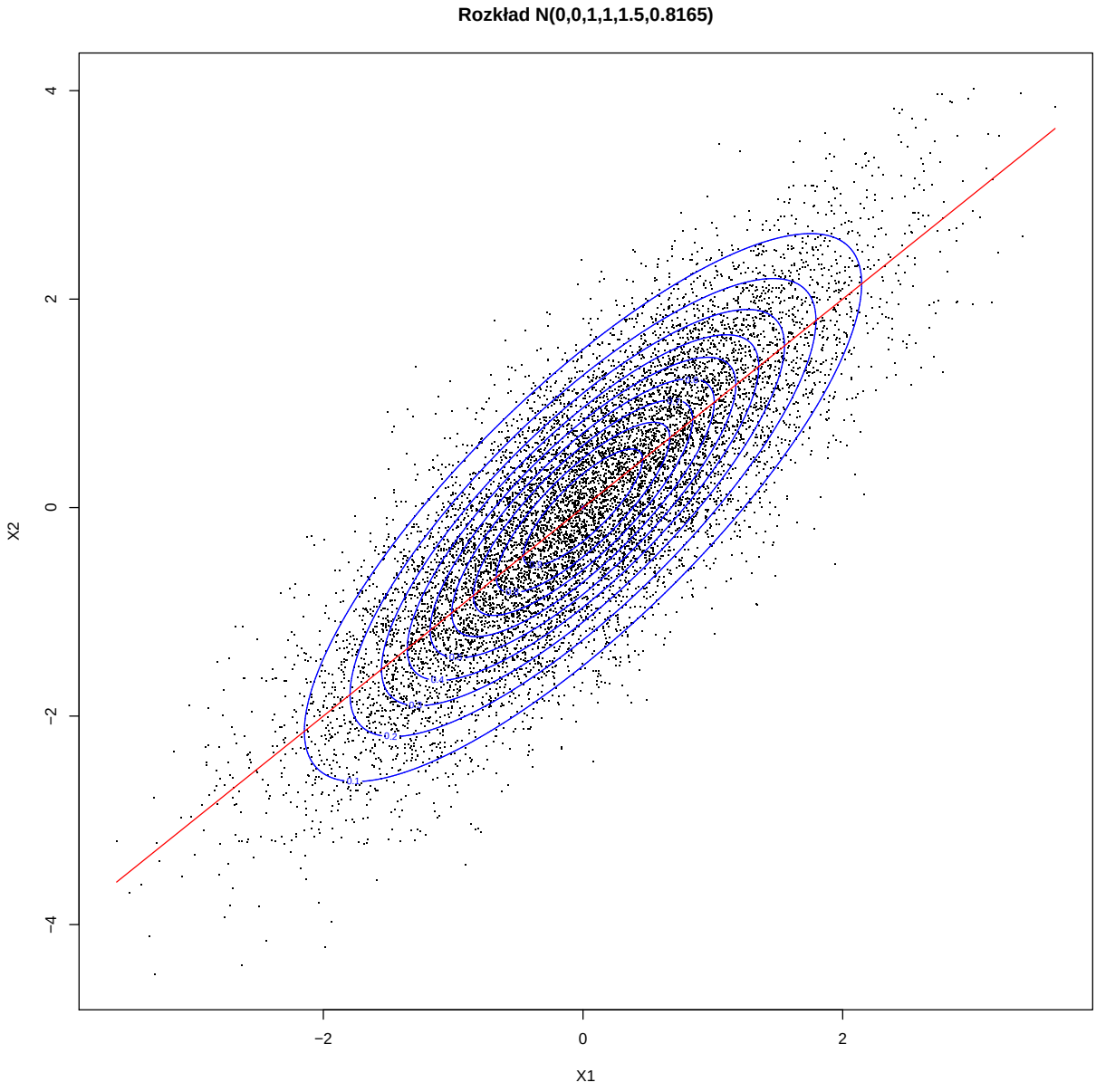
$$p(x_2|x_1) = \frac{1}{\sqrt{2\pi}\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2\sigma_2^2(1-\rho^2)}\Big(x_2 - \rho\frac{\sigma_2}{\sigma_1}x_1\Big)^2\right].$$

To znaczy, że $\mathrm{N}(0, \sigma_1^2)$ jest rozkładem brzegowym X_1 oraz

$$\mathrm{N}\left(\rho\frac{\sigma_2}{\sigma_1}x_1, \sigma_2^2(1-\rho^2)\right)$$

jest rozkładem warunkowym X_2 dla $X_1 = x_1$. Algorytm jest więc następujący.

`Gen` $X_1, X_2 \sim \mathrm{N}(0,1)$
$X_1 := \sigma_1 X_1$
$X_2 := \rho(\sigma_2/\sigma_1)X_1 + \sigma_2\sqrt{1-\rho^2}X_2$



Rysunek 1.3: Próbka z dwuwymiarowego rozkładu normalnego, poziomice gęstości i funkcja regresji $x_2 = \mathbb{E}(X_2|X_1 = x_1)$.

Efekt działania tego algorytmu widać na Rysunku 1.3. W tym konkretnym przykładzie X_1 ma rozkład brzegowy $\mathrm{N}(0,1)$, zaś X_2 ma rozkład warunkowy $\mathrm{N}(X_1, 0.5)$. Zauważmy, że $\mathrm{Var} X_2 = 1.5$ i $\mathrm{Cov}(X_1, X_2) = 1$. Wykresem funkcji regresji $\mathbb{E}(X_2|X_1 = x_1)$ jest prosta $x_2 = x_1$, przedstawiona na wykresie. Pokazane są też poziomice gęstości (elipsy). Warto zwrócić uwagę, że funkcja regresji nie pokrywa się ze wspólną osią tych elips. Dlaczego? Jak obliczyć oś?

Uogólnienie na przypadek rozkładu normalnego $\mathrm{N}(\mu_1, \mu_2, \sigma_1^2, \sigma_2^2, \rho)$, z niezerowymi średnimi, jest banalne. Uogólnienie na wyższe wymiary też nie jest skomplikowane. Okazuje się jednak, że metoda rozkładów warunkowych dla rozkładów normalnych prowadzi do algorytmu identycznego jak otrzymany metodą przekształceń, patrz Przykład 1.6.3. △

Metoda kompozycji i marginalizacja

Metoda kompozycji jest niezwykle prostą techniką generowania zmiennych losowych. Załóżmy, że docelowy rozkład jest mieszanką rozkładów prawdopodobieństwa, czyli jego gęstość jest kombinacją wypukłą postaci

$$p(x) = \sum_{i=1}^{k} \alpha_i p_i(x), \qquad \left(\alpha_i \geqslant 0, \sum_{i=1}^{k} \alpha_i = 1 \right).$$

Jeśli umiemy losować z każdej gęstości p_i to możemy uruchomić dwuetapowe losowanie:

Gen $I \sim \alpha(\cdot)$; *{ to znaczy* $\mathbb{P}(I = i) = \alpha_i$ *}*
Gen $X \sim p_I$; *{ jeśli* $I = i$ *to uruchamiamy generator rozkładu* p_i *}*
return X

Jasne, że na wyjściu mamy $X \sim p$. W istocie jest to szczególny przypadek metody rozkładów warunkowych. Istotnie, kompozycja polega na wygenerowaniu pary (I, X) z rozkładu łącznego tak, aby otrzymać docelowy rozkład brzegowy zmiennej X.

1.6.2 Przykład (Rozkład Laplace'a)**.** Rozkład Laplace'a (podwójny rozkład wykładniczy) ma gęstość

$$p(x) = \frac{1}{2\lambda} \mathrm{e}^{-\lambda|x|}.$$

Można go „skomponować" z dwóch połówek rozkładu wykładniczego:

Gen $W \sim \mathrm{Ex}(\lambda)$; *{ generujemy z rozkładu wykładniczego }*
Gen $U \sim \mathrm{U}(0,1)$;
if $U < 1/2$ **then** $X := W$ **else** $X := -W$ *{ losowo zmieniamy znak }*
return X

△

Przykłady zastosowania metody rozkładów warunkowych i marginalizacji do generowania z „trudnych" jednowymiarowych rozkładów ciągłych są podane w Zadaniach 1.6 i 1.7.

Poniżej przedstawiam metody symulacji dla kilku typów rozkładów wielowymiarowych, które mi się wydają szczególnie ważne i ciekawe.

Rozkłady sferyczne i eliptyczne

Dwuwymiarowy rozkład normalny, omówiony w 1.6.1, ma poziomice gęstości będące elipsami. W wielu wymiarach, poziomice gęstości normalnej są elipsoidami. Wielowymiarowe rozkłady normalne należą do rodziny rozkładów eliptycznych. W szczególnym przypadku, gdy elipsoidy są sferami, mówimy o rozkładach sferycznych.

1.6.3 Przykład (Wielowymiarowy rozkład normalny)**.** Rozważmy niezależne zmienne losowe $Z_1, \ldots, Z_d \sim \mathrm{N}(0,1)$. Wektor $Z = (Z_1, \ldots, Z_d)^\top$ ma d-wymiarowy rozkład normalny $\mathrm{N}(0, I)$ o gęstości

$$p_Z(z) = (2\pi)^{-d/2} \exp\left[-\frac{1}{2} z^\top z\right].$$

Jeżeli teraz A jest nieosobliwą macierzą $(d \times d)$ to przekształcenie $z \mapsto x = \mathrm{A}z$ jest dyfeomorfizmem z jakobianem $\det \mathrm{A}$. Z Twierdzenia 1.3.1 wynika, że wektor losowy $X = \mathrm{A}Z$ ma rozkład normalny o gęstości

$$p_X(x) = (2\pi)^{-d/2} (\det \mathrm{A})^{-1/2} \exp\left[-\frac{1}{2} x^\top \Sigma^{-1} x\right],$$

gdzie $\Sigma = \mathrm{A}\mathrm{A}^\top$. Innymi słowy, $X \sim \mathrm{N}(0, \Sigma)$. Algorytm generacji jest oczywisty:

`Gen` $Z \sim \mathrm{N}(0, I)$
$X := \mathrm{A}Z$

Jeśli dana jest macierz kowariancji Σ wektora X, to przed uruchomieniem algorytmu trzeba znaleźć taką macierz A, żeby $\Sigma = \mathrm{A}\mathrm{A}^\top$. Istnieje wiele takich macierzy, ale najlepiej skorzystać z rozkładu Choleskiego i wybrać macierz trójkątną. △

Rozważmy teraz $\mathrm{U}(B^d)$, **rozkład jednostajny na kuli**

$$B^d = \{x \in \mathbb{R}^d : |x|^2 \leqslant 1\}$$

i $\mathrm{U}(S^{d-1})$, **rozkład jednostajny na sferze**

$$S^{d-1} = \{x \in \mathbb{R}^d : |x|^2 = 1\}.$$

Oczywiście, $|x|$ oznacza normę euklidesową, $|x| = (x_1^2 + \cdots + x_d^2)^{1/2} = (x^\top x)^{1/2}$. Rozkład $\mathrm{U}(B^d)$ ma po prostu stałą gęstość względem d-wymiarowej miary Lebesgue'a na kuli. Rozkład $\mathrm{U}(S^{d-1})$ ma stałą gęstość względem $(d-1)$-wymiarowej miary „powierzchniowej" na sferze. Oba te rozkłady są niezmiennicze względem *obrotów* (liniowych przekształceń ortogonalnych) $\mathbb{R}^d$. Takie rozkłady nazywamy *sferycznie symetrycznymi* lub krócej: **sferycznymi**. Zauważmy, że zmienną losową o rozkładzie $\mathrm{U}(S^{d-1})$ możemy interpretować jako losowo wybrany kierunek w przestrzeni $d-1$-wymiarowej. Algorytmy „poszukiwań losowych" często wymagają generowania takich losowych kierunków.

Rozkłady jednostajne na kuli i sferze są blisko ze sobą związane.

- Jeśli $V = (V_1, \ldots, V_d) \sim \mathrm{U}(B^d)$ i $R = |V|$ to

$$Y = \frac{V}{R} = \left(\frac{V_1}{R}, \ldots, \frac{V_d}{R}\right) \sim \mathrm{U}(S^{d-1}).$$

  Łatwo też zauważyć, że R jest zmienną losową o rozkładzie $\mathbb{P}(R \leqslant r) = r^d$ niezależną od Y.

- Jeśli $Y \sim \mathrm{U}(S^{d-1})$ i R jest niezależną zmienną losową o rozkładzie $\mathbb{P}(R \leqslant r) = r^d$ to $V = RY = (RY_1, \ldots, RV_d) \sim \mathrm{U}(B^d)$.

Zmienną R łatwo wygenerować metodą odwracania dystrybuanty.

Najprostszy wydaje się algorytm eliminacji:

`repeat`
  `Gen` $V_1, \ldots, V_d \sim \mathrm{U}(-1, 1)$
`until` $R^2 = V_1^2 + \ldots + V_d^2 \leqslant 1$

Na wyjściu otrzymujemy, zgodnie z żądaniem

$$V = (V_1, \ldots, V_d) \sim \mathrm{U}(B^d).$$

W istocie, dokładnie ta metoda, dla $d = 2$, jest częścią algorytmu biegunowego Marsaglii (Zadanie 1.4). Problem w tym, że w wyższych wymiarach efektywnosc eliminacji gwałtownie maleje. Prawdopodobieństwo akceptacji jest równe stosunkowi „objętości" kuli B^d do kostki $[-1, 1]^d$. Ze znanego wzoru na objętość kuli d-wymiarowej wynika, że

$$\frac{|B^d|}{2^d} = \frac{2\pi^{d/2}}{d\Gamma(d/2)} \cdot \frac{1}{2^d} = \frac{\pi^{d/2}}{d2^{d-1}\Gamma(d/2)} \longrightarrow_{d\to\infty} 0.$$

Zbieżność do zera jest bardzo szybka. Dla dużego d kula jest znikomą częścią opisanej na niej kostki.

Inna metoda, którą z powodzeniem stosuje się dla $d = 2$ jest związana ze współrzędnymi biegunowymi:

`Gen` $\Phi \sim \mathrm{U}(0, 2\pi)$;
$Y_1 := \cos\Phi$; $Y_2 := \sin\Phi$;
`Gen` U; $R := \sqrt{U}$;
$V_1 := Y_1 \cdot R$; $V_2 := Y_2 \cdot R$;

Na wyjściu $(Y_1, Y_2) \sim \mathrm{U}(S^1)$ i $(V_1, V_2) \sim \mathrm{U}(B^2)$. Jest to część algorytmu Boxa-Müllera. Uogólnienie na przypadek $d > 2$ nie jest jednak ani proste, ani efektywne. Mechaniczne zastąpienie, współrzędnych biegunowych przez współrzędne sferyczne (dla, powiedzmy $d = 3$) prowadzi do *niepoprawnych wyników* (Ćwiczenie 1.10).

Poprawny i efektywny algorytm jest podany poniżej.

`Gen` $Z_1, \ldots, Z_d \sim_{\text{i.i.d.}} \mathrm{N}(0, 1)$;
$R := (Z_1^2 + \cdots + Z_d^2)^{1/2}$;
$Y_1 := Z_1/R, \ldots, Y_d := Z_d/R$;
`Gen` U; $R := U^{1/d}$;
$V_1 := Y_1 \cdot R, \ldots, V_d := Y_d \cdot R$;

Na wyjściu $Y \sim \mathrm{U}(S^{d-1})$ i $V \sim \mathrm{U}(B^d)$. Jak widać, algorytm polega na normowaniu punktów wylosowanych ze sferycznie symetrycznego rozkładu normalnego.

1.6.4 Przykład (Wielowymiarowe rozkłady Studenta)**.** Niech $Z = (Z_1, \ldots, Z_d)^\top$ będzie wektorem losowym o rozkładzie $\mathrm{N}(0, I)$, zaś R^2 – niezależną zmienną losową o rozkładzie $\chi^2(n)$. Wektor

$$(Y_1, \ldots, Y_d)^\top = \frac{(Z_1, \ldots, Z_d)^\top}{\sqrt{R^2/n}}$$

ma, z definicji, *Sferyczny rozkład t-Studenta z n stopniami swobody*. Gęstość tego rozkładu (z dokładnością do stałej normującej) jest równa

$$p(y) = p(y_1, \ldots, y_d) \propto \left[1 + \frac{1}{n}\Big(\sum y_i^2\Big)\right]^{-(n+d)/2} = \left[1 + \frac{1}{n}|y|^2\right]^{-(n+d)/2}.$$

W przypadku jednowymiarowym, a więc przyjmując $d = 1$, otrzymujemy dobrze znane rozklady t-Studenta z n stopniami swobody o gęstości

$$p(y) \propto \frac{1}{(1 + y^2/n)^{(n+1)/2}}$$

W szczególnym przypadku, biorąc za liczbę stopni swobody $n = 1$, otrzymujemy rozkłady Cauchy'ego. Na przykład, dwuwymiarowy rozkład Cauchy'ego ma taką gęstość:

$$p(y_1, y_2) \propto \frac{1}{(1 + y_1^2 + y_2^2)^{3/2}}.$$

△

Użytecznym uogólnieniem rozkładów sferycznych są rozkłady eliptyczne. Są one określone w następujący sposób. Niech Σ będzie macierzą symetryczną i nieosobliwą. Nazwijmy *uogólnionym obrotem* przekształcenie liniowe, które zachowuje uogólnioną normę $|x|_{\Sigma^{-1}} = (x^\top \Sigma^{-1} x)^{1/2}$. Rozkład jest z definicji *eliptycznie konturowany* lub krócej **eliptyczny**, gdy jest niezmienniczy względem uogólnionych obrotów (dla ustalonej macierzy Σ).

Rozklady Dirichleta

1.6.5 Definicja. *Mówimy, że n-wymiarowa zmienna losowa X ma rozkład **Dirichleta**,*

$$X = (X_1, \ldots, X_n) \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_n)$$

jesli $X_1 + \cdots + X_n = 1$ i zmienne $X_1, \ldots, X_{n-1}$ mają gęstość

$$p(x_1, \ldots, x_{n-1}) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1) \ldots \Gamma(\alpha_n)} x_1^{\alpha_1 - 1} \ldots x_{n-1}^{\alpha_{n-1} - 1} (1 - x_1 - \ldots x_{n-1})^{\alpha_n - 1}.$$

Parametry $\alpha_1, \ldots, \alpha_n$ mogą być dowolnymi liczbami dodatnimi.

Uwaga. Rozkłady Dirichleta dla $n = 2$ są to w istocie rozkłady beta:

$(X_1, X_2) \sim \mathrm{Dir}(\alpha_1, \alpha_2)$ wtedy i tylko wtedy, gdy $X_1 \sim \mathrm{Beta}(\alpha_1, \alpha_2)$ i $X_2 = 1 - X_1$.

1.6.6 Wniosek. *Jeśli $U_1, \ldots, U_{n-1}$ są niezależnymi zmiennymi o jednakowym rozkładzie jednostajnym* $\mathrm{U}(0, 1)$ *i*

$$U_{1:n} < \ldots < U_{n-1:n-1}$$

oznaczają statystyki pozycyjne, to spacje

$$X_i = U_{i:n} - U_{i-1:n}, \quad X_n = 1 - U_{n-1:n}$$

mają rozklad $\mathrm{Dir}(1, \ldots, 1)$.

1.6.7 Twierdzenie. *Jeśli $Y_1, \ldots, Y_n$ są niezależnymi zmiennymi losowymi o rozkładach gamma,*

$$Y_i \sim \mathrm{Gamma}(\alpha_i)$$

i $S = Y_1 + \cdots + Y_n$ to

$$(X_1, \ldots, X_n) = \left(\frac{Y_1}{S}, \ldots, \frac{Y_n}{S} \right) \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_n).$$

Wektor losowy X jest niezależny od S.

Dowód. Obliczymy łączną gęstość zmiennych losowych $S, X_1, \ldots, X_{n-1}$. Ze wzoru na przekształcenie gęstości wynika, że

$$\begin{aligned} &p_{S,X_1,\ldots,X_{n-1}}(s, x_1, \ldots, x_{n-1}) = p_{Y_1,\ldots,Y_n}(x_1 s, \ldots, x_n s) \\ &\propto (x_1 s)^{\alpha_1 - 1} \mathrm{e}^{-x_1 s} \ldots (x_n s)^{\alpha_n - 1} \mathrm{e}^{-x_n s} \left| \frac{\partial(y_1, \ldots, y_n)}{\partial(s, x_1, \ldots x_{n-1})} \right| \\ &\propto x_1^{\alpha_1 - 1} \ldots x_n^{\alpha_n - 1} s^{\alpha_1 + \cdots + \alpha_n - 1} \mathrm{e}^{-s}, \end{aligned}$$

ponieważ jakobian przekształcenia odwrotnego jest równy s^{n-1}. Wystarczy teraz zauważyć, że

$$x_1^{\alpha_1 - 1} \cdots x_n^{\alpha_n - 1} \propto \mathrm{Dir},$$

$$s^{\alpha_1 + \cdots + \alpha_n - 1} \mathrm{e}^{-s} \propto \mathrm{Gamma}.$$

□

1.6.8 Wniosek. *Dla niezależnych zmiennych losowych o jednakowym rozkładzie wykładniczym,*

$$Y_1, \ldots, Y_n \sim \mathrm{Ex}(1),$$

jeśli $S = Y_1 + \cdots + Y_n$ *to*

$$(X_1, \ldots, X_n) = \left(\frac{Y_1}{S}, \ldots, \frac{Y_n}{S} \right) \sim \mathrm{Dir}(1, \ldots, 1)$$

Z 1.6.6 i 1.6.8 wynika, że następujące dwa algorytmy:

`Gen` $U_1, \ldots, U_{n-1}$`;`
`Sort` $(U_1, \ldots, U_{n-1})$; $U_0 = 0$; $U_n = 1$`;`
`for i:=1 to` n `do` $X_i := U_i - U_{i-1}$

oraz

`Gen` $Y_1, \ldots, Y_n \sim \mathrm{Ex}(1)$
$S := Y_1 + \cdots + Y_n$
`for i:=1 to` n `do` $X_i := Y_i / S$

dają te same wyniki.

Wiele ciekawych własności rozkładów Dirichleta wynika niemal natychmiast z 1.6.7 (choć nie tak łatwo wyprowadzić je posługując się wzorem na gęstość). Mam na myśli przede wszystkim zasadniczą własność „grupowania zmiennych".

1.6.9 Wniosek. *Rozważmy rozbicie zbioru indeksów na sumę rozłącznych podzbiorów:*

$$\{1,\dots,n\} = \bigcup_{j=1}^{k} I_j.$$

Jeżeli $(X_1,\dots,X_n) \sim \mathrm{Dir}(\alpha_1,\dots,\alpha_n)$ *i rozważymy „zgrupowane zmienne”*

$$S_j = \sum_{i\in I_j} X_i,$$

to wektor tych zmiennych ma też rozkład Dirichleta,

$$(S_1,\dots,S_k) \sim \mathrm{Dir}(\beta_1,\dots,\beta_k),$$

gdzie

$$\beta_j = \sum_{i\in I_j} \alpha_i.$$

Co więcej, każdy z wektorów $(X_i/S_j)_{i\in I_j}$ *ma rozkład Dirichleta* $\mathrm{Dir}(\alpha_i)_{i\in I_j}$ *i wszystkie te wektory są niezależne od* $(S_1,\dots,S_k)$.

1.6.10 Przykład. Wniosek 1.6.9 razem z 1.6.6 pozwala szybko generować wybrane statystyki pozycyjne. Na przykład łączny rozkład dwóch statystyk pozycyjnych z rozkładu jednostajnego jest wyznaczony przez rozkład trzech „zgrupowanych spacji”:

$$(U_{k:n-1},\ U_{l:n-1} - U_{k:n-1},\ 1 - U_{l:n-1}) \sim \mathrm{Dir}(k, l-k, n-l)$$

△

1.6.11 Przykład (Rozkład dwumianowy)**.** Aby wygenerować zmienną o rozkładzie dwumianowym $\mathrm{Bin}(n,p)$ wystarczy rozpoznać między którymi statystykami pozycyjnymi z rozkładu $\mathrm{U}(0,1)$ leży liczba p. Nie musimy w tym celu generować *wszystkich* staystyk pozycyjnych, możemy wybierać „najbardziej prawdopodobne”. W połączeniu 1.6.10 daje to następujący algorytm.

$k := n;\ \theta := p;\ X := 0;$

```
repeat
  i := ⌊1 + kθ⌋;
  Gen V ~ Beta(i, k + 1 − i);
  if θ < V then
    begin θ := θ/V; k := i − 1 end
                  else
    begin X := X + i; θ = (θ − V)/(1 − V); k := k − i end

until k = 0
```

△

Najprostsza metoda generowania zmiennych o rozkładzie Dirichleta opiera sie bezpośrednio na Twierdzeniu 1.6.7. Inna metoda wykorzystuje następujący fakt, który jest w istocie szczególnym przypadkiem „reguły grupowania" 1.6.9.

1.6.12 Wniosek. *Jeśli* $(X_1, \ldots, X_n) \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_n)$ *i* $S_k = X_1 + \cdots + X_k$ *dla* $k = 1, \ldots, n$, *to zmienne*

$$B_1 = \frac{X_1}{X_1 + \cdots + X_n}, B_2 = \frac{X_2}{X_2 + \cdots + X_n}, \ldots, B_{n-1} = \frac{X_{n-1}}{X_{n-1} + X_n}$$

są niezależne i

$$B_k \sim \mathrm{Beta}(\alpha_k, \alpha_{k+1} + \cdots + \alpha_n).$$

Odwrotnie, jeśli zmienne $B_1, \ldots, B_n$ *są niezależne i mają odpowiednie rozkłady beta (wzór powyżej), to wektor* $(X_1, \ldots, X_n)$ *ma rozkład Dirichleta.*

Oczywiście, jeśli wygenerujemy niezależne zmienne $B_1, \ldots, B_n$ o rozkładach beta, to zmienne $X_1, \ldots, X_n$ łatwo „odzyskać" przy pomocy wzorów:

$$\begin{aligned}
X_1 &= B_1 \\
X_2 &= (1 - B_1)B_2 \\
X_3 &= (1 - B_1)(1 - B_2)B_3 \\
&\cdots\cdots \\
X_{n-1} &= (1 - B_1) \cdots (1 - B_{n-2})B_{n-1} \\
X_n &= (1 - B_1) \cdots (1 - B_{n-2})(1 - B_{n-1})
\end{aligned}$$

Powyższe równania określają algorytm generowania zmiennych o rozkładzie $\mathrm{Dir}(\alpha_1, \ldots, \alpha_n)$.

1.7 Rozkłady dyskretne

Jak wylosować zmienną losową I o rozkładzie $\mathbb{P}(I = i) = p_i$ mając dane $p_1, p_2, \ldots$? Metoda odwracania dystrybuanty w przypadku rozkładów dyskretnych (Przykład 1.2.5) przyjmuje następującą postać. Obliczamy $s_1, s_2, \ldots$, gdzie $s_k = \sum_{i=1}^{k} p_i$. Losujemy $U \sim \mathrm{U}(0, 1)$ i szukamy przedziału $]s_{I-1}, s_I]$ w którym leży U.

1.7.1 Przykład (Algorytm prymitywny)**.** W zasadzie można to zrobić tak:

```
Gen U, I := 1
  while s_I ⩽ U do I := I + 1;
return I
```

△

Problemem jest mała efektywność tego algorytmu. Jedno z możliwych ulepszeń polega na bardziej „inteligentnym" lokalizowaniu przedziału $]s_{I-1}, s_I] \ni U$, na przykład metodą bisekcji.

Przedstawię teraz piękną metodę opartą na innym pomyśle.

1.7.2 Przykład (Metoda „Alias")**.** Przypuśćmy, że mamy dwa ciągi liczb: $q_1, \ldots, q_m$, gdzie $0 < q_i < 1$ oraz $a_1, \ldots, a_m$, gdzie $a_i \in \{1, \ldots, m\}$ (to są owe „aliasy"). Rozpatrzmy taki algorytm:

`Gen` $K \sim \mathrm{U}\{1, \ldots, m\}$`;`
`Gen` U`;`
`if` $U < q_K$ `then` $I := K$ `else` $I := a_K$`;`
`return` I

Jest jasne, że

$$\mathbb{P}(I = i) = \frac{1}{m}\left(q_i + \sum_{j:a_j=i}(1 - q_j)\right).$$

Jeśli mamy zadany rozkład prawdopodobieństwa $(p_1, \ldots, p_m)$ i chcemy, żeby $\mathbb{P}(I = i) = p_i$, to musimy dobrać odpowiednio q_i i a_i. Można zawsze tak zrobić, i to na wiele różnych sposobów. Opracowanie algorytmu dobierania wektorów q i a do zadanego p pozostawiamy jako ćwiczenie. △

Metoda „Alias" ma również swój odpowiednik dla rozkładów ciągłych. Prosty przykład pokazany jest w Zadaniu 1.8.

Zadanie 1.9 podaje jeszcze inną, ogólną metodę losowania z rozkładu dyskretnego.

Przedstawimy teraz kilka metod losowania prostych „obiektów kombinatorycznych".

Pobieranie próbki bez zwracania

Spośród r obiektów chcemy wybrać losowo n tak, aby każdy z $\binom{r}{n}$ podzbiorów miał jednakowe prawdopodobieństwo. Oczywiście, można losować tak, jak ze zwracaniem, tylko odrzucać elementy wylosowane powtórnie.

Algorytm I. W tablicy $c(1), \ldots, c(r)$ zaznaczamy, które elementy zostały wybrane.

```
for i := 1 to r do c(i) := false;
i := 0
repeat
  repeat Gen K ~ U{1,...,r} until c(K) = false;
  c(K) := true; i := i + 1;
until i = n
```

Pewnym ulepszeniem tej prymitywnej metody jest następujący algorytm.

Algorytm II. Tablica $c(1), \ldots, c(r)$ ma takie samo znaczenie jak w poprzednim przykładzie. Będziemy teraz „przeglądać" elementy $1, \ldots, r$ po kolei, decydując o zaliczeniu do próbki kolejnego elementu zgodnie z odpowiednim prawdopodobieństwem *warunkowym*. Niech i oznacza liczbę wybranych, zaś $t-1$ - liczbę przejrzanych poprzednio elementów.

```
for i := 1 to r c(i) := false;
t := 1; i := 0
repeat
  Gen U;
  if U ⩽ (n - i)/(r - (t - 1)) then
    begin c(t) := true; i := i + 1 end;
  t := t + 1;
until i = n
```

Algorytm III. Tablica $s(1), \ldots, s(n)$ będzie teraz zawierała *numery* wybieranych elementów. Podobnie jak poprzednio, i – oznacza liczbę elementów wybranych, $t-1$ – liczbę przejrzanych.

```
for i := 1 to n do s(i) := i;
for t := n + 1 to r do
  begin
  Gen U;
  if U ⩽ n/t then
    begin Gen K ~ U{1,...,n}; s(K) := t end
  end
```

Uzasadnienie poprawności tego algorytmu jest rekurencyjne. Załóżmy, że przed t-tym losowaniem każda próbka wybrana ze zbioru $\{1,\ldots,t-1\}$ ma prawdopodobieństwo

$$\binom{t-1}{n}^{-1} = \frac{n!}{(t-1)\cdots(t-n)}.$$

W kroku t ta próbka „przeżywa" czyli pozostaje bez zmiany z prawdopodobieństwem $1-n/t$ (jest to prawdopodobieństwo, że próbka wylosowana ze zbioru $\{1,\ldots,t\}$ nie zawiera elementu t. Zatem po kroku t każda próbka nie zawierająca elementu t ma prawdopodobieństwo

$$\frac{n!}{(t-1)\cdots(t-n)} \cdot \frac{t-n}{t} = \binom{t}{n}^{-1}.$$

Z prawdopodobieństwem n/t „podmieniamy" jeden z elementów próbki (losowo wybrany) na element t. Sprawdzenie, że każda probka zawierająca element t ma po t-tym losowaniu jednakowe prawdopodobieństwo – pozostawiam jako ćwiczenie.

Permutacje losowe

Przez permutację losową rozumiemy uporządkowanie n elementów wygenerowane zgodnie z rozkładem jednostajnym na przestrzeni wszystkich $n!$ możliwych uporządkowań. Permutację liczb $1,\ldots,n$ zapiszemy w tablicy $\sigma(1),\ldots,\sigma(n)$. Łatwo sprawdzić poprawność następującego algorytmu.

```
for i := 1 to n do σ(i) := i;
for i := 1 to n − 1 do
  begin
  Gen J ∼ U{i, i + 1, ..., n};
  Swap(σ(i), σ(J))
  end
```

Funkcja `Swap` zamienia miejscami elementy $\sigma(i)$ i $\sigma(J)$.

Na zkończenie należy się uwaga na temat algorytmów losowania „obiektów kombinatorycznych" w R. Losowanie próbki (bez zwracania albo ze zwracaniem) można wykonać przy pomocy uniwersalnej funkcji `sample`. Jeśli chodzi o permutacje losowe, to można użyć dość prymitywnego, ale prostego kodu

```
> U <- runif(n)
> Perm <- sort(U, index.return = TRUE)$ix
```

1.8 Zadania i uzupełnienia

Zadania teoretyczne

1.6 Zadanie. Skonstruuj generator zmiennych losowych o gęstości

$$p(x) \propto \int_1^\infty y^{-n} e^{-xy} dy, \quad (x > 0).$$

Użyj metdy rozkładów warunkowych i marginalizcji: wygeneruj zmienną (X, Y) o gęstości $p(x, y) \propto y^{-n} e^{-xy}$. Przy okazji oblicz stałą normującą.

1.7 Zadanie. Zaproponuj algorytm generujący zmienną losową X o gęstości

$$p(x) = -\log x, \quad (0 < x < 1).$$

Sugestia: Rozważ gęstość $p(x, y) = \frac{1}{y}\mathbb{1}(0 < x < y < 1)$ i przypomnij sobie metodę rozkładów warunkowych.

1.8 Zadanie. Rozpatrz algorytm:

`Gen` $X \sim U(0, 1)$

`Gen` $U \sim U(0, 1)$

`if` $U < X$ `then return` X `else return` $1 - X$

Jaki jest rozkład prawdopodobieństwa zmiennej X na wyjściu? Wyjaśnij związek z metodą Alias (Przykład 1.7.2).

1.9 Zadanie. . Jeśli $W_1, \ldots, W_k$ są niezależne i każda ma rozkład wykładniczy, $X_i \sim \mathrm{Ex}(\lambda_i)$ to $\min(W_1, \ldots, W_k) \sim \mathrm{Ex}(\sum \lambda_i$. Jeśli J jest numerem zmiennej, dla której minimum jest przyjmowane, to $\mathbb{P}(J = j) = \lambda_j / \sum \lambda_i$. *Wskazówka:* Najpierw rozpatrz przypadek $k = 2$.

1.10 Zadanie. Uzasadnij poprawność algorytmu generującego permutacje losowe. Możesz też napisać kod w R i sprawdzić jego poprawność wykonując symulacje (jak?).

Nastepne zadania dotyczą statystyk pozycyjnych. Bezpośrednia metoda symulowania statystyk pozycyjnych polega na wygenerowaniu próbki z danego rozkładu i jej uporządkowaniu. Lepszy sposób oparty jest na Wniosku 1.6.6 w połączeniu z prostym spostrzeżeniem, które jest sformułowane w Zadaniu 1.11 poniżej.

1.11 Zadanie. Udowodnij, że jeśli $U_1, \ldots, U_n \sim_{\text{i.i.d.}} \mathrm{U}(0, 1)$ i $X_i = F^-(U_i)$ to $(F^-(U_{1:n}), \ldots, F^-(U_{n:n})$ jest wektorem statystyk porządkowych $(X_{1:n} \leqslant \cdots \leqslant X_{n:n})$ z rozkładu o dystrybuancie F.

Z Wniosku 1.6.6 i Twierdzenia 1.6.7 wynika reprezentacja statystyk pozycyjnych jako unormowanych sum zmiennych wykładniczych. To nie tylko ułatwia symulacje, ale podpowiada postać twierdzeń granicznych dla statystyk pozycyjnych i prowadzi do prostych dowodów.

1.12 Zadanie. Niech $\underline{M}_n = \min(U_1, \ldots, U_n)$ i $\overline{M}_n = \max(U_1, \ldots, U_n)$ dla $U_1, \ldots, U_n \sim_{\text{i.i.d.}} \mathrm{U}(0,1)$.

- Jaki jest rozkład prawdopodobieństwa $\underline{M}_n$? Jak dobrać ciąg b_n, żeby ciąg $b_n\underline{M}_n$ miał niezdegenerowany rozkład graniczny? Zidentyfikować ten rozkład.
- Niech
$$R_n = \underline{M}_n + 1 - \overline{M}_n.$$
Jaki rozkład prawdopodobieństwa ma zmienna R_n? Jak dobrać ciąg b_n, żeby ciąg b_nR_n miał niezdegenerowany rozkład graniczny? Zidentyfikować ten rozkład.
- Niech
$$C_n = \underline{M}_n + \overline{M}_n - 1$$
Jak dobrać ciąg b_n, żeby ciąg b_nC_n miał niezdegenerowany rozkład graniczny? Zidentyfikować ten rozkład.

1.13 Zadanie. Niech $U_1, \ldots, U_n \sim_{\text{i.i.d.}} \mathrm{U}(0,1)$ i $M_n = \mathrm{med}(U_1, \ldots, U_n)$. Załóżmy, że n jest liczbą nieparzystą. Jaki jest rozkład prawdopodobieństwa M_n? Udowodnij, że
$$b_n(M_n - 1/2) \to_{\mathrm{d}} \mathrm{N}(0,1).$$
Jak dobrać ciąg b_n, żeby to było prawdą?

Ćwiczenia komputerowe

1.9 Ćwiczenie. Napisz funkcję `rmult.norm(n,V)`, która generuje n niezależnych wektorów losowych o d-wymiarowym rozkładzie normalnym $\mathrm{N}(0, V)$, gdzie V jest $d \times d$ macierzą wariancji-kowariancji. *Wskazówka:* Funkcja `chol` oblicza rozkład Choleskiego macierzy. Najwygodniejszym formatem wyjścia jest *macierz* o wymiarach $n \times d$. Można dokonać transformacji liniowej wszystkich kolumn macierzy używając mnożenia macierzowego `%*%`.

Uwaga: W pakiecie `mvtnorm` istnieje gotowa funkcja `rmvnorm`. Porównaj jej działanie ze swoim kodem.

1.10 Ćwiczenie. Rozważmy punkty produkowane przez następujący algorytm:

`Gen` $\Phi \sim \mathrm{U}(0, 2\pi)$; `Gen` $\Psi \sim \mathrm{U}\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)$

$X := \cos\Phi\cos\Psi$; $Y := \sin\Phi\cos\Psi$; $Z := \sin\Psi$

Wygeneruj dużą liczbę punktów (X, Y, Z), przyjrzyj się nim (jak?) i przekonaj się, że *nie mają* one rozkładu jednostajnego na sferze S^2.

1.11 Ćwiczenie. Rozważmy punkty produkowane przez następujący algorytm:

`Gen` $\Phi \sim \mathrm{U}(0, 2\pi)$; `Gen` $Z \sim \mathrm{U}(-1, 1)$

$X := \sqrt{1 - Z^2}\cos\Phi$; $Y := \sqrt{1 - Z^2}\sin\Phi$

`return` (X, Y, Z)

Wygeneruj dużą liczbę punktów (X, Y, Z), przyjrzyj się nim i przekonaj się, że mają rozkład jednostajny na sferze S^2.

1.12 Ćwiczenie. Napisz funkcję `runisphere`, która generuje n niezależnych wektorów o rozkładzie jednostajnym na sferze $(d-1)$-wymiarowej.

- Wygeneruj dużą próbkę 3-wymiarowych wektorów $(X_1, X_2, X_3) \sim_{\text{i.i.d.}} \mathrm{U}(S^2)$. Zbadaj 1-wymiarowe rozkłady brzegowe. Jaki jest rozkład, powiedzmy, X_1?
- Wygeneruj dużą próbkę 4-wymiarowych wektorów $(X_1, X_2, X_3, X_4) \sim_{\text{i.i.d.}} \mathrm{U}(S^3)$. Zbadaj 2-wymiarowe rozkłady brzegowe. Jaki jest rozkład, powiedzmy, (X_1, X_2)?
- Zgadnij tezę następującego twierdzenia: *Jeśli* $(X_1, \ldots, X_d) \sim \mathrm{U}(S^{d-1})$ *to* $(X_1, \ldots, X_{d-2}) \sim \ldots ? \ldots$.

1.13 Ćwiczenie. Wygeneruj próbkę (X_i, Y_i), $i = 1, \ldots, n$ z 2-wymiarowego rozkładu Cauchy'ego. (rozkład t-Studenta z 1 stopniem swobody). Porównaj z rozkładem par (X_i, Y_i), gdzie X_i i Y_i są niezależne o 1-wymiarowym rozkładzie Cauchy'ego. Naszkicuj wykresy 2-wymiarowych gęstości w obu przypadkach. Użyj funkcji `contour` do narysowania poziomic gęstości.

Uwaga: W tym ćwiczeniu chodzi o narysowanie poziomic *teoretycznych* gęstości, przedstawionych odpowiednimi wzorami. Funkcja `contour` wymaga przygotowania danych przy użyciu funkcji `outer`, co wymaga pewnego wysiłku.

1.14 Ćwiczenie. Napisz funkcję `rdirich`, która generuje wektor o rozkładzie Dirichleta.

- Użyj generatora Beta.
- Użyj generatora Gamma.

Nastepne ćwiczenia dotyczą statystyk pozycyjnych. Porównajmy metodę generacji sformułowaną w Zadaniu 1.11 z metodą bezpośrednią.

1.15 Ćwiczenie. Wygeneruj wektor statystyk porządkowych $(X_{1:n} \leqslant \cdots \leqslant X_{n:n})$, czyli uporządkowaną próbkę $(X_1 \leqslant \cdots \leqslant X_n)$ z rozkładu wykładniczego $F = \mathrm{Ex}(1)$.

- Bezpośrednio, poprzez uporządkowanie próbki.
- Użyj Wniosku 1.6.8, własności rozkładu Dirichleta i Zadania 1.11. Sprawdź, że poniższe kawałki kodu są równoważne:

```
> X <- rexp(n)
> X <- sort(X,decreasing=TRUE)
```

oraz

```
> X <- rexp(n)
> X <- -log(cumsum(X)/sum(X))
```

- Oblicz 1-wymiarową gęstość $p_{X_{k:n}}(x)$ teoretycznie i porównaj z rezultatami doświadczenia.

Czasami nie trzeba i nie warto generować wszystkich statystyk pozycyjnych, tylko te, które nas interesują. Ilustruje to następne ćwiczenie.

1.16 Ćwiczenie. Użyj generatora Dirichleta do wyprodukowania pary wybranych statystyk porządkowych z rozkładu $\mathrm{U}(0,1)$, na przykład $(U_{2:5}, U_{4:5})$, bez generowania wszystkich statystyk porządkowych. Wyprodukuj próbkę (powiedzmy, wektory `U25` i `U45` zawierające m niezależnych par).

- Naszkicuj poziomice gęstości 2-wymiarowej zmiennych $(U_{2:5}, U_{4:5})$. Porównaj z estymatorem gęstości obliczonym na podstawie próbki – funkcja `kde2d` w pakiecie `MASS`.

  Sugestia: Napisz po prostu `contour(kde2d(U25,U45))`. Szkicowanie poziomic teoretycznej gęstości jest trudniejsze.

- Oblicz kowariancję $\mathrm{Cov}(U_{2:5}, U_{4:5})$ teoretycznie. Porównaj z wynikami symulacji.

1.17 Ćwiczenie. Na bezludnej wyspie żyją dwa gatunki ptaszków: A i B. Ornitolog zjawia się na tej wyspie i obserwuje ptaszki pojawiające się zgodnie ze schematem Bernoulliego. Niech ϑ oznacza prawdopodobieństwo natrafienia na gatunek A, zaś $1-\vartheta$ prawdopodobieństwo natrafienia na gatunek B. Ponieważ ornitolog nie znał dotychczas tych gatunków, nadaje im nazwy zgodnie z kolejnością napotkania po raz pierwszy. Niech $\tilde{\vartheta}$ będzie prawdopodobieństwem spotkania ptaszka tego gatunku, który został oznaczony jako pierwszy. Jaki jest rozkład $\tilde{\vartheta}$ przy założeniu, że ϑ jest zmienną losową?

Matematyczne sformułowanie zadania jest następujące: Niech ϑ będzie zmienną losową o gęstości p na przedziale $[0,1]$. Zmienna $\tilde{\vartheta}$ ma warunkowo rozkład dwupunktowy:

$$\mathbb{P}(\tilde{\vartheta} = \theta | \vartheta = \theta) = \theta, \qquad \mathbb{P}(\tilde{\vartheta} = 1-\theta | \vartheta = \theta) = 1-\theta.$$

Na początku załóżmy, że ϑ ma rozkład jednostajny $\mathrm{U}(0,1)$.

- Obliczyć $\mathbb{E}(\tilde{\vartheta}|\vartheta)$,
- Obliczyć $\mathbb{E}(\tilde{\vartheta})$ w zalezności od $\mathbb{E}(\vartheta)$,
- Obliczyć gęstość rozkładu prawdopodobieństwa zmiennej $\tilde{\vartheta}$.

Jaka będzie odpowiedź, jeśli założymy, że ϑ ma rozkład $\mathrm{Beta}(2,1)$, o gęstości $p(\theta) = 2\theta\mathbb{1}(0 \leqslant \theta \leqslant 1)$? Jeśli $\vartheta \sim p$, gdzie p jest *dowolną* gęstością na przedziale $[0,1]$?

- Znajdźmy rozwiązanie doświadczalnie. A może symulacje naprowadzą nas na właściwe rozwiązanie?
- Pojawiające się w ten sposób rozkłady zmiennej losowej $\tilde{\vartheta}$ nazywają się „Invariant under size-biased sampling" (ISBS) Skąd ta nazwa? Podać kilka przykładów rozkładów ISBS (rozkłady empiryczne, obliczone teoretycznie gęstości).
- Sprobować uogólnić na przypadek > 2 gatunków ptaszków.

Rozdział 2

Symulowanie procesów stochastycznych

Pełny tytuł tego rozdziału powinien brzmieć „Symulacje Niektórych Procesów Stochastycznych, Bardzo Subiektywnie Wybranych Spośród Mnóstwa Innych". Nie będę szczegółowo tłumaczył, skąd pochodzi mój subiektywny wybór. Zrezygnowałem z próby przedstawienia procesów z czasem ciągłym i równocześnie ciągłą przestrzenią stanów, bo to temat oddzielny i obszerny.

2.1 Procesy Poissona

Jednorodny proces Poissona na półprostej

2.1.1 Definicja. *Rozważmy niezależne zmienne losowe $W_1, \ldots, W_k, \ldots$ o jednakowym rozkładzie wykładniczym, $X_i \sim \mathrm{Ex}(\lambda)$ i utwórzmy kolejne sumy*

$$T_0 = 0,\ T_1 = W_1,\ T_2 = W_1 + W_2,\ \ldots,\ T_k = W_1 + \cdots + W_k,\ \ldots$$

Niech, dla $t \geqslant 0$,

$$N(t) = \max\{k : T_k \leqslant t\}.$$

Rodzinę zmiennych losowych $N(t)$ nazywamy ***procesem Poissona****.*

Proces Poissona dobrze jest wyobrażać sobie jako *losowy zbiór punktów* $\{T_1, T_2, \ldots, T_k, \ldots\}$ na półprostej $[0, \infty[$: Zmienna $N(t)$ oznacza liczbę punktów, które „wpadły" w odcinek $]0, t]$. Wygodnie będzie używać symbolu

$$N(s, t) = N(t) - N(s)$$

dla oznaczenia liczby punktów, które „wpadły" w odcinek $]s, t]$.

2.1.2 Stwierdzenie. *Jeśli $N(t)$ jest procesem Poissona, to*

$$\mathbb{P}(N(t)=k)=e^{-\lambda t}\frac{(\lambda t)^k}{k!}.$$

Dowód. Zauważmy, że

$$T_k\sim \text{Gamma}(k,\lambda).$$

Wobec tego ze wzoru na prawdopodobieństwo całkowite wynika, że

$$\begin{aligned}\mathbb{P}(N(t)=k)&=\mathbb{P}(T_k\leqslant t, T_{k+1}>t)\\&=\int_0^t \mathbb{P}(T_{k+1}>t|T_k=s)p_{T_k}(s)\,\mathrm{d}s\\&=\int_0^t \mathbb{P}(W_{k+1}>t-s|T_k=s)p_{T_k}(s)\,\mathrm{d}s\\&=\int_0^t \mathrm{e}^{-\lambda(t-s)}\frac{\lambda^k}{\Gamma(k)}s^{k-1}\mathrm{e}^{-\lambda s}\mathrm{d}s\\&=\mathrm{e}^{-\lambda t}\frac{\lambda^k}{\Gamma(k)}\int_0^t s^{k-1}\mathrm{d}s=\mathrm{e}^{-\lambda t}\frac{\lambda^k}{(k-1)!}\frac{t^k}{k}=\mathrm{e}^{-\lambda t}\frac{(\lambda t)^k}{k!}.\end{aligned}$$

□

Oczywiście $\mathbb{E}N(t)=\lambda t$. Liczba

$$\lambda=\frac{1}{\mathbb{E}W_i}=\frac{\mathbb{E}N(t)}{t}$$

jest nazywana *intensywnością procesu.*

2.1.3 Stwierdzenie. *Jeśli*

$$0<t_1<t_2<\cdots<t_i<\cdots,$$

to zmienne losowe $N(t_1), N(t_1,t_2),\ldots$ są niezależne i każda z nich ma rozkład Poissona:

$$N(t_{i-1},t_i)\sim\text{Poiss}(\lambda(t_i-t_{i-1})).$$

Dowód. Pokażemy, że warunkowo, dla $N(t_1)=k$, ciąg zmiennych losowych

$$T_{k+1}-t_1, W_{k+2}, W_{k+3}\ldots,\quad \text{jest iid}\sim\text{Ex}(\lambda).$$

Wynika to z własności braku pamięci rozkładu wykładniczego. W istocie, dla ustalonych $N(t_1)=k$ i $S_k=s$ mamy

$$\begin{aligned}&\mathbb{P}(T_{k+1}-t_1>t|N(t_1)=k,T_k=s)\\&=\mathbb{P}(T_{k+1}-t_1>t|T_k=s,T_{k+1}>t_1)\\&=\mathbb{P}(W_{k+1}>t_1+t-s|W_{k+1}>t_1-s)\\&=\mathbb{P}(W_{k+1}>t)=\mathrm{e}^{-\lambda t}.\end{aligned}$$

Fakt, że zmienne $W_{k+2}, W_{k+3} \dots$ są niezależne od zdarzenia $N(t_1) = k$ jest oczywisty. Pokazaliśmy w ten sposób, że losowy zbiór punktów $\{T_{k+1} - t_1, T_{k+2} - t_1, \dots\}$ ma (warunkowo, dla $N(t_1) = k$) taki sam rozkład prawdopodobieństwa, jak $\{T_1, T_2, \dots\}$. Proces Poissona *obserwowany od momentu t_1 jest kopią wyjściowego procesu.* Wynika stąd w szczególności, że zmienna losowa $N(t_2, t_1)$ jest niezależna od $N(t_1)$ i $N(t_2, t_1) \sim \text{Poiss}(\lambda(t_2 - t_1))$. Dalsza część dowodu przebiega analogicznie i ją pominiemy. □

Metoda generowania procesu Poissona oparta na Definicji 2.1.1 jest raczej oczywista. Nie jest to jednak *jedyna* metoda. Inny sposób generowania (i inny sposób patrzenia na proces Poissona) jest związany z następującym faktem.

2.1.4 Stwierdzenie. *Warunkowo, dla $N(t) = n$, ciąg zmiennych losowych*

$$T_1, \dots, T_n$$

ma rozkład taki sam, jak ciąg statystyk pozycyjnych

$$U_{1:n}, \dots, U_{n:n}$$

z rozkładu $U(0, t)$.

Dowód. Obliczmy warunkową gęstość zmiennych losowych $T_1, \dots, T_n$, jeśli $N(t) = n$:

$$\begin{aligned} p_{T_1,\dots,T_n|N(t)}(t_1, \dots, t_n|n) &= \frac{p_{T_1,\dots,T_n}(t_1, \dots, t_n) \cdot \mathbb{P}(N(t) = n | T_n = t_n)}{\mathbb{P}(N(t) = n)} \\ &= \frac{p_{W_1}(t_1) p_{W_2}(t_2 - t_1) \cdots p_{W_n}(t_n - t_{n-1}) \cdot \mathbb{P}(W_{n+1} > t - s)}{\mathbb{P}(N(t) = n)} \\ &= \frac{\lambda^n \mathrm{e}^{-\lambda t_1} \mathrm{e}^{-\lambda(t_2 - t_1)} \cdots \mathrm{e}^{-\lambda(t_n - t_{n-1})} \cdot \mathrm{e}^{-\lambda(t - t_n)}}{(\lambda t)^n \mathrm{e}^{-\lambda t} / n!} \\ &= \frac{n!}{t^n}, \end{aligned}$$

dla $0 \leqslant t_1 \leqslant \cdots \leqslant t_n \leqslant t$. □

Wynika stąd następujący sposób generowania procesu Poissona na przedziale $[0, t]$.

`Gen` $N \sim \text{Poiss}(\lambda t)$
`for` $i = 1$ `to` N `do Gen` $U_i \sim \text{U}(0, t)$;
`Sort` $(U_1, \dots, U_N)$
$(T_1, \dots, T_N) := (U_{1:N}, \dots, U_{N:N})$

Co ważniejsze, Stwierdzenia 2.1.2, 2.1.3 i 2.1.4 wskazują, jak powinny wyglądać uogólnienia procesu Poissona i jak generować takie ogólniejsze procesy. Zanim tym się zajmiemy, zróbmy dygresję i podajmy pewną „infinitezymalną" charakteryzację „zwykłego" procesu Poissona. Nietrudno sprawdzić, że proces Poissona spełnia następujące równania:

$$\begin{aligned} &\mathbb{P}(N(t+h) = n+1 | N(t) = n) = \lambda h + o(h), \\ &\mathbb{P}(N(t+h) = n | N(t) = n) = 1 - \lambda h + o(h), \quad h \searrow 0, \end{aligned} \tag{2.1.5}$$

gdzie $o(h)$ oznacza funkcję taką, że $\lim_{h \searrow 0} o(h)/h = 0$. Odwrotnie, można udowodnić, że równanie (2.1.5), przy pewnych naturalnych założeniach, charakteryzuje proces Poissona (Twierdzenie 2.4.8). Ten fakt nie jest bezpośrednio używany w symulacjach, ale wprowadza intuicyjny sposób określania procesów, użytecznych w probabilistycznym modelowaniu zjawisk. W podobnym języku łatwo formułować założenia o tak zwanych procesach urodzin i śmierci. Wrócimy do tego przy okazji omawiania procesów Markowa z czasem ciągłym.

Niejednorodne procesy Poissona w przestrzeni

Naturalne uogólnienia procesu Poissona polegają na tym, że rozważa się losowe zbiory punktów w przestrzeni o dowolnym wymiarze i dopuszcza się różną intensywność pojawiania się punktów w różnych rejonach przestrzeni. Niech $\mathcal{X}$ będzie przestrzenią polską. Czytelnik, który nie lubi abstrakcji może założyć, że $\mathcal{X} \subseteq \mathbb{R}^d$.

Musimy najpierw wprowadzić odpowiednie oznaczenia. Rozważmy ciąg zmiennych losowych o wartościach w $\mathcal{X}$ (punktów losowych):

$$X_1, \ldots, X_n, \ldots$$

(może to być ciag skonczony lub nie, liczba tych punktów może być zmienną losową). Niech, dla $A \subseteq \mathcal{X}$,

$$N(A) = \#\{i : X_i \in A\}$$

oznacza liczbę punktów, które „wpadły do zbioru A" (przy tym dopuszczamy wartość $N(A) = \infty$ i umawiamy się liczyć powtarzające się punkty tyle razy, ile razy występują w ciągu).

Niech teraz μ będzie miarą na (σ-ciele borelowskich podzbiorów) przestrzeni $\mathcal{X}$.

2.1.6 Definicja. *$N = N(\cdot)$ jest procesem Poissona z miarą intensywnosci μ, symbolicznie $N \sim \mathrm{PP}(\mu)$, jesli*

- *dla parami rozłącznych zbiorów $A_1, \ldots, A_i \subseteq \mathcal{X}$, odpowiadające im zmienne losowe $N(A_1), \ldots, N(A_i)$ są niezależne;*
- $\mathbb{P}(N(A) = n) = \mathrm{e}^{-\mu(A)} \dfrac{\mu(A)^n}{n!}$, *dla każdego $A \subseteq \mathcal{X}$ takiego, że $\mu(A) < \infty$ i dla $n = 0, 1, \ldots$.*

Z elementarnych własności rozkładu Poissona wynika następujący wniosek.

2.1.7 Wniosek. *Rozważymy rozbicie zbioru A o skończonej mierze intensywności, $\mu(A) < \infty$, na rozłączną sumę $A = A_1 \cup \cdots \cup A_k$. Wtedy*

$$\begin{aligned}\mathbb{P}\left(N(A_1) = n_1, \ldots, N(A_k) = n_k | N(A) = n\right) \\ = \frac{n!}{n_1! \cdots n_k!} \mu(A_1)^{n_1} \cdots \mu(A_k)^{n_k}, \qquad (n_1 + \cdots + n_k = n).\end{aligned}$$

Jeśli natomiast $\mu(A) = \infty$ to łatwo zauważyć, że $N(A) = \infty$ z prawdopodobieństwem 1.

Z Definicji 2.1.6 i Wniosku 2.1.7 natychmiast wynika następujący algorytm generowania procesu Poissona. Załóżmy, że interesuje nas „fragment" procesu w zbiorze $A \subseteq \mathcal{X}$ o skończonej mierze intensywności. W praktyce zawsze symulacje muszą się do takiego fragmentu ograniczać. Zauważmy, że *unormowana* miara $\mu(\cdot)/\mu(A)$ jest rozkładem prawdopodobieństwa (zmienna losowa X ma ten rozkład, jeśli $\mathbb{P}(X \in B) = \mu(B)/\mu(A)$, dla $B \subseteq A$).

`Gen` $N \sim \text{Poiss}(\mu(A))$
`for` $i = 1$ `to` N `do Gen` $X_i \sim \mu(\cdot)/\mu(A)$

Należy rozumieć, że formalnie definiujemy $N(B) = \#\{i : X_i \in B\}$ dla $B \subseteq A$, w istocie jednak za realizację procesu uważamy zbiór punktów $\{X_1, \ldots, X_N\} \subset A$ („zapominamy" o uporządkowaniu punktów). Widać, że to jest proste uogólnienie analogicznego algorytmu dla „zwykłego" procesu Poissona, podanego wcześniej.

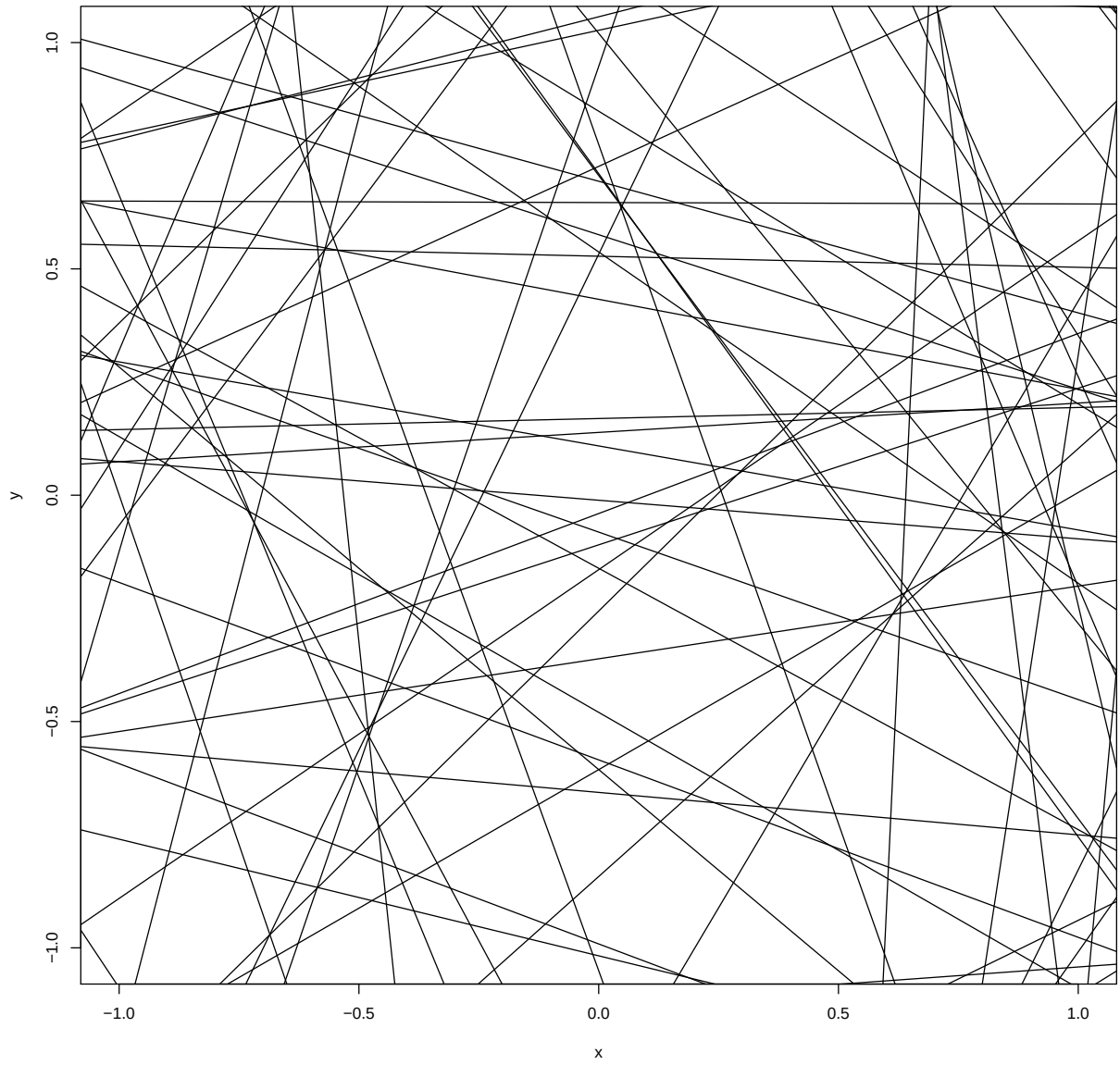
Można również rozbić zbiór A na rozłączną sumę $A = A_1 \cup \cdots \cup A_k$ i generować *niezależnie* fragmenty procesu w każdej części A_j.

2.1.8 Przykład. Jednorodny proces Poissona na kole $B^2 = \{x^2 + y^2 \leqslant 1\}$ można wygenerować w następujący sposób. Powiedzmy, że intensywność (punktów na jednostkę pola) jest λ, to znaczy $\mu(B) = \lambda|B|$, gdzie $|B|$ jest polem (miarą Lebesgue'a) zbioru B.

Najpierw generujemy $N \sim \text{Poiss}(\lambda\pi)$, następnie punkty $(X_1, Y_1), \ldots, (X_N, Y_N)$ niezależnie z rozkładu $\text{U}(B^2)$. To wszystko. △

Ciekawszy jest następny przykład.

2.1.9 Przykład (Jednorodny proces Poissona w przestrzeni prostych)**.** Prostą na płaszczyźnie można sparametryzować podając kąt $\varphi \in [0, 2\pi[$ jako tworzy prostopadła do prostej z osią poziomą oraz odległość $r \geqslant 0$ prostej od początku układu. Każda prosta jest więc opisana przez parę liczb (φ, r) czyli punkt przestrzeni $\mathcal{L} = [0, 2\pi[\times [0, \infty[$. Jeśli teraz wygenerujemy jednorodny proces Poissona na tej przestrzeni, to znaczy proces o intensywności $\mu(B) = \lambda|B|$, $B \subseteq \mathcal{L}$, to można się spodziewać zbioru „losowo położonych prostych". To widać na Rysunku 2.1.



Rysunek 2.1: Proces Poissona w przestrzeni prostych.

W istocie, wybór parametryzacji zapewnia, że rozkład prawdopodobieństwa procesu nie zależy od wyboru układu współrzędnych. Dowód, czy nawet precyzyjne sformułowanie tego stwierdzenia przekracza ramy tego skryptu. Intuicyjnie chodzi o to, że na obrazku „nie ma wyróżnionego kierunku ani wyróżnionego punktu". Nie można sensownie zdefiniować pojęcia „losowej prostej" ale każdy przyzna, że proces Poissona o którym mowa można uznać za uściślenie intuicyjnie rozumianego pojęcia „losowego *zbioru* prostych". △

Ciekawe, że podstawowe metody generowania zmiennych losowych mają swoje odpowiedniki dla procesów Poissona. Rolę rozkładu prawdopodobieństwa przejmuje miara intensywności.

2.1.10 Przykład (Odwracanie dystrybuanty)**.** Dla miary intensywności λ na przestrzeni jednowymiarowej można zdefiniować *dystrybuantę* tej miary. Dla uproszczenia rozważmy przestrzeń $\mathcal{X} = [0, \infty[$ i założymy, że każdy zbiór ograniczony ma miarę skończoną. Niech $\Lambda(t) = \lambda([0, t])$. Funkcję $\Lambda : [0, \infty[\to [0, \infty[$ nazwiemy dystrybuantą. Jest ona niemalejąca, prawostronnie ciągła, ale granica w nieskończoności $\Lambda(\infty) = \lim_{x\to\infty} \Lambda(x)$ może być dowolnym elementem z $[0, \infty]$. Dla procesu Poissona na $[0, \infty[$ wygodnie wrócić do prostszych oznaczeń, pisząc $N(t) = N([0, t])$ jak we wcześniej rozpatrywanym przypadku jednorodnym.

Niech $J(t)$ będzie jednorodnym procesem Poissona na $[0, \infty[$ z intensywnoscia równą 1. Wtedy

$$N(t) = J(\Lambda(t))$$

jest niejednorodnym procesem Poissona z miarą intensywności λ. W istocie, jeśli $0 < t_1 < t_2 < \cdots$ to $N(t_1), N(t_2) - N(t_1), \ldots$ są niezależne i maja rozkłady odpowiednio $\mathrm{Poiss}(\Lambda(t_1)), \mathrm{Poiss}(\Lambda(t_2) - \Lambda(t_1)), \ldots$. Zauważmy, że jeśli $R_1 < R_2 < \cdots$ oznaczają punkty skoku procesu $J(\cdot)$ to $N(t) = \max\{k : R_k \leqslant \Lambda(t)\} = \max\{k : \Lambda^-(R_k) \leqslant t\}$, gdzie Λ^- jest uogólnioną funkcją odwrotną do dystrybuanty. Wobec tego punktami skoku procesu N są $T_k = \Lambda^-(R_k)$. Algorytm jest taki:

`Gen` $N \sim \mathrm{Poiss}(\Lambda(t))$`;`

`for` $i := 1$ `to` N `do Gen` $R_i \sim \mathrm{U}(0, \Lambda(t))$; $\quad T_i := \Lambda^-(R_i)$

Pominęliśmy tutaj sortowanie skoków i założyliśmy, że symulacje ograniczamy do odcinka $[0, t]$. △

2.1.11 Przykład (Przerzedzanie)**.** To jest odpowiednik metody eliminacji. Załóżmy, że mamy dwie miary intensywności: μ o gęstości m i λ o gęstości l. To znaczy, że $\lambda(B) = \int_B l(x)\mathrm{d}x$ i $\mu(B) = \int_B m(x)\mathrm{d}x$ dla dowolnego zbioru $B \subseteq \mathcal{X}$. Załóżmy, że $l(x) \leqslant m(x)$ i przypuśćmy, że umiemy generować proces Poissona o intensywności μ. Niech $X_1, \ldots, X_N$ będą punktami tego procesu w zborze A o skończonej mierze μ (wiemy, że $N \sim \mathrm{Poiss}(\mu(A))$. Punkt X_i *akceptujemy* z prawdopodobieństwem $l(X_i)/m(X_i)$ (pozostawiamy w zbiorze) lub odrzucamy (usuwamy ze zbioru) z prawdopodobieństwem $1 - l(X_i)/m(X_i)$. Liczba pozostawionych punktów L ma rozkład $\mathrm{Poiss}(\lambda(A))$, zaś każdy z tych punktów ma rozkład o gęstości $l(x)/\lambda(A)$, gdzie $\lambda(A) = \int_A l(x)\mathrm{d}x$. Te punkty tworzą proces Poissona z miarą intensywności λ.

`Gen` $\mathbb{X} = \{X_1, \ldots, X_N\} \sim \mathrm{PP}(\mu(\cdot))$`;`

`for` $i := 1$ `to` N `Gen` $U_i \sim \mathrm{U}(0, 1)$`;`

  `if` $U_i > l(X_i)/m(X_i)$ `then` $\mathbb{X} := \mathbb{X} \setminus X_i$`;`

`return` $\mathbb{X} = \{X'_1, \ldots, X'_L\} \sim \mathrm{PP}(\lambda(\cdot))$

△

2.1.12 Przykład (Superpozycja)**.** To jest z kolei odpowiednik metody *kompozycji*. Metoda opiera się na następującym prostym fakcie. Jezeli $N_1(\cdot), \ldots, N_k(\cdot)$ są niezależnymi procesami Poissona z miarami intensywnosci odpowiednio $\mu_1(\cdot), \ldots, \mu_k(\cdot)$, to $N(\cdot) = \sum_i N_i(\cdot)$ jest procesem Poissona z intensywnością $\mu(\cdot) = \sum_i \mu_i(\cdot)$. Dodawanie należy tu rozumieć w dosłaowny sposób, to znaczy $N(A)$ jest określone jako $\sum_i N_i(A)$ dla każdego zbioru A. Jeśli utożsamimy procesy z losowymi zbiorami punktów to odpowiada temu operacja brania sumy mnogościowej (złączenia zbiorów). Niech $X_{j,1}, \ldots, X_{j,N_j}$ będą punktami j-tego procesu w zborze A o skończonej mierze μ_j.

$\mathbb{X} = \emptyset$;
`for` $j := 1$ `to` k `do`
  `begin`
  `Gen` $\mathbb{X}_j = \{X_{j,1}, \ldots, X_{j,N_j}\} \sim \mathrm{PP}(\mu_j(\cdot))$;
  $\mathbb{X} := \mathbb{X} \cup \mathbb{X}_j$;
  `end`
`return` $\mathbb{X} = \{X'_1, \ldots, X'_N\} \sim \mathrm{PP}(\mu(\cdot))$ `{ mamy tu` $N = \sum_j N_j$ `}`

△

Wygodnie jest utożsamiać procesy Poissona z losowymi zbiorami punktów, jak uczyniliśmy w ostatnich przykładach (i mniej jawnie w wielu miejscach wcześniej). Te zbiory można rozumieć w zwykłym sensie, dodawać, odejmować tak jak w teorii mnogości pod warunkiem, że ich elementy się *nie powtarzają*. W praktyce mamy najczęściej do czynienia z intensywnościami, które mają gęstości „w zwykłym sensie", czyli względem miary Lebesgue'a. Wtedy, z prawdopodobieństwem 1, punkty procesu Poissona nie powtarzają się.

2.2 Symulowanie łańcuchów i procesów Markowa

Czas dyskretny, przestrzeń dyskretna

Zacznijmy od najprostszej sytuacji. Rozważmy skończony lub przeliczalny zbiór $\mathcal{X}$, który będziemy nazywali *przestrzenią stanów*. Czasem wygodnie przyjąć, że $\mathcal{X} = \{1, 2, \ldots, d\}$ lub $\mathcal{X} = \{1, 2, \ldots\}$. Jest to tylko umowne ponumerowanie stanów.

2.2.1 Definicja. *(i) Ciąg $X_0, X_1, \dots, X_n, \dots$ zmiennych losowych o wartościach w przestrzeni $\mathcal{X}$ nazywamy* ***łańcuchem Markowa****, jeśli dla każdego $n = 1, 2, \dots$ i dla każdego ciągu punktów $x_0, x_1, \dots, x_n, x_{n+1} \in \mathcal{X}$,*

$$\begin{aligned}\mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_1 = x_1, X_0 = x_0) \\ = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n),\end{aligned}$$

(o ile tylko $\mathbb{P}(X_n = x_n, X_{n-1} = x_{n-1}, \dots, X_1 = x_1, X_0 = x_0) > 0$, czyli prawdopodobieństwo warunkowe w tym wzorze ma sens).

(ii) Łańcuch Markowa nazywamy ***jednorodnym****, jeśli dla dowolnych stanów x i x' i każdego n możemy napisać*

$$\mathbb{P}(X_{n+1} = x' | X_n = x) = P(x, x'),$$

to znaczy prawdopodobieństwo warunkowe w powyższym wzorze zależy tylko od x i x', ale nie zależy od n.

Jeśli $X_n = x$, to mówimy, że łańcuch w chwili n znajduje się w stanie $x \in \mathcal{X}$. Warunek (i) w Definicji 2.2.1 znaczy tyle, że przyszła ewolucja łańcucha zależy od stanu obecnego, ale nie zależy od przeszłości. Łańcuch jest jednorodny, jeśli prawo ewolucji łańcucha nie zmienia się w czasie. W dalszym ciągu rozpatrywać będziemy głównie łańcuchy jednorodne. Macierz

$$P = (P(x, x'))_{x,x' \in \mathcal{X}} = \begin{pmatrix} P(1,1) & \cdots & P(1,x') & \cdots & \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ P(x,1) & \cdots & P(x,x') & \cdots & \\ \vdots & \ddots & \vdots & \ddots & \end{pmatrix}$$

nazywamy *macierzą* (prawdopodobieństw) *przejścia* łańcucha. Jest to macierz kwadratowa $(d \times d)$, jeśli przestrzeń $\mathcal{X}$ jest skończona, ale może być „macierzą" o nieskończonej liczbie wierszy i kolumn. Macierz P jest stochastyczna, to znaczy $P(x, x') \geqslant 0$ dla dowolnych stanów $x, x' \in \mathcal{X}$ oraz $\sum_{x'} P(x, x') = 1$ dla każdego $x \in \mathcal{X}$. Jeśli $\nu(x) = \mathbb{P}(X_0 = x)$, to (być może nieskończony) wektor wierszowy

$$\nu^\top = \big(\nu(1), \dots, \nu(x), \dots\big)$$

nazywamy *rozkładem początkowym* łańcucha (oczywiście, $\sum_x \nu(x) = 1$). Jest jasne, że

$$\begin{aligned}\mathbb{P}(X_0 = x_0, X_1 = x_1, \dots, X_{n-1} = x_{n-1}, X_n = x_n) \\ = \nu(x_0) P(x_0, x_1) \cdots P(x_{n-1}, x_n).\end{aligned} \tag{2.2.2}$$

Ten wzór określa jednoznacznie łączny rozkład prawdopodobieństwa zmiennych $X_0, X_1, \dots, X_n, \dots$ i może być przyjęty za definicję jednorodnego łańcucha Markowa. W tym sensie możemy utożsamić łańcuch z parą (ν, P): łączny rozkład prawdopodobieństwa jest wyznaczony przez podanie rozkładu początkowego i macierzy przejścia.

Opiszemy teraz bardzo ogólną konstrukcję, która jest podstawą algorytmów generujących łańcuchy Markowa. Wyobraźmy sobie, jak zwykle, że mamy do dyspozycji ciąg $U = U_0, U_1, \ldots, U_n, \ldots$ „liczb losowych", produkowanych przez komputerowy generator, czyli z teoretycznego punktu widzenia ciąg *niezależnych zmiennych losowych o jednakowym rozkładzie*, $\mathrm{U}(0,1)$. Niech $\phi : [0,1] \to \mathcal{X}$ i $\psi : \mathcal{X} \times [0,1] \to \mathcal{X}$ będą takimi funkcjami, że

$$\begin{aligned} &\mathbb{P}\left(\psi(U) = x\right) = \nu(x) \quad \text{dla każdego } x \in \mathcal{X}, \\ &\mathbb{P}\left(\phi(x,U) = x'\right) = P(x,x') \quad \text{dla dowolnych } x, x' \in \mathcal{X}. \end{aligned} \tag{2.2.3}$$

Jeśli określimy $X_0, X_1, \ldots, X_n, \ldots$ rekurencyjnie w następujący sposób:

$$X_0 = \psi(U_0), \qquad X_{n+1} = \phi\left(X_n, U_{n+1}\right), \tag{2.2.4}$$

to jest jasne, że tak otrzymany ciąg zmiennych losowych jest łańcuchem Markowa z rozkładem początkowym ν i macierzą przejścia P.

Na zakończenie tego podrozdziału zróbmy kilka prostych uwag.

- Gdy symulujemy łańcuch Markowa, kodujemy funkcję ϕ zazwyczaj w postaci R-owej funkcji `krok()` tak, że instrukcja `X<-krok(X)` aktualizuje $X := X_n$ do $X := X_{n+1}$. Jawne zapisanie macierzy P w pamięci jest niemal zawsze zbędne, a często niemozliwe.
- Można sobie wyobrażać, że dla ustalonego x, zbiór $\{u : \phi(x,u) = y\}$ jest *odcinkiem* długości $P(x,y)$, ale nie musimy tego żądać. Nie jest istotne, że zmienne U_i mają rozkład $\mathrm{U}(0,1)$. Moglibyśmy założyć, że są określone na dość dowolnej przestrzeni $\mathcal{U}$. Istotne jest, że te zmienne są niezależne, mają jednakowy rozkład i zachodzi wzór (2.2.3). W praktyce zazwyczaj do zrealizowania jednego „kroku" łańcucha Markowa generujemy więcej niż jedną „liczbę losową".

Czas dyskretny, przestrzeń ciągła

Przypadek ciągłej lub raczej *ogólnej* przestrzeni stanów jest w istocie równie prosty, tylko oznaczenia trzeba trochę zmienić i sumy zamienić na całki. Dla prostoty przyjmijmy, że $\mathcal{X} \subseteq \mathbb{R}^d$. Poniżej sformułujemy definicję w taki sposób, żeby podkreślić analogię do przypadku dyskretnego i pominiemy subtelności teoretyczne. Zwróćmy tylko uwagę, że prawdopodobieństwo warunkowe nie może tu być zdefiniowane tak elementarnie jak w Definicji 2.2.1, bo zdarzenie warunkujące może mieć prawdopodobieństwo zero.

2.2.5 Definicja. *(i) Ciąg $X_0, X_1, \ldots, X_n, \ldots$ zmiennych losowych o wartościach w $\mathcal{X}$ nazywamy **łańcuchem Markowa**, jeśli dla każdego $n = 1, 2, \ldots$, dla każdego ciągu $x_0, x_1, \ldots, x_n$ punktów przestrzeni $\mathcal{X}$ oraz dowolnego (borelowskiego) zbioru $B \subseteq \mathcal{X}$,*

$$\begin{aligned} &\mathbb{P}(X_{n+1} \in B | X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_1 = x_1, X_0 = x_0) \\ &\quad = \mathbb{P}(X_{n+1} \in B | X_n = x_n), \end{aligned}$$

(dla prawie wszystkich punktów $(x_n, x_{n-1}, \ldots, x_1, x_0)$*).*

(ii) Łańcuch Markowa nazywamy ***jednorodnym****, jeśli dla dowolnego zbioru* $B \subseteq \mathcal{X}$ *i każdego* n *możemy napisać*

$$\mathbb{P}(X_{n+1} \in B | X_n = x) = P(x, B)$$

(dla prawie wszystkich punktów x*).*

Funkcja $P(\cdot, \cdot)$, której argumentami są *punkt* x *i zbiór* B, jest nazywana *jądrem przejścia*. Ważne dla nas jest to, że dla ustalonego $x \in \mathcal{X}$, jądro $P(x, \cdot)$ rozważane jako funkcja *zbioru* jest rozkładem prawdopodobieństwa. W wielu przykładach jest to rozkład zadany przez gęstość,

$$P(x, B) = \int_B p(x, x')\mathrm{d}x'.$$

Wtedy odpowiednikiem wzoru (2.2.2) jest następujący wzór na łączną gęstość:

$$p(x_0, x_1, \ldots, x_{n-1}, x_n) = \nu(x_0)p(x_0, x_1) \cdots p(x_{n-1}, x_n), \tag{2.2.6}$$

gdzie $\nu(\cdot)$ jest gęstością rozkładu początkowego, zaś $p(x_i, x_{i+1}) = p(x_{i+1}|x_i)$ jest gęstością warunkową. Sformułowaliśmy Definicję 2.2.5 nieco ogólniej głównie dlatego, że w symulacjach ważną rolę odgrywają łańcuchy dla których prawdopodobieństwo przejścia nie ma gęstości. Przykładem może być łańcuch, który z niezerowym prawdopodobieństwem potrafi „stać w miejscu", to znaczy $P(x, \{x\}) =: \alpha(x) > 0$, i ma prawdopodobieństwo przejścia postaci, powiedzmy

$$P(x, B) = \alpha(x)\mathbb{1}(x \in B) + (1 - \alpha(x)) \int_B p(x, x')\mathrm{d}x'.$$

Łańcuch Markowa na ogólnej przestrzeni stanów można rekurencyjnie generować w sposób opisany wzorem (2.2.4), to znaczy $X_0 = \psi(U_0)$ i $X_{n+1} = \phi(X_n, U_{n+1})$ dla ciągu i.i.d. $U_0, U_1, \ldots, U_n, \ldots$ z rozkładu $\mathrm{U}(0, 1)$. Niech $\nu(\cdot)$ oznacza rozkład początkowy. Funkcje ψ i ϕ muszą spełniać warunki

$$\begin{aligned} &\mathbb{P}(\psi(U) \in B) = \nu(B) \quad \text{dla każdego (mierzalnego) } B \subseteq \mathcal{X}, \\ &\mathbb{P}(\phi(x, U) \in B) = P(x, B) \quad \text{dla dowolnych } x \in \mathcal{X}, B \subseteq \mathcal{X}. \end{aligned} \tag{2.2.7}$$

Różnica między (2.2.3) i (2.2.7) jest tylko taka, że w ogólnej sytuacji rozkłady prawdopodobieństwa zmiennych losowych $\psi(U)$ i $\phi(x, U)$, czyli odpowiednio $\nu(\cdot)$ i $P(x, \cdot)$ nie muszą być dyskretne. Funkcje ψ i ϕ w abstrakcyjnym schemacie (2.2.7) możemy zastąpić w zasadzie dowolną metodą generacji zmiennych o zadanym rozkładzie, przy czym (dla ϕ) ten zadany rozkład zależy od x.

2.2.8 Przykład (Proces AR(1))**.** Niech $\mathcal{X} = \mathbb{R}$ i rozważmy jądro przejścia

$$P(x, B) = \mathbb{P}(X_{n+1} \in B | X_n = x) = \int_B \frac{1}{\sqrt{2\pi}v} \exp\left[-\frac{1}{2v^2}(x' - x)^2\right] \mathrm{d}x'.$$

Równoważnie, $X_{n+1}|X_n = x \sim \mathrm{N}(x, v^2)$ lub $X_{n+1} = X_n + W_{n+1}$, gdzie W_{n+1} jest zmienną losową niezależną od X_n, o rozkładzie $\mathrm{N}(0, v^2)$. Napisana w R funkcja `krok()`, która aktualizuje $X := X_n$ do $X := X_{n+1}$ może być, powiedzmy, taka

```
krok <- function(X)
return (rnorm(1,mean=X,sd=v))
```

Łańcuch Markowa $X_0, \ldots, X_n, \ldots$ jest specjalnym przypadkiem procesu autoregresji 1-go rzędu, czyli „AR(1)". Ogólniejsze procesy autoregresji są omówione w Podrozdziale 2.3. △

Czas ciągły, przestrzeń dyskretna

Rozważmy proces stochastyczny $(X(t), t \geqslant 0)$, czyli rodzinę zmiennych losowych o wartościach w *skończonej lub przeliczalnej przestrzeni stanów* $\mathcal{X}$, indeksowaną przez parameter t, który interpretujemy jako *ciągły czas*. Pojedyncze stany oznaczamy przez $x, x', x'', \ldots \in \mathcal{X}$ lub podobnie. Załóżmy, że trajektorie są prawostronnie ciągłymi funkcjami mającymi lewostronne granice (prawie na pewno).

2.2.9 Definicja. *(i) Mówimy, że $X(t)$ jest* ***procesem Markowa****, jeśli dla dowolnych $x, x' \in \mathcal{X}$ oraz $s, t \geqslant 0$,*

$$\mathbb{P}\left(X(s+t) = x' | X(s) = x, X(u), 0 \leqslant u < s\right) = \mathbb{P}\left(X(t+s) = x' | X(s) = x\right).$$

(ii) Proces Markowa nazywamy ***jednorodnym****, jeśli dla dowolnych stanów x i x' i każdego t i s możemy napisać*

$$\mathbb{P}\left(X(s+t) = x' | X(s) = x\right) = P^t(x, x'),$$

to znaczy prawdopodobieństwo warunkowe w powyższym wzorze zależy tylko od x, x' i t, ale nie zależy od s.

Sens założenia (i) jest taki sam jak w Definicji 2.2.1. Żądamy mianowicie, żeby zmienna $X(t+s)$ była *warunowo niezależna* od zmiennych $X(u), 0 \leqslant u < s$ pod warunkiem zdarzenia $X(s) = x$. Oczywiście, $P^t(x, y)$ jest *prawdopodobieństwem przejścia w ciągu czasu t*. Rozkład początkowy oznaczymy przez $\nu(x) = \mathbb{P}\left(X(0) = x\right)$. Ponieważ $\mathcal{X}$ jest skończona, ν może być przedstawiony jako wektor a P^t jako macierz.

Odpowiednikiem macierzy przejścia jest *macierz intensywności przejść* zdefiniowana następująco.

$$Q(x,x') = \lim_{t\to 0} \frac{1}{h}\left[P^h(x,x') - I(x,x')\right],$$

gdzie $I = P^0$ jest macierzą identycznościową. Można udowodnić, że granice istnieją. Tak więc $Q = \frac{\mathrm{d}}{\mathrm{d}t}P^t|_{t=0}$ i $P^h = I + hQ + o(h)$ przy $h \searrow 0$. Innymi słowy, $Q(x,x')$ jest rzeczywiście „intensywnością skoków z x do x'" (na jednostkę czasu). Mamy wzór analogiczny do (2.1.5):

$$\begin{aligned}\mathbb{P}\left(X(t+h) = x' | X(t) = x\right) &= hQ(x,x') + o(h) \text{ dla } x \neq y;\\ \mathbb{P}\left(X(t+h) = x | X(t) = x\right) &= 1 + hQ(x,x) + o(h), \quad h \searrow 0.\end{aligned}$$

Zauważmy, że mamy $\sum_{x'} Q(x,x') = 0$. Wspomnijmy jeszcze mimochodem, że macierze przejścia wyrażają się przez macierz intensywności Q przy pomocy „macierzowej funkcji wykładniczej": $P^t = \exp[tQ]$. Dla uproszczenia notacji, niech $Q(x) = -Q(x,x) = \sum_{x'\neq x} Q(x,x')$ oznacza "intensywność wszystkich skoków ze stanu x".

Z tego co zostało powiedziane łatwo się domyślić, że generowanie procesu Markowa można zorganizować podobnie jak dla procesu Poissona, poprzez *czasy skoków*. Niech $0 < T_1 < T_2 < \cdots < T_n < \cdots$ będą kolejnymi momentami skoków,

$$\begin{aligned}T_1 &= \inf\left\{t > 0 : X(t) \neq X(0)\right\};\\ T_{n+1} &= \inf\left\{t > T_n : X(t) \neq X(T_n)\right\}.\end{aligned}$$

Jeśli $X(T_n) = x$, to czas oczekiwania na następny skok, $W_{n+1} = T_{n+1} - T_n$ zależy tylko od x i ma rozkład *wykładniczy* z parametrem $Q(x)$. W szczególności, mamy $\mathbb{E}\left(W_{n+1} | X(T_n) = x\right) = 1/Q(x)$. Zadanie 2.4 wyjaśnia, dlaczego pojawia się rozkład wykładniczy. Jeśli obserwujemy proces w momentach skoków, czyli skupimy uwagę na ciągu $\hat{X}_n = X(T_n)$ to otrzymujemy łańcuch Markowa z czasem dyskretnym, nazywany czasem „szkieletem". Jego prawdopodobieństwa przejścia są takie:

$$\hat{P}(x,x') = \mathbb{P}(\hat{X}_{n+1} = x' | \hat{X}_n = x) = \begin{cases} Q(x,x')/Q(x) & \text{jeśli } x \neq x';\\ 0 & \text{jeśli } x = x'.\end{cases} \tag{2.2.10}$$

Oczywiście, cała trajektoria procesu $X(t)$ jest w pełni wyznaczona przez momenty skoków $T_1, \ldots, T_n, \ldots$ i kolejne stany łańcucha szkieletowego $\hat{X}_0, \hat{X}_1, \ldots, \hat{X}_n, \ldots$. Po prostu, $X(t) = \hat{X}_n$ dla $T_n \leqslant t < T_{n+1}$.

Następujący, sławny *algorytm Gillespie'go* formalizuje opisany powyżej sposób generacji.

```
Gen X̂_0 ~ ν(·);
T_0 := 0;
for i := 1 to ∞ do
  begin
  Gen W_i ~ Ex(Q(X̂_{i-1}));
  T_i := T_{i-1} + W_i;
  Gen X̂_i ~ P̂(X̂_{i-1}, · ); { P̂ dane wzorem powyżej }
  end
```

W praktyce trzeba ustalić sobie jakiś skończony horyzont czasowy $t_{\max}$ i zakończyć symulacje gdy $T_i > t_{\max}$. Zauważmy, że ten algorytm nadaje się do symulowania procesu Markowa na *nieskończonej* ale *przeliczalnej* przestrzeni stanów, na przykład $\mathcal{X} = \{0, 1, 2, \ldots\}$. W istocie, metoda została wynaleziona przez Gillespie'go do symulowania wielowymiarowych procesów urodzin i śmierci (głównie w zastosowaniach do chemii). Te procesy mają najczęściej przestrzeń stanów postaci $\mathcal{X} = \{0, 1, 2, \ldots\}^d$; stanem układu jest układ liczb $x = (x(1), \ldots, x(d))$, gdzie $x(i)$ jest liczbą osobników (na przykład cząstek) typu i.

Czasami warto zmodyfikować bazowy algorytm Gillespie'go w następujący sposób. Generowanie procesu Markowa można rozbić na dwie fazy: najpierw wygenerować „potencjalne czasy skoków" a następnie symulować „szkielet", który będzie skakał wyłącznie w poprzednio otrzymanych momentach (ale nie koniecznie we wszystkich spośród nich). Opiszemy dokładniej tę konstrukcję, która bazuje na własnościach procesu Poissona. Wyobraźmy sobie najpierw, że dla każdego stanu $x \in \mathcal{X}$ symulujemy niezależnie jednorodny proces Poissona $\mathbb{R}^x$ o punktach skoku $R_1^x < \cdots < R_k^x < \cdots$ przy czym ten proces ma intensywność $Q(x)$. Jeśli teraz, w drugiej fazie $\hat{X}_{i-1} = X(T_{i-1}) = x$, to za następny moment skoku wybierzemy najbliższy punkt procesu $\mathbb{R}^x$ na prawo od T_{i-1}, czyli $T_i = \min\{R_k^x : R_k^x > T_{i-1}\}$. Z własności procesu Poissona (patrz Stwierdzenie 2.1.3 i jego dowód) wynika, że zmienna $T_i - T_{i-1}$ ma rozkład wykładniczy z parametrem $Q(x)$ i metoda jest poprawna. Naprawdę nie ma nawet potrzeby generowania wszystkich procesów Poissona $\mathbb{R}^x$. Warto posłużyć się metodą *przerzedzania* i najpierw wygenerować jeden proces o dostatecznie dużej intensywności a następnie „w miarę potrzeby" go przerzedzać. Niech $\mathbb{R}^*$ będzie procesem Poissona o intensywności $Q^* \geqslant \max_x Q(x)$ i oznaczmy jego punkty skoków przez $\{R_1 < \cdots < R_k < \cdots\}$. Jeśli w drugiej fazie algorytmu mamy $X(R_{i-1}) = x$ w pewnym momencie $R_{i-1} \in \mathbb{R}^*$, to zaczynamy przerzedzać proces $\mathbb{R}^*$ w taki sposób, aby otrzymać proces o intensywności $Q(x) \leqslant Q^*$. Dla każdego punktu R_j, $j \geqslant i$ powinniśmy rzucić monetą i z prawdopodobieństwem $(1-Q(x))/Q^*$ ten punkt usunąć. Ale jeśli usuniemy punkt R_i to znaczy, że w tym punkcie *nie ma skoku*, czyli $X(R_i) = x$. Jeśli punkt R_i zostawimy, to wykonujemy skok, losując kolejny stan z prawdopodobieństwem $\hat{P}(x, \cdot)$. W rezultacie, stan $X(R_i)$ jest wylosowany z rozkładu praw-

dopodobieństwa

$$\tilde{P}(x,y) = \begin{cases} Q(x,x')/Q^* & \text{jeśli } x \neq x'; \\ 1 - Q(x)/Q^* & \text{jeśli } x = x'. \end{cases} \tag{2.2.11}$$

Niech $\tilde{X}_n = X(R_i)$ będzie „nadmiarowym szkieletem" procesu. Jest to łańcuch Markowa, który (w odróżnieniu od „cieńszego szkieletu" $\hat{X}_n$) *może* w jednym kroku pozostać w tym samym stanie i ma prawdopodobieństwa przejścia $\mathbb{P}(\tilde{X}_n = x' | \tilde{X}_{n-1} = x) = \tilde{P}(x,x')$.

Odpowiada temu następujący dwufazowy algorytm.

Faza I:

`Gen` $\mathbb{R} \sim \mathrm{PP}(Q^*)$ *{ proces Poissona }*

Faza II:

`Gen` $\tilde{X}_0 \sim \nu(\cdot)$;
`for` $i := 1$ `to` ∞ `do Gen` $\tilde{X}_i \sim \tilde{P}(\hat{X}_{i-1}, \cdot)$; *{* $\tilde{P}$ *dane wzorem powyżej }*

W tym ostatnim algorytmie pojawia problem, jeśli $\max_x Q(x) = \infty$, co jest możliwe dla nieskończonej przestrzeni $\mathcal{X}$ i jest typową sytuacją dla procesów urodzin i śmierci. Żeby sobie z tym poradzić, można wziąć Q^* dostatecznie duże, żeby „na ogół" wystarczało, a w mało prawdopodobnym przypadku zawitania do stanu x z $Q(x) > Q^*$ przerzucać się na pierwszy, jednofazowy algorym.

2.3 Stacjonarne procesy Gaussowskie

Ograniczymy się do dwóch klas procesów, często używanych do modelowania różnych zjawisk. Będą to procesy z czasem dyskretnym i przestrzenią stanów $\mathbb{R}$, to znaczy ciągi (zależnych) zmiennych losowych $X_0, X_1, \ldots, X_n, \ldots$ o wartościach rzeczywistych. Niech W_i będą niezależnymi zmiennymi losowymi o jednakowym rozkładzie $\mathrm{N}(0, v^2)$. Wygodnie rozważyć tutaj dwustronnie nieskończony ciąg $\ldots, W_{-1}, W_0, W_1, \ldots, W_n, \ldots$.

2.3.1 Definicja. *Proces* ***ruchomych średnich*** *rzędu q, w skrócie MA(q) jest określony równaniem*

$$X_n = \beta_1 W_{n-1} + \cdots + \beta_q W_{n-q}, \qquad (n = 0, 1, \ldots),$$

gdzie $\beta_1, \ldots, \beta_q$ jest ustalonym ciągiem współczynników.

Sposób generowania takiego procesu jest oczywisty i wynika wprost z definicji. Co więcej widać, że proces MA(q) jest *stacjonarny*, to znaczy łączny rozkład prawdopodobieństwa zmiennych $X_0, X_1, \ldots, X_n$ jest taki sam jak zmiennych $X_k, X_{k+1}, \ldots, X_{k+n-1}$, dla dowolnych n i k. Intuicyjnie, proces nie zmienia się po „przesunięciu czasu" o k jednostek.

2.3.2 Definicja. *Proces **autoregresji** rzędu p, w skrócie $AR(p)$ jest określony równaniem rekurencyjnym*

$$X_n = \alpha_1 X_{n-1} + \cdots + \alpha_p X_{n-p} + W_n, \qquad (n = p, p+1, \ldots),$$

gdzie $\alpha_1, \ldots, \alpha_p$ jest ustalonym ciągiem współczynników.

Procesy autoregresji wydają się bardzo odpowiednie do modelowania „szeregów czasowych": stan układu w chwili n zależy od stanów przeszłych i dotatkowo jeszcze od przypadku. Procesy AR(1), w szczególności są łańcuchami Markowa. Sposób generowania procesów AR(p) jest też bezpośrednio widoczny z definicji. Pojawia się jednak pewien problem. Jak znaleźć $X_0, \ldots, X_{p-1}$ na początku algorytmu w taki sposób, żeby proces był stacjonarny? Jest to o tyle istotne, że rzeczywiste procesy (na przykład czeregi czasowe w zastosowaniach ekonomicznych) specjaliści uznają za stacjonarne, przynajmniej w przybliżeniu. Załóżmy, że spełniony jest następujący konieczny i dostateczny warunek istnienia stacjonarnego procesu AR(p).

2.3.3 Założenie. *Wielomian charakterystyczny $A(z) = 1 - \alpha_1 z - \ldots - \alpha_p z^p$ nie ma zer w kole $\{|z| \leqslant 1\}$.*

Rozważmy dla wygody oznaczeń podwójnie nieskończony proces

$$\ldots, X_{-1}, X_0, X_1, \ldots$$

spełniający równanie autoregresji rzędu p. Załóżmy, że ten proces jest stacjonarny i wektor $X_0, \ldots, X_{p-1}$ rozkład normalny, $\mathrm{N}(0, \Sigma)$. Stacjonarność implikuje, że elementy macierzy Σ muszą być postaci $\mathrm{Cov}(X_i, X_j) = \sigma^2 \rho_{i-j}$. Mamy przy tym $\rho_{-k} = \rho_k$, co może być traktowane jako wygodna konwencja (po to właśnie „rozszerzamy" proces w obie strony). Zastosowanie równania definiującego autoregresję prowadzi do wniosku, że

$$\begin{aligned}\sigma^2 \rho_k &= \mathrm{Cov}(X_0, X_k) = \mathrm{Cov}(X_0, \alpha_1 X_{k-1} + \cdots + \alpha_p X_{k-p} + W_k) \\ &= \sigma^2 \alpha_1 \rho_{k-1} + \sigma^2 \alpha_2 \rho_{k-2} + \cdots + \sigma^2 \alpha_p \rho_{k-p}.\end{aligned}$$

Podobnie,

$$\begin{aligned}\sigma^2 &= \mathrm{Var}(X_0) = \mathrm{Cov}(X_0, \alpha_1 X_{-1} + \cdots + \alpha_p X_{-p} + W_0) \\ &= \sigma^2 \alpha_1 \rho_1 + \sigma^2 \alpha_2 \rho_2 + \cdots + \sigma^2 \alpha_p \rho_p + v^2,\end{aligned}$$

gdzie $v^2 = \mathrm{Var}(W_0)$. Otrzymujemy następujący układ równań na współczynniki autokorelacji ρ_k (są to sławne równania Yule'a-Walkera):

$$\text{(2.3.4)} \qquad \begin{cases} \rho_1 = \alpha_1 + \alpha_2 \rho_1 + \alpha_3 \rho_2 + \ldots + \alpha_p \rho_{p-1}, \\ \rho_2 = \alpha_1 \rho_1 + \alpha_2 + \alpha_3 \rho_1 + \ldots + \alpha_p \rho_{p-2}, \\ \ldots, \\ \rho_p = \alpha_1 \rho_{p-1} + \alpha_2 \rho_{p-2} + \alpha_3 \rho_2 + \ldots + \alpha_p, \end{cases}$$

Można pokazać, że przy Założeniu 2.3.3 ten układ ma dokładnie jedno rozwiązanie. Ponadto mamy równanie na wariancję stacjonarną:

$$\sigma^2 = \frac{v^2}{1 - \rho_1\alpha_1 - \ldots - \alpha_p\rho_p} \tag{2.3.5}$$

Metoda generowania ciągu $X_0, X_1, \ldots, X_p, \ldots$, który jest *stacjonarnym* procesem AR(p) jest następująca. Znajdujemy rozwiązanie układu równań (2.3.4), wariancję obliczamy ze wzoru (2.3.5) i tworzymy macierz $\Sigma = (\sigma^2\rho_{i-j})_{i,j=0,\ldots,p-1}$. Najpierw generujemy wektor losowy $(X_0, X_1, \ldots, X_{p-1}) \sim \mathrm{N}(0, \Sigma)$. Następnie generujemy rekurencyjnie $X_p, X_{p+1}, \ldots$ używając równania autoregresji. Aby się przekonać, że tak generowany proces jest stacjonarny, wystarczy sprawdzić że identyczne są rozkłady wektorów $(X_0, X_1, \ldots, X_{p-1})$ i $(X_1, X_2, \ldots, X_p)$. W tym celu obliczamy wariancję „nowej zmiennej" X_p i jej kowariancje z poprzednimi zmiennymi. Korzystamy ze wzoru $X_p = \alpha_1 X_{p-1} + \cdots + \alpha_p X_0$, ze znajomości struktury kowariancji zmiennych $X_{p-1}, \ldots, X_0$ i z równań Y-W. Sprawdzamy kolejno równości $\mathrm{Cov}(X_p, X_{p-k}) = \sigma^2\rho_k$ dla $k = 1, \ldots, p$ i następnie równość $\mathrm{Var}\, X_p = \sigma^2$ co kończy dowód.

2.4 Zadania i uzupełnienia

Zadania teoretyczne

2.1 Zadanie. Łączny rozkład zmiennych losowych N i N_1 jest zdefiniowany przez podanie rozkładu brzegowego $N \sim \mathrm{Poiss}(\lambda)$ i warunkowego $(N_1|N = n) \sim \mathrm{Bin}(n, \theta)$. Jeśli $N_0 = N - N_1$ to zmienne losowe N_1 i N_0 są niezależne, $N_1 \sim \mathrm{Poiss}(\theta\lambda)$, $N_0 \sim \mathrm{Poiss}((1-\theta)\lambda)$.

Udowodnij. Wyjaśnij związek z procedurą „przerzedzania" dla procesu Poissona.

2.2 Zadanie. Odwrotnie, jeśli zmienne losowe N_1 i N_0 są niezależne, $N_1 \sim \mathrm{Poiss}(\lambda_1)$, $N_0 \sim \mathrm{Poiss}(\lambda_0)$ i $N = N_1 + N_0$, to $N \sim \mathrm{Poiss}(\lambda = \lambda_1 + \lambda_0)$ i warunkowo $(N_1|N = n) \sim \mathrm{Bin}(n, \theta = \lambda_1/\lambda)$.

Udowodnij. Wyjaśnij związek z procedurą „superpozycji" procesów Poissona.

2.3 Zadanie. Załóżmy, że zmienne losowe N i $T_1, \ldots, T_n, \ldots$ są niezależne, $T_i \sim \mathrm{Ex}(\lambda)$ i $N \sim \mathrm{Geo}(\theta)$, czyli $\mathbb{P}(N = k) = (1-\theta)^{k-1}\theta$ dla $k = 1, 2, \ldots$. Niech $T = \sum_{i=1}^{N} T_i$. Pokaż, że $T \sim \mathrm{Ex}(\theta\lambda)$.

Wyjaśnij związek z procedurą „przerzedzania" dla procesu Poissona.

2.4 Zadanie. Jeżeli $T \sim \mathrm{Ex}(\lambda)$ to

$$\mathbb{P}(T \leqslant t + h | T > t) = \lambda h + o(h)$$

przy $h \searrow 0$. Odwrotnie, jeśli powyższe równanie zachodzi dla $T \geqslant 0$ i dystrybuanta $F(t) = \mathbb{P}(T \leqslant t)$ jest różniczkowalna, to T ma rozkład wykładniczy.

2.5 Zadanie. Łańcuch Markowa na przestrzeni stanów $\mathcal{X} = \{1, 2\}$ ma macierz przejścia

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}.$$

- Oblicz rozkład stacjonarny π w zależności od α i β.
- Oblicz prawdopodobieństwa przejścia w n krokach, czyli macierz P^n. *Wskazówka:* Wystarczy obliczyć np. $p_n = P^n(1,1) = \mathbb{P}(X_n = 1|X_0 = 1)$. Łatwo ułożyć równanie rekurencyjne: $p_n = p_n(1-\alpha) + (1-p_n)\beta$ i rozwiązać to równanie.
- Wygeneruj trajektorię $X_0, X_1, \ldots, X_n$ i narysuj jej wykres dla różnych punktów początkowych (możesz użyć np. funkcji `plot(...,type='s')`. Porównaj przebieg procesu np. dla $\alpha = \beta = 0.1$ i dla $\alpha = \beta = 0.9$ (zauważ, że rozkłady stacjonarne w obu przypadkach są identyczne).
- Wyestymuj rozkład stacjonarny π na podstawie symulowanej trajektorii (dla różnych wartości α, β i porównaj z teorią. *Wskazówka:* Mocne Prawo Wielkich Liczb dla łańcuchów Markowa pozwala estymować π na podstawie jednej (możliwie długiej) trajektorii (np. $n = 1000000$). Istotnie,
$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(X_i = x) \to_{\text{p.n.}} \pi(x).$$
- Wygeneruj trajektorię łańcucha stacjonarnego (zacznij symulacje od $X_0 \sim \pi$).

2.6 Zadanie. Rozważmy proces AR(1), czyli łańcuch Markowa na przestrzeni $\mathcal{X} = \mathbb{R}$ zdefiniowany równaniem rekurencyjnym

$$X_{n+1} = \alpha X_n + W_{n+1},$$

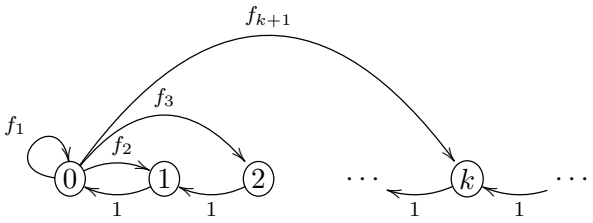
gdzie $W_1, W_2, \ldots$ są niezależnymi zmiennymi losowymi o rozkładzie $N(0, v^2)$. Powtórz serię doświadczeń i obliczeń podobnie jak w poprzednim zadaniu:

- Oblicz rozkład stacjonarny π w zależności od α.
- Wyprowadź wzór na X_n w zależności od X_0 i „innowacji" $W_1, \ldots, W_n$. Podaj rozkład warunkowy $X_n|X_0 = x$.
- Wygeneruj trajektorię $X_0, X_1, \ldots, X_n$ i narysuj jej wykres dla różnych punktów początkowych (możesz użyć np. funkcji `plot(...,type='l')`. Porównaj przebieg procesu np. dla $\alpha = 0.9$ i dla $\alpha = -0.9$ (zauważ, że rozkłady stacjonarne w obu przypadkach są identyczne).
- Wyestymuj rozkład stacjonarny π na podstawie symulowanej trajektorii (dla różnych wartości α i porównaj z teorią. *Wskazówka:* Mocne Prawo Wielkich Liczb dla łańcuchów Markowa pozwala estymować π na podstawie jednej (możliwie długiej) trajektorii (np. $n = 1000000$). Istotnie,
$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(X_i \in A) \to_{\text{p.n.}} \pi(A).$$
- Wygeneruj trajektorię łańcucha stacjonarnego (zacznij symulacje od $X_0 \sim \pi$).

2.7 Zadanie. Niech $(f_1, \ldots, f_k, \ldots)$ będzie rozkładem prawdopodobieństwa, to znaczy $f_k \geqslant 0$ i $\sum_{k=1}^{\infty} f_k = 1$. Rozważmy łańcuch Markowa na przestrzeni $\mathcal{X} = \{0, 1, \ldots\}$ o następujących prawdopodobieństwach przejścia

$$\mathbb{P}(X_{n+1} = k-1 | X_n = k) = 1 \text{ dla } k = 1, 2, \ldots,$$
$$\mathbb{P}(X_{n+1} = k | X_n = 0) = f_k \text{ dla } k = 1, 2, \ldots.$$

Przedstawia to następujący diagram:



- Oblicz rozkład stacjonarny π, jeśli istnieje. Zbadaj istnienie rozkładu stacjonarnego w zależności od ciągu $(f_1, \ldots, f_k, \ldots)$. *Wskazówka:* Jeśli π jest rozkładem stacjonarnym, to nietrudno wyrazić $\pi(k)$ w zależności od $\pi(0)$.
- Rozważ kilka szczególnych przypadków:
  - $f_k = \frac{1}{m}$ dla $k = 1, \ldots, m$ (rozkład jednostajny na zbiorze $\{1, \ldots, m\}$).
  - $f_k = (1-\theta)^{k-1}\theta$ dla $k = 1, 2, \ldots$ (rozkład geometryczny na zbiorze $\{1, 2, \ldots\}$)..
  - $f_k = \frac{1}{k(k+1)}$ dla $k = 1, 2, \ldots$.

  Przeprowadź obliczenia teoretyczne i porównaj z wynikami symulacji.

  Uwaga: W ostatnim przykładzie konieczna jest ostrożność w interpretacji wyników symulacyjnych. Zastanów się, jaka jest granica

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(X_n = k).$$

2.8 Zadanie. Reguła przejścia łańcucha Markowa na przestrzeni $\mathcal{X} = [0, 1]$ jest zdefiniowana następujaco: dla danego X_n,

$$\begin{cases} X_{n+1} \sim \mathrm{U}(0, X_n) & \text{z prawdopodobieństwem } a; \\ X_{n+1} \sim \mathrm{U}(X_n, 1) & \text{z prawdopodobieństwem } 1-a \end{cases} \tag{2.4.1}$$

$(0 < a < 1)$.

- Znajdź rozkład stacjonarny teoretycznie.
- Wyestymuj rozkład stacjonarny symulacyjnie (np. histogram). *Wskazówka:* PWL dla łańcuchów Markowa pozwala estymować ten rozkład z pojedynczej trajektorii.

2.9 Zadanie. Proces Markowa z czasem ciągłym przestrzeni stanów $\mathcal{X} = \{1, 2\}$ ma macierz intensywnosci przejścia

$$Q = \begin{pmatrix} -\alpha & \alpha \\ \beta & -\beta \end{pmatrix}.$$

- Oblicz rozkład stacjonarny π w zależności od α i β.
- Wygeneruj trajektorię $(X(s), 0 \leqslant s \leqslant t)$ i narysuj jej wykres (np. `plot(...,type='s')`). Porównaj przebieg procesu np. dla $\alpha = \beta = 0.1$ i dla $\alpha = \beta = 0.9$ (zauważ, że rozkłady stacjonarne w obu przypadkach są identyczne).
- Wyestymuj rozkład stacjonarny π na podstawie symulowanej trajektorii (dla różnych wartości α, β) i porównaj z teorią. *Wskazówka:* Mocne Prawo Wielkich Liczb dla procesów Markowa z czasem ciągłym ma postać:

  $$\frac{1}{t}\int_0^t \mathbb{1}(X(s) = x)\mathrm{d}s \to_{\text{p.n.}} \pi(x),$$

  przy $t \to \infty$. Dla naszego procesu MPWL jest spełnione, więc wystarczy symulować *jedną, długą* trajektorię. Zwróć uwagę, że trzeba wziąć pod uwagę nie tylko trajektorię łańcucha szkieletowego (ta ma zawsze postać $(1, 2, 1, 2, \ldots, 1, 2, \ldots)$ lub $(2, 1, 2, 1, \ldots, 2, 1, \ldots)$), ale również czasy przebywania w obu stanach.
- Wygeneruj trajektorię procesu stacjonarnego (zacznij symulacje od $X(0) \sim \pi$).

Model Chandrasekhara/Smoluchowskiego (proces narodzin i śmierci)

2.10 Zadanie (1 pudełko)**.** Rozważmy „pudełko" zawierające pewną liczbę „cząstek". Niech $X(t)$ oznacza liczbę cząstek znajdujących się w pudełku w chwili $t \geqslant 0$. Ewolucję procesu (w czasie ciągłym) opisują następujące prawa:

- $\star$ Do pudełka wpadają nowe cząstki, przy czym strumień tych wpadających cząstek stanowi jednorodny w czasie proces Poissona z intensywnością $a_\star$.
- $\dagger$ Każda cząstka znajduje sie w pudełku przez okres czasu będący zmienną losową o rozkładzie wykładniczym z parametrem $a_\dagger$ (o wartości średniej $1/a_\dagger$), po czym opuszcza pudełko bezpowrotnie. Czasy przebywania w pudełku dla poszczególnych cząstek są stochastycznie niezależne.

Schematycznie:

$$\star \xrightarrow{a_\star} \square \xrightarrow{a_\dagger} \dagger$$

Opisz prawa ewolucji w postaci

- $\star$ $\mathbb{P}(X(t+h) = x+1 | X(t) = x) = \ldots + o(h)$ dla $h \searrow 0$.
- $\dagger$ $\mathbb{P}(X(t+h) = x-1 | X(t) = x) = \ldots + o(h)$ dla $h \searrow 0$.

Równoważnie, napisz wzory na intensywności przejść $Q(x, x')$ procesu Markowa $X(t)$.

- Przeprowadź symulację tego procesu. Sugerowane wartości parametrów: $a_\star = 10$, $a_\dagger = 2$. Zrób wykres kilku trajektorii, dla różnych stanów początkowych.
- Zbadaj symulacyjnie zbieżność procesu do rozkładu stacjonarnego:

$$\lim_{t\to\infty} \mathbb{P}(X(t) = x) = \pi(x).$$

  Oszacuj stacjonarną wartość oczekiwaną

$$\lim_{t\to\infty} \mathbb{E}X(t) = ?$$

  Zbadaj symulacyjnie rozkład stacjonarny (np. `barplot`).

  Wskazówka: Podobnie jak w Zadaniu 2.9, wystarczy symulować *jedną, długą* trajektorię.
- Zgadnij (powiedzmy, na podstawie symulacji) postać rozkładu stacjonarnego i udowodnij, że to jest rzeczywiście rozkład stacjonarny.

2.11 Zadanie (2 pudełka)**.** Rozważmy teraz 2 pudełka i następujący schemat przepływu cząstek:

$$\star \xrightarrow{a_{\star 1}} \boxed{1} \xrightarrow{a_{12}} \boxed{2} \xrightarrow{a_{2\dagger}} \dagger$$

Reguły ewolucji są podobne jak poprzednio, tylko każda cząstka opuszczająca pudełko X_1 momentalnie przechodzi do pudełka X_2.

- Napisz wzory na intensywnosci przejścia 2-wymiarowego procesu Markowa $X(t) = (X_1(t), X_2(t))$.
- Przeprowadź symulacje podobne jak w przypadku jednego pudełka. Zbadaj istnienie i postać 2-wymiarowego rozkładu stacjonarnego $\pi(x_1, x_2)$.
- Oszacuj symulacyjnie $\lim_{t\to\infty} \mathrm{Cov}(X_1(t), X_2(t))$. Zastanów się jak to zrobić i zinterpretuj wynik.

Model Ehrenfestów (wersja z czasem ciągłym)

2.12 Zadanie. Pudełko jest podzielone na 2 części, z niewielkim otworem w przegrodzie. W pudełku jest r cząstek, które mogą się przemieszczać (losowo i niezależnie) z lewej części do prawej i odwrotnie. Dla pojedynczej cząstki, intensywność przejść z lewej części do prawej jest a, zaś z prawej do lewej b:

$$\boxed{\text{L}} \underset{b}{\overset{a}{\rightleftarrows}} \boxed{\text{P}}$$

Jeśli więc $X(t)$ oznacza liczbę cząstek w lewej części, to ewolucja układu jest opisana równaniami

$$\mathbb{P}(X(t+h) = x-1 | X(t) = x) = axh + o(h) \quad \text{dla } h \searrow 0,$$
$$\mathbb{P}(X(t+h) = x+1 | X(t) = x) = b(r-x)h + o(h) \quad \text{dla } h \searrow 0.$$

- Przeprowadź symulacje i rozpoznaj rozkład stacjonarny.
- Udowodnij wynik teoretycznie.
- Wyjaśnij związek z Zadaniem 2.9.

Stochastyczne modele epidemiologiczne

Zjawiska opisywane na poziomie „mikro" przez procesy Markowa, zachowują się na poziomie „makro" jak funkcje deterministyczne będące rozwiązaniami równań różniczkowych. Wyjaśnia to bardzo prosty (ale, niestety, bardzo aktualny w czasie pisania tego skryptu) model „wykładniczej fazy epidemii".

2.13 Zadanie (Początek epidemii)**.** Rozważmy bardzo dużą populację[1]. Niech $I(t)$ oznacza liczbę zarażonych w chwili t. Zakładamy, że w krótkim okresie czasu h każdy osobnik I zaraża średnio αh innych ludzi, ale przy tym z prawdopodobieństwem βh zdrowieje lub umiera i przestaje zarażać. Deterministyczny model jest następujący:

$$I(t+h) = I(t) + (\alpha - \beta)I(t)h + o(h) \quad \text{dla } h \searrow 0. \tag{2.4.2}$$

(Traktujemy $I(t)$ jako wielkość ciągłą.) Oczywiście, (2.4.2) sprowadza się do zwyczajnego równania różniczkowego:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = (\alpha - \beta)I(t). \tag{2.4.3}$$

Popatrzmy na to samo zjawisko z bliska (w skali „mikro"). Potraktujmy $I(t)$ jako zmienną losową o wartościach całkowitoliczbowych. Ewolucja procesu $I(t)$ jest opisana równaniami

$$\begin{aligned} &\mathbb{P}(I(t+h) = i+1 | I(t) = i) = \alpha i h + o(h) \text{ dla } h \searrow 0, \\ &\mathbb{P}(I(t+h) = i-1 | I(t) = i) = \beta i h + o(h) \text{ dla } h \searrow 0, \\ &\mathbb{P}(I(t+h) = i | I(t) = i) = 1 - (\alpha+\beta) i h + o(h) \text{ dla } h \searrow 0. \end{aligned} \tag{2.4.4}$$

- Napisz rozwiązanie równań (2.4.3) z warunkiem początkowym $I(0) = 1$.
- Przeprowadź symulacje procesu Markowa opisanego równaniami (2.4.4) (algorytm Gillespie'go). Przyjmij warunek początkowy jak wyżej: $I(0) = 1$. Prześledź i narysuj kilka trajektorii procesu losowego $I(t)$, na tle rozwiązania równania różniczkowego. Wybierz kilka wartości (α, β) takich, że $\alpha > \beta$. Co się dzieje, jeśli ten warunek *nie* jest spełniony?
- Zrób rysunek w skali logarytmicznej (na osi I).
- Wygeneruj *wiele* trajektorii procesu losowego, oblicz i narysuj $\mathbb{E}I(t)$, $\mathrm{med}I(t)$, parę kwantyli (metodą Monte Carlo), porównaj z równaniem różniczkowym.

[1] To założenie pozwala zignorować fakt, że przyrost $I(t)$ zmniejsza liczbę osób narażonych na zarażenie. To uproszczenie jest uzasadnione na początku epidemii.

2.14 Zadanie (Model SIR)**.** Rozważmy populację złożoną z ℓ osobników. Niech $I(t)$ oznacza liczbę zarażonych w chwili t, zaś $R(t)$ łączną liczbę uodpornionych i zmarłych. Liczba osobników narażonych na zakażenie jest równa $S(t) = \ell - I(t) - R(t)$. Osobnik typu S może przejść do kategorii I, a stąd do kategorii R (stąd nazwa „model SIR"). Schematycznie:

$$\boxed{\text{S}} \xrightarrow{\alpha} \boxed{\text{I}} \xrightarrow{\beta} \boxed{\text{R}}$$

Zakładamy, że w krótkim okresie czasu h każdy osobnik I zaraża średnio $\alpha(S(t)/\ell)h$[2] innych ludzi, a z prawdopodobieństwem βh zdrowieje lub umiera i przestaje zarażać. Klasyczny model deterministyczny jest następujący:

$$\begin{aligned} I(t+h) &= I(t) + \left(\alpha\frac{S(t)}{\ell} - \beta\right) I(t)h + o(h) \quad \text{dla } h \searrow 0, \\ S(t+h) &= S(t) - \alpha\frac{S(t)}{\ell} I(t)h + o(h) \quad \text{dla } h \searrow 0, \end{aligned} \tag{2.4.5}$$

$R(t) = \ell - I(t) - S(t)$. Oczywiście, (2.4.2) jest układem równań różniczkowych zwyczajnych:

$$\begin{aligned} \frac{\mathrm{d}I(t)}{\mathrm{d}t} &= \left(\alpha\frac{S(t)}{\ell} - \beta\right) I(t), \\ \frac{\mathrm{d}S(t)}{\mathrm{d}t} &= -\alpha\frac{S(t)}{\ell} I(t). \end{aligned} \tag{2.4.6}$$

W skali „mikro" traktujmy $I(t)$ i $S(t)$ jako zmienne losowe o wartościach całkowitoliczbowych. Stan układu jest parą $(I(t), S(t)) = (i, s)$. Ewolucja procesu jest opisana równaniami

$$\begin{aligned} &\mathbb{P}(I(t+h) = i+1, S(t) = s-1 | I(t) = i, S(t) = s) = \alpha\frac{S(t)}{\ell} ih + o(h) \text{ dla } h \searrow 0, \\ &\mathbb{P}(I(t+h) = i-1, S(t+h) = s | I(t) = i, S(t) = s) = \beta ih + o(h) \text{ dla } h \searrow 0, \\ &\mathbb{P}(I(t+h) = i, S(t+h) = s | I(t) = i, S(t) = s) = 1 - \left(\beta + \alpha\frac{S(t)}{\ell}\right) ih + o(h) \text{ dla } h \searrow 0. \end{aligned} \tag{2.4.7}$$

- Przeprowadź symulacje procesu Markowa opisanego równaniami (2.4.7) (algorytm Gillespie'go). Przyjmij warunki początkowe: $I(0) = 1$, $S(0) = \ell - 1$. Prześledź i narysuj kilka trajektorii procesu losowego $I(t)$. Przykładowe wartości $(\alpha, \beta, \ell) = (2, 1, 1000)$ lub $(2, 1, 5000)$. Zorientuj się, jakie są losowe fluktuacje przebiegu procesu (maksymalna liczba zarażonych, czas największego nasilenia epidemii, wreszcie liczba „uodpornionych" po wygaśnięciu zarazy).

- Porównaj z rozwiązaniem równania różniczkowego (2.4.6). *Wskazówka:* Funkcja `ODE` w pakiecie `deSolve` rozwiązuje numerycznie zwyczajne równania różniczkowe.

[2] Jeśli $S(t)/\ell \simeq 1$ to dostajemy model „fazy wykładniczej" przedstawiony w zadaniu poprzednim.

Charakteryzacja procesu Poissona

Następujące twierdzenie pokazuje, że własności (2.1.5), w połaczeniu z pewnymi naturalnymi założeniami, w pełni charakteryzują proces Poissona.

2.4.8 Twierdzenie. *Załóżmy, że $(N(t) : t \geqslant 0)$ jest procesem o wartościach w $\{0, 1, 2, \ldots\}$, stacjonarnych i niezależnych przyrostach (to znaczy $N(t) - N(s)$ jest niezależne od $(N(u), u \leqslant s)$ i ma rozkład zależny tylko od $t - s$ dla dowolnych $0 < s < t$) oraz, że trajektorie $N(t)$ są prawostronnie ciągłymi funkcjami mającymi lewostronne granice (prawie na pewno). Jeżeli $N(0) = 0$ i spełnione są następujące warunki:*

$$(i) \quad \lim_{t\to 0} \frac{\mathbb{P}(N(t) = 1)}{t} = \lambda,$$

$$(ii) \quad \lim_{t\to 0} \frac{\mathbb{P}(N(t) \geqslant 2)}{t} = 0,$$

to $N(\cdot)$ jest jednorodnym procesem Poissona z intensywnością λ

Bardzo prosto można zauważyć, że proces Poissona $(N(t) : t \geqslant 0)$ o intensywności λ ma własności wymienione w Twierdzeniu 2.4.8. Ciekawe jest, że te własności *w pełni charakteryzują* proces Poissona.

Dowód Tw. 2.4.8 – szkic. Pokażemy tylko, że

$$p_n(t) := \mathbb{P}(N(t) = n) = \mathrm{e}^{-\lambda t}\frac{(\lambda t)^n}{n!}.$$

Najpierw zajmiemy się funkcją $p_0(t) = \mathbb{P}(N(t) = 0)$. Z niezależności i jednorodności przyrostów wynika tożsamość

$$p_0(t + h) = p_0(h)p_0(t).$$

Stąd

$$\frac{p_0(t + h) - p_0(t)}{h} = \frac{p_0(h) - 1}{h}p_0(t) = \left[- \frac{p_1(h)}{h} - \frac{\sum_{i=2}^{\infty} p_i(h)}{h}\right]p_0(h).$$

Przejdźmy do granicy z $h \to 0$ i skorzystajmy z własności (i) i (ii). Dostajemy proste równanie różniczkowe:

$$p_0'(t) = -\lambda p_0(t).$$

Rozwiązanie tego równania z warunkiem początkowym $p_0(0) = 1$ jest funkcja

$$p_0(t) = \mathrm{e}^{-\lambda t}.$$

Bardzo podobnie obliczamy kolejne funkcje p_n. Postępujemy rekurencyjnie: zakładamy, że znamy p_{n-1} i układamy równanie różniczkowe dla funkcji p_n. Podobnie jak poprzednio,

$$p_n(t + h) = p_n(t)p_0(h) + p_{n-1}(t)p_1(h) + \sum_{i=2}^{n} p_{n-1}(t)p_i(h),$$

a zatem

$$\frac{p_n(t+h)-p_n(t)}{h} = \frac{p_0(h)-1}{h}p_n(t) + \frac{p_1(h)}{h}p_{n-1}(t) + \frac{1}{h}\sum_{i=2}^{n} p_{n-i}(t)p_i(h).$$

Korzystając z własności (i) i (ii) otrzymujemy równanie

$$p_n'(t) = -\lambda p_n(t) + \lambda p_{n-1}(t).$$

To równanie można rozwiązać metodą uzmiennienia stałej: poszukujemy rozwiązania postaci $p_n(t) = c(t)\mathrm{e}^{-\lambda t}$. Zakładamy przy tym indukcyjnie, że $p_{n-1}(t) = (\lambda t)^{n-1}\mathrm{e}^{-\lambda t}/(n-1)!$ i mamy oczywisty warunek początkowy $p_n(0) = 0$. Stąd już łatwo dostać dowodzony wzór na p_n.

Na koniec zauważmy, że z postaci funkcji p_0 łatwo wywnioskować jaki ma rozkład zmienna $T_1 = \inf\{t : N(t) > 0\}$. Istotnie, $\mathbb{P}(T_1 > t) = \mathbb{P}(N(t) = 0) = p_0(t) = \mathrm{e}^{-\lambda t}$. □

Część II

Algorytmy Monte Carlo

Rozdział 3

Niezależne Monte Carlo

Metody Monte Carlo (MC) polegają na wykorzystanu generowanej komputerowo losowości do obliczania pewnych wielkości deterministycznych (do rozwiązywania zadań niekoniecznie związanych z rachunkiem prawdopodobieństwa). Duża część zastosowań (MC) wiąże się z obliczaniem całek lub sum. Typowe zadanie polega na obliczeniu wartości oczekiwanej

$$\theta = \mathbb{E}_\pi f(X) = \int_{\mathcal{X}} f(x)\pi(\mathrm{d}x),$$

gdzie X jest zmienną losową o rozkładzie prawdopodobieństwa π na przestrzeni $\mathcal{X}$, zaś $f : \mathcal{X} \to \mathbb{R}$. Zazwyczaj $\mathcal{X}$ jest podzbiorem wielowymiarowej przestrzeni euklidesowej lub jest zbiorem skończonym ale bardzo licznym (na przykład zbiorem pewnych obiektów kombinatorycznych). Jeśli $\mathcal{X} \subseteq \mathbb{R}^d$ i rozkład π jest opisany przez gęstość p to całka określająca wartość oczekiwaną jest zwykłą całką Lebesgue'a. W tym przypadku zatem możemy napisać

$$\theta = \int_{\mathcal{X}} f(x)p(x)\mathrm{d}x.$$

Równie ważny w zastosowaniach jest przypadek *dyskretnej* przestrzeni $\mathcal{X}$. Jeśli $p(x) = \mathbb{P}(X = x)$, to

$$\theta = \sum_{x\in\mathcal{X}} f(x)p(x).$$

W dalszym ciągu tego rozdziału utożsamiamy rozkład π z funkcją p. Zwróćmy uwagę, że przedstawiamy tu θ jako wartość oczekiwaną tylko dla wygody oznaczeń; w istocie *każda* całka, suma, (a także prawdopodobieństwo zdarzenia losowego) jest wartością oczekiwaną.

Na pierwszy rzut oka nie widać czym polega problem! Sumowanie wykonuje każdy kalkulator, całki się sprawnie oblicza numerycznie. Ale nie zawsze. Metody MC przydają się w sytuacjach gdy spotyka się z następującymi trudnościami (lub przynajmniej którąś z nich).

- Przestrzeń $\mathcal{X}$ jest ogromna. To znaczy wymiar d jest bardzo duży lub skończona przestrzeń zawiera astronomicznie dużą liczbę punktów.

- Rozkład prawdopodobieństwa π jest „skupiony" w małej części ogromnej przestrzeni $\mathcal{X}$.
- Nie ma podstaw do zakładania, że funkcja f jest w jakimkolwiek sensie „gładka" (co jest warunkiem stosowania standardowych numerycznych metod całkowania).
- Gęstość p rozkładu π jest znana tylko z dokładnością do stałej normującej. Innymi słowy, umiemy obliczać $\tilde{p}(x) = cp(x)$ ale nie znamy stałej $c = \int \tilde{p}(x)\mathrm{d}x$. Czasem zadanie polega właśnie na obliczeniu tej stałej (c jest nazwane „funkcją podziału" lub sumą statystyczną).

Prosta metoda MC (po angielsku nazywana bardziej brutalnie: *Crude Monte Carlo*, czyli CMC) nasuwa się sama. Należy wygenerować n niezależnych zmiennych losowych $X_1, \ldots, X_n$ o jednakowym rozkładzie π i za estymator wartości oczekiwanej wziąć średnią z próbki,

$$\hat{\theta}_n = \hat{\theta}_n^{\mathrm{CMC}} = \frac{1}{n}\sum_{i=1}^{n} f(X_i).$$

Mocne Prawo Wielkich Liczb (MPWL) gwarantuje, że $\hat{\theta}_n \to \theta$ prawie na pewno, gdy $n \to \infty$. W terminologii statystycznej, $\hat{\theta}_n$ jest *mocno zgodnym* estymatorem obliczanej wielkości. Zadanie wydaje się rozwiązane. Są jednak dwa zasadnicze kłopoty.

- Estymator $\hat{\theta}_n$ skonstruowany metodą CMC może się zbliżać do θ przerażająco wolno.
- Co robić, jeśli nie umiemy losować z rozkładu π?

3.1 Losowanie istotne

Zadziwiająco skutecznym sposobem na *oba* przedstawione wyżej kłopoty jest **losowanie istotne** (*Importance Sampling*, w skrócie IS). Przypuśćmy, że umiemy losować z rozkładu o gęstości q. Zauważmy, że

$$\theta = \int_{\mathcal{X}} \frac{p(x)}{q(x)} f(x)q(x)\mathrm{d}x = \mathbb{E}_q \frac{p(X)}{q(X)} f(X) = \mathbb{E}_q w(X)f(X),$$

gdzie $w(x) = p(x)/q(x)$. Piszemy tu wzory dla całek ale oczywiście dla sum jest tak samo. Niech $X_1, \ldots, X_n$ będą niezależnymi zmiennymi losowymi o jednakowym rozkładzie q,

$$\hat{\theta}_n = \hat{\theta}_n^{\mathrm{IS1}} = \sum_{i=1}^{n} W_i f(X_i), \tag{3.1.1}$$

gdzie

$$W_i = w(X_i) = \frac{p(X_i)}{q(X_i)}$$

traktujemy jako *wagi* wylosowanych punktów X_i. Z tego co wyżej powiedzieliśmy wynika, że $\mathbb{E}_q\hat{\theta}_n = \theta$ oraz $\hat{\theta}_n \to \theta$ prawie na pewno, gdy $n \to \infty$. Mamy więc estymator *nieobciążony* i *zgodny*. Milcząco założyliśmy, że $q(X_i) > 0$ w każdym wylosowanym punkcie, czyli, że $\{x : p(x) > 0\} \subseteq \{x : q(x) > 0\}$. Z tym zwykle nie ma wielkiego kłopotu. Ponadto musimy założyć, że umiemy obliczać funkcję w w każdym wylosowanym punkcie. Jeśli znamy tylko $\tilde{p}(x) = zp(x)$ a nie znamy stałej z to jest kłopot. Możemy tylko obliczyć $\tilde{W}_i = \tilde{p}(X_i)/q(X_i) = zW_i$. Używamy zatem nieco innej postaci estymatora IS, mianowicie

$$\hat{\theta}_n = \hat{\theta}_n^{\mathrm{IS2}} = \frac{\sum_{i=1}^n \tilde{W}_i f(X_i)}{\sum_{i=1}^n \tilde{W}_i}. \tag{3.1.2}$$

Możemy jeszcze zapisać estymator IS2 w zgrabnej formie

$$\hat{\theta}_n^{\mathrm{IS2}} = \sum_{i=1}^n \bar{W}_i f(X_i),$$

gdzie $\bar{W}_i = \tilde{W}_i / \sum_j \tilde{W}_j$ są *unormowanymi wagami*. Należy jednak pamiętać, że suma w mianowniku, $\sum_j \tilde{W}_j$, jest zmienną losową.

Ponieważ $\frac{1}{n}\sum_{i=1}^n \tilde{W}_i f(X_i) \to z\theta$ i $\frac{1}{n}\sum_{i=1}^n \tilde{W}_i \to z$, więc $\hat{\theta}_n^{\mathrm{IS2}} \to \theta$ (nieznany czynnik z skraca się). Estymator IS2 jest *zgodny*. Jest jednak, w przeciwieństwie do IS1, *obciążony* (wartość oczekiwana ilorazu nie jest równa ilorazowi wartości oczekiwanych). Przy okazji otrzymujemy zgodny i nieobciążony estymator stałej normującej z:

$$\hat{Z}_n = \frac{1}{n}\sum_{i=1}^n \tilde{W}_i. \tag{3.1.3}$$

Estymator IS2 jest znacznie częściej używany niż IS1. Oprócz uniezależnienia się od stałej normującej, jest jeszcze inny powód. Okazuje się, że (pomimo obciążenia) estymator IS2 może być w niektórych przykładach bardziej efektywny.

3.2 Dokładność i efektywność estymatorów MC

Naturalne jest żądanie, aby estymator był mocno zgodny, $\hat{\theta}_n \to \theta$ prawie na pewno przy $n \to \infty$. Znaczy to, że zbliżymy się dowolnie blisko aproksymowanej wielkości, gdy tylko dostatecznie długo przedłużamy symulacje. Chcielibyśmy jednak wiedzieć coś konkretniejszego, oszacować jak szybko *błąd aproksymacji* $\hat{\theta}_n - \theta$ maleje do zera i jakie n wystarczy do osiągnięcia wystarczającej dokładności. Przedstawimy teraz najprostsze i najczęściej stosowane w

praktyce podejście, oparte na tzw. asymptotycznej wariancji i konstrukcji asymptotycznych przedziałów ufności. W typowej sytuacji estymator $\hat{\theta}_n$ ma następującą własność, nazywaną *asymptotyczną normalnością*:

$$\sqrt{n}\left(\hat{\theta}_n - \theta\right) \to \mathrm{N}(0, \sigma^2), \qquad (n \to \infty)$$

w sensie zbieżności według rozkładu. W konsekwencji, dla ustalonej liczby $a > 0$,

$$\mathbb{P}\left(|\hat{\theta}_n - \theta| \leqslant \frac{z\sigma}{\sqrt{n}}\right) \to \Phi(z) - \Phi(-z) = 2\Phi(z) - 1, \qquad (n \to \infty),$$

gdzie Φ jest dystrybuantą rozkładu $\mathrm{N}(0,1)$. Często nie znamy *asymptotycznej wariancji* σ^2 ale umiemy skonstruować jej zgodny estymator $\hat{\sigma}_n^2$. Dla ustalonej „małej" dodatniej liczby α łatwo dobrać kwantyl rozkładu normalnego $z = z_{1-\alpha/2}$ tak, żeby $2\Phi(z) - 1 = 1 - \alpha$. Typowo $\alpha = 0.05$ i $z = z_{0.975} = 1.9600 \approx 2$. Otrzymujemy

$$\mathbb{P}\left(|\hat{\theta}_n - \theta| \leqslant \frac{z_{1-\alpha/2}\hat{\sigma}_n}{\sqrt{n}}\right) \to 1 - \alpha. \quad n \to \infty.$$

Ten wzór interpretuje się w praktyce tak: estymator $\hat{\theta}_n$ ma błąd nie przekraczający $z\hat{\sigma}_n/\sqrt{n}$ z prawdopodobieństwem około $1-\alpha$. W żargonie statystycznym $1-\alpha$ jest nazywane asymptotycznym poziomem ufności.

Chociaż opisame powyżej podejście ma swoje słabe strony, to prowadzi do prostego kryterium porównywania estymatorów. Przypuśćmy, że mamy dwa estymatory $\hat{\theta}_n^{\mathrm{I}}$ i $\hat{\theta}_n^{\mathrm{II}}$, oba asymptotycznie normalne, o asymptotycznej wariancji σ_{I}^2 i σ_{II}^2 odpowiednio. Przypuśćmy dalej, że dla obliczenia pierwszego z tych estymatorów generujemy n_{I}^2 punktów, zaś dla drugiego n_{I}^2. Błędy obu estymatorów na tym mym poziomie istotności są ograniczone przez, odpowiednio, $z\sigma_{\mathrm{I}}/\sqrt{n_{\mathrm{I}}}$ i $z\sigma_{\mathrm{II}}/\sqrt{n_{\mathrm{II}}}$. Przyrównując te wyrażenia do siebie dochodzimy do wniosku, że oba estymatory osiągają podobną dokładność, jeśli $n_{\mathrm{I}}/n_{\mathrm{II}} = \sigma_{\mathrm{II}}^2/\sigma_{\mathrm{I}}^2$. Liczbę

$$\mathrm{eff}(\hat{\theta}_n^{\mathrm{I}}, \hat{\theta}_n^{\mathrm{II}}) = \frac{\sigma_{\mathrm{II}}^2}{\sigma_{\mathrm{I}}^2}$$

nazywamy *względną efektywnością* (asymptotyczną). Czasami dobrze jest wybrać za „naturalny punkt odniesienia" estymator CMC i zdefiniować *efektywność* estymatora $\hat{\theta}_n$ o asymptotycznej wariancji σ^2 jako

$$\mathrm{eff}(\hat{\theta}_n) = \mathrm{eff}(\hat{\theta}_n, \hat{\theta}_n^{\mathrm{CMC}}) = \frac{\sigma_{\mathrm{CMC}}^2}{\sigma^2}.$$

Mówi się też, że jeśli wygenerujemy próbkę n punktów i obliczymy $\hat{\theta}_n$ to „efektywna liczność próbki" (ESS, czyli *effective sample size*) jest $n/\mathrm{eff}(\hat{\theta}_n)$. Tyle bowiem należałoby wygenerować punktów stosując CMC, żeby osiągnąć podobną dokładność.

Asymptotyczna normalność estymatora CMC wynika wprost z Centralnego Twierdzenia Granicznego (CTG). Istotnie, jeśli generujemy niezależnie $X_1, \ldots, X_n$ o jednakowym rozkładzie π, to

$$\begin{aligned}\sqrt{n}\left(\hat{\theta}_n^{\mathrm{CMC}} - \theta\right) &= \frac{1}{\sqrt{n}}\sum_{i=1}^{n}(f(X_i) - \theta) \\ &\to \mathrm{N}(0, \sigma^2_{\mathrm{CMC}}), \qquad (n \to \infty),\end{aligned}$$

gdzie

$$\sigma^2_{\mathrm{CMC}} = \mathrm{Var}_p f(X) = \int (f(x) - \theta)^2 \, p(x)\mathrm{d}x.$$

Zupełnie podobnie, dla losowania istotnego w formie (3.1.1) otrzymujemy asymptotyczną normalność i przy tym

$$\begin{aligned}\sigma^2_{\mathrm{IS1}} = \mathrm{Var}_q w(X) f(X) &= \mathbb{E}_q \frac{\left(f(X)p(X) - \theta q(X)\right)^2}{q(X)^2} \\ &= \int \frac{\left(f(x)p(x) - \theta q(x)\right)^2}{q(x)}\mathrm{d}x.\end{aligned}$$

Z tego wzorku widać, że estymator może mieć wariancję *zero* jeśli $q(x) \propto f(x)p(x)$. Niestety, żeby obliczyć $q(x)$ potrzebna jest znajomość współczynnika proporcjonalności, który jest równy... θ. Nie wszystko jednak stracone. Pozostaje ważna reguła heurystyczna:

Gęstość $q(x)$ należy tak dobierać, aby jej „kształt" był zbliżony do funkcji $f(x)p(x)$.

Dla drugiej wersji losowania istotnego, (3.1.2), asymptotyczna normalność wynika z następujących rozważań. Mamy mianowicie

$$\begin{aligned}\sqrt{n}\left(\hat{\theta}_n^{\mathrm{IS2}} - \theta\right) &= \frac{\frac{1}{\sqrt{n}}\sum_{i=1}^{n}(W_i f(X_i) - \theta W_i)}{\frac{1}{n}\sum_{i=1}^{n} W_i} \\ &\to \mathrm{N}(0, \sigma^2_{\mathrm{IS2}}), \qquad (n \to \infty),\end{aligned}$$

gdzie $\mathrm{N}(0, \sigma^2_{\mathrm{IS2}})$ jest granicznym rozkładem *licznika*. Istotnie, asymptotyczna normalność licznika wynika z CTG. Stosując PWL do mianownika wnioskujemy, że $\frac{1}{n}\sum_{i=1}^{n} W_i \to 1$ i wystarczy powołać się na lemat Słuckiego. Asymptotyczna wariancja σ^2_{IS2} jest dana wzorem

$$\begin{aligned}\sigma^2_{\mathrm{IS2}} &= \mathrm{Var}_q w(X)(f(X) - \theta) \\ &= \mathrm{Var}_q w(X) f(X) - 2\theta \mathrm{Cov}_q(w(X), w(X)f(X)) + \theta^2 \mathrm{Var}_q w(X) \\ &= \sigma^2_{\mathrm{IS1}} + \theta\left[-2\mathrm{Cov}_q(w(X), w(X)f(X)) + \theta \mathrm{Var}_q w(X)\right].\end{aligned}$$

Wyrażenie w kwadratowym nawiasie może być ujemne, jeśli jest duża dodatnia korelacja zmiennych $w(X)$ i $w(X)f(X)$. W tej sytuacji estymator IS2 jest lepszy od IS1. Okazuje się więc rzecz na pozór paradoksalna: dzielenie przez estymator jedynki może poprawić estymator. Poza tym oba estymatory IS1 i IS2 mogą mieć mniejszą (asymptotyczną) wariancję, niż CMC. Jeśli efektywność jest większa niż 100%, to używa się czasem określenia „estymator *superefektywny*".

3.3 Przykłady

Przytoczę 2 nietrywialne przykłady ilustrujące potęgę losowania istotnego. Pierwszy przykład dotyczy obliczania sum i zliczania obiektów kombinatorycznych. Zastosujemy tu metodę IS2.

3.3.1 Przykład (Nie-samo-przecinające się błądzenia)**.** Po angielsku nazywają się *Self Avoiding Walks*, w skrócie SAW. Niech $\mathbb{Z}^d$ będzie d-wymiarową kratą całkowitoliczbową. Mówimy, że ciąg $s = (0 = s_0, s_1, \ldots, s_k)$ punktów kraty jest SAW-em jeśli

- każde dwa kolejne punkty s_{i-1} i s_i sąsiadują ze sobą, czyli różnią się o ± 1 na dokładnie jednej współrzędnej,
- żadne dwa punkty nie zajmują tego samego miejsca, czyli $s_i \neq s_j$ dla $i \neq j$.

Zbiór wszystkich SAW-ów o k ogniwach w $\mathbb{Z}^d$ oznaczymy SAW_k^d, a dla d i k ustalonych w skrócie SAW. Przykład $s \in \mathrm{SAW}_{15}^2$ widać na Rysunku 3.1.

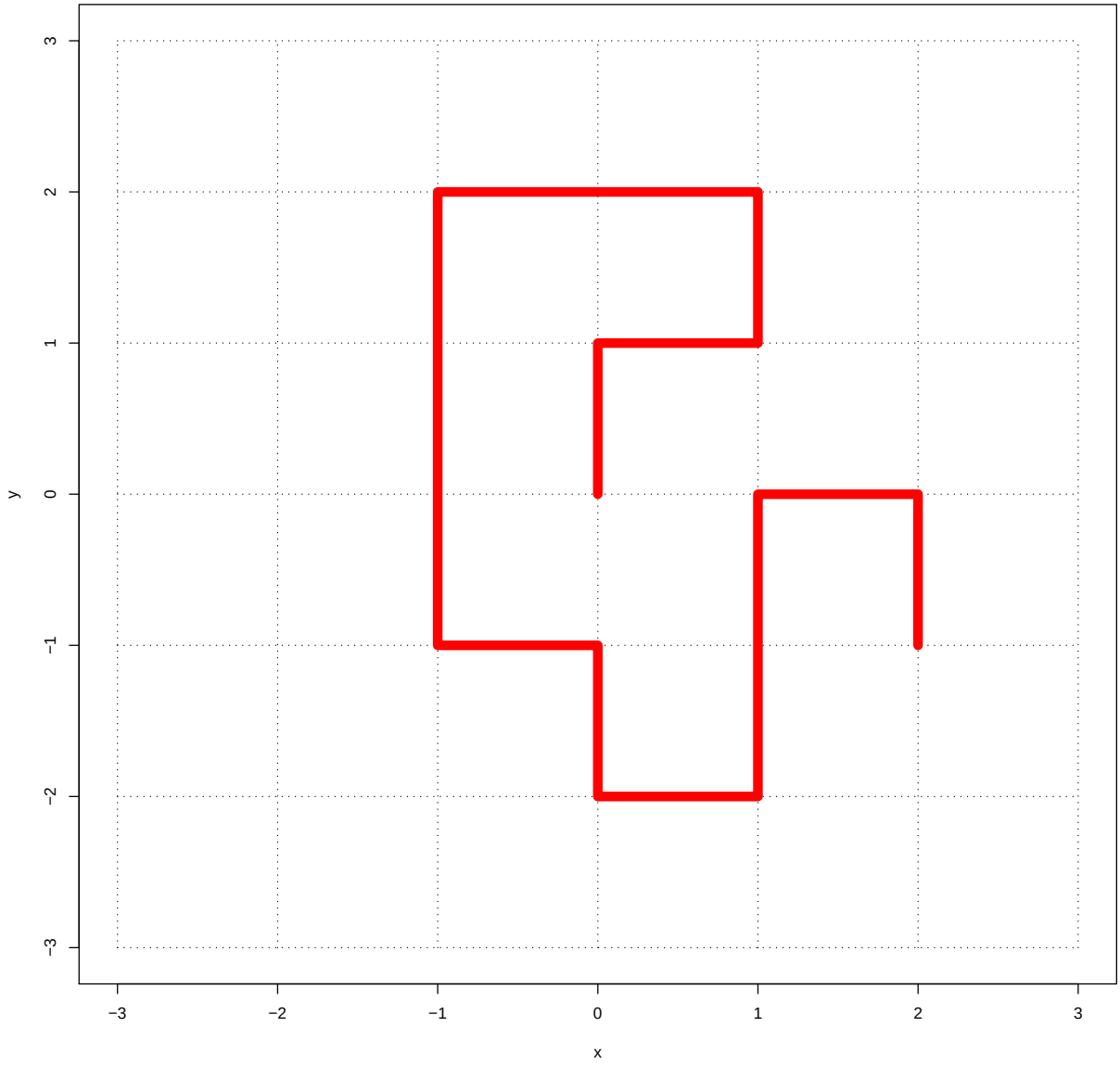
Natychmiast nasuwa się bardzo proste pytanie:

- Jak policzyć SAW-y, czyli obliczyć $|\mathrm{SAW}_k^d|$, liczbę elementów zbioru?

Zainteresujmy się teraz „losowo wybranym SAW-em". Rozumiemy przez to zmienną losową S o rozkładzie jednostajnym w zbiorze SAW_k^d, czyli taką, że $\mathbb{P}(S = s) = 1/L_{k,d}$ dla każdego $s \in \mathrm{SAW}_k^d$. W skrócie, $S \sim \mathrm{U}(\mathrm{SAW})$. Niech $\ell(s_k)$ oznacza odległość euklidesową końca SAW-a, czyli punktu s_k, od początku, czyli punktu 0. Na przykład dla łańcuszka widocznego na rysunku mamy $\ell(s_{15}) = \sqrt{5}$. Można zadać sobie pytanie, jaki jest średni kwadrat takiej odległości, czyli

- Jak obliczyć $\overline{\ell^2} = \overline{\ell^2_{d,k}} = \mathbb{E}\ell(S_k)^2$, przy założeniu, że $S \sim \mathrm{U}(\mathrm{SAW})$?

Można zastosować prostą metodę MC i eliminację. Niech WALK_k^d oznacza zbiór wszystkich „błądzeń", czyli ciągów $s = (0 = s_0, s_1, \ldots, s_k)$ niekoniecznie spełniających warunek „$s_i \neq s_j$ dla $i \neq j$". Oczywiście $|\mathrm{WALK}_k^d| = (2d)^k$ i metoda generowania „losowego błądzenia" (z rozkładu $\mathrm{U}(\mathrm{WALK})$) jest bardzo prosta: kolejno losujemy pojedyncze kroki, wybierając zawsze jedną z $2d$ możliwości. Żeby otrzymać „losowy SAW", stosujemy eliminację. Ten sposób pozwala w zasadzie estymować $\overline{\ell^2}$ (przez uśrednianie długości zaakceptowanych błądzeń) oraz SAW/WALK (przez zanotowanie frakcji akceptowanych błądzeń). Niestety, metoda jest bardzo nieefektywna, bo dla dużych k prawdopodobieństwo akceptacji szybko zbliża sie do zera.



Rysunek 3.1: Przykład SAW-a.

Metoda „wzrostu" zaproponowana przez Rosenbluthów polega na losowaniu kolejnych kroków błądzenia spośród „dopuszczalnych punktów", to znaczy punktów wcześniej nie odwiedzonych. W każdym kroku, z wyjątkiem pierwszego mamy co najwyżej $2d-1$ możliwości. W błądzeniu widocznym na rysunku kolejne kroki wybieraliśmy spośród:

$$4,3,3,3, \quad 2,3,2,2, \quad 3,2,3,3, \quad 2,1,3,2$$

możliwych. Nasz SAW został zatem wylosowany z prawdopodobieństwem

$$\frac{1}{4}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot \quad \frac{1}{2}\cdot\frac{1}{3}\cdot\frac{1}{2}\cdot\frac{1}{2}\cdot \quad \frac{1}{3}\cdot\frac{1}{2}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot \quad \frac{1}{2}\cdot\frac{1}{1}\cdot\frac{1}{3}\cdot\frac{1}{2}$$

Powiedzmy ogólniej, że przy budowaniu SAW-a $s = (0 = s_0, s_1, \ldots, s_k)$ mamy kolejno

$$m_1 = 2d, m_2, \ldots, m_k$$

możliwości (nie jest przy tym wykluczone, że w pewnym kroku *nie mamy żadnej możliwości*, $m_i = 0$). Używając terminologii i oznaczeń związanych z losowaniem istotnym powiemy, że

$$q(s) = \frac{1}{m_1}\cdot\frac{1}{m_2}, \cdots \frac{1}{m_k}.$$

jest gęstością instrumentalną dla $s \in \mathrm{SAW}$, gęstość docelowa jest stała, równa $p(s) = 1/Z$, zatem funkcja podziału jest po prostu liczbą SAW-ów: $Z = |\mathrm{SAW}|$. Wagi przypisujemy zgodnie ze wzorem $w(s) = m_1 \cdot n_2 \cdots m_k$ (jeśli wygenerowanie SAW-a się nie udało, $m_i = 0$ dla pewnego i, to waga jest zero). Niech teraz $S(1), \ldots, S(n)$ będą niezależnymi błądzeniami losowanymi metodą Rosenbluthów. Zgodnie z ogólnymi zasadami losowania istotnego,

$$\frac{\sum w(S(i))\ell(S(i))}{\sum w(S(i))} \quad \text{jest estymatorem } \overline{\ell^2},$$

$$\sum w(S(i))/n \quad \text{jest estymatorem } |\mathrm{SAW}|.$$

W ostatnim wzorze należy *uwzględniać* błądzenia o wadze zero, czyli „nieudane SAW-y". $\triangle$

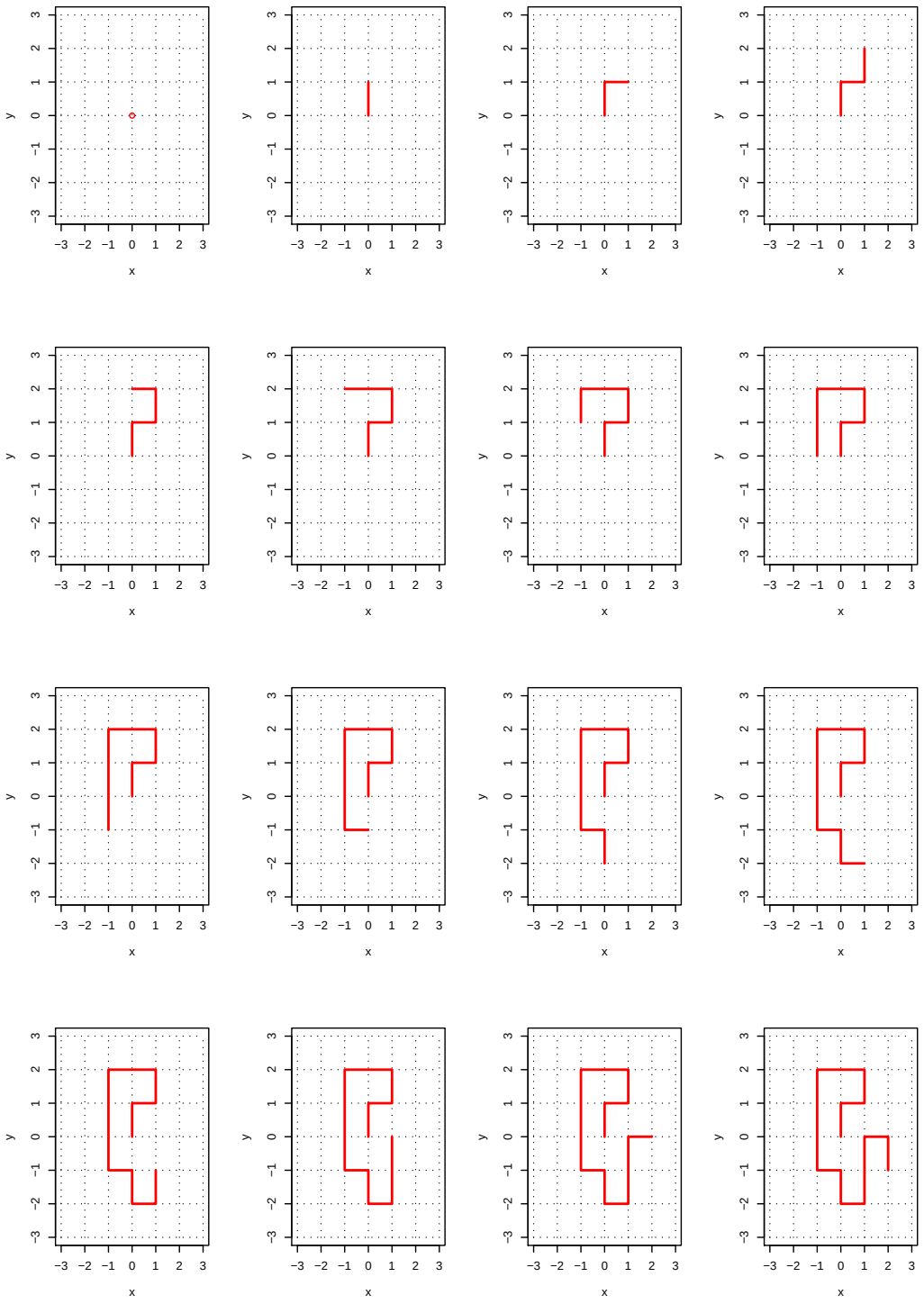
Drugi przykład dotyczy obliczania „bardzo małego" prawdopodobieństwa pewnego zdarzenia. Techniki MC stosowane typu problemach określa się terminem „symulacja zdarzeń rzadkich" (*rare event simulation*).

3.3.2 Przykład (Prawdopodobieństwo ruiny i wykładnicza zamiana miary)**.** Rozpatrzymy najprostszy model procesu opisującego straty i przychody w ubezpieczeniowej „teorii ryzyka". Niech $Y_1, Y_2, \ldots$ będą zmiennymi losowymi oznaczającymi *straty netto* (straty − przychody) towarzystwa ubezpieczeniowego w kolejnych okresach czasu. Założymy (co jest dużym uproszczeniem), że te zmienne są niezależne i mają jednakowy rozkład o gęstości $p(y)$. Tak zwana „nadwyżka ubezpieczyciela" na koniec n-go roku jest równa

$$u - S_n = u - \sum_{i=1}^{n} Y_i,$$

gdzie u jest rezerwą początkową. Interesuje nas prawdopodobieństwo zdarzenia, polegającego na tym, że $u - S_n < 0$ dla pewnego n. Mówimy wtedy (znowu w dużym uproszczeniu) o „ruinie ubezpieczyciela". Wygodnie jest przyjąć następujące oznaczenia i konwencje. Zmienna losowa

$$R = \begin{cases} \min\{n : S_n > u\} & \text{jeśli takie } n \text{ istnieje;} \\ \infty & \text{jeśli } S_n \leqslant u \text{ dla wszystkich } n \end{cases}$$



Rysunek 3.2: Generowanie SAW-a metodą „wzrostu”.

oznacza czas oczekiwania na ruinę, przy czym jeśli ruina nigdy nie nastąpi to ten czas uznajemy za nieskończony. Przy takiej umowie, prawdopodobieństwo ruiny możemy zapisać jako $\psi = \mathbb{P}_p(R < \infty)$. Wskaźnik p przy symbolu wartości oczekiwanej przypomina, że chodzi tu o „oryginalny" proces, dla którego $Y_i \sim p$.

Obliczenie ψ analitycznie jest możliwe tylko w bardzo specjalnych przykładach. Motody numeryczne istnieją, ale też nie są łatwe. Pokażemy sposób obliczania ψ metodą Monte Carlo, który stanowi jeden z najpiękniejszych, klasycznych, przykładów losowania istotnego. Przyjmiemy bardzo rozsądne założenie, że $\mathbb{E}_p Y_i < 0$. *Funkcję tworzącą momenty*, która odpowiada gęstości p określamy wzorem

$$M_p(t) = \mathbb{E}_p \mathrm{e}^{tY_i} = \int_{-\infty}^{\infty} \mathrm{e}^{ty} p(y) \mathrm{d}y.$$

Założymy, że ta funkcja przyjmuje wartości skończone przynajmniej w pewnym otoczeniu zera i istnieje takie $r > 0$, że

$$M_p(t) = 1.$$

Liczba r jest nazywana *współczynnikiem dopasowania* i odgrywa tu kluczową rolę.

Metoda *wykładniczej zamiany miary* jest specjalnym przypadkiem losowania istotnego. Żeby określić instrumentalny rozkład prawdopodobieństwa, połóżmy

$$q(y) = \mathrm{e}^{ry} p(y).$$

Z definicji współczynnika dopasowania wynika, że q jest gęstością prawdopodobieństwa, to znaczy $\int q(y)\mathrm{d}y = 1$. Generuje się ciąg $Y_1, Y_2, \ldots$ jednakowo rozłożonych zmiennych losowych o gęstości q. Dla utworzonego w ten sposób „instrumentalnego procesu" używać będziemy symboli $\mathbb{P}_q$ i $\mathbb{E}_q$. Zauważmy, że

$$\begin{aligned} \mathbb{E}_q Y_i = \int y q(y) \mathrm{d}y &= \int y \mathrm{e}^{ry} p(y) \mathrm{d}y \\ &= \frac{\mathrm{d}}{\mathrm{d}r} \int \mathrm{e}^{ry} p(y) \mathrm{d}y \\ &= M_p'(r) > 0, \end{aligned}$$

ponieważ funkcja tworząca momenty $M_p(\cdot)$ jest wypukła, $M_p'(0) < 0$ i $M_p(0) = M_p(r) = 1$. Po zamianie miary, proces $u - S_n$ na mocy Prawa Wielkich Liczb zmierza prawie na pewno do minus nieskończoności, a zatem ruina następuje z prawdopodobieństwem jeden, $\mathbb{P}_q(R < \infty) = 1$. Pokażemy, jak wyrazić prstwo ruiny dla oryginalnego procesu w terminach procesu instrumentalnego. Niech

$$\mathcal{R}_n = \{(y_1, \ldots, y_n) : y_1 \leqslant u, \ldots, y_1 + \cdots + y_{n-1} \leqslant u, y_1 + \cdots + y_{n-1} + y_n > u\},$$

Innymi słowy, zdarzenie $\{R = n\}$ zachodzi gdy $(Y_1, \ldots, Y_n) \in \mathcal{R}_n$. Mamy zatem

$$\begin{aligned}\mathbb{P}_p(R = n) &= \int \cdots \int_{\mathcal{R}_n} p(y_1) \cdots p(y_n) \mathrm{d}y_1 \cdots \mathrm{d}y_n \\ &= \int \cdots \int_{\mathcal{R}_n} \mathrm{e}^{-ry_1} q(y_1) \cdots \mathrm{e}^{-ry_n} q(y_n) \mathrm{d}y_1 \cdots \mathrm{d}y_n \\ &= \int \cdots \int_{\mathcal{R}_n} \mathrm{e}^{-r(y_1+\cdots+y_n)} q(y_1) \cdots q(y_n) \mathrm{d}y_1 \cdots \mathrm{d}y_n \\ &= \mathbb{E}_q \mathrm{e}^{rS_n} \mathbb{1}(R = n).\end{aligned}$$

Weźmy sumę powyższych równości dla $n = 1, 2, \ldots$ i skorzystajmy z faktu, że $\sum_{n=1}^{\infty} \mathbb{P}_q(R = n) = 1$. Dochodzimy do wzoru

$$\mathbb{P}_p(R < \infty) = \mathbb{E}_q \exp\{-rS_R\} = \mathrm{e}^{-ru} \mathbb{E}_q \exp\{-r(S_R - u)\}. \tag{3.3.3}$$

Ten fakt jest podstawą algorytmu Monte Carlo:

$\hat{\psi} := 0$; [$\hat{\psi}$ będzie estymatorem prawdopodobieństwa ruiny]
for $k := 1$ to m do
  begin
  $S := 0$;
  repeat
    Gen $Y \sim q$; $S := S + Y$;
    until $S > u$;
  $\hat{\psi} := \hat{\psi} + \exp\{-r(S - u)\}$;
  end
$\hat{\psi} := \mathrm{e}^{-ru} \hat{\psi}/m$

Algorytm jest prosty i efektywny. Trochę to zadziwiające, że w celu obliczenia prawdopodobieństwa ruiny generuje się proces dla którego ruina jest pewna. Po chwili zastanowienia można jednak zauważyć, że wykładnicza zamiana miary realizuje podstawową ideę losowania istotnego: rozkład instrumentalny „naśladuje" proces docelowy *ograniczony do zdarzenia* $\{R < \infty\}$.

Ciekawe, że wykładnicza zamiana miary nie tylko jest techniką Monte Carlo, ale jest też techniką *dowodzenia twierdzeń*! Aby się o tym przekonać, zauważmy, że „po drodze" udowodniliśmy nierówność $\psi \leqslant \mathrm{e}^{-ru}$. Wynika to z podstawowego wzoru (3.3.3) gdyż $\exp\{-r(S_R - u)\} \leqslant 1$. Jest to sławna nierówność Lundberga i wcale nie jest ona oczywista. △

3.4 Inne metody redukcji wariancji

W tym podrozdziale omówię niektóre metody redukcji wariancji dla klasycznych algorytmów Monte Carlo, w których losujemy próbki niezależnie, z jednakowego rozkładu. Wiemy, że dla takich algorytmów wariancja estymatora zachowuje się jak const/n. Jedyne, co możemy zrobić – to konstruować takie algorytmy, dla których stała „const" jest możliwie mała. Najważniejszą z tych metod faktycznie już poznaliśmy: jest to *losowanie istotne*, omówione w Podrozdziale 3.1. Odpowiedni wybór „rozkładu instrumentalnego" może zmniejszyć wariancję setki tysięcy razy! Istnieje jeszcze kilka innych, bardzo skutecznych technik. Do podstawowych należą: losowanie warstwowe, metoda zmiennych kontrolnych, metoda zmiennych antytetycznych. Możliwe są niezliczone modyfikacje i kombinacje tych metod. Materiał zawarty w tym rozdziale jest w dużym stopniu zaczerpnięty z monografii Ripleya ??

Losowanie warstwowe

Tak jak poprzednio, zadanie polega na obliczeniu wielkości

$$\theta = \mathbb{E}_\pi f(X) = \int_{\mathcal{X}} f(x)\pi(x)\mathrm{d}x,$$

gdzie rozkład prawdopodobieństwa i jego gęstość dla uproszczenia oznaczamy tym samym symbolem π. **Losowanie warstwowe** polega na tym, że rozbijamy przestrzeń X na sumę k rozłącznych podzbiorów (warstw),

$$\mathcal{X} = \bigcup_{h=1}^{k} A_h, \qquad A_h \cap A_g = \emptyset \quad (h \neq g),$$

i losujemy k niezależnych próbek, po jednej z każdej warstwy. Niech π_h będzie gęstością rozkładu *warunkowego* zmiennej X przy $X \in A_h$, czyli

$$\pi_h(x) = \frac{\pi(x)}{p_h}\mathbb{1}(x \in A_h), \qquad \text{gdzie} \quad p_h = \pi(A_h) = \int_{A_h} \pi(x)\mathrm{d}x.$$

Widać natychmiast, że $\int_B \pi_h(x)\mathrm{d}x = \pi(X \in B | X \in A_h)$. Rozbijamy teraz całkę, którą chcemy obliczyć:

$$\theta = \sum_h p_h \int f(x)\pi_h(x)\mathrm{d}x.$$

Możemy użyć następującego estymatora warstwowego:

$$\hat{\theta}_n^{\mathrm{stra}} = \sum_h \frac{p_h}{n_h} \sum_{i=1}^{n_h} f(X_{hi}), \tag{3.4.1}$$

gdzie

$$X_{h1}, \ldots, X_{hn_h} \sim_{\text{i.i.d}} \pi_h,$$

jest próbką rozmiaru n_h wylosowaną z h-tej warstwy ($h = 1, \ldots, k$). Jest to estymator nieobciążony,

$$\text{Var}\hat{\theta}_n^{\text{stra}} = \sum_h \frac{p_h^2}{n_h}\sigma_h^2, \tag{3.4.2}$$

gdzie $\sigma_h^2 = \text{Var}_\pi(f(X)|X \in A_h)$.

Porównajmy estymator warstwowy (3.4.1) ze „zwykłym" estymatorem

$$\hat{\theta}_n^{\text{CMC}} = \frac{1}{n}\sum_{i=1}^n f(X_i).$$

opartym na jednej próbce

$$X_1, \ldots, X_n \sim_{\text{i.i.d}} \pi.$$

Oczywiście

$$\text{Var}\hat{\theta}_n^{\text{CMC}} = \frac{1}{n}\sigma^2,$$

gdzie $\sigma^2 = \text{Var}_\pi f(X)$. Żeby porównanie było „uczciwe" rozważmy próbkę liczności $n = \sum_h n_h$. Jeśli decydujemy się na sumaryczną liczność próbki n, to możemy w różny sposób „rozdzielić" to n pomiędzy warstwami. To się nazywa *alokacja* próbki.

3.4.3 Stwierdzenie (Alokacja proporcjonalna)**.** *Jeżeli $n_h = p_h n$ dla $n = 1, \ldots, k$, to*

$$\text{Var}\hat{\theta}_n^{\text{stra}} = \frac{1}{n}\sum_h p_h\sigma_h^2 \leqslant \frac{1}{n}\sigma^2 = \text{Var}\hat{\theta}_n^{\text{CMC}}.$$

Dowód. Wyrażenie na wariancję wynika znatychmiast z podstawienia $n_h = p_h n$ w ogólnym wzorze (3.4.2). Nierówność wynika z następującej tożsamości:

$$\sigma^2 = \sum_h p_h\sigma_h^2 + \sum_h p_h(\theta_h - \theta)^2, \tag{3.4.4}$$

gdzie $\theta_h = \int f(x)\pi_h(x)\mathrm{d}x = \mathbb{E}_\pi(f(X)|X \in A_h)$. Jest to klasyczny wzór na dekompozycję wariancji na „wariancję wewnątrz warstw" (pierwszy składnik w (3.4.4)) i „wariancję pomiędzy warstwami" (drugi składnik w (3.4.4)). Zdefiniujemy zmienną losową H jako „numer warstwy do której wpada $X \sim \pi$", czyli $\{H = h\} = \{X \in A_h\}$. Możemy teraz wzór (3.4.4) przepisać w dobrze znanej postaci

$$\text{Var}\, f(X) = \mathbb{E}\text{Var}(f(X)|H) + \text{Var}\mathbb{E}(f(X)|H).$$

□

Wzór (3.4.4) podpowiada, jak dzielić przestrzeń na warstwy. Największy zysk w porównaniu z losowaniem nie-warstwowym jest wtedy, gdy „wariancja międzywarstwowa" jest dużo większa od „wewnątrzwarstwowej" Warstwy należy więc wybierać tak, żeby funkcja $\pi(c)f(x)$ była możliwie bliska stałej na każdym zbiorze A_h i różniła się jak najbardziej pomiędzy poszczególnymi zbiorami.

Stwierdzenie 3.4.3 pokazuje, że zawsze zyskujemy na losowaniu warstwowym, jeśli zastosujemy najprostszą, proporcjonalną alokację. Okazuje się, że nie jest to alokacja najlepsza. Jerzy Spława-Neyman odkrył prostą regułę wyznaczania alokacji optymalnej. Wychodzimy od wzoru (3.4.2) i staramy się zoptymalizować prawą stronę przy warunku $n = \sum_h n_h$. Poszukujemy zatem rozwiązania zadania

$$\sum_h \frac{p_h^2}{n_h}\sigma_h^2 = \min\,! \qquad \Big(\sum_h n_h - n = 0\Big) \tag{3.4.5}$$

(względem zmiennych n_h). Zastosujmy metodę mnożników Lagrange'a. Szukamy minimum

$$\mathcal{L} = \sum_h \frac{p_h^2}{n_h}\sigma_h^2 + \lambda\left(\sum_h n_h - n\right) = \min\,! \tag{3.4.6}$$

Obliczamy pochodną i przyrównujemy do zera:

$$\frac{\partial}{\partial n_h}\mathcal{L} = -\frac{p_h^2}{n_h^2}\sigma_h^2 + \lambda = 0. \tag{3.4.7}$$

Stąd natychmiast otrzymujemy rozwiązanie: $n_h \propto \sigma_h p_h$.

3.4.8 Stwierdzenie (Alokacja optymalna, J. Neyman)**.** *Estymator warstwowy ma najmniejszą wariancję jeśli alokacja n losowanych punktów jest dana wzorem*

$$n_h = \frac{\sigma_h p_h}{\sum_g \sigma_g p_g}, \qquad (\ h = 1, \ldots, k\ ).$$

Zignorowaliśmy tutaj niewygodne wymaganie, że liczności próbek n_h muszą być całkowite. Gdyby to wziąć pod uwagę, rozwiązanie stałoby się skomplikowane, a zysk praktyczny z tego byłby znikomy. W praktyce rozwiązanie neymanowskie zaokrągla się do liczb całkowitych i koniec. Ważniejszy jest inny problem. Żeby wyznaczyć alokację optymalną, trzeba znać nie tylko prawdopodobieństwa p_h ale i wariancje warstwowe σ_h^2. W praktyce często opłaca się wylosować wstępne próbki, na podstawie których *estymuje* się wariancje σ_h^2. Dopiero potem alokuje się dużą, roboczą próbkę rozmiaru n, która jest wykorzystana do obliczania docelowej całki.

Zmienne kontrolne

Idea zmiennych kontrolnych polega na rozbiciu docelowej całki (wartości oczekiwanej) na dwa składniki, z których jeden umiemy obliczyć analitycznie. Metodę Monte Carlo stosujemy do drugiego składnika. Przedstawmy wielkość obliczaną w postaci

$$\theta = \mathbb{E}_\pi f(X) = \int_{\mathcal{X}} f(x)\pi(x)\mathrm{d}x = \int_{\mathcal{X}} [f(x) - k(x)]\pi(x)\mathrm{d}x + \int_{\mathcal{X}} k(x)\pi(x)\mathrm{d}x.$$

Dążymy do tego, żeby całka funkcji k była obliczona analitycznie (lub numerycznie) a różnica $f-k$ była możliwie bliska stałej, bo wtedy wariancja metody Monte Carlo jest mała. Funkcję k lub zmienną losową $k(X)$ nazywamy zmienną kontrolną.

Dla uproszczenia połóżmy $Y = f(X)$, gdzie $X \sim \pi$. Przypuśćmy, że zmiennej kontrolnej będziemy szukać pośród kombinacji liniowych funkcji $k_1, \ldots, k_d$ o znanych całkach. Niech $Z_j = k_j(X)$ i $Z = (Z_1, \ldots, Z_d)^\top$. Przy tym stale pamiętajmy, że $X \sim \pi$ i będziemy pomijać indeks π przy wartościach oczekiwanych i wariancjach. Zatem

$$k(x) = \sum_{j=1}^{d} \beta_j k_j(x).$$

Innymi słowy poszukujemy wektora współczynników $\beta = (\beta_1, \ldots, \beta_d)^\top$ który minimalizuje wariancję $\mathrm{Var}(Y - \beta^\top Z)$. Wykorzystamy następujący standardowy wynik z teorii regresji liniowej.

3.4.9 Stwierdzenie. *Niech Y i Z będą zmiennymi losowymi o skończonych drugich momentach, wymiaru odpowiednio* 1 *i d. Zakładamy dodatkowo odwracalność macierzy wariancji-kowariancji* $\mathrm{VAR}(Z)$*. Kowariancję pomiędzy Y i Z traktujemy jako wektor wierszowy i oznaczamy przez* $\mathrm{COV}(Y, Z)$ *Spośród zmiennych losowych postaci $Y - \beta^\top Z$, najmniejszą wariancję otrzymujemy dla*

$$\beta_*^\top = \mathrm{COV}(Y, Z)\mathrm{VAR}(Z)^{-1}.$$

Ta najmniejsza wartość wariancji jest równa

$$\mathrm{Var}(Y - \beta_*^\top Z) = \mathrm{Var}Y - \mathrm{COV}(Y, Z)\mathrm{VAR}(Z)^{-1}\mathrm{COV}(Z, Y).$$

Przepiszmy tem wynik w bardziej sugestywnej formie. Niech $\mathrm{Var}Y = \sigma^2$. Można pokazać, że β_* maksymalizuje korelację pomiędzy Y i $\beta^\top Z$. Napiszmy

$$\begin{aligned} \varrho_{Y.Z} &= \max_\beta \mathrm{corr}(Y, \beta^\top Z) = \mathrm{corr}(Y, \beta_*^\top Z) \\ &= \sqrt{\frac{\mathrm{COV}(Y, Z)\mathrm{VAR}(Z)^{-1}\mathrm{COV}(Z, Y)}{\mathrm{Var}Y}}. \end{aligned} \tag{3.4.10}$$

Niech teraz $\hat{\theta}_n^{\text{contr}}$ będzie estymatorem zmiennych kontrolnych. To znaczy, że losujemy próbkę $X_1, \ldots, X_n \sim \pi$,

$$\hat{\theta}_n^{\text{contr}} = \frac{1}{n} \sum (Y_i - \beta_*^\top Z_i) + \beta_*^\top \mu,$$

gdzie $Z_i^\top = (Z_{i1}, \ldots, Z_{in}) = (f_1(X_i), \ldots, f_d(X_i))$, zaś $\mu_j = \mathbb{E} f_j(X)$ są obliczone analitycznie lub numerycznie $\mu = (\mu_1, \ldots, \mu_d)^\top$. Wariancja estymatora jest wyrażona wzorem

$$\text{Var}\hat{\theta}_n^{\text{contr}} = \frac{1}{n}\sigma^2(1 - \varrho_{Y.Z}^2),$$

co należy porównać z wariancją σ^2/n „zwykłego" estymatora.

Zróbmy podobną uwagę, jak w poprzednim podrozdziale. Optymalny wybór współczynników regresji wymaga znajomości wariancji i kowariancji, których obliczenie może być (i zazwyczaj jest!) trudniejsze niż wyjściowe zadanie obliczenia wartości oczekiwanej. Niemniej, można najpierw wylosować *wstępną* próbkę, wyestymować potrzebne wariancje i kowariancje (nawet niezbyt dokładnie) po to, żeby dla dużej, *roboczej* próbki skonstruować dobre zmienne kontrolne.

Wspomnijmy na koniec, że dobieranie zmiennej kontrolnej metodą regresji liniowej nie jest jedynym sposobem. W konkretnych przykładach można spotkać najróżniejsze, bardzo pomysłowe konstrukcje, realizujące podstawową ideę zmiennych kontrolnych.

Zmienne antytetyczne

Przypuśćmy, że estymujemy wielkość $\theta = \mathbb{E}_\pi f(X)$. Jeśli mamy dwie zmienne losowe X i X' o jednakowym rozkładzie π ale nie zakładamy ich niezależności, to

$$\text{Var}\frac{f(X) + f(X')}{2} = \frac{1}{2}\text{Var}(X)\,[1 + \text{corr}(f(X), f(X'))] = \frac{1}{2}\sigma^2(1 + \varrho).$$

Jeśli $\varrho = \text{corr}(f(X), f(X')) < 0$, to wariancja w powyższym wzorze jest *mniejsza* niż $\sigma^2/2$, czyli mniejsza niż w przypadku niezależności X i X'. To sugeruje, żeby zamiast losować *niezależnie* n zmiennych $X_1, \ldots, X_n$, wygenerować *pary zmiennych ujemnie skorelowanych.* Załóżmy, że $n = 2k$ i mamy k niezależnych par $(X_1, X_1'), \ldots, (X_k, X_k')$, a więc łącznie n zmiennych. Porównajmy wariancję „zwykłego" estymatora $\hat{\theta}_n^{\text{CMC}}$ i estymatora $\hat{\theta}_n^{\text{ant}}$ wykorzystującego ujemne skorelowanie par:

$$\hat{\theta}_n^{\text{CMC}} = \frac{1}{n}\sum_{i=1}^{n} f(X_i), \qquad \text{Var}\hat{\theta}_n^{\text{CMC}} = \frac{\sigma^2}{n}$$

$$\hat{\theta}_n^{\text{ant}} = \frac{1}{n}\sum_{i=1}^{n/2} [f(X_i) = f(X_i')], \qquad \text{Var}\hat{\theta}_n^{\text{ant}} = \frac{\sigma^2}{n}(1 + \varrho).$$

Wariancja estymatora $\hat{\theta}_n^{\text{ant}}$ jest tym mniejsza, im ϱ bliższe -1 (im bardziej ujemnie skorelowane są pary). Standardowym sposobem generowania ujemnie skorelowanych par jest odwracanie dystrybuanty z użyciem „odwróconych loczb losowych".

3.4.11 Stwierdzenie. *Jeśli* $h :]0,1[\to \mathbb{R}$ *jest funkcją monotoniczną różną od stałej,* $\int_0^1 h(u)^2 \mathrm{d}u < \infty$ *i* $U \sim \mathrm{U}(0,1)$*, to*

$$\mathrm{Cov}(h(U), h(1-U)) < 0.$$

Dowód. Bez straty ogólności załóżmy, że h jest niemalejąca. Niech $\mu = \mathbb{E}h(U) = \int_0^1 h(u)\mathrm{d}u$ i $t = \sup\{u : h(1-u) > \mu\}$. Łatwo zauważyć, że $0 < t < 1$. Zauważmy, że

$$\begin{aligned}
\mathrm{Cov}(h(U), h(1-U)) &= \mathbb{E}h(U)[h(1-U) - \mu] \\
&= \int_0^1 h(u)[h(1-u) - \mu]\mathrm{d}u \\
&= \int_0^t h(u)[h(1-u) - \mu]\mathrm{d}u + \int_t^1 h(u)[h(1-u) - \mu]\mathrm{d}u \\
&< \int_0^t h(t)[h(1-u) - \mu]\mathrm{d}u + \int_t^1 h(t)[h(1-u) - \mu]\mathrm{d}u \\
&= h(t) \int_0^1 [h(1-u) - \mu]\mathrm{d}u = 0,
\end{aligned}$$

ponieważ dla $0 < u < t$ mamy $h(1-u) - \mu > 0$ i dla $t < u < 1$ mamy $h(1-u) - \mu \geqslant 0$. □

Przypomnijmy, że dla dowolnej dystrybuanty G określamy uogólnioną funkcję odwrotną G^- następującym wzorem:

$$G^-(u) = \inf\{x : G(x) \geqslant u\}.$$

Natychmiast wnioskujemy ze Stwierdzenia 3.4.11, że $\mathrm{Cov}(G^-(U), G^-(1-U)) < 0$. W ten sposób możemy produkować ujemnie skorelowane pary zmiennych o zadanej dystrybuancie. Okazuje się, że są to najbardziej ujemnie skorelowane pary. Jeśli $X \sim G$ i $X' \sim G$, to

$$\mathrm{Cov}(X, X') \geqslant \mathrm{Cov}(G^-(U), G^-(1-U)).$$

Powyższy fakt ma oczywiste znaczenie z punktu widzenia metod Monte Carlo i wynika z ogólniejszego twierdzenia, podanego w następujcym podrozdziale (Twierdzenie 3.5.1).

3.5 Zadania i uzupełnienia

Zadania teoretyczne

3.1 Zadanie. Udowodnić Stwierdzenie 3.4.9.

3.2 Zadanie. Udowodnić wzór 3.4.10.

Ćwiczenia: obliczanie całek, redukcja wariancji

3.1 Ćwiczenie. Oblicz $\theta = \mathbb{P}(Z > 4)$, gdzie $Z \sim \mathrm{N}(0,1)$. Zastosuj dwa schematy obliczeń MC:

- Prymitywna metoda Monte Carlo: bezpośrednio z definicji $\hat{\theta}_n = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(Z_i > 4)$, gdzie $Z_1, \ldots, Z_n \sim_{\text{i.i.d.}} \mathrm{N}(0,1)$.
- Losowanie ważone z rozkładem instrumentalnym o gęstości $q(z) = \exp[-(z-4)]\mathbb{1}(z > 4)$ (jest to przesunięty rozkład wykładniczy).

Podać asymptotyczne przedziały ufności dla obu metod. Porównac z dokładnym wynikiem obliczonym przez `pnorm()`.

3.2 Ćwiczenie. Obliczyć kilkoma wariantami metody MC całkę

$$\theta = \int_2^\infty \frac{\mathrm{d}x}{\pi(1+x^2)}.$$

Oszacować na podstawie symulacji lub/i obliczyć teoretycznie wariancję σ^2 i „typowy błąd estymacji" 2σ (gdzie $\mathbb{P}(|\hat{\theta}_n - \theta| \leqslant 2\sigma/\sqrt{n}) \simeq 0.95$)). Porównać.

- Estymator MC oparty bezpośrednio na spostrzeżeniu, że

  $$\theta = \mathbb{P}(X > 2)$$

  dla $X \sim$ Cauchy.

- Estymator wykorzystujący dodatkowo symetrię,

  $$\theta = \frac{1}{2}\mathbb{P}(|X| > 2)$$

  dla $X \sim$ Cauchy.

- Estymator wykorzystujący przejście do dopełnienia i „proste" MC,

  $$\theta = \frac{1}{2} - \int_0^2 \frac{\mathrm{d}x}{\pi(1+x^2)} = \frac{1}{2} - 2\mathbb{E}\frac{1}{\pi(1+U^2)},$$

  dla $U \sim \mathrm{U}(0,2)$.

- Estymator wykorzystujący przejście do dopełnienia i metodę zmiennych antytetycznych,

  $$\theta = \frac{1}{2} - \int_0^2 \frac{\mathrm{d}x}{\pi(1+x^2)} = \frac{1}{2} - \mathbb{E}\frac{1}{\pi(1+U^2)} - \mathbb{E}\frac{1}{\pi(1+(2-U)^2)},$$

  dla $U \sim \mathrm{U}(0,2)$.

- Estymator wykorzystujący „chytry chwyt”,

$$\theta = \int_0^{1/2} \frac{\mathrm{d}y}{\pi(1+y^2)},$$

gdzie równość wynika z podstawienia $y = 1/x$.

- Estymator wykorzystujący metodę zmiennych kontrolnych, opartą na przybliżeniu

$$\frac{1}{1+x^2} \quad \text{przez wielomian} \quad 1 - a_2x^2 + a_4x^4.$$

Zakładamy, że umiemy obliczyć analitycznie $\int_0^2 x^2\mathrm{d}x$ i $\int_0^2 x^4\mathrm{d}x$ Znaleźć „dobre” współczynniki a_2 i a_4 symulacyjnie, dopasowując model regresji liniowej.

3.3 Ćwiczenie. Zadanie o ruinie z dwiema barierami. Niech $Y = Y_1, \ldots, Y_n, \ldots$ będą niezależnymi zmiennymi losowymi o jednakowym rozkładzie i $S_n = Y_1 + \cdots + Y_n$. Dla ustalonych $u, b > 0$, niech $R = \min\{n : S_n > u \text{ lub } S_n < -b\}$. Obliczyć $\theta = \mathbb{P}(S_R > u)$. Zakładamy, że $\mathbb{E}Y < 0$.

- Obliczyć θ „prostą” metodą MC.
- Obliczyć θ metodą losowania istotnego z wykładniczą zamianą miary.
- Dla obu metod oszacować liczbę m doświadczeń potrzebnych do osiągnięcia „dokładności względnej” 0.1, czyli $\mathbb{P}(|\hat{\theta}_m - \theta| \leqslant 0.1\theta) \simeq 0.95)$.

Jednostronna wersja tego zadania z $b = \infty$ jest opisana w skrypcie.

a) Przyjąć $u = b = 10$, $Y \sim \mathrm{N}(-\mu, 1)$, $\mu = 0$, $\mu = 0.1$, $\mu = 0.5$, $\mu = 1$, $\mu = 2$.

b) Przyjąć, że Y ma rozkład dwupunktowy: $\mathbb{P}(Y = 1) = p$, $\mathbb{P}(Y = -1) = 1 - p$, gdzie $p < 1/2$ (poeksperymentować z różnymi wartościami p).

W tym drugim przypadku można wyprowadzić dokładny wzór na ψ i porównać z symulacjami.

Ograniczenia Frecheta

3.5.1 Twierdzenie. *Jeżeli $X \sim F$, $Y \sim G$ oraz $\mathbb{E}X^2 < \infty$, $\mathbb{E}Y^2 < \infty$ to*

$$\mathrm{Cov}(F^-(U), G^-(1-U)) \leqslant \mathrm{Cov}(X, Y) \leqslant \mathrm{Cov}(F^-(U), G^-(U)).$$

Twierdzenie 3.5.1 wynika z trzech poniższych faktów. Każdy z nich jest sam w sobie interesujący. Zaczniemy od sławnego wyniku Frecheta.

3.5.2 Twierdzenie (Ograniczenia Frecheta)**.** *Jeżeli* $\mathbb{P}(X \leqslant x) = F(x)$, $\mathbb{P}(Y \leqslant x) = G(y)$ *oraz* $\mathbb{P}(X \leqslant x, Y \leqslant y) = H(x,y)$ *oznaczają, odpowiednio, dystrybuanty brzegowe oraz łączną dystrybuantę dwóch zmiennych losowych to*

$$\max(0, F(x) + G(y) - 1) \leqslant H(x,y) \leqslant \min(F(x), G(y)).$$

Istnieją rozkłady łączne o brzegowych F *i* G*, dla których jest osiągane ograniczenie dolne i ograniczenie górne.*

Dowód. Ograniczenie górne wynika z oczywistych nierówności

$$\begin{aligned}\mathbb{P}(X \leqslant x, Y \leqslant y) &\leqslant \mathbb{P}(X \leqslant x) = F(x),\\ \mathbb{P}(X \leqslant x, Y \leqslant y) &\leqslant \mathbb{P}(Y \leqslant y) = G(y).\end{aligned}$$

Ograniczenie dolne jest równie proste:

$$\begin{aligned}\mathbb{P}(X \leqslant x, Y \leqslant y) &= \mathbb{P}(X \leqslant x) - \mathbb{P}(X \leqslant x, Y > y)\\ &\geqslant \mathbb{P}(X \leqslant x) - \mathbb{P}(Y > y) = F(x) - [1 - G(y)].\end{aligned}$$

□

Pozostała do pokazania osiągalność. Następujący lemat jest jednym z piękniejszych przykładów „symulacyjnego punktu widzenia" w rachunku prawdopodobieństwa.

3.5.3 Lemat. *Jeśli* $U \sim \mathrm{U}(0,1)$ *to*

$$\begin{aligned}\mathbb{P}(F^-(U) \leqslant x, G^-(U) \leqslant y) &= \min(F(x), G(y));\\ \mathbb{P}(F^-(U) \leqslant x, G^-(1-U) \leqslant y) &= \max(0, F(x) + G(y) - 1).\end{aligned}$$

Dowód. Pierwsza równość jest oczywista:

$$\begin{aligned}\mathbb{P}(F^-(U) \leqslant x, G^-(U) \leqslant y) &= \mathbb{P}(U \leqslant F(x), U \leqslant G(y))\\ &= \min(F(x), G(y)).\end{aligned}$$

Druga równość też jest oczywista:

$$\begin{aligned}\mathbb{P}(F^-(U) \leqslant x, G^-(1-U) \leqslant y) &= \mathbb{P}(U \leqslant F(x), 1 - U \leqslant G(y))\\ &= \mathbb{P}(1 - G(y) \leqslant U \leqslant F(x))\\ &= \max(0, F(x) - [1 - G(y)]).\end{aligned}$$

□

Głębokie twierdzenie Frecheta składa się więc z kilku dość oczywistych spostrzeżeń. Zeby udowodnić Twierdzenie 3.5.1 potrzeba jeszcze jednego ciekawego lematu.

3.5.4 Lemat. *Niech $F(x)$, $G(y)$ i $H(x,y)$ oznaczają, odpowiednio, dystrybuanty brzegowe oraz łączną dystrybuantę zmiennych losowych X i Y. Jeśli $\mathbb{E}X^2 < \infty$, $\mathbb{E}Y^2 < \infty$ to*

$$\mathrm{Cov}(X,Y) = \iint [H(x,y) - F(x)G(y)]\mathrm{d}x\mathrm{d}y.$$

Dowód. Niech (X_1,Y_1) i (X_2,Y_2) będą niezależnymi parami o jednakowym rozkładzie takim jak para (X,Y). Wtedy

$$\begin{aligned}
2\mathrm{Cov}(X,Y) &= \mathbb{E}(X_1 - X_2)(Y_1 - Y_2) \\
&= \mathbb{E}\iint [\mathbb{1}(X_1 \leqslant x) - \mathbb{1}(X_2 \leqslant x)][\mathbb{1}(Y_1 \leqslant y) - \mathbb{1}(Y_2 \leqslant y)]\mathrm{d}x\mathrm{d}y \\
&= \iint \mathbb{E}[\mathbb{1}(X_1 \leqslant x) - \mathbb{1}(X_2 \leqslant x)][\mathbb{1}(Y_1 \leqslant y) - \mathbb{1}(Y_2 \leqslant y)]\mathrm{d}x\mathrm{d}y \\
&= \iint [\mathbb{P}(X_1 \leqslant x, Y_1 \leqslant y) + \mathbb{P}(X_2 \leqslant x, Y_2 \leqslant y) \\
&\qquad\qquad - \mathbb{P}(X_1 \leqslant x, Y_2 \leqslant y) - \mathbb{P}(X_2 \leqslant x, Y_1 \leqslant y)]\mathrm{d}x\mathrm{d}y \\
&= \iint [2\mathbb{P}(X \leqslant x, Y \leqslant y) - 2\mathbb{P}(X \leqslant x)\mathbb{P}(Y \leqslant y)]\mathrm{d}x\mathrm{d}y.
\end{aligned}$$

□

Rozdział 4

Markowowskie Monte Carlo, MCMC

4.1 Co to jest MCMC ?

W pewnych sytuacjach okazuje się, że wygenerowanie zmiennej losowej X z interesującego nas rozkładu prawdopodobieństwa π jest praktycznie niemożliwe. Wyobraźmy sobie, że π jest bardzo skomplikowanym rozkładem na „ogromnej", wielowymiarowej przestrzeni $\mathcal{X}$. Zazwyczaj ten rozkład jest dany poprzez podanie funkcji proporcjonalnej do gęstości, $\tilde{p} \propto p$, ale bez znajomości stałej normującej $z = \int \tilde{p}$. („Ogromna" przestrzeń $\mathcal{X}$ może być zbiorem skończonym, ale bardzo licznym. Czasami rozkład π jest jednostajny, o gęstości $p \propto 1$, ale jego nośnik jest bardzo „skomplikowanym" zbiorem.) Metody typu eliminacji/akceptacji mogą zawieść. Skrót MCMC oznacza *Markov Chain Monte Carlo*, czyli po polsku algorytmy MC wykorzystujące łańcuchy Markowa. Podstawowa idea jest taka: jeśli nie umiemy generować zmiennej losowej X o rozkładzie π to zadowolimy się generowaniem ciągu zmiennych losowych $X_0, X_1, \ldots, X_n, \ldots$, który w pewnym sensie zbliża się, zmierza do rozkładu π.

W moich wykładach ścisłe przedstawienie teorii MCMC jest możliwe tylko w ograniczonym zakresie. W obecnym rozdziale skupię się na głównych ideach MCMC, przedstawię podstawowe algorytmy i kilka motywujących przykładów, pokażę wyniki symulacji, ale niemal nic nie udowodnię. Sprobuję to częściowo naprawić w Rozdziale 6, który w całości będzie poświęcony łańcuchom Markowa i algorytmom MCMC na skończonej przestrzeni $\mathcal{X}$. W tej specjalnej sytuacji podam dowody (a przynajmniej szkice dowodów) podstawowych twierdzeń. Zastosowania MCMC w przypadku „ciągłej" przestrzeni $\mathcal{X}$ (powiedzmy, $\mathcal{X} \subseteq \mathbb{R}^d$) są przynajmniej równie ważne. Zobaczymy to na paru przykładach. Algorytmy MCMC pracujące na przestrzeni ciągłej są w zasadzie takie same jak te w przypadku przestrzeni skończonej, ale ich analiza robi się trudniejsza, wymaga więcej abstrakcyjnej matematyki – i w rezultacie wykracza poza zakres mojego skryptu. Ograniczę się w tej materii do kilku skromnych uwag. Bardziej systematyczne, a przy tym dość przystępne przedstawienie ogólnej teorii można znaleźć w [14], [5] lub [9].

Łańcuchy Markowa

Klasyczne metody MCMC, jak sama nazwa wskazuje, opierają się na generowaniu łańcucha Markowa. Co prawda, rozwijają się obecnie bardziej wyrafinowane metody MCMC (zwane adaptacyjnymi), które wykorzystują procesy niejednorodne a nawet nie-markowowskie. Na razie ograniczymy się do rozpatrzenia sytuacji, gdy generowany ciąg zmiennych losowych X_n jest *jednorodnym łańcuch liczbę krokówem Markowa* na przestrzeni $\mathcal{X}$. Jeśli $\mathcal{X}$ jest zbiorem skończonym lub przeliczalnym, możemy posługiwać się Definicją 2.2.1, w ogólnym przypadku – Definicją 2.2.5. Przypomnijmy oznaczenie *prawdopodobieństw przejścia* łańcucha: dla $x \in \mathcal{X}$ oraz $B \subseteq \mathcal{X}$,

$$\mathbb{P}(X_{n+1} \in B | X_n = x) = P(x, B).$$

W przypadku przestrzeni skończonej wygodniej posługiwać się *macierzą przejścia* o elementach

$$\mathbb{P}(X_{n+1} = x' | X_n = x) = P(x, x').$$

Metoda generowania łańcuchów Markowa jest dość oczywista i sprowadza się do wzorów (2.2.3) w przypadku przestrzeni dyskretnej i (2.2.7) w przypadku ogólnym.

Rozkład stacjonarny

Niech π oznacza docelowy rozkład prawdopodobieństwa. Chcemy tak generować łańcuch X_n, czyli tak wybrać prawdopodobieństwa przejścia P, aby uzyskać zbieżność do rozkładu π. Spróbujemy uściślić co to znaczy, w jakim sensie rozumiemy zbieżność.

4.1.1 Definicja. *Mówimy, że π jest **rozkładem stacjonarnym** (lub rozkładem równowagi) łańcucha Markowa o prawdopodobieństwach przejścia P, jeśli dla każdego (mierzalnego) zbioru $B \subseteq \mathcal{X}$ mamy*

$$\pi(B) = \int_{\mathcal{X}} \pi(\mathrm{d}x) P(x, B).$$

W przypadku dyskretnej przestrzeni $\mathcal{X}$ równoważne sformułowanie jest takie: dla każdego stanu x',

$$\pi(x') = \sum_{x \in \mathcal{X}} \pi(x) P(x, x').$$

Utożsamiając P z macierzą a π z wektorem, zapiszemy powyższą równość w postaci $\pi^\top = \pi^\top P$. Ten krótki zapis będziemy stosowali (umownie) również w ogólnej sytuacji. Jeśli rozkład początkowy jest rozkładem stacjonarnym, $\mathbb{P}(X_0 \in \cdot) = \pi(\cdot)$, to dla każdego n mamy $\mathbb{P}(X_n \in \cdot) = \pi(\cdot)$. Co więcej, w takiej sytuacji łączny rozkład zmiennych $X_n, X_{n+1}, \ldots$ jest taki sam, jak rozkład zmiennych $X_0, X_1, \ldots$. Mówimy, że łańcuch jest w położeniu równowagi lub, ze jest *procesem stacjonarnym*. To uzasadnia nazwę rozkładu stacjonarnego. Oczywiście, łańcuchy generowane przez algorytmy MCMC nie są w stanie równowagi, bo z

założenia nie umiemy wygenerować $X_0 \sim \pi$. Generujemy X_0 z pewnego innego rozkładu ν, nazywanego rozkładem początkowym. Gdy zajdzie potrzeba, żeby uwidocznić zależność od rozkładu początkowego, będziemy używali oznaczeń $\mathbb{P}_\nu(\cdots)$ i $\mathbb{E}_\nu(\cdots)$. Przeważnie start jest po prostu deterministyczny, czyli ν jest rozkładem skupionym w pewnym punkcie $x \in \mathcal{X}$. Piszemy wtedy $\mathbb{P}_x(\cdots)$ i $\mathbb{E}_x(\cdots)$.

Twierdzenia graniczne dla łańcuchów Markowa

Naszkicujemy podstawowe twierdzenia graniczne dla łańcuchów Markowa. Dokładniejsze sformułowania i niektóre dowody pojawią się w następnym rozdziale i będą ograniczone do przypadku skończonej przestrzeni $\mathcal{X}$. Bardziej dociekliwych Czytelników muszę odesłać do przeglądowych prac [14, **?**] i skryptu Geyera [5].

Jeśli π jest rozkładem stacjonarnym, to przy pewnych założeniach uzyskuje się tak zwane *Słabe Twierdzenie Ergodyczne* (STE). Jego tezą jest zbieżność rozkładów prawdopodobieństwa zmiennych losowych X_n do π w następującym sensie: dla dowolnego (mierzalnego) zbioru $B \subseteq \mathcal{X}$ i dowolnego rozkładu początkowego ν mamy

$$\mathbb{P}_\nu(X_n \in B) \to \pi(B) \qquad (n \to \infty). \tag{4.1.2}$$

Dla skończonej przestrzeni $\mathcal{X}$ równoważne jest stwierdzenie, że dla każdego $x \in \mathcal{X}$,

$$\mathbb{P}_\nu(X_n = x) \to \pi(x) \qquad (n \to \infty).$$

STE dla skończonej przestrzeni $\mathcal{X}$ udowodnimy w rozdziale (Twierdzenie 6.3.8). Na razie poprzestańmy na następującym prostym spostrzeżeniu.

Uwaga. Jeżeli zachodzi teza STE dla pewnego rozkładu granicznego π_∞, czyli $P^n(x, B) \to \pi_\infty(B)$ dla dowolnych $x \in \mathcal{X}$, $B \subseteq \mathcal{X}$, to π_∞ jest rozkładem stacjonarnym. W istocie, wystarczy przejść do granicy w równości

$$\begin{array}{ccc} P^{n+1}(x,B) & = & \int P^n(x,\mathrm{d}x')P(x',B) \\ \downarrow & & \downarrow \\ \pi_\infty(B) & & \int \pi_\infty(\mathrm{d}x')P(x',B). \end{array}$$

Co więcej, π_∞ jest *jedynym* rozkładem stacjonarnym.

Uwaga. W teorii łańcuchów Markowa rozważa się różne pojęcia zbieżności rozkładów. Zauważmy, że w powyżej przytoczonej tezie STE oraz w Uwadze 4.1 mamy do czynienia z silniejszym rodzajem zbieżności, niż poznana na rachunku prawdopodobieństwa zbieżność słaba (według rozkładu), oznaczana $\to_{\mathrm{d}}$.

Sformułujemy teraz odpowiednik *Mocnego Prawa Wielkich Liczb* (PWL) dla łańcuchów Markowa. Rozważmy funkcję $f : \mathcal{X} \to \mathbb{R}$. Wartość oczekiwana funkcji f względem rozkładu π jest określona jako całka

$$\mathbb{E}_\pi f = \int_{\mathcal{X}} f(x)\pi(\mathrm{d}x).$$

Jeżeli rozkład π ma gęstość p względem miary Lebesgue'a to jest to „zwyczajna całka",

$$\mathbb{E}_\pi f = \int_{\mathcal{X}} f(x)p(x)\mathrm{d}x.$$

W przypadku dyskretnej przestrzeni $\mathcal{X}$ jest to suma

$$\mathbb{E}_\pi f = \sum_{x\in\mathcal{X}} f(x)\pi(x).$$

Jeśli załóżmy π jest rozkładem stacjonarnym łańcucha Markowa X_n, to możemy oczekiwać, że zachodzi zbieżność średnich do granicznej wartości oczekiwanej,

$$\frac{1}{n}\sum_{i=0}^{n-1} f(X_i) \longrightarrow \mathbb{E}_\pi f \qquad (n\to\infty) \tag{4.1.3}$$

z prawdopodobieństwem 1. W istocie, można udowodnić (4.1.3) przy pewnych dodatkowych założeniach. Mówimy wtedy, że zachodzi PWL lub Mocne Twierdzenie Ergodyczne. Ze względu na zastosowania MCMC, wymagamy aby (4.1.3) zachodziło dla *dowolnego rozkładu początkowego* ν (a nie dla $\nu=\pi$, czyli dla łańcucha stacjonarnego). Jedną z wersji PWL dla łańcuchów Markowa przedstawimy, wraz z pięknym i prostym dowodem, w Rozdziale 6 rozdziale (Twierdzenie 6.2.5).

Centralne Twierdzenie Graniczne (CTG) dla łańcuchów Markowa ma tezę następującej postaci. Dla dowolnego rozkładu początkowego ν zachodzi zbieżność według rozkładu:

$$\frac{1}{\sqrt{n}}\left(\sum_{i=0}^{n-1}[f(X_i)-\mathbb{E}_\pi f]\right) \longrightarrow \mathrm{N}\big(0,\sigma^2_{\mathrm{as}}\big) \quad (n\to\infty). \tag{4.1.4}$$

Liczba $\sigma^2_{\mathrm{as}} = \sigma^2_{\mathrm{as}}(P,f)$, zwana asymptotyczną wariancją, nie zależy od rozkładu początkowego ν, zależy zaś od macierzy przejścia P i funkcji f. Ponadto można udowodnić następujący fakt: dla dowolnego rozkładu początkowego ν,

$$\frac{1}{n}\mathrm{Var}_\nu\left(\sum_{i=0}^{n-1} f(X_i)\right) \longrightarrow \sigma^2_{\mathrm{as}}. \tag{4.1.5}$$

Przy pewnych dodatkowych założeniach, asymptotyczną wariancję można wyrazić w terminach „stacjonarnych kowariancji" jak następuje. Mamy

$$\sigma^2_{\mathrm{as}} = \mathrm{Var}_\pi f(X_0) + 2\sum_{n=1}^{\infty}\mathrm{Cov}_\pi\big(f(X_0),f(X_n)\big), \tag{4.1.6}$$

gdzie Var_π i Cov_π oznaczają, oczywiście, wariancję i kowariancję obliczoną przy założeniu, że łańcuch jest stacjonarny. Niech

$$\begin{aligned}\sigma^2 =& \mathrm{Var}_\pi f(X_0);\\ \sigma^2\rho_n =& \mathrm{Cov}_\pi\big[f(X_0),f(X_n)\big], \quad \rho_n = \mathrm{corr}_\pi\big[f(X_0),f(X_n)\big].\end{aligned}$$

Przyjmijmy jeszcze, że $\rho_n = \rho_{-n}(f)$. To określenie jest naturalne bo łańcuch stacjonarny można „przedłużyć wstecz". Wzór (4.1.6) można przepisać tak:

$$\sigma^2_{\text{as}} = \sigma^2 \left(1 + 2\sum_{n=1}^{\infty} \rho_n\right) = \sigma^2 \sum_{n=-\infty}^{\infty} \rho_n.$$

Później pojawi się kilka innych wzorów na asymptotyczną wariancję.

Szkic dowodu wzoru (4.1.6). Załóżmy, że rozkładem początkowym jest π i skorzystamy ze stacjonarności łańcucha:

$$\begin{aligned}
\frac{1}{n}\text{Var}_\pi\left(\sum_{i=0}^{n-1} f(X_i)\right) &= \frac{1}{n}\sum_{i=0}^{n-1}\text{Var}_\pi(X_i) + \frac{2}{n}\sum_{i=0}^{n-1}\sum_{j=i}^{n-1}\text{Cov}_\pi(f(X_i), f(X_j)) \\
&= \text{Var}_\pi f(X_0) + 2\sum_{k=1}^{n-1}\frac{n-k}{n}\text{Cov}_\pi(f(X_0), f(X_k)) \\
&\to \text{Var}_\pi f(X_0) + 2\sum_{k=1}^{\infty}\text{Cov}_\pi(f(X_0), f(X_k)), \quad (n \to \infty).
\end{aligned}$$

Przejście do granicy w ostatniej linijce jest uzasadnione elementarnym faktem, że dla dowolnego ciągu liczbowego a_n mamy $\lim_{n\to\infty}\sum_{k=1}^{n-1}\frac{n-k}{n}a_k = \sum_{k=1}^{\infty} a_k$, o ile szereg po prawej stronie równości jest zbieżny. Wyprowadziliśmy wzór (4.1.6) dla $\nu = \pi$.

Pominiemy uzasadnienie tego, że granica ciągu $\text{Var}_\nu \sum_{i=0}^{n-1} f(X_i)/n$ nie zależy of ν. □

W tym miejscu chcę podkreślić różnicę między stacjonarną wariancją $\sigma^2 = \text{Var}_\pi f = \text{Var}_\pi f(X_n)$ i asymptotyczną wariancją σ^2_{as}. W większości zastosowań kowariancje we wzorze (4.1.6) są dodatnie (zmienne losowe $f(X_0)$ i $f(X_k)$ są dodatnio skorelowane). W rezultacie σ^2_{as} jest dużo większa od σ^2. To jest cena, którą płacimy za używanie łańcucha Markowa zamiast ciągu zmiennych niezależnych, jak w Rozdziale 3.

Zauważmy, że algorytmy Monte Carlo przeważnie mają za zadanie obliczyć pewną wartość oczekiwaną, a więc wielkość postaci $\theta = \mathbb{E}_\pi f$. Jeśli potrafimy generować łańcuch Markowa zbieżny do π, to naturalnym estymatorem $\mathbb{E}_\pi f$ jest $\hat{\theta}_n = \frac{1}{n}\sum_{i=0}^{n-1} f(X_i)$. W praktyce niemal zawsze odrzuca się początkowy odcinek trajektorii długości b. Liczba kroków b (*burn-in time*) jest „czasem po którym łańcuch zbliża się dostatecznie do rozkładu stacjonarnego" (przeważnie wybór b jest raczej arbitralny i heurystyczny). Do obliczania estymatora używamy tylko dalszej części trajektorii:

$$\hat{\theta}_{b,n} = \frac{1}{n}\sum_{i=b}^{b+n-1} f(X_i).$$

Zauważmy, że graniczne zachowanie estymatora $\hat{\theta}_{b,n}$ jest takie samo, jak estymatora $\hat{\theta}_n$, ponieważ zmienia się tylko rozkład początkowy: łańcuch startuje z X_b zamiast z X_0. Możemy tezy PWL oraz CTG zapisać w skrócie tak:

$$\hat{\theta}_{b,n} \longrightarrow_{\text{p.n.}} \theta \qquad (n \to \infty),$$
$$\sqrt{n}\left(\hat{\theta}_{b,n} - \theta\right) \longrightarrow_{\text{d}} \mathrm{N}\big(0, \sigma^2_{\text{as}}\big), \quad (n \to \infty).$$

PWL gwarantuje *zgodność* estymatora, a więc w pewnym sensie *poprawność* metody. Jest to, rzecz jasna, zaledwie wstęp do dokładniejszej analizy algorytmu. Graniczne zachowanie wariancji estymatora $\hat{\theta}_n$ wyjaśnia wzór (4.1.5). Uzupełnijmy to (pomijając chwilowo uzasadnienie) opisem granicznego zachowania *obciążenia*: przy $n \to \infty$,

$$\mathrm{Var}_\nu(\hat{\theta}_{b,n}) = \frac{1}{n}\sigma^2_{\text{as}} + o\left(\frac{1}{n}\right),$$
$$\mathbb{E}_\nu \hat{\theta}_{b,n} - \theta = O\left(\frac{1}{n}\right).$$

Oczywiście, naturalną miarą jakości estymatora jest *błąd średniokwadratowy* (BŚK). Ponieważ BŚK jest sumą wariancji i *kwadratu* obciążenia to, przynajmniej w granicy dla $n \to \infty$, wariancja ma dominujący wpływ, zaś obciążenie staje się zaniedbywalne:

$$\mathbb{E}_\nu\left(\hat{\theta}_{b,n} - \theta\right)^2 = \frac{1}{n}\sigma^2_{\text{as}} + o\left(\frac{1}{n}\right). \tag{4.1.7}$$

Zauważmy, że CTG może służyć do budowania asymptotycznych przedziałów ufności dla estymowanej wielkości θ: jeśli przyjmiemy poziom ufności $1-\alpha$ i dobierzemy odpowiedni kwantyl z rozkładu normalnego, to

$$\lim_{n\to\infty} \mathbb{P}\left(|\hat{\theta}_{b,n} - \theta| \leqslant \frac{z\sigma_{\text{as}}}{\sqrt{n}}\right) = \Phi(z) - \Phi(-z) = 1 - \alpha. \tag{4.1.8}$$

Oczywiście, $\Phi(z)$ oznacza to dystrybuantę rozkładu $\mathrm{N}(0,1)$ i $\Phi(z) = 1 - \alpha/2$.

Potrzebne jest jeszcze oszacowanie asymptotycznej wariancji σ^2_{as}, co nie jest wcale łatwe. Przedstawię jeden ze sposobów estymacji σ^2_{as}, metodę *batch means* (średnich blokowych). Podzielmy trajektorię łańcucha Markowa długości n na k „bloków" długości m każdy (zatem $n = km$):

$$\underbrace{X_b, X_{b+1}, \ldots, X_{b+m-1}}_{\text{blok 1}}, \underbrace{X_{b+m}, X_{b+m+1}, \ldots, X_{b+2m-1}}_{\text{blok 2}}, \ldots, \underbrace{X_{b+km}, X_{b+km+1}, \ldots, X_{b+(k+1)m-1}}_{\text{blok } k}.$$

Oznaczmy przez $\bar{\theta}_j$ średnią j obliczoną z j-tego bloku:

$$\bar{\theta}_j = \frac{1}{m}\sum_{i=b+jm}^{b+(j+1)m-1} f(X_i).$$

Estymatorem wariancji asymptotycznej jest

$$\hat{\sigma}^2_{\text{as}} = \frac{m}{k}\sum_{j=1}^{k}(\bar{\theta}_j - \hat{\theta}_{b,n})^2,$$

gdzie $\hat{\theta}_{b,n}$ jest estymatorem obliczonym na podstawie trajektorii długości $n = km$. Estymator $\hat{\sigma}_{\text{as}}$ jest, przy pewnych założeniach zgodny w następującym sensie: $\hat{\sigma}_{\text{as}} \to \sigma_{\text{as}}$ jeżeli jednocześnie $m \to \infty$ i $k \to \infty$ (coraz więcej coraz dłuższych bloków!). Ten fakt nie powinien dziwić w świetle tego, co powiedzieliśmy wcześniej.

Sławne i ważne twierdzenia graniczne sformułowane w tym podrozdziale nie są, niestety, całkowicie zadowalającym narzędziem analizy algorytmów Monte Carlo. Algorytmy wykorzystujące łańcuchy Markowa są użyteczne wtedy, gdy osiągają wystarczającą dokładność dla liczby kroków n *znikomo małej* w porównaniu z rozmiarem przestrzeni stanów. W przeciwnym przypadku można po prostu deterministycznie „przejrzeć wszystkie stany" i dokładnie obliczyć interesującą nas wielkość. Niemniej, twierdzenia graniczne są interesujące z jakościowego punktu widzenia.

4.2 Zadania i uzupełnienia

4.1 Zadanie. Łańcuch Markowa na przestrzeni stanów $\mathcal{X} = \{1, 2\}$ ma macierz przejścia

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}.$$

Niech $f(x) = x$ dla $x \in \mathcal{X}$ (rozważamy funkcję „tożsamość").

- Oblicz asymptotyczną wariancję σ^2_{as} w zależności od α i β. *Wskazówka:* Wykorzystaj rozwiązanie Zadania 2.5, czyli postać macierzy P^n do obliczenia $\text{Cov}_\pi(X_0, X_n)$, a następnie użyj wzoru (4.1.6).
- Oblicz σ^2_{as} symulacyjnie, generując k niezależnych trajektorii długości m. W tym prostym przykładzie (*toy example*) umiemy generować łańcuchy stacjonarne. W zastosowaniach MCMC to jest, oczywiście, wykluczone! Porównaj wyniki dla, powiedzmy, $\alpha = \beta = 0.1$ i dla $\alpha = \beta = 0.9$ (zauważ, że rozkłady stacjonarne w obu przypadkach są identyczne). Porównaj ze wzorem teoretycznym.

4.2 Zadanie. Rozważmy proces AR(1), czyli łańcuch Markowa na przestrzeni $\mathcal{X} = \mathbb{R}$ zdefiniowany równaniem rekurencyjnym

$$X_{n+1} = \alpha X_n + W_{n+1},$$

gdzie $W_1, W_2, \ldots$ są niezależnymi zmiennymi losowymi o rozkładzie $N(0, v^2)$. Podobnie jak w poprzednim zadaniu, rozważamy funkcję $f(x) = x$.

- Oblicz σ^2_{as} w zależności od α. *Wskazówka:* Przypomnij sobie Zadanie 2.6. Najpierw oblicz $\text{Cov}_\pi(X_0, X_n)$, a następnie użyj wzoru (4.1.6).
- Oblicz σ^2_{as} symulacyjnie, tak jak w zadaniu poprzednim. Porównaj wyniki dla, powiedzmy, $\alpha = 0.9$ i dla $\alpha = -0.9$ (zauważ, że rozkłady stacjonarne w obu przypadkach są identyczne).

4.3 Podstawowe algorytmy MCMC

Zadanie rozpatrywane w tym rozdziale jest następujące. Dla danego rozkładu π na przestrzeni $\mathcal{X}$ chcemy znaleźć sposób generowania łańcucha Markowa, który jest zbieżny do tego rozkładu. Szukamy takiego jądra (macierzy) przejścia P, że P ma rozkład stacjonarny π (Definicja 4.1.1).

Odwracalność

Najważniejsze algorytmy MCMC są oparte na idei odwracalności łańcucha Markowa.

4.3.1 Definicja. *Łańcuch o jądrze P jest odwracalny względem rozkładu prawdopodobieństwa π, jeśli dla dowolnych $A, B \subseteq \mathcal{X}$ mamy*

$$\int_A \pi(\mathrm{d}x)P(x, B) = \int_B \pi(\mathrm{d}x')P(x', A).$$

W skrócie,

$$\pi(\mathrm{d}x)P(x, \mathrm{d}x') = \pi(\mathrm{d}x')P(x', \mathrm{d}x).$$

Odwracalność implikuje, że rozkład π jest stacjonarny. Jest to dlatego ważne, że sprawdzanie odwracalności jest stosunkowo łatwe.

4.3.2 Twierdzenie. *Jeśli $\pi(\mathrm{d}x)P(x, \mathrm{d}x') = \pi(\mathrm{d}x')P(x', \mathrm{d}x)$ to $\pi^\top = \pi^\top P$.*

Dowód. $\int_{\mathcal{X}} \pi(\mathrm{d}x)P(x, B) = \int_B \pi(\mathrm{d}x')P(x', \mathcal{X}) = \int_B \pi(\mathrm{d}x') = \pi(B)$. □

Algorytm Metropolisa-Hastingsa

To jest pierwszy historycznie i wciąż najważniejszy algorytm MCMC. Zakładamy, że umiemy generować łańcuch Markowa z pewnym jądrem q. Pomysł Metropolisa polega na tym, żeby zmodyfikować ten łańcuch wprowadzając specjalnie dobraną regułę akceptacji w taki sposób,

żeby wymusić zbieżność do zadanego rozkładu π. W dalszym ciągu systematycznie utożsamiamy rozkłady prawdopodobieństwa z ich gęstościami, aby nie mnożyć oznaczeń. Mamy zatem:

- Rozkład docelowy: $\pi(\mathrm{d}x) = \pi(x)\mathrm{d}x$.
- Rozkład „propozycji": $q(x, \mathrm{d}x') = q(x, x')\mathrm{d}x'$.
- Prawdopodobieństwo akceptacji:

$$a(x, x') = \frac{\pi(x')q(x', x)}{\pi(x)q(x, x')} \wedge 1. \tag{4.3.3}$$

Algorytm Metropolisa-Hastingsa (MH) interpretujemy jako „błądzenie losowe" zgodnie z jądrem przejścia q, zmodyfikowane poprzez odrzucanie niektórych ruchów, przy czym reguła akceptacji/odrzucania zależy w specjalny sposób od π. Pojedynczy krok algorytmu jest następujący.

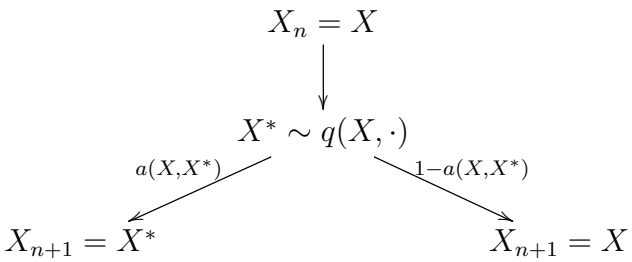
```
function KrokMH(X)
  Gen X* ~ q(X, ·); { propozycja }
  Gen U ~ U(0, 1)
  if U > a(X, X*) then X' := X* { ruch zaakceptowany z pr-stwem a(X, X') }
                  else X' := X { ruch odrzucony z pr-stwem 1 − a(X, X') }
KrokMH := X'
```

Oczywista jest analogia z podstawową metodą eliminacji. Zasadnicza różnica polega na tym, że w algorytmie MH nie „odrzucamy" zmiennej losowej, tylko „odrzucamy propozycje ruchu" i stoimy w miejscu. Algorytm MH wymaga znajomości gęstości π tylko z dokładnością do proporcjonalności, *bez stałej normującej*.

Łańcuch Markowa $X_0, X_1, \ldots, X_n, \ldots$ powstaje zgodnie z następującym schematem:

```
Gen X_0 ~ ν; { start }
for n := 1 to ∞
    begin
    X_n := KrokMH(X_{n−1}) { krok }
    end
```

Graficznie to można przedstawić w takiej postaci.



Jądro przejścia M-H jest następujące:

$$P(x, B) = \int_B \mathrm{d}x'\, q(x, x')a(x, x') + \mathbb{1}(x \in B) \int_{\mathcal{X}} \mathrm{d}x'\, q(x, x')[1 - a(x, x')].$$

Dla przestrzeni skończonej, jądro łańcucha MH redukuje się do macierzy prawdopodobieństw przejścia. Wzór jest w tym przypadku bardzo prosty: dla $x \neq x'$,

$$P(x, x') = q(x, x')a(x, x').$$

4.3.4 Twierdzenie. *Jądro przejścia MH jest odwracalne względem π.*

Dowód. Ograniczmy się do przestrzeni skończonej, żeby nie komplikować oznaczeń. W ogólnym przypadku dowód jest w zasadzie taki sam, tylko napisy stają się mniej czytelne. Niech (bez straty ogólności)

$$a(x, x') = \frac{\pi(x')q(x', x)}{\pi(x)q(x, x')} \leqslant 1, \qquad a(x', x) = 1.$$

Wtedy

$$\begin{aligned} \pi(x)P(x, x') &= \pi(x)q(x, x')a(x, x') \\ &= \pi(x)q(x, x')\frac{\pi(x')q(x', x)}{\pi(x)q(x, x')} \\ &= \pi(x')q(x', x) \\ &= \pi(x')P(x', x) \quad \text{bo} \quad a(x', x) = 1. \end{aligned}$$

□

Uwagi historyczne:

- Metropolis w roku 1953 zaproponował algorytm, w którym zakłada się symetrię rozkładu propozycji, $q(x, x') = q(x', x)$. Warto zauważyć, że wtedy łańcuch odpowiadający q (błądzenie bez eliminacji ruchów) ma rozkład stacjonarny *jednostajny*. Reguła akceptacji przybiera postać

$$a(x, x') = \frac{\pi(x')}{\pi(x)} \wedge 1.$$

- Hastings w roku 1970 uogólnił rozważania na przypadek niesymetrycznego q.

Alternatywna reguła obliczania prawdopodobieństwa akceptacji, znana jako reguła Barkera, jest następująca:

$$a(x, x') = \frac{\pi(x')q(x', x)}{\pi(x')q(x', x) + \pi(x)q(x, x')}. \tag{4.3.5}$$

Jeśli w funkcji $krokMH$ użyjemy tak określonej funkcji akceptacji, to również otrzymamy łańcuch π-odwracalny (Zadanie 4.3). Intuicje stojące za algorytmem Metropolisa najlepiej zilustrować, rozpatrując rodzinę rozkładów (na skończonej przestrzeni $\mathcal{X}$) postaci

$$\pi_\beta = \frac{1}{z_\beta} \exp[-\beta H(x)], \tag{4.3.6}$$

gdzie $z_\beta = \sum_x \exp[-\beta H(x)]$ jest stałą normującą. Są to tak zwane rozkłady Gibbsa. (Każdy rozkład na przestrzeni skończonej może być napisany w postaci rozkładu Gibbsa, jeśli przyjmiemy konwencję $\exp[-\infty] = 0$. Chodzi o interpretację fizyczną: $H(x)$ traktujemy jako energię „stanu" x, zaś β jest „odwrotnością temperatury". Załóżmy, że macierz propozycji q jest symetryczna. Łatwo sprawdzić, że reguły akceptacji Metropolisa i Barkera przybierają następującą postać.

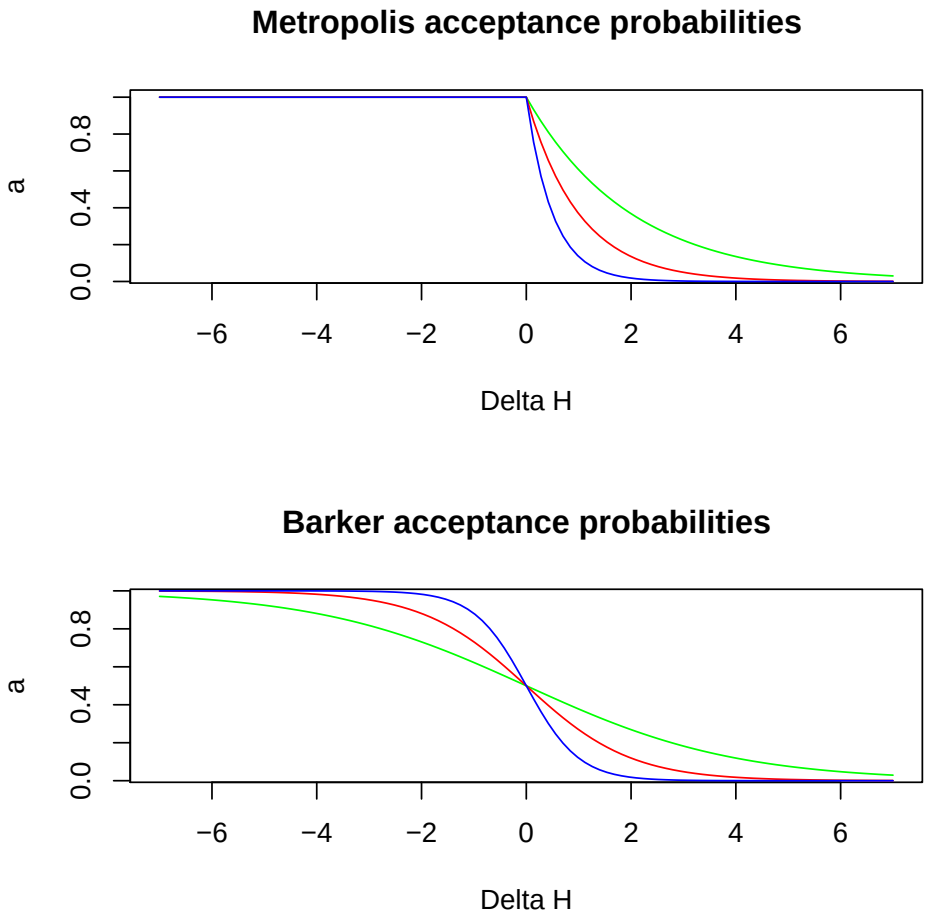
$$\begin{aligned} a_{\mathrm{Met}}(x, x') &= \exp\left[-\beta \max(H(x') - H(x), 0)\right], \\ a_{\mathrm{Bar}}(x, x') &= \frac{\exp\left[-\beta(H(x') - H(x))\right]}{1 + \exp\left[-\beta(H(x') - H(x))\right]}. \end{aligned}$$

Obie funkcje akceptacji są przedstawione na Rysunku 4.1. Zauważmy, że $a(x, x')$ dąży do funkcji zero-jedynkowej $\mathbb{1}(H(x') \leqslant H(x))$ przy $\beta \to \infty$. Jeśli tempratura spada do zera, to akceptujemy tylko ruchy zmniejszające energię i odrzucamy propozycje ruchów zwiększających energię. Algorytm Metropolisa ma bliski związek z zadaniem minimalizacji funkcji H (Zadanie 4.4 i komentarz do tego zadania).

Próbnik Gibbsa

Drugim podstawowym algorytmem MCMC jest próbnik Gibbsa (PG) (*Gibbs Sampler*, GS). Załóżmy, że przestrzeń na której żyje docelowy rozkład π ma strukturę produktową: $\mathcal{X} = \prod_{i=1}^d \mathcal{X}_i$. Łańcuch Markowa na przestrzeni $\mathcal{X}$ będziemy teraz oznaczali $X(0), X(1), \ldots, X(n), \ldots$, gdzie $X(n)$ jest wektorem $(X_1(n), \ldots, X_d(n))$. Ponadto przyjmijmy następujące oznaczenia:

- Jeśli $\mathcal{X} \ni x = (x_i)_{i=1}^d$ to $x_{-i} = (x_j)_{j \neq i}$: wektor z pominiętą i-tą współrzędną.
- Rozkład docelowy (gęstość): $\pi(\mathrm{d}x) = \pi(x)\mathrm{d}x$.



Rysunek 4.1: Prawdopodobieństwa akceptacji Metropolisa i Barkera dla różnych wartości parametru β (odwrotnej temperatury) .

- Pełne rozkłady warunkowe (*full conditionals*):

$$\pi(x_i|x_{-i}) = \frac{\pi(x)}{\pi(x_{-i})}$$

Mały krok PG jest zmianą i-tej współrzędnej (wylosowaniem nowej wartości z rozkładu warunkowego):

$$\begin{aligned} X &= (X_1, \ldots, X_i, \ldots, X_d) \\ &\downarrow \\ \text{Gen } X'_i &\sim \pi(\cdot|X_{-i}) \\ &\downarrow \\ X' &= (X_1, \ldots, X'_i, \ldots, X_d). \end{aligned}$$

Prawdopodobieństwo przejścia małego kroku PG (w przypadku przestrzeni skończonej) jest takie:

$$P_i(x, x') = \pi(x'_i|x_{-i})\mathbb{1}(x_{-i} = x'_{-i}).$$

4.3.7 Twierdzenie. *Mały krok PG jest π-odwracalny.*

Dowód. Niech $x_{-i} = x'_{-i}$. Wtedy

$$\begin{aligned}\pi(x)P_i(x, x') &= \pi(x)\pi(x'_i|x_{-i}) \\ &= \pi(x_{-i})\pi(x_i|x_{-i})\pi(x'_i|x_{-i}) \\ &= \pi(x'_{-i})\pi(x_i|x_{-i})\pi(x'_i|x'_{-i}) \\ &= \pi(x')P_i(x', x).\end{aligned}$$

(skorzystaliśmy z symetrii). □

Trzeba jeszcze zadbać o to, żeby łańcuch generowany przez PG był nieprzywiedlny. Musimy zmieniać wszystkie współrzędne, nie tylko jedną. Istnieją dwie zasadnicze odmiany próbnika Gibbsa, różniące się sposbem wyboru współrzędnych do zmiany.

- Losowy wybór współrzędnych, „LosPG".
- Systematyczny wybór współrzędnych, „SystemPG".

Losowy PG. Wybieramy współrzędną i-tą z prawdopodobieństwem $c(i)$.

```
function LosPG(X)
  Gen I ~ c(·);
  Gen X'_I := π(·|X_{-I}); { zmieniamy I-tą współrzędną }
  X'_{-I} := X_{-I}; { wszystkie inne współrzędne pozostawiamy bez zmian }
LosPG := X'
```

Systematyczny PG. Współrzędne są zmieniane w porządku cyklicznym.

```
function SystemPG(X)
  begin
  Gen X'_1 ~ π(·|X_2, ..., X_d);
  Gen X'_2 ~ π(·|X'_1, X_3, ..., X_d);
  ...
  Gen X'_d ~ π(·|X'_1, ..., X'_{d-1});
  SystemPG := X'
  end
```

Oczywiście, łańcuch $X(0), X(1), \ldots, X(n), \ldots$ generujemy powtarzając instrukcję $X_n := LosPG(X_{n-1})$ lub $X_n := SystemPG(X_{n-1})$.

Jądro przejścia w „dużym" kroku losowego PG jest takie:

$$P = \sum_{i=1}^{d} c(i) P_i.$$

Losowy PG jest **odwracalny**.

Jądro przejścia w „dużym" kroku systematycznego PGjest następujące:

$$P = P_1 P_2 \cdots P_d.$$

Systematyczny PG **nie jest odwracalny**. Ale jest π-stacjonarny, bo $\pi^\top P_1 P_2 \cdots P_d = \pi^\top$.

Przy projektowaniu konkretnych realizacji PG pojawia się szereg problemów, ważnych zarówno z praktycznego jak i teoretycznego punktu widzenia. Jak wybrać rozkład $c(\cdot)$ w losowym PG? Jest raczej jasne, że niektóre współrzędne powinny być zmieniane częściej, a inne rzadziej. Jak dobrać kolejność współrzędnych w systematycznym PG? Ta kolejność ma wpływ na tempo zbieżności łańcucha. Wreszcie, w wielu przykładach można zmieniać całe „bloki" współrzędnych na raz.

Systematyczny PG jest uważany za bardziej efektywny i częściej stosowany w praktyce. Z drugiej strony jest trudniejszy do analizy teoretycznej, niż losowy PG.

4.3.8 Przykład. Dwa poniższe rysunki pokazują pracę (systematycznego) próbnika Gibbsa w prostym przykładzie 2-wymiarowym. Docelowy rozkład ma postać

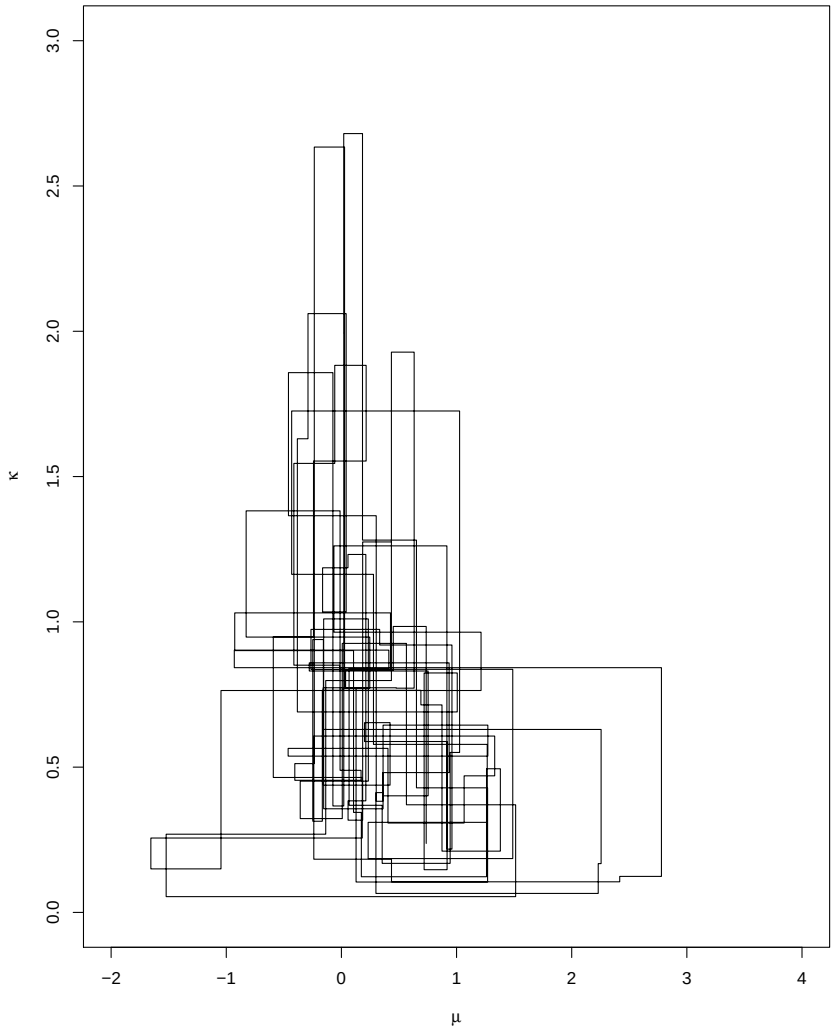
$$\pi(\mu, \kappa) \propto \exp\left[-\kappa\left(\frac{s^2}{2} + \frac{n}{2}(\mu - \bar{y})^2\right) - \frac{v^2}{2}(\mu - m)^2\right] \kappa^{n/2-1},$$

gdzie $n = 5$, $s^2 = 5$, $\bar{y} = 0$, $m = 5$, $v^2 = 0.2$. jest jasne, że $\kappa|\mu \sim \text{Gamma}(\cdots)$ i $\mu|\kappa \sim \text{N}(\cdots)$, więc PG jest łatwy do implementacji. Motywacją tego przykładu jest pewien bayesowski model statystyczny, który przedstawię (w znacznie większej ogólności) w następnym rozdziale. △

4.4 Zadania i uzupełnienia

4.1 Ćwiczenie. Przeprowadzić symulacje w Przykładzie 4.3.8.

4.3 Zadanie. Sprawdzić, że zastosowanie reguły akceptacji Barkera (4.3.5) zamiast (4.3.3) w algorytmie $krokMH$ prowadzi do łańcucha π-odwracalnego.

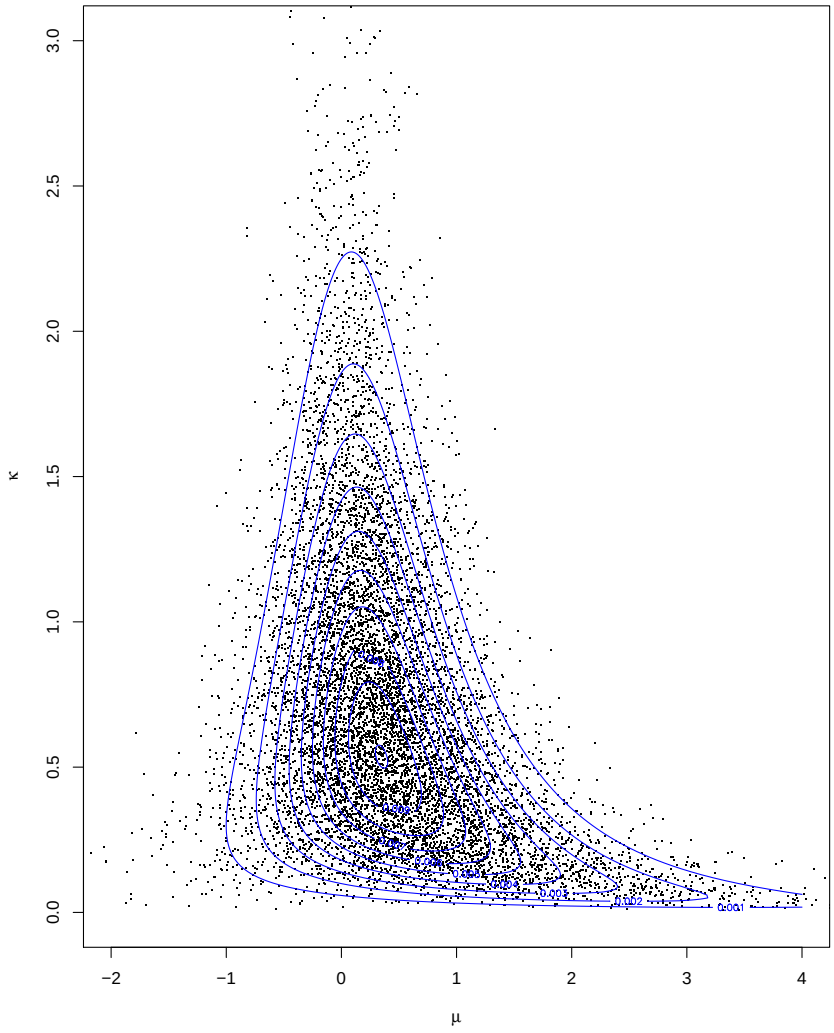


Rysunek 4.2: Trajektoria próbnika Gibbsa w przestrzeni dwuwymiarowej.

4.4 Zadanie. Jeśli $\mathcal{X}$ jest przestrzenią skończoną i rozpatrujemy rodzinę rozkładów Gibbsa (4.3.6), to $\pi_\beta \to \mathrm{U}(\mathcal{X}_{\min})$ przy $\beta \to \infty$, gdzie $\mathcal{X}_{\min} = \{x \in \mathcal{X} : H(x) = \min_{x' \in \mathcal{X}} H(x')\}$. Innymi słowy,

$$\pi_\beta(x) \to \begin{cases} 0 & \text{jeśli } x \notin \mathcal{X}_{\min}; \\ 1/|\mathcal{X}_{\min}| & \text{jeśli } x \in \mathcal{X}_{\min}, \end{cases} \qquad (\beta \to \infty).$$

Wiemy, jak skonstruować algorytm Metropolisa zbieżny do π_β. Prawdopodobieństwo akceptacji zależy od β. Jeśli w przebiegu algorytmu będziemy zwiększać β do nieskończoności, to można się spodziewać zbieżności do rozkładu skupionego na minimach energii, $\mathrm{U}(\mathcal{X}_{\min})$. W ten sposób powstaje algorytm minimalizacji zwany Symulowanym Wyżarzaniem (*Simulated Annealing, SA).*



Rysunek 4.3: Chmurka punktów wygenerowanych przez próbnik Gibbsa i poziomice gęstości docelowej.

Analiza algorytmów SA jest bez porównania trudniejsza, niż „zwykłego" algorytmu Metropolisa, ponieważ generowany przez SA łańcuch Markowa jest niejednorodny (prawdopodobieństwa przejścia zmieniają się z kroku na krok). Warunki zapewniające zbieżność algorytmu SA do $\mathrm{U}(\mathcal{X}_{\min})$ są skomplikowane.

Rozdział 5

Przykłady zastosowań MCMC

5.1 Statystyka bayesowska

Algorytmy MCMC zrewolucjonizowały statystykę bayesowską. Stworzyły możliwość obliczania (w przybliżeniu) rozkładów *a posteriori* w sytuacji, gdy dokładne, analityczne wyrażenia są niedostępne. W ten sposób statystycy uwolnili się od konieczności używania nadmiernie uproszczonych modeli. Zaczęli śmiało budować modele coraz bardziej realistyczne, zwykle o strukturze hierarchicznej. Przedstawię to na dwóch dość typowych przykładach. Pierwszy z nich jest oparty na pracy [15], drugi na [12]. Inne przykłady i doskonały wstęp do tematyki zastosowań MCMC można znaleźć w pracy Geyera [4].

Hierarchiczny model klasyfikacji

5.1.1 Przykład (Statystyka małych obszarów)**.** Zacznijmy od opisania problemu tak zwanych „małych obszarów", który jest dość ważny w dziedzinie badań reprezentacyjnych, czyli w tak zwanej „statystyce oficjalnej". *Małe obszary* to pod-populacje w których rozmiar próbki nie jest wystarczający, aby zastosować „zwykłe" estymatory (średnie z próbki). Podejście bayesowskie pozwala „pożyczać informację" z innych obszarów. Zakłada się, że z każdym małym obszarem związany jest nieznany parametr, który staramy się estymować. Obserwacje pochodzące z określonego obszaru mają rozkład prawdopodobieństwa zależny od odpowiadającego temu obszarowi parameru. Parametry, zgodnie z filozofią bayesowską, traktuje się jak zmienne losowe. W najprostszej wersji taki model jest zbudowany w sposób opisany poniżej. △

Model bayesowski

- $y_{ij} \sim \mathrm{N}(\theta_i, \sigma^2)$ – badana cecha dla j-tej wylosowanej jednostki i-tego obszaru, $(j = 1, \ldots, n_i)$, $(i = 1, \ldots, k)$,
- $\theta_i \sim \mathrm{N}(\mu, \upsilon^2)$ – interesująca nas średnia w i-tym obszarze,
- μ – średnia w całej populacji.

Ciekawe, że ten sam model pojawia się w różnych innych zastosowanianich, na przykład w matematyce ubezpieczeniowej. Przytoczymy klasyczny rezultat dotyczący tego modelu, aby wyjaśnić na czym polega wspomniane „pożyczanie informacji".

Estymator bayesowski

W modelu przedstawionym powyżej, łatwo obliczyć estymator bayesowski (przy kwadratowej funkcji straty), czyli wartość oczekiwaną *a posteriori*. Następujący wzór jest bardzo dobrze znany specjalistom od małych obszarów i aktuariuszom.

$$\hat{\theta}_i = \mathbb{E}(\theta_i|y) = z_i\bar{y}_i + (1 - z_i)\mu, \qquad z_i = \frac{n_i\upsilon^2}{n_i\upsilon^2 + \sigma^2}.$$

Estymator bayesowski dla i-tego obszaru jest średnią ważoną $\bar{y}$ (estymatora opartego na danych z tego obszaru) i wielkości μ, która opisuje całą populację, a nie tylko i-ty obszar. Niestety, proste estymator napisany powyżej zależy od parametrów μ, σ i v, które w praktyce są nieznane i które trzeba estymować. Konsekwentnie bayesowskie podejście polega na traktowaniu również tych parametrów jako zmiennych losowych, czyli nałożeniu na nie rozkładów a priori. Powstaje w ten sposób model hierarchiczny.

Hierarchiczny model bayesowski

Uzupełnijmy rozpatrywany powyżej model, dobudowując „wyższe piętra" hierarchii. potraktujemy mianowicie parametry rozkładów *a priori*: μ, σ i v jako zmienne losowe i wyspecyfikujemy ich rozkłady *a priori*.

- $y_{ij} \sim \mathrm{N}(\theta_i, \sigma^2)$,
- $\theta_i \sim \mathrm{N}(\mu, \upsilon^2)$,
- $\mu \sim \mathrm{N}(m, \tau^2)$,
- $\sigma^{-2} \sim \mathrm{Gamma}(p, \lambda)$,

- $\upsilon^{-2} \sim \text{Gamma}(q, \kappa)$.

Zakładamy przy tym, że μ, σ i υ są *a priori* niezależne (niestety, są one zależne *a posteriori*). Na szczycie hierarchii mamy „hiperparametry" m, τ, p, λ, q, κ, o których musimy założyć, ze są znanymi liczbami.

Łączny rozkład prawdopodobieństwa wszystkich zmiennych losowych w modelu ma postać

$$p(y, \theta, \mu, \sigma^{-2}, \upsilon^{-2}) = p(y|\theta, \sigma^{-2})p(\theta|\mu, \upsilon^{-2})p(\mu)p(\sigma^{-2})p(\upsilon^{-2}).$$

We wzorze powyżej i w dalej traktujemy (trochę nieformalnie) σ^{-2} i υ^{-2} jako pojedyncze symbole nowych zmiennych, żeby nie mnożyć oznaczeń. Rozkład prawdopodobieństwa *a posteriori* jest więc taki:

$$p(\theta, \mu, \sigma^{-2}, \upsilon^{-2}|y) = \frac{p(y, \theta, \mu, \sigma^{-2}, \upsilon^{-2})}{p(y)}.$$

To jest rozkład „docelowy" π, na przestrzeni $\mathcal{X} = \mathbb{R}^{k+3}$, ze nieznaną stałą normującą $1/p(y)$. Choć wygląda na papierze dość prosto, ale obliczenie rozkładów brzegowych, wartości oczekiwanych i innych charakterystyk jest, łagodnie mówiąc, trudne.

Opiszemy teraz jak jest skonstruowany **próbnik Gibbsa w modelu hierarchicznym**. Rozkłady warunkowe poszczególnych współrzędnych są proste i łatwe do generowania. Można te rozkłady „odczytać" uważnie patrząc na rozkład łączny:

$$\begin{aligned} p(\theta, \mu, \upsilon^{-2}, \sigma^{-2}|y) \propto (\sigma^{-2})^{n/2} \exp\left\{-\frac{\sigma^{-2}}{2}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\theta_i)^2\right\} \\ \cdot (\upsilon^{-2})^{k/2} \exp\left\{-\frac{\upsilon^{-2}}{2}\sum_{i=1}^{k}(\theta_i-\mu)^2\right\} \\ \cdot \exp\left\{-\frac{\tau^{-2}}{2}(\mu-m)^2\right\} \\ \cdot (\sigma^{-2})^{p-1} \exp\{-\lambda\sigma^{-2}\} \\ \cdot (\upsilon^{-2})^{q-1} \exp\{-\kappa\upsilon^{-2}\}. \end{aligned}$$

Dla ustalenia uwagi zajmijmy się rozkładem warunkowym zmiennej υ^{-2}. Kolorem niebieskim oznaczyliśmy te czynniki łącznej gęstości, które zawierają υ^{-2}. Pozostałe, czarne czynniki traktujemy jako stałe. Stąd widać, jak wygląda rozkład warunkowy υ^{-2}„ przynajmniej z dokładnością do proporcjonalności:

$$\begin{aligned} p(\upsilon^{-2}|y, \theta, \mu, \sigma^{-2}) \propto (\upsilon^{-2})^{k/2+q-1} \\ \cdot \exp\left\{-\left(\frac{1}{2}\sum_{i=1}^{k}(\theta_i-\mu)^2+\kappa\right)\upsilon^{-2}\right\}. \end{aligned}$$

Jest to zatem rozkład Gamma$(k/2+q, \sum_{i=1}^{k}(\theta_i-\mu)^2/2+\kappa)$. Zupełnie podobnie rozpoznajemy inne (pełne) rozkłady warunkowe:

$$\begin{aligned}
\upsilon^{-2}|y,\theta,\mu,\sigma^{-2} &\sim \text{Gamma}\left(\frac{k}{2}+q, \frac{1}{2}\sum_{i=1}^{k}(\theta_i-\mu)^2+\kappa\right),\\
\sigma^{-2}|y,\theta,\mu,\upsilon^{-2} &\sim \text{Gamma}\left(\frac{n}{2}+p, \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{n_i}(y_{ij}-\theta_i)^2+\lambda\right),\\
\mu|y,\theta,\sigma^{-2},\upsilon^{-2} &\sim \text{N}\left(\frac{k\tau^2}{k\tau^2+\upsilon^2}\bar{\theta}+\frac{\upsilon^2}{k\tau^2+\upsilon^2}m, \frac{\tau^2\upsilon^2}{k\tau^2+\upsilon^2}\right),\\
\theta_i|y,\theta_{-i},\mu,\sigma^{-2},\upsilon^{-2} &\sim \text{N}\left(\frac{n\upsilon^2}{n\upsilon^2+\sigma^2}\bar{y}_i+\frac{\sigma^2}{n\upsilon^2+\sigma^2}\mu, \frac{\upsilon^2\sigma^2}{n\upsilon^2+\sigma^2}\right),
\end{aligned}$$

gdzie, rzecz jasna, $n=\sum n_i$, $\bar{\theta}=\sum_i \theta_i/k$ i $\theta_{-i}=(\theta_k)_{k\neq i}$. Zwróćmy uwagę, że współrzędne wektora θ są warunkowo niezależne (pełny rozkład warunkowy θ_i nie zależy od θ_{-i}). Dzięki temy możemy w próbniku Gibbsa potraktować θ jako cały „blok" współrządnych i zmieniać „na raz".

Próbnik Gibbsa ma w tym modelu przestrzeń stanów $\mathcal{X}$ składającą się z punktów $x=(\theta,\mu,\sigma^{-2},\upsilon^{-2})\in\mathbb{R}^{k+3}$. Reguła przejścia próbnika w wersji systematycznej (duży krok „SystemPG"),

$$\underbrace{\left(\theta,\mu,\sigma^{-2},\upsilon^{-2}\right)}_{X_t} \longmapsto \underbrace{\left(\theta,\mu,\sigma^{-2},\upsilon^{-2}\right)}_{X_{t+1}},$$

jest złożona z następujących „małych kroków":

- Wylosuj $\upsilon^{-2} \sim p(\upsilon^{-2}|y,\theta,\mu,\sigma^{-2}\quad) = \text{Gamma}(\ldots)$,
- Wylosuj $\sigma^{-2} \sim p(\sigma^{-2}|y,\theta,\mu,\quad \upsilon^{-2}) = \text{Gamma}(\ldots)$,
- Wylosuj $\mu \sim p(\mu\ |y,\theta,\ \sigma^{-2},\upsilon^{-2}) = \text{N}(\ldots)$,
- Wylosuj $\theta \sim p(\theta\ |y,\ \mu,\sigma^{-2},\upsilon^{-2}) = \text{N}(\ldots)$.

Łańcuch Markowa jest zbieżny do rozkładu *a posteriori*:

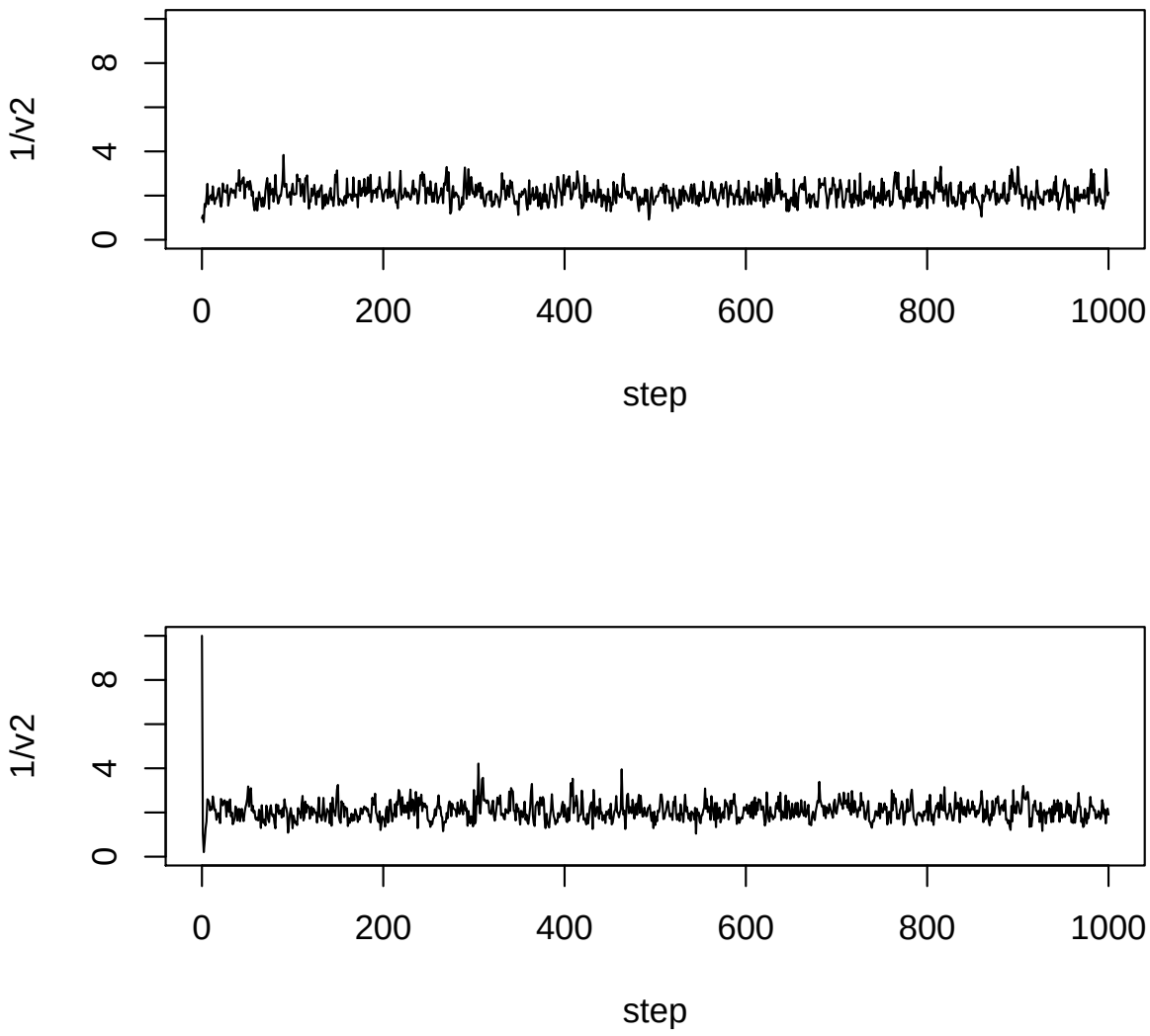
$$X_t \to \pi(\cdot) = p(\theta,\mu,\sigma^{-2},\upsilon^{-2}|y).$$

Najbardziej interesujące są w tym hierarchicznym modelu zmienne θ_i (pozostałe zmienne można uznać za „parametry zkłócające). Dla ustalenia uwagi zajmijmy się zmienną θ_1 (powiedzmy, wartością średnią w pierwszym małym obszarze). *Estymator bayesowski* jest to wartość oczekiwana *a posteriori* tej zmiennej:

$$\mathbb{E}(\theta_1|y) = \int\cdots\int \theta_1 p(\theta,\mu,\sigma^{-2},\upsilon^{-2}|y)\mathrm{d}\theta_2\cdots\mathrm{d}\theta_k\mathrm{d}\mu\mathrm{d}\sigma^{-2}\mathrm{d}\upsilon^{-2}.$$

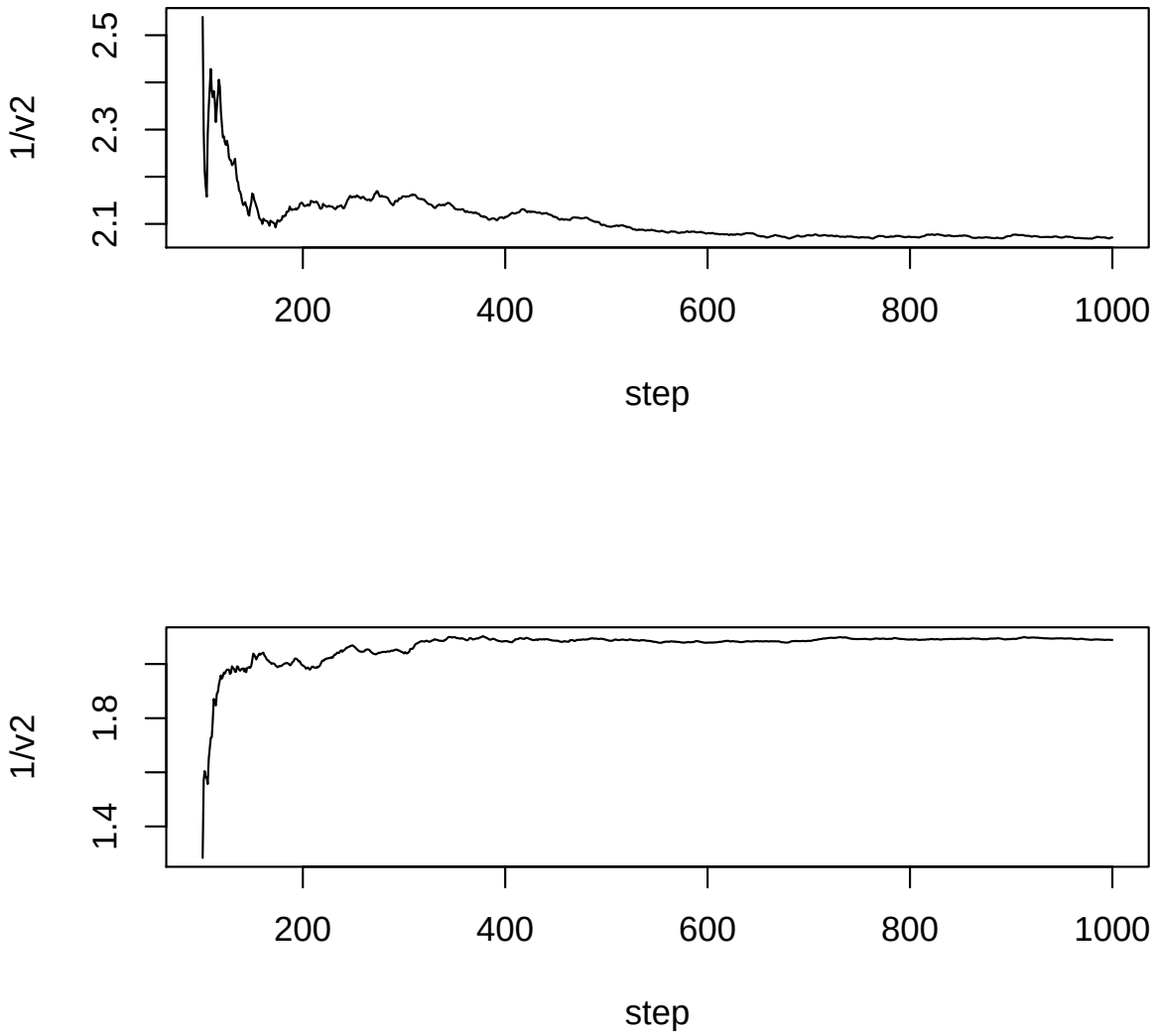
Aproksymacją MCMC interesującej nas wielkości są średnie wzdłuż trajektorii łańcucha:

$$\theta_1(X_0), \theta_1(X_1), \ldots, \theta_1(X_t), \ldots,$$
gdzie $\theta_1(x) = \theta_1$ dla $x = (\theta_1, \ldots, \theta_k, \mu, \sigma^{-2}, \upsilon^{-2})$.



Rysunek 5.1: Trajektorie zmiennej υ^{-2} dla dwóch punktów startowych. Hierarchiczny model komponentów wariancyjnych.

Na Rysunku 5.1 pokazane są dwie przykładowe trajektorie współrzędnej υ^{-2} dla PG poruszającego się po przestrzeni $k + 3 = 1003$ wymiarowej (model uwzględniający 1000 małych



Rysunek 5.2: Skumulowane średnie zmiennej v^{-2} dla dwóch punktów startowych.Hierarchiczny model komponentów wariancyjnych.

obszarów). Dwie trajektorie odpowiadają dwu różnym punktom startowym. Dla innych zmiennych rysunki wyglądają bardzo podobnie. Uderzające jest to, jak szybko trajektoria zdaje się „osiągać" rozkład stacjonarny, przynajmniej wizualnie. Na Rysunku 5.2 pokazane są kolejne „skumulowane" średnie dla tych samych dwóch trajektorii zmiennej v^{-2}.

Model mieszanek normalnych

Tak, jak w modelu komponentów wariancyjnych, rozważamy obserwacje podzielone na grupy. Różnica jest taka, że podział jest „ukryty". Dane nie zawierają informacji o tym, które jednostki pochodzą z tej samej grupy, a które z różnych grup. Zakładamy, że mamy próbkę losową z mieszanki rozkładów prawdopodobieństwa $\sum_{j=1}^{k} q_j P_j(\cdot)$, gdzie $\sum_{j=1}^{k} q_j = 1$. Każdy z rozkładów $P_j(\cdot)$ zależy od nieznanych parametrów. Prawdopodobieństwa q_j też są nieznane. W modelu, który rozpatrzymy zakłada się, że liczba komponentów k jest znana. W dalszym ciągu rozpatrujemy mieszanki rozkładów normalnych i następującą hierarchię rozkładów *a priori*:

- $Y_1, \ldots, Y_n \sim_{\text{i.i.d.}} \sum_{j=1}^{k} q_j \mathrm{N}(\mu_j, \sigma_j^2)$,
- $(q_1, \ldots, q_k) \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_k)$,
- $\mu_1, \ldots, \mu_k \sim_{\text{i.i.d.}} \mathrm{N}(m, v^2)$,
- $\sigma_1^{-2}, \ldots, \sigma_k^{-2} \sim_{\text{i.i.d.}} \mathrm{Gamma}(\gamma, \lambda)$.

Hiperparametry $m, v^2, \alpha, \gamma, \lambda$ są ustalone i znane. Rozkładem docelowym jest rozkład *a posteriori* $\pi(q, \mu, \sigma^{-2}|y)$, gdzie $\sigma^{-2} = (\sigma_1^{-2}, \ldots, \sigma_k^{-2})$.

Przedstawię poniżej próbnik Gibbsa (PG) wykorzystujący ideę *zmiennych pomocniczych*. W naszym modelu, te zmienne pomocnicze, $c_1, \ldots, c_n$, po prostu wskazją, do którego komponentu mieszanki należą poszczególne obserwacje. Innymi słowy,

- $\mathbb{P}(c_i = j|q) = q_j$ (*a priori*),
- $Y_i|c_i = j \sim \mathrm{N}(\mu_j, \sigma_j^2)$ niezależnie od reszty zmiennych.

Ogólnie mówiąc, zmienne pomocnicze ułatwiają konstrukcję algorytmów MCMC. W modelu mieszanek, są same w sobie interesujące.

Łączna gęstość a posteriori w naszym modelu jest następująca:

$$\pi(q, c, \mu, \sigma^{-2}|y) \propto \prod_{i=1}^{n} q_{c_i} (\sigma_{c_i}^{-2})^{1/2} \exp\left(-\frac{\sigma_{c_i}^{-2}}{2}(y_i - \mu_{c_i})^2\right) \cdot \prod_{j=1}^{k} q_j^{\alpha_j - 1} \cdot \prod_{j=1}^{k} \exp\left(-\frac{1}{2v^2}(\mu_j - m)^2\right) \cdot \prod_{j=1}^{k} (\sigma_j^{-2})^{\gamma-1} \exp\left(-\lambda \sigma_j^{-2}\right).$$

Z tego wzoru łatwo wydobyć postać pełnych rozkładów warunkowych (*full conditionals*).

- Rozkład warunkowy c. Niezależnie dla $i = 1, \ldots, n$ mamy

$$\mathbb{P}(c_i = j | q, \mu, \sigma^{-2}, y) \propto q_j (\sigma_j^{-2})^{1/2} \exp\left(-\frac{\sigma_j^{-2}}{2} (y_i - \mu_j)^2 \right).$$

  Jest to łatwy do symulowania rozkład dyskretny na zbiorze $\{1, \ldots, k\}$.

- Rozkład warunkowy q. Grupując wyrażenia zawierające q_j, dostajemy

$$\pi(q | c, \mu, \sigma^{-2}, y) \propto \prod_{j=1}^{k} q_j^{\alpha_j + n_j - 1},$$

  gdzie $n_j = \sum_{i=1}^{n} \mathbb{1}(c_i = j)$. Rozkładem warunkowym jest więc $\mathrm{Dir}(\alpha_k + n_k, \ldots, \alpha_k + n_k)$.

- Rozkład warunkowy μ. Przepiszmy tę część wzoru na łączną gęstość *a posteriori*, która zawiera μ w następujący sposób:

$$\pi(\mu | q, c, \sigma^{-2}, y) \propto \prod_{j=1}^{k} \exp\left(-\frac{\sigma_j^{-2}}{2} \sum_{i: c_i = j} (y_i - \mu_j)^2 \right) \exp\left(-\frac{1}{2v^2} (\mu_j - m)^2 \right).$$

  Niezależnie dla $j = 1, \ldots, k$, obliczamy rozkład μ_j sprowadzając funkcje kwadratowe „pod znakiem exp" do postaci kanonicznej, otrzymując:

$$\mu_j \sim \mathrm{N}\left(z_j \bar{y_j} + (1 - z_j) m, \frac{v^2}{n_j \sigma_j^{-2} + 1} \right),$$
$$z_j = \frac{n_j \sigma_j^{-2} v^2}{n_j \sigma_j^{-2} v^2 + 1}, \quad \bar{y_j} = \frac{1}{n_j} \sum_{i: c_i = j} y_i.$$

  Zwróćmy uwagę na to, że w każdym kroku PG używamy bieżących wartości zmiennych c_i, czyli aktualizujemy przynależność obserwacji do grup.

- Rozkład warunkowy σ^{-2}. Podobnie jak w poprzednim punkcie, przepisujemy część wzoru na łączną gęstość *a posteriori*, która zawiera σ^{-2} i rozpoznajemy, że pełny rozkład warunkowy jest niezależny dla każdej współrzędnej i równy

$$\sigma_j^{-2} \sim \mathrm{Gamma}\left(\gamma + \frac{n_j}{2}, \lambda + \frac{1}{2} \sum_{i: c_i = j} (y_i - \mu_j)^2 \right).$$

5.2 Markowowskie pola losowe

Model auto-logistyczny

Niech $x = (x_1, \ldots, x_d)$ będzie wektorem (konfiguracją) binarnych zmiennych losowych na przestrzeni $\mathcal{X} = \{0,1\}^d$. Rozważmy następujący *rozkład Gibbsa*:

$$\pi(x) = \frac{1}{z} \exp\left\{ \sum_{i,j=1}^{d} \alpha_{ij} x_i x_j \right\}.$$

Rolę parametru gra macierz $\alpha = (\alpha_{ij})$. Zakłada się, bez straty ogólności, że jest to macierz symetryczna. Stała normująca $z = \sum_{x \in \mathcal{X}} \exp\left\{ \sum_{i,j=1}^{d} \alpha_{ij} x_i x_j \right\}$ jest typowo (dla dużych d) *niemożliwa do obliczenia*.

Próbnik Gibbsa pozwala łatwo symulować konfiguracje o rozkładzie p_α w modelu auto-logistycznym. „Pełne" rozkłady warunkowe (*full conditionals*) są identyczne, jak w modelu regresji logistycznej:

$$\pi(x_i = 1 \mid x_{-i}) = \frac{\exp\left(\alpha_{ii} + \sum\limits_{\substack{j=1 \\ j \neq i}}^{d} \alpha_{ij} x_j \right)}{1 + \exp\left(\alpha_{ii} + \sum\limits_{\substack{j=1 \\ j \neq i}}^{d} \alpha_{ij} x_j \right)},$$

gdzie $x_{-i} = (x_j, j \neq i)$ (zmienna x_i odgrywa rolę „zmiennej odpowiedzi, zaś x_{-i} są „zmiennymi" objaśniającymi.

5.2.1 Przykład (Statystyka przestrzenna)**.** W zastosowaniach „przestrzennych" indeks $i \in \{1, \ldots, d\}$ interpretuje się jako „miejsce". Zbiór miejsc wyposażony jest w strukturę grafu. Krawędzie łączą miejsca „sąsiadujące". Piszemy $i \sim j$. Tego typu modele mogą opisywać na przykład rozprzestrzenianie się chorób lub występowanie pewnych gatunków. Wartosc $x_i = 1$ oznacza obecność gatunku lub występowanie choroby w miejscu i. Najprostszy model zakłada, że każda zmienna x_i zależy tylko od swoich „sąsiadów" i to w podobny sposób w całym rozpatrywanym obszarze. W takim modelu mamy tylko dwa parametry $\alpha = (\alpha_0, \alpha_1)$:

$$\alpha_{ij} = \begin{cases} 0 & i \not\sim j, i \neq j; \\ \alpha_1 & i \sim j; \\ \alpha_0 & i = j. \end{cases}$$

Parametr α_0 opisuje „skłonność" pojedynczej zmiennej do przyjmowania wartości 1, zaś parametr α_1 odpowiada za zależność od zmiennych sąsiadujących (zakaźność choroby, powiedzmy). W typowej dla statystyki przestrzennej sytuacji, rozpatruje się nawet dziesiątki tysięcy „miejsc". Stała $z(\alpha)$ jest wtedy sumą niewyobrażalnie wielu (dokładnie 2^d) składników. $\triangle$

Markowowskie pola losowe

Markowowskie pola losowe są uogólnieniem Przykładu 5.2.1. Niech $(\mathcal{S}, \mathcal{E})$ będzie nieskierowanym grafem. Wyobraźmy sobie, że elementy $s \in S$ reprezentują „miejsca" w przestrzeni lub na płaszczyźnie, zaś krawędzie grafu łączą miejsca „sąsiadujące" ze sobą. Taka interpretacja jest związana z zastosowaniami do statystyki „przestrzennej" i przetwarzania obrazów. Model, który przedstawimy ma również zupełnie inne interpretacje, ale pozostaniemy przy sugestywnej terminologii „przestrzennej":

- $\mathcal{S}$ — zbiór miejsc,
- $\{s, t\} \in \mathcal{E}$ — miejsca s i t sąsiadują — będziemy wtedy pisać $s \sim t$,
- $\partial t = \{s : s \sim t\}$ — zbiór sąsiadów miejsca t.

Niech $\Lambda = \{1, \ldots, l\}$ będzie skończonym zbiorem. Powiedzmy, że elementy $a \in \Lambda$ są „kolorami" które mogą być przypisane elementom zbioru $\mathcal{S}$. Konfiguracją nazywamy dowolną funkcję $x : \mathcal{S} \to \Lambda$. Będziemy mówić że $x_s = x(s)$ jest s-tą współrzędną konfiguracji x i stosować oznaczenia podobne jak dla wektorów:

$$x = (x_s) = (x_s : s \in \mathcal{S}).$$

W zadaniach przetwarzania obrazów, miejsca są pikslami na ekranie i konfigurację utożsamiamy z ich pokolorowaniem, a więc z cyfrową reprezentacją obrazu. Zbiór Λ gra rolę „palety kolorów". Niekiedy założenie o skończoności zbioru Λ staje się niewygodne. Dla czarno-szaro-białych obrazów „pomalowanych" różnymi odcieniami szarości, wygodnie przyjąć, że $\Lambda = [0, 1]$ lub $\lambda = [0, \infty[$. Tego typu modyfikacje są dość oczywiste i nie będę się nad tym zatrzymywał. Dla ustalenia uwagi, wzory w tym podrozdziale dotyczą przypadku skończonego zbioru „kolorów". Przestrzenią konfiguracji jest zbiór $\mathcal{X} = \Lambda^{\mathcal{S}}$. Dla konfiguracji x i miejsca t, niech

- $x_{-t} = (x_s : s \neq t) = (x_s : s \in \mathcal{S} \setminus \{t\})$ — konfiguracja z pominiętą t-tą współrzędną,
- $x_{\partial t} = (x_s : s \in \partial t)$ — konfiguracja ograniczona do sąsiadów miejsca t.

Jeśli $H : \mathcal{X} \to \mathbb{R}$ i $\beta \geqslant 0$ to **rozkładem Gibbsa** nazywamy rozkład prawdopodobieństwa na przestrzeni konfiguracji dany wzorem

$$\pi_\beta(x) = \frac{1}{z(\beta)} \exp[-\beta H(x)].$$

Ze względu na inspiracje pochodzące z fizyki statystycznej, funkcję H nazywamy energią, β jest (z dokładnością do stałej) odwrotnością temperatury. Stała normująca wyraża się wzorem

$$z(\beta) = \sum_{x \in \mathcal{X}} \exp[-\beta H(x)].$$

i jest typowo *niemożliwa do obliczenia.*

Oczywiście, każdy rozkład prawdopodobieństwa π na $\mathcal{X}$ daje się zapisać jako rozkład Gibbsa, jeśli położyć $H(x) = -\log \pi(x)$, $\beta = 1$ i umownie przyjąć, że $-\log 0 = \infty$ (czyli konfiguracje niemożliwe mają nieskończoną energię). Nie o to jednak chodzi. Ciekawe są rozkłady Gibbsa, dla których fukcja energii ma specjalną postać związaną z topologią grafu „sąsiedztw". Ograniczymy się do ważnej podklasy markowowskich pól losowych (MPL), mianowicie do sytuacji gdy energia jest sumą „oddziaływań" lub „interakcji" między parami miejsc sąsiadujących i składników zależnych od pojedynczych miejsc. Dokładniej, załóżymy że

$$H(x) = \sum_{s\sim t} V(x_s, x_t) + \sum_s U_s(x_s), \tag{5.2.2}$$

dla pewnych funkcji $V : \Lambda \times \Lambda \to \mathbb{R}$ i $U_s : \Lambda \to \mathbb{R}$. Funkcja $V(x_s, x_t)$ opisuje „potencjał interakcji pomiędzy s i t", zaś $U_s(a)$ jest wielkością związaną z „tendencją miejsca s do przybrania koloru a. Zwróćmy uwagę, że potencjał V jest jednorodny ($V(a, b)$ zależy tylko od „kolorów" $a, b \in \Lambda$ ale nie od miejsc), zaś $U_s(a)$ może zależeć zarówno od $a \in \Lambda$ jak i od $s \in \mathcal{S}$. W modelach fizyki statystycznej zazwyczaj $U_s(a) = U(a)$ jest jednorodnym „oddziaływaniem zewnętrznym" ale w modelach rekonstrukcji obrazów nie można tego zakładać.

5.2.3 Przykład (Model Pottsa)**.** Niech Λ będzie zbiorem skończonym i

$$H(x) = \alpha \sum_{s\sim t} \mathbb{1}(x_s \neq x_t).$$

Ta funkcja opisuje „tendencję sąsiednich miejsc do przybierania tego samego koloru". Jeśli $\alpha > 0$ to preferowane są konfiguracje złożone z dużych, jednobarwnych plam. △

Generowanie markowowskich pól losowych

Użyteczność MPL w różnorodnych zastosowaniach związana jest z istnieniem efektywnych algorytmów symulacyjnych MCMC. Zarówno próbnik Gibbsa, jak i algorytm Metropolisa są w zastosowaniach do MPL wyjątkowo proste. Próbnik Gibbsa opiera się na następującym fakcie.

5.2.4 Twierdzenie (Pełne rozkłady warunkowe dla MPL)**.** *Jeżeli π_β jest rozkładem Gibbsa z energią daną wzorem (5.2.2), to*

$$\pi_\beta(x_s|x_{-s}) = \pi_\beta(x_s|x_{\partial s}) = \frac{1}{z_s(\beta)} \exp\left[-\beta H_s(x)\right],$$

gdzie

$$H_s(x) = \sum_{t\in\partial s} V(x_s, x_t) + U(x_s),$$

$z_s(\beta) = \sum_{a\in\Lambda} \exp[H_s(x_{a\rightsquigarrow s})]$. *Symbol $x_{a\rightsquigarrow s}$ oznacza konfigurację powstałą z x przez wpisanie koloru a w miejscu s.*

Dowód. Skorzystamy z elementarnej definicji prawdopodobieństwa warunkowego (poniżej piszemy $\pi_\beta(\cdot) = \pi(\cdot)$, bo parametr β jest ustalony):

$$\begin{aligned}
\pi(x_s|x_{-s}) &= \frac{\pi(x)}{\pi(x_{-s})} = \frac{\pi(x)}{\sum\limits_a \pi(x_{a\rightsquigarrow s})} \\
&= \frac{\exp -\beta H(x)}{\sum_a \exp -\beta H(x_{a\rightsquigarrow s})} \\
&= \frac{\exp -\beta \left(\sum\limits_{t:t\sim s} V(x_s, x_t) + \sum\limits_{t\sim w, t\neq s, w\neq s} V(x_t, x_w) + U_s(x_s) + \sum\limits_{t\neq s} U_t(x_t) \right)}{\sum\limits_a \exp -\beta \left(\sum\limits_{t:t\sim s} V(a, x_t) + \sum\limits_{t\sim w, t\neq s, w\neq s} V(x_t, x_w) + U_s(a) + \sum\limits_{t\neq s} U_t(x_t) \right)} \\
&= \frac{\exp -\beta \left(\sum\limits_{t:t\sim s} V(x_s, x_t) + U_s(x_s) \right)}{\sum\limits_a \exp -\beta \left(\sum\limits_{t:t\sim s} V(a, x_t) + U_s(a) \right)} \\
&= \frac{\exp -\beta H_s(x)}{z_s(\beta)}.
\end{aligned}$$

Ponieważ otrzymany wynik zależy tylko od x_s i $x_{\partial s}$, więc $\pi(x_s|x_{-s}) = \pi(x_s|x_{\partial s})$. Ten wniosek jest pewną formą własności Markowa. □

Zauważmy, że obliczenie $H_s(x)$ jest łatwe, bo suma $\sum_{t\in\partial s} \cdots$ zawiera tylko tyle składników, ile jest sąsiadów miejsca s. Obliczenie $z_s(\beta)$ też jest łatwe, bo suma $\sum_{a\in\Lambda} \cdots$ zawiera tylko $l = |\Lambda|$ składników. Ale nawet nie musimy obliczać stałej normującej $z_s(\beta)$ żeby generować z rozkładu

$$\pi(x_s = a|x_{\partial s}) \propto \exp -\beta \left(\sum_{t:t\sim s} V(a, x_t) + U_s(a) \right). \tag{5.2.5}$$

Na tym opiera się implementacja próbnika Gibbsa. Wersję PG z „systemstycznym przeglądem miejsc" można zapisać tak:

```
for s ∈ 𝒮 do
    begin
    Gen a ∼ π(x_s = ·|x_∂s);
    x := x_{a⇝s}
    end
```

Faktycznie już ten algorytm spotkaliśmy na początku tego rozdziału, dla szczególnego przypadku modelu auto-logistycznego.

Rekonstrukcja obrazów

Bayesowski model rekonstrukcji obrazów został zaproponowany w pracy Gemana i Gemana w 1987 roku. Potem zdobył dużą popularność i odniósł wiele sukcesów. Model łączy idee zaczerpnięte ze statystyki bayesowskiej i fizyki statystycznej. Cyfrową reprezentację obrazu utożsamiamy z konfiguracją kolorów na wierzchołkach grafu, czyli z elementem przestrzeni $\mathcal{X} = \Lambda^{\mathcal{S}}$. Przyjmijmy, że „idealny obraz", czyli to co chcielibyśmy zrekonstruować jest konfiguracją $x = (x_s) \in \mathcal{X}$. Niestety, obraz jest „zakłócony" lub „zaszumiony". Możemy tylko obserwować konfigurację $y = (y_s) \in \mathcal{Y}$ reprezentującą zakłócony obraz. Zbiór kolorów w obrazie y nie musi być identyczny jak w obrazie x. Ważne jest to, że zniekształcenie modelujemy probabilistycznie przy pomocy rodziny rozkładów warunkowych $f(y|x)$. Dodatkowo zakładamy, że obraz x pojawia się losowo, zgodnie z rozkładem prawdopodobieństwa $\pi(x)$. Innymi słowy, „idealny" obraz x oraz „zniekształcony" obraz y traktujemy jako realizacje zmiennych losowych $X : \Omega \to \mathcal{X}$ i $Y : \Omega \to \mathcal{Y}$,

$$\pi(x) = \mathbb{P}(X = x), \qquad f(y|x) = \mathbb{P}(Y = y|X = x).$$

W ten sposób buduje się statystyczny model bayesowski, w którym

- Y jest obserwowaną zmienną losową,
- x jest nieznanym parametrem traktowanym jako zmienna losowa X.

Oczywiście, π gra rolę rozkładu *a priori*, zaś f jest wiarogodnością. Być może użycie literki x na oznaczenie parametru jest niezgodne z tradycyjnymi oznaczeniami statystycznymi, ale z drugiej strony jest wygodne. Wzór Bayesa mówi, że rozkład *a posteriori* jest następujący.

$$\pi_y(x) = \mathbb{P}(X = x|Y = y) \propto f(y|x)\pi(x).$$

Pomysł Gemana i Gemana polegał na tym, żeby modelować rozkład *a priori* π jako MPL. Załóżmy, że π jest rozkładem Gibbsa,

$$\pi(x) \propto \exp\left(-H(x)\right), \tag{5.2.6}$$

gdzie

$$H(x) = \alpha \sum_{s \sim t} V(x_s, x_t). \tag{5.2.7}$$

Energia „*a priori*" zawiera tu tylko składniki reprezentujące oddziaływania między parami miejsc sąsiednich. Funkcja $V(a, b)$ zazwyczaj ma najmniejszą wartość dla $a = b$ i rośnie wraz z „odległością" między a i b (jakkolwiek tę odległość zdefiniujemy). W ten sposób „nagradza" konfiguracje w których sąsiednie miejsca są podobnie pokolorowane. Im większy parametr $\alpha > 0$, tym bardziej prawdopodobne są obrazy zawierające jednolite plamy kolorów.

Trzeba jeszcze założyć coś o „wiarogodności" f. Dla uproszczenia opiszę tylko najprostszy model, w którym kolor y_s na obserwowanym obrazie zależy tylko od koloru x_s na obrazie idealnym. Intuicyjnie znaczy to, że „zaszumienie" ma ściśle lokalny charakter. Matematycznie znaczy to, że

$$f(y|x) = \prod_s f(y_s|x_s)$$

(pozwolę sobie na odrobinę nieścisłości aby uniknąć nowego symbolu na oznaczenie $f(y_s|x_s)$). Zapiszemy teraz „wiarogodność" f w postaci zlogarytmowanej. Jeśli położymy $-\log f(y_s|x_s) = U_s(x_s)$, to otrzymujemy następujący wzór:

$$\pi_y(x) \propto \exp\left(-H_y(x)\right), \tag{5.2.8}$$

gdzie

$$H_y(x) = \alpha \sum_{s\sim t} V(x_s, x_t) - \sum_s \log f(y_s|x_s). \tag{5.2.9}$$

Okazuje się zatem, że rozkład *a posteriori* ma podobną postać do rozkładu *a priori*. Też jest rozkładem Gibbsa, a różnica polega tylko na dodaniu składników reprezentujących oddziaływania zewnętrzne $U_s(x_s) = -\log f(y_s|x_s)$. Pamiętajmy przy tym, że y jest w świecie bayesowskim ustalone. W modelu rekonstrukcji obrazów „oddziaływania zewnętrzne" zależą od y i „wymuszają podobieństwo" rekonstruowanego obrazu do obserwacji. Z kolei „oddziaływania między parami" są odpowiedzialne za wygładzenie obrazu. Lepiej to wyjaśnimy na przykładzie.

5.2.10 Przykład (Losowe „przekłamanie koloru" i wygładzanie Pottsa)**.** Załóżmy, że $\Lambda = \{1, \ldots, l\}$ jest naprawdę paletą kolorów, na przykład

$$\Lambda = \{Czerwony, Niebieski, Pomaraczowy, Zielony\}.$$

Przypuśćmy, że mechanizm losowego „przekłamania" polega na tym, że w każdym pikslu, kolor obecny w idealnym obrazie x jest z prawdopodobieństwem $1-\varepsilon$ niezmieniony, a z prawdopodobieństwem ε zmienia się na losowo wybrany inny kolor. Tak więc zarówno x jak i y należą do tej samej przestrzeni $\Lambda^{\mathcal{S}}$,

$$f(y_s|x_s) = \begin{cases} 1-\varepsilon & \text{dla } y_s = x_s; \\ \varepsilon/(l-1) & \text{dla } y_s \neq x_s. \end{cases}$$

Można za rozkład *a priori* przyjąć rozkład Pottsa z Przykładu 5.2.3. Rozkład *a posteriori* ma funkcję energii daną następującym wzorem (z $J > 0$):

$$H_y(x) = \alpha \sum_{s\sim t} \mathbb{1}(x_s \neq x_t) - \sum_s \left[\log(1-\varepsilon)\mathbb{1}(x_s = y_s) + \log(\varepsilon/(l-1))\mathbb{1}(x_s \neq y_s)\right].$$

Pierwszy składnik w tym wzorze pochodzi od rozkładu a priori (z modelu Pottsa) i „nagradza" konfiguracje w których dużo sąsiednich punktów jest pomalowanych na ten sam kolor. Powoduje to, że obrazy x składające się z jednolotych dużych „plam" są preferowane. Drugi składnik pochodzi od obserwowanegj konfiguracji y i jest najmniejszy dla $x = y$. Powoduje to, ze obrazy x mało się różniące od y są bardziej prawdopodobne. Rozkład *a posteriori* jest pewnym kompromisem pomiędzy tymi dwoma konkurującymi składnikami. Parametr α jest „wagą" pierwszego składnika i dlatego odgrywa rolę „parametru wygładzającego". Im większe α tym odtwarzany obraz będzie bardziej regularny (a tym mnie będzie starał się upodobnić do y). I odwrotnie, małe α powoduje ściślejsze dopasowanie x do y ale mniejszą „regularność" x. △

Jeszcze lepiej to samo widać na przykładzie tak zwanego „szumu gaussowskiego".

5.2.11 Przykład (Addytywny szum gaussowski)**.** Załóżmy, że x jest konfiguracją „poziomów szarości" czyli, powiedzmy, $\Lambda \subseteq [0, \infty[$. Mechanizm losowego „zaszumienia" polega na tym, że zamiast poziomu szarości x_s obserwujemy $y_s \sim \mathrm{N}(x_s, \sigma^2)$. Innymi słowy,

$$f(y_s|x_s) \propto \exp\left[-\frac{1}{2\sigma^2}(y_s - x_s)^2\right].$$

Przestrzenią obserwowanych konfiguracji y jest tutaj (formalnie) $\mathbb{R}^{\mathcal{S}}$ (faktycznie, raczej $[0, \infty[^{\mathcal{S}}$). Rozkład *a posteriori* ma funkcję energii daną następującym wzorem:

$$H_y(x) = \alpha \sum_{s \sim t} V(x_s \neq x_t) + \frac{1}{2\sigma^2} \sum_s (y_s - x_s)^2.$$

Jeśli rozpatrujemy model ze skończoną liczbą poziomów szarości dla konfiguracji x to można pierwszy składnik określić tak jak w poprzednim przykładzie, czyli zapożyczyć z modelu Pottsa. Bardziej naturalne jest określenie $V(a, b)$ w taki sposób, aby większe różnice pomiędzy poziomami a i b były silniej karane. Parametr α jest, jak poprzednio, odpowiedzialny za stopień wygładzenia. △

5.3 Zadania i uzupełnienia

5.1 Ćwiczenie. Niech X będzie losową macierzą $d \times d$ o elementach 0 lub 1, to znaczy przestrzenią stanów jest $\mathcal{X} = \{0, 1\}^{d \times d}$. Relacja „sąsiedztwa" na kwadracie $\{1, \ldots, d\}^2$ jest następująca:

$$(i, j) \sim (k, l) \quad \text{wtw} \quad |i - k| + |j - l| = 1.$$

Rozważamy rozkład Gibbsa

$$\pi(x) \propto \exp[-\beta H(x)],$$

gdzie funkcja energii jest zdefiniowana wzorem

$$H(x) = \alpha_0 \sum_{(i,j)} x_{ij} + \alpha_1 \sum_{(i,j)\sim(k,l)} x_{ij}x_{kl}.$$

Zadanie polega na próbkowaniu X z rozkładu π (w przybliżeniu) przy pomocy MCMC.

- Można wybrać albo algorytm Metropolisa albo próbnik Gibbsa.
- Można wybrać albo systematyczny przegląd miejsc (po kolei zmieniamy $i = 1, \ldots, d$, $j = 1, \ldots, d$) albo losowy wybór miejsc: $(i, j) \sim U(\{0, 1\}^{d\times d})$.

Sugestia: Żeby sobie ułatwić, można zdefiniować macierz X jako macierz $(d+2) \times (d+2)$, w której pierwszy i ostatni wiersz oraz pierwsza i ostatnia kolumna są zerowe.

Uwaga: Można się pobawić zmieniając parametry i rysując wylosowane macierze. To ciekawe i pouczające. Ale przede wszystkim należy skontrolować poprawność.

- Przeprowadzić doświadczenie dla parametrów: $d = 20$, $\alpha_0 = 4$, $\alpha_1 = -2$, $\beta = 0.5$.
- Wyestymować rozkłady statystyk dostatecznych

  $$S = \sum_{(i,j)} X_{ij}, \qquad N = \sum_{(i,j)\sim(k,l)} X_{ij}X_{kl}.$$

  Obliczamy estymatory $\mathbb{E}_\pi(S)$ i $\mathbb{E}_\pi(N)$.
- Wyestymować rozkad Boltzmanna (rozkład na poziomach energii): jest to z definicji rozkład dyskretny $\mathbb{P}_\pi(H(X) = h)$ dla różnych wartości h. Najlepiej zrobić histogram.

5.2 Ćwiczenie. Rozpatrzyć dokładnie przypadek macierzy 2×2. Możliwych stanów jest $2^4 = 16$, ale możliwych poziomów energetycznych jeszcze mniej. Zauważmy, że

$$H(x) = \alpha_0 S(x) + \alpha_1 N(x),$$

gdzie $S(x) = \sum_{(i,j)} x_{ij}$, $N(x) = \sum_{(i,j)\sim(k,l)} x_{ij}x_{kl}$. *Uwaga:* W tej ostatniej sumie każda para *nieuporządkowana* sąsiednich wierzchołków liczy się 1 raz.

Można *obliczyć* rozkład Gibbsa i rozkład Boltzmanna teoretycznie i porównać z symulacjami!

- Obliczyć rozkład Gibbsa $x \mapsto \pi(x)$, $x \in \{0, 1\}^{2\times2}$.
- Obliczyć rozkład łączny (S, N).
- Obliczyć rozkład Boltzmanna $h \mapsto \mathbb{P}_\pi(H(X) = h)$, dla wszystkich możliwych poziomów h.
- Uruchomić symulacje (dla $\alpha_0 = 4$, $\alpha_1 = -2$, $\beta = 0.5$) i porównać z wartościami obliczonymi (patrz powyżej).

Rozdział 6

Elementy teorii łańcuchów Markowa

6.1 Podstawowe określenia i oznaczenia

W tym rozdziale rozważamy jednorodny łańcuch Markowa $X_0, X_1, \ldots, X_n, \ldots$ na skończonej przestrzeni stanów $\mathcal{X} = \{1, \ldots, d\}$. Będziemy posługiwać się wygodną i zwięzłą notacją wektorowo-macierzową. Macierz przejścia o wymiarach $d \times d$ oznaczamy $P = (P(x, x'))_{x,x' \in \mathcal{X}}$. Rozkład początkowy utożsamiamy z wektorem wierszowym $\nu^\top = \big(\nu(1), \ldots, \nu(x), \ldots, \nu(d)\big)$. W dalszym ciągu, mówiąc o łańcuchu Markowa, będziemy mieli na myśli ustaloną macierz przejścia P i dowolnie wybrany rozkład początkowy ν. Przyjmujemy oznaczenie $\mathbb{P}_\nu(\cdot)$. W szczególności, $\mathbb{P}_x(\cdot) = \mathbb{P}(\cdot | X_0 = x)$, dla $x \in \mathcal{X}$. Analogicznie będziemy oznaczali wartość oczekiwaną: $\mathbb{E}_\nu$ lub $\mathbb{E}_x$. Zauważmy, że $\mathbb{P}(X_{n+2} = x'' | X_n = x) = \sum_{x'} P(x, x')P(z, x') = P^2(x, x'')$. Ogólniej, macierz przejścia w m krokach jest m-tą potęgą macierzy P:

$$\mathbb{P}(X_{n+m} = x' | X_n = x) = P^m(x, x').$$

Rozkład *brzegowy* zmiennej losowej X_n jest wektorem $\nu^\top P^n$:

$$\mathbb{P}(X_n = x) = \big(\nu^\top P^n\big)(x).$$

Interesują nas głównie łańcuchy, które „zmierzają w kierunku położenia równowagi". Aby uściślić co to znaczy „równowaga", przypomnijmy pojęcie stacjonarności. Rozkład π jest stacjonarny jeśli dla każdego stanu x',

$$\pi(x') = \sum_x \pi(x) P(x, x').$$

W notacji macierzowej: $\pi^\top = \pi^\top P$. Stąd oczywiście wynika, że $\pi^\top = \pi^\top P^n$.

Mówimy, że łańcuch jest nieprzywiedlny, jeśli dla dowolnych stanów $x, x' \in \mathcal{X}$ istnieje n takie, że $P^n(x, x') > 0$ (można przejść z x do x').

Poniższy prosty fakt można uzasadnić na wiele sposobów. W następnym podrozdziale przytoczymy, wraz z dowodem, piękne twierdzenie Kaca (Twierdzenie 6.2.4), które implikuje Twierdzenie 6.1.1.

6.1.1 Twierdzenie. *Jeśli łańcuch Markowa jest nieprzywiedlny, to istnieje dokładnie jeden rozkład stacjonarny π, przy tym $\pi(x) > 0$ dla każdego $x \in \mathcal{X}$.*

Uwaga. Podkreślmy stale obowiązujące w tym rozdziale założenie, że *przestrzeń stanów jest skończona*. To założenie jest istotne w Twierdzeniu 6.1.1 i to samo dotyczy dalszych rozważań. Istnieją co prawda odpowiedniki sformułowanych tu twierdzeń dla przypadku ogólnej przestrzeni stanów (nieskończonej, a nawet „ciągłej" takiej jak $\mathbb{R}^d$) ale wymagają one dodatkowych, niełatwych do sprawdzenia założeń. Przystępny i bardzo elegancki wykład teorii łańcuchów Markowa na ogólnej przestrzeni stanów można znaleźć w pracy Nummelina [9]. Przeglądowy artykuł Robertsa i Rosenthala [14] zawiera dużo dodatkowych informacji na ten temat. Obie cytowane prace koncentrują się na tych własnościach łańcuchów, które są istotne z punktu widzenia algorytmów Monte Carlo. Z kolei piękna książka Brémaud [2] ogranicza się do przestrzeni dyskretnych (skończonych lub przeliczalnych).

6.2 Regeneracja

Przedstawimy w tym podrozdziale konstrukcję, która prowadzi do łatwch i eleganckich dowodów twierdzeń granicznych. Podstawowa idea jest następująca. Wyróżnia się jeden ustalony stan, powiedzmy $x_* \in \mathcal{X}$. W każdym momencie wpadnięcia w x_* następuje „odnowienie" i dalsza ewolucja łańcucha jest niezależna od przeszłości.

Niech, dla ustalonego $x_* \in \mathcal{X}$,

$$T = T^{x_*} = \min\{n > 0 : X_n = x_*\}. \tag{6.2.1}$$

Przyjmujemy przy tym naturalną konwencję: $T = \infty$, jeśli $X_n \neq x_*$ dla każdego $n \geqslant 1$. Zmienna losowa T jest więc czasem pierwszego dojścia do stanu x_*. Jeśli założymy, że łańcuch startuje z punktu x_*, to T jest czasem pierwszego powrotu.

6.2.2 Lemat. *Jeżeli łańcuch jest nieprzywiedlny, to istnieją stałe c i $\gamma < 1$ takie, że dla dowolnego rozkładu początkowego ν, dowolnego x_* i $T = T^{x_*}$,*

$$\mathbb{P}_\nu(T > n) \leqslant c\gamma^n.$$

Dowód. Dla uproszczenia przyjmijmy dodatkowe założenie, że łańcuch jest nieokresowy. Wtedy dla dostatecznie dużych k wszystkie elementy macierzy P^k są niezerowe. Ustalmy k i znajdźmy liczbę $\delta > 0$ taką, że $P^k(x, x_*) \geqslant \delta$ dla wszystkich x (jest to możliwe, bo łańcuch

ma skończoną liczbę stanów). Dla dowolnego n, dobierzmy takie m, że $mk \leqslant n < (m+1)k$. Mamy wówczas

$$\begin{aligned}\mathbb{P}_\nu(T > n) &\leqslant \mathbb{P}_\nu(T > mk)\\ &\leqslant \mathbb{P}_\nu(X_0 \neq x_*, X_k \neq x_*, \ldots, X_{mk} \neq x_*)\\ &= \sum_{x_0 \neq x_*, x_1 \neq x_*, \ldots, x_m \neq x_*} \nu(x_0) P^k(x_0, x_1) \cdots P^k(x_{m-1}, x_m)\\ &\leqslant (1-\delta)^m \leqslant c\gamma^n,\end{aligned}$$

dla $\gamma = (1-\delta)^{1/k}$ i $c = (1-\delta)^{-1}$.

W przypadku łańcucha okresowego dowód nieco się komplikuje i, choć nie jest trudny, zostanie pominięty. □

6.2.3 Wniosek. *Dla łańcucha nieprzywiedlnego, dla dowolnego rozkładu początkowego ν, dowolnego x_* i $T = T^{x_*}$ mamy $\mathbb{P}_\nu(T < \infty) = 1$, a zatem $\mathbb{E}_\nu T < \infty$. Co więcej, istnieje funkcja tworząca momenty $\mathbb{E}_\nu \exp(\lambda T) < \infty$ przynajmniej dla pewnych dostatecznie małych wartości $\lambda > 0$ (w istocie dla $\lambda < -\log \gamma$).*

Podamy teraz bardzo ciekawą interpretację rozkładu stacjonarnego, wykazując przy okazji jego istnienie (Twierdzenie 6.1.1). Ustalmy dowolnie wybrany stan x_*. Udowodnimy, że średni czas, spędzony przez łańcuch w stanie x pomiędzy wyjściem z x_* i pierwszym powrotem do x_* jest *proporcjonalny* do $\pi(x)$, prawdopodobieństwa stacjonarnego.

6.2.4 Twierdzenie (Kaca)**.** *Załóżmy, że łańcuch jest nieprzywiedlny. Ustalmy $x_* \in \mathcal{X}$ i zdefiniujmy miarę α wzorem*

$$\alpha(x) = \mathbb{E}_{x_*} \sum_{i=0}^{T-1} \mathbb{1}(X_i = x) = \mathbb{E}_{x_*} \sum_{i=1}^{T} \mathbb{1}(X_i = x),$$

gdzie $T = T^{x_*}$. Wtedy:

(i) Miara α jest stacjonarna, czyli $\alpha^\top P = \alpha^\top$.

(ii) Miara α jest skończona, $\alpha(\mathcal{X}) = \mathbb{E}_{x_*}(T) = m < \infty$.

(iii) Unormowana miara $\alpha/m = \pi$ jest jedynym rozkładem stacjonarnym.

Dowód. Dla uproszczenia będziemy pisali $\mathbb{P}_{x_*} = \mathbb{P}$ i $\mathbb{E}_{x_*} = \mathbb{E}$. Zauważmy, że

$$\begin{aligned}\alpha(x) &= \mathbb{E} \sum_{i=0}^{T-1} \mathbb{1}(X_i = x) = \mathbb{E} \sum_{i=0}^{\infty} \mathbb{1}(X_i = x, T > i)\\ &= \sum_{i=0}^{\infty} \mathbb{P}(X_i = x, T > i).\end{aligned}$$

Udowodnimy teraz (i). Jeśli $x \neq x_*$, to

$$\begin{aligned}\sum_{x'} \alpha(x')P(x',x) &= \sum_{x'}\sum_{i=0}^{\infty} \mathbb{P}(X_i = x', T > i)P(x',x) \\ &= \sum_{i=0}^{\infty}\sum_{x'} \mathbb{P}(X_i = x', T > i)P(x',x) \\ &= \sum_{i=0}^{\infty} \mathbb{P}(X_{i+1} = x, T > i+1) = \sum_{i=1}^{\infty} \mathbb{P}(X_i = x, T > i) \\ &= \alpha(x),\end{aligned}$$

ponieważ $\mathbb{P}(X_0 = x) = 0$, bo $\mathbb{P}(X_0 = x_*) = 1$. Dla $x = x_*$ mamy z kolei

$$\begin{aligned}\sum_{x'} \alpha(x')P(x',x_*) &= \sum_{x'}\sum_{i=0}^{\infty} \mathbb{P}(X_i = x', T > i)P(x,x_*) \\ &= \sum_{i=0}^{\infty}\sum_{x'} \mathbb{P}(X_i = x', T > i)P(x,x_*) \\ &= \sum_{i=0}^{\infty} \mathbb{P}(X_{i+1} = x_*, T = i+1) = \sum_{i=1}^{\infty} \mathbb{P}(T = i) \\ &= 1 = \alpha(x_*),\end{aligned}$$

co kończy dowód (i).

Część (ii) jest łatwa. Równość

$$\alpha(\mathcal{X}) = \sum_x \alpha(x) = \mathbb{E}T$$

wynika wprost z definicji miary α. Fakt, że $m = \mathbb{E}T < \infty$ jest wnioskiem z Lematu 6.2.2.

Punkt (iii): istnienie rozkładu stacjonarnego

$$\pi(x) = \frac{\alpha(x)}{m}.$$

jest natychmiastowym wnioskiem z (i) i (ii). Jednoznaczność rozkładu stacjonarnego dla jest nietrudna do bezpośredniego udowodnienia. Pozostawiamy to jako ćwiczenie. W najbardziej interesującym nas przypadku łańcucha nieokresowego, jednoznaczność wyniknie też ze Słabego Twierdzenia Ergodycznego, które udowodnimy w następnym podrozdziale. □

Odnotujmy ważny wniosek wynikający z powyższego twierdzenia:

$$\pi(x_*) = \frac{1}{\mathbb{E}_{x_*}(T^{x_*})}.$$

Zjawisko odnowienia, czyli regeneracji pozwala sprowadzić badanie łańcuchów Markowa do rozpatrywania niezależnych zmiennych losowych, a więc do bardzo prostej i dobrze znanej sytuacji. Aby wyjaśnić to bliżej, zauważmy następującą oczywistą równość. Pamiętamy, że $T = T^{x_*}$ dla ustalonego x_*. Na mocy własności Markowa i jednorodności,

$$\begin{aligned}&\mathbb{P}(X_{n+1} = x_1, \ldots, X_{n+k} = x_k | T = n)\\ &\mathbb{P}(X_{n+1} = x_1, \ldots, X_{n+k} = x_k | X_n = x_*)\\ &\quad = \mathbb{P}_{x_*}(X_1 = x_1, \ldots, X_k = x_k).\end{aligned}$$

Zatem warunkowo, dla $T = n$, łańcuch „regeneruje się w momencie n" i zaczyna się zachowywać dokładnie tak, jak łańcuch który wystartował z punktu z w chwili 0. Niezależnie od przeszłości!

Zdefiniujmy teraz kolejne momenty odnowienia, czyli czasy odwiedzin stanu x_*:

$$\begin{aligned}T = T_1 &= \min\{n > 0 : X_n = x_*\},\\ T_k &= \min\{n > T_{k-1} : X_n = x_*\}.\end{aligned}$$

Momenty $0 < T_1 < \cdots < T_k < \cdots$ dzielą trajektorię łańcucha na następujące „losowe wycieczki", czyli losowej długości ciągi zmiennych losowych:

$$\begin{array}{ccccc}\underbrace{X_0, \ldots, X_{T_1-1}}_{T_1}, & \underbrace{X_{T_1}, \ldots, X_{T_2-1}}_{T_2-T_1}, & \underbrace{X_{T_2}, \ldots, X_{T_3-1}}_{T_3-T_2}, & \cdots\\ & \uparrow & \uparrow & \\ & X_{T_1} = x_* & X_{T_2} = x_* & \cdots\end{array}$$

Wycieczka zaczyna się w punkcie x_* i kończy tuż przed powrotem do x_*. Oznaczmy k-tą wycieczkę symbolem Ξ_k:

$$\begin{aligned}\Xi = \Xi_1 &= (X_0, \ldots, X_{T-1}, T),\\ \Xi_k &= (X_{T_{k-1}}, \ldots, X_{T_k-1}, T_k - T_{k-1})\end{aligned}$$

Z tego, co powiedzieliśmy wcześniej wynika, że wszystkie „wycieczki" są niezależne. Co więcej wycieczki Ξ_k mają ten sam rozkład, z wyjątkiem być może początkowej, czyli Ξ_1. Jeśli rozkład początkowy jest skupiony w punkcie x_*, to również wycieczka Ξ_1 ma ten sam rozkład (0 jest wtedy momentem odnowienia).

Podejście regeneracyjne, czyli rozbicie łańcucha na niezależne wycieczki prowadzi do ładnych i łatwych dowodów PWL i CTG dla łańcuchów Markowa. Sformułujemy najpierw pewną wersję *Mocnego Prawa Wielkich Liczb*. Rozważmy funkcję f o wartościach rzeczywistych, określoną na przestrzeni stanów. Przypomnijmy, że $\mathbb{E}_\pi f = \sum_{x\in\mathcal{X}} \pi(x) f(x)$.

6.2.5 Twierdzenie (Mocne Twierdzenie Ergodyczne)**.** *Jeśli X_n jest nieprzywiedlnym łańcuchem Markowa, to dla dowolnego rozkładu początkowego ν i każdej funkcji $f : \mathcal{X} \to \mathbb{R}$,*

$$\frac{1}{n}\sum_{i=0}^{n-1} f(X_n) \longrightarrow \mathbb{E}_\pi f \qquad (n \to \infty)$$

z prawdopodobieństwem 1.

Dowód. Zdefiniujmy sumy blokowe:

$$\Xi_0(f) = \sum_{i=0}^{T-1} f(X_i),$$
$$\Xi_k(f) = \sum_{i=T_k}^{T_{k+1}-1} f(X_i).$$

Niech $N(n) = \max\{k : T_k \leqslant n\}$, czyli $T_{N(n)}$ jest ostatnią regeneracją przed momentem n:

$$\begin{array}{lllll} 0,\ldots,T_1-1, & T_1,\ldots\ldots, & T_{N(n)},\ldots,n, & \ldots, T_{N(n)+1}-1, & T_{N(n)+1},\ldots \\ & \uparrow & \uparrow \quad \uparrow & & \uparrow \\ & X = x_* & X = x_* \quad \bullet & & X = x_* \end{array}$$

Oczywiście,

$$T_{N(n)} \leqslant n < T_{N(n)+1}. \tag{6.2.6}$$

Wiemy, że $\mathbb{E}_{x_*} T = m < \infty$. Wiemy, że T_k jest sumą k niezależnych zmiennych losowych (długości wycieczek), przy tym wszystkie składniki z wyjątkiem pierwszego mają ten sam rozkład o wartości oczekiwanej m. Wnioskujemy, że $T_k/k \to m$ z prawdopodobieństwem 1, na mocy *zwykłego Prawa Wielkich Liczb*. Rzecz jasna, tak samo $T_{k+1}/k \to m$. Podzielmy nierówność (6.2.6) stronami przez $N(n)$ i przejdźmy do granicy (korzystając z tego, że $N(n) \to \infty$ prawie na pewno). Twierdzenie o trzech ciągach pozwala wywnioskować, że

$$\frac{N(n)}{n} \to \frac{1}{m} \text{ p.n.}$$

Załóżmy teraz, że $f \geqslant 0$ i powtórzmy bardzo podobne rozumowanie dla sum

$$\sum_{j=1}^{N(n)} \Xi_j(f) \leqslant S_n(f) = \sum_{i=0}^{n-1} f(X_i) \leqslant \sum_{j=0}^{N(n)+1} \Xi_j(f). \tag{6.2.7}$$

Po lewej i po prawej stronie mamy sumy niezależnych składników $\Xi_j(f)$. Korzystamy z PWL dla niezależnych zmiennych, dzielimy (6.2.7) stronami przez $N(n)$ i przejchodzimy do granicy. Otrzymujemy

$$\frac{S_n(f)}{N(n)} \to \mathbb{E}_{x_*} \Xi(f) \text{ p.n.}$$

a więc

$$\frac{S_n(f)}{n} \to \frac{\mathbb{E}_{x_*} \Xi(f)}{m} = \frac{1}{m} \sum_x \alpha(x) f(x) = \sum_x \pi(x) f(x) \text{ p.n.}$$

Ostatnia równość wynika z Twierdzenia Kaca. Przypomnijmy, że $\alpha(x)$ jest „średnim czasem spędzonym w stanie x" podczas pojedynczej wycieczki.

Jeśli funkcja f nie jest nieujemna, to możemy zastosować rozkład $f = f^+ - f^-$ i wykorzystać już udowodniony wynik. □

Na podobnej idei oparty jest „regeneracyjny" dowód Centralnego Twierdzenia Granicznego (istnieją też zupełnie inne dowody).

6.2.8 Twierdzenie (Centralne Twierdzenie Graniczne)**.** *Jeśli X_n jest łańcuchem nieprzywiedlnym, to dla dowolnego rozkładu początkowego ν i każdej funkcji $f : \mathcal{X} \to \mathbb{R}$,*

$$\frac{1}{\sqrt{n}} \left(\sum_{i=0}^{n-1} [f(X_i) - \mathbb{E}_\pi f] \right) \longrightarrow_d \mathrm{N}\big(0, \sigma^2_{\mathrm{as}}\big), \quad (n \to \infty).$$

Ponadto, dla dowolnego rozkładu początkowego ν zachodzi wzór (4.1.5), czyli $\mathrm{Var}_\nu \sum_{i=0}^{n-1} f(X_i)/n \to \sigma^2_{\mathrm{as}}$ *przy $n \to \infty$, gdzie*

$$\sigma^2_{\mathrm{as}} = \mathrm{Var}_{x_*} \Xi(f) / \mathbb{E}_{x_*} T.$$

Szkic dowodu. Trochę więcej jest tu technicznych zawiłości niż w dowodzie PWL, wobec tego zdecydowałem się pominąć szczegóły. W istocie, przedstawię tylko bardzo pobieżnie główną ideę. Bez straty ogólności załóżmy, że $\pi^\top f = 0$. Tak jak w dowodzie PWL, sumę $S_n(f) = \sum_{i=0}^{n-1} f(X_i)$ przybliżamy sumą niezależnych składników, które odpowiadają całkowitym wycieczkom: $S_n(f) \simeq S_{T_{N(n)}}(f) = \sum_{j=1}^{N(n)} \Xi_j(f)$. Ze zwykłego CTG dla niezależnych zmiennych o jednakowym rozkładzie otrzymujemy

$$\frac{1}{\sqrt{k}} \sum_{j=1}^{k} \Xi_j(f) \to_{\mathrm{d}} \mathrm{N}(0, \mathrm{Var}_{x_*} \Xi(f)).$$

Jeśli „podstawimy" w miejsce k zmienną losową $N(n)$ i wykorzystamy fakt, że $N(n) \simeq n/m$ (PWL gwarantuje, że $N(n)/n \to 1/m$), to nie powinien dziwić następujący wniosek:

$$\frac{1}{\sqrt{n}} S_n(f) \to_{\mathrm{d}} \mathrm{N}(0, \mathrm{Var}_{x_*} \Xi(f)/m).$$

W ten sposób „udowodniliśmy" tezę. □

Zauważmy, że w Twierdzeniu 6.2.8 pojawiło się inne wyrażenie na asymptotyczną wariancję niż wzory (4.1.5) i (4.1.6) z Rozdziału 4. Te wzory są równoważne (przynajmniej dla łańcuchów nieprzywiedlnych na przestrzeni skończonej), ale pominiemy dowód tego faktu.

6.3 *Coupling*

Coupling to złączanie, zlepianie (brak dobrego odpowiednika tego terminu w języku polskim). W kontekście łańcuchów Markowa jest to metoda, która pozwala udowodnić zbieżność do rozkładu stacjonarnego i daje w wielu przypadkach dobre oszacowania szybkości zbieżności. Co więcej, idea couplingu stoi za algorytmami *perfect sampling* (losowanie dokładne z rozkładu stacjonarnego).

Coupling ma długą historię. Wolfgang Döblin w 1938 roku podał dowód Słabego Twierdzenia Ergodycznego oparty na tej idei (tak zwana „metoda dwóch cząstek Döblina"). W tym podrozdziale przytoczę dowód Döblina. Przedstawię pierwszy algorytm losowania dokładnego (CFTP), według przełomowej pracy [10] Proppa i Wilsona z 1996 roku.

Odległość pełnego wahania

Najpierw zajmiemy się określeniem odległości między rozkładami. Dla naszych celów najbardziej przydatna będzie następująca metryka. Niech ν i λ będą dwoma rozkładami prawdopodobieństwa na skończonej przestrzeni $\mathcal{X}$. Odległość *pełnego wahania* pomiędzy ν i λ określamy wzorem

$$\|\nu - \lambda\|_{\mathrm{tv}} = \max_{A \subseteq \mathcal{X}} |\nu(A) - \lambda(A)|.$$

Jak zwykle, możemy utożsamić rozkład prawdopodobieństwa na $\mathcal{X}$ z funkcją, przypisującą prawdopodobieństwa pojedynczym punktom $x \in \mathcal{X}$. Zauważmy, że

$$\|\nu - \lambda\|_{\mathrm{tv}} = \frac{1}{2} \sum_{x \in \mathcal{X}} |\nu(x) - \lambda(x)|.$$

Istotnie, ponieważ rozpatrujemy dwie miary probabilistyczne, dla których $\nu(\mathcal{X}) = \lambda(\mathcal{X}) = 1$, więc $\|\nu - \lambda\| = \nu(B) - \lambda(B)$ dla $B = \{x : \nu(x) > \lambda(x)\}$. Ale $\sum_{x \in B}(\nu(x) - \lambda(x)) = \sum_{x \in \mathcal{X} \setminus B}(\lambda(x) - \nu(x)) = \frac{1}{2} \sum_{x \in \mathcal{X}} |\nu(x) - \lambda(x)|$.

Dla zmiennej losowej $X : \Omega \to \mathcal{X}$ napis $X \sim \nu$ oznacza (jak zwykle) fakt, że X ma rozkład prawdopodobieństwa ν, czyli $\mathbb{P}(X = x) = \nu(x)$,

6.3.1 Lemat. *Jeżeli $X, Y : \Omega \to \mathcal{X}$ są dwiema zmiennymi losowymi określonymi na tej samej przestrzeni probabilistycznej i $X \sim \nu$ i $Y \sim \lambda$, to*

$$\|\nu - \lambda\|_{\mathrm{tv}} \leqslant \mathbb{P}(X \neq Y).$$

Dowód. Niech $d = \mathbb{P}(X \neq Y)$. Dla dowolnego $A \subseteq \mathcal{X}$ mamy

$$\nu(A) = \mathbb{P}(X \in A) \leqslant \mathbb{P}(Y \in A) + \mathbb{P}(X \neq Y) = \lambda(A) + d.$$

Symetrycznie, $\nu(A) \leqslant \lambda(A) + d$. Zatem $\|\nu - \lambda\|_{\mathrm{tv}} \leqslant d$. □

Lemat 6.3.1 daje się w pewnym sensie odwrócić. Co prawda, to nie będzie potrzebne w dowodzie Słabego Twierdzenia Ergodycznego, ale jest interesujące i ważne.

6.3.2 Lemat. *Jeżeli ν i λ są rozkładami prawdopodobieństwa na $\mathcal{X}$, to istnieją zmienne losowe X i Y określone na tej samej przestrzeni probabilistycznej, takie, że $X \sim \nu$ i $Y \sim \lambda$ i*

$$\|\nu - \lambda\|_{\mathrm{tv}} = \mathbb{P}(X \neq Y).$$

Dowód. Niech $\|\nu - \lambda\|_{\mathrm{tv}} = d$. Bez straty ogólności możemy przyjąć, że X i Y są zmiennymi losowymi określonymi na przestrzeni probabilistycznej $\Omega = \mathcal{X} \times \mathcal{X}$. Należy podać *łączny* rozkład zmiennych losowych X i Y, czyli miarę probabilistyczną χ na $\mathcal{X} \times \mathcal{X}$ taką, że $\sum_y \chi(x,y) = \nu(x)$, $\sum_x \chi(x,y) = \lambda(y)$ i $\sum_x \chi(x,x) = 1 - d$.

Niech

$$\chi(x,x) = \min\left(\nu(x), \lambda(x)\right) = \begin{cases} \nu(x) & \text{dla } x \in A; \\ \lambda(x) & \text{dla } x \in B, \end{cases}$$

gdzie $A = \{x : \nu(x) \leqslant \lambda(x)\}$ i $B = \{x : \nu(x) > \lambda(x)\}$.

Mamy oczywiście $d = 1 - \sum_x \chi(x,x)$ i jest jasne, że tabelka łącznego rozkładu $\chi(x,y) = \mathbb{P}(X = x, Y = y)$ musi być postaci macierzy blokowej

$$\begin{array}{r} {\scriptstyle x \in A}\left\{ \right. \\ {\scriptstyle x \in B}\left\{ \right. \\ \\ \end{array} \begin{array}{cc} D_A & 0 \\ G & D_B \\ \underbrace{}_{y \in A} & \underbrace{}_{y \in B} \end{array},$$

gdzie D_A i D_B są macierzami diagonalnymi. Pozostaje tylko odpowiednio „rozmieścić pozostałą masę prawdopodobieństwa” d w macierzy G. Możemy na przykład przyjąć, dla $x \in B$ i $y \in A$,

$$\chi(x,y) = \frac{1}{d}\left(\nu(x) - \lambda(x)\right)\left(\lambda(y) - \nu(y)\right).$$

Mamy wtedy $\sum_{y \in A} \chi(x,y) = \nu(x) - \lambda(x)$, więc $\sum_y \chi(x,y) = \nu(x)$ dla $x \in B$ i podobnie $\sum_x \chi(x,y) = \lambda(x)$ dla $y \in A$. Określony przez nas rozkład łączny χ ma więc masę $1 - d$ na przekątnej i żądane rozkłady brzegowe. □

Słabe Twierdzenie Ergodyczne dla łańcuchów Markowa (*via coupling*)

Rozważmy „podwójny" łańcuch Markowa (X_n, Y_n) na przestrzeni stanów $\mathcal{X}\times\mathcal{X}$. Przypuśćmy, że każda z dwóch „współrzędnych", oddzielnie rozpatrywana, jest łańcuchem o macierzy przejścia P. Mówiąc dokładniej, zakładamy, że

$$\begin{aligned}\mathbb{P}\left(X_{n+1}=x', Y_{n+1}=y' | X_n=x, Y_n=y, X_{n-1}, Y_{n-1}, \ldots, X_0, Y_0\right)\\ = \bar{P}\big((x,y),(x',y')\big),\end{aligned}$$

gdzie macierz przejścia $\bar{P}$ podwójnego łańcucha spełnia następujące warunki:

$$\begin{aligned}\sum_{y'} \bar{P}\big((x,y),(x',y')\big) &= P(x,x') \quad \text{dla dowolnych } (x,x',y)\\ \sum_{x'} \bar{P}\big((x,y),(x',y')\big) &= P(y,y') \quad \text{dla dowolnych } (y,y',x).\end{aligned} \tag{6.3.3}$$

Widać, że $X_0, X_1, \ldots, X_n, \ldots$ jest łańcuchem Markowa z prawdopodobieństwami przejścia P i to samo można powiedzieć o $Y_0, Y_1, \ldots, Y_n, \ldots$. Załóżmy ponadto, że od momentu, gdy oba łańcuchy się spotkają, dalej „poruszają się" już razem. Innymi słowy,

$$\bar{P}\big((x,y),(x',y')\big) = \begin{cases} P(x,x') & \text{jeśli } x'=y',\\ 0 & \text{jeśli } x'\neq y'.\end{cases} \tag{6.3.4}$$

Nazwiemy konstrukcję takiej pary *zlepianiem* łańcuchów (może lepiej pozostać przy angielskim terminie *coupling*). Aby zrozumieć *coupling*, przypomnijmy konstrukcję, która jest podstawą algorytmów generujących łańcuchy Markowa, opisaną w Rozdziale 2: porównaj równania (2.2.3) i (2.2.4). Niech $U = U_0, U_1, \ldots, U_n, \ldots$ będzie ciągiem „liczb losowych", produkowanych przez komputerowy generator, traktowanych jako *niezależne zmienne losowych o jednakowym rozkładzie*, $\mathrm{U}(0,1)$. Niech $\psi : [0,1] \to \mathcal{X}$ i $\phi : \mathcal{X}\times[0,1] \to \mathcal{X}$ będą takimi funkcjami, że $\mathbb{P}\left(\psi(U)=x\right) = \nu(x)$ dla każdego $x\in\mathcal{X}$ i

$$\mathbb{P}\left(\phi(x,U)=x'\right) = P(x,x') \quad \text{dla dowolnych } x,x'\in\mathcal{X}. \tag{6.3.5}$$

Powiemy, że funkcja ϕ jest spełniająca równanie (6.3.5) zgodna z P (lub „realizuje prawdopodobieństwa przejścia P"). Jeśli generujemy dwie „kopie" łańcucha X_n i Y_n używając tej samej funkcji ϕ i tych samych liczb losowych U_i, to widać, że spełnione będą równania (6.3.3) i (6.3.4).

Oznaczmy przez T moment spotkania się łańcuchów:

$$T = \min\{n>0 : X_n = Y_n\}. \tag{6.3.6}$$

Podstawową rolę odgrywa następujące spostrzeżenie:

$$\|\mathbb{P}(X_n\in\cdot) - \mathbb{P}(Y_n\in\cdot)\|_{\mathrm{tv}} \leqslant \mathbb{P}(X_n\neq Y_n) = \mathbb{P}(T>n).$$

Jeśli teraz łańcuch Y_n „wystartuje" z rozkładu stacjonarnego, czyli $Y_0 \sim \pi$ to $Y_n \sim \pi$ dla każdego n i otrzymujemy

$$\|\mathbb{P}(X_n \in \cdot) - \pi(\cdot)\|_{\text{tv}} \leqslant \mathbb{P}(T > n). \tag{6.3.7}$$

Aby udowodnić zbieżność $\mathbb{P}(X_n \in \cdot) \to \pi(\cdot)$ wystarczy skonstruować parę łańcuchów, które się spotkają z prawdopodobieństwem 1: $\mathbb{P}(T < \infty) = 1$. Możemy teraz udowodnić (4.1.2), przynajmniej dla łańcuchów na skończonej przestrzeni stanów.

6.3.8 Twierdzenie (Słabe Twierdzenie Ergodyczne)**.** *Jeśli łańcuch Markowa na skończonej przestrzeni stanów jest nieprzywiedlny i nieokresowy, to*

$$\|\mathbb{P}(X_n \in \cdot) - \pi(\cdot)\|_{\text{tv}} \to 0.$$

Dowód. Rozważmy parę łańcuchów (X_n, Y_n), które poruszają się *niezależnie* aż do momentu spotkania.

$$\bar{P}\big((x,y),(x',y')\big) = \begin{cases} P(x,x')P(y,y') & \text{jeśli } x \neq y, \\ P(x,x') & \text{jeśli } x = y \text{ i } x' = y', \\ 0 & \text{jeśli } x = y \text{ i } x' \neq y'. \end{cases} \tag{6.3.9}$$

Żeby pokazać, że $\mathbb{P}(T < \infty) = 1$ wystarczy zauważyć, że do przed momentem spotkania, łańcuch podwójny ewoluuje zgodnie z prawdopodobieństwami przejścia

$$\tilde{P}\big((x,y),(x',y')\big) = P(x,x')P(y,y').$$

Łańcuch odpowiadający $\tilde{P}$ jest nieprzywiedlny. Istotnie, możemy znaleźć takie n_0, że dla $n \geqslant n_0$ wszystkie elementy macierzy P^n są niezerowe. Stąd $\tilde{P}^n\big((x,y),(x',y')\big) = P^n(x,x')P^n(y,y') > 0$ dla dowolnych x, x', y, y'. Wystarczy teraz powołać się na Wniosek 6.2.3: podwójny łańcuch z prawdopodobieństwem 1 prędzej czy później dojdzie do każdego punktu przestrzeni $\mathcal{X} \times \mathcal{X}$, a zatem musi dojść do „przekątnej" $\{(x,x) : x \in \mathcal{X}\}$. □

Uwaga. W dowodzie Twierdzenia 6.3.8 wykorzystaliśmy w istotny sposób nieokresowość macierzy przejścia P (dla pojedynczego łańcucha), choć to mogło nie być wyraźnie widoczne. Jeśli P jest nieprzywiedlna ale okresowa, wtedy $\tilde{P}$ jest nieprzywiedlna. Na przykład, niech $\mathcal{X} = \{0,1\}$ i

$$P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Wtedy, oczywiście, $\tilde{P}^n((0,0),(0,1)) = 0$ bo $P^n(0,0) = 0$ dla nieparzystych n zaś $P^n(0,1) = 0$ dla parzystych n.

Ten sam trywialny przykład pokazuje, że dla łańcuchów okresowych teza Słabego Twierdzenia Ergodycznego nie jest prawdziwa.

W istocie, przytoczony przez nas dowód Twierdzenia 6.3.8 daje nieco więcej, niż tylko zbieżność rozkładów. Z Wniosku 6.2.3 wynika, że

$$\|\mathbb{P}(X_n \in \cdot) - \pi(\cdot)\|_{\mathrm{tv}} \leqslant c\gamma^n$$

dla pewnych stałych $c < \infty$ i $\gamma < 1$. Takie ogólnikowe stwierdzenie nie jest wystarczające. Dla niektórych łańcuchów używanych w algorytmach MCMC znane są jawne oszacowania, z konkretnymi stałymi. Przykład ponożej pokazuje, że użycie niezależnych kopii łańcucha w dowodzie Twierdzenia 6.3.8, wzór (6.3.9) jest konstrukcją dalece nieoptymalną. Można skonstruować pary łańcuchów (X_n, Y_n) znacznie szybciej „zmierzające do spotkania".

6.3.10 Przykład (Błądzenie po kostce)**.** Niech $\mathcal{X} = \{0,1\}^n$ i $\pi = \mathrm{U}(\mathcal{X})$, czyli $\pi(x) = 1/2^n$ dla każdego x. Rozważmy łańcuch Markowa X_n, którego krok polega na wylosowaniu jednej, losowo wybranej współrzędnej z rozkładu $(1/2, 1/2)$ na zbiorze $\{0,1\}$ i pozostawieniu pozostałych współrzędnych bez zmian. Formalnie,

$$P(x, x') = \frac{1}{2d} \sum_{i=1}^{d} \mathbb{1}(x_{-i} = x'_{-i}).$$

Jest to zatem „losowe błądzenie" wzdłuż krawędzi n-wymiarowej kostki lub inaczej próbnik Gibbsa. Rzecz jasna, dokładne genrowanie z rozkładu jednostajnego na kostce jest łatwe i nie ptrzebujemy do tego łańcuchów Markowa, ale nie o to teraz chodzi. Chcemy zilustrować jak metoda sprzęgania pozwala oszacować szybkość zbieżności łańcucha na możliwie prostym przykładzie. Skonstruujmy parę łańcuchów sprzężonych w taki sposób: wybieramy współrzędną i oraz losujemy jej nową wartość z rozkładu $(1/2, 1/2)$ po czym zmieniamy w ten sam sposób obie kopie. Formalnie,

$$\bar{P}((x,y),(y,y')) = \frac{1}{2d} \sum_{i=1}^{d} \mathbb{1}(x_{-i} = x'_{-i}, y_{-i} = y'_{-i}, x'_i = y'_i).$$

Jest jasne, że to jest poprawny *coupling*, to znaczy spełnione są równania (6.3.3) i (6.3.4). Spotkanie obu kopii nastąpi *najpóźniej* w momencie gdy każda ze współrzędnych zostanie wybrana przynajmniej raz. Zatem

$$\mathbb{P}(T > n) \leqslant \left(1 - \frac{1}{d}\right)^n.$$

Nie trudno wyobrazić sobie, że dla *niezależnego* couplingu określonego wzorem (6.3.9), czas oczekiwania na spotkanie obu kopii jest na ogół dużo, dużo dłuższy. △

6.4 Symulacja doskonała dla łańcuchów Markowa

Pomysłowy algorytm, wynaleziony przez Proppa i Wilsona w 1996 roku, pozwala generować próbki *dokładnie* z rozkładu stacjonarnego łańcucha Markowa. To zadziwiające osiągnięcie,

ponieważ metody MCMC (markowowskie algorytmy Monte Carlo) zostały stworzone z myślą o trudnych rozkładach prawdopodobieństwa, z których nie potrafimy losować dokładnie! Przełomowy artykuł Proppa i Wilsona [10] dał początek całej obszernej dziedzinie badań nad algorytmami typu „*perfect sampling*" (próbkowanie dokładne albo doskonałe).

Niech P będzie macierzą *nieprzywiedlnego i nieokresowego* łańcucha Markowa na *skończonej* przestrzeni $\mathcal{X}$. Niech π będzie rozkładem stacjonarnym dla P (wiadomo, że taki rozkład istnieje i jest jednoznacznie wyznaczony). Niech $\phi : \mathcal{X} \times [0,1] \to \mathcal{X}$ będzie funkcją spełniającą równanie (6.3.5), czyli „realizującą prawdopodobieństwa przejścia P. Krok łańcucha Markowa realizujemy używając (powiedzmy) ciągu $U_i \sim_{\text{i.i.d.}} \mathrm{U}(0,1)$ w następujący sposób:

$$X_{n+1} = \phi\left(X_n, U_n\right). \tag{6.4.1}$$

Otrzymujemy łańcuch Markowa o macierzy przejścia P. Powiedzmy, że stan początkowy wybieramy deterministycznie, $X_0 = x$. Wiadomo, że rozkład prawdopodobieństwa zmiennej losowej X_n zmierza, przy $n \to \infty$ do rozkładu stacjonarnego π, niezależnie od wyboru $x \in \mathcal{X}$.

CFTP

Przedstawię algorytm z pracy [10], nazwany przez autorów *Coupling From The Past*, CFTP.

Pomysł jest taki: wyobrażamy sobie, że łańcuch startuje w dalekiej przeszłości z różnych punktów przestrzeni. Jeśli dwie trajektorie łańcucha spotkają się w tym samym czasie w tym samym stanie, to dalej poruszają się wspólnie (na tym polega *coupling*). Jeśli okaże się, że *wszystkie* trajektorie, startujące ze wszystkich stanów się spotkały i złączyły (nastąpiła „koalescencja"), to ta wspólna trajektoria jest trajektorią łańcucha stacjonarnego.

Dla $n = 1, 2, \ldots$, zdefiniujmy funkcję $\Phi_{-n} : \mathcal{X} \times [0,1]^n \to \mathcal{X}$ w następujący sposób:

$$\Phi_{-n}(\cdot, U_{-1}, \ldots, U_{-n}) = \phi(\cdot, U_{-1}) \circ \cdots \circ \phi(\cdot, U_{-n}). \tag{6.4.2}$$

Innymi słowy, $\Phi_{-n}(x) = \phi(\cdots \phi(\phi(x, U_{-n}), U_{-n+1}) \cdots, U_{-1})$. Patrzymy na Φ_{-n} jak na „losową funkcję" $\mathcal{X} \to \mathcal{X}$.

Poniższe założenie odgrywa kluczową rolę.

6.4.3 Założenie. *Z prawdopodobieństwem 1 istnieje takie n, że Φ_{-n} jest funkcją stałą.*

Jeśli Φ_{-n} jest funkcją stałą, to Φ_{-m} jest również funkcją stałą dla $-m < -n$. Założenie 6.4.3 zapewnia, że prawie na pewno istnieje granica

$$\Phi_{-\infty}(\cdot, U_{-1}, \ldots, U_{-n}, \cdots) = \lim_{n\to\infty} \Phi_{-n}(\cdot, U_{-1}, \ldots, U_{-n}). \tag{6.4.4}$$

Możemy zdefiniować zmienne losowe

$$X_0 = \Phi_{-\infty}(\cdot, U_{-1}, U_{-2}, \cdots),$$
$$X_{-1} = \Phi_{-\infty}(\cdot, U_{-2}, U_{-3}, \cdots). \tag{6.4.5}$$

6.4.6 Twierdzenie. *Jeśli spełnione jest Założenie 6.4.3 to $\pi(x) = \mathbb{P}(X_0 = x)$ jest rozkładem stacjonarnym dla macierzy przejścia P.*

Dowód. Ponieważ

$$\Phi_{-\infty}(\cdot, U_{-1}, U_{-2}, \cdots) = \phi(\cdot, U_{-1}) \circ \Phi_{-\infty}(\cdot, U_{-2}, U_{-3}, \cdots),$$

więc $X_0 = \phi(X_{-1}, U_{-1})$, przy tym U_{-1} jest niezależne od X_{-1}. Stąd mamy

$$\mathbb{P}(X_0 = x' | X_{-1} = x) = P(x, x'),$$

na mocy wzoru (2.2.3). Z drugiej strony, $X_{-1} =_{\mathrm{d}} X_0$, zatem $\pi(x') = \sum_x \pi(x) P(x, x')$. □

Twierdzenie 6.4.6 w istocie opisuje algorytm generowania $X_0 \sim \pi$. To jest tak zwana symulacja dokładna lub doskonała (*perfect simulation*).

Algorytm CFTP (*Coupling From The Past*)

$n := 0$; $\Phi_0 := \mathrm{Id}$

`repeat`

  `Gen` U_{-n-1}

  $\Phi_{-n-1} := \Phi_{-n} \circ \phi(\cdot, U_{-n-1})$

  $n := n + 1$

`until` Φ_{-n} `jest funkcją stałą`

`return` $X_0 := \Phi_{-n}(x)$ `{ ta wartość jest taka sama dla dowolnie wybranego` $x \in \mathcal{X}$ `}`

Oczywiście, Id oznacza odworowanie tożsamościowe.

Uwaga. Występujące w naszych wzorach zmienne $U_{-1}, \ldots, U_{-n}, \cdots \sim_{\text{i.i.d.}} \mathrm{U}(0,1)$ odgrywają rolę „źródła losowości". Istotna jest tylko niezależność tych zmiennych i wzór (6.3.5). Zamiast „liczby losowej", czyli zmiennej losowej $U \sim \mathrm{U}(0,1)$ moglibyśmy użyć dowolnej zmiennej losowej Z w dowolnej przestrzeni $\mathcal{Z}$, byleby spełniona była relacja $\mathbb{P}(\phi(x, Z) = x') = P(x, x')$. Do wygenerowania takiej zmiennej Z można użyć dowolnie wielu „liczb losowych".

W algorytmie CFTP niezwykle ważne jest to, że przy „cofnięciu się w czasie", czyli przejściu od $\Phi_{-n}(\cdot, U_{-1}, \ldots, U_{-n})$ do $\Phi_{-n-1}(\cdot, U_{-1}, \ldots, U_{-n}, U_{-n-1}) = \Phi_{-n}(\cdot, U_{-1}, \ldots, U_{-n}) \circ \phi(\cdot, U_{-n-1})$, trzeba używać tych samych „liczb losowych" $U_{-1}, \ldots, U_{-n}$, a nie generować ich od nowa. Kontrprzykład, zaczerpnięty z oryginalnej pracy Proppa i Wilsona, podany jest W Ćwiczeniu 6.1.

Przykład poniżej odpowiada (w moim przekonaniu) na pytanie: „dlaczego konstruujemy *couping* wstecz, a nie w przód?"

6.4.7 Przykład. Rozpatrzmy przestrzeń stanów $\{0,1\}$ i macierz (nieprzywiedlną i nieokresową)

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ 1 & 0 \end{pmatrix}.$$

Łatwo zauważyć, że dwie kopie łańcucha starujące z 0 i z 1 nie mogą się spotkać w 1. Analiza działania CFTP w tym przykładzie jest naszkicowana w Zadaniu 6.1. △

Przyjrzyjmy się teraz Założeniu 6.4.3.

6.4.8 Stwierdzenie. *Jeżeli istnieje n takie, że $\mathbb{P}(\Phi_{-n}$ jest funkcją stałą $) \geqslant \alpha > 0$ to Założenie 6.4.3 jest spełnione.*

Dowód. Jak już zauważyliśmy, złożenie dwóch funkcji jest funkcją stałą jeśli przynajmniej jedna ze składanych funkcji jest stała. Ponieważ

$$\Phi_{-n-k}(\cdot, U_{-1}, \dots, U_{-n}, U_{-n-1}, \dots, U_{-n-k})) = \Phi_{-n}(\cdot, U_{-1}, \dots, U_{-n}) \circ \Phi_{-k}(\cdot, U_{-n-1}, \dots, U_{-n-k})$$

i obie funkcje po prawej stronie tego wzoru są niezależne, to

$$\mathbb{P}(\Phi_{-n-k} \text{ nie jest funkcją stałą }) \leqslant \mathbb{P}(\Phi_{-n} \text{ nie jest funkcją stałą })\mathbb{P}(\Phi_{-k} \text{ nie jest funkcją stałą }).$$

Stąd przez indukcję wnioskujemy, że

$$\mathbb{P}(\Phi_{-nm} \text{ jest funkcją stałą }) \geqslant 1 - (1-\alpha)^m \to_{m\to\infty} 1.$$

Ciąg zdarzeń losowych $C_m = \{\Phi_{-nm}$ jest funkcją stałą$\}$ jest niemalejący, zatem $\mathbb{P}(\bigcup_{m=1}^{\infty} C_m) = 1$. □

6.4.9 Stwierdzenie. *Jeśli macierz przejścia P jest nieprzywiedlna i nieokresowa, to istnieje funkcja ϕ spełniająca równanie (2.2.3) taka, że Założenie 6.4.3 jest spełnione.*

Zanim przejdziemy do dowodu, zwróćmy uwagę na pewną subtelność w sformułowaniu tego stwierdzenia. Nie jest prawdą, że dowolna funkcja „realizująca" prawdopodobieństwa przejścia P spełnia założenie 6.4.3. Kontrprzykład jest niezwykle prosty:

6.4.10 Przykład. Niech $\mathcal{X} = \{0,1\}$,

$$P = \begin{pmatrix} 1-\alpha & \alpha \\ \alpha & 1-\alpha \end{pmatrix}.$$

Jeśli przyjmiemy

$$\phi(x,u) = \begin{cases} x & \text{jeśli } u > \alpha; \\ 1-x & \text{jeśli } u \leqslant \alpha, \end{cases}$$

to ta funkcja jest zgodna z P, ale dwie kopie łańcucha startujące z 0 i z 1, używające funkcji ϕ oraz tych samych liczb losowych *nigdy się nie spotkają*. △

Dowód Stwierdzenia 6.4.9. Opiszę jedną z wielu możliwych konstrukcji: powiedzmy, że jest to „coupling niezależny". Pojedynczą „liczbę losową" U można „rozmnożyć" na $|\mathcal{X}|$ niezależnych zmiennych losowych $\left(U^{(x)}, x \in \mathcal{X}\right)$. Możemy zdefiniować funkcję ϕ w następujący sposób:

$$\phi(x, U) = \phi_0(x, U^{(x)}),$$

gdzie funkcja ϕ_0 jest zgodna z P, czyli spełnia (2.2.3). Chwila zastanowienia wystarczy, żeby sobie uświadomić, że jeśli trajektorie łańcucha startują z różnych punktów, używają funkcji ϕ i tych samych „rozmnożonych" liczb losowych, $U_n \equiv (U_n^{(x)}, x \in \mathcal{X})$, to ewoluują niezależnie aż do momentu spotkania, a później się sklejają i wędrują razem. Skorzystamy ze Stwierdzenia 6.4.8. Pokażemy, że istnieje takie n, że z niezerowym prawdopodobieństwem wszystkie trajektorie złączą się po co najwyżej n krokach. Wybierzmy dowolny, ustalony stan $x_* \in \mathcal{X}$. Ponieważ P jest nieprzywiedlna i nieokresowa, to istnieje n takie, że $p = P^n(x, x_*) > 0$ dla dowolnego stanu x. Wszystkie trajektorie się spotkają i zlepią (w najgorszym przypadku w punkcie x_* po n krokach) z prawdopodobieństwem co najmniej $\alpha = p^{|\mathcal{X}|}$.

Zauważmy, że określone wyżej α jest zazwyczaj astronomicznie małe. Nasze oszacowanie prawdopodobieństwa „koagulacji" jest co prawda konserwatywne (zaniżone), ale ważniejsze jest to, że sama strategia „niezależnego couplingu" jest bardzo daleka od optymalnej. Można zdefiniować funkcję ϕ inaczej, tak aby byłą zgodna z P ale jednocześnie „sprzyjała łączeniu się trajektorii". □

Symulacja doskonała dla łańcuchów monotonicznych

Sprawdzanie, czy Φ_{-n} jest funkcją stałą może być, w ogólnej sytuacji, dla dużej przestrzeni stanów, zadaniem niewdzięcznym. Sprawa się bardzo upraszcza jeśli przestrzeń jest zbiorem częściowo uporządkowanym i łańcuch jest „monotoniczny" w odpowiednim sensie.

Niech $\preceq$ będzie relacją częściowego porządku w zbiorze $\mathcal{X}$. Pnadto załóżmy, że istnieje w tym zbiorze element najmniejszy, oznaczany $\hat{0}$ i element największy, oznaczany $\hat{1}$. O funkcji ϕ, która realizuje krok łańcucha Markowa założymy, że

$$x \preceq x' \text{ implikuje } \phi(x, u) \preceq \phi(x', u) \text{ dla każdego } u. \tag{6.4.11}$$

Powyższy warunek precyzuje, w jakim sensie rozumiemy monotoniczność łańcucha. (Zauważmy, że jest to warunek sformułowany w terminach funkcji ϕ, a nie macierzy P.)

Jeśli spełnione są powyższe założenia, to zamiast kontrolowania wszystkich trajektorii, startujących z różnych punktów x, wystarczy sprawdzić, czy zlepiły się 2 trajektorie: ta startująca z $\hat{0}$ i ta startująca z $\hat{1}$. Jeśli $\Phi_{-n}(\hat{0}) = \Phi_{-n}(\hat{1})$ to wiemy, że *wszystkie* trajektorie się zlepiły. Algorytm CFTP można implementować efektywnie.

Efektywna wersja CFTP uwzględnia następujące spostrzeżenie.

Nie jest rozsądną strategią cofanie się o jeden krok wstecz, aby dojść do zmiennej losowej $-C$, gdzie $C = \min\{n : \Phi_{-n}(\hat{0}) = \Phi_{-n}(\hat{1})\}$. Lepiej cofać się w postępie geometrycznym, z $-n$ do $-2n$. W ten sposób oszacujemy C z góry, przy tym przeszacowanie nie będzie nigdy większe niż dwukrotne.

Algorytm CFTP dla łańcucha monotonicznego

$n := 1$;
`Gen` U_{-1}
`repeat`
  $\underline{X} = \hat{0}$; $\overline{X} = \hat{1}$
  { Generujemy $\overline{X}_{-n+1}, \ldots, \overline{X}_0$ *}*
  `for` $i := -n$ `to` -1 `do`
    `begin`
    $\underline{X} := \phi(\underline{X}, U_{-i})$; $\overline{X} := \phi(\overline{X}, U_{-i})$
    `end`
  `Gen` $U_{-n-1}, \ldots, U_{-2n}$
  $n := 2n$;
`until` $\underline{X} = \overline{X}$
`return` $\overline{X}$

Próbnik Gibbsa dla modelu Isinga (modelu auto-logistycznego) jest świetnym przykładem zastosowania CFTP. Przypomnijmy (Przykład 5.2.1): $\mathcal{X} = \{0,1\}^{\mathcal{S}}$, gdzie zbiór „miejsc" $\mathcal{S}$ jest wyposażony w relację sąsiedztwa ($s \sim t$). Rozważymy funkcję energii

$$H(x) = -\alpha_0 \sum_t x_t - \alpha_1 \sum_{s \sim t} x_s x_t,$$

i rozkład Gibbsa (z odwrotnością temperatury $\beta = 1$)

$$\pi(x) \propto \exp[-H(x)].$$

Istotny jest znak współczynnika interakcji we wzorze na energię. Zakładamy, że $\alpha_1 > 0$. Rozważamy model „przyciągający": preferowane są konfiguracje, w których jedynki sąsiadują z jedynkami (model „ferromagnetyczny" w interpretacji Isinga).

Próbnik Gibbsa generuje

$$\pi(x_t = 1 \mid x_{-t}) = \frac{\exp\left(\alpha_0 + \alpha_1 \sum_{s \sim t} x_t\right)}{1 + \exp\left(\alpha_0 + \alpha_1 \sum_{s \sim t} x_s\right)},$$

W przestrzeni konfiguracji $\mathcal{X}$ mamy naturalną relację częściowego porządku: $x \preceq x' \Leftrightarrow \forall_t (x_t \leqslant x'_t)$. Jeśli losowanie z pełnego rozkładu warunkowego $\pi(x_t = 1 \mid x_{-t})$ realizujemy w naturalny sposób,

to znaczy

$$x_t := \mathbb{1}\left(u < \frac{\exp\left(\alpha_0 + \alpha_1 \sum_{s\sim t} x_t\right)}{1 + \exp\left(\alpha_0 + \alpha_1 \sum_{s\sim t} x_s\right)}\right),$$

to spełniony jest warunek monotoniczności (6.4.11) i możemy zastosować CFTP w efektywny sposób. Oczywiście, 1 duży krok systematycznego PG składa się z $|\mathcal{S}|$ małych kroków i wymaga tyleż „liczb losowych".

6.5 Zadania i uzupełnienia

6.1 Zadanie. Niech $\mathcal{X} = \{0, 1\}$. Funkcję ϕ określamy wzorem

$$\phi(0, U) = \begin{cases} 0 & \text{jeśli } U < \alpha \text{ ('Orzeł')} ; \\ 1 & \text{jeśli } U \geqslant \alpha \text{ ('Reszka')} . \end{cases}$$
$$\phi(1, U) = 0.$$

- Prześledzić działanie algorytmu CFTP: dla jakich ciągów „rzutów monetą" $\Phi_{-\infty} = 0$, a dla jakich $\Phi_{-\infty} = 1$. Obliczyć bezpośrednio, nie odwołując się do Twierdzenia 6.4.6, $\mathbb{P}(X_0 = x)$, dla $x = 0, 1$.
- Porównać z rozkładem stacjonarnym w Przykładzie 6.4.7.

6.1 Ćwiczenie. Niech $\mathcal{X} = \{0, 1, 2\}$. Funkcję ϕ określamy wzorem

$$\phi(x, U) = \begin{cases} \max(x - 1, 0) & \text{jeśli } U < 1/2; \\ \min(x + 1, 2) & \text{jeśli } U \geqslant 1/2. \end{cases}$$

- Napisać macierz przejścia P i znaleźć rozkład stacjonarny π.
- Dlaczego spełnione jest Założenie 6.4.3?
- Zakodować CFTP i skontrolować, że na wyjściu mamy $X_0 \sim \pi$.
- Uruchomić niepoprawną wersję CFTP (poniżej) i zbadać rozkład X_0.

Niepoprawna wersja CFTP

$n := 0$; $\Phi_0 := \mathrm{Id}$

`repeat`

  `Gen` $U_{-1}, \ldots, U_{-n}, U_{-n-1}$

  $\Phi_{-n-1} = \phi(\cdot, U_{-1}) \circ \cdots \circ \phi(\cdot, U_{-n-1})$

  $n := n + 1$

`until` Φ_{-n} `jest funkcją stałą`

`return` $X_0 := \Phi_{-n}(x)$ *{ ta wartość jest taka sama dla dowolnie wybranego* $x \in \mathcal{X}$ *}*

6.2 Ćwiczenie. Zakodować CFTP dla modelu Isinga.

- Uruchomić program na przykładzie kraty 2×2 tak, jak w Ćwiczeniu 5.2. Skontrolować poprawność.
- Uruchomić program w sytuacji opisanej w Ćwiczeniu 5.1. Porównać wyniki symulacji „tradycyjnej" i „doskonałej".

Uwaga. Nie należy przechowywać coraz dłuższych ciągów „liczb losowych" U_{-i} tym bardziej, że faktycznie jeden krok łańcucha wymaga użycia wielu „liczb losowych". Zamiast tego lepiej zapamiętywać „ziarno" generatora „liczb losowych".

Rozdział 7

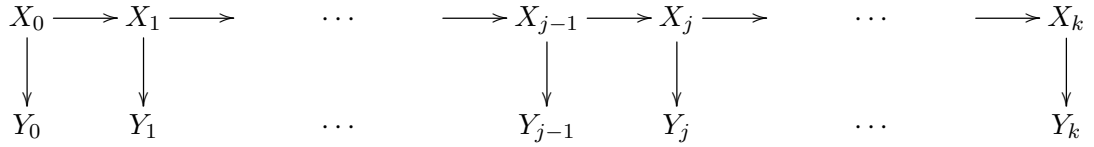
Sekwencyjne Monte Carlo

7.1 Ukryty model Markowa

Rozważamy parę procesów stochastycznych z czasem dyskretnym: $(X_{0:k}, Y_{0:k})$.

- $X = X_{0:k} = (X_0, X_1, \ldots, X_k)$ jest łańcuchem Markowa (nieobserwowalnym).
- $Y = Y_{0:k} = (Y_0, Y_1, \ldots, Y_k)$ jest procesem obserwacji (interpretujemy $Y_j = y_j$ jako informację o zmiennej X_j, z losowym błędem).
- Łączny rozkład prawdopodobieństwa: $p(x, y) = p(x)p(y|x)$, gdzie $x = x_{0:k}$, $y = y_{0:k}$.
- Interesuje nas rozkład a posteriori $p(x|y)$.

Graficzne przedstawienie struktury zależności zmiennych w naszym modelu jest następujące:



Uwaga. Używamy notacji zwięzłej i sugestywnej, ale niejednoznacznej. Symbol p jest ogólnym oznaczeniem prawdopodobieństwa lub gęstości prawdopodobieństwa i w zależności od kontekstu oznacza różne funkcje. Dla uniknięcia nieporozumień, poniżej wprowadzimy jawne oznaczenia dla prawdopodobieństw przejścia, rozkładu początkowego i funkcji wiarygodności.

- Łańcuch Markowa $X_0, X_1, \ldots, X_k$ ma prawdopodobieństwa przejścia $T(x_{j-1}, x_j) = p(x_j|x_{j-1}) = p(x_j|x_{0:j-1})$. Rozkład początkowy oznaczymy przez $\nu(x_0) = p(x_0)$.
- Zmienna losowa Y_j zależy tylko od X_j. Funkcję wiarygodności oznaczymy przez $w(x_j, y_j) = p(y_j|x_j)$.

W tej notacji, łańcuch $X_{0:k}$ ma rozkład prawdopodobieństwa

$$p(x_{0:k}) = \nu(x_0) \prod_{j=1}^{k} T(x_{j-1}, x_j). \tag{7.1.1}$$

Zgodnie z terminologią statystyki bayesowskiej jest to rozkład *a priori* (przed zaobserwowaniem danych).

Łączny rozkład prawdopodobieństwa zmiennych $X_{0:k}, Y_{0:k}$ jest dany wzorem

$$p(x_{0:k}, y_{0:k}) = \nu(x_0) w(x_0, y_0) \prod_{j=1}^{k} T(x_{j-1}, x_j) w(x_j, y_j).$$

Uwaga. W tym wzorze lewa strona oznacza gęstość na przestrzeni $\mathcal{X}^{k+1} \times \mathcal{Y}^{k+1}$, wyposażonej w miarę $\mathrm{d}x_{0:k}\mathrm{d}y_{0:k} = \prod_j \mathrm{d}x_j \prod_j \mathrm{d}y_j$. Na ogół jest to po prostu miara Lebesgue'a (gdy $\mathcal{X} \subseteq \mathcal{R}^d$ i $\mathcal{Y} \subseteq \mathcal{R}^g$) lub miara licząca (gdy $\mathcal{X}$ i $\mathcal{Y}$ są skończone). Oczywiście, $\nu(\cdot)$ i $T(x_{j-1}, \cdot)$ są gęstościami na $(\mathcal{X}, \mathrm{d}x_j)$, zaś $w(x_j, \cdot)$ jest gęstością na $(\mathcal{Y}, \mathrm{d}y_j)$.

Zakładamy, że funkcje ν, T i w są znane (nie zależą od nieznanych parametrów). Interesuje nas rozkład *a posteriori*, który oznaczymy

$$\pi(x_{0:k}) = p(x_{0:k}|y_{0:k}).$$

Ponieważ obserwacje y_j są, jak zwykle w statystyce bayesowskiej, traktowane jako stałe, w naszej notacji pomijamy zależność od $y_{0:k}$.

Ze wzoru Bayesa mamy

$$\pi(x_{0:k}) = \frac{1}{z} \nu(x_0) w(x_0, y_0) \prod_{t=1}^{k} T(x_{t-1}, x_t) w(x_t, y_t), \tag{7.1.2}$$

gdzie stała normująca z jest rozkładem brzegowym obserwacji, $p(y_{0:k})$. Jest to bardzo nieprzyjemna (zazwyczaj niemożliwa do obliczenia) całka

$$z = p(y_{0:k}) = \int_{\mathcal{X}^{k+1}} \nu(x_0) w(x_0, y_0) \prod_{t=1}^{k} T(x_{t-1}, x_t) w(x_t, y_t) \mathrm{d}x_{0:k}.$$

Sekwencyjne algorytmy Monte Carlo (SMC) pozwalają przybliżać (w odpowiednim sensie) docelowy rozkład *a posteriori* π przy pomocy rozkładów empirycznych symulowanych zmiennych losowowych. „Przy okazji" otrzymuje się przybliżenie (estymator Monte Carlo $\hat{Z}$) stałej z.

7.2 Algorytmy SIS i PF

Rozkładem docelowym jest rozkład *a posteriori* $\pi(x_{0:k}) = p(x_{0:k}|y_{0:k})$ na przestrzeni $\mathcal{X}^{k+1}$, czyli na przestrzeni trajektorii ukrytego łańcucha $X_{0:k}$, wzór (7.1.2).

Algorytm SIS, sekwencyjne losowanie istotne jest to po prostu algorytm losowania istotnego (IS2), w którym losowanie i ważenie wykonuje się sekwencyjnie (krok po kroku). Rozkładem instrumentalnym jest rozkład *a priori* dany wzorem (7.1.1). Losuje się n trajektorii długości $k+1$. Oznaczymy te trajektorie $\xi^i_{0:k}$, dla $i = 1, \ldots, n$. Będziemy mówili obrazowo, że ξ^i_t jest „położeniem i-tej cząstki w chwili t". Trajektorii $\xi^i_{0:k}$ przypisujemy wagę $w(\xi^i_{0:k}, y_{0:k})$. Algorytm SIS kolejno generuje punkty ξ^i_t i oblicza wagi $w(\xi^i_{0:t}, y_{0:t})$, dla $t = 0, 1, \ldots, k$.

Algorytm PF (filtr cząsteczkowy, Particle Filter) różni się od algorytmu SIS wprowadzeniem kroku repróbkowania (*resampling*). W każdej chwili t „redukujemy wagi do 1" przez wylosowanie (ze zwracaniem) n punktów ze zbioru n-elementowego $\{\xi^1_{t-1}, \ldots, \xi^n_{t-1}\}$, z prawdopodobieństwami proporcjonalnymi do wag: $\{w(\xi^1_{t-1}, y_{t-1}), \ldots, w(\xi^n_{t-1}, y_{t-1})\}$. „Populacja" cząsteczek ewoluuje zgodnie z zasadą „doboru naturalnego": cząsteczki z wyższymi wagami mają więcej potomstwa, rozmnażają się, a cząsteczki z małymi wagami wymierają bezpotomnie.

Algorytm SIS (*Sequential Importance Sampling*)

Input: $\nu(\cdot)$, $T(\cdot,\cdot)$, $w(\cdot, y_t)_{t=0,1,\ldots,k}$.
{ Inicjalizacja }
  for $i = 1, \ldots, n$
  Gen $\xi^i_0 \sim \nu(\cdot)$;
  Oblicz $W^i_0 = w(\xi^i_0, y_0)$;
  endfor
{ Główna pętla }
for $t = 1, \ldots, k$
  for $i = 1, \ldots, n$
  { Propagacja }
  Draw $\xi^i_t \sim T(\xi^i_{t-1}, \cdot)$; $\xi^i_{0:t} := (\xi^i_{0:t-1}, \xi^i_t)$;
  { Ważenie }
  Oblicz $W^i_t = w(\xi^i_t, y_t)$; $W^i_{0:t} = W^i_{0:t-1} W^i_t$;
  endfor
 endfor
 Output: $(\xi^i_{0:k}, W^i_{0:k})_{i=1,\ldots,n}$, $\hat{Z} = \bar{W}_{0:k} = \frac{1}{n}\sum_{i=1}^n W^i_{0:k}$.

W poniższym pseudo-kodzie, cząsteczka ξ^i_t w momencie t wybiera (losuje) rodzica $\xi^{A^i_t}_{t-1}$: w ten sposób realizuje się losowanie ze zwracaniem.

Algorytm PF (*Particle Filter*)

Input: $\nu(\cdot)$, $T(\cdot,\cdot)$, $w(\cdot, y_t)_{t=0,1,\ldots,k}$.

{ Inicjalizacja }

for $i = 1, \ldots, n$

  Gen $\xi_0^i \sim \nu(\cdot)$;

  Oblicz wagę $W_0^i = w(\xi_0^i, y_0)$;

endfor

{ Główna pętla }

for $t = 1, \ldots, k$

  for $i = 1, \ldots, n$

  { Repróbkowanie }

  Wylosuj A_t^i z prawdopodobieństwem $\mathbb{P}(A_t^i = j) \propto W_{t-1}^j \ \ (j = 1, \ldots, n)$;

  { Propagacja }

  Gen $\xi_t^i \sim T(\xi_{t-1}^{A_t^i}, \cdot\,)$; $\xi_{0:t}^i := (\xi_{0:t-1}^{A_t^i}, \xi_t^i)$;

  { Ważenie }

  Oblicz $W_t^i = w(\xi_t^i, y_t)$;

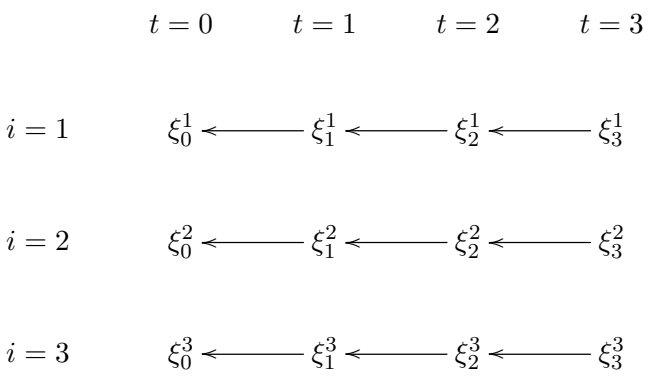
  endfor

endfor

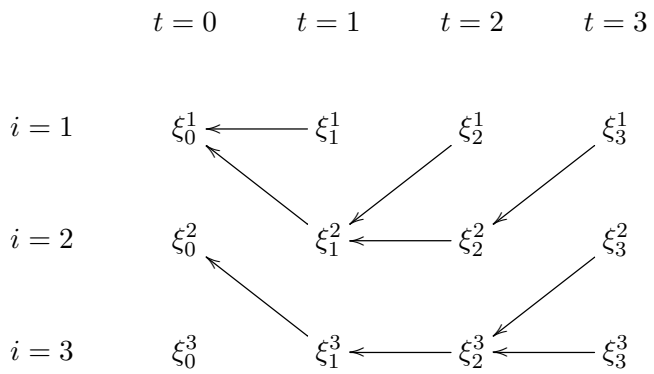
Output: $(\xi_{0:k}^i, W_k^i)_{i=1,\ldots,n}$, $\hat{Z} = \prod_{t=0}^k \bar{W}_t$.

Oczywiście, $\bar{W}_t$ oznacza $\frac{1}{n}\sum_{i=1}^n W_t^i$.

Zależności pomiędzy zmiennymi dla algorytmów SIS i PF obrazują następujące dwa schematy. Strzałki wskazują *rodziców*. Dla algorytmu SIS mówimy, że rodzicem ξ_t^i jest ξ_{t-1}^i co wyraża fakt, że $\xi_t^i \sim T(\xi_{t-1}^i, \cdot)$.



Przykładową strukturę zależności dla algorytmu PF przedstawia następny diagram.



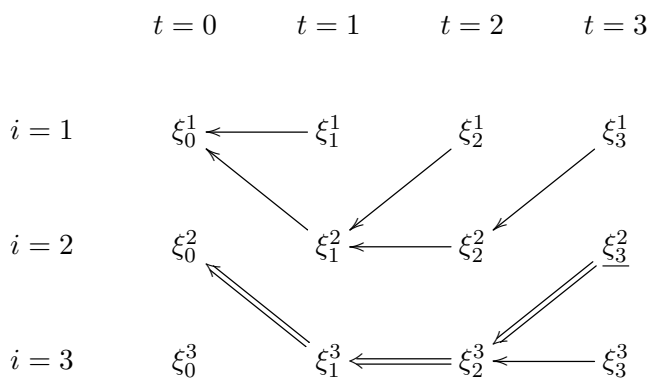
Mamy tutaj $A_1^1 = 1$, $A_1^2 = 1$, $A_1^3 = 2$, $A_2^1 = 2$ itd. W kroku propagacji losujemy zatem $\xi_1^1 \sim T(\xi_0^1, \cdot)$, $\xi_1^2 \sim T(\xi_0^1, \cdot)$ itd. Ogólnie, $\xi_t^i \sim T(\xi_{t-1}^{A_t^i}, \cdot)$.

Na końcu algorytmu PF dodamy jeszcze jeden krok: wylosujemy *jedną* z cząsteczek ξ_k^i z prawdopodobieństwem proporcjonalnym do wagi W_k^i. Jeśli tą wylosowaną cząsteczką będzie ξ_k^s, to $\xi_{0:k}^s$ oznacza linię jej przodków. Formalnie, możemy zdefiniować przez indukcję wsteczną linię przodków s, mianowicie $B_k = s$, $B_{t-1} = A_t^{B_t}$ dla $t = k, \ldots, 1$ i wtedy $\xi_{0:k}^s = (\xi_0^{B_0}, \xi_1^{B_1}, \ldots, \xi_k^s)$. Proszę prześledzić komentarz dotyczący przykładowego schematu pod pseudo-kodem. Zwróćmy uwagę na instrukcję $\xi_{0:t}^i := (\xi_{0:t-1}^{A_t^i}, \xi_t^i)$ w algorytnie PF, w której ukryta jest indukcja wsteczna. W istocie, „dolepiamy" tu nowe położenie ξ_t^i do dotychczasowej linii przodków $\xi_{0:t-1}^{A_t^i}$.

Algorytm PF z losowaniem wyróżnionej cząsteczki:

```
...
{ Ciąg instrukcji identyczny jak w algorytmie PF }
...
{ Losowanie wyróżnionej cząsteczki }
Wylosuj S z prawdopodobieństwem P(S = s) ∝ W_k^s (s = 1, ..., n); X*_{1:k} := ξ^S_{0:k}
Output: X*_{1:k}, Ẑ* = ∏_{t=0}^{k} W̄_t. { Ẑ* jest tylko nowym oznaczeniem Ẑ }
```

Wróćmy do przykładu przedstawionego na poprzednim schemacie i wyobraźmy sobie, że wyróżniona została cząsteczka ξ_3^2 (to znaczy, w końcowym momencie $t = 3$ wylosowaliśmy $S = 2$).



Linia przodków wyróżnionej cząsteczki jest oznaczona podwójną linią: zgodnie z naszą konwencją oznaczeniową jest to ciąg $\xi^2_{0:3} = (\xi^2_0, \xi^3_1, \xi^3_2, \xi^2_3)$.

7.3 Cząsteczkowe algorytmy MCMC: *pMCMC*

Każdy z algorytmów pMCMC generuje ciąg $(X(m), m = 0, 1, \ldots)$ zmiennych losowych w przestrzeni $\mathcal{X}^{k+1}$, czyli w przestrzeni trajektorii ukrytego łańcucha $X_{0:k}$. Ten ciąg jest zbieżny do rozkładu docelowego, którym jest rozkład *a posteriori* $\pi(x_{0:k}) = p(x_{0:k}|y_{0:k})$, wzór (7.1.2). Żeby zrozumieć działanie tych algorytmów, trzeba zdefinować łańcuch Markowa na rozszerzonej przestrzeni stanów, mianowicie *na przestrzeni konfiguracji algorytmu PF*. Przez konfigurację rozumiemy rodzinę wszystkich zmiennych losowych produkowanych przez PF, czyli $(\xi^i_t, A^i_t)_{i=1,\ldots,n}$ oraz S.

Cząsteczkowy niezależny algorytm Metropolisa-Hastingsa (pIMH)

Niezależny algorytm Metropolisa-Hastingsa to algorytm, w którym propozycje losuje się niezależnie z tego samego rozkładu o gęstości q. (Markowowska zależność pomiędzy kolejnymi krokami wynika z reguły akceptacji która w tym przypadku przyjmuje postać

$$a(x, x') = \frac{\pi(x')q(x)}{\pi(x)q(x')} \wedge 1. \tag{7.3.1}$$

Algorytm pIMH generuje łańcuch Markowa $(X(m), \hat{Z}(m))$ na rozszerzonej przestrzeni $\mathcal{X}^{k+1} \times \mathbb{R}$, gdzie $X(m) = X_{0:k}(m)$ jest aktualną trajektorią ukrytego procesu $X_{0:k}$, zaś $\hat{Z}(m)$ jest aktualnym estymatorem stałej normującej z. Poniżej opisujemy regułę aktualizacji, określającą prawdopodobieństwo przejścia ze stanu $(X(m-1), \hat{Z}(m-1)) = (X_{0:k}, \hat{Z})$ do stanu $(X(m), \hat{z}(m)) = (X'_{0:k}, \hat{Z}')$.

Algorytm pIMH (*particle Independent Metropolis-Hastings*), 1 krok

Input: $X_{0:k}$, $\hat{Z}$ { zapamiętane z poprzedniego kroku }

Wykonaj PF i zapamiętaj:

- Nową wyróżnioną trajektorię $X^*_{0:k}$,
- Nowy $\hat{Z}^*$.

Gen $U \sim \mathrm{U}(0,1)$

if $U \leqslant \hat{Z}^*/\hat{Z}$ then

$X'_{0:k} := X^*_{0:k}$; $\hat{Z}' := \hat{Z}^*$ { ruch zaakceptowany }

else

$X'_{0:k} := X_{0:k}$; $\hat{Z}' := \hat{Z}$ { ruch odrzucony }

endif

return $X'_{0:k}$, $\hat{Z}'$.

7.3.2 Twierdzenie. *pIMH zachowuje docelowy rozkład a posteriori*

$$\begin{aligned}\pi(x_{0:k}) = p(x_{0:k}|y_{0:k}) &= \frac{p(x_{0:k}, y_{0:k})}{p(y_{0:k})} \\ &= \frac{1}{z}\nu(x_0)w(x_0,y_0)\prod_{t=1}^{k} T(x_{t-1},x_t)w(x_t,y_t).\end{aligned}$$

Dokładniej, rozkład stacjonarny łańcucha pIMH na rozszerzonej przestrzeni stanów, zmarginalizowany do rozkładu na przestrzeni trajektorii (zapominamy o współrzędnej $\hat{z}$), jest rozkładem a posteriori $p(x_{0:k}|y_{0:k})$. Ciąg $X(m)$ zmierza do tego rozkładu docelowego.

Dowód tego twierdzenia podam w następnym podrozdziale.

Opiszę teraz kolejny algorytm rodziny pMCMC, mianowicie cząsteczkowy próbnik Gibbsa (pGS, *particle Gibbs Sampler*). Pomysł polega na tym, żeby w pojedynczym kroku łańcucha Markowa, trajektorię *jednej z cząsteczek* wziąć z poprzedniego kroku (w pseudo-kodzie poniżej jest to trajektoria ostatniej, n-tej cząsteczki) i traktować jako ustaloną. Pozostałe cząsteczki ewoluują tak, jak w algorytmie PF (przy losowaniu rodzica mogą wybrać również cząsteczkę n). Na końcu „zapominamy" o ustalonej trajektorii i wybieramy nową wyróżnioną trajektorię identycznie jak w algorytmie PF. Ta nowa trajektoria będzie przekazana do następnego kroku. Poniższy pseudo-kod przedstawia regułę aktualizacji $X(m-1) = X_{0:k}$ (wejście) do $X(m) = X'_{0:k}$ (wyjście).

Algorytm pGS (*particle Gibbs Sampler*), 1 krok

Input: $\nu(\cdot)$, $T(\cdot,\cdot)$, $w(\cdot, y_t)_{t=0,1,\ldots,k}$, $X_{0:k}$.

{ *Trajektoria n-tej cząsteczki $\xi^n_{0:t}$ jest ustalona* }

for $t = 1, \ldots, k$

$\xi^n_t := X_t$; $W^n_t := w(\xi^n_t, y_t)$; $A^n_t := t-1$;

endfor

{ *Wykonujemy ,,warunkowy PF'', przy ustalonej trajektorii $\xi^n_{0:t}$:* }

{ *Inicjalizacja pozostałych cząsteczek* }

for $i = 1, \ldots, n-1$

Gen $\xi^i_0 \sim \nu(\cdot)$;

Oblicz $W^i_0 := w(\xi^i_0, y_0)$;

endfor

{ *Główna pętla* }

for $t = 1, \ldots, k$}

for $i = 1, \ldots, n-1$

{ *Repróbkowanie* }

Wylosuj A^i_t z prawdopodobieństwem $\mathbb{P}(A^i_t = j) \propto W^j_{t-1}$ $(j = 1, \ldots, n)$;

{ *Propagacja* }

Gen $\xi^i_t \sim T(\xi^{A^i_t}_{t-1}, \cdot)$; $\xi^i_{0:t} := (\xi^{A^i_t}_{0:t-1}, \xi^i_t)$;

{ *Ważenie* }

Oblicz $W^i_t = w(\xi^i_t, y_t)$;

endfor

endfor

{ *Wybierz nową wyróżnioną trajektorię* }

Wylosuj S' z prawdopodobieństwem $\mathbb{P}(S' = s') \propto W^{s'}_k$ $(s' = 1, \ldots, n)$;

$X'_{0:k} := \xi^{S'}_{0:k}$.

Output: $X'_{0:k}$.

7.3.3 Twierdzenie. *Algorytm pGS zachowuje rozkład a posteriori*

$$\pi(x_{0:k}) = p(x_{0:k}|y_{0:k}).$$

Łańcuch Markowa $X(m)$ generowany przez algorytm pGS zmierza do tego rozkładu.

Dowód tego twierdzenia podam w następnym podrozdziale.

Co prawda Twierdzenie 7.3.3 zapewnia, że algorytm pGS jest poprawny, ale szybkość zbieżności do rozkładu docelowego nie jest satysfakcjonująca. Na czym polega problem, wyjaśnia Rysunek **??**. Chodzi o to, że wiele trajektorii o różnych końcach może mieć ten sam początek. W rezultacie, wybierając nowy koniec wyróżnionej trajektorii, S', mamy dużą szansę na to, że jej początek będzie taki sam jak początek starej trajektorii, odziedziczonej z kroku poprzedniego. To zjawisko, określane jako „degeneracja trajektorii", powoduje, że pGS nieźle estymuje rozkład *a posteriori* blisko końca, ale marnie estymuje rozkład *a posteriori* blisko początku. Jest jednak prosty sposób poprawienia pGS przez wprowadzenie „losowania przodków" (*ancestor sampling*).

Algorytm pGAS (*particle Gibbs with Ancestor Sampling*)

Input: $\nu(\cdot)$, $T(\cdot,\cdot)$, $w(\cdot, y_t)_{t=0,1,\ldots,k}$, $X_{0:k}$.

{ *Tak samo jak pGS, ale NIE wykonujemy instrukcji* $A_t^n := t-1$ }

for $t = 1, \ldots, k$

$\xi_t^n := X_t$; $W_t^n := w(\xi_t^n, y_t)$;

endfor

{ *Tak samo jak pGS, tylko w głównej pętli dodajemy losowanie* A_t^n*:* }

for $t = 1, \ldots, k$}

for $i = 1, \ldots, n-1$

{ *Repróbkowanie* }

Wylosuj A_t^i z prawdopodobieństwem $\mathbb{P}(A_t^i = j) \propto W_{t-1}^j$ $(j = 1, \ldots, n)$;

{ *Propagacja* }

Gen $\xi_t^i \sim T(\xi_{t-1}^{A_t^i}, \cdot)$; $\xi_{0:t}^i := (\xi_{0:t-1}^{A_t^i}, \xi_t^i)$;

{ *Ważenie* }

Oblicz $W_t^i = w(\xi_t^i, y_t)$;

endfor

Wylosuj A_t^n z prawdopodobieństwem $\mathbb{P}(A_t^n = j) \propto W_{t-1}^j T(\xi_{t-1}^j, \xi_t^n)$
$(j = 1, \ldots, n)$;

$\xi_{0:t}^n := (\xi_{0:t-1}^{A_t^n}, \xi_t^n)$;

endfor

{ *Dalej tak samo jak pGS* }

7.3.4 Twierdzenie. *Algorytm pGAS zachowuje rozkład a posteriori*

$$\pi(x_{0:k}) = p(x_{0:k}|y_{0:k}).$$

Łańcuch Markowa $X(m)$ generowany przez algorytm pGAS zmierza do tego rozkładu.

Porównanie Rysunku 7.3 z Rysunkiem 7.2 pokazuje efekt „losowania przodków".

7.4 Kluczowy lemat i dowody poprawności algorytmów pMCMC

Rozkład *extended proposal*

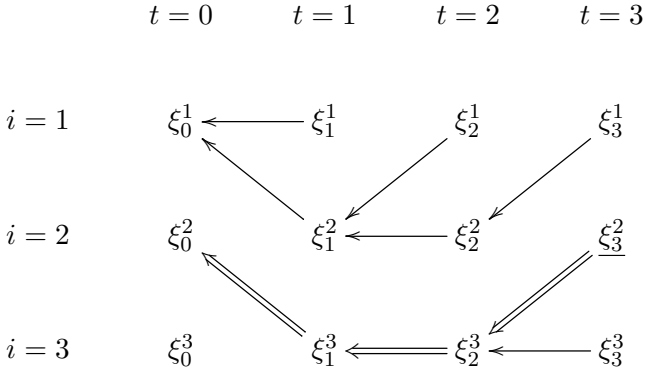
Łączny rozkład prawdopodobieństwa wszystkich zmiennych w algorytmie PF, czyli $\xi_{0:k}^{1:n}$ i $A_{1:k}^{1:n}$ i S nazywamy rozszerzonym rozkładem propozycji, *extended proposal*. (Najważniejszy pomysł polega na rozważeniu rozkładu prawdopodobieństwa na rozszerzonej przestrzeni stanów, składającej się z wielu „próbnych" trajektorii ukrytego łańcucha Markowa.) Ten rozkład będziemy oznaczali symbolem ψ. Wartości zmiennych losowych A, W, $\hat{Z}$ będziemy oznaczali małymi literami a, w, $\hat{z}$. Chwila zastanowienia wystarczy, żeby zauważyć, że

$$\psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}) = \prod_{i=1}^{n} \nu(\xi_0^i) \prod_{t=1}^{k} \prod_{i=1}^{n} \frac{W_{t-1}^{a_t^i}}{n\bar{W}_{t-1}} T\left(\xi_{t-1}^{a_t^i}, \xi_t^i\right),$$

gdzie $\bar{W}_t = \frac{1}{n}\sum_{i=1}^{n} W_t^i$ i $W_t^i = w(\xi_t^i, y_t)$. Jeśli rozważymy dodatkowo wybór wyróżnionej cząsteczki, to otrzymamy rozkład

$$\psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s) = \psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}) \frac{W_k^s}{n\bar{W}_k}.$$

Pamiętajmy, że trajektoria wyróżnionej cząsteczki jest zdefiniowana jako $\xi_{1:k}^s = (\xi_0^{b_0}, \xi_1^{b_1}, \ldots, \xi_k^{b_k})$, gdzie $b_k = s$, $b_{t-1} = a_t^{b_t}$, dla $t = k, k-1, \ldots, 1$. Wróćmy do przykładowego, wcześniej już rozpatrywanego grafu:



$$\begin{aligned}
\psi = {} & \nu(\xi_0^1)\nu(\xi_0^2)\nu(\xi_0^3) \\
& \times \frac{w_0^1}{n\bar{w}_0} T(\xi_0^1, \xi_1^1) \frac{w_0^1}{n\bar{w}_0} T(\xi_0^1, \xi_1^2) \frac{w_0^2}{n\bar{w}_0} T(\xi_0^2, \xi_1^3) \\
& \times \frac{w_1^2}{n\bar{w}_1} T(\xi_1^2, \xi_2^1) \frac{w_1^2}{n\bar{w}_1} T(\xi_1^2, \xi_2^2) \frac{w_1^3}{n\bar{w}_1} T(\xi_1^3, \xi_2^3) \\
& \times \frac{w_2^2}{n\bar{w}_2} T(\xi_2^2, \xi_3^1) \frac{w_2^3}{n\bar{w}_2} T(\xi_2^3, \xi_3^2) \frac{w_2^3}{n\bar{w}_2} T(\xi_2^3, \xi_3^3) \\
& \times \frac{w_3^2}{n\bar{w}_3}
\end{aligned}$$

Rozkłady *extended proposal* i *extended target*

Rozszerzony rozkład propozycji, *extended proposal* jest dany wzorem:

$$\psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s) = \prod_{i=1}^{n} \nu(\xi_0^i) \left[\prod_{t=1}^{k} \prod_{i=1}^{n} \frac{W_{t-1}^{a_t^i}}{n\bar{W}_{t-1}} T\left(\xi_{t-1}^{a_t^i}, \xi_t^i\right) \right] \frac{W_k^s}{n\bar{W}_k}.$$

Kluczowe jest następujące wyrażenie:

$$\begin{aligned} \psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s)\frac{\hat{z}}{z} = \underbrace{\frac{1}{z}\nu(\xi_0^{b_0}) \left[\prod_{t=1}^{k} W_{t-1}^{b_{t-1}} T\left(\xi_{t-1}^{b_{t-1}}, \xi_t^{b_t}\right) \right] W_k^{b_k}}_{\pi(\xi_0^{b_0}, \xi_1^{b_1}, \ldots, \xi_k^{b_k})} \frac{1}{n^{k+1}} \\ \times \prod_{i \neq b_0}^{n} \nu(\xi_0^i) \prod_{t=1}^{k} \prod_{\substack{i=1 \\ i \neq b_t}}^{n} \frac{W_{t-1}^{a_t^i}}{n\bar{W}_{t-1}} T\left(\xi_{t-1}^{a_t^i}, \xi_t^i\right). \end{aligned} \tag{7.4.1}$$

Dowód równości (7.4.1). Dowód polega na przestawieniu kolejności czynników we wzorze na ψ. Na początku grupujemy czynniki „wzdłóż wyróżnionej trajektorii ". Przypomnijmy, że $\hat{z} = \prod_{t=0}^{k} \bar{W}_t$. Ponieważ rozważamy ψ pomnożone przez $\hat{z}/z$, to skracają się mianowniki $\bar{w}_0, \bar{w}_1 \ldots \bar{w}_k$ i otrzymujemy wyrażenie oznaczone symbolem π (pojawia się jeszcze czynnik $1/n^{k+1}$). Pozostałe czynniki odpowiadają tym parom (a_t^i, i), które nie leżą na wyróżnionej trajektorii, czyli $i \neq b_t$. □

Najważniejszy lemat sprowadza się do odpowiedniej interpretacji czynników we wzorze (7.4.1).

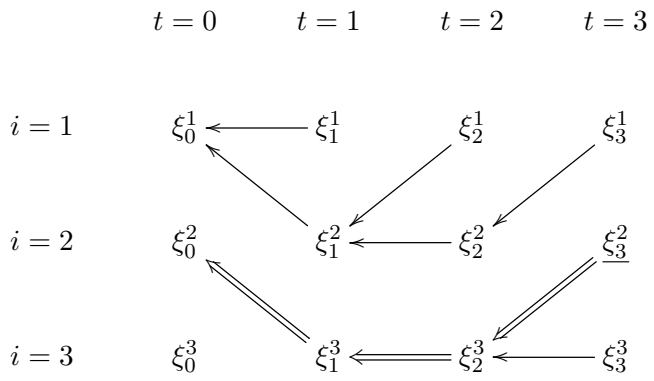
7.4.2 Lemat (Kluczowy!)**.**

$$\begin{aligned} \psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s)\frac{\hat{z}}{z} &= \underbrace{\pi(\xi_0^{b_0}, \xi_1^{b_1}, \ldots, \xi_k^{b_k})}_{\textit{rozkład docelowy}} \frac{1}{n^{k+1}} \\ &\quad \times \underbrace{\psi_{\text{cond}}(\xi_{0:k}^{1:n}, a_{0:k}^{1:n} | \xi_{0:k}^{b_{0:k}})}_{\textit{rozkład „warunkowego PF"}} \\ &= \phi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s). \end{aligned}$$

Symbol ϕ po prawej stronie oznacza rozszerzony rozkład docelowy, czyli extended target.

Dowód. Wystarczy zauważyć, że π we wzorze (7.4.1) jest (prawidłowo unormowanym) rozkładem a posteriori $p(x_{0:k}|y_{0:k})$ dla $x_{0:k} = \xi_{0:k}^s$. Czynnik oznaczony ψ_{cond} jest rozkładem prawdopodobieństwa „warunkowego PF" określonego w algorytmie pGS. Czynnik $1/n^{k+1}$ jest obecny dlatego, że w każdym czasie $t = 0, 1, \ldots, k$, wyróżniona cząsteczka może mieć numer $i = 1, \ldots, n$. („Warunkowy PF" został zapisany w ten sposób, że wyróżniona cząsteczka ma zawsze numer n). □

Prześledźmy ten dowód kluczowego lematu na przykładzie:



Konfiguracja pokazana na tym rysunku jest wylosowana z prawdopodobieństwem ψ,

$$
\begin{aligned}
\psi &= \nu(\xi_0^1)\nu(\xi_0^2)\nu(\xi_0^3) \\
&\times \frac{w_0^1}{n\bar{w}_0} T(\xi_0^1,\xi_1^1) \frac{w_0^1}{n\bar{w}_0} T(\xi_0^1,\xi_1^2) \frac{w_0^2}{n\bar{w}_0} T(\xi_0^2,\xi_1^3) \\
&\times \frac{w_1^2}{n\bar{w}_1} T(\xi_1^2,\xi_2^1) \frac{w_1^2}{n\bar{w}_1} T(\xi_1^2,\xi_2^2) \frac{w_1^3}{n\bar{w}_1} T(\xi_1^3,\xi_2^3) \\
&\times \frac{w_2^2}{n\bar{w}_2} T(\xi_2^2,\xi_3^1) \frac{w_2^3}{n\bar{w}_2} T(\xi_2^3,\xi_3^2) \frac{w_2^3}{n\bar{w}_2} T(\xi_2^3,\xi_3^3) \\
&\times \frac{w_3^2}{n\bar{w}_3} \\
&= \nu(\xi_0^2) w_0^2 T(\xi_0^2,\xi_1^3) w_1^3 T(\xi_1^3,\xi_2^3) w_2^3 T(\xi_2^3,\xi_3^2) w_3^2 \cdot \frac{1}{n\bar{w}_0}\frac{1}{n\bar{w}_1}\frac{1}{n\bar{w}_2}\frac{1}{n\bar{w}_3} \\
&\times \nu(\xi_0^1) \qquad \nu(\xi_0^3) \\
&\times \frac{w_0^1}{n\bar{w}_0} T(\xi_0^1,\xi_1^1) \frac{w_0^1}{n\bar{w}_0} T(\xi_0^1,\xi_1^2) \\
&\times \frac{w_1^2}{n\bar{w}_1} T(\xi_1^2,\xi_2^1) \frac{w_1^2}{n\bar{w}_1} T(\xi_1^2,\xi_2^2) \\
&\times \frac{w_2^2}{n\bar{w}_2} T(\xi_2^2,\xi_3^1) \qquad\qquad \frac{w_2^3}{n\bar{w}_2} T(\xi_2^3,\xi_3^3) \\
&= \pi(\xi_0^2,\xi_1^3,\xi_2^3,\xi_3^2) \cdot \frac{z}{n^4 \hat{z}} \\
&\times \psi_{\text{cond}}(\cdots).
\end{aligned}
$$

Czerwonym kolorem są oznaczone czynniki odpowiadające wyróżnionej trajektorii (podwójne strzałki w grafie).

Możemy teraz wyjaśnić nazwy nadane rozkładom ψ i ϕ powyżej:

- $\psi =$ *extended proposal* – łączny rozkład prawdopodobieństwa wszystkich zmiennych w algorytmie PF (używany jako rozkład propozycji w algorytmie pIMH).
- $\phi =$ *extended target* – rozkład dla którego rozkładem brzegowym jest π, czyli rozkład docelowy (*target*).

Innymi słowy, rozszerzony rozkład propozycji ϕ możemy zdefiniować w następujący sposób:

- Wyobrażamy sobie, że losujemy trajektorię $X_{0:k} \sim \pi$. (Oczywiście, tego nie umiemy zrobić, ale można sobie wyobrazić.)
- Wybieramy losowo ciąg indeksów $b_{0:k}$ ze zbioru $\{1, \dots, n\}^{k+1}$.
- Umieszczamy trajektorię $\xi_{0:k}^{b_{0:k}} := X_{0:k}$.
- Uruchamiamy „warunkowy PF".

Zasadnicze twierdzenia dotyczące algorytmów pMCMC są łatwymi wnioskami z Lematu 7.4.2. Na początek przytoczymy dowód nieobciążoności estymatora $\hat{Z}$.

7.4.3 Twierdzenie. *Na wyjściu algorytmu PF,* $\hat{Z} = \prod_{t=0}^{k} \bar{W}_t$ *jest nieobciążonym estymatorem stałej normującej* Z*,*

$$\mathbb{E}_\psi \hat{Z} = z.$$

Dowód. Ponieważ

$$\psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s)\frac{\hat{z}}{z} = \phi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s),$$

więc

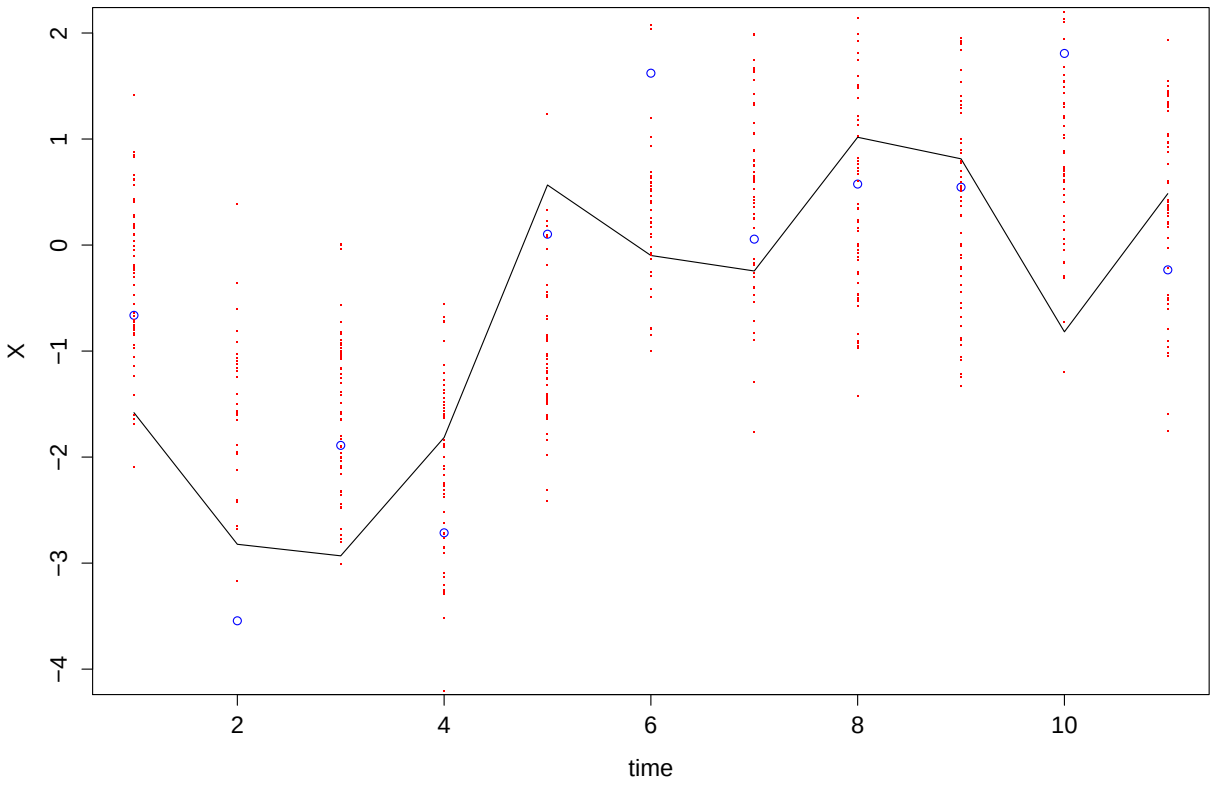
$$\begin{aligned}\mathbb{E}_\psi \frac{\hat{Z}}{z} &= \sum_{\xi,a,s} \psi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s)\frac{\hat{z}}{z} = \sum_{\xi,a,s} \phi(\xi_{0:k}^{1:n}, a_{1:k}^{1:n}, s) \\ &= 1.\end{aligned}$$

□

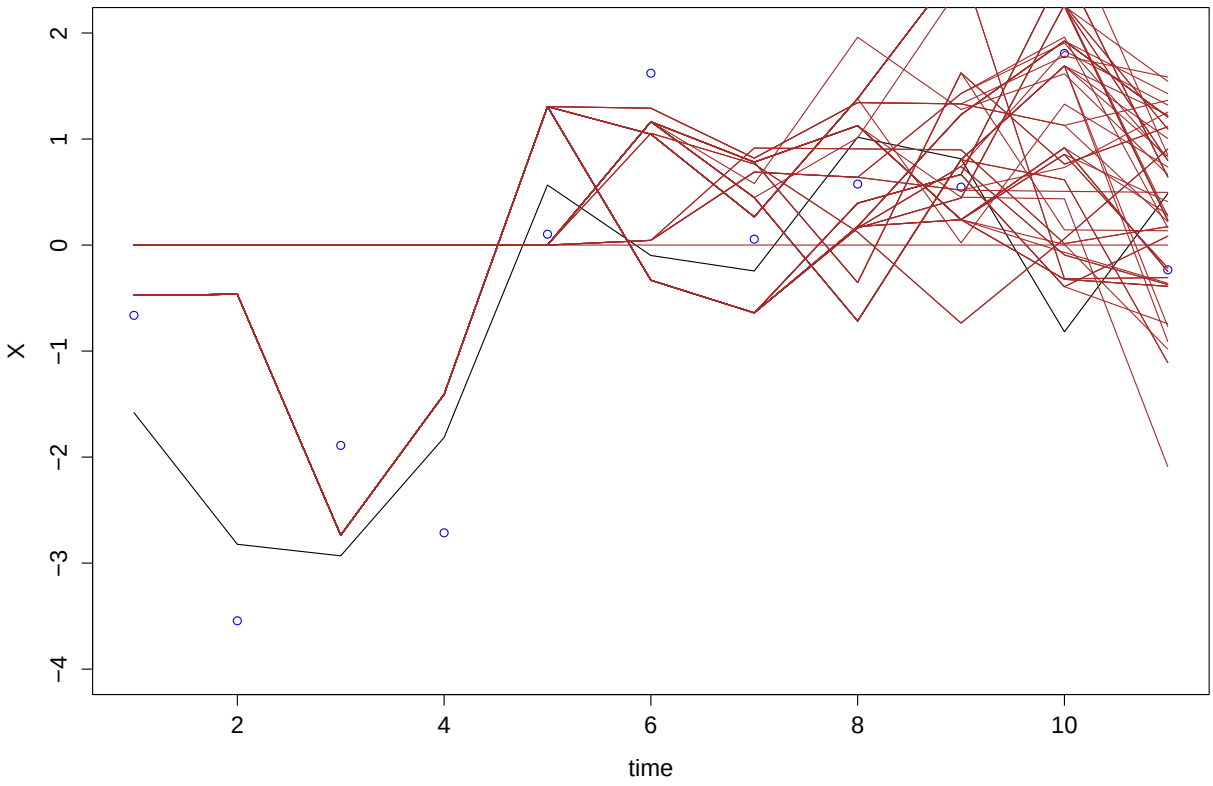
Dowód Twierdzenia 7.3.2. Symulujemy łańcuch Markowa $\underline{X}(m)$ na *przestrzeni konfiguracji* PF, zgodnie z regułą IMH. Zauważmy, że konfiguracja PF jest ważonym grafem $\underline{x} = (\xi_{0:k}^{1:n}, a_{0:k}^{1:n}, s) \in \mathcal{X}^{n\times(k+1)} \times (1:n)^{n\times k} \times (1:n)$. Stosujemy regułę akceptacji MH, przyjmując za ψ za rozkład propozycji i ϕ za rozkład docelowy. Z Kluczowego Lematu wynika, że

$$\begin{aligned}\alpha(\underline{x}, \underline{x}^*) &= \frac{\phi(\underline{x}^*)\psi(\underline{x})}{\psi(\underline{x}^*)\phi(\underline{x})} \wedge 1 \\ &= \frac{\hat{z}^*}{\hat{z}} \wedge 1.\end{aligned}$$

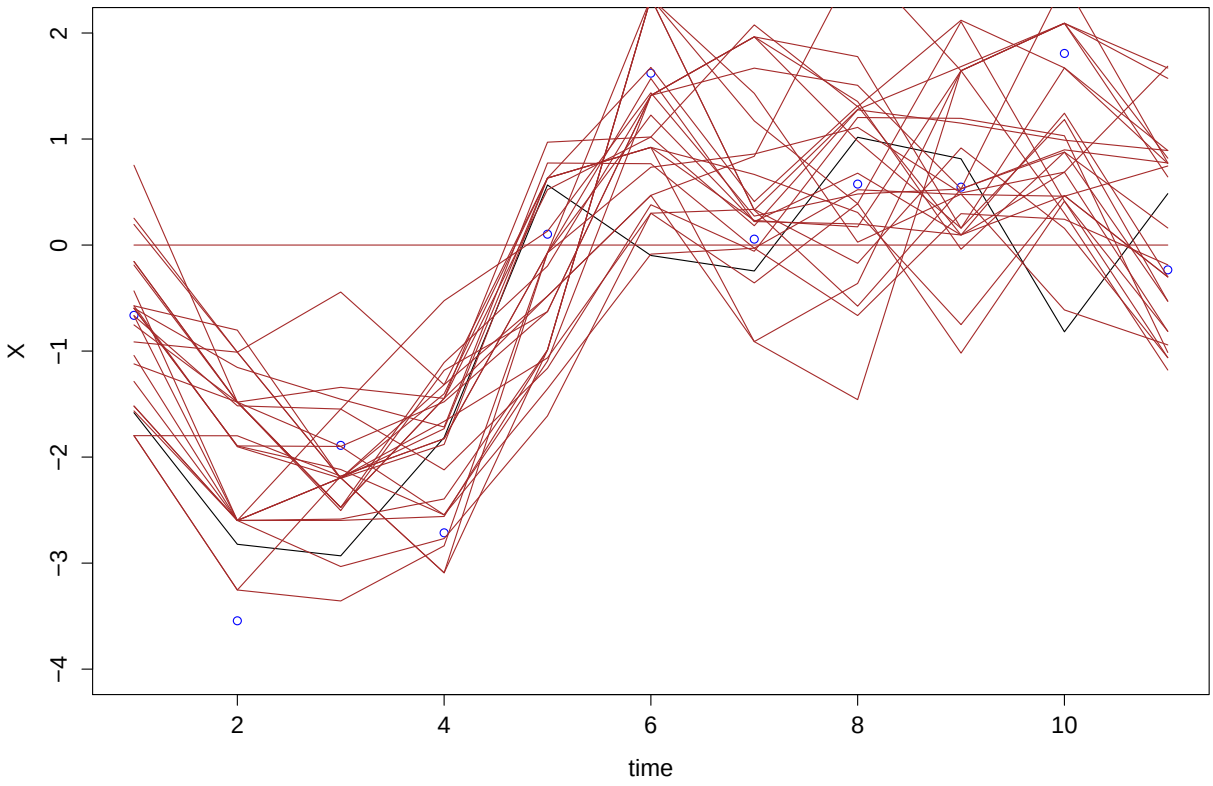
Łańcuch $\underline{X}(m)$ zachowuje więc rozszerzony rozkład docelowy ϕ. Wiemy (z Kluczowego Lematu), że ϕ zmarginalizowany do wyróżnionej trajektorii jest rozkładem docelowym π. □



Rysunek 7.1: „Chmurki punktów” generowane przez PF.



Rysunek 7.2: Trajektorie generowane przez pGS.



Rysunek 7.3: Trajektorie generowane przez pGAS.

Bibliografia

[1] S. Asmussen and P.W. Glynn: *Stochastic Simulation, Algorithms and Analysis*, Springer, 2007.

[2] P. Bremaud: *Markov Chains: Gibbs Fields, Monte Carlo Simulation and Queues*, Springer Verlag, 1999.

[3] S. Geman and D. Geman (1984): Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE-PAMI*, 6, 721–741.

[4] C.J. Geyer (1992): Practical Markov Chain Monte Carlo. *Statistical Science* 7 (4), 473–511.

[5] C.J. Geyer (1995, 2005): *Markov chain Monte Carlo Lecture Notes.* Dostępne na `www.stat.umn.edu/geyer`.

[6] W.K. Hastings (1970): Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* 57, 97–109.

[7] F.K.C. Kingman: *Procesy Poissona*, PWN 2002.

[8] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller (1953): Equation of state calculation by fast computing machines, *Journal of Chemical Physics*, 21 (6), 1087–1092.

[9] E. Nummelin (2002): MC's for MCMC'ists. *International Statistical Review*, 70, 215–240.

[10] J.G. Propp, D.B. Wilson (1996): Exact Sampling with Coupled Markov Chains and Applications to Statistical Mechanics, *Random Structures and Algorithms* 9, 232–252.

[11] B.D. Ripley: *Stochastic Simulation*, Wiley & Sons, 1987.

[12] J. Diebolt, C.P. Robert (1994): Estimation of Finite Mixture Distributions Through Bayesian Sampling, *Journal of the Royal Statistical Society, Series B* 56, 2, 363–375.

[13] C.P. Robert, G. Casella: *Monte Carlo Statistical Methods*, Springer 2004.

[14] G.O. Roberts, J.S. Rosenthal (2004): General state space Markov chains and MCMC algorithms. *Probability Surveys* 1, 20–71.

[15] J.S. Rosenthal (1995): Rates of convergence for Gibbs sampling for variance component models, *Annals of Statistics* 23, 740–761.

[16] M. Rybiński: *Krótkie wprowadzenie do R dla programistów, z elementami statystyki opisowej*, WMIM UW 2009.

[17] R. Zieliński, R. Wieczorkowski: *Komputerowe generatory liczb losowych*, WNT, Warszawa, 1997.

[18] R Development Core Team (2011). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.