

Sprawozdanie z zadania egzaminacyjnego nr 1

Piotr Piesiak

14 kwietnia 2021

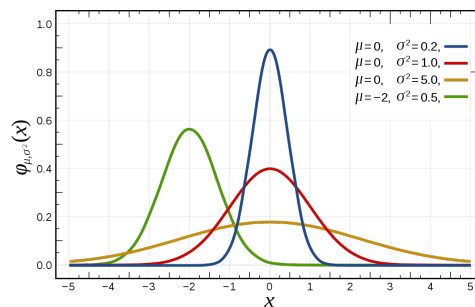
1 Rozkład normalny

Rozkład normalny możemy bardzo często zaobserwować w naturze. W statystyce używamy go do opisu wielu obserwacji takich jak inteligencja, wzrost, czy natężenie źródła światła, przez co jest jednym z najważniejszych rozkładów prawdopodobieństwa. Gęstość rozkładu normalnego przyjmuje dwa parametry ($X \sim N(\mu, \sigma^2)$):

1. μ - oznaczające średnią
2. σ^2 - oznaczające wariancję, σ jest odchyleniem standardowym

i określa się wzorem

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), x \in \mathbb{R}$$



Rysunek 1: wykres gęstości prawdopodobieństwa rozkładu normalnego dla różnych parametrów (źródło: wikipedia/Rozkład_normalny)

1.1 Standardowy rozkład normalny

Standardowy rozkład normalny jest szczególnym przypadkiem rozkładu normalnego dla $\mu = 0$ oraz $\sigma^2 = 1$, ma gęstość określoną wzorem:

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), x \in \mathbb{R}$$

W tym zadaniu zajmiemy się obliczeniem dystrybuanty standardowego rozkładu normalnego, czyli wartości funkcji:

$$\Phi(t) = \int_{-\infty}^t f(x) dx = \int_{-\infty}^t \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) dx, t \in \mathbb{R}$$

2 Obliczenie funkcji dystrybuanty

Niestety, pomimo że wiemy:

$$\int_{-\infty}^{\infty} \exp(-x^2) dx = \sqrt{\pi}$$

to całki oznaczonej z $\exp(-x^2)$ nie da się obliczyć dokładnie metodą analityczną i jesteśmy zmuszeni do korzystania z przybliżeń pochodzących z tablic (dostępnych w internecie) lub własnego programu przybliżającego całkę. My skorzystamy z własnego programu, spróbujemy więc ułatwić zadanie komputerowi i pozbyć się granicy *inf* w liczeniu całki.

2.1 Symetryczność funkcji gęstości

Zauważmy, że funkcja gęstości rozkładu normalnego jest symetryczna. Łatwo to zaobserwować na wykresie (rys. 1), lub też wywnioskować ze wzoru funkcji:

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi}} \exp\left(-\frac{x^2}{2}\right) = \frac{1}{\sqrt{2 \cdot \pi}} \exp\left(-\frac{(-x)^2}{2}\right) = f(-x)$$

Korzystając zatem z symetryczności funkcji gęstości (f) możemy zapisać:

$$\Phi(t) = \int_{-\infty}^t f(x) dx = 2 \cdot \int_0^t f(x) dx = 2 \cdot \int_0^t \frac{1}{\sqrt{2 \cdot \pi}} \exp\left(-\frac{x^2}{2}\right) dx$$

2.2 Przekształcenie funkcji dystrybuanty

Skorzystajmy teraz z własności funkcji gęstości, a mianowicie:

$$\int_{-\infty}^{\infty} f(x) dx = 1$$

korzystając z obserwacji 2.1:

$$\int_{-\infty}^{\infty} f(x) dx = 2 \cdot \int_0^{\infty} f(x) dx = 1$$

czyli

$$\int_{-\infty}^0 f(x) dx = \int_0^{\infty} f(x) dx = \frac{1}{2}$$

Po tych kilku prostych obserwacjach wróćmy do naszego problemu obliczania dystrybuanty:

$$\Phi(t) = \int_{-\infty}^t f(x) dx = \int_{-\infty}^t \frac{1}{\sqrt{2 \cdot \pi}} \exp\left(-\frac{x^2}{2}\right) dx, t \in \mathbb{R}$$

Rozpatrzmy dwa przypadki:

1. $t \geq 0$

$$\Phi(t) = \int_{-\infty}^0 f(x) dx + \int_0^t f(x) dx = \int_{-\infty}^0 \frac{1}{\sqrt{2 \cdot \pi}} \exp\left(-\frac{x^2}{2}\right) dx + \int_0^t \frac{1}{\sqrt{2 \cdot \pi}} \exp\left(-\frac{x^2}{2}\right) dx$$

czyli:

$$\Phi(t) = \frac{1}{2} + \frac{1}{\sqrt{2 \cdot \pi}} \cdot \int_0^t \exp\left(-\frac{x^2}{2}\right) dx$$

2. $t < 0$

Skorzystajmy jeszcze raz z symetryczności funkcji gęstości:

$$\int_{-\infty}^t f(x) dx \stackrel{t \leq 0}{=} \int_{-t}^{\infty} f(x) dx$$

$$\frac{1}{2} = \int_0^{\infty} f(x) dx = \int_0^{-t} f(x) dx + \int_{-t}^{\infty} f(x) dx$$

czyli wzór na dystrybuantę wygląda:

$$\Phi(t) = \frac{1}{2} - \int_0^{-t} f(x) dx = \frac{1}{2} - \frac{1}{\sqrt{2 \cdot \pi}} \cdot \int_0^{-t} \exp\left(-\frac{x^2}{2}\right) dx$$

Zauważmy, że po przekształceniu w oby przypadkach problem obliczenia dystrybuanty sprowadza się do obliczenia wartości całki $I(a)$:

$$I(a) = \int_0^a \exp\left(-\frac{x^2}{2}\right) dx$$

2.3 Szkic algorytmu obliczającego dystrybuantę

Zakładając, że potrafimy obliczyć całkę $I(a)$ możemy skonstruować algorytm, który obliczałby dystrybuantę:

```
import math
def phi(t):
    if t >= 0:
        return 1/2 + 1/(math.sqrt(2 * math.pi)) * I(t)
    else:
        return 1/2 - 1/(math.sqrt(2 * math.pi)) * I(-t)
```

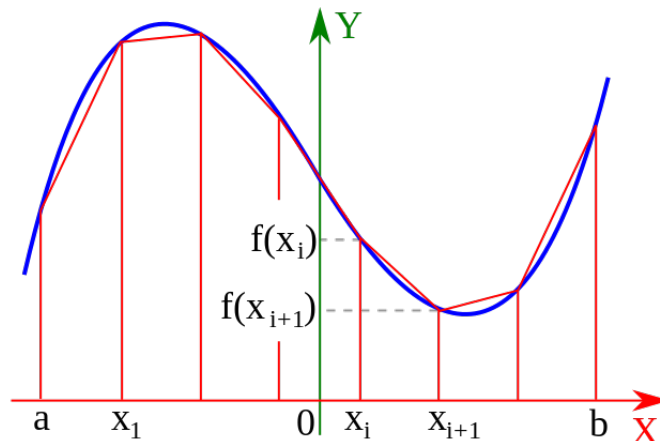
3 Metody całkowania numerycznego

3.1 Metoda złożonych trapezów

W metodzie tej dzielimy przedział całkowania (bardzo intuicyjnie) $(n + 1)$ równoodległymi punktami na podprzedziały:

$$a = x_0 < x_1 < x_2 < x_3 < \dots < x_{n-1} < b = x_n$$

i pole pod wykresem między punktami x_i, x_{i+1} przybliżamy polem trapezu tzn. $\frac{(f(x_i) + f(x_{i+1})) \cdot (x_{i+1} - x_i)}{2}$ (rys. 2)



Rysunek 2: Zobrazowanie działania metody (źródło: wikipedia/Wzór_trapezów)

Ostatecznie otrzymujemy wzór:

$$\begin{aligned} \int_a^b f(x) dx &\approx \sum_{i=0}^{n-1} \frac{(f(x_i) + f(x_{i+1})) \cdot (x_{i+1} - x_i)}{2} = \sum_{i=0}^{n-1} \frac{(f(x_i) + f(x_{i+1})) \cdot (\frac{b-a}{n})}{2} = \\ &= \frac{b-a}{2 \cdot n} \cdot (f(a) + 2 \cdot f(x_1) + 2 \cdot f(x_2) + \dots + 2 \cdot f(x_{n-1}) + f(b)) \end{aligned}$$

Prosta implementacja tej metody wygląda:

```
def compositeTrapezoid(n,a,b):
    h = (b-a)/n
    sum = 0
    for i in range(1,n):
        sum += f(a + i*h)
    return (1/2)*h*(f(a) + 2*sum + f(b))
```

3.2 Metoda Romberga

W tej metodzie dzielimy przedział całkowania (a, b) na 2^n równych części:

$$a = x_0 < x_1 < x_2 < x_3 < \dots < x_{2^n-1} < b = x_{2^n}$$

Niech $T(n)$ oznacza wynik złożonego wzoru trapezów dla n przedziałów. Możemy przedstawić metodę Romberga rekurencyjnie:

$$\begin{cases} R_{0,i} & : T(2^n) \\ R_{m,i} & : \frac{4^m \cdot R_{m-1,i+1} - R_{m-1,i}}{4^m - 1} \end{cases}$$

Sprowadza się to do obliczenia współczynników tablicy Romberga, czyli wyznaczenia rekurencyjnie kolejnych kolum. Otrzymamy w ten sposób coraz lepsze przybliżenie funkcji:

$R_{0,0}$
 $R_{0,1} R_{1,0}$
 $R_{0,2} R_{1,1} R_{2,0}$
 $R_{0,3} R_{1,2} R_{2,1} R_{3,0}$
 \dots

Implementacja wykorzystująca tę rekurencję wygląda następująco: (procedura RombergsMethod zwraca kolejne wiersze tablicy Romberga - oszczędzając w ten sposób czas i pamięć):

```
def RombergsMethod(n,a,b, Rombergs_tab):
    new_R_tab = [compositeTrapezoid(n,a,b)]
    for r in range(len(Rombergs_tab)):
        new_R_tab.append(1/(4**(r+1) - 1) * ((4**(r+1)) * (new_R_tab[r]) - Rombergs_tab[r]))
    Rombergs_tab = new_R_tab
    return Rombergs_tab
```

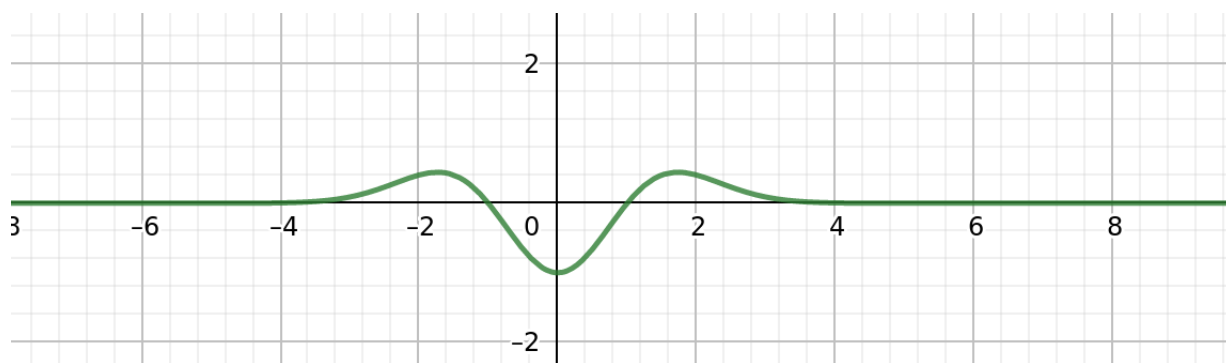
3.3 Analiza błędu

W przypadku metody złożonych trapezów, funkcja błędu wynosi:

$$E(N) = -\frac{(b-a)^3}{12N^2} \cdot f''(\xi), \xi \in [a, b]$$

$$f''(\xi) = \left(\exp\left(-\frac{\xi^2}{2}\right) \right)'' = (\xi^2 - 1) \cdot \exp\left(-\frac{\xi^2}{2}\right)$$

Zobaczmy, że dla większych ξ funkcja wykładnicza zawsze będzie przeważać i zbliżać do 0 wartość f'' , zatem wystarczy popatrzeć na zachowanie funkcji w okolicach 0:



Rysunek 3: Wykres drugiej pochodnej (wygenerowany przez geogebra.org)

Widać, że maksimum funkcji jest na pewno mniejsze niż 1, zatem funkcję błędu możemy oszacować przez $(N - \text{liczba podprzedziałów}, a, b - \text{krańce przedziału})$:

$$|E(N)| \leq \frac{(b-a)^3}{12N^2}$$

Pozostaje nam rozwiązać nierówność (względem N):

$$\frac{(b-a)^3}{12N^2} \leq 10^{-8}$$

$$N \geq \sqrt{\frac{(b-a)^3 \cdot 10^8}{12}}$$

Zatem, wystarczy sprawdzić na wejściu programu, ile podprzedziałów potrzebujemy aby uzyskać zadaną dokładność. W przypadku metody Romberga (wiemy z Analizy Numerycznej że działa ona mniej więcej 2 razy lepiej) możemy podzielić na tyle samo podprzedziałów, aby mieć pewność, że metoda zwróci nam wynik z zadaną dokładnością. Na przykład dla $(b-a) = 1$, potrzebujemy $N \geq 10^4$ podprzedziałów.

4 Program i wyniki eksperymentów

Pełny program, który wykorzystuje wszystkie powyższe obserwacje znajduje się w pliku `CDF_standard_ND.py`. Oto kilka wyników eksperymentów:

Funkcja I liczy wartość całki $I(a) = \int_0^a \exp\left(-\frac{x^2}{2}\right) dx$, natomiast Φ liczy wartość dystrybuanty:

$$\Phi(t) = \int_{-\infty}^t f(x) dx = \int_{-\infty}^t \frac{1}{\sqrt{2 \cdot \pi}} \exp\left(-\frac{x^2}{2}\right) dx, t \in \mathbb{R}$$

Kilka wyników dla numerycznego liczenia wartości całki:

I(0.2)

```
Minimum N required for given error: 512
Composite Trapezoid for 512 subsets: 0.19867462622631904
Romberg's method for 512 subsets: 0.19867462871909305

Composite Trapezoid for 1024 subsets: 0.19867462622631904
Romberg's method for 1024 subsets: 0.19867462871909328
Composite Trapezoid for 2048 subsets: 0.19867462622631904
Romberg's method for 2048 subsets: 0.19867462871909292
Composite Trapezoid for 4096 subsets: 0.19867462622631904
Romberg's method for 4096 subsets: 0.19867462871909317
Composite Trapezoid for 8192 subsets: 0.19867462622631904
Romberg's method for 8192 subsets: 0.19867462871909328
Composite Trapezoid for 16384 subsets: 0.19867462622631904
Romberg's method for 16384 subsets: 0.19867462871909214
```

I(2)

```
Minimum N required for given error: 8192
Composite Trapezoid for 8192 subsets: 1.1962880119781758
Romberg's method for 8192 subsets: 1.1962880133226144

Composite Trapezoid for 1024 subsets: 1.1962880119781758
Romberg's method for 1024 subsets: 1.1962880133226106
Composite Trapezoid for 2048 subsets: 1.1962880119781758
Romberg's method for 2048 subsets: 1.196288013322609
Composite Trapezoid for 4096 subsets: 1.1962880119781758
Romberg's method for 4096 subsets: 1.1962880133226013
Composite Trapezoid for 8192 subsets: 1.1962880119781758
Romberg's method for 8192 subsets: 1.1962880133226144
Composite Trapezoid for 16384 subsets: 1.1962880119781758
Romberg's method for 16384 subsets: 1.1962880133226068
```

Możemy zauważyć, że metoda Romberga jest dokładniejsza i szybciej zbiega do dokładnego wyniku.

Zaaplikujmy zatem metodę Romberga do wyliczenia naszej dystrybuanty:

```
phi(-2)
0.02275013194817671
```

```
phi(1.2)
0.8849303297782914
```

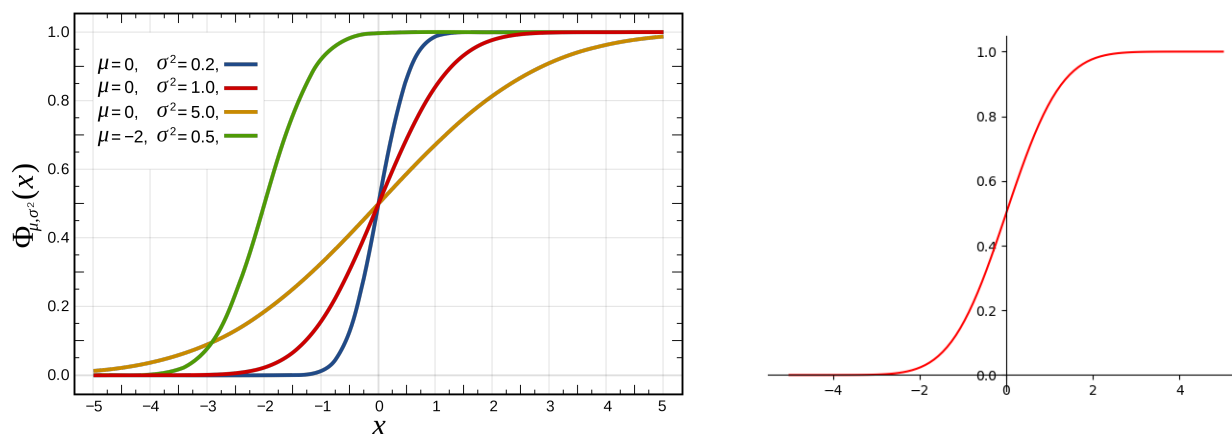
```
phi(0.2)
0.579259709439103
```

```
phi(3)
0.9986501019683678
```

```
phi(5)
0.999997133484154
```

```
phi(8)
0.999999999999609
```

Zgodnie z oczekiwaniami wartości dystrybuanty oddalone o kilka sigm są coraz bliżej jedynki lub zera. Dodatkowo możemy sprawdzić, że wykres wygenerowany przez naszą dystrybuantę jest poprawny:



Rysunek 4: Wykresy dystrybuanty rozkładu normalnego, po lewej widać wzorcowe wykresy z Wikipedii, po prawej wygenerowany przez nasz program dla funkcji Phi.