Błażej Kapkowski, Konrad Konsek 10.04.2024

**„Laboratorium” 5**

**Aproksymacja**

**Dane techniczne:**

Język: Python

Translator: Visual Studio Code

Procesor: AMD Ryzen 7 5800H

System operacyjny: Windows 11

**Realizacja ćwiczenia:**

**Zadanie 1**

Zadanie polegało na wykonaniu aproksymacji średniokwadratowej punktowej populacji Stanów Zjednoczonych w przedziale [1900, 1980] wielomianami stopnia m dla 0 ≤ m ≤ 6. Następnie dla każdego stopnia wielomianu należało dokonać ekstrapolacji do roku 1990 i porównać otrzymane wartości z prawdziwą wartością dla roku 1990. Na koniec należało obliczyć błąd względny ekstrapolacji dla roku 1990 oraz wykorzystać kryterium informacyjne Akaikego (AICc), aby wybrać optymalny stopień wielomianu.

Wybór stopni wielomianu:

Tworzymy tablicę stopni wielomianu (degrees) od 0 do 6, które będziemy testować podczas aproksymacji.

degrees = np.arange(7)

Aproksymacja i ekstrapolacja:

Następnie dla każdego stopnia wielomianu wykonujemy aproksymację średniokwadratową punktową populacji w Stanach Zjednoczonych w latach 1900-1980. Na tej podstawie dokonujemy ekstrapolacji uzyskanych wielomianów do roku 1990.

year\_1990 = 1990

predicted\_populations = np.zeros((len(degrees), len(years)))

extrapolated\_values = np.zeros(len(degrees))

for i, degree in enumerate(degrees):

    coeffs = np.polyfit(years, population, degree)

    predicted\_populations[i] = np.polyval(coeffs, years)

    extrapolated\_values[i] = np.polyval(coeffs, year\_1990)

Obliczenie błędów względnych:

Dla każdego stopnia wielomianu obliczamy błąd względny ekstrapolacji, porównując uzyskane wartości z prawdziwą wartością populacji dla roku 1990.

true\_value\_1990 = 248709873

relative\_errors = np.abs(extrapolated\_values - true\_value\_1990) / true\_value\_1990

Obliczenie kryterium informacyjnego Akaikego (AICc):

Dla każdego stopnia wielomianu obliczamy wartość kryterium informacyjnego Akaikego z korektą (AICc), które pozwoli nam wybrać optymalny stopień wielomianu.

n = len(years)

AICc\_values = np.zeros(len(degrees))

for i, degree in enumerate(degrees):

    k = degree + 1

    residual = population - predicted\_populations[i]

    RSS = np.sum(residual \*\* 2)

    # Obliczenie AICc

    AICc\_values[i] = 2 \* k + n \* np.log(RSS/n) + 2 \* k \* (k + 1) / (n - k - 1)

Wybór optymalnego stopnia wielomianu:

Na koniec wyświetlamy wartości AICc dla każdego stopnia wielomianu i wybieramy stopień, dla którego AICc jest najmniejsze. Jest to optymalny stopień wielomianu według kryterium informacyjnego Akaikego.

best\_degree\_index = np.argmin(AICc\_values)

best\_degree = degrees[best\_degree\_index]

Wyniki:

Stopień wielomianu m = 0:

Wartość ekstrapolowana do 1990: 143369177.44

Błąd względny: 42.35%

Stopień wielomianu m = 1:

Wartość ekstrapolowana do 1990: 235808109.03

Błąd względny: 5.19%

Stopień wielomianu m = 2:

Wartość ekstrapolowana do 1990: 254712944.64

Błąd względny: 2.41%

Stopień wielomianu m = 3:

Wartość ekstrapolowana do 1990: 261439379.59

Błąd względny: 5.12%

Stopień wielomianu m = 4:

Wartość ekstrapolowana do 1990: 243106971.64

Błąd względny: 2.25%

Stopień wielomianu m = 5:

Wartość ekstrapolowana do 1990: 220442967.00

Błąd względny: 11.37%

Stopień wielomianu m = 6:

Wartość ekstrapolowana do 1990: 255056576.00

Błąd względny: 2.55%

Stopień wielomianu m = 0: AICc = 321.0109750531991

Stopień wielomianu m = 1: AICc = 289.0564781232789

Stopień wielomianu m = 2: AICc = 279.4533738965903

Stopień wielomianu m = 3: AICc = 284.88040168494643

Stopień wielomianu m = 4: AICc = 290.92858151471336

Stopień wielomianu m = 5: AICc = 311.25819549763605

Stopień wielomianu m = 6: AICc = 381.26317684077264

Najlepszy stopień wielomianu według AICc: m = 2

Wnioski:

Wartość błędu względnego najmniejsza jest dla stopnia wielomianu m=4, ale wartość kryterium informacyjnego Akaikego (AICc) wskazuje na optymalność stopnia m=2.

Takie rozbieżności mogą wynikać z faktu, że różne kryteria oceniają modelowanie z różnych perspektyw. Błąd względny bierze pod uwagę dokładność aproksymacji w konkretnym punkcie (wartość populacji dla roku 1990), natomiast kryterium AICc uwzględnia zrównoważenie między dokładnością aproksymacji a złożonością modelu.

W związku z tym, możemy stwierdzić, że m=2 jest preferowanym stopniem wielomianu, ponieważ otrzymaliśmy najmniejszą wartość kryterium AICc, co oznacza, że ten model jest najbardziej odpowiednią reprezentacją danych populacyjnych. Jednakże, jeśli skupimy się wyłącznie na dokładności aproksymacji w konkretnym punkcie (wartości populacji dla roku 1990), to model m=4 wydaje się być lepszym wyborem.

**Zadanie 2**

Wybór metody aproksymacji:

W celu wykonania aproksymacji wybraliśmy wielomiany Czebyszewa jako bazę aproksymującą. Wielomiany Czebyszewa są szczególnie skuteczne, ponieważ minimalizują efekt Rungego, który może prowadzić do niestabilności aproksymacji.

Generacja węzłów Czebyszewa:

Pierwszym krokiem było wygenerowanie węzłów Czebyszewa, które będą używane jako punkty interpolacyjne. Wykorzystaliśmy funkcję chebpts1() z biblioteki NumPy, która generuje węzły Czebyszewa na przedziale [-1, 1].

nodes = chebyshev.chebpts1(n\_nodes)

Następnie, węzły Czebyszewa zostały przeskalowane do odpowiedniego przedziału, w tym przypadku [0, 2]. Wykonaliśmy to przekształcenie za pomocą prostej operacji arytmetycznej.

nodes\_scaled = 1 + (1 - nodes)

Kolejnym krokiem było obliczenie wartości funkcji f(x) = w wygenerowanych węzłach Czebyszewa.

def f(x):

    return np.sqrt(x)

values = f(nodes\_scaled)

Dyskretne przekształcenie kosinusowe (DCT):

Wykorzystaliśmy dyskretne przekształcenie kosinusowe (DCT) do obliczenia współczynników aproksymacyjnego wielomianu Czebyszewa. DCT pozwala efektywnie przekształcić wartości funkcji w przestrzeń współczynników wielomianu.

coeffs = dct(values, norm='ortho')

Utworzenie wielomianu aproksymacyjnego:

Na podstawie obliczonych współczynników utworzyliśmy wielomian aproksymacyjny, który stanowi przybliżenie funkcji f(x) = w przedziale [0, 2] za pomocą wielomianu drugiego stopnia.

approx\_polynomial = chebyshev.Chebyshev(coeffs)

Wyniki:  
Współczynniki aproksymacyjnego wielomianu Czebyszewa: [ 2.40872262 0.4440995 -0.02882671]

Wielomian aproksymacyjny: 2.40872262 + 0.4440995 T\_1(x) - 0.02882671 T\_2(x).

**Wnioski:**