Metody numeryczne

Sprawozdanie 5

Diagonalizacja macierzy symetrycznej metodą potęgową z redukcją Hotellinga.

Kateryna Andrusiak

3 kwietnia 2020

1. Wstęp teoretyczny

Metoda potęgowa -metoda wielokrotnego przybliżania, stosowana do wyznaczania pojedynczych wartości własnych i wektorów własnych. Ta metoda działa w następujący sposób:

Załóżmy, że istnieje n liniowo niezależnych wektorów własnych macierzy A, stanowią bazę przestrzeni liniowej

$$\vec{x}_1, \vec{x}_2, \vec{x}_3, \dots, \vec{x}_n \tag{1}$$

Wówczas dla dowolnego wektora $\vec{v}_0$

$$\vec{v}_0 = \sum_{i=1}^{n} a_i \vec{x}_i \tag{2}$$

Jeśli λ_i stanowią wartości własne macierzy

$$A\vec{v}_0 = \sum_{i=1}^{n} a_i \lambda_i \vec{x}_i \tag{3}$$

$$\vec{v}_m = A^m \vec{v}_0 = \sum_{i=1}^{n} a_i \lambda_i \vec{x}_i \tag{4}$$

Zakładamy, że wartości własne tworzą ciąg

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \cdots \geq |\lambda_n| \tag{5}$$

Jeśli λ_1 jest dominującą wartością własną, oraz wektor $\vec{v}_0$ ma składową w kierunku $\vec{x}_1$ to wówczas zachodzi

$$\lim_{m\to\infty} \frac{A^m \vec{v}_0}{\lambda_1^m} = a_1 \vec{x}_1 \tag{6}$$

Z czego można wysnuć wniosek, że wartość własną można obliczyć następująco

$$\lambda_1 = \lim_{m\to\infty} \frac{\vec{y}^T \vec{v}_{m+1}}{\vec{y}^T \vec{v}_m} \tag{7}$$

Dla dowolnego wektora y nieortogonalnego do $\vec{x}_1$.

Zazwyczaj $\vec{y}$ ma 1 na pozycji elementu o największym module w $\vec{v}_{m+1}$ a na pozostałych 0.

<u>Zbieżność metody</u> zależy od $\left(\frac{\lambda_i}{\lambda_1}\right)^m$ ale również od współczynników a_i czyli od wyboru $\vec{v}_0$.

$$\vec{v}_m = \lambda_1^m \left[a_1 \vec{x}_1 + \sum_{i=2}^{n} \left(\frac{\lambda_i}{\lambda_1} \right)^m a_i \vec{x}_i \right] \tag{8}$$

Jeśli wartość własna o największym module jest zespolona to ciąg nie jest zbieżny.

Wyznaczenie wektora własnego $\vec{x}_1$. Ponieważ $\vec{v}_m \approx \lambda_1^m a_1 \vec{x}_1$, więc unormowany wektor własny będzie miał postać

$$\vec{x}_1 = \frac{\vec{v}_m}{|\vec{v}_m|} \tag{9}$$

Jeśli wartość własna jest pierwiastkiem wielokrotnym równania charakterystycznego to metoda jest zbieżna, bo składnik z λ_1 dominuje

$$\vec{v}_m = A^m \vec{v}_0 = \lambda_1^m \sum_{i=1}^{k} a_i \vec{x}_i + \sum_{i=k+1}^{n} \lambda_i^m a_i \vec{x}_i \tag{10}$$

Redukcja Hotellinga.

Za wektor $\vec{v}$ przyjmujemy lewy wektor własny przynależny do wartości własnej λ_1 . Ale na ogół nie znamy lewych wektorów.

Metoda jest więc skuteczna tylko w przypadku macierzy symetrycznych, wtedy lewe wektory są identyczne z prawymi.

$$\vec{v} = \vec{x}_1 \tag{11}$$

$$W_1 = A - \lambda_1 \vec{x}_1 \vec{x}_1^T \tag{12}$$

lub rekurencyjnie

$$W_0 = A \tag{13}$$

$$W_i = W_{i-1} - \lambda_{i-1} \vec{x}_{i-1} \vec{x}_{i-1}^T \qquad i = 1,2,\dots,n-1 \tag{14}$$

gdzie $\vec{x}(\vec{x})^T$ jest iloczynem zewnętrznym/tensorowym.

2. Problem

Naszym zadaniem było rozwiązanie macierzowego problemu własnego

$$A\vec{x} = \lambda\vec{x} \qquad \{\lambda_1, \lambda_2, \dots, \lambda_n\}, \qquad \{\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n\} \tag{15}$$

przy użyciu metody iteracyjnej.

W ogólnym przypadku (macierz niesymetryczna) iteracyjne wyznaczanie wartości i wektorów własnych wymaga zastosowania zaawansowanych metod np. metody Arnoldiego zaimplementowanej w pakiecie ARPACK.

Jeśli jednak macierz jest symetryczna, wówczas możemy użyć prostej metody potęgowej. W wersji podstawowej metoda pozwala iteracyjnie wyznaczać pojedynczą wartość i odpowiadający jej wektor własny, ale po modyfikacji np. tzw. redukcji Hotellinga umożliwia wyznaczanie kolejnych par $(\lambda_i, \vec{x}_i)$.

Naszym zadaniem było zaimplementowanie tej metody i wyznaczenie po kolei wszystkich par $(\lambda_i, \vec{x}_i)$.

Najpierw tworzymy macierz A, w naszym przypadku musi ona być symetryczna:

$$A_{ij} = \frac{1+|i+j|}{1+|i-j|} \qquad i,j = 0,1,\dots,n-1 \tag{16}$$

Wartości własne wyznaczymy iteracyjnie, przy użyciu metody potęgowej (korzystając z redukcji macierzy Hotellinga) zgodnie z poniższym algorytmem

```
Utwórz macierze: A, W, X
Utwórz wektory: x_old, x_new
 W = A (inicjalizacja macierzy iterującej W – będziemy modyfikować)
   for(k=0; k < K_val; k++){
        x_old = [1,1,...,1] (inicjalizacja wektora startowego)
        x_old = x_old / ||x_old||_2 (normalizacja wektora)
        for(m=1; m <= IT_MAX; m++){
            x_new = W x_old
            λ_k = (x_new)^T x_old
            x_old = x_new / ||x_new||_2 (normalizacja wektora)
        }
        W = W − λ_k x_old (x_old)^T (iloczyn zewnętrzny/tensorowy)
        X_{*,k} = x_old (zachowujemy wektor własny x_old w k−tej kolumnie macierzy X)
   }
```

gdzie:

- k - numer wyznaczanej wartości własnej
- i - numer iteracji dla określonego k
- A - macierz pierwotna
- W - macierz iteracji (podlega modyfikacji)
- λ_k - przybliżenie k-tej wartości własnej w m-tej iteracji
- $\boldsymbol{x}_{new}$ - m-te przybliżenie k-tego wektora własnego
- $K_{val} = n$ - liczba wartości własnych do wyznaczenia
- $IT\_MAX = 12$ - maksymalna liczba iteracji dla każdego k

W kolumnach macierzy X zachowujemy wyznaczone wektory własne:

$$X = [\vec{x}_0, \vec{x}_1, \dots, \vec{x}_{n-1}] \tag{17}$$

Nastepnie wyznaczamy postać macierzy D zdefiniowanej jako iloczyn:

$$D = X^T A X \tag{18}$$

3. Wyniki

Macierz D otrzymana po 12 iteracjach:

$$\boldsymbol{D} = \begin{bmatrix} 24.5585 & -7.84263e-05 & -5.55944e-08 & -4.79673e-10 & -6.58698e-12 & -1.77636e-14 & -1.12688e-14 \\ -7.84263e-05 & 8.85168 & -0.042035 & -0.000362672 & -4.97635e-06 & -1.18577e-08 & -1.80828e-14 \\ -5.55944e-08 & -0.042035 & 5.86619 & -0.0672484 & -0.000896966 & -2.13685e-06 & -2.77844e-12 \\ -4.79673e-10 & -0.000362672 & -0.0672484 & 4.29449 & -0.054224 & -0.000122061 & -1.58562e-10 \\ -6.58708e-12 & -4.97635e-06 & -0.000896966 & -0.054224 & 3.05855 & -0.00709218 & -8.86873e-09 \\ -1.71252e-14 & -1.18577e-08 & -2.13685e-06 & -0.000122061 & -0.00709218 & 1.81862 & -4.19876e-06 \\ -1.1522e-14 & -1.8101e-14 & -2.77851e-12 & -1.58561e-10 & -8.86873e-09 & -4.19876e-06 & 0.552968 \end{bmatrix}$$

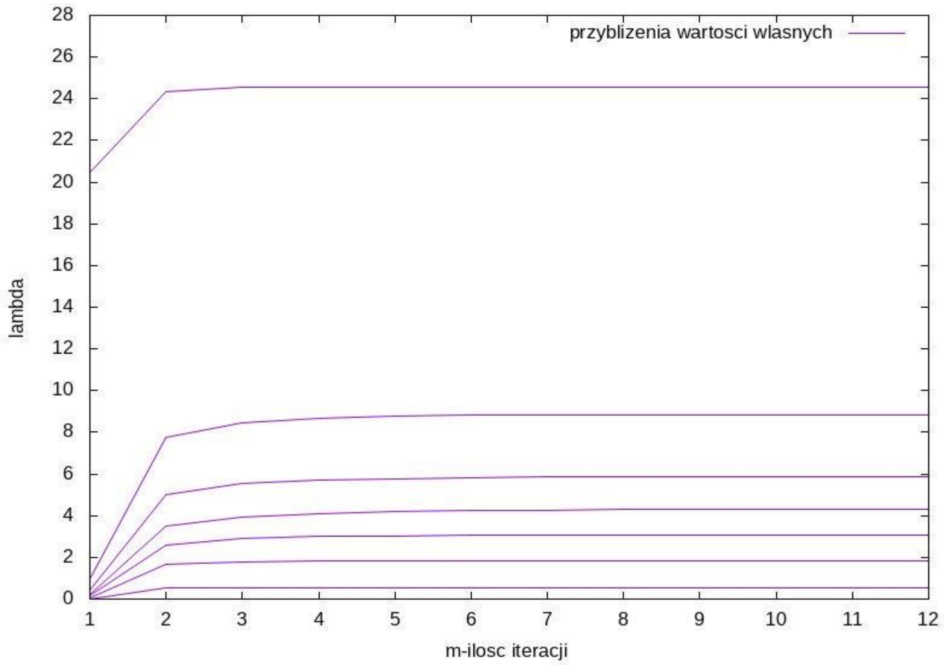
Otrzymane wartości własne: 24.5585, 8.85168, 5.86619, 4.29449, 3.05855, 1.81862, 0.552968

Jak widać, macierz D jest symetryczna, na diagonali otrzymaliśmy wartości własne, elementy pozadiagonalne są bliskie zeru.

Zwiększając liczbę iteracji, poprawiamy dokładność i D robi się diagonalna. Dla IT_Max=30 (liczba iteracji) otrzymaliśmy macierz jeszcze bardziej przybliżoną do diagonalnej.

Proces iteracyjny powinien zatrzymać się sam, jeśli wszystkie elementy pozadiagonalne są równe zeru lub zakończyć pracę po ustalonej liczbie iteracji.

Rysunek 1. Kolejne przybliżenia znalezionych wartości własnych.



Jak widać z rysunku już po drugiej iteracji przybliżenie wartości własnych staje się bliskie oczekiwanej wartości.

Wartości własne stabilizują się różnie, na przykład (dla k=0 oraz k=1):

kolejno: nr wartości, nr iteracji, przybliżona wartość

0 1 20.485714
0 2 24.327365
0 3 24.534951
0 4 24.555684
0 5 24.558121
0 6 24.558425
0 7 24.558464
0 8 24.558469 ← tu widać stabilizację wyniku na 8 miejscu znaczącym
0 9 24.558469
0 10 24.558469
0 11 24.558469
0 12 24.558469

1 1 0.975911
1 2 7.732824
1 3 8.435153
1 4 8.682327
1 5 8.780323
1 6 8.821095
1 7 8.838471
1 8 8.845972
1 9 8.849235
1 10 8.850661 ← tu widać stabilizację wyniku na 3 miejscu znaczącym
1 11 8.851285
1 12 8.851559

4. Wnioski

Przy użyciu metody potęgowej (korzystając z redukcji macierzy Hotellinga) iteracyjnie wyznaczyliśmy wartości własne oraz wektory własne i rozwiązaliśmy macierzowy problem własny. Metoda potęgowa, jest metodą, która powinna zwracać tym bardziej precyzyjne wyniki im więcej iteracji zadamy do wykonania programowi. Zwiększając liczbę iteracji, otrzymujemy co raz bardziej dokładne wartości własne.

Wykorzystanie metody potęgowej razem z redukcją Hotellinga pozwala nam znaleźć żądaną ilość par wartości oraz wektorów własnych, ale w tylko w tym przypadku, gdy macierz jest symetryczna.