Wstęp do metod numerycznych

14. Numeryczne zagadnienie własne

P. F. Góra

http://th-www.if.uj.edu.pl/zfs/gora/

2019

Zagadnienie własne

Definicja: Niech $\mathbf{A} \in \mathbb{C}^{N\times N}$. Liczbę $\lambda \in \mathbb{C}$ nazywam *wartością własną macierzy* $\mathbf{A}$, jeżeli istnieje niezerowy wektor $\mathbf{x} \in \mathbf{C}^N$ taki, że

$$\mathbf{A}\mathbf{x} = \lambda \mathbf{x}\,. \tag{1}$$

Wektor $\mathbf{x}$ nazywam wówczas *wektorem własnym macierzy* $\mathbf{A}$ *do wartości własnej* λ.

Definicja jest sformułowana dla macierzy zespolonych, my jednak — poza przypadkiem macierzy hermitowskich — będziemy zajmować się macierzami rzeczywistymi. Należy jednak pamiętać, że także macierze rzeczywiste (niesymetryczne) mogą mieć zespolone wartości własne.

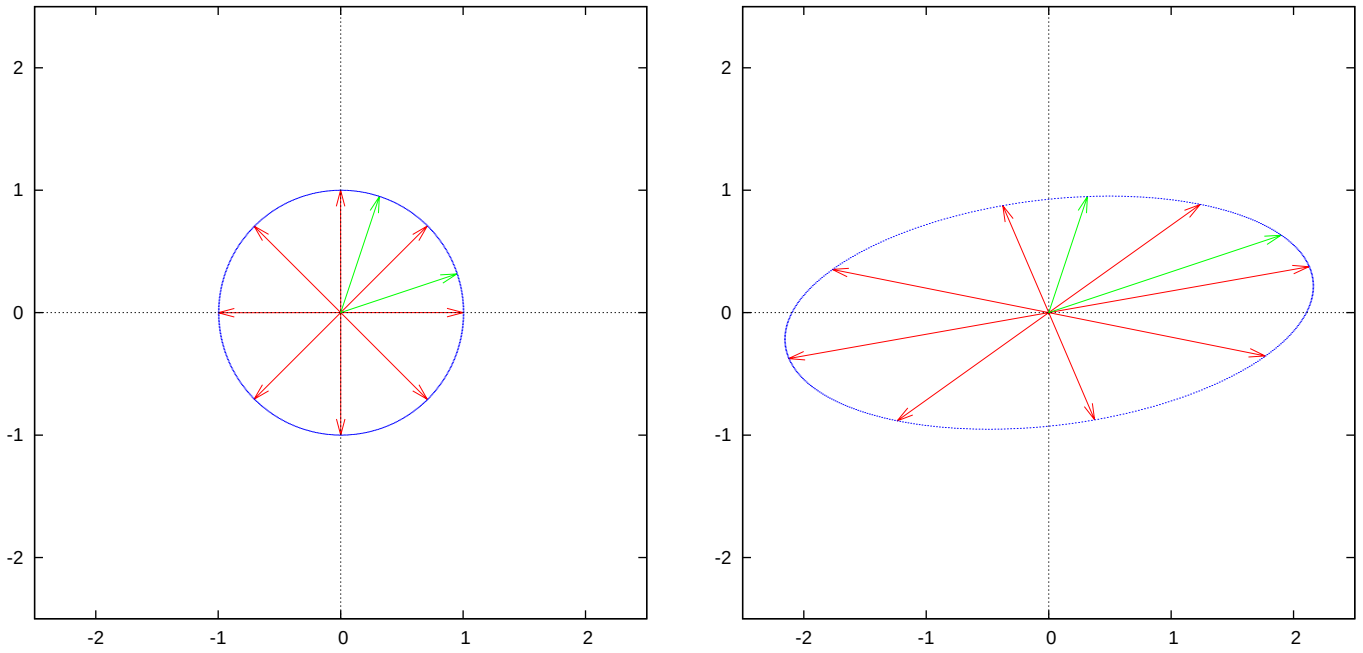
Problem poszukiwania wartości własnych macierzy nazywa się *zagadnieniem własnym*.

Przykład

Rozważmy macierz

$$\begin{bmatrix} \frac{17}{8} & -\frac{3}{8} \\ \frac{3}{8} & \frac{7}{8} \end{bmatrix}. \tag{2}$$

Jej unormowanymi wektorami własnymi są $\left[\frac{1}{\sqrt{10}}, \frac{3}{\sqrt{10}}\right]^T$, $\left[\frac{3}{\sqrt{10}}, \frac{1}{\sqrt{10}}\right]^T$, a odpowiadającymi im wartościami własnymi liczby $1, 2$. Rysunek na następnej stronie przedstawia rodzinę wektorów jednostkowych (lewy panel) oraz tę samą rodzinę przekształconą przez macierz (2). Wektory własne zaznaczone są na zielono.



Rodzina wektorów jednostkowych przekształconych przez macierz (2). Wektory własne tej macierzy zaznaczone są na zielono. Jeden z nich, odpowiadający wartości własnej 1, nie zmienia się, drugi, odpowiadający wartości własnej 2, zachowuje kierunek, ale jego długość rośnie dwukrotnie.

PageRank Algorithm

Przypuśćmy, że chcemy opracować ranking stron WWW, umożliwiający ich pozycjonowanie w wyszukiwarce. Zauważmy, że w danej chwili istnieje skończona liczba stron WWW, możemy je zatem — przynajmniej teoretycznie — ponumerować.

Co można uznać za kryterium "ważności" strony, wpływające na jej pozycję w rankingu? Może się wydawać, że strona jest tym ważniejsza, im więcej linków z innych stron do niej prowadzi. Jednak po chwili zastanowienia widać, że to kryterium może być mylące: Jeżeli na naszą stronę prowadzi dużo linków z innych "nieważnych" stron, nasza strona i tak może być trudna do znalezienia, gdyż użytkownicy być może nigdy nie znajdą *tamtych* stron, a więc nie trafią na naszą. Jeśli natomiast na naszą stronę

prowadzi tylko jeden link, ale ze strony, która sama jest bardzo popularna, nasza strona może zyskać wielu użytkowników.

Widać zatem, że przy obliczaniu rankingu naszej strony, należy uwzględnić rankingi stron, które linkują do naszej.

Oznaczmy r_i — ranking i-tej strony. Niech $k \to i$ oznacza, że strona k-ta linkuje do strony i-tej. Proponujemy zatem

$$r_i = C \cdot \sum_{k\,:\; k \to i} r_k \,, \tag{3}$$

gdzie $C > 0$ jest stałą proporcjonalności, a sumowanie rozciąga się po wszystkich stronach k, które linkują do strony i.

Zauważmy jednak, że jeśli k-ta strona zawiera wiele linków, użytkownik może kliknąć dowolny z nich, niekoniecznie ten prowadzący do strony i-tej. Modyfikujemy zatem powyższą definicję następująco:

$$r_i = C \cdot \sum_{k\,:\; k \to i} r_k / l_k \,, \tag{4}$$

gdzie l_k oznacza liczbę linków na stronie k-tej.

Spróbujmy nieco uprościć notację. W tym celu wprowadzam oznaczenie

$$p_{ik} = \begin{cases} l_k^{-1} & \text{jeżeli } k \to i \\ 0 & \text{poza tym} \end{cases} \tag{5}$$

W tych oznaczeniach definicję (4) można zapisać jako

$$r_i = C \cdot \sum_k p_{ik} r_k \,. \tag{6}$$

Zauważmy, że sumujemy już po *wszystkich* k, gdyż zerowe wartości p_{ij} wyeliminują składowe pochodzące od stron nie linkujących do strony i.

Niech $\mathbf{P}$ oznacza macierz, której elementami są liczby p_{ij}, i niech $\mathbf{r}$ oznacza wektor rankingów. Widzimy, że równanie (6) możemy zapisać jako

$$\mathbf{Pr} = \lambda \mathbf{r}\,, \tag{7}$$

gdzie $\lambda = 1/C$. *Wektor rankingów jest wektorem własnym macierzy* $\mathbf{P}$*.* Opisana wyżej koncepcja jest podstawą algorytmu PageRank, wymyślonego przez Sergeya Brina i Larry'ego Page'a, twórców Google'a.

Równanie charakterystyczne

Równanie (1) można zapisać w postaci

$$(\mathbf{A} - \lambda \mathbb{I})\,\mathbf{x} = 0\,. \tag{8}$$

Jeżeli λ jest wartością własną, równanie to musi mieć niezerowe rozwiązanie ze względu na $\mathbf{x}$. Jest to możliwe tylko w wypadku

$$\det\left(\mathbf{A} - \lambda \mathbb{I}\right) = 0\,. \tag{9}$$

Równanie (9) nazywane jest *równaniem charakterystycznym macierzy* $\mathbf{A}$. Jest to równanie wielomianowe stopnia N. Widzimy, że każda wartość własna jest pierwiastkiem równania charakterystycznego, ale stwierdzenie odwrotne niekoniecznie jest prawdą.

Zbiór wszystkich rozwiązań równania (9) nazywam *widmem macierzy* $\mathbf{A}$.

Macierz $\mathbf{A} \in \mathbb{C}^{N \times N}$, która ma N niezależnych liniowo wektorów własnych, nazywam *macierzą diagonalizowalną* lub *normalną*. Łatwo sprawdzić, że jeżeli $\mathbf{X} \in \mathbb{C}^{N \times N}$ jest macierzą, której kolumnami są kolejne, liniowo niezależne wektory własne diagonalizowalnej macierzy $\mathbf{A}$, zachodzi

$$\mathbf{X}^{-1}\mathbf{A}\mathbf{X} = \mathrm{diag}\{\lambda_1, \lambda_2, \ldots, \lambda_N\}\,. \tag{10}$$

Uwaga! Poza przypadkami, które daje się rozwiązać analitycznie i poza pewnymi przypadkami specjalnymi, które wykraczają poza zakres tego wykładu, próba poszukiwania wartości własnych macierzy poprzez rozwiązywanie jej równania charakterystycznego na ogół jest numerycznie nieefektywna. Trzeba znaleźć jakieś inne metody.

Metoda potęgowa

Niech $\mathbf{A} \in \mathbb{R}^{N \times N}$ będzie macierzą symetryczną, $\mathbf{A} = \mathbf{A}^T$. Wiadomo, że macierz taka jest diagonalizowalna, ma rzeczywiste wartości własne, a jej unormowane wektory własne tworzą bazę ortonormalną w $\mathbb{R}^N$. Niech bazą tą będzie $\{\mathbf{e}_i\}_{i=1}^N$, przy czym $\mathbf{A}\mathbf{e}_i = \lambda_i \mathbf{e}_i$. Przyjmijmy dodatkowo, że wartości własne macierzy $\mathbf{A}$ są dodatnie i uporządkowane $\lambda_1 > \lambda_2 > \cdots > \lambda_N > 0$.

Weźmy wektor $\mathbf{y} \in \mathbb{R}^N$. Posiada on rozkład

$$\mathbf{y} = \sum_{i=1}^{N} \beta_i \mathbf{e}_i \,. \tag{11}$$

Obliczamy

$$\mathbf{A}\mathbf{y} = \mathbf{A}\sum_{i=1}^{N}\beta_i\mathbf{e}_i = \sum_{i=1}^{N}\beta_i\mathbf{A}\mathbf{e}_i = \sum_{i=1}^{N}\beta_i\lambda_i\mathbf{e}_i \tag{12a}$$

$$\mathbf{A}^2\mathbf{y} = \mathbf{A}\sum_{i=1}^{N}\beta_i\lambda_i\mathbf{e}_i = \sum_{i=1}^{N}\beta_i\lambda_i^2\mathbf{e}_i \tag{12b}$$

$$\mathbf{A}^k\mathbf{y} = \sum_{i=1}^{N}\beta_i\lambda_i^k\mathbf{e}_i \tag{12c}$$

Widzimy, że dla dostatecznie dużych k wyraz zawierający λ_1^k będzie dominował w sumie po prawej stronie (12c): pozostałe współczynniki będą zaniedbywalnie małe, a zatem prawa strona (12c) dla dostatecznie dużych k będzie dążyć do wektora proporcjonalnego do $\mathbf{e}_1$, czyli do wektora własnego do największej wartości własnej.

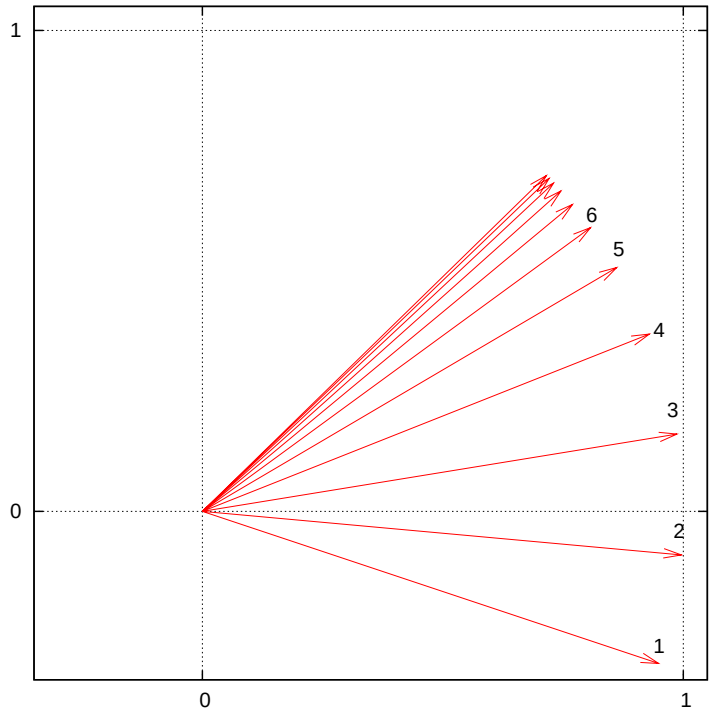
A zatem iteracja ($\|\mathbf{y}_1\| = 1$, poza tym dowolny)

$$\begin{aligned} \mathbf{A}\mathbf{y}_k &= \mathbf{z}_k \\ \mathbf{y}_{k+1} &= \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|} \end{aligned} \tag{13}$$

zbiega się, przy przyjętych założeniach, do unormowanego wektora własnego macierzy $\mathbf{A}$, odpowiadającego największej wartości własnej.

Gdy iteracja (13) zbiegnie się do punktu stałego (kolejne wektory $\mathbf{y}_k \simeq \mathbf{e}_1$ przestaną się zauważalnie zmieniać), wartość własną obliczamy jako $\lambda_1 = \|\mathbf{z}_k\|$.

Przykład



Kolejne wektory generowane przez metodę potęgową, czyli iterację (13), dla macierzy $\begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix}$.

Druga wartość własna

Kiedy w sumie (12c) wyraz z wektorem własnym do największej wartości własnej **nie będzie** dominować? Wtedy i tylko wtedy, gdy współczynnik odpowiadający temu wektorowi w rozwinięciu (11) będzie znikać, $\beta_1 = 0$. To sugeruje sposób na szukanie wektora własnego odpowiadającego *drugiej* co do wielkości wartości własnej: należy upewnić się, że wektor $\mathbf{y}_1$ i wszystkie jego iteraty są prostopadłe do uprzednio znalezionego $\mathbf{e}_1$.

Iteracja ($\|\mathbf{y}_1\| = 1$, $\mathbf{e}_1^T \mathbf{y}_1 = 0$, poza tym dowolny)

$$\begin{aligned} \mathbf{A}\mathbf{y}_k &= \mathbf{z}_k \\ \mathbf{z}_k &= \mathbf{z}_k - \mathbf{e}_1 \left(\mathbf{e}_1^T \mathbf{z}_k \right) \\ \mathbf{y}_{k+1} &= \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|} \end{aligned} \tag{14}$$

zbiega się, przy przyjętych założeniach, do unormowanego wektora własnego macierzy $\mathbf{A}$, odpowiadającego drugiej co do wielkości wartości własnej. Drugi krok powyższego algorytmu oznacza ortogonalizację. Teoretycznie, skoro wektor startowy iteracji jest ortogonalny do $\mathbf{e}_1$, krok tn możnaby pominąć. Tak byłoby w arytmetyce dokładnej. W arytmetyce przybliżonej zakumulowany błąd obcięcia może spowodować, że wygenerowane wektory będą miały niezerowy rzut na kierunek $\mathbf{e}_1$, który w ten sposób zacznie dominować. Dlatego reortogonalizacja, jeśli nie w każdej iteracji, to przynajmniej co kilka, jest nieodzowna, podrażając koszt numeryczny całego algorytmu.

Odwrotna metoda potęgowa

Przypuśćmy, że zamiast największej, szukamy najmniejszej (lecz większej od zera) wartości własnej. Jak łatwo sprawdzić, jeżeli $\mathbf{A}\mathbf{e}_i = \lambda_i \mathbf{e}_i$, to $\mathbf{A}^{-1}\mathbf{e}_i = \lambda_i^{-1}\mathbf{e}_i$, a zatem najmniejsza wartość własna jest największą wartością własną macierzy odwrotnej. Prowadzimy więc iterację

$$\begin{aligned} \mathbf{A}^{-1}\mathbf{y}_k &= \mathbf{z}_k \\ \mathbf{y}_{k+1} &= \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|} \end{aligned} \tag{15}$$

Należy pamiętać, że zapis $\mathbf{A}^{-1}\mathbf{y}_k = \mathbf{z}_k$ należy rozumieć jako konieczność rozwiązania równania $\mathbf{A}\mathbf{z}_k = \mathbf{y}_k$. Widać, że kolejne równania rozwiązujemy korzystając z dokonanego *tylko raz* rozkładu *LU* (lub rozkładu Cholesky'ego, jeśli macierz jest dodatnio określona — tak, jak przy obowiązujących dotąd założeniach).

Uwagi o metodzie potęgowej

1. Jeżeli nie ograniczamy się do macierzy dodatnio określonych, dominującą wartością własną będzie wartość własna o największym module. Podobnie, dominującą wartością własną macierzy odwrotnej będzie wartość własna macierzy nieodwróconej o najmniejszym module; trzeba jednak pamiętać, że macierz odwrotna jest określona tylko wtedy, gdy żadna wartość własna nie jest zerowa.

2. Ujemne wartości własne rozpoznajemy po tym, że po ustabilizowaniu się kierunku wektora własnego, jego współrzędne zmieniają znak w kolejnych iteracjach.

3. Jeżeli przy pomocy metody potęgowej trzeba znaleźć więcej niż kilka wartości i wektorów własnych, metoda, z uwagi na konieczność reortogonalizacji, staje się bardzo kosztowna.

4. Jeżeli w widmie macierzy pojawiają się wartości własne bardzo zbliżone *co do modułu* — na przykład 2 oraz 1.99999999, ale też 1 oraz -0.99999999 — zbieżność będzie bardzo wolna, a nawet można spodziewać się stagnacji.

5. Metoda potęgowa *nie działa* dla macierzy niesymetrycznych!

Transformacja podobieństwa

Zależność (10) jest szczególnym przypadkiem *transformacji podobieństwa*. Ogólnie, dwie macierze $\mathbf{A}$, $\mathbf{B}$ nazywam macierzami podobnymi, jeżeli istnieje taka nieosobliwa macierz $\mathbf{S}$, że zachodzi

$$\mathbf{B} = \mathbf{S}^{-1}\mathbf{A}\mathbf{S}\,. \tag{16}$$

Macierze podobne mają takie same widma, co wynika z faktu, że jeśli spełnione jest równanie (1), spełnione jest także równanie

$$\mathbf{S}^{-1}\left(\mathbf{A} - \lambda\mathbb{I}\right)\mathbf{S}\mathbf{S}^{-1}\mathbf{x} = 0\,, \tag{17}$$

a zatem wektor $\mathbf{S}^{-1}\mathbf{x}$ jest wektorem własnym macierzy (16) do wartości własnej λ.

Ortogonalna transformacja podobieństwa

Na ogół zamiast ogólnej transformacji podobieństwa, używa się *ortogonalnej transformacji podobieństwa*

$$\mathbf{B} = \mathbf{O}^T \mathbf{A} \mathbf{O}\,, \tag{18}$$

gdzie $\mathbf{O}^T\mathbf{O} = \mathbb{I}$. Co więcej, typowo wykonuje się cały ciąg transformacji $\mathbf{A}_2 = \mathbf{O}_1^T \mathbf{A}_1 \mathbf{O}_1$, $\mathbf{A}_3 = \mathbf{O}_2^T \mathbf{A}_2 \mathbf{O}_2$ itd, czyli

$$\mathbf{A}_{k+1} = \underbrace{\mathbf{O}_k^T \ldots \mathbf{O}_2^T \mathbf{O}_1^T}_{\mathbf{P}_k^T} \mathbf{A}_1 \underbrace{\mathbf{O}_1 \mathbf{O}_2 \ldots \mathbf{O}_k}_{\mathbf{P}_k} \tag{19}$$

$\mathbf{P}_k$ jest złożeniem macierzy ortogonalnych, a więc samo jest macierzą ortogonalną.

Ponieważ kolejne transformacje wykonuje się w kolejności “od wewnątrz”, nie musimy pamiętać każdej z nich z osobna. Innymi słowy, jeżeli

$$\begin{aligned}
\mathbf{P}_0 &= \mathbb{I} \\
\mathbf{A}_1 &= \mathbf{Q}_1^T \mathbf{A} \mathbf{Q}_1 \\
\mathbf{P}_1 &= \mathbf{P}_0 \mathbf{Q}_1 \\
\mathbf{A}_2 &= \mathbf{Q}_2^T \mathbf{A}_1 \mathbf{Q}_2 \\
\mathbf{P}_2 &= \mathbf{P}_1 \mathbf{Q}_2
\end{aligned}$$

. . . . . .

$$\begin{aligned}
\mathbf{A}_{\text{diag}} = \mathbf{A}_s &= \mathbf{Q}_s^T \mathbf{A}_{s-1} \mathbf{Q}_s \\
\mathbf{P}_s &= \mathbf{P}_{s-1} \mathbf{Q}_s
\end{aligned} \tag{20}$$

zakumulowana macierz $\mathbf{P}_s$ **jest** macierzą, której kolumnami są wektory własne macierzy $\mathbf{A}$, porównaj (10). Macierz $\mathbf{A}_{\text{diag}}$ może być “numerycznie diagonalna”, to znaczy jej pozadiagonalne elementy mogą być zaniedbywalnie małe.

Uwaga: Jeżeli nie musimy znaleźć wszystkich wektorów własnych i interesują nas tylko wartości własne, nie musimy zapamiętywać zakumulowanych macierzy transformacji, co istotnie przyczynia się do zmniejszenia numerycznego kosztu całej procedury.

Algorytm *QR*

Niech macierz $\mathbf{A}$ posiada faktoryzację *QR*, $\mathbf{A} = \mathbf{QR}$. Obliczmy czemu równa się iloczyn tych czynników wziętych w odwrotnej kolejności.

$$\mathbf{A}' = \mathbf{RQ} = \mathbf{Q}^T \mathbf{AQ} \,. \tag{21}$$

Widzimy, że wymnożenie czynników faktoryzacji *QR* w odwróconej kolejności stanowi ortogonalną transformację podobieństwa macierzy $\mathbf{A}$.

Procedurę tę można iterować. Zaczynamy od $\mathbf{A}_1 = \mathbf{A}$.

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{Q}_1\mathbf{R}_1 \\ \mathbf{A}_2 = \mathbf{R}_1\mathbf{Q}_1 &= \mathbf{Q}_2\mathbf{R}_2 \\ \mathbf{A}_3 = \mathbf{R}_2\mathbf{Q}_2 &= \mathbf{Q}_3\mathbf{R}_3 \end{aligned}$$

. . . . . .

$$\mathbf{A}_n = \mathbf{R}_{n-1}\mathbf{Q}_{n-1} = \mathbf{Q}_n\mathbf{R}_n \tag{22}$$

Każda prawa strona $\mathbf{Q}_k\mathbf{R}_k$ reprezentuje faktoryzację *QR* dokonywaną w k-tym kroku iteracji. Macierz $\mathbf{A}_n$ jest ortogonalnie podobna do macierzy $\mathbf{A}$. Ponadto powyższy algorytm zachowuje symetrię, postać trójdiagonalną symetryczną i postać Hessenberga (patrz niżej).

Zachodzi (nieoczywiste)

Twierdzenie: Jeżeli macierz $\mathbf{A}$ jest diagonalizowalna i jej wszystkie wartości własne są rzeczywiste i parami różne od siebie, iteracja (22) jest zbieżna do macierzy trójkątnej górnej, w której na głównej przekątnej stoją kolejne wartości własne.
Jeżeli niektóre wartości własne są zdegenerowane lub zespolone, iteracja (22) jest zbieżna do macierzy trójkątnej górnej z dodatkowymi klatkami na przekątnej, których wartości własne (do znalezienia "ręcznie") odpowiadają owym wyróżnionym wartościom własnym.

Problem polega na tym, że faktoryzacja *QR* ma złożoność obliczeniową $O(N^3)$ na każdą iterację, co czyni ten algorytm zdecydowanie zbyt drogim. Na szczęście złożoność obliczeniowa faktoryzacji *QR* jest znacznie mniejsza dla macierzy rzadkich.

Algorytm *QR* dla macierzy trójdiagonalnych, symetrycznych

Jak pamiętamy*, faktoryzację *QR* macierzy trójdiagonalnej, symetrycznej można znaleźć za pomocą obrotów Givensa w czasie liniowym:

$$\mathbf{R} = \mathbf{G}_{N-1}\cdots\mathbf{G}_2\mathbf{G}_1\mathbf{A}\,, \tag{23a}$$

$$\mathbf{A} = \underbrace{\mathbf{G}_1^T\mathbf{G}_2^T\cdots\mathbf{G}_{N-1}^T}_{\mathbf{Q}}\mathbf{R}\,, \tag{23b}$$

a wobec tego

$$\mathbf{A}' = \mathbf{RQ} = \underbrace{\mathbf{G}_{N-1}\ldots\mathbf{G}_2\underbrace{\mathbf{G}_1\mathbf{A}\mathbf{G}_1^T}\underbrace{\mathbf{G}_2^T}\ldots\mathbf{G}_{N-1}^T}_{} \tag{23c}$$

Ponieważ macierz $\mathbf{R}$ ma tylko dwa pasma ponad diagonalą, a macierze $\mathbf{G}_k$ tylko jedno pasmo pod, macierz $\mathbf{A}'$ nie tylko jest symetryczna, ale sama trójdiagonala. Ponieważ nie musimy zapamiętywać poszczególnych $\mathbf{G}_k$, tylko uzywamy ich w miarę ich obliczania, jeden krok algorytmu *QR* dla macierzy trójdiagonalnej, symetrycznej, wykonujemy w czasie $O(N)$, w wyniku otrzymując także macierz trójdiagonalną, symetryczną.

*☺

Redukcja do postaci trójdiagonalnej symetrycznej

Niech $\mathbf{A}_1 \in \mathbb{R}^{N\times N}$, $\mathbf{A}_1 = \mathbf{A}_1^T$.

Niech teraz

$$\mathbf{P}_1 = \left[\begin{array}{c|ccc} 1 & 0 & 0 & \dots & 0 \\ \hline 0 & & & & \\ 0 & & & & \\ 0 & & {}^{(N-1)}\mathbf{P}_1 & & \\ 0 & & & & \end{array}\right] \tag{24}$$

gdzie ${}^{(N-1)}\mathbf{P}_1 \in \mathbb{R}^{(N-1)\times(N-1)}$ jest transformacją Householdera, zbudowaną na pierwszej kolumnie macierzy $\mathbf{A}_1$, poczynając od drugiego (poddiagonalnego) elementu. Cała macierz $\mathbf{P}_1$ jest ortogonalna.

Obliczam

$$\mathbf{P}_1\mathbf{A}_1\mathbf{P}_1^T = \mathbf{A}_2 = \begin{bmatrix} \bullet & \bullet & & & & \\ \bullet & \bullet & \bullet & \bullet & \bullet & \cdots \\ & \bullet & \bullet & \bullet & \bullet & \cdots \\ & \bullet & \bullet & \bullet & \bullet & \cdots \\ & \bullet & \bullet & \bullet & \bullet & \cdots \\ & \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{25}$$

Transformacja podobieństwa (25) zeruje pierwszą kolumnę macierzy poczynając od trzeciego elementu, *a także pierwszy wiersz*, za co odpowiedzialna jest transformacja $\mathbf{P}_1^T$ działająca od prawej. Wynika to z faktu, że macierz $\mathbf{A}_1$ jest symetryczna.

W następnym kroku tworzymy

$$\mathbf{P}_2 = \left[\begin{array}{cc|ccc} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ \hline 0 & 0 & \multicolumn{3}{c}{\multirow{4}{*}{$^{(N-2)}\mathbf{P}_2$}} \\ 0 & 0 & & & & \\ 0 & 0 & & & & \\ 0 & 0 & & & & \end{array}\right] \tag{26}$$

gdzie $^{(N-2)}\mathbf{P}_2 \in \mathbb{R}^{(N-2)\times(N-2)}$ jest transformacją Householdera, zbudowaną na drugiej kolumnie macierzy $\mathbf{A}_2$, poczynającpoczynając od trzeciego (poddiagonalnego) elementu. Cała macierz $\mathbf{P}_2$ jest ortogonalna. Stojąca w lewym górnym rogu macierz jednostkowa służy do tego, żeby nie zepsuć struktury, którą osiągnęlimy w poprzednim kroku.

Obliczam

$$\mathbf{P}_2\mathbf{A}_2\mathbf{P}_2^T = \mathbf{A}_3 = \begin{bmatrix} \bullet & \bullet & & & & & \\ \bullet & \bullet & \bullet & & & & \\ & \bullet & \bullet & \bullet & \bullet & \cdots \\ & & \bullet & \bullet & \bullet & \cdots \\ & & \bullet & \bullet & \bullet & \cdots \\ & & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{27}$$

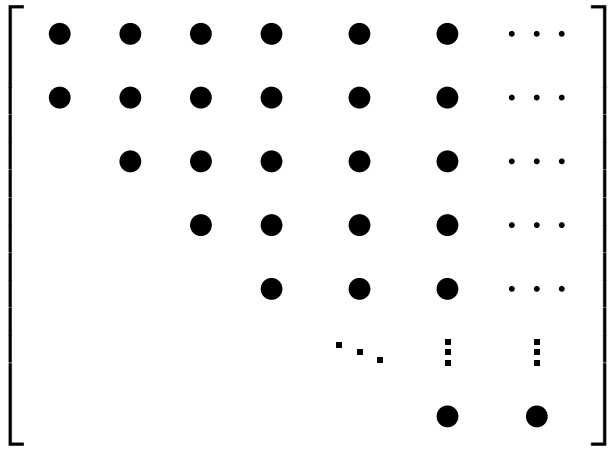
Transformacja podobieństwa (25) zeruje drugą kolumnę macierzy poczynając od czwartego elementu, *a także drugi wiersz*.

I tak dalej. Widać, że po $N-1$ krokach cała macierz zostanie sprowadzona do postaci trójdiagonalnej.

I najważniejsze: Złożoność obliczeniowa faktoryzacji *QR* macierzy trójdiagonalnej wynosi $O(N)$.

Redukcja do postaci Hessenberga

Jeżeli macierz **nie** jest symetryczna, $\mathbf{A}_1^T \neq \mathbf{A}$, algorytm opisany na stronie 27 i następnych nie prowadzi do postaci trójdiagonalnej (gdyż k-ty wiersz jest różny od k-tej kolumny), ale do górnej *postaci Hessenberga*:



(28)

Jest to macierz trójkątna górna z jedną dodatkową diagonalą pod diagonalą główną. Złożoność obliczeniowa faktoryzacji *QR* macierzy w postaci Hessenberga wynosi $O(N^2)$.

Algorytm diagonalizacji

Wobec tego, co powiedziano wyżej, **diagonalizacja małych i średnich macierzy** wygląda następująco:

- Za pomocą transformacji Householdera, opisanej na stronie 27 i następnych, doprowadzamy macierz do prostszej postaci.
  - Dla macierzy symetrycznych, postacią "prostą" jest postać trójdiagonalna symetryczna.
  - Dla macierzy niesymetrycznych, postacią "prostą" jest postać Hessenberga.
- Macierz w "prostej" postaci diagonalizujemy za pomocą algorytmu *QR*, gdyż dla tych postaci faktoryzacja *QR*, przeprowadzana w każdym kroku algorytmu, jest numerycznie tania.

Uwagi

Jeżeli musimy znać wektory własne, akumulujemy wszystkie wykonywane transformacje podobieństwa. Jeżeli chcemy znać tylko wartości własne, nie musimy akumulowac transformacji, co może być istotną oszczędnością czasu.

Dla dobrze rozdzielonych wartości własnych, algorytm *QR* jest dość szybko zbieżny. Jeżeli wartości własne leżą blisko siebie, zbieżność może być powolna. W takiej sytuacji algorytm *QR* można przyspieszyć za pomocą techniki zwanej *implicit shifts*.

W wypadku macierzy dużych, o $N \gg 1000$, konieczność znajomości *wszystkich* wartości i wektorów własnych występuje bardzo rzadko. Na ogół potrzebne jest kilka-kilkanaście dominujących wartości. Do tego celu służą specjalne algorytmy (Lanczosa, Davidsona), które wykraczają poza program tego wykładu.

Wartości własne macierzy hermitowskiej

Niech $\mathbf{H} \in \mathbb{C}^{N\times N}$ będzie macierzą hermitowską, $\mathbf{H}^\dagger = \mathbf{H}$. Jak wiadomo, ma ona rzeczywiste wartości własne. Niech

$$\mathbf{H}\mathbf{u} = \lambda \mathbf{u}\,. \tag{29}$$

Rozłóżmy $\mathbf{H}$ na część rzeczywistą i urojoną, $\mathbf{H} = \mathbf{A} + i\mathbf{B}$, $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$. Łatwo pokazać, że $\mathbf{A}^T = \mathbf{A}$, $\mathbf{B}^T = -\mathbf{B}$. Niech $\mathbf{u} = \mathbf{x} + i\mathbf{y}$, $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$. Z (29) otrzymujemy

$$(\mathbf{A} + i\mathbf{B})(\mathbf{x} + i\mathbf{y}) = \lambda(\mathbf{x} + i\mathbf{y}) \tag{30a}$$

$$\mathbf{A}\mathbf{x} - \mathbf{B}\mathbf{y} + i(\mathbf{B}\mathbf{x} + \mathbf{A}\mathbf{y}) = \lambda\mathbf{x} + i\lambda\mathbf{y} \tag{30b}$$

Równość w (30b) musi zachodzić jednocześnie dla części rzeczywistych i urojonych. Zapisując te równości w notacji macierzowej, otrzymujemy

$$\underbrace{\begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix}}_{\widetilde{\mathbf{H}}} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} \tag{31}$$

Macierz $\widetilde{\mathbf{H}}$ jest symetryczna i rzeczywista, a zatem wydaje się, że problem diagonalizacji macierzy hermitowskiej został sprowadzony do problemu diagonalizacji macierzy symetrycznej, rzeczywistej.

Jest jednak pewien *problem*: macierz $\mathbf{H} \in \mathbb{C}^{N\times N}$ ma N wartości własnych. Macierz $\widetilde{\mathbf{H}} \in \mathbb{R}^{2N\times 2N}$ ma $2N$ wartości własnych.

Okazuje się, że przy przejściu od $\mathbf{H}$ do $\widetilde{\mathbf{H}}$, każda wartość własna się klonuje. Aby to zobaczyć, pomnóżmy obie strony (29) przez $-i$. Postępując jak poprzednio, zamiast (30b) otrzymujemy

$$\begin{aligned} -i(\mathbf{Ax}-\mathbf{By})+\mathbf{Bx}+\mathbf{Ay} &= -i\lambda\mathbf{x}+\lambda\mathbf{y} && \text{(32a)} \\ \mathbf{Ay}-\mathbf{B}(-\mathbf{x})+i(\mathbf{By}+\mathbf{A}(-\mathbf{x})) &= \lambda\mathbf{y}+i\lambda(-\mathbf{x}) && \text{(32b)} \end{aligned}$$

$$\begin{bmatrix} \mathbf{A} & -\mathbf{B} \\ \mathbf{B} & \mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ -\mathbf{x} \end{bmatrix} = \lambda \begin{bmatrix} \mathbf{y} \\ -\mathbf{x} \end{bmatrix} \qquad \text{(33)}$$

Porównując (31) i (33) widzimy, że wektory $\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}$ i $\begin{bmatrix} \mathbf{y} \\ -\mathbf{x} \end{bmatrix}$ są wektorami własnymi macierzy $\widetilde{\mathbf{H}}$ do *tej samej* wartości własnej λ. Ponieważ wektory te są ortogonalne, wartość własna λ (rozumiana jako wartość własna $\widetilde{\mathbf{H}}$, nie $\mathbf{H}$) jest (co najmniej) dwukrotnie zdegenerowana.

Rezolwenta

Niech $\mathbf{A} \in \mathbb{C}^{N\times N}$ i niech pewna liczba $\xi \in \mathbb{C}$ **nie** należy do widma $\mathbf{A}$. Wynika stąd, że $\det(\mathbf{A} - \xi\mathbb{I}) \neq 0$. Macierz $\mathbf{Z} = (\mathbf{A} - \xi\mathbb{I})^{-1}$ nazywam rezolwentą macierzy $\mathbf{A}$. (Rezolwenta jest funkcją argumentu ξ.) Niech λ i $\mathbf{u}$ będą wartością własną i odpowiadającym jej wektorem własnym $\mathbf{A}$:

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}\,. \tag{34a}$$

Z (34a) wynika ($\xi \neq \lambda$)

$$\mathbf{A}\mathbf{u} - \xi\mathbf{u} = \lambda\mathbf{u} - \xi\mathbf{u} \tag{34b}$$

$$(\mathbf{A} - \xi\mathbb{I})\mathbf{u} = (\lambda - \xi)\mathbf{u} \tag{34c}$$

$$\frac{1}{\lambda - \xi}\mathbf{u} = (\mathbf{A} - \xi\mathbb{I})^{-1}\mathbf{u} = \mathbf{Z}\mathbf{u}\,. \tag{34d}$$

Ostatnie równanie oznacza, że $\mathbf{u}$ jest wektorem własnym rezolwenty do wartości własnej $(\lambda - \xi)^{-1}$.

Transformacja Möbiusa

Algorytmy znajdywania wartości własnych *dużych* macierzy są na ogół szybko zbieżne do wartości skrajnych, izolowanych. Często znajomość takich wartości własnych wystarcza nam ze względów praktycznych. Jeżeli jednak chcemy dokładnie obliczyć dwie zbliżone wartości własne, leżące gdzieś "w środku" widma, zbieżność może być bardzo wolna. Można ją przyspiezyć za pomocą transformacji Möbiusa.

Niech $\mathbf{u}_1, \mathbf{u}_2$ będą dwoma wektorami własnymi pewnej macierzy $\mathbf{A}$, zaś λ_1, λ_2 będą odpowiadającymi im wartościami własnymi, przy czym $\exists \tau : \lambda_1 = \tau + \varepsilon_1, \lambda_2 = \tau + \varepsilon_2$, $|\varepsilon_{1,2}| \ll 1$ oraz τ *nie* jest wartością własną $\mathbf{A}$. Rozważmy rezolwentę $(\mathbf{A} - \tau \mathbb{I})^{-1}$. Wektory $\mathbf{u}_{1,2}$ są jej wektorami własnymi do wartości własnych $(\tau + \varepsilon_{1,2} - \tau)^{-1} = \varepsilon_{1,2}^{-1}$. Różnica tych wartości własnych wynosi $\left|\frac{\varepsilon_1 - \varepsilon_2}{\varepsilon_1 \varepsilon_2}\right|$, co jest liczbą dużą, gdyż $\varepsilon_{1,2}$ są małe. Widzimy, że

zbliżone wartości własne macierzy odpowiadają dobrze rozseparowanym wartościom własnym rezolwenty, wektory własne zaś są takie same.

Dobre rozseparowanie wartości własnych rezolwenty powinno sprawić, że procedura poszukiwania tych wartości własnych powinna być szybko zbieżna, co zrekompensuje koszt obliczania samej rezolwenty. Dodatkowym ułatwieniem jest fakt, że przy obliczaniu rezolwenty (ściślej: wyników działania rezolwenty na jakieś wektory), możemy cały czas korzystać z tej samej faktoryzacji.

Poszukiwanie wektora własnego gdy znana jest wartość własna

Jak wspomniano wyżej, poszukiwanie samych wartości własnych jest znacznie tańsze, niż jednoczesne poszukiwanie wartości i wektorów własnych. Bardzo często zdarza się, że chcemy poznać wszystkie (lub większość) wartości własnych, ale tylko kilka wektorów własnych, odpowiadających wybranym (na ogół skrajnym lub w inny sposób charakterystycznym) wartościom własnym.

Przyjmijmy, że λ jest wartością własną macierzy $\mathbf{A}$, której wektory własne stanowią bazę w $\mathbb{R}^N$ i niech $\tau \simeq \lambda$ *nie* należy do widma $\mathbf{A}$. (Jako τ można brać przybliżenie λ otrzymane *numerycznie*, bo ono powinno być bliskie "prawdziwej" wartości własnej, ale różne od niej.)

Rozważamy rezolwentę $(\mathbf{A}-\tau\mathbb{I})^{-1}$. Jej dominującą wartością własną będzie $(\lambda-\tau)^{-1}$. Podobnie jak w metodzie potęgowej, iteracja ($\|\mathbf{y}_1\|=1$, poza tym dowolny)

$$\begin{aligned}(\mathbf{A}-\tau\mathbb{I})^{-1}\mathbf{y}_k &= \mathbf{z}_k \\ \mathbf{y}_{k+1} &= \frac{\mathbf{z}_k}{\|\mathbf{z}_k\|}\end{aligned} \tag{35}$$

zbiega się, przy przyjętych założeniach, do unormowanego wektora własnego macierzy $(\mathbf{A}-\tau\mathbb{I})^{-1}$, odpowiadającego wartości własnej $(\lambda-\tau)^{-1}$ (trzeba uważać na przypadek ujemnych wartości własnych rezolwenty, objawiający się oscylacyjnymi zmianami wektora własnego!). Ten sam wektor jest jednocześnie wektorem własnym $\mathbf{A}$, odpowiadającym wartości własnej $\lambda\simeq\tau$.

W iteracji (35) wykorzystujemy cały czas ten sam rozkład *LU*. Ważne jest, aby obliczenia (i sam rozkład) prowadzić w podwyższonej precyzji, gdyż macierz $\mathbf{A}-\tau\mathbb{I}$ jest źle uwarunkowana.

Uogólnione zagadnienie własne

Zagadnienie własne oznacza analizę macierzy postaci $\mathbf{A} - \lambda\mathbb{I}$. Jeśli macierz jednostkową zastąpimy jakąś inną macierzą, otrzymamy $\mathbf{A} - \lambda\mathbf{B}$, co prowadzi do uogólnionego zagadnienia własnego:

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}\,, \tag{36}$$

gdzie $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{N\times N}$. Liczby λ nazywamy uogólnionymi wartościami własnymi, a wektory $\mathbf{x}$ uogólnionymi wektorami własnymi. Zagadnienia tego typu pojawiają się przy rozwiązywaniu pewnych problemów fizycznych.

Załóżmy dodatkowo, że $\mathbf{B}$ jest symetryczna i dodatnio określona. Wówczas istnieje faktoryzacja Cholesky'ego $\mathbf{B} = \mathbf{C}\mathbf{C}^T$.

Problem (36) mogę zapisać jako

$$\mathbf{C}^{-1}\mathbf{A}\left(\mathbf{C}^T\right)^{-1}\mathbf{C}^T\mathbf{x} = \lambda\mathbf{C}^T\mathbf{x} \tag{37a}$$

$$\tilde{\mathbf{A}}\mathbf{y} = \lambda\mathbf{y}\,, \tag{37b}$$

gdzie $\tilde{\mathbf{A}} = \mathbf{C}^{-1}\mathbf{A}\left(\mathbf{C}^T\right)^{-1}$, $\mathbf{y} = \mathbf{C}^T\mathbf{x}$. Jak widać, uogólniony problem własny (36) jest równoważny "zwykłemu" problemowi własnemu (37b). Ponadto, jeżeli $\mathbf{A}$ jest symetryczna (i dodatnio określona), także $\tilde{\mathbf{A}}$ jest symetryczna (i dodatnio określona). Aby numerycznie rozwiązać problem własny (37b), należy znaleźć $\mathbf{C}^{-1}$ — jest to jedna z niewielu sytuacji, w których odwrotność jakiejś macierzy trzeba znaleźć *explicite*. Skądinąd odwrotność macierzy trójkątnej znajduje się stosunkowo szybko.

Niech $\tilde{\mathbf{A}}$ będzie symetryczna i dodatnio określona i niech $\lambda_1 \neq \lambda_2$. Wówczas odpowiednie wektory własne są prostopadłe, $\mathbf{y}_1^T \mathbf{y}_2 = 0$. Mamy

$$0 = \mathbf{y}_1^T \mathbf{y}_2 = \left(\mathbf{C}^T \mathbf{x}_1\right)^T \mathbf{C}^T \mathbf{x}_2 = \mathbf{x}_1^T \mathbf{C}\mathbf{C}^T \mathbf{x}_2 = \mathbf{x}_1^T \mathbf{B} \mathbf{x}_2 \,. \qquad (38)$$

Jak widzimy, w przypadku macierzy symetrycznych uogólnione wektory własne odpowiadające różnym uogólnionym wartościom własnym są sprzężone względem macierzy $\mathbf{B}$, przy dodatkowym założeniu, że $\mathbf{B}$ jest dodatnio określona. Można również powiedzieć, że uogólnione wektory własne są ortogonalne względem iloczynu skalarnego generowanego przez symetryczną i dodatnio określoną macierz $\mathbf{B}$. Uogólnione wektory własne stanowią także bazę w przestrzeni $\mathbb{R}^N$, nieortogonalną względem "naturalnego" iloczynu skalarnego.