Sprawozdanie z laboratorium nr 4

Aproksymacja - metoda najmniejszych kwadratów

Wojciech Matys

Wydział WIMiIP

Kierunek: Inżynieria obliczeniowa

Grupa lab nr 4

Data wykonania: 04.04.2024

Cel ćwiczenia

Celem laboratorium było zapoznanie się z aproksymacją metodą najmniejszych kwadratów oraz implentacja jej w wybranym przez siebie języku, ja zdecydowałem się na C++.

Wstęp teoretyczny

Aproksymacja jest procesem znajdowania funkcji lub modelu, który najlepiej odpowiada zestawowi danych pomiarowych. Głównym celem aproksymacji jest znalezienie funkcji, która pozwala przewidywać wartości dla danych nieobjętych oryginalnym zestawem danych, bazując na tych dostępnych.

W metodach numerycznych, aproksymacja polega na dopasowywaniu określonej klasy funkcji matematycznych do danych pomiarowych za pomocą różnych technik, takich jak metoda najmniejszych kwadratów. Kluczowym aspektem aproksymacji jest minimalizacja błędu pomiędzy rzeczywistymi wartościami danych, a wartościami przewidywanymi przez funkcję aproksymującą. Dzięki temu procesowi, możemy uzyskać model, który jak najwierniej odwzorowuje zachowanie danych pomiarowych i może być używany do przewidywania nowych wartości z pewnym stopniem pewności.

Aproksymacja metodą najmniejszych kwadratów

Aproksymacja metodą najmniejszych kwadratów jest ważnym narzędziem w analizie matematycznej, służącym do znalezienia funkcji, która najlepiej pasuje do zestawu danych pomiarowych. W przeciwieństwie do interpolacji, która stara się dokładnie odwzorować wszystkie punkty danych, aproksymacja MNK koncentruje się na minimalizacji różnicy między wartościami rzeczywistymi a wartościami przewidywanymi przez funkcję dopasowującą.

Metoda najmniejszych kwadratów (MNK) jest narzędziem obliczeniowym w statystyce, które stosuje się do wyznaczania linii regresji dla dostarczonych danych. Może być używana do oszacowania zarówno zależności liniowych, jak i nieliniowych. W dzisiejszych laboratoriach skupiliśmy się na oszacowaniu zależności liniowych. MNK stara się aproksymować nasze dane przy użyciu prostej linii, takiej że suma kwadratów błędów dopasowania jest jak najniższa. Błąd dopasowania to różnica między faktyczną wartością punktu danych, a wartością, którą punkt miałby na linii aproksymacyjnej. W celu tego, zakładamy równanie parametryczne prostej w postaci (równanie 1).

$$y = a_0 + a_1 x \qquad (1)$$

Gdzie:

x - zmienna zależna

y - zmienna niezależna

a_0 - współczynnik kierunkowy prostej

a_1 - wyraz wolny

Współczynniki a_0 i a_1 są szukanymi w metodzie regresji liniowej. Poszukiwanie tych parametrów realizujemy poprzez minimalizację sumy kwadratów różnic pomiędzy wartościami zmiennych zależnych a wartościami przewidywanymi:

$$S(a,b) = \sum_{i=1}^{n} [y_i - y(x_i)]^2 = \sum_{i=1}^{n} [y_i - a_0 - a_1 x_i]^2 \qquad (2)$$

Funkcja wielu zmiennych ma minimum w punkcie, dla którego pochodne cząstkowe po tej funkcji po wszystkich zmiennych wynoszą 0.

$$\frac{\partial S(a_0, a_1)}{\partial a_0} = 0 \qquad (3)$$

$$\frac{\partial S(a_0, a_1)}{\partial a_1} = 0 \qquad (4)$$

Po wyliczeniu pochodnych możemy wyznaczyć parametry a_0 i a_1:

$$a_0 = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \qquad (5)$$

$$a_1 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \quad (6)$$

Współczynnik korelacji (R) jest miarą określającą stopień związku między zmiennymi x i y. W kontekście aproksymacji metodą najmniejszych kwadratów, współczynnik korelacji pomaga ocenić, jak silna jest korelacja między wartościami zmiennej niezależnej x i zależnej y. Gdy wartość współczynnika korelacji jest bliska 1, oznacza to silną dodatnią korelację między zmiennymi, co sugeruje, że zmienne te są dobrze skorelowane.

Wartość współczynnika korelacji R mieści się w zakresie od -1 do 1. Wartość bliska -1 wskazuje na silną ujemną korelację, czyli gdy jedna zmienna rośnie, druga maleje. Natomiast wartość bliska 1 oznacza silną dodatnią korelację, gdzie obie zmienne rosną lub maleją równocześnie. W przypadku, gdy R = 1, zmienne x i y są całkowicie skorelowane, co sugeruje istnienie pewnej zależności funkcjonalnej między nimi.

Współczynnik korelacji (R) można obliczyć za pomocą wzoru, który określa stosunek kowariancji między zmiennymi a iloczynu odchyleń standardowych obu zmiennych.

$$R = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\ \sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \quad (7)$$

Implementacja numeryczna

Aproksymacja metodą najmniejszych kwadratów implementuję w języku programowania c++, korzystając z środowiska Visual Studio

```
int main() {
    int rozmiar;
    fstream czytaj;
    czytaj.open("MN3.txt");
    czytaj >> rozmiar;
    cout << rozmiar;

    int** M = new int* [rozmiar];
    for (int i = 0; i < rozmiar; i++)
        M[i] = new int[2]; // Teraz przyjmujemy dwie kolumny

    cout << endl;
    for (int i = 0; i < rozmiar; i++) {
        for (int j = 0; j < 2; j++) {
            czytaj >> M[i][j];
            cout << M[i][j] << " ";
        }
        cout << endl;
    }
```

(1. Fragment kodu przedstawiający odczyt z pliku)

(1) Ta część kodu odpowiada za odczytanie danych z pliku tekstowego i zapisanie ich w tablicy dwuwymiarowej. Rozmiar tablicy jest odczytywany z pierwszej linii pliku, a pozostałe dane są odczytywane wiersz po wierszu. Odczytane dane są wyświetlane na ekranie.

```
void linear_approx(int** M, int rozmiar, double& a0, double& a1) {
    double sum_x = 0, sum_y = 0, sum_x2 = 0, sum_xy = 0;

    // Obliczanie sum potrzebnych do obliczenia współczynników a0 i a1
    for (int i = 0; i < rozmiar; i++) {
        double x = M[i][0];
        double y = M[i][1];
        sum_x += x;
        sum_y += y;
        sum_x2 += x * x;
        sum_xy += x * y;
    }

    // Obliczanie współczynnika a1
    a1 = (rozmiar * sum_xy - sum_x * sum_y) / (rozmiar * sum_x2 - sum_x * sum_x);

    // Obliczanie współczynnika a0
    a0 = (sum_y * sum_x2 - sum_x * sum_xy) / (rozmiar * sum_x2 - sum_x * sum_x);
}
```

(2. Fragment kodu przedstawiający funkcję służącą do obliczania współczynników)

(2) Kroki działania funkcji `***linear_approx***`:

1. Inicjalizacja zmiennych: Na początku funkcji inicjalizowane są zmienne `sum_x`, `sum_y`, `sum_x2`, `sum_xy` na wartość 0. Te zmienne będą przechowywać sumy potrzebne do obliczenia współczynników prostej regresji liniowej.

2. Iteracja po danych: Następnie funkcja przechodzi przez każdy punkt danych z tablicy dwuwymiarowej `M`. Dla każdego punktu, pobiera wartość x (zmienna niezależna) z pierwszej kolumny i wartość y (zmienna zależna) z drugiej kolumny.

3. Obliczenie sum: Dla każdego punktu danych funkcja dodaje wartości x i y do odpowiednich sum `sum_x` i `sum_y`. Dodatkowo, oblicza sumę kwadratów wartości x i sumę iloczynów x i y, zapisując je odpowiednio w zmiennych `sum_x2` i `sum_xy`.

4. Obliczenie współczynnika a1: Po zakończeniu iteracji funkcja oblicza współczynnik kierunkowy (a1) prostej regresji liniowej. Wzór na a_1 to:

$$a_1 = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

gdzie `n` to liczba punktów danych.

5. Obliczenie współczynnika a0: Następnie funkcja oblicza wyraz wolny (a0) prostej regresji liniowej. Wzór na a_0 to:

$$a_0 = \frac{\sum_{i=1}^{n} y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i y_i}{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

6. Zapisanie wartości współczynników: Ostatecznie, obliczone współczynniki `$a_0$` i `$a_1$` są przypisywane do zmiennych referencyjnych a0 i a1, które są przekazane do funkcji jako argumenty.

```c
double correlation_coefficient(int** M, int rozmiar, double a0, double a1) {
    double sum_x = 0, sum_y = 0, sum_xy = 0, sum_x_squared = 0, sum_y_squared = 0;
    for (int i = 0; i < rozmiar; i++) {
        double x = M[i][0];
        double y = M[i][1];
        sum_x += x;
        sum_y += y;
        sum_xy += x * y;
        sum_x_squared += x * x;
        sum_y_squared += y * y;
    }
    double r_numerator = rozmiar * sum_xy - sum_x * sum_y;
    double r_denominator = sqrt((rozmiar * sum_x_squared - sum_x * sum_x) * (rozmiar * sum_y_squared - sum_y * sum_y));
    double r = r_numerator / r_denominator;
    return r;
}
```

(3. Fragment kodu przedstawiający funkcję służącą do obliczania współczynnika korelacji)

(3) Kroki działania funkcji `***correlation_coefficient***`:

1. Inicjalizacja zmiennych: Na początku funkcji inicjalizowane są zmienne `sum_x`, `sum_y`, `sum_xy`, `sum_x_squared`, `sum_y_squared` na wartość 0. Te zmienne będą przechowywać sumy potrzebne do obliczenia współczynnika korelacji.

2. Iteracja po danych: Następnie funkcja przechodzi przez każdy punkt danych z tablicy dwuwymiarowej `M`. Dla każdego punktu, pobiera wartość x (zmienna niezależna) z pierwszej kolumny i wartość y (zmienna zależna) z drugiej kolumny.

3. Obliczenie sum: Dla każdego punktu danych funkcja dodaje wartości x i y do odpowiednich sum `sum_x` i `sum_y`. Dodatkowo, oblicza sumę iloczynów x i y, sumę kwadratów wartości x i sumę kwadratów wartości y, zapisując je odpowiednio w zmiennych `sum_xy`, `sum_x_squared` i `sum_y_squared`.

4. Obliczenie współczynnika korelacji: Po zakończeniu iteracji funkcja oblicza współczynnik korelacji (r) za pomocą wzoru Pearsona. Wzór ten mierzy siłę i kierunek zależności liniowej między zmiennymi x i y. Wzór na r to:

$$R = \frac{n \sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2} \sqrt{n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}}$$

gdzie `n` to liczba punktów danych.

5. Zwrócenie wartości współczynnika korelacji: Ostatecznie, obliczony współczynnik korelacji `r` jest zwracany jako wynik funkcji.

```
    double a0, a1;
    linear_approx(M, rozmiar, a0, a1);

    cout << "\nWyniki aproksymacji:\n";
    cout << "a1 = " << a1 << ", a0 = " << a0 << endl;

    double r = correlation_coefficient(M, rozmiar, a0, a1);
    cout << "Wspolczynnik korelacji: " << r << endl;

    // Zwolnij pamięć
    for (int i = 0; i < rozmiar; ++i) {
        delete[] M[i];
    }
    delete[] M;

    return 0;
```

(4. Fragment kodu odpowiedzialny za wypisywanie wyników i zwolnienie pamięci)

(4) Ten fragment kodu jest odpowiedzialny za:

1. Wywołanie funkcji `linear_approx` w celu obliczenia współczynników a0 i a1 dla aproksymacji liniowej na podstawie danych zawartych w tablicy dwuwymiarowej `M`.

2. Wyświetlenie wyników aproksymacji na standardowym wyjściu za pomocą instrukcji wyjścia `cout`. Wyświetla wartości współczynników a0 i a1.

3. Wywołanie funkcji `correlation_coefficient` w celu obliczenia współczynnika korelacji (r) między zmiennymi x i y na podstawie współczynników a0 i a1 oraz danych zawartych w tablicy dwuwymiarowej `M`.

4. Wyświetlenie wyniku współczynnika korelacji na standardowym wyjściu za pomocą instrukcji wyjścia `cout`.

5. Zwolnienie pamięci, która została zaalokowana na tablicę dwuwymiarową `M` za pomocą instrukcji `delete`.

6. Zakończenie programu zwracając wartość 0.

Testy jednostkowe

Test 1

Dane wejściowe programu:

7

4 1

6 4

8 5

10 9

12 9

14 12

16 12

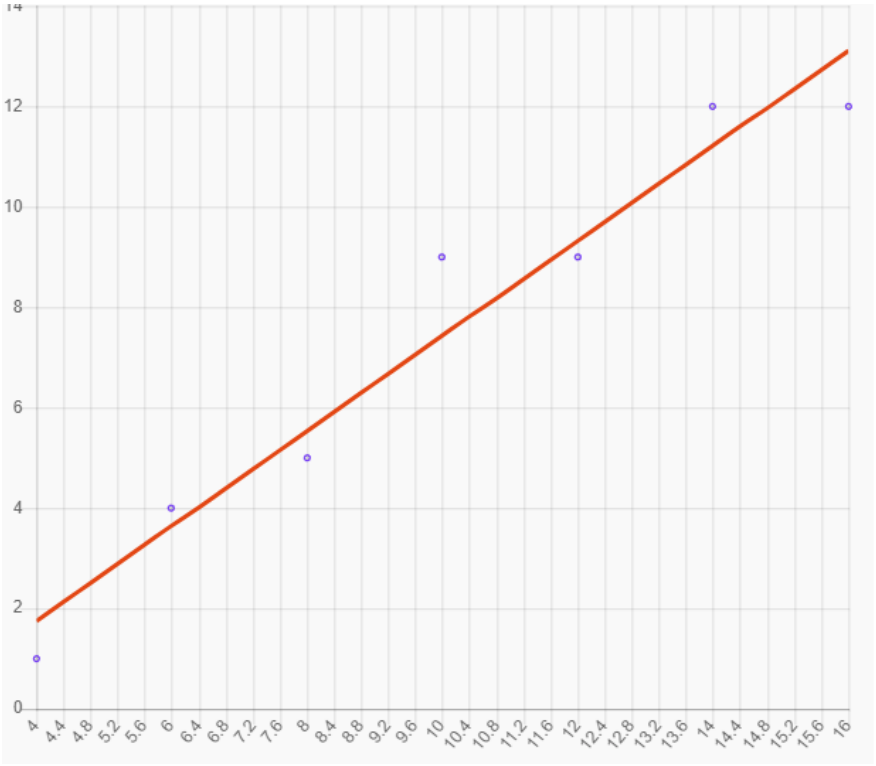
```
7
4 1
6 4
8 5
10 9
12 9
14 12
16 12

Wyniki aproksymacji:
a1 = 0.946429, a0 = -2.03571
Wspolczynnik korelacji: 0.974159
```

(1. Wynik programu)

(1) Wyniki aproksymacji:

a1 = 0.946429, a0 = -2.03571

Wspolczynnik korelacji: 0.974159



(Wykres obrazujący przypadek testowy 1)

Test 2

Dane wejściowe programu:

5

4 4

6 6

8 8

10 10

12 12

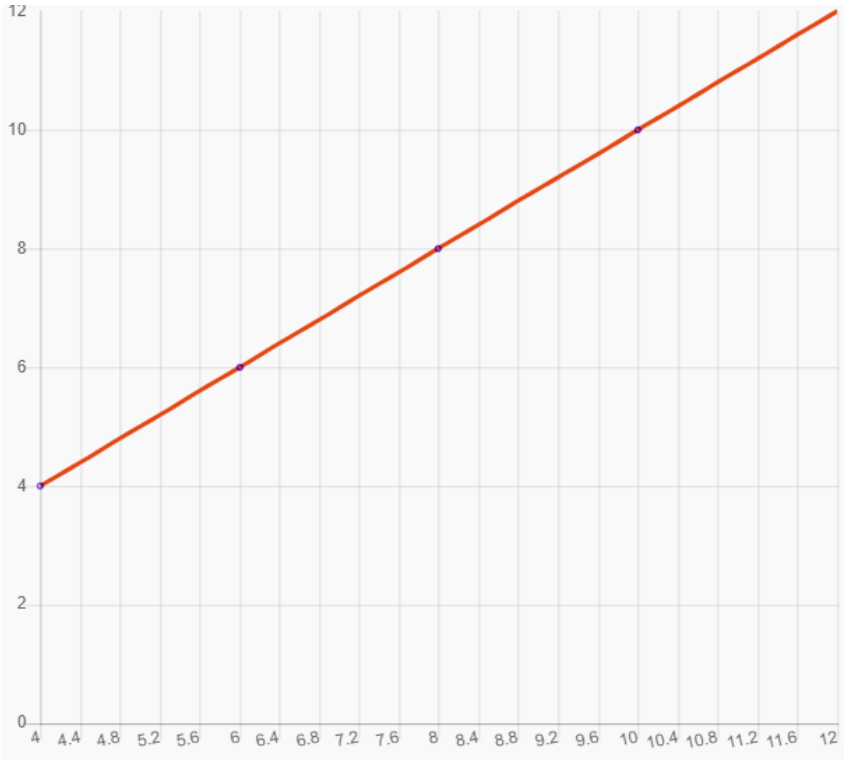
```
5
4 4
6 6
8 8
10 10
12 12

Wyniki aproksymacji:
a1 = 1, a0 = 0
Wspolczynnik korelacji: 1
```

(2. Wynik programu)

Wyniki aproksymacji:

a1 = 1, a0 = 0

Wspolczynnik korelacji: 1



(Wykres obrazujący przypadek testowy 2)

Test 3

Dane wejściowe programu:

5

1 0

2 1

3 3.5

3.5 4

5 7

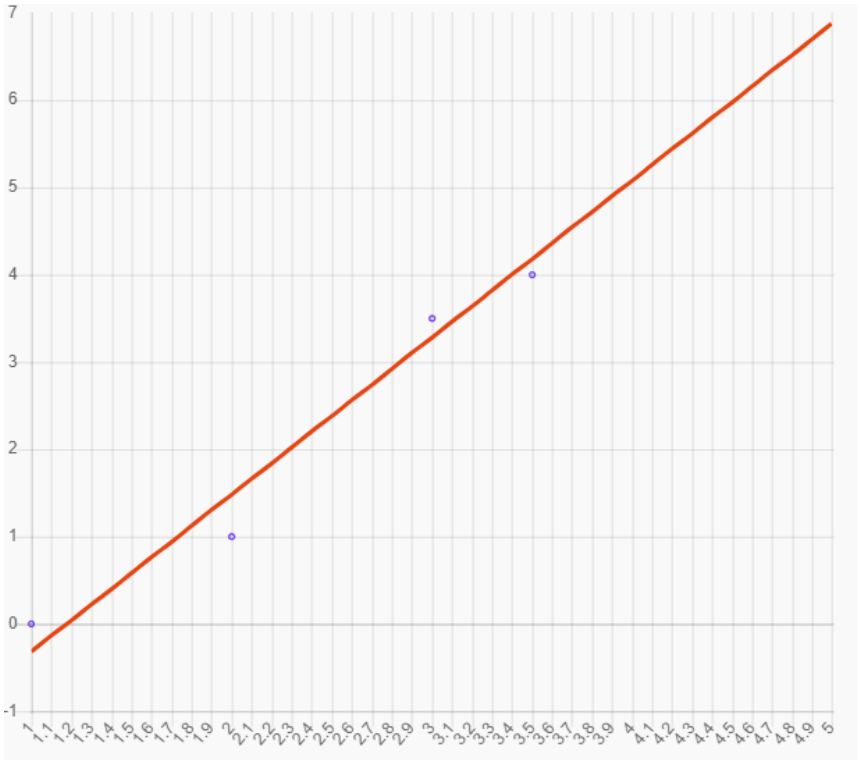
```
5
1 0
2 1
3 3.5
3.5 4
5 7

Wyniki aproksymacji:
a1 = 1.79891, a0 = -2.11685
Wspolczynnik korelacji: 0.992889
```

(3. Wynik programu)

Wyniki aproksymacji:

a1 = 1.79891, a0 = -2.11685

Wspolczynnik korelacji: 0.992889



(Wykres obrazujący przypadek testowy 3)

Opracowanie wyników testów

Poniższa analiza prezentuje wyniki testów przeprowadzonych na implementacji modelu regresji liniowej na wybranych zestawach danych. Dokonano porównania tych wyników z analogicznymi wynikami uzyskanymi za pomocą programu Microsoft Excel, który posłużył jako standardowy punkt odniesienia. Analiza przeprowadzonych testów wykazała wysoki poziom skuteczności metody najmniejszych kwadratów w aproksymacji danych.

Numer testu	a_1 EXCEL	a_0 EXCEL	R EXCEL	a_1	a_0	R
1	0.946429	-2.03571	0,9741663	0.946429	-2.03571	0.974159
2	1	0	1	1	0	1
3	1.79891	-2.11685	0.992889	1.79891	-2.11685	0.992889

Wnioski

Metoda najmniejszych kwadratów jest efektywnym narzędziem do przybliżania danych numerycznych poprzez dopasowanie funkcji do zestawu danych. Jej głównym celem jest minimalizacja sumy kwadratów różnic pomiędzy wartościami rzeczywistymi a wartościami przybliżonymi.

Współczynnik korelacji (R) odgrywa kluczową rolę w ocenie jakości dopasowania funkcji aproksymującej do danych. Kiedy jego wartość zbliża się do 1, oznacza to silną korelację między zmiennymi, co sugeruje wysoką jakość dopasowania.

Implementacja metody najmniejszych kwadratów w programie komputerowym umożliwia szybkie i efektywne przetwarzanie danych oraz modelowanie funkcji. Program pozwala na automatyczne obliczanie parametrów aproksymacji oraz współczynnika korelacji, co ułatwia pracę badawczą i inżynieryjną.

Porównanie wyników uzyskanych z programu z wynikami uzyskanymi za pomocą innych narzędzi, takich jak arkusz kalkulacyjny Excel, umożliwia weryfikację poprawności działania programu i zwiększa wiarygodność analizy danych.

Dzięki przeprowadzeniu tego ćwiczenia zdobyliśmy praktyczne umiejętności w analizie danych oraz modelowaniu funkcji. Pozwoliło nam to również na lepsze zrozumienie zasad działania metody najmniejszych kwadratów oraz jej praktycznych zastosowań w różnych dziedzinach, takich jak nauki techniczne i inżynieryjne.

Wiedza i doświadczenie zdobyte podczas tego ćwiczenia będą wartościowe w naszej dalszej pracy badawczej oraz inżynieryjnej, szczególnie w kontekście analizy danych i tworzenia modeli numerycznych. Pomogą nam one skutecznie wykorzystać metodę najmniejszych kwadratów w praktyce do analizy, interpretacji danych oraz prognozowania zachowań systemów i procesów.