

Programowanie i metody numeryczne:

Zagadnienie własne (Eigenproblem)

Rozkład SVD

Dyskretna Transformata Fouriera (i jej podobne)

Jędrzej Wardyn

06.06.2024

Types of Matrices

- ▶ Symmetric matrix: $A = A^T$
- ▶ Hermitian matrix: $A = A^\dagger$
- ▶ Unitary matrix: $UU^\dagger = U^\dagger U = I$
- ▶ Orthogonal matrix: $A^T A = A A^T = I$
- ▶ Diagonal matrix: A is diagonal if $A_{ij} = 0$ for $i \neq j$

Eigenproblem (równanie własne)

$$A\mathbf{x}_L = \lambda_L\mathbf{x}_L$$

$$\mathbf{x}_R A = \lambda_R\mathbf{x}_R$$

dla $A = A^T$ zbiory wartości własnych są takie same :

$$\{\lambda_L\} = \{\lambda_R\}$$

Dla ogólnej macierzy, jeśli zbierzemy wszystkie wektory własne do macierzy $\mathbf{X}_L, \mathbf{X}_R$, otrzymamy:

$$(\mathbf{X}_L \cdot \mathbf{X}_R) \cdot \text{diag}(\lambda_1 \dots \lambda_N) = \text{diag}(\lambda_1 \dots \lambda_N) \cdot (\mathbf{X}_L \cdot \mathbf{X}_R)$$

Diagnoalizacja macierzy jako transformacja podobieństwa

Możemy przedstawić zdiagonalizowaną macierz jako:

$$\text{diag}(\lambda_1 \dots \lambda_N) = \mathbf{X}_R^{-1} \cdot \mathbf{A} \cdot \mathbf{X}_R$$

Co jest szczególnym przypadkiem dla transformacji podobieństwa:

$$\mathbf{A} \rightarrow \mathbf{Z}^{-1} \cdot \mathbf{A} \cdot \mathbf{Z}$$

$$\begin{aligned} \det |\mathbf{Z}^{-1} \cdot \mathbf{A} \cdot \mathbf{Z} - \lambda \mathbf{1}| &= \det |\mathbf{Z}^{-1} \cdot (\mathbf{A} - \lambda \mathbf{1}) \cdot \mathbf{Z}| \\ &= \det |\mathbf{Z}| \det |\mathbf{A} - \lambda \mathbf{1}| \det |\mathbf{Z}^{-1}| \\ &= \det |\mathbf{A} - \lambda \mathbf{1}| \end{aligned}$$

Strategia występująca w większości metod
diagonalizacyjnych:

$$\mathbf{A} \rightarrow \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1$$

Strategia występująca w większości metod
diagonalizacyjnych:

$$\mathbf{A} \rightarrow \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \rightarrow \mathbf{P}_2^{-1} \cdot \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \cdot \mathbf{P}_2$$

Strategia występująca w większości metod diagonalizacyjnych:

$$\begin{aligned} \mathbf{A} &\rightarrow \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \rightarrow \mathbf{P}_2^{-1} \cdot \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \cdot \mathbf{P}_2 \\ &\rightarrow \mathbf{P}_3^{-1} \cdot \mathbf{P}_2^{-1} \cdot \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \mathbf{P}_3 \rightarrow \text{etc.} \end{aligned}$$

Stosujemy kolejno transformacje tak długo aż nie otrzymamy odpowiedniego:

$$X_R = \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \mathbf{P}_3 \cdot \dots$$

Strategia występująca w większości metod diagonalizacyjnych:

$$\begin{aligned} \mathbf{A} &\rightarrow \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \rightarrow \mathbf{P}_2^{-1} \cdot \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \cdot \mathbf{P}_2 \\ &\rightarrow \mathbf{P}_3^{-1} \cdot \mathbf{P}_2^{-1} \cdot \mathbf{P}_1^{-1} \cdot \mathbf{A} \cdot \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \mathbf{P}_3 \rightarrow \text{etc.} \end{aligned}$$

Stosujemy kolejno transformacje tak długo aż nie otrzymamy odpowiedniego:

$$\mathbf{X}_R = \mathbf{P}_1 \cdot \mathbf{P}_2 \cdot \mathbf{P}_3 \cdot \dots$$

Głównymi metodami transformacji są:

1. Otrzymuje \mathbf{P}_i (Jacobi, Hausholder)
2. Metody faktoryzacji: rozdziela $\mathbf{A} = \mathbf{F}_L \cdot \mathbf{F}_R$ (Metoda QR)

W ogólności najlepiej stosować obie w zależności od przypadku

Macierze niehermitowskie: Metoda Arnoldi do wartości własnych

Wykorzystujemy macierz Krylova:

$$K_n = [b \quad Ab \quad A^2b \quad \dots \quad A^{n-1}b].$$

A następnie stosujemy ortogonalizację Grama-Schmidta uzyskując wektory, które mogą przybliżyć nam wektory własne.

Dla problemów wykraczających poza systemy izolowane:

$$\dot{\rho} = -\frac{i}{\hbar}[H, \rho] + \sum_i \gamma_i \left(L_i \rho L_i^\dagger - \frac{1}{2} \{L_i^\dagger L_i, \rho\} \right),$$

gdzie L_i -operatory skoku

Dla macierzy hermitowskich metoda Arnoldi jest tożsama z metodą Lanczos.

Uogólnione zagadnienie własne $\mathbf{A} \cdot \mathbf{x} = \lambda \mathbf{B} \cdot \mathbf{x}$

Gdy \mathbf{B} jest odwracalna (nieosobliwa):

$$(\mathbf{B}^{-1} \cdot \mathbf{A}) \cdot \mathbf{x} = \lambda \mathbf{x}$$

Zwykle \mathbf{A} and \mathbf{B} symetryczne, a \mathbf{B} jest określona dodatnio.

$\mathbf{B}^{-1} \cdot \mathbf{A}$ jest niesymetryczna, ale możemy zastosować rozkład

Choleskiego $\mathbf{B} = \mathbf{L} \cdot \mathbf{L}^T$. Mnożąc przez \mathbf{L}^{-1} dostajemy:

$$\mathbf{C} \cdot (\mathbf{L}^T \cdot \mathbf{x}) = \lambda (\mathbf{L}^T \cdot \mathbf{x}) \quad \mathbf{C} = \mathbf{L}^{-1} \cdot \mathbf{A} \cdot (\mathbf{L}^{-1})^T$$

Jeśli \mathbf{C} symetryczna to jej funkcje własne: $\mathbf{L}^T \cdot \mathbf{x}$. Najlepiej jest rozwiązać:

$$\mathbf{Y} \cdot \mathbf{L}^T = \mathbf{A}$$

Dla dolnego trójkąta symetrycznej macierzy \mathbf{C}

Układy równań różniczkowych

$$\mathbf{x}' = A\mathbf{x}, \quad \mathbf{x} = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \quad \mathbf{x}' = \begin{pmatrix} x'_1(t) \\ x'_2(t) \\ \vdots \\ x'_n(t) \end{pmatrix}$$

Jeśli macierz A jest diagonalna ($a_{ij} = 0$ gdy $i \neq j$), wtedy układ redukuje się do:

$$x'_1 = a_{11}x_1 \quad x'_2 = a_{22}x_2 \quad \dots \quad x'_n = a_{nn}x_n$$

A rozwiązaniem jest trywialnie:

$$x_1 = k_1 e^{a_{11}t} \quad x_2 = k_2 e^{a_{22}t} \quad \dots \quad x_n = k_n e^{a_{nn}t}.$$

Układy równań różniczkowych

$$\mathbf{x}' = A\mathbf{x}, \quad \mathbf{x} = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \quad \mathbf{x}' = \begin{pmatrix} x'_1(t) \\ x'_2(t) \\ \vdots \\ x'_n(t) \end{pmatrix}$$

Ogólniejsze A (diagonalizowalne), mamy: $D = M^{-1}AM$,
where M – macierz modalna dla A . Używając $A = MDM^{-1}$,
sprowadzamy problem do postaci

$$M^{-1}\mathbf{x}' = D(M^{-1}\mathbf{x}) \text{ lub } \mathbf{y}' = D\mathbf{y}, \quad \mathbf{x} = M\mathbf{y}.$$

Rozwiązaniem jest wtedy:

$$y_1 = k_1 e^{\lambda_1 t} \quad y_2 = k_2 e^{\lambda_2 t} \quad \vdots \quad y_n = k_n e^{\lambda_n t}$$

Macierz modalna M (przykład)

$$J = M^{-1}AM$$

$$A = \begin{pmatrix} -1 & 0 & -1 & 1 & 1 & 3 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 2 & -1 & -1 & -6 & 0 \\ -2 & 0 & -1 & 2 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ -1 & -1 & 0 & 1 & 2 & 4 & 1 \end{pmatrix}$$

$$M = \begin{pmatrix} 0 & 1 & -1 & 0 & 0 & -2 & 1 \\ 0 & 3 & 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & 1 & 0 & 2 & 0 \\ -2 & 0 & -1 & 0 & 0 & -2 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & -1 & 0 \end{pmatrix}$$

$$J = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix},$$

Układy równań różniczkowych

$$\mathbf{x}' = A\mathbf{x}, \quad \mathbf{x} = \begin{pmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{pmatrix}, \quad \mathbf{x}' = \begin{pmatrix} x'_1(t) \\ x'_2(t) \\ \vdots \\ x'_n(t) \end{pmatrix}$$

Gdy A nie jest diagonalizowalna wybieramy takie M takie, że $J = M^{-1}AM$, gdzie J - Macierz Jordana od A

Otrzymujemy układ $\mathbf{y}' = J\mathbf{y}$:

$$y'_1 = \lambda_1 y_1 + \epsilon_1 y_2 \quad y'_{n-1} = \lambda_{n-1} y_{n-1} + \epsilon_{n-1} y_n \quad \dots \quad y'_n = \lambda_n y_n$$

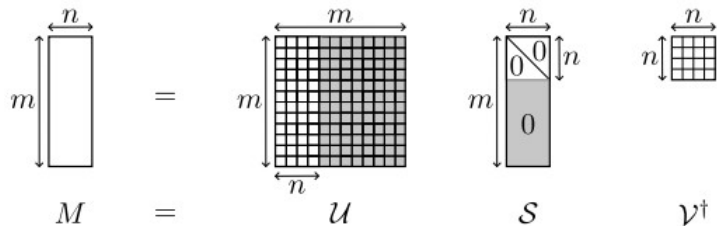
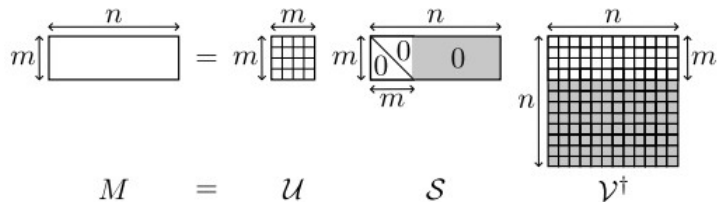
gdzie mamy λ_i – wartości własne, ϵ_i – zera i jedynki z naddiagonali J .

Przykład SVD

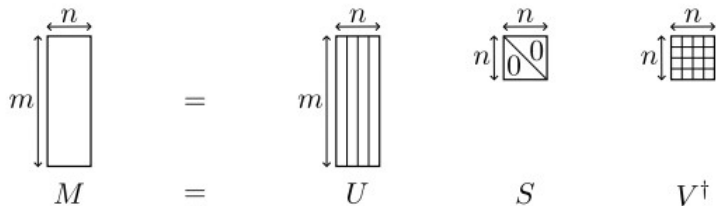
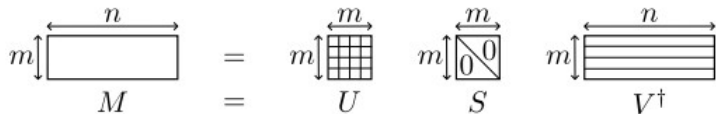
$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{bmatrix} = U\Sigma V^T \quad \Sigma = \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$U = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad V = \begin{bmatrix} 0 & 0 & \frac{\sqrt{5}}{5} & 0 & \frac{-2\sqrt{5}}{5} \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{2\sqrt{5}}{5} & 0 & \frac{\sqrt{5}}{5} \end{bmatrix}$$

Schemat SVD (pełne)



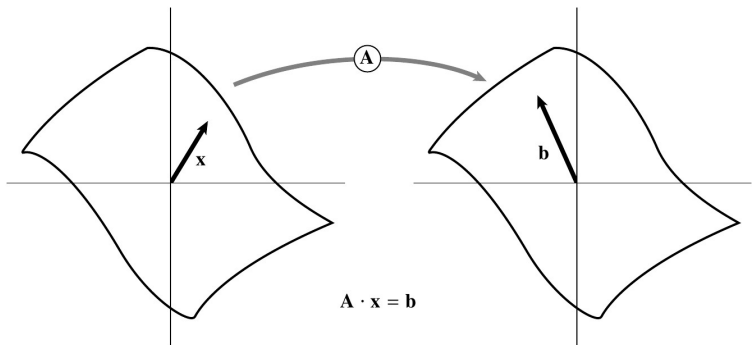
Schemat SVD (zredukowane)



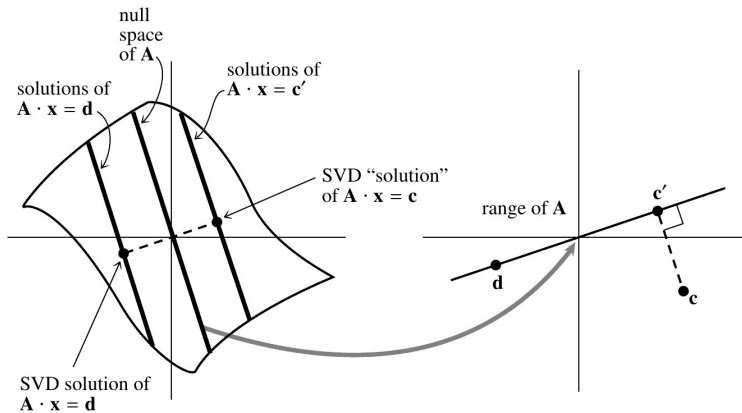
Czym graficznie są wektory własne?

Graficzne przedstawienie wektorów własnych od 3Blue1Brown

Co pokazuje nam rozkład SVD?



Co pokazuje nam rozkład SVD?



Układy z mniejszą liczbą równań niż niewiadomych (niedookreślone)

Gdy mamy N szukanych oraz M równań to możemy nie dostać unikalnego rozwiązania, tylko rodzinę rozwiązań w liczbie $N - M$. Rozwiązania mogą być dość bliskie zeru lub mogą być zerami i trzeba na to uważać [w sensie numerycznym].

Układy z większą liczbą równań niż niewiadomych (nadokreślone)

Na podstawie serii wyników równoważnych pomiarów, w naukach przyrodniczych za **ocenę prawdziwej**, często określanej też mianem: **dokładnej**, wartości wielkości fizycznej, którą mierzymy, przyjmuje się **średnią arytmetyczną** \bar{x} , którą dla serii N wartości x_i , $i = 1, 2, \dots, N$, definiujemy jako:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i,$$

za miarę zaś rozrzutu uzyskanych wartości, czyli szerokości rozkładu, przyjmujemy wielkość s_x , której kwadrat jest proporcjonalny do minimalnej sumy kwadratów reszt:

$$s_x^2 \propto \sum_{i=1}^N (x_i - \bar{x})^2.$$

Układy z większą liczbą równań niż niewiadomych (nadokreślone)

$$\begin{pmatrix} \mathbf{A} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \end{pmatrix} \quad (2.6.11)$$

The proofs that we gave above for the square case apply without modification to the case of more equations than unknowns. The least-squares solution vector \mathbf{x} is

$$\begin{pmatrix} \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{V} \end{pmatrix} \cdot \begin{pmatrix} \text{diag}(1/w_j) \end{pmatrix} \cdot \begin{pmatrix} \mathbf{U}^T \end{pmatrix} \cdot \begin{pmatrix} \mathbf{b} \end{pmatrix}$$

Kompresja obrazu

Kompresja obrazu za pomocą SVD

Podstawowe biblioteki

Fortran/C++:

- ▶ **LAPACK** (Linear Algebra Package)
- ▶ **BLAS** (Basic Linear Algebra Subprograms)
- ▶ **Eigen** (C++) (łatwa w obsłudze ale ograniczona)
- ▶ **ARPACK** (do metody Arnoldi macierzy rzadkich)

Python:

- ▶ **SciPy**: (LAPACK i BLAS)
- ▶ **NumPy**

Do obliczeń tensorowych lista bibliotek:

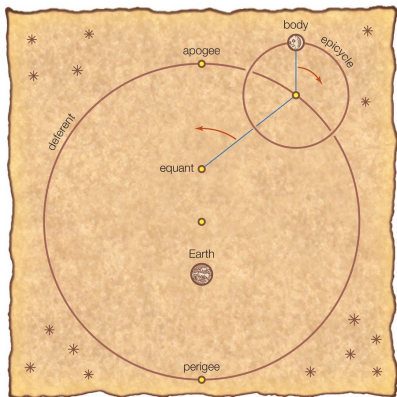
<http://tensornetwork.org/software/>

Transformata Fouriera: wstępnie

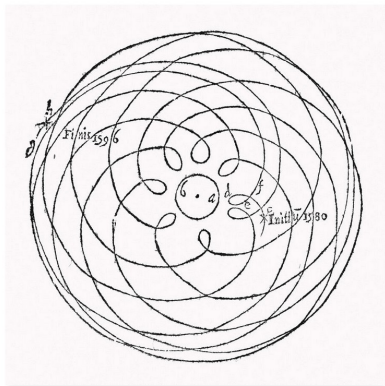
Starożytna Transformata Fouriera

Co to może być? Proszę zgadnąć

Starożytna Transformata Fouriera: Epicykle!



© 2012 Encyclopædia Britannica, Inc.



Transformata Fouriera

Ograniczenia w skończonej ilości okręgów: zjawisko Gibbsa

Możemy narysować dowolną krzywą w 2D ruchami po okręgach

Link do kodu do wyrysowania dowolnych krzywych

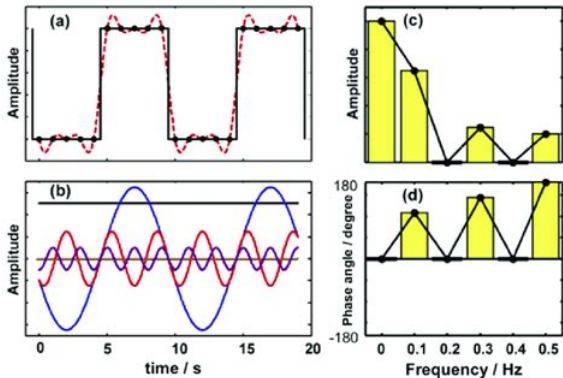
Veritasium na temat początków transformaty Fouriera: analogowy komputer

Dyskretna transformata Fouriera

Transformata Fouriera ciągła i dyskretna

Transformata Fouriera przelicza nam funkcję z dziedziny czasu (ogólnie jakiejś zmiennej) do dziedziny częstotliwości.

Dyskretna wersja robi to dla skończonego zbioru częstotliwości.



Transformata Fouriera: wzór

Szereg fouriera dany jest przez:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi \frac{n}{P}x} \quad c_n = \frac{1}{P} \int_{-P/2}^{P/2} f(x) e^{-i2\pi \frac{n}{P}x} dx.$$

gdzie $x \in [-P/2, P/2]$,

Transformata Fouriera: wzór

Szereg fouriera dany jest przez:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi \frac{n}{P}x} \quad c_n = \frac{1}{P} \int_{-P/2}^{P/2} f(x) e^{-i2\pi \frac{n}{P}x} dx.$$

gdzie $x \in [-P/2, P/2]$, gdy $P \rightarrow \infty$ dostajemy transformatę Fouriera:

$$\widehat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\xi x} dx.$$

Transformata Fouriera: wzór

Szereg fouriera dany jest przez:

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{i2\pi \frac{n}{P}x} \quad c_n = \frac{1}{P} \int_{-P/2}^{P/2} f(x) e^{-i2\pi \frac{n}{P}x} dx.$$

gdzie $x \in [-P/2, P/2]$, gdy $P \rightarrow \infty$ dostajemy transformatę Fouriera:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi\xi x} dx.$$

Dyskretna transformata Fouriera to szereg ucięty na N wyrazów:

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{k}{N}n}, \quad k = 0, 1, 2, \dots, N-1,$$

$$(x_0, x_1, \dots, x_{N-1}) \in \mathbb{C}^N$$

Twierdzenie Nyquista–Shannona: możemy odtwarzać sygnały ciągłe dyskretnymi

Sampling Theorem and Aliasing

For any sampling interval Δ , there is also a special frequency f_c , called the *Nyquist critical frequency*, given by

$$f_c \equiv \frac{1}{2\Delta} \quad (12.1.2)$$

If a sine wave of the Nyquist critical frequency is sampled at its positive peak value, then the next sample will be at its negative trough value, the sample after that at the positive peak again, and so on. Expressed otherwise: *Critical sampling of a sine wave is two sample points per cycle.* One frequently chooses to measure time in units of the sampling interval Δ . In this case the Nyquist critical frequency is just the constant $1/2$.

Jak wydajnie liczyć transformatę Fouriera numerycznie?

Jak wydajnie liczyć transformatę Fouriera numerycznie?

Zacznijmy od prostszego pytania:

Jak wydajnie liczyć transformatę Fouriera numerycznie?

Zacznijmy od prostszego pytania:

Jak pomnożyć dwa wielomiany ze sobą?

Jak pomnożyć dwa wielomiany ze sobą?

Mamy dwa wielomiany:

$$A(x) = x^2 + 3x + 2$$

$$B(x) = 2x^2 + 1$$

Jak pomnożyć dwa wielomiany ze sobą?

Mamy dwa wielomiany:

$$A(x) = x^2 + 3x + 2$$

$$B(x) = 2x^2 + 1$$

To oczywiście:

$$C(x) = A(x) \cdot B(x) = 2 + 3x^2 + 5x^2 + 6x^3 + 2x^4$$

Jak pomnożyć dwa wielomiany ze sobą?

Mamy dwa wielomiany:

$$A(x) = x^2 + 3x + 2$$

$$B(x) = 2x^2 + 1$$

To oczywiście:

$$C(x) = A(x) \cdot B(x) = 2 + 3x^2 + 5x^2 + 6x^3 + 2x^4$$

Reprezentacja wielomianów współczynnikami

W ogólności:

$$A(x) = a_n x^d + a_{d-1} x^{d-1} + \dots + a_2 x^2 + a_1 x + a_0,$$

$$B(x) = b_n x^d + b_{d-1} x^{d-1} + \dots + b_2 x^2 + b_1 x + b_0,$$

To oczywiste:

$$C(x) = A(x) \cdot B(x) = (\text{mnożenie każdego elementu z każdym})$$

W czym problem?

Reprezentacja wielomianów współczynnikami

W ogólności:

$$A(x) = a_n x^d + a_{d-1} x^{d-1} + \dots + a_2 x^2 + a_1 x + a_0,$$

$$B(x) = b_n x^d + b_{d-1} x^{d-1} + \dots + b_2 x^2 + b_1 x + b_0,$$

To oczywiste:

$$C(x) = A(x) \cdot B(x) = (\text{mnożenie każdego elementu z każdym})$$

W czym problem?

W złożoności obliczeniowej tego procesu:

Reprezentacja wielomianów współczynnikami

W ogólności:

$$A(x) = a_n x^d + a_{d-1} x^{d-1} + \dots + a_2 x^2 + a_1 x + a_0,$$

$$B(x) = b_n x^d + b_{d-1} x^{d-1} + \dots + b_2 x^2 + b_1 x + b_0,$$

To oczywiste:

$$C(x) = A(x) \cdot B(x) = (\text{mnożenie każdego elementu z każdym})$$

W czym problem?

W złożoności obliczeniowej tego procesu: $O(d^2)$: wolno.

Reprezentacja za pomocą wartości w punktach

$$A(x) = a_n x^d + a_{d-1} x^{d-1} + \dots + a_2 x^2 + a_1 x + a_0,$$

$$B(x) = b_n x^d + b_{d-1} x^{d-1} + \dots + b_2 x^2 + b_1 x + b_0,$$

Możemy zrobić reprezentację za pomocą wartości w $d + 1$ punktach:

$$\{(x_0, A(x_0)), (x_1, A(x_1)), \dots, (x_d, A(x_d))\}$$

$$\{(x_0, B(x_0)), (x_1, B(x_1)), \dots, (x_d, B(x_d))\}$$

Reprezentacja za pomocą wartości w punktach

$$A(x) = a_n x^d + a_{d-1} x^{d-1} + \dots + a_2 x^2 + a_1 x + a_0,$$

$$B(x) = b_n x^d + b_{d-1} x^{d-1} + \dots + b_2 x^2 + b_1 x + b_0,$$

Możemy zrobić reprezentację za pomocą wartości w $d + 1$ punktach:

$$\{(x_0, A(x_0)), (x_1, A(x_1)), \dots, (x_d, A(x_d))\}$$

$$\{(x_0, B(x_0)), (x_1, B(x_1)), \dots, (x_d, B(x_d))\}$$

Wtedy mnożymy tylko dwa wektory o rozmiarze $d + 1$, co ma złożoność

Reprezentacja za pomocą wartości w punktach

$$A(x) = a_n x^d + a_{d-1} x^{d-1} + \dots + a_2 x^2 + a_1 x + a_0,$$

$$B(x) = b_n x^d + b_{d-1} x^{d-1} + \dots + b_2 x^2 + b_1 x + b_0,$$

Możemy zrobić reprezentację za pomocą wartości w $d + 1$ punktach:

$$\{(x_0, A(x_0)), (x_1, A(x_1)), \dots, (x_d, A(x_d))\}$$

$$\{(x_0, B(x_0)), (x_1, B(x_1)), \dots, (x_d, B(x_d))\}$$

Wtedy mnożymy tylko dwa wektory o rozmiarze $d + 1$, co ma złożoność $O(d)$.

Reprezentacja za pomocą wartości w punktach

$$A(x) = a_n x^d + a_{d-1} x^{d-1} + \dots + a_2 x^2 + a_1 x + a_0,$$

$$B(x) = b_n x^d + b_{d-1} x^{d-1} + \dots + b_2 x^2 + b_1 x + b_0,$$

Możemy zrobić reprezentację za pomocą wartości w $d + 1$ punktach:

$$\{(x_0, A(x_0)), (x_1, A(x_1)), \dots, (x_d, A(x_d))\}$$

$$\{(x_0, B(x_0)), (x_1, B(x_1)), \dots, (x_d, B(x_d))\}$$

Wtedy mnożymy tylko dwa wektory o rozmiarze $d + 1$, co ma złożoność $O(d)$.

Ale jak wrócić do postaci ze współczynnikami?

Podział na mniejsze wielomiany

$$P(x) = p_0 + p_1x + p_2x^2 + \cdots + p_{n-1}x^{n-1}$$

Wyznaczamy wartości na n punktach $\pm x_1, \pm x_2, \dots, \pm x_{n/2}$

$$P(x) = P_e(x^2) + xP_o(x^2)$$

Podział na mniejsze wielomiany

$$P(x) = p_0 + p_1x + p_2x^2 + \dots + p_{n-1}x^{n-1}$$

Wyznaczamy wartości na n punktach $\pm x_1, \pm x_2, \dots, \pm x_{n/2}$

$$P(x) = P_e(x^2) + xP_o(x^2)$$

$$\left. \begin{aligned} P(x_i) &= P_e(x_i^2) + x_i P_o(x_i^2) \\ P(-x_i) &= P_e(x_i^2) - x_i P_o(x_i^2) \end{aligned} \right\} \text{Znacząco się nakładają}$$

Podział na mniejsze wielomiany

$$P(x) = p_0 + p_1x + p_2x^2 + \dots + p_{n-1}x^{n-1}$$

Wyznaczamy wartości na n punktach $\pm x_1, \pm x_2, \dots, \pm x_{n/2}$

$$P(x) = P_e(x^2) + xP_o(x^2)$$

$$\left. \begin{aligned} P(x_i) &= P_e(x_i^2) + x_i P_o(x_i^2) \\ P(-x_i) &= P_e(x_i^2) - x_i P_o(x_i^2) \end{aligned} \right\} \text{Znacząco się nakładają}$$

$P_e(x^2)$ i $P_o(x^2)$ są stopnia $n/2 - 1$

Wyznaczamy wartości $P_e(x^2)$ i $P_o(x^2)$

w punktach $x_1^2, x_2^2, \dots, x_{n/2}^2$ ($n/2$ punktów)

Slajdy powstały na bazie tego filmu [\[Link\]](#)

Szybka Transformata Fouriera

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \omega_N^{-nk}, \quad k = 0, 1, 2, \dots, N-1,$$

$$(x_0, x_1, \dots, x_{N-1}) \in \mathbb{C}^N$$

$$(y_0, y_1, \dots, y_{N-1}) \in \mathbb{C}^N$$

Veritasium na temat Szybkiej transformaty Fouriera

Kwantowa Transformata Fouriera

$$y_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x_n \omega_N^{-nk}, \quad k = 0, 1, 2, \dots, N-1,$$

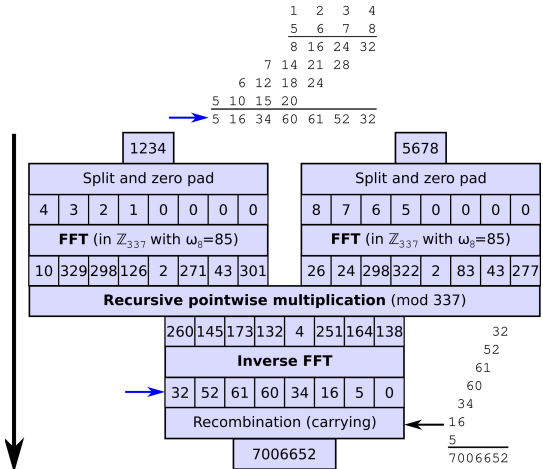
$$(x_0, x_1, \dots, x_{N-1}) \in \mathbb{C}^N$$

$$(y_0, y_1, \dots, y_{N-1}) \in \mathbb{C}^N$$

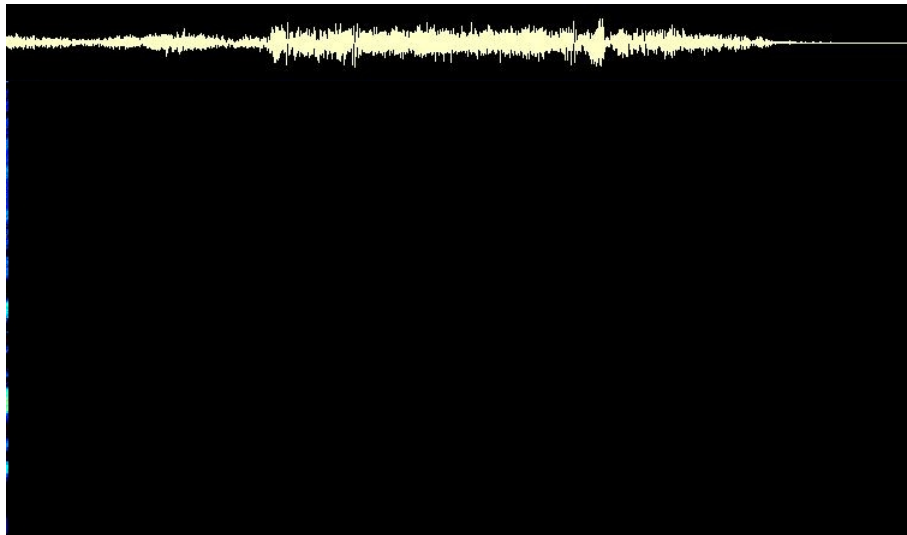
Veritasium na temat kwantowej transformaty Fouriera

Zastosowania (poza epicyklami)

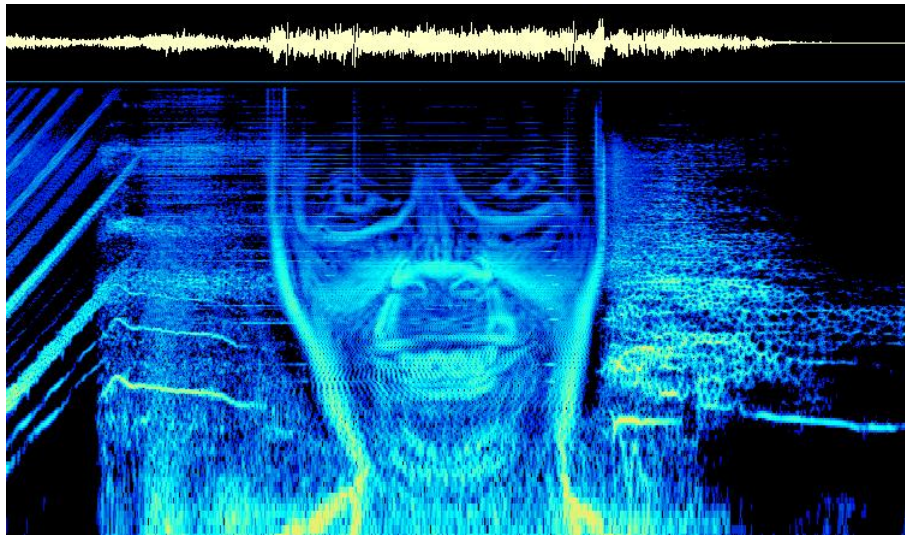
Metoda Szybszego Mnożenia Schönhage-Strassen



Short-time Fourier transform



Aphex Twin Windowlicker track 2



Short-time Fourier transform

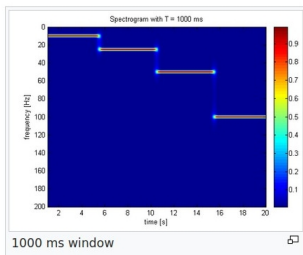
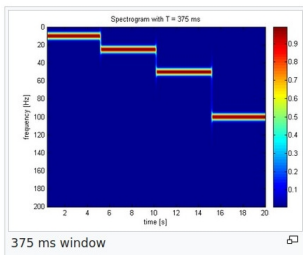
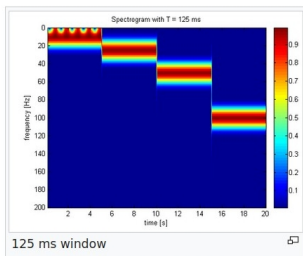
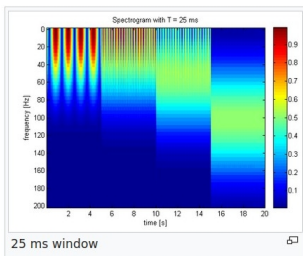
W wersji ciągłej:

$$\mathbf{STFT}\{x(t)\}(\tau, \omega) \equiv X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)w(t - \tau)e^{-i\omega t} dt$$

W wersji dyskretnej:

$$\mathbf{STFT}\{x[n]\}(m, \omega) \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n - m]e^{-i\omega n}$$

Short-time Fourier transform: nieoznaczoność ogranicza dokładność

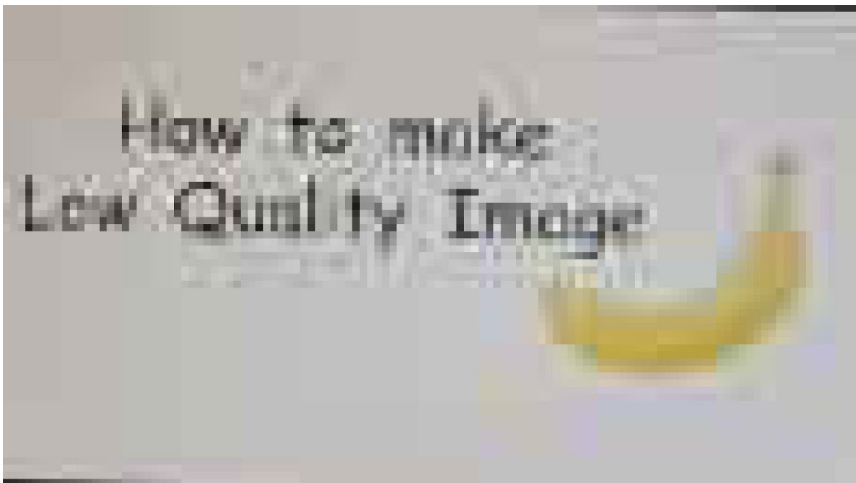


Short-time Fourier transform: nieoznaczoność ogranicza dokładność

$$\left(\int_{-\infty}^{\infty} (x - x_0)^2 |f(x)|^2 dx \right) \left(\int_{-\infty}^{\infty} (\xi - \xi_0)^2 |\hat{f}(\xi)|^2 d\xi \right) \geq \frac{1}{16\pi^2}$$

Reguła nieoznaczoności Heisenberga wynika bezpośrednio z natury falowej mechaniki kwantowej (jak widać). I ogranicza nas też w kontekście sygnału (Limit Gabora):

$$\sigma_t \cdot \sigma_f \geq \frac{1}{4\pi} \approx 0.08$$



Obrazy w formacie .jpg wykorzystują
Dyskretną zmodyfikowaną transformatę kosinusową

Zmodyfikowana dyskretna transformata kosinusowa

$$X_k = \sum_{n=0}^{2N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} + \frac{N}{2} \right) \left(k + \frac{1}{2} \right) \right]$$

Odrzucając funkcję $\sin(x) = (e^x - e^{-x})/2i$ mamy tylko liczby rzeczywiste $\cos(x) = (e^x + e^{-x})/2$: uproszczenie obliczeń.

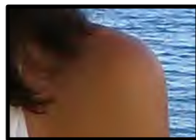
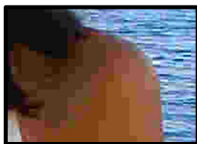
Transformata falkowa (wavelet transform)



Jpeg

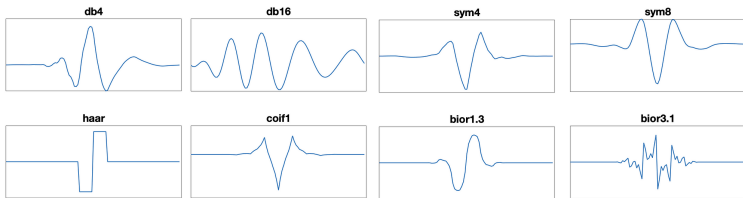


Jpeg 2000



Obrazy w formacie .jp2 wykorzystują
dyskretną transformatę falkową

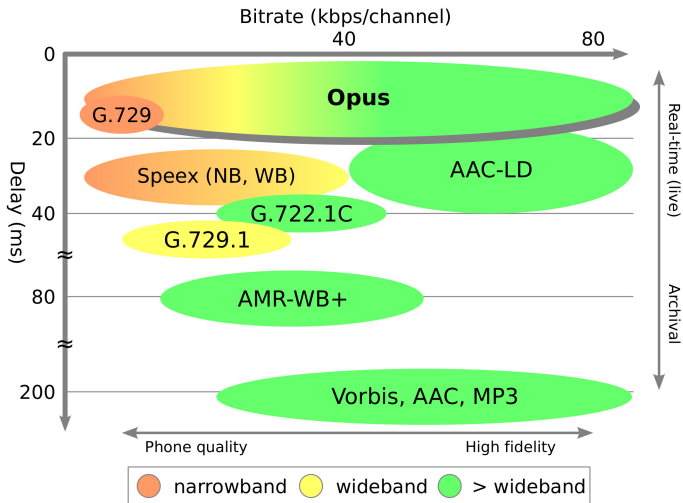
Falki (Wavelets)



Comparison with Fourier transform and time-frequency analysis [\[edit \]](#)

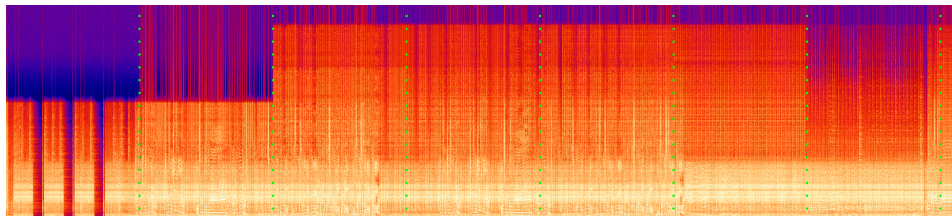
Transform	Representation	Input
Fourier transform	$\hat{X}(f) = \int_{-\infty}^{\infty} x(t)e^{-i2\pi ft} dt$	f : frequency
Time-frequency analysis	$X(t, f)$	t time; f frequency
Wavelet transform	$X(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \overline{\Psi\left(\frac{t-b}{a}\right)} x(t) dt$	a scaling ; b time shift factor

Kompresja stratna audio: MP3 oparte o TF



Kompresja stratna, na przykładzie OPUS

32kbit/s do 160 kbit/s



I wiele wiele innych zastosowań

- ▶ we wszelkiej analizie fal elektromagnetycznych (sieci komórkowe, łączność radiowa ogólniej, astronomia)
- ▶ w spektrometrii mas
- ▶ Jako fragment innej metody numerycznej

Źródła

- ▶ *Numerical Recipes 3rd Edition: The Art of Scientific Computing* 3rd Edition
- ▶ <https://pastebin.com/QCgwT6VD>



Dziękuję za uwagę

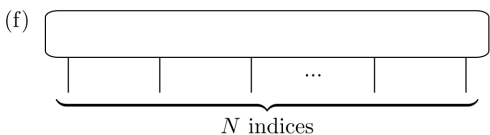
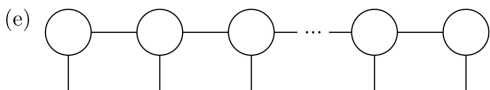
Tensor diagrams

(a) $v_j = \text{circle}(v) \downarrow j$

(b) $A_{ij} = \text{circle}(A) \downarrow i, j$

(c) $T_{jkl} = \text{circle}(T) \downarrow j, k, l$

(d) $A_{ij}T_{jkl} = \text{circle}(T) \downarrow j, k, l \text{ connected to } \text{circle}(A) \downarrow i$

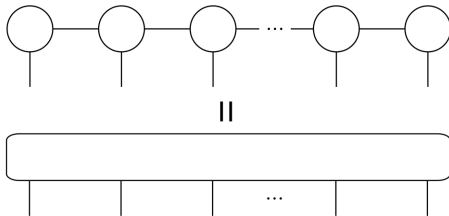


Matrix product state (Tensor train)

We can express tensor $T_{s_1 s_2 \dots s_L}$ as a Matrix product state (MPS):

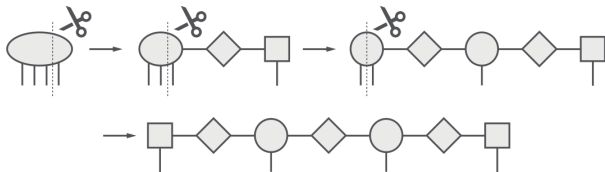
$$T_{s_1 s_2 \dots s_L} = \sum_{\alpha} A_{\alpha_1}^{s_1} A_{\alpha_1, \alpha_2}^{s_2} A_{\alpha_2, \alpha_3}^{s_3} \dots A_{\alpha_{L-1}, \alpha_L}^{s_{L-1}} A_{\alpha_L}^{s_L}$$

where $A_{\alpha_i, \alpha_{i+1}}^{s_i}$ are matrices, s_i are outer indices, α_i indices are contracted/summed over to recover tensor $T_{s_1 s_2 \dots s_L}$.

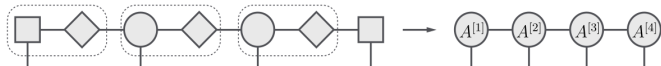


Construction of MPS via SVD

Arbitrary four-party state $|\psi\rangle$ represented by tensor is bipartited into two parts, where one leg undergoes SVD, process is repeated for entire tensor.



Then tensors can be grouped into an MPS.



Singular value decomposition (SVD)

