

Laboratorium 11 - Optymalizacja

Błażej Naziemiec i Szymon Żuk

17 czerwca 2025

Wstęp

W tym laboratorium zajmiemy się optymalizacją kosztu obliczeń funkcji. W tym celu wykonamy dwa zadania. Pierwsze z nich polega na porównaniu rozwiązania problemu predykcji typu komórek nowotworowych za pomocą metody najmniejszych kwadratów oraz regresji liniowej. W drugim zadaniu musimy stworzyć algorytm, który jak najmniejszym kosztem znajdzie ścieżkę dla robota z punktu startowego do punktu końcowego.

Zadanie 1

Na początku zaimportowaliśmy dane odnośnie komórek nowotworowych, które otrzymaliśmy w laboratorium 2. Następnie skopiowaliśmy kod z laba 2 do wyznaczania predykcji metodą najmniejszych kwadratów. Następnie zaimplementowaliśmy funkcję rozwiązującą ten problem metodą spadku wzdłuż gradientu. Przyjeliśmy $\alpha = \frac{1}{\lambda_{\max} + \lambda_{\min}}$, gdzie λ to wartość własna macierzy $A^T * A$. Gradient $\nabla f(x_k)$ obliczyliśmy ze wzoru $2A^T(Ax - b)$. Wyniki oraz czasy wykonania przedstawiliśmy poniżej.

Wyniki otrzymane z kodu z laboratorium 2:

Czas wykonania: 0.0566 sekund

```
((58, 194, 6, 2, 0.9692307692307692),  
 (55, 185, 15, 5, 0.9230769230769231),  
 (58, 194, 6, 2, 0.9692307692307692),  
 (55, 199, 1, 5, 0.9769230769230769))
```

Rodzaj metody	true positive	true negative	false positive	false negative	accuracy
Liniowa	58	194	6	2	96.92%
Kwadratowa	55	185	15	5	92.31%
Lambda	58	194	6	2	96.92%
SVD	55	199	1	5	97.69%

Tabela 1. Wyniki predykcji typu komórek nowotworowych dla różnych macierzy metody najmniejszych kwadratów

Analizując wyniki z tabeli 1 możemy zauważyć, że najlepsze wyniki uzyskaliśmy dla macierzy SVD i jego dokładność wynosi 97.69%. Następnie wykonaliśmy operację regresji liniowej dla tych samych danych, a wyniki przedstawiliśmy w tabeli 2.

Wyniki otrzymane z gradientu:

Czas wykonania: 3.5047 sekund

((58, 177, 23, 2, 0.9038461538461539), 100000)

true positive	true negative	false positive	false negative	accuracy	liczba iteracji
58	177	23	2	90.38%	100000

Tabela 2. Wyniki spadku wzdłuż gradientu dla danych komórek nowotworowych

Z tabeli 2 wynika, że dokładność wynosi 90.38%, co jest gorszym wynikiem niż w przypadku metody najmniejszych kwadratów dla każdego typu macierzy. W związku z tym możemy stwierdzić, że metoda najmniejszych kwadratów jest lepsza w tym przypadku. Warto również zwrócić uwagę na teoretyczne złożoności czasowe obu rozwiązań. Metoda najmniejszych kwadratów ma teoretyczną złożoność obliczeniową $O(nm^2 + m^3)$, a spadek wzdłuż gradientu $O(nm \log(\epsilon))$. W praktyce metoda najmniejszych kwadratów była znacznie szybsza, pomimo tego, że obliczono w niej wynik na 4 różne sposoby.

Zadanie 2

Zaimplementowaliśmy funkcję kosztu i funkcję obliczającą gradient funkcji kosztu. Następnie zainicjalizowaliśmy dane początkowe. Macierze X i R wypełniliśmy danymi z rozkładu normalnego za pomocą funkcji `numpy.random.uniform`. Potem zaimplementowaliśmy funkcję minimalizującą koszt metodą spadku wzdłuż gradientu. W implementacji wykorzystaliśmy

funkcję *scipy.optimize.golden* do przeszukiwania liniowego metodą złotego podziału. Potem utworzyliśmy wykresy otrzymanych wartości funkcji kosztu oraz macierzy X i R .

Podpunkt 1). Wyznaczenie $\nabla F = [\frac{\partial F}{\partial \mathbf{x}_0}, \dots, \frac{\partial F}{\partial \mathbf{x}_n}]$

$$F(\mathbf{x}) = \lambda_1 \sum_{i=0}^{n-1} \sum_{j=1}^{k-1} \frac{1}{\epsilon + \|\mathbf{x}_i - \mathbf{r}_j\|^2} + \lambda_2 \sum_{i=0}^{n-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2$$

$$\nabla F(\mathbf{x}) = \left[\frac{\partial F}{\partial \mathbf{x}_0}, \frac{\partial F}{\partial \mathbf{x}_1}, \dots, \frac{\partial F}{\partial \mathbf{x}_n} \right]$$

$$F(\mathbf{x}) = F_1(\mathbf{x}) + F_2(\mathbf{x})$$

$$F_1(\mathbf{x}) = \lambda_1 \sum_{i=0}^{n-1} \sum_{j=1}^{k-1} \frac{1}{\epsilon + \|\mathbf{x}_i - \mathbf{r}_j\|^2}$$

$$F_2(\mathbf{x}) = \lambda_2 \sum_{i=0}^{n-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\|^2$$

$$\frac{\partial}{\partial \mathbf{x}_t} \left(\frac{1}{\epsilon + \|\mathbf{x}_t - \mathbf{r}_j\|^2} \right) = - \frac{1}{(\epsilon + \|\mathbf{x}_t - \mathbf{r}_j\|^2)^2} \cdot 2(\mathbf{x}_t - \mathbf{r}_j)$$

$$\frac{\partial F_1}{\partial \mathbf{x}_t} = -2\lambda_1 \sum_{j=1}^{k-1} \frac{\mathbf{x}_t - \mathbf{r}_j}{(\epsilon + \|\mathbf{x}_t - \mathbf{r}_j\|^2)^2}$$

$$\frac{\partial}{\partial \mathbf{x}_t} \|\mathbf{x}_{t+1} - \mathbf{x}_t\|^2 = -2(\mathbf{x}_{t+1} - \mathbf{x}_t)$$

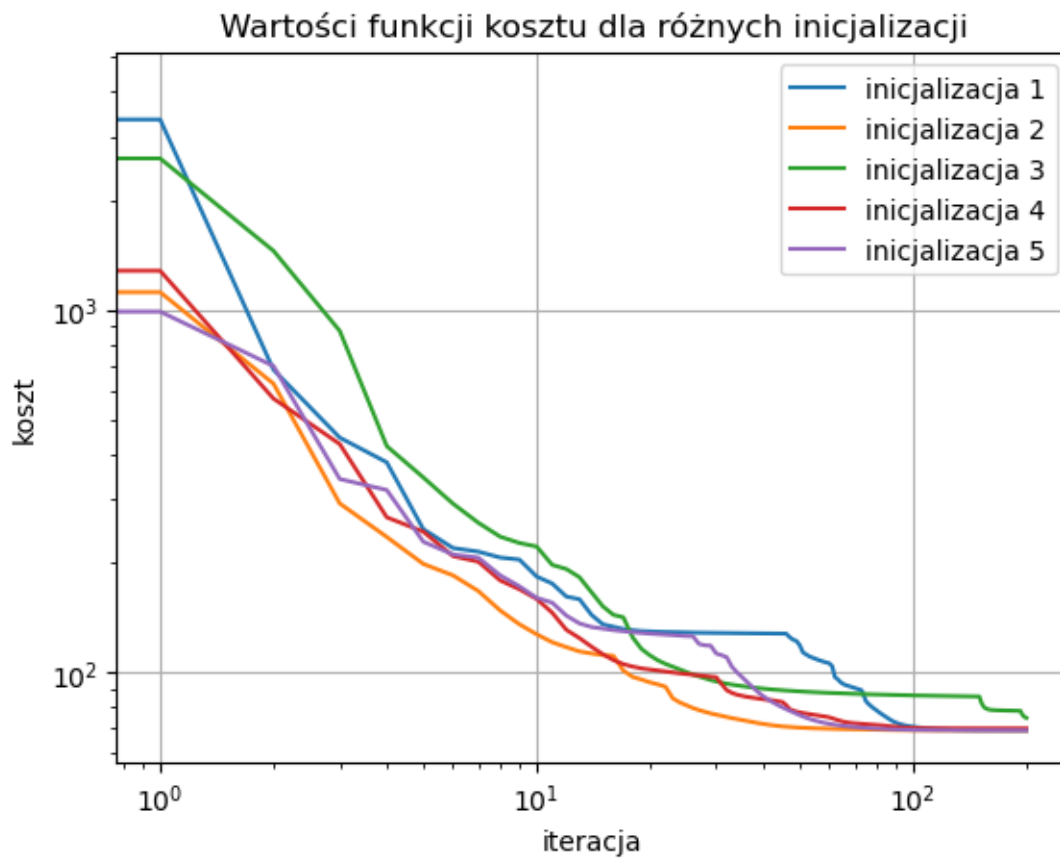
$$\frac{\partial}{\partial \mathbf{x}_t} \|\mathbf{x}_t - \mathbf{x}_{t-1}\|^2 = 2(\mathbf{x}_t - \mathbf{x}_{t-1})$$

$$\frac{\partial F_2}{\partial \mathbf{x}_t} = 2\lambda_2 ((\mathbf{x}_t - \mathbf{x}_{t-1}) - (\mathbf{x}_{t+1} - \mathbf{x}_t)) = 2\lambda_2 (2\mathbf{x}_t - \mathbf{x}_{t-1} - \mathbf{x}_{t+1})$$

$$\frac{\partial F}{\partial \mathbf{x}_t} = -2\lambda_1 \sum_{j=1}^{k-1} \frac{\mathbf{x}_t - \mathbf{r}_j}{(\epsilon + \|\mathbf{x}_t - \mathbf{r}_j\|^2)^2} + 2\lambda_2 (\mathbf{x}_t - \mathbf{x}_{t-1} - \mathbf{x}_{t+1}) \text{ dla } t = 1, 2, \dots, n-1$$

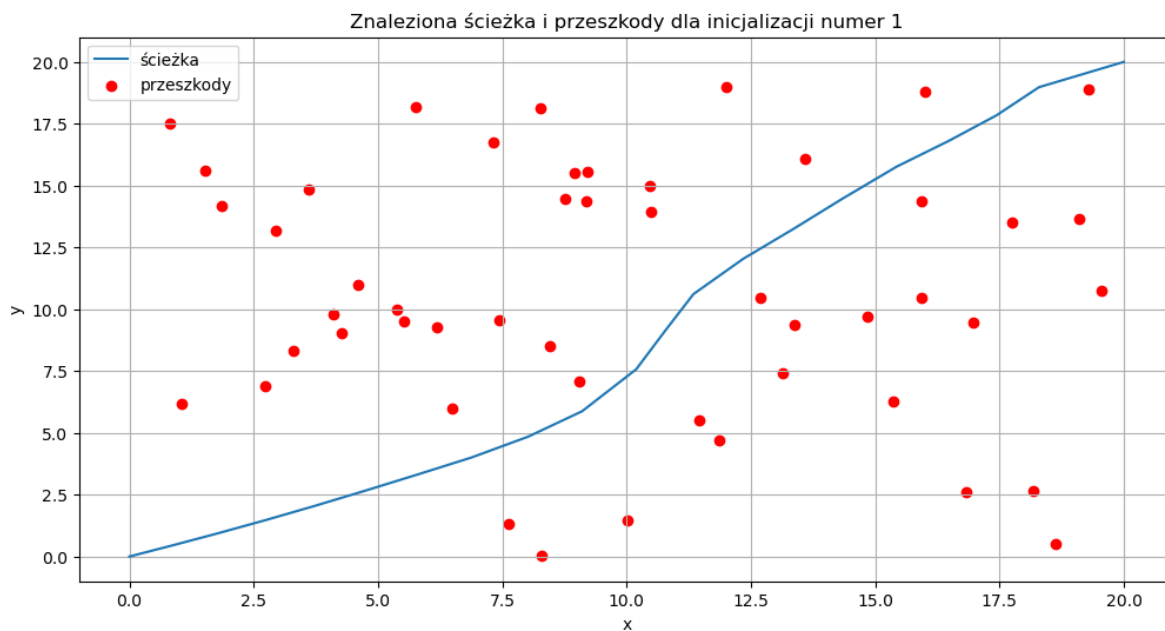
Podpunkt 2). Algorytm rozpoczynamy od zainicjalizowania punktów wartościami z początkowego przedziału, z ustalonymi punktami początkowym i końcowym. Następnie, w każdej iteracji, obliczany jest gradient funkcji kosztu, który wskazuje kierunek największego spadku kosztu. Potem znajdujemy optymalną długość kroku wzdłuż kierunku gradientu, za pomocą metody złotego podziału, minimalizując funkcję kosztu. Wyznaczona długość kroku jest następnie używana do aktualizacji pozycji wszystkich punktów na ścieżce, z wyjątkiem punktów początkowego i końcowego, które pozostają niezmiennie. Proces ten powtarza się przez określoną liczbę iteracji lub do momentu, gdy poprawa kosztu spadnie poniżej ustalonego progu tolerancji.

Podpunkt 3). Wyniki działania algorytmu

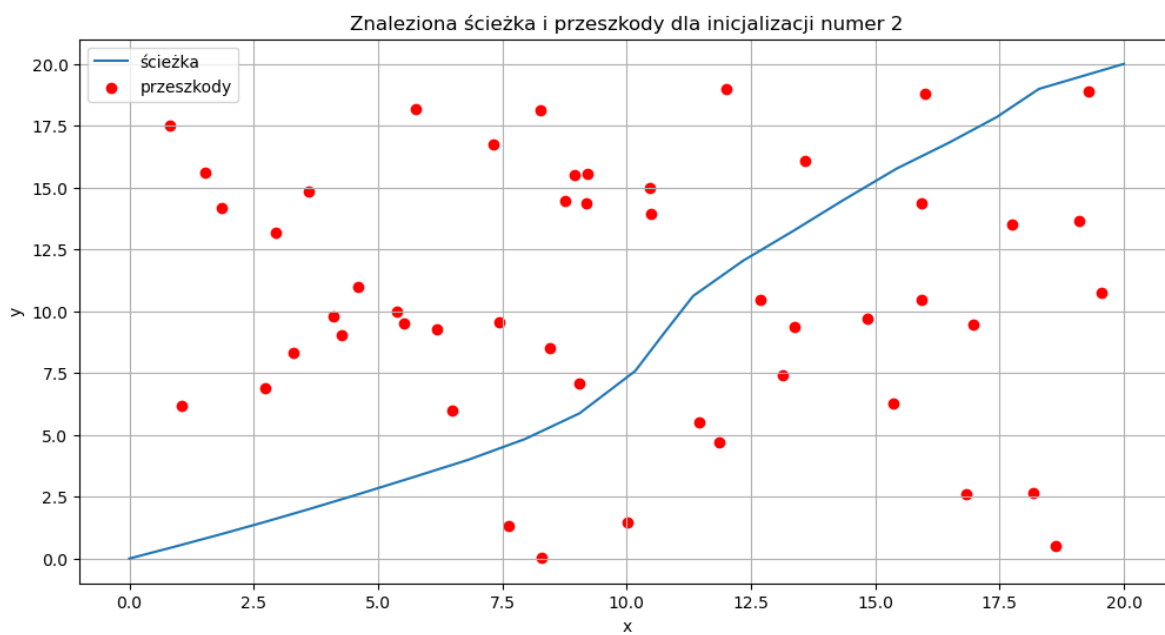


Wykres 1. Wartości funkcji kosztu dla różnych inicjalizacji

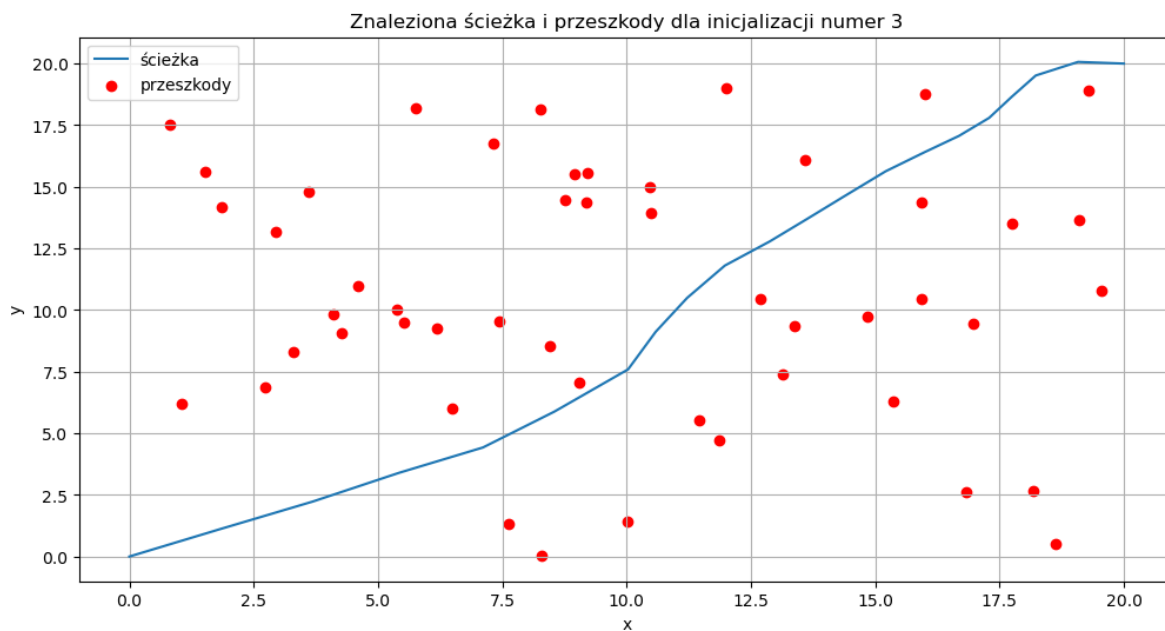
Jak widać na wykresie wartości funkcji kosztu maleją w podobny sposób dla każdej inicjalizacji. Początkowa losowa wartość macierzy X daje koszt większy niż 1000. Koszt ten ostatecznie spada poniżej 100.



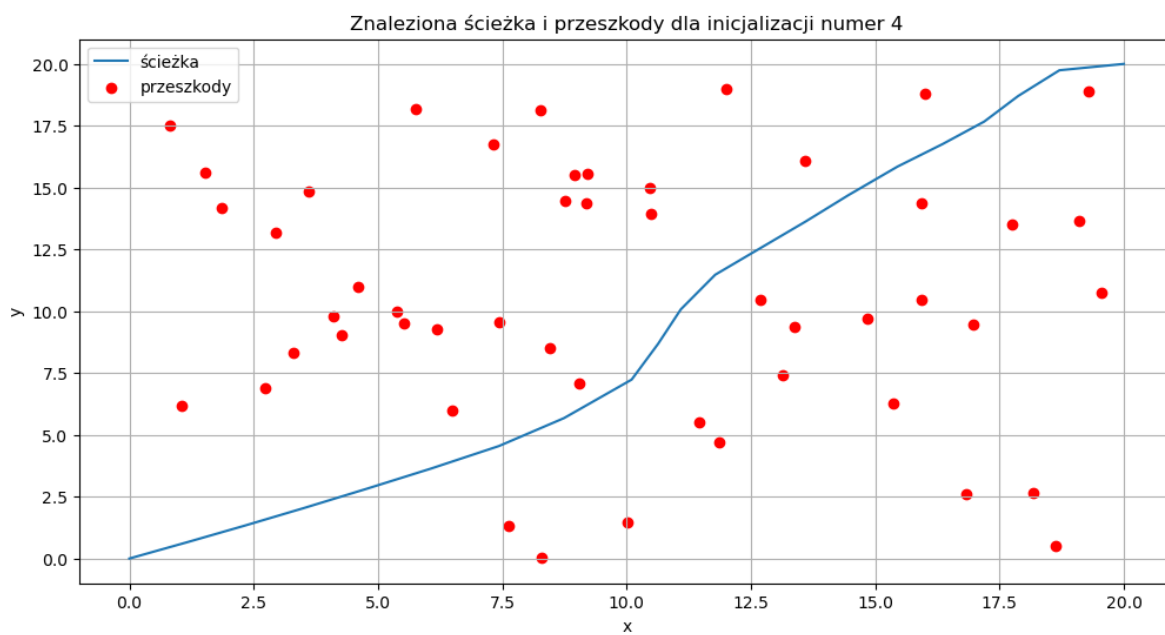
Wykres 2: Znaleziona ścieżka i przeszkody dla inicjalizacji numer 1



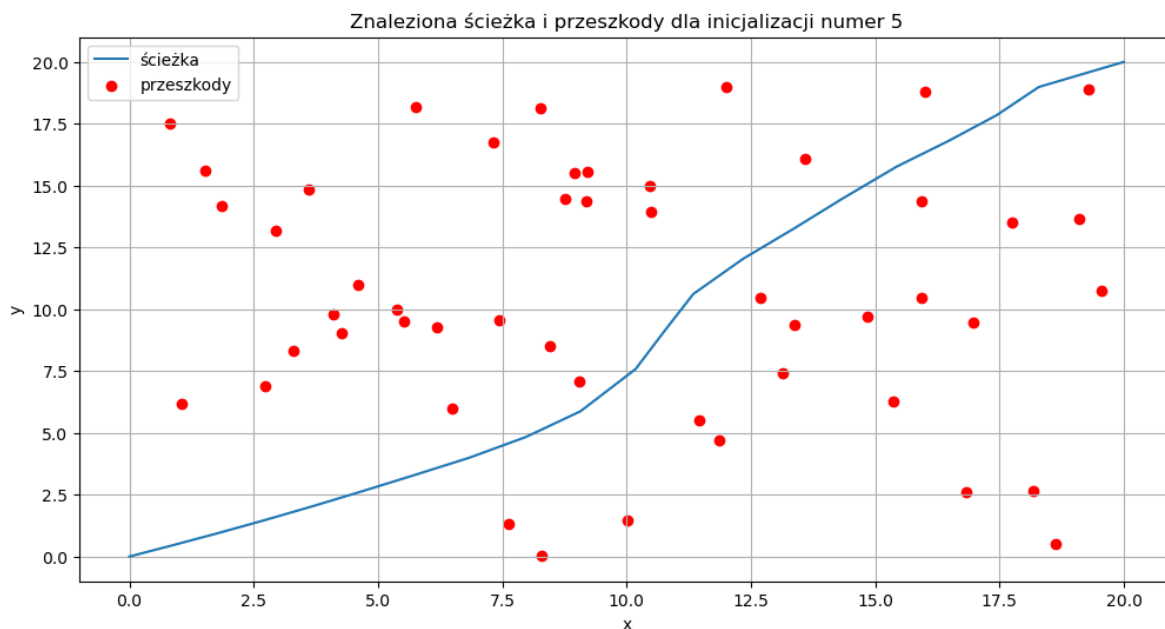
Wykres 3: Znaleziona ścieżka i przeszkody dla inicjalizacji numer 2



Wykres 4: Znaleziona ścieżka i przeszkody dla inicjalizacji numer 3



Wykres 5: Znaleziona ścieżka i przeszkody dla inicjalizacji numer 4



Wykres 6: Znalezione ścieżki i przeszkody dla inicjalizacji numer 5

Jak widać na wykresach, otrzymane ścieżki nieznacznie różnią się w zależności od inicjalizacji macierzy X i są optymalne. Pokazuje to efektywność algorytmu spadku wzdłuż gradientu.

Podsumowując algorytm spadku wzdłuż gradientu jest efektywną metodą optymalizacji. W pierwszym zadaniu okazał się on mniej dokładny i wolniejszy od metody najmniejszych kwadratów, lecz dał w miarę akceptowalne wyniki. W drugim zadaniu algorytm dał bardzo optymalne wyniki niezależnie od punktu startowego.

Bibliografia

- Materiały zamieszczone na platformie Microsoft Teams w zespole *MOwNiT 2025* w zakładce *Materiały z zajęć/lab11/lab11-intro.pdf*