Diagonalizacja macierzy symetrycznej metodą potegową z redukcją Hotellinga.

Tomasz Chwiej

31 marca 2020

1 Wprowadzenie

Naszym zadaniem jest rozwiązanie macierzowego problemu własnego

$$A\boldsymbol{x} = \lambda \boldsymbol{x}, \qquad \{\lambda_1.\lambda_2, \ldots, \lambda_n\}, \quad \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \tag{1}$$

przy użyciu metody iteracyjnej. W ogólnym przypadku (macierz niesymetryczna) iteracyjne wyznaczanie wartości i wektorów wsłanych wymaga zastosowania zaawansowanych metod np. metody Arnoldiego zaimplementowanej w pakiecie ARPACK (pakiet ten stworzony został 25 lat temu i do tej pory oprócz może pakietu FEAST nie wymyślono nic lepszego). Jeśli jednak macierz jest symetryczna, wóczas możemy użyć prostej metody potęgowej. Podstawy metody opisane są w wykładzie ("Wyznaczanie wartości i wektorów własnych macierzy") na stronie 9 i 10. W wersji podstawowej metoda pozwala iteracyjnie wyznaczać pojedynczą wartość i odpowiadający jej wektor własny, ale po modyfikacji np. tzw. redukcji Hotellinga (str. 11 wykładu) umożliwia wyznaczanie kolejnych par $(\lambda_i, \boldsymbol{x}_i)$. Naszym zadaniem jest zaimplementowanie tej metody i wyznaczenie po kolei wszystkich par $(\lambda_i, \boldsymbol{x}_i)$. Uwaga: w projekcie NIE WYKORZYSTUJEMY biblioteki numerycznej GSL, wszystkie operacje wykonujemy na macierzach i wektorach, które tworzymy w standardowy sposób w C.

2 Zadania do wykonania

1. Utworzyć macierz symetryczną A o liczbie kolumn/wierszy $n = 7$, której elementy są dane wzorem
$$A_{ij} = \frac{1 + |i + j|}{1 + |i - j|} \tag{2}$$
gdzie: $i, j = 0, 1, \ldots, n - 1$. Macierz jest symetryczna więc ma wszystkie wartości własne rzeczywste, podobnie jak składowe wszystkich wektorów własnych.

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

Utwórz macierze: A, W, X
Utwórz wektory: $\boldsymbol{x}_{old}$, $\boldsymbol{x}_{new}$
$W = A$ (inicjalizacja macierzy iterującej W – będziemy modyfikować)
  for(k=0; $k < K_{val}$; k++){
      $\boldsymbol{x}_{old} = [1, 1, \ldots, 1]$ (inicjalizacja wektora startowego)
      $\boldsymbol{x}_{old} = \dfrac{\boldsymbol{x}_{old}}{\|\boldsymbol{x}_{old}\|_2}$ (normalizacja wektora)
      for(m=1; $m <= IT\_MAX$; m++){
        $\boldsymbol{x}_{new} = W\boldsymbol{x}_{old}$
        $\lambda_k = (\boldsymbol{x}_{new})^T \boldsymbol{x}_{old}$

10 $\boldsymbol{x}_{old} = \dfrac{\boldsymbol{x}_{new}}{\|\boldsymbol{x}_{new}\|_2}$ (normalizacja wektora)
11 }
12 $W = W - \lambda_k \boldsymbol{x}_{old}(\boldsymbol{x}_{old})^T$ (iloczyn zewnętrzny/tensorowy)
13 $X_{*,k} = \boldsymbol{x}_{old}$ (zachowujemy wektor własny $\boldsymbol{x}_{old}$ w k−tej kolumnie macierzy X)
14 }

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.

Działanie algorytmu:
W linii 4 rozpoczyna się zewnętrzna pętla, w której wyznaczamy kolejne (indeks - k) wartości (λ_k) i wketory własne ($\boldsymbol{x}_k = \boldsymbol{x}_{new}$). W linii 7 zaczyna się włąściwa pętla iterująca. W niej poprawaiamy aktualne przybliezenia λ_k i $\boldsymbol{x}_k$, pętla kończy się po IT_MAX iteracjach, ale moglibyśmy ją zakończyć wcześniej, w momencie gdy dwa kolejne przybliżenia λ_k niewiele się różnią. Po zakończeniu wewnętrznej pętli, zachowyjemy wartość i wektor własny (linia 13). Uwaga: gdybyśmy skasowali linię (12) wówczas po wykonaniu kolejnej iteracji k, otrzymalibyśmy ten sam wynik (wartość i wektor własny) bo powtórzylibyśmy obliczenia. Aby temu zapobiec, z macierzy iteracji W musimy usunąć informację o kierunku wektora $\boldsymbol{x}_{k-1}$. W tym celu od macierzy W odejmujemy wyraz $\boldsymbol{x}_k \boldsymbol{x}_k^T$ przemnożony przez λ_k. Wyraz $\boldsymbol{x}_k \boldsymbol{x}_k^T$ jest tzw. iloczynem zewnętrznym (tensorowym - patrz wykład str.4). Jak on działa? Jeśli z lewej strony przemnożymy go przez wektor $\boldsymbol{y}$ to wynik będzie następujący

$$-\lambda_k \boldsymbol{x}_k \underbrace{\left(\boldsymbol{x}_k^T \boldsymbol{y}\right)}_{=c\ =\text{il. skalar.}} = -\lambda_k c \boldsymbol{x}_k \tag{3}$$

czyli wycina on z wektora $\boldsymbol{y}$ (znak '-') wkład pochodzący od wektora (oznaczającego też kierunek) $\boldsymbol{x}_k$. Ten zabieg pozwala usunąć informacje o znalezionych wektorach z macierzy W i poszukiwać kolejne.

3. Dla każdego k zapisać kolejne m przybliżeń wartości własnych λ_k do pliku. W kolumnach macierzy X zachowujemy wyznaczone wektory własne

$$X = [\boldsymbol{x}_0, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n-1}] \tag{4}$$

4. Wyznaczyć postać macierzy D zdefiniowanej jako iloczyn

$$D = X^T A X \tag{5}$$

Macierz D zapisać do pliku - jaką powinna mieć ona postać?

5. W sprawozdaniu przedyskutować kolejność znalezionych wartości własnych, liczbę iteracji potrzebną do znalezienia każdej z nich oraz postać macierzy D. Proszę skomentować wielkość elementów pozadiagonalnych, można sprawdzić jak się one zmienią, gdy maksymalna liczba iteracji wzrośnie np. do $IT\_MAX = 30$. Proces iteracyjny powinien zatrzymać się sam, jeśli spełniony jest odpowiedni warunek (tzw. warunek STOP-u) lub zakończyć pracę po ustalonej liczbie iteracji. Jaki warunek STOP-u moglibyśmy przyjąć w naszym problemie? Sporządzić rysunek, na którym proszę umieścić kolejne przybliżenia znalezionych wartości własnych - w komentarzu proszę napisać jak szybko stabilizują się wartości własne.

3 Uwagi

Do wyznaczania iloczynów: macierz-wektor, wektor-wektor, macierz-macierz oraz modyfikacji macierzy W_{k+1} proszę stworzyć oddzielne funkcje. Dzięki temu kod zyska na przejrzystości. $\|\boldsymbol{x}\|_2$ to norma euklidesowa wektora.

4 Przykładowe wyniki

Trzy wartości własne o największym module to: 24.5585, 8.85168, 5.86604.