

# Rachunek Prawdopodobieństwa i Statystyka

Rozwiązanie zadania z kolokwium nr 1

Michał Mikołajczyk, Wojciech Adamiec

19 kwietnia 2020

## Treść zadania:

Standardowy rozkład normalny ma gęstość określoną wzorem:

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

Dla dystrybuanty otrzymujemy wyrażenie:

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

która to całka nie ma przedstawienia za pomocą funkcji elementarnych. Ponieważ gęstość jest funkcją parzystą zatem zachodzi związek:

$$\Phi(t) = 1 - \Phi(-t) \quad (3)$$

Redukujemy zatem to zadanie do obliczenia wartości poniższej całki dla ustalonego  $t > 0$ :

$$G(t) = \int_0^t \exp\left(\frac{-x^2}{2}\right) dx. \quad (4)$$

## Rozwiązanie:

1. *Dowód (3):*

Zauważmy, że:

$$\begin{aligned} \int_{-\infty}^{\infty} f(x) dx &= 1, \\ f(x) &= f(-x). \end{aligned} \quad (5)$$

Wtedy:

$$\begin{aligned} \Phi(t) &= \int_{-\infty}^t f(x) dx \\ &= \int_{-\infty}^{\infty} f(x) dx - \int_t^{\infty} f(x) dx \\ &\stackrel{(5)}{=} 1 - \int_t^{\infty} f(x) dx. \end{aligned} \quad (6)$$

Skoro  $f(x) = f(-x)$ , to:

$$\int_t^{\infty} f(x) dx = \int_{-\infty}^{-t} f(x) dx = \Phi(-t). \quad (7)$$

Zatem  $\Phi(t) = 1 - \Phi(-t)$ .



2. Metoda obliczania całki  $G(t) - (4)$ :

Korzystamy z metody Romberga. Nasze rozwiązanie bazowaliśmy na wersji znajdującej się na stronie wikipedii – [Romberg's method](#).

3. Obliczanie  $\Phi(t)$  na podstawie  $G(t)$ :

(a) Jeśli  $t < 0$  wówczas naszym wynikiem będzie:  $1 - \Phi(-t)$

(b) Dla  $t \geq 0$  mamy  $\Phi(t) = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \cdot G(t)$ , gdzie  $\frac{1}{2}$  to pole pod wykresem naszego rozkładu w przedziale  $(-\infty, 0]$ .

### Informacje techniczne:

Do dokumentacji jest dołączony program napisany w języku *Python* obliczający zarówno  $\Phi(t)$  jak i samo  $G(t)$ . Dodatkowo dla wygody dołączamy również dokument w formacie *jupyter* z tymi samymi programami oraz z kilkoma przykładami.

### Kod źródłowy:

#### Metoda Romberga

```
def romberg(f, a, b, max_steps, acc):
    Rp = [0 for _ in range(max_steps)] # Previous row
    Rc = [0 for _ in range(max_steps)] # Current row
    h = (b-a) # Step size
    Rp[0] = (f(a) + f(b)) * h * 0.5 # First trapezoidal step

    for i in range(1, max_steps):
        h /= 2
        c = 0
        ep = 1 << (i-1) # 2^(n-1)

        for j in range(1, ep+1):
            c += f(a+(2*j-1)*h)

        Rc[0] = h*c + 0.5 * Rp[0] # R(i,0)

        for j in range(1, i+1):
            n_k = 4**j
            Rc[j] = (n_k * Rc[j-1] - Rp[j-1]) / (n_k-1) # compute R(i,j)

        if i > 1 and abs(Rp[i-1] - Rc[i]) < acc:
            return Rc[i-1]

    # swap Rn and Rc as we only need the last row
    rt = Rp
    Rp = Rc
    Rc = rt

    return Rp[max_steps-1] # return our best guess
```

Definicja funkcji  $G(t)$

```
from math import exp
def G(t):
    def internal(x):
        return exp(-(x**2) / 2)
    return romberg(internal, 0, t, 1000, 0.00001)
```

Definicja funkcji  $\Phi(t)$

```
from math import sqrt, pi
def phi(t):
    if t < 0:
        return 1 - phi(-t)
    return 0.5 + 1/sqrt(2 * pi) * G(t)
```