Opis modelu matematycznego - dopasowanie modelu logistycznego z członem wielomianowym do danych COVID-19

Artur Bieniek

"Jest 01:37, zabieram się do pisania swojej pierwszej w życiu notatki w TEX-u, to będzie ciekawa noc..." - powiedział na pewno ktoś kiedyś

1 Wstęp

Celem niniejszej notatki jest przedstawienie modelu matematycznego, który służy do dopasowania danych dotyczących liczby aktywnych przypadków w pandemii. Model ten łączy funkcję logistyczną z dodatkowym składnikiem wielomianowym, aby umożliwić uchwycenie zarówno charakterystyki szybkiego wzrostu w początkowych etapach pandemii, jak i późniejszego wygaszania wzrostu.

2 Model Matematyczny

Model matematyczny, który wykorzystuję, jest rozszerzoną wersją klasycznego modelu logistycznego. W skład modelu wchodzi człon logistyczny oraz jeden dodatkowy człon wielomianowy, który wprowadza większą elastyczność w modelowaniu krzywej.

Model ma postać:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} + (ax^4 + bx^3)$$

gdzie:

- $f(x)$ - liczba aktywnych przypadków w dniu x (liczba dni od pierwszego przypadku),
- L - maksymalna wartość, do której zbliża się liczba przypadków (asymptota),
- k - współczynnik szybkości wzrostu przypadków,
- x_0 - punkt infleksji, który określa dzień, w którym tempo wzrostu zaczyna zwalniać,

- $(ax^4 + bx^3)$ - wielomian stopnia 4 (człon $(cx^2 + dx + e)$ nie okazał się grać dużej roli w kształcie funkcji),
- x - liczba dni od pierwszego przypadku.

2.1 Człon Logistyczny

Człon logistyczny:

$$\frac{L}{1 + e^{-k(x-x_0)}}$$

jest klasycznym modelem wzrostu, który początkowo wykazuje szybki przyrost, a następnie wygasza wzrost w miarę zbliżania się do wartości asymptotycznej L. Funkcja ta dobrze opisuje wiele procesów biologicznych i epidemiologicznych, w tym rozprzestrzenianie się chorób zakaźnych, które w początkowej fazie mają gwałtowny wzrost, a później spowalniają w wyniku różnych czynników, takich jak wyczerpywanie się zasobów lub wprowadzenie środków zapobiegawczych.

2.2 Człon wielomianowy

Człon wielomianowy:

$$(ax^4 + bx^3)$$

wprowadza dodatkową elastyczność do modelu, pozwalając na uchwycenie nieliniowych zmian w przebiegu epidemii. Współczynniki niższych potęg można zupełnie pominąć bez utraty dopasowania. Zwiększenie stopnia wielomianu powyżej 6 prowadzi do wystąpienia poważnych problemów numerycznych, ponieważ docelowe wartości współczynników a, b są bardzo małymi ułamkami.

3 Dopasowanie Modelu do Danych

Aby dopasować model do rzeczywistych danych, wykorzystuję metodę aproksymacji średniokwadratowej, której celem jest minimalizacja sumy kwadratów różnic między wartościami rzeczywistymi a wartościami przewidywanymi przez model. Dopasowanie to wykonuję przy pomocy funkcji `curve_fit` z biblioteki `scipy.optimize`, która pozwala na optymalizację parametrów modelu.

Funkcja `curve_fit` wymaga podania następujących argumentów:

- Funkcja modelu, która opisuje zależność między zmiennymi niezależnymi a zależnymi,
- Zmienna niezależna (x) - w tym przypadku liczba dni od pierwszego przypadku,
- Zmienna zależna (y) - liczba aktywnych przypadków,
- Początkowe wartości parametrów (szacowane na podstawie danych lub wiedzy na temat modelu)

- $maxfev$ - liczbę maksymalnych iteracji, domyślnie jest ona zbyt mała

Na podstawie tych danych, funkcja `curve_fit` zwróci optymalne wartości parametrów L, k, x_0, a i b, które najlepiej pasują do rzeczywistego przebiegu danych.

4 Interpretacja Parametrów

Po dopasowaniu modelu do danych, otrzymujemy optymalne wartości parametrów, które pozwalają na interpretację zachowania epidemii:

- L wskazuje maksymalną liczbę przypadków, do której może dojść w populacji.
- k określa szybkość wzrostu liczby przypadków w początkowej fazie epidemii.
- x_0 informuje o dniu, w którym tempo wzrostu przypadków zaczyna zwalniać.
- a i b umożliwiają dopasowanie krzywej do nieliniowych trendów, które mogą występować w danych.

5 Dane

Zgodnie z zasadą "śmieci na wejściu - śmieci na wyjściu" szukałem przez chwilę w internecie dobrego, zaufanego źródła danych w strawnym formacie, jednak ostatecznie zdecydowałem się wykorzystać zarekomendowany zestaw danych z tego linku. Był on o tyle wygodny, że wystarczyło zmienić nazwę kolumny z sumaryczną liczbą przypadków zachorowań od początku epidemii i używać wartości z kolejnych wierszy jako wskazań dla kolejnych dni.

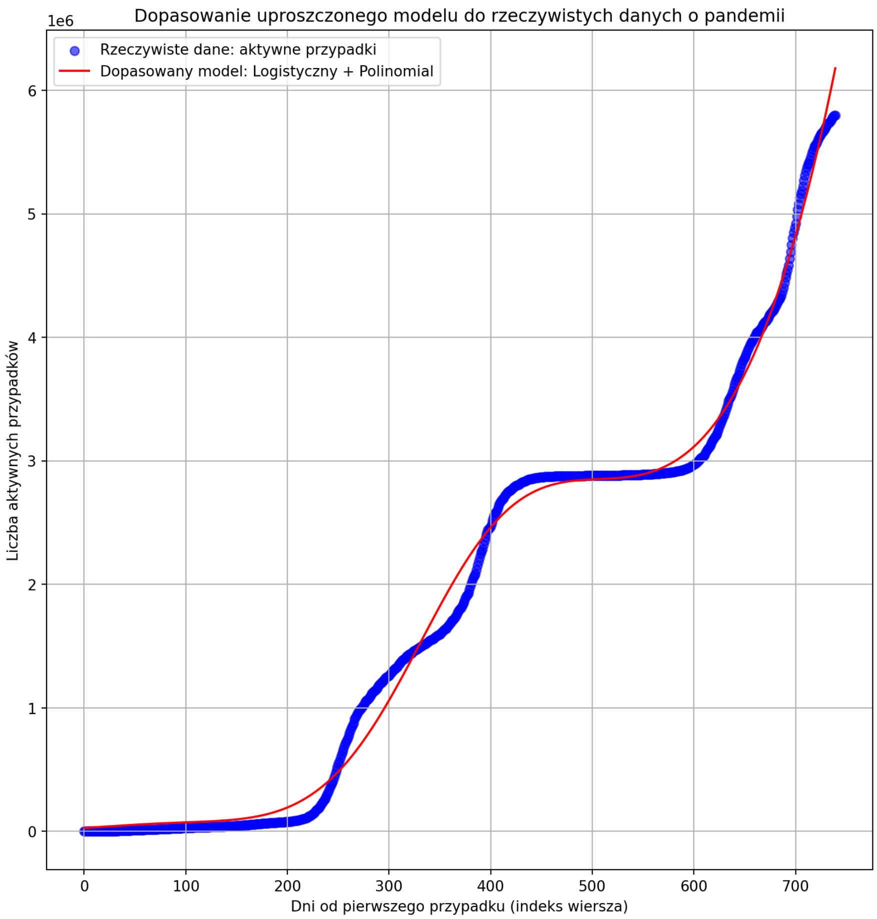
6 Kod w Pythonie

```
import pandas as pd
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt

data = pd.read_csv('covid.csv')

def model(x, L, k, x0, a, b):
    logistic_part = L / (1 + np.exp(-k * (x - x0)))
    polynomial_part = a*x**4+b*x**3
    return logistic_part + polynomial_part

x_real = np.arange(len(data))
y_real = data['ActiveCases'].values

initial_guess = [max(y_real), 0.1, len(data) // 2, 0.1, 0.5]
params_real, covariance_real = curve_fit(model, x_real, y_real, p0=initial_guess

L_real, k_real, x0_real, a_real, b_real = params_real

y_fit_real = model(x_real, L_real, k_real, x0_real, a_real, b_real)

plt.figure(figsize=(12, 7))
plt.scatter(x_real, y_real, label='Rzeczywiste dane: aktywne przypadki', color='
plt.plot(x_real, y_fit_real, label='Dopasowany model: Logistyczny + Polinomial',
plt.title('Dopasowanie uproszczonego modelu do rzeczywistych danych o pandemii')
plt.xlabel('Dni od pierwszego przypadku (indeks wiersza)')
plt.ylabel('Liczba aktywnych przypadk w')
plt.legend()
plt.grid(True)
plt.show()

print(f"Fitted parameters: L = {L_real:.2f}, k = {k_real:.4f}, x0 = {x0_real:.2f
```

7 Osiągnięte parametry

- L = 6454970.70
- k = 0.0161
- x0 = 336.46
- a = 0.000103
- b = -0.077099

8 Efekt



9 Podsumowanie

Trzeba przyznać zgodnie z prawdą, że opracowany przeze mnie model jest prawdopodobnie bezużyteczny. Niemniej jednak, tworząc go:

- Dowiedziałem się, że istnieje coś takiego jak model logistyczny
- Nauczyłem się świadomie korzystać z potężnego narzędzia do dopasowywania modeli (scipy)
- Napisałem pierwszy raz w życiu cokolwiek w LaTeX, przy okazji całkiem nieźle się go nauczyłem

Także zdecydowanie nie można odmówić temu zadaniu wartości edukacyjnej.