

# Zastosowanie metody Cross-entropy do symulacji rzadkich zdarzeń

Antoni Bieniasz

Antoni Bieniasz

Uniwersytet Wrocławski

15 kwietnia 2026

# Estymacja całki

Mamy całkę:

$$I = \int_E k(x)f(x)dx = \mathbb{E}k(X)$$

gdzie  $E$  jest pewnym podzbiorem  $\mathbb{R}$  albo  $\mathbb{R}^d$ ,  $f$  to funkcja gęstości prawdopodobieństwa zmiennej losowej  $X \in E$  a funkcja  $k : \mathbb{R}^d \rightarrow \mathbb{R}$  spełnia  $\int_E (k(x))^2 f(x) dx < \infty$ .

# CMC

Do przybliżenia wartości danej całki możemy skorzystać z estymatora CMC (Crude Monte Carlo estimator):

$$\hat{Y}_R^{CMC} = \frac{1}{R} \sum_{j=1}^R Y_j,$$

gdzie  $Y_1 = k(X_1), \dots, Y_R = k(X_R)$  to niezależne próbki  $Y = k(X), X \sim f$ .

Tak otrzymany estymator ma jednak zazwyczaj dużą wariancję względem innych estymatorów, między innymi względem estymatora otrzymywanego poprzez Importance Sampling.

# Importance Sampling

Niech  $\tilde{X}$  będzie zmienną losową z funkcją gęstości  $\tilde{f}$ . Zakładamy, że jest ona dodatnia na  $E$ . Wtedy:

$$I = \int_E k(x) \frac{f(x)}{\tilde{f}(x)} \tilde{f}(x) dx = \mathbb{E}k_1(\tilde{X}),$$

czyli  $k_1(x) = k(x) \frac{f(x)}{\tilde{f}(x)}$ .

Definiujemy estymator Importance Sampling jako:

$$\hat{Y}_R^{IS} = \frac{1}{R} \sum_{j=1}^R k(\tilde{X}_j) \frac{f(\tilde{X}_j)}{\tilde{f}(\tilde{X}_j)}$$

## Wybór $\tilde{f}$

Poprzez właściwy wybór  $\tilde{f}$  możemy osiągnąć istotnie mniejszą wariancję w porównaniu do tej pochodzącej od estymatora CMC. Teoretycznie najlepsza gęstość prawdopodobieństwa pozwalająca otrzymać estymator o wariancji równej 0 to:

$$g^* = \frac{k(x)f(x)}{I}.$$

W praktyce jednak nie możemy go użyć, bo wymaga wiedzy o wartości  $I$ .

# Metoda Cross-Entropy

Możemy jednak szukać rozkładu prawdopodobieństwa o gęstości "bliskiej"  $g^*$ . W tym celu skorzystamy z dywergencji Kullbacka-Leiblera pomiędzy  $g^*$  a szukaną gęstością  $\tilde{f}_\theta$ . Jest ona definiowana jako:

$$\begin{aligned}\mathcal{D}(g^*, \tilde{f}_\theta) &= \int g^*(x) \log g^*(x) dx - \int g^*(x) \log \tilde{f}_\theta(x) dx \\ &= E_{g^*} \log \left( \frac{g^*(X)}{\tilde{f}_\theta(X)} \right)\end{aligned}$$

$\theta$  opisuje parametry rozkładu (w przypadku rozkładu normalnego są to średnia (wektor średnich), wariancja (macierz kowariancji)).

# Minimalizacja dywergencji

Celem naszym jest znalezienie takiego  $\theta$ , że dywergencja między  $g^*$  a  $\tilde{f}$  jest jak najmniejsza. (szukany rozkład  $\tilde{f}$  jest jak "najbliżej" tego optymalnego  $g^*$ ). Mamy:

$$\begin{aligned}\theta^* &= \arg \min_{\theta} \mathcal{D}(g^*, \tilde{f}_{\theta}) \\ &= \arg \max_{\theta} \int g^*(x) \log \tilde{f}_{\theta}(x) dx \\ &= \arg \max_{\theta} \int \frac{k(x)f(x)}{I} \log \tilde{f}_{\theta}(x) dx \\ &= \arg \max_{\theta} \int k(x) \log \tilde{f}_{\theta}(x) f(x) dx \\ &= \arg \max_{\theta} \mathbb{E}_f [k(X) \log \tilde{f}_{\theta}(X)]\end{aligned}$$

# Minimalizacja dywergencji

Znalezienie dokładnej wartości  $\theta^*$  poprzez  $\arg \max_{\theta} \mathbb{E}_f [k(X) \log \tilde{f}_{\theta}(X)]$  jest często trudne. Zamiast tego można jednak wysymulować próbę  $X_1, \dots, X_n$  z rozkładu o gęstości  $f$  i estymować  $\theta^*$  poprzez:

$$\hat{\theta}^* = \arg \max_{\theta} \frac{1}{N} \sum_{i=1}^N k(X_i) \log \tilde{f}_{\theta}(X_i)$$

W tym celu można skorzystać z metody gradientowej.

# Rzadkie zdarzenia

Rzadkie zdarzenia to zdarzenia, których prawdopodobieństwo jest niskie. Dokładna wartość nie jest konkretnie określona, przykładowo może ono być rzędu  $10^{-3}$  lub  $10^{-5}$ .

Rozważamy rodzinę rozkładów prawdopodobieństwa  $\{f_\theta(x)\}$  sparametryzowaną przez  $\theta \in \Theta$ . Niech  $g$  będzie funkcją rzeczywistą na  $\mathbb{R}^d$ . Chcemy wiedzieć jakie jest prawdopodobieństwo, że  $g(X) \geq \gamma$  ( $g(X) \leq \gamma$ ), przy ustalonej  $\gamma \in \mathbb{R}$ ,  $X \sim f_\theta$ . Mamy (dla  $g(X) \geq \gamma$ ):

$$I = \mathbb{P}_\theta(g(X) \geq \gamma) = \int \mathbb{I}(g(x) \geq \gamma) f_\theta(x) dx = \int k(x) f_\theta(x) dx$$

# Badany problem

Mamy całkę:

$$P_f = \int k(x) f_\theta(x) dx = \int \mathbb{I}(g(x) \leq 0) f_\theta(x) dx,$$

gdzie  $f_\theta(x)$  jest gęstością 2-wymiarowego standardowego rozkładu normalnego, a  $g(x)$  to taka funkcja, że  $P_f$  jest rzędu  $10^{-3}$ . Chcemy znaleźć tę wartość poprzez Importance Sampling z rozkładu uzyskanego poprzez metodę Cross-Entropy.

# Metoda Gradient Descent

Estymator  $\hat{\theta}^*$  maksymalizuje poniższe wyrażenie:

$$\hat{\theta}^* = \arg \max_{\theta} \frac{1}{N} \sum_{i=1}^N k(X_i) \log \tilde{f}_{\theta}(X_i)$$

W celu jego obliczenia skorzystamy z metody gradient descent.

# Metoda Gradient Descent

Zakładamy, że  $f \in C^1$ . Na początku wybieramy tempo zmiany funkcji (oznaczamy jako  $t$ ). Określamy liczbę kroków algorytmu (np. 1000). Losujemy początkowe parametry  $\theta = (\theta_1, \dots, \theta_n)$

- 1 Obliczamy gradient:  $\frac{\partial f}{\partial \theta} = (\frac{\partial f}{\partial \theta_1}, \dots, \frac{\partial f}{\partial \theta_n})$ .
- 2 Obliczamy wartość funkcji  $f$  w  $\theta - t \cdot \frac{\partial f}{\partial \theta}$ .
- 3 Powtarzamy w kolejnym kroku daną procedurę.

Działanie całej metody opiera się na fakcie, wyznaczenia przez  $t \cdot \frac{\partial f}{\partial \theta}$  kierunku, w którym funkcja maleje najszybciej.

## Przypadek 2-wymiarowego rozkładu normalnego

W przypadku 2-wymiarowego rozkładu normalnego parametrami są średnie  $\mu_1, \mu_2$  oraz wariancje  $\sigma_1^2, \sigma_2^2$ , a także kowariancja  $cov(X_1, X_2)$ . O ile średnie mogą przyjmować dowolne wartości rzeczywiste, to trzeba w pewien sposób zapewnić nieujemność wariancji. Korzystamy w tym celu z rozkładu Choleskiego dla macierzy kowariancji:

$$\Sigma = \begin{pmatrix} a & 0 \\ b & c \end{pmatrix} \begin{pmatrix} a & b \\ 0 & c \end{pmatrix} = \begin{pmatrix} a^2 & ab \\ ab & b^2 + c^2 \end{pmatrix}.$$

Parametrami, dla których liczymy gradient są więc średnie oraz  $a, b, c$ .

Implementacja poprzez bibliotekę torch w Pythonie:

```
lr = 0.05 #t
num_epochs = 3000

mu = torch.randn(2, requires_grad=True, device=device)

alpha = torch.randn((), requires_grad=True, device=device)
beta = torch.randn((), requires_grad=True, device=device)
gamma = torch.randn((), requires_grad=True, device=device)

optimizer = torch.optim.Adam([mu, alpha, beta, gamma], lr=lr)
```

## Wyniki i porównanie z inną pracą

Mamy funkcje: Convex limit-state function:

$$g_1(x) = 0.1(x_1 - x_2)^2 - \frac{1}{\sqrt{2}}(x_1 + x_2) + 2.5$$

oraz Parabolic limit-state function:

$$g_2(x) = b - x_2 - \kappa(x_1 - e)^2,$$

gdzie  $b = 5$ ,  $\kappa = 0.5$  i  $e = 0.1$ .

Poprzez metodę Cross-entropy wybrano dla nich optymalny rozkład 2-wymiarowy normalny do Importance Sampling.

## Wyniki i porównanie z inną pracą

Następnie dla 500, 1000, oraz 2000 próbek przeprowadzono 500 symulacji w celu wyestymowania  $P_f$  oraz współczynnika kowariancji  $\delta_{P_f}$ , który można rozpisać jako:

$$\delta_{P_f} = \sqrt{\frac{1 - P_f}{n_s P_f}},$$

gdzie  $n_s$  to rozmiar próby (500, 1000 lub 2000).

Wyniki porównano z metodami SiS oraz SuS z taką samą liczbą próbek oraz symulacji. Estymator  $\hat{P}_f$  porównano z  $P_f$  wyestymowanym poprzez CMC na podstawie wystarczająco dużej próby.

# Convex limit-state function

$P_f$  obliczone na podstawie symulacji Monte Carlo z  $10^7$  próbkami:  
 $4.21 \times 10^{-3}$

Liczba próbek	SuS	SuS	SiS	SiS	Cross-ent.	Cross-ent.
parametr	$\hat{P}_f$	$\delta_{\hat{P}_f}$	$\hat{P}_f$	$\delta_{\hat{P}_f}$	$\hat{P}_f$	$\delta_{\hat{P}_f}$
500	$4.28 \times 10^{-3}$	0.28	$3.90 \times 10^{-3}$	0.20	$4.226 \times 10^{-3}$	0.69
1000	$4.25 \times 10^{-3}$	0.21	$3.97 \times 10^{-3}$	0.12	$4.217 \times 10^{-3}$	0.49
2000	$4.26 \times 10^{-3}$	0.13	$4.13 \times 10^{-3}$	0.07	$4.211 \times 10^{-3}$	0.34

# Parabolic limit-state function

$P_f$  obliczone na podstawie symulacji Monte Carlo z  $10^7$  próbkami:  
 $3.01 \times 10^{-3}$

Liczba próbek	SuS	SuS	SiS	SiS	Cross-ent.	Cross-ent.
parametr	$\hat{P}_f$	$\delta_{\hat{P}_f}$	$\hat{P}_f$	$\delta_{\hat{P}_f}$	$\hat{P}_f$	$\delta_{\hat{P}_f}$
500	$3.09 \times 10^{-3}$	0.30	$2.47 \times 10^{-3}$	0.24	$3.019 \times 10^{-3}$	0.82
1000	$3.06 \times 10^{-3}$	0.22	$2.76 \times 10^{-3}$	0.15	$3.320 \times 10^{-3}$	0.58
2000	$2.99 \times 10^{-3}$	0.15	$2.83 \times 10^{-3}$	0.10	$2.984 \times 10^{-3}$	0.41