Aproksymacja średniokwadratowa wielomianami trygonometrycznymi

Łukasz Wala
AGH, Wydział Informatyki, Elektroniki i Telekomunikacji
Metody Obliczeniowe w Nauce i Technice 2021/2022

Kraków, 30 kwietnia 2022

1 Opis problemu

Główną ideą zadania jest zbadanie zachowania funkcji przybliżonej za pomocą aproksymacji średniokwadratowej wielomianami trygonometrycznymi.

Badana funkcja:

$$f(x) = x^2 - m \cdot \cos\left(\frac{\pi x}{k}\right)$$

Gdzie $k = \frac{1}{2}$, $m = 4$ oraz $x \in [-6, 6]$.

2 Opracowanie

2.1 Wyprowadzenie

W aproksymacji średniokwadratowej poszukiwana jest wartość minimalna sumy kwadratów różnic funkcji aproksymowanej $F(x)$ oraz funkcji aproksymującej $f(x)$ z uwzględnieniem funcji wagowej $w(x)$ większej od zera (tutaj $\forall x \in D : w(x) = 1$). Funkcją aproksymującą ma być wielomian trygonometryczny o postaci

$$f(x) = a_0 + \sum_{j=1}^{m} b_j \cos(jx) \; + \; \sum_{j=1}^{m} a_j \sin(jx)$$

Więc błąd średniokwadratowy przyjmuje postać

$$H(a_0, ..., a_m, b_1, ..., b_m) = \sum_{i=1}^{n} w(x_i) \left[F(x_i) - (a_0 + \sum_{j=1}^{m} b_j \cos(jx_i) \; + \; \sum_{j=1}^{m} a_j \sin(jx_i)) \right]^2$$

Aby funkcja przyjmowała wartość minimalną względem współczynnka c, pochodna funkcji względem tego współczynnika musi wynosić zero

$$\frac{\partial H}{\partial c} = 0, c \in \{a_0, ..., a_m, b_1, ..., b_m\}$$

Na przykład dla a_k

$$-2\sum_{i=1}^{n} w(x_i)\left[F(x_i) - (a_0 + \sum_{j=1}^{m} b_j \sin(jx_i) \ + \sum_{j=1}^{m} a_j \cos(jx_i))\right]\cos(kx_i) = 0$$

Po przekształceniach dla a_0

$$\sum_{i=0}^{n} w(x_i)a_0 + \sum_{j=1}^{m}\left(\sum_{i=0}^{n} w(x_i)\cos(jx_i)\right)a_j + \sum_{j=1}^{m}\left(\sum_{i=0}^{n} w(x_i)\sin(jx_i)\right)b_j = \sum_{i=0}^{n} w(x_i)F(x_i)$$

Dla $a_k, k \in \{1, 2, ..., m\}$:

$$\sum_{i=0}^{n} w(x_i)\cos(kx_i)a_0 + \sum_{j=1}^{m}\left(\sum_{i=0}^{n} w(x_i)\cos(kx_i)\cos(jx_i)\right)a_j + \sum_{j=1}^{m}\left(\sum_{i=0}^{n} w(x_i)\cos(kx_i)\sin(jx_i)\right)b_j = \sum_{i=0}^{n} w(x_i)F(x_i)\cos(kx_i)$$

Oraz dla $b_k, k \in \{1, 2, ..., m\}$

$$\sum_{i=0}^{n} w(x_i)\sin(kx_i)a_0 + \sum_{j=1}^{m}\left(\sum_{i=0}^{n} w(x_i)\sin(kx_i)\cos(jx_i)\right)a_j + \sum_{j=1}^{m}\left(\sum_{i=0}^{n} w(x_i)\sin(kx_i)\sin(jx_i)\right)b_j = \sum_{i=0}^{n} w(x_i)F(x_i)\sin(kx_i)$$

Z tych równań można zbudować układ

$$\begin{bmatrix}
\sum_{i=0}^{n} w(x_i) & \sum_{i=0}^{n} w(x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\sin(1\cdot x_i) & \dots \\
\sum_{i=0}^{n} w(x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\cos(1\cdot x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\cos(1\cdot x_i)\sin(1\cdot x_i) & \dots \\
\sum_{i=0}^{n} w(x_i)\sin(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\sin(1\cdot x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\sin(1\cdot x_i)\sin(1\cdot x_i) & \dots \\
\sum_{i=0}^{n} w(x_i)\cos(2\cdot x_i) & \sum_{i=0}^{n} w(x_i)\cos(2\cdot x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\cos(2\cdot x_i)\sin(1\cdot x_i) & \dots \\
\sum_{i=0}^{n} w(x_i)\sin(2\cdot x_i) & \sum_{i=0}^{n} w(x_i)\sin(2\cdot x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\sin(2\cdot x_i)\sin(1\cdot x_i) & \dots \\
\vdots & \vdots & \vdots & \vdots \\
\sum_{i=0}^{n} w(x_i)\cos(m\cdot x_i) & \sum_{i=0}^{n} w(x_i)\cos(m\cdot x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\cos(m\cdot x_i)\sin(1\cdot x_i) & \dots \\
\sum_{i=0}^{n} w(x_i)\sin(m\cdot x_i) & \sum_{i=0}^{n} w(x_i)\sin(m\cdot x_i)\cos(1\cdot x_i) & \sum_{i=0}^{n} w(x_i)\sin(m\cdot x_i)\sin(1\cdot x_i) & \dots
\end{bmatrix}.$$

$$
\cdot \begin{bmatrix} a_0 \\ a_1 \\ b_1 \\ a_2 \\ b_2 \\ \vdots \\ a_m \\ b_m \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} w(x_i)F(x_i) \\ \sum_{i=0}^{n} w(x_i)F(x_i)\cos(1\cdot x_i) \\ \sum_{i=0}^{n} w(x_i)F(x_i)\sin(1\cdot x_i) \\ \sum_{i=0}^{n} w(x_i)F(x_i)\cos(2\cdot x_i) \\ \sum_{i=0}^{n} w(x_i)F(x_i)\sin(2\cdot x_i) \\ \vdots \\ \sum_{i=0}^{n} w(x_i)F(x_i)\cos(m\cdot x_i) \\ \sum_{i=0}^{n} w(x_i)F(x_i)\sin(m\cdot x_i) \end{bmatrix}
$$

Do stworzenia i rozwiązania powyższego układu równań użyta została funkcja *linalg.solve* z pakietu *numpy* w języku Python.

2.2 Wykresy

2.3 Dokładności

3 Wnioski