Artur Gęsiarz,
Kwiecień 12, 2024

Laboratorium nr 5
MOwNiT – Aproksymacja

1. Treść zadań

1.1. Zadanie 1

Wykonam aproksymacja średniokwadratową punktową populacji Stanów Zjednoczonych w przedziale [1900, 1980] wielomianami stopnia m dla $0 <= m <= 6$.

Dla każdego m dokonam ekstrapolacji wielomianu do roku 1990. Porównam otrzymaną wartość z prawdziwą wartością dla roku 1990, wynoszącą 248 709 873. Odpowiem na pytanie ile wynosi błąd względny ekstrapolacji dla roku 1990 oraz dla jakiego m błąd względny był najmniejszy.

Zbyt niski stopień wielomianu oznacza, że model nie jest w stanie uwzględnić zmienności danych (duże obciążenie). Zbyt wysoki stopień wielomianu oznacza z kolei, że model uwzględnia szum lub błędy danych (duża wariancja), co w szczególności obserwowaliśmy w przypadku interpolacji. Wielomian stopnia m posiada k = m + 1 parametrów. Stopień wielomianu, m, jest hiperparametrem modelu. Do wyboru optymalnego stopnia wielomianu można posłużyć się kryterium informacyjnym Akaikego (ang. Akaike infor- mation criterion):

$$AIC = 2k + n * \ln\left(\frac{\sum_{i=1}^{n}(y_i - \hat{y}(x_i))^2}{n}\right)$$

gdzie $y_i \in (i = 1, \dots, n)$ oznacza prawdziwą liczbę osób w roku xi, natomiast y^(xi) liczbę osób przewidywaną przez model, tzn wartość wielomianu y^(x).
Ponieważ rozmiar próbki jest niewielki (dane z dziewięciu lat, n = 9), n/k < 40, należy użyć wzoru ze składnikiem korygującym:

$$AIC_c = AIC + \frac{2k(k+1)}{n-k-1}$$

Mniejsze wartości kryterium oznaczają lepszy model. Czy wyznaczony w ten sposób stopień m, odpowiadający najmniejszej wartości AICc, pokrywa się z wartością z poprzedniego podpunktu?

1.2. Zadanie 2

Wykonam aproksymacje średniokwadratową ciągła funkcji $f(x) = \sqrt{x}$ w przedziale [0,2] wielomianem drugiego stopnia.

Wykonam to używając wielomianów Czebyszewa. Aproksymacja ta jest tańszym obliczeniowo zamiennikiem aproksymacji jednostajnej.

2. Rozwiązanie zadań

2.1. Rozwiązanie zadania pierwszego

2.1.1. Populacja Stanów Zjednoczonych na przestrzeni lat

```
population_US = {
    1900: 76212168,
    1910: 92228496,
    1920: 106021537,
    1930: 123202624,
    1940: 132164569,
    1950: 151325798,
    1960: 179323175,
    1970: 203302031,
    1980: 226542199
}
```

2.1.2. Funkcja do aproksymacji średniokwadratowej punktowej

```
def polyfit_and_extrapolate(years, populations, degree, extrapolate_year):

    coeffs = np.polyfit(years, populations, degree)
    extrapolated_population = np.polyval(coeffs, extrapolate_year)

    return extrapolated_population
```

2.1.3. Rok do ekstrapolacji

```
extrapolate_year = 1990
```

2.1.4. Prawdziwa wartość populacji dla roku 1990

```
true_population_1990 = 248709873
```

2.1.5. Wartości wielomianów dla różnych stopni

```
errors = []
for degree in range(7):
    extrapolated_population = polyfit_and_extrapolate(list(population_US.keys()), list(population_US.values
     ()), degree, extrapolate_year)
    relative_error = abs(true_population_1990 - extrapolated_population) / true_population_1990
    errors.append(relative_error)

    print(f"Błąd względny dla stopnia {degree} wynosi: {relative_error}")
```

2.1.6. Znalezienie najmniejszego błędu względnego i odpowiadającego stopnia wielomianu

```
min_error_index = np.argmin(errors)
min_error_degree = min_error_index
min_error = errors[min_error_index]

print(f"Najmniejszy błąd względny wynosi: {min_error} dla stopnia {min_error_degree}")
```

2.1.7. Funkcja obliczająca AICc

```
def calculate_AICc(n, k, sum_squared_error):
    AIC = 2 * k + n * np.log(sum_squared_error)
    AICc = AIC + (2 * k * (k + 1)) / (n - k - 1)
    return AICc
```

2.1.8. Obliczenie sumy kwadratów różnic miedzy wartościami rzeczywistymi a wartościami aproksymacji

```
sum_squared_error = np.sum((np.array(list(population_US.values())) - polyfit_and_extrapolate(list
 (population_US.keys()), list(population_US.values()), min_error_degree, list(population_US.keys())) ) ** 2)
```

2.1.9. Obliczenie AICc dla każdego stopnia wielomianu

```
AICc_values = []
for degree in range(7):
    k = degree + 1  # liczba parametrów w modelu
    AICc = calculate_AICc(len(population_US), k, sum_squared_error)
    AICc_values.append(AICc)
    print(f"AICc dla stopnia {degree}: {AICc}")
```

2.1.10. Wybór stopnia wielomianu odpowiadającego najmniejszej wartości AICc

```
optimal_degree = np.argmin(AICc_values)
print(f"Optymalny stopień wielomianu wg AICc: {optimal_degree}")
```

2.1.11. Sprawdzenie czy optymalny stopień zgadza się z wynikiem z poprzedniego podpunktu

```
if optimal_degree == min_error_degree:
    print("Optymalny stopień wielomianu wyznaczony za pomocą AICc pokrywa się z poprzednim wynikiem.")
else:
    print("Optymalny stopień wielomianu wyznaczony za pomocą AICc nie pokrywa się z poprzednim wynikiem.")
print(f"Opitmal Degree: {optimal_degree}.\nMin Error Degree: {min_error_degree}.")
```

2.2. Rozwiązanie zadania drugiego

2.2.1. Funkcja pierwiastka kwadratowego

```
def sqrt_func(x):
    return np.sqrt(x)
```

2.2.2. Funkcja generująca węzły Czebyszewa

```
def chebyshev_nodes(n):
    k = np.arange(1, n + 1)
    nodes = np.cos((2 * k - 1) * np.pi / (2 * n))
    return nodes
```

2.2.3. Aproksymacja wielomianowa wielomianem stopnia 2 przy użyciu węzłów Czebyszewa

```
def chebyshev_approximation(f, n):
    """
    :param f:
    :param n:
    :return: Współczynniki aproksymacji wielomianem Czebyszewa oraz wygenerowane węzły Czebyszew
    """
    nodes = chebyshev_nodes(n)

    # Obliczanie wartości funkcji w węzłach
    values = f(nodes)

    # Obliczanie współczynników aproksymacji poprzez funkcje chebfit, ktora
    # na podstawie nodow oraz wartosci oblicza najlepsze dopasowanie do wspolcznnikow
    # dla stopnia drugiego wielomianu
    coeffs = np.polynomial.chebyshev.chebfit(nodes, values, 2)

    return coeffs, nodes
```

2.2.4. Aproksymacja wielomianowa

```
degree = 2
coeffs, nodes = chebyshev_approximation(sqrt_func, degree + 1)
```

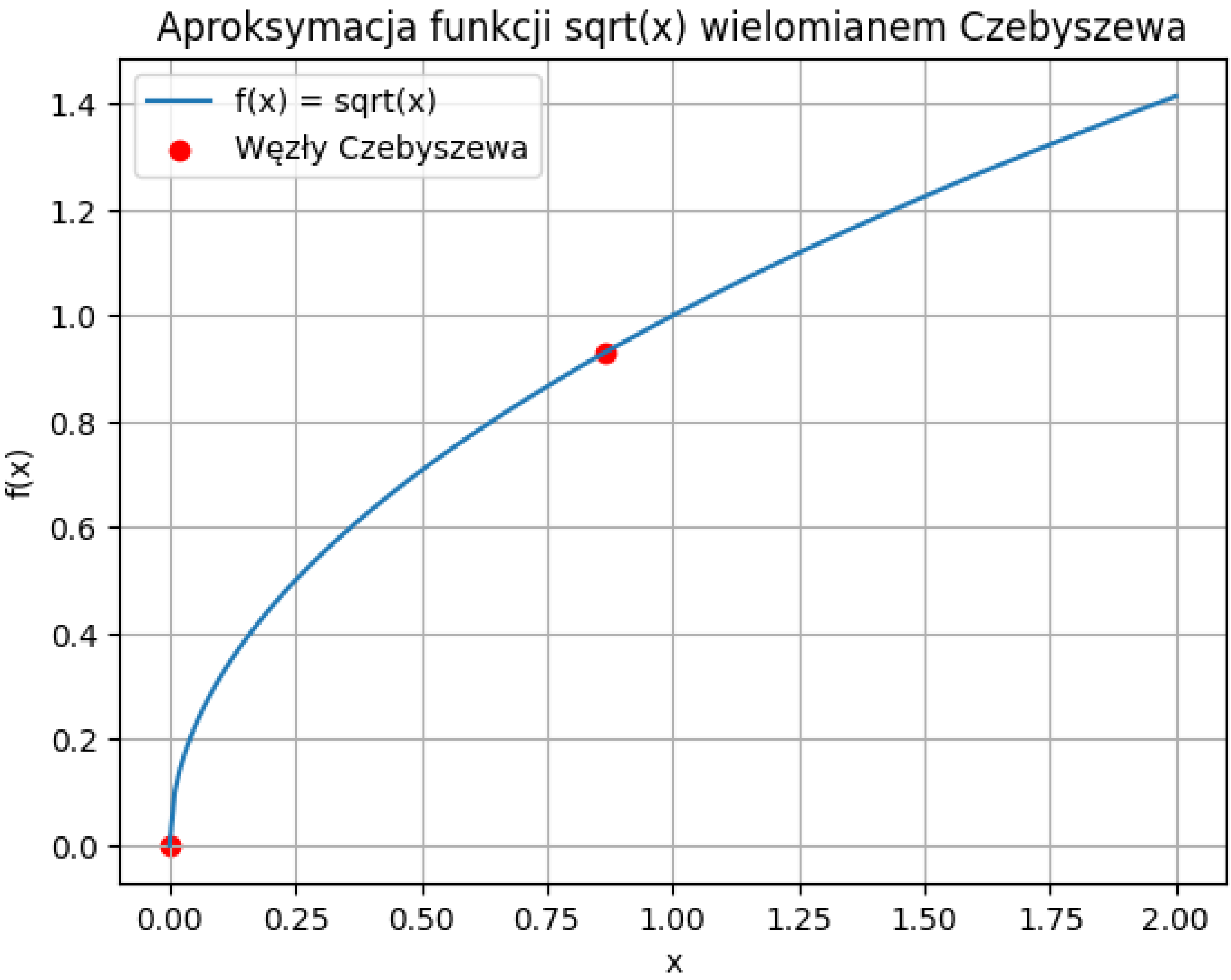
2.2.5. Wyświetlanie wyników

```
x_values = np.linspace(0, 2, 200)
y_values = sqrt_func(x_values)
approx_values = np.polynomial.chebyshev.chebval(x_values, coeffs)

plt.plot(x_values, y_values, label="f(x) = sqrt(x)")
plt.scatter(nodes, sqrt_func(nodes), color='red', label='Węzły Czebyszewa')
plt.xlabel('x')
plt.ylabel('f(x)')
plt.title('Aproksymacja funkcji sqrt(x) wielomianem Czebyszewa')
plt.legend()
plt.grid(True)
plt.show()
```

3. Wykresy

3.1. Aproksymacja funkcji pierwiastka z x, wielomianem Czebyszewa



Wykres 1. Wykres przedstawiający aproksymacje funkcji wielomianem Czebyszewa

4. Tabele

4.1. Tabela błędów względnych dla różnych stopni wielomianu:

Błąd względny dla stopnia:	Wartość błędu
0	0.423
1	0.051
2	0.024
3	0.051
4	0.022
5	0.113
6	0.025

Tabela 1. Tabela błędów względnych dla różnych stopni wielomianu

4.2. Tabela obliczeń AICc dla każdego stopnia wielomianu:

Stopień wielomianu:	Wartość AICc
0	283.27
1	286.70
2	291.50
3	298.70
4	310.70
5	334.70
6	406.70

Tabela 2. Tabela kolejnych obliczeń AICc dla poszczególnych stopni wielomianu

5. Wnioski

Na podstawie danych i wyników otrzymanych z zadania, możemy wyciągnąć kilka wniosków

Błąd względny dla różnych stopni wielomianów pokazuje, jak dobrze dany stopień wielomianu dopasowuje się do danych. W tym przypadku, stopień wielomianu 4 wykazuje najmniejszy błąd względny, co sugeruje, że jest to najlepszy stopień wielomianu dla tej aproksymacji.

Wyniki AICc sugerują, że optymalnym stopniem wielomianu jest stopień 0. Oznacza to, że według kryterium informacyjnego Akaikego, najprostszy model (liniowy) jest preferowany. Jest to ciekawe, ponieważ optymalny stopień wyznaczony za pomocą AICc nie pokrywa się z wynikiem minimalnego błędu względnego.

Istnieje rozbieżność między wynikiem optymalnego stopnia wielomianu wyznaczonym za pomocą minimalnego błędu względnego, a wynikiem wyznaczonym za pomocą AICc. Jest to dość nietypowa sytuacja, która sugeruje, że różne kryteria wyboru modelu mogą prowadzić do różnych wniosków. Może to wynikać z różnych założeń lub preferencji dotyczących złożoności modelu.

Analiza AICc sugeruje, że najprostszy model (stopień 0) jest preferowany z punktu widzenia złożoności, podczas gdy analiza minimalnego błędu względnego sugeruje, że stopień 4 jest najlepszym modelem z punktu widzenia dopasowania do danych. To otwiera dyskusję na temat kompromisu między złożonością modelu a jego zdolnością do opisywania danych.

Zauważone zostało też że, aproksymacja wielomianami Czebyszewa jest tańszym obliczeniowo zamiennikiem aproksymacji jednostajnej głównie ze względu na sposób doboru węzłów i obliczania współczynników aproksymacji.

W aproksymacji jednostajnej węzły są równomiernie rozmieszczone na zadanym przedziale, co może prowadzić do zwiększonej gęstości w obszarach, gdzie funkcja ma duże zmiany, a także zbędnej gęstości w obszarach o małej zmienności. Natomiast w aproksymacji Czebyszewa węzły są dobrze rozłożone na podstawie miejsc zerowych wielomianów Czebyszewa, co pozwala na bardziej efektywne dopasowanie wielomianu do funkcji.

6. Bibliografia
Wykład MOwNiT - prowadzony przez dr. Inż. K. Rycerz
Prezentacje – dr. Inż. M. Kuta

7. Dodatkowe informacje
Rozwiązanie obu zdań odpowiednio znajdują się w pliku ex1.ipynb, oraz ex2.ipynb