Bootcamp Data Science
Zajęcia 4

Przemysław Spurek

Liczby zespolone

$\mathbb{C} = \mathbb{R} \times \mathbb{R}$ - zbiór liczb zespolonych z działaniami:

- $\forall (a, b), (c, d) \in \mathbb{C}\colon (a, b) + (c, d) = (a + c, b + d)$,
- $\forall (a, b), (c, d) \in \mathbb{C}\colon (ac - bd, ad + bc)$.

Liczby zespolone możemy zatem zapisywać także w postaci:

- $i = \sqrt{-1}$ - jednostka urojona
- $\forall (a, b) \in \mathbb{C}\colon (a, b) = a + bi$

Liczby zespolone

Kilka specjalnych funkcji oraz sposobów reprezentacji:

- $\mathrm{Re}(a+bi) = a$ - **część rzeczywista**,
- $\mathrm{Im}(a+bi) = b$ - **część urojona**,
- $\overline{a+bi} = a - bi$ - **sprzężenie** liczby zespolonej,
- $|a+bi| = \sqrt{a^2+b^2}$ - **moduł** liczby zespolonej (odległość od zera $(0,0)$) lub inaczej $|z| = \sqrt{(\mathrm{Re}\, z)^2 + (\mathrm{Im}\, z)^2}$,
- $\phi = \begin{cases} \arccos \frac{a}{|z|}, & \text{jeżeli } b \geq 0 \text{ i } z \neq (0,0) \\ -\arccos \frac{a}{|z|}, & \text{jezeli } b < 0 \text{ i } z \neq (0,0) \end{cases}$, to **argument główny** liczby zespolonej ($\phi \in [0, 2\pi]$).

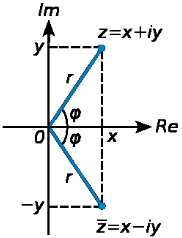
Liczby zespolone

- Liczba zespolona $z \neq (0,0)$ może być przedstawiona jako:

$$z = |z|(\cos\phi + i \sin\phi), \tag{1}$$

gdzie:

- $|z|$, to **moduł** liczby zespolonej,
- ϕ, to argument główny liczby zespolonej,



Liczby zespolone

- Liczba zespolona $z \in \mathbb{C}$ może być przedstawiona jako:

$$z = |z| \cdot e^{i\phi}$$

gdzie:

- $|z|$, to **moduł** liczby zespolonej,
- ϕ, to argument główny liczby zespolonej,

Stąd na podstawie układu równań

$$\begin{cases} e^{i\phi} & = \cos\phi + i\sin\phi \\ e^{-i\phi} & = \cos\phi - i\sin\phi \end{cases}$$

otrzymujemy $\sin\phi = \frac{e^{i\phi} - e^{-i\phi}}{2i}$ oraz $\cos\phi = \frac{e^{i\phi} + e^{-i\phi}}{2}$.

Własności fal

Własności fal

- amplituda – maksymalne odchylenie A od położenia równowagi

Własności fal

- amplituda – maksymalne odchylenie A od położenia równowagi
- długość fali – odległość λ pomiędzy kolejnymi powtórzeniami kształtu fali (np. grzbiety, doliny)

Własności fal

- amplituda – maksymalne odchylenie A od położenia równowagi
- długość fali – odległość λ pomiędzy kolejnymi powtórzeniami kształtu fali (np. grzbiety, doliny)
- okres – odstęp czasu T między momentami, gdy grzbiety (doliny) dwóch sąsiadujących fal przechodzą przez ten sam punkt.

https://github.com/przem85/statistics/blob/master/D16/D16_Z5.ipynb

Szereg Fouriera

Niech dana będzie funkcja okresowa $f : \mathbb{R} \to \mathbb{R}$ o okresie $T \in \mathbb{R}^+$, bezwzględnie całkowalna w przedziale $\left[\frac{-T}{2}, \frac{T}{2}\right]$.
Trygonometrycznym szeregiem Fouriera funkcji f nazywamy szereg funkcyjny następującej postaci:

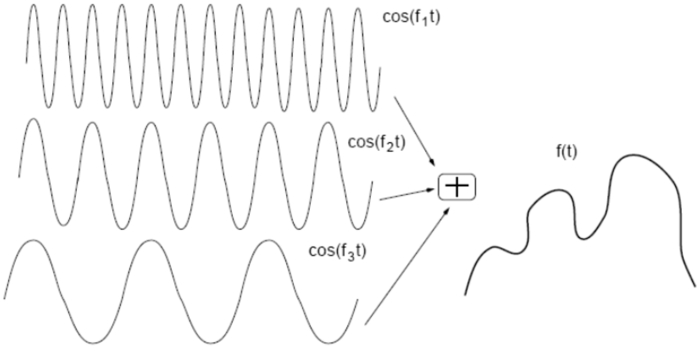
$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos\left(\frac{2n\pi}{T}x\right) + b_n \sin\left(\frac{2n\pi}{T}x\right)\right) \quad (1.1)$$

O współczynnikach określonych następującymi wzorami:

$$a_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \cos\left(\frac{2n\pi}{T}x\right) dx, \qquad n = 0, 1, 2, \ldots$$

$$b_n = \frac{2}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} f(x) \sin\left(\frac{2n\pi}{T}x\right) dx, \qquad n = 1, 2, 3, \ldots$$

Transformata Fouriera



https:
//github.com/przem85/statistics/blob/master/D16/D16_Z1.ipynb

Transformata Fouriera

Transformacja Fouriera rozkłada funkcję okresową na szereg funkcji okresowych tak, że uzyskana transformata podaje w jaki sposób poszczególne częstotliwości składają się na pierwotną funkcję.

Transformata Fouriera

Definition

Dla funkcji $f(x)$ **transformatę Fouriera** definiujemy następująco:

$$F(f(x)) = F(s) = \int f(x) \cdot e^{-2\pi ixs} dx.$$

Definition

Dla funkcji $f(x)$ **transformatę Fouriera** definiujemy następująco:

$$F(f(x)) = F(s) = \int f(x) \cdot e^{-2\pi ixs} dx.$$

Zauważmy, że przy tak postawionej definicji możemy $f(x)$ zapisać w następujący sposób:

$$F^{-1}(F(s)) = f(x) = \int F(s) \cdot e^{2\pi ixs} ds.$$

gdzie

$$e^{\pm\alpha} = \cos(\alpha) \pm i \sin(\alpha)$$

Transformata Fouriera

W praktyce często zmienna x oznacza czas (w sekundach), a argument transformaty s oznacza częstotliwość (w $Hz = \frac{1}{s}$).

Zauważmy,że:

$$f(x) = E(x) + O(x),$$

gdzie $E(x)$ jest pewną funkcją parzystą zmiennej x, natomiast $O(x)$ jest pewną funkcją nieparzystą.

Wtedy transformata Fouriera funkcji f redukuje się do postaci:

$$2\int_0^\infty E(x)\cdot\cos(2\pi xs)dx - 2i\int_0^\infty O(x)\sin(2\pi xs)dx$$

Transformata Fouriera

Stąd łatwo widać, że jeżeli funkcja jest parzysta, to jej transformata jest parzysta, a jeżeli funkcja jest nieparzysta, to jej transformata jest również nieparzysta.

$$2\int_0^\infty E(x)\cdot\cos(2\pi xs)dx - 2i\int_0^\infty O(x)\sin(2\pi xs)dx$$

Transformata Fouriera

Ponieważ w praktyce w wyniku pomiarów otrzymujemy dane o charakterze dyskretnym, a nie ciągłym, konieczne jest zdefiniowanie dyskretnego odpowiednika ciągłej transformaty Fouriera (zastępuje się całkę poprzez sumę):

Definition

Dla N-elementowego ciągu x_n **dyskretną transformatę Fouriera** definiujemy następująco:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}, \quad k = 0, 1, \ldots, N-1$$

Liczenie DFT z definicji:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}, \quad k = 0, 1, \ldots, N-1$$

gdzie $x_0, \ldots, x_{N-1}$ to próbki sygnału, wymagało (w latach 60-tych) tak ogromnych mocy obliczeniowych, że maszyny z tego okresu ograniczały użycie tego algorytmu.

https://github.com/przem85/statistics/blob/master/D16/D16_Z2.ipynb

Rok 1965 przyniósł rewolucję. J. Cooley i J. Tuckey opublikowali pracę pod tytułem "An Algorithm for the machine computation of complex Fourier series", w której opracowali szybszy algorytm liczenia dyskretnej transformaty Fouriera powszechnie znany jako szybka transformata Fouriera (ang. FFT – Fast Fourier Transform).

FFT jest to DFT ze zmniejszoną liczbą niezbędnych operacji arytmetycznych. Celem FFT jest zmniejszenie długiego algorytmu obliczeniowego przez jego podział na krótsze i prostsze obliczenia DFT i skrócenie czasu obliczeń. Istnieją różne algorytmy FFT.

Transformata Fouriera

Sama idea algorytmu opiera się na tzw. lemacie Danielsona-Lanczosa. Odkryli oni, że pojedyńcza DFT o długości N, jest równoważna sumie dwóch transformat o długości $N/2$, jedna z nich jest złożona z nieparzystych punków spośród oryginalnych N, a druga z parzystych.

$$X_k = \underbrace{\sum_{m=0}^{N/2-1} x_{2m} e^{-\frac{2\pi i}{N/2} mk}}_{\text{DFT of even-indexed part of } x_m} + e^{-\frac{2\pi i}{N} k} \underbrace{\sum_{m=0}^{N/2-1} x_{2m+1} e^{-\frac{2\pi i}{N/2} mk}}_{\text{DFT of odd-indexed part of } x_m} =$$

$$= E_k + e^{-\frac{2\pi i}{N} k} O_k.$$

http://en.wikipedia.org/wiki/Cooley%E2%80%93Tukey_FFT_algorithm

Zadania - filtrowanie sztucznego sygnału

https://github.com/przem85/statistics/blob/master/D16/D16_Z3.ipynb

https://github.com/przem85/statistics/blob/master/D16/D16_Z4.ipynb

https://github.com/przem85/statistics/blob/master/D16/D16_Z4.ipynb

Zadania - filtrowanie dźwięku

https://github.com/przem85/statistics/blob/master/D16/D16_Z6.ipynb