Laboratorium z rachunku prawdopodobieństwa

Zadania treningowe przed kolokwium zaliczeniowym
(wersja `Python`)

Spacer losowy i prawo iterowanego logarytmu

Rozważmy następujący prosty spacer losowy po osi liczbowej: umieszczamy pionek w punkcie 0, a następnie w każdym ruchu przesuwamy go losowo o 1 w prawo lub o 1 w lewo. Łącznie wykonujemy w ten sposób n ruchów. Formalnie definiujemy taki spacer w następujący sposób. Niech $X_1, X_2, \ldots, X_n$ będą niezależnymi zmiennymi losowymi, takimi, że

$$\mathbb{P}(X_i = -1) = \mathbb{P}(X_i = 1) = \frac{1}{2}, \quad i = 1, 2, \ldots, n$$

Zmienną losową X_i interpretujemy jako zmianę pozycji pionka w i-tym kroku – w skrócie będziemy ją nazywać **i-tym krokiem**. Wówczas

$$S_i = X_1 + X_2 + \ldots + X_i, \quad i = 1, 2, \ldots, n$$

oznacza **pozycję** pionka po n krokach. Ciąg $(S_1, S_2, \ldots)$ kolejnych pozycji nazywamy **trajektorią** spaceru losowego. Na gruncie teorii możemy rozważać nieskończone spacery losowe, tzn. przyjmować $n = \infty$. Takie nieskończone spacery losowe mają następującą własność.

Twierdzenie (prawo iterowanego logarytmu). Niech $S_1, S_2, \ldots$ będzie trajektorią nieskończonego prostego spaceru losowego. Wówczas

$$\limsup_{n\to\infty} \frac{S_n}{\sqrt{2n\log\log n}} = -\liminf_{n\to\infty} \frac{S_n}{\sqrt{2n\log\log n}} = 1 \quad \text{z prawdopodobiestwem 1.}$$

Tutaj log oznacza logarytm naturalny. Sprawdzimy działanie prawa iterowanego logarytmu na skończonym spacerze losowym.

W poniższych zadaniach powinno się używać bibliotek:

[ ]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Zadanie 1. Wylosuj 30 razy po $n = 200$ kroków spaceru losowego i zapisz je w macierzy `numpy.ndarray` z 200 wierszami i 30 kolumnami, tj. rozmiaru (200,30)

Przytatne funkcje i parametry: `np.choice()`

Zadanie 2. Wykorzystaj kroki spacerów losowych z utworzonej macierzy numpy do obliczenia trajektorii. Wynikiem powinna być macierz numpy, zawierająca w każdej z 30 kolumn trajektorię jednego spaceru losowego długości $n = 200$.

Przydatne funkcje: `np.cumsum()`

Zadanie 3. Narysuj wykresy dwóch funkcji $f(x) = \sqrt{2x \log\log x}$ oraz $g(x) = -f(x)$. Skala na osi X powinna przebiegać od 0 do 200. Skalę na osi Y dobierz optymalnie do wartości obu funkcji. Wykresy powinny być narysowane czarną linią przerywaną grubości 2.

Przydatne funkcje i argumenty: `plt.plot()`, `np.sqrt()`, `np.log()`, `plt.xlim()`, `plt.ylim()`, `plt.legend()`, `np.e`

Zadanie 3.5. Zastanów się, dlaczego przy tworzeniu wykresów wyświetla się (może zależeć od Twojego wyboru wartości na osi X) komunikat ostrzegawczy. Dla jakich x funkcje $f(x)$ i $g(x)$ są dobrze zdefiniowane?

Zadanie 4. Dorysuj do wykresu wykresy liniowe dla wszystkich 30 trajektorii spacerów losowych. Użyj funkcjii `plt.cm.rainbow()` do zdefiniowana 30 kolorów – jednego dla każdej trajektorii. Ustaw poziom krycia `alpha` tak, aby wykresy funkcji narysowane w poprzednim zadaniu pozostały wyraźne. Trajektorie powinny być narysowane liniami ciągłymi grubości 1.

Przydatne funkcje i argumenty: `np.linspace()`, `plt.plot()`, `plt.legend()`, `plt.show()`, `plt.cm.rainbow()`

```
# Kolory dla trajektorii definiujemy następująco:
colors = plt.cm.rainbow(np.linspace(0, 1, num_walks)) #tutaj: num_walks=30
```

Rozkład wykładniczy i gamma

Zmienna losowa X o rozkładzie wykładniczym z parametrem **intenstywności** $\lambda > 0$ ma gęstość i dystrybuantę:

$$f(x) = \lambda e^{-\lambda x}, \quad F(x) = 1 - e^{-\lambda x}, \qquad x \geq 0.$$

Zmienna losowe Y ma rozkład Gamma z parametrami **kształtu** (shape) α oraz **skali** (scale) $\beta > 0$ jeśli ma gęstość

$$f(x) = \frac{1}{\beta^\alpha \Gamma(\alpha)} x^{\alpha-1} e^{-\frac{x}{\beta}}, x \geq 0.$$

Uwaga: Czasami podawany jest rozkład z parametrami kształtu $\alpha > 0$ oraz parametrem **intensywności**, który jest odwrotnością parametru skali, tj. $\lambda = 1/\beta$, wówczas gęstość wyraża się wzorem

$$f(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\lambda x}, x \geq 0.$$

Przez rozkład Gamma(α, β) rozumiemy rozkład z parametrem **kształtu** α oraz **skali** β

Załóżmy, że zmienne losowe X i Y są niezależne i mają rozkład wykładniczy Exp(λ). Wówczas suma $Z = X + Y$ ma rozkład Gamma$(2, \lambda)$.

Zadanie 5. Wylosuj dwie stuelementowe próby `X`, `Y` z rozkładu Exp(2).

Przydatne funkcje: `numpy.random.exponential()`

Uwaga: funkcja `numpy.random.exponential()` przyjmuje parametr skali. Zatem wylosowanie Exp(λ) wygląda tak:

Zadanie 6. Użyj wektorów `X` i `Y` do stworzenia wektora `Z=X+Y`.

Zadanie 7. Narysuj histogram gęstościowy zmiennej `Z`.

Przydatne funkcje i argumenty: `np.hist()`, `plt.hist()`

Zadanie 8. Dorysuj do histogramu wykres gęstości rozkładu Gamma(2,2).

Przydatne funkcje i argumenty: `scipy.stats.gamma.pdf`

[ ]:
```
# Gamma: symulowanie zmiennych losowych, funkcja gęstości, dystrybuanta
# przy użyciu biblioteki scipy.stats:

from scipy.stats import gamma

shape=3
scale=1/5

gamma_samples = gamma.rvs(a=3, scale=scale, size=100) # probka rozm. 100
x=np.linspace(0,5,200)

gamma_pdf = gamma.pdf(x, a=shape, scale=scale)
gamma_cdf = gamma.cdf(x, a=shape, scale=scale)

plt.plot(x,gamma_pdf, label='pdf')
plt.plot(x,gamma_cdf, label='cdf')
plt.legend()
plt.show()
```