Часткова проблема власних значень. Сингулярне розкладення матриці

Часткова проблема власних значень

- В багатьох задачах потрібно знайти не всі власні значення, а лише деякі: наприклад, максимальне за модулем власне число та відповідний вектор, мінімальне за модулем власне число та відповідний вектор. Розглянемо алгоритми, які дозволяють розв’язати часткову проблему.

Степеневий метод

- Застосовується для знаходження максимального по модулю власного числа матриці А та відповідного власного вектора.

$x_1, x_2, \ldots x_n$ – базис.

max ILI (для матриці А) = 1/min ILI (для матриці A^(-1))

Нехай максимальне за модулем значення дійсне, і немає кратних власних значень

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \ldots \geq |\lambda_n|$$

Довільний y_0:

$$y_1 = Ay_0,\ y_2 = Ay_1, \ldots, y_i = Ay_{i-1}$$

Запишемо розкладення $y_0 \ldots y_i$ за базисом $x_1, \ldots x_n$

При цьому координата y_0<>0, тож $y_1, y_2, \ldots, y_i$ теж. Нехай a_1<>0

$$\begin{cases} y_0 = a_1 x_1 + a_2 x_2 + \ldots a_n x_n = \sum_{l=1}^{n} a_l x_l; \\ y_1 = Ay_0 = a_1\lambda_1 x_1 + a_2\lambda_2 x_2 + \ldots a_n\lambda_n x_n = \sum_{l=1}^{n} a_l\lambda_l x_l; \\ \ldots.. \\ y_i = A^i y_0 = a_1\lambda_1^i x_1 + a_2\lambda_2^i x_2 + \ldots a_n\lambda_n^i x_n = \sum_{l=1}^{n} a_l\lambda_l^i x_l \end{cases}$$

…………………………………
Звідки?

Розкладемо y_k ($k=0,1,\dots,i$) за базисом $u_1, u_2, \dots, u_n$:

$$\underline{y_k = \sum_{j=1}^{n} b_{jk} u_j}, \quad k = 0,1,\dots,i$$

Знайдемо розкладення y_k ($k=0,1,\dots,i$) через базис $u_1, u_2, \dots, u_n$ підставляючи розкладення :

$$\underline{y_k} = \sum_{l=1}^{n} a_l \lambda_l^k x_l = \sum_{l=1}^{n} a_l \lambda_l^k \sum_{j=1}^{n} c_{jl} u_j = \underline{\sum_{j=1}^{n} \sum_{l=1}^{n} a_l \lambda_l^k c_{jl} u_j}$$

Порівняємо розкладення та врахуємо лінійну незалежність $\{u_j\}$

$$b_{jk} = \sum_{l=1}^{n} a_l \lambda_l^k c_{jl} \qquad b_{jk+1} = \sum_{l=1}^{n} a_l \lambda_l^{k+1} c_{jl}$$

Розглянемо відношення компонент для двох сусідніх ітерацій:

$$\frac{b_{jk+1}}{b_{jk}} = \frac{a_1\lambda_1^{k+1}c_{j1} + a_2\lambda_2^{k+1}c_{j2} + \ldots + a_n\lambda_n^{k+1}c_{jn}}{a_1\lambda_1^{k}c_{j1} + a_2\lambda_2^{k}c_{j2} + \ldots + a_n\lambda_n^{k}c_{jn}} =$$

$$= \lambda_1 \frac{1 + \cancel{\frac{a_2c_{j2}}{a_1c_{j1}}\left(\frac{\lambda_2}{\lambda_1}\right)^{k+1}} + \ldots + \frac{a_nc_{jn}}{a_1c_{j1}}\left(\frac{\lambda_n}{\lambda_1}\right)^{\cancel{k+1}}}{1 + \cancel{\frac{a_2c_{j2}}{a_1c_{j1}}\left(\frac{\lambda_2}{\lambda_1}\right)^{k}} + \ldots + \cancel{\frac{a_nc_{jn}}{a_1c_{j1}}\left(\frac{\lambda_n}{\lambda_1}\right)^{k}}}$$

Оскільки $a_1 <> 0$, та існує j $c_{j1} <> 0$

$$\forall l \left|\frac{\lambda_l}{\lambda_1}\right| < 1$$

$$\frac{b_{jk+1}}{b_{jk}} = \lambda_1\left(1 + O\left[\left(\frac{\lambda_2}{\lambda_1}\right)^k + \left(\frac{\lambda_3}{\lambda_1}\right)^k + \ldots + \left(\frac{\lambda_n}{\lambda_1}\right)^k\right]\right) \longrightarrow \lim \frac{b_{jk+1}}{b_{jk}} = \lambda_1$$

Власний вектор при максимальному власному числі

$$y_k = A^k y_0$$

Наближено = власному вектору, який відповідає 1 власному числу

$$A^k y_0 = a_1\lambda_1^k x_1 + \sum_{l=2}^{n} a_l \lambda_l^k x_l = a_1\lambda_1^k \left(x_1 + \cancel{\sum_{l=2}^{n} \frac{a_l}{a_1}\left(\frac{\lambda_l}{\lambda_1}\right)^k} x_l\right) \approx a_1\lambda_1^k x_1$$

при великих k

Зворотний степеневий метод зі зсувом, пошук мінімального за модулем власного числа

- Ітерації дають збіг до максимального за модулем власного числа матриці А, а отже, 1/min_за_ модулем для A^{-1}.

Збіжність методу повільна, тому для прискорення використовують <u>зсув</u>.

Нехай відоме наближене значення $\tilde{\lambda}_k$ деякого власного числа, не обов'язково мінімального за модулем. Тоді зсунута матриця має власні значення $\lambda_k - \tilde{\lambda}_k$. Для матриці $(A - \tilde{\lambda}_k I)$ власне значення $\lambda_k - \tilde{\lambda}_k$ є найменшим за модулем. Зворотні ітерації зі зсунутою матрицею набувають виду:

$$(A - \tilde{\lambda}_k I)x^{(n+1)} = x^{(n)}$$

$$\lambda_k - \tilde{\lambda}_k$$

Після кожної ітерації <u>слід нормувати</u> одержаний вектор, для запобігання появі надто великих значень.

Зворотний степеневий метод зі змінним зсувом

$$(A-\tilde{\lambda}_k^{(n)}I)x^{(n+1)}=x^{(n)}$$

$$\tilde{\lambda}_k^{(n+1)}=\tilde{\lambda}_k^{(n)}+<\frac{x_i^{(n)}}{x_i^{(n+1)}}>$$

<….> - усереднення за усіма компонентами.

З урахуванням нормування:

$$(A-\tilde{\lambda}_k^{(n)}I)y^{(n+1)}=x^{(n)}$$

$$\tilde{\lambda}_k^{(n+1)}=\tilde{\lambda}_k^{(n)}+<\frac{x_i^{(n)}}{y_i^{(n+1)}}>$$

$$x^{(n+1)}=\frac{y^{(n)}}{\|y^{(n)}\|}.$$

Метод скалярних добутків

- є модифікацією степеневого методу та використовується для знаходження максимального за модулем власного значення дійсної матриці. Особливо зручний для симетричних матриць.

$$\max_i |\lambda_i| \approx \frac{(y_k, y'_k)}{(y_{k-1}, y'_k)} = \frac{(A^k y_0, A'^k y_0)}{(A^{k-1} y_0, A'^k y_0)}$$

Сингулярне розкладення матриць

- узагальнення власних чисел на випадок прямокутних матриць- називається сингулярними числами.

Тут https://www.alglib.net/matrixops/general/svd.php наведено огляд існуючих методів сингулярного розкладення. Готова реалізація методу SVD (singular value decomposition) є не лише на мові Python, а й на мові C++.

Сингулярним розкладенням матриць називається її факторизація виду

A=USV`, де

U- ортогональна матриця mxm ,

V- ортогональна матриця nxn,

S- діагональна матриця mxn, де по діагоналі розташовані сингулярні числа $\sigma_{ii}=\sigma_i\geq 0$.

Або ж,

$$U'AV=S.$$

Задача: підібрати U,V щоби перетворити вихідну матрицю на діагональну

Етапи алгоритму сингулярного розкладення матриці

- 1)Приведення до двухдіагональної форми (ненульовими є елементи діагоналі та першої над-діагоналі). Введення нулів може відбуватись із використанням так званих хаусхолдерових перетворень. Вводити нулі до матриці А за допомогою, наприклад, гаусівського виключення не можна, оскільки відповідне перетворення не буде ортогональним.
- 2)Ітераційний процес, де наддіагональні елементи зменшуються до нульових значень із заданою точністю. Використовується різновид QR-алгоритма, використовуються матриці обертання Т.

Приведення до двудіагональної форми

- За першим стовпцем-вектором a_1 матриці А будуємо вектор u_1, який відрізняється від a_1 лише першою координатою: вона збільшена на $||a1||$ (тут мається на увазі евклідова норма).

$$U_1 = I - \beta_1^{-1} u_1 u_1', \ (1)$$

$$\beta_1 = \frac{||u_1||^2}{2}. \ (2)$$

$$A_1 = A,$$

далі знаходимо $A_2 = U_1 A_1$

(використовуючи (1) та (2)):

$$U_1 a_1 = (I - \beta_1^{-1} u_1 u_1') a_1 = a_1 - \beta_1^{-1} u_1 u_1' a_1 = a_1 - u_1,$$

оскільки

$$\beta_1^{-1} u_1' a_1 = \frac{2}{||u_1||^2} \left(a_1 + ||a_1|| l_1, a_1\right) = \frac{2(a_1 + ||a_1|| l_1, a_1)}{(a_1 + ||a_1|| l_1, a_1 + ||a_1|| l_1)} = \frac{2||a_1||^2 + 2a_{11} ||a_1||}{2||a_1||^2 + 2a_{11} ||a_1||} = 1,$$

де a_{11} – перша координата вектора a_1.

Перший стовпець після перетворення u_1

$$u_1 a_1 = a_1 - u_1 = (-||a_1||, 0, 0 \ldots 0)'.$$
$$||u_1 a_1|| = ||a_1||.$$

Другий стовпець матриці А дорівнює:

$$u_1 a_2 = a_2 - \beta_1^{-1} u_1 u_1' a_2 = a_2 - c_{12} u_1,\ c_{12} = \beta_1^{-1}(u_1', a_2).$$

Третій стовпець матриці А дорівнює:

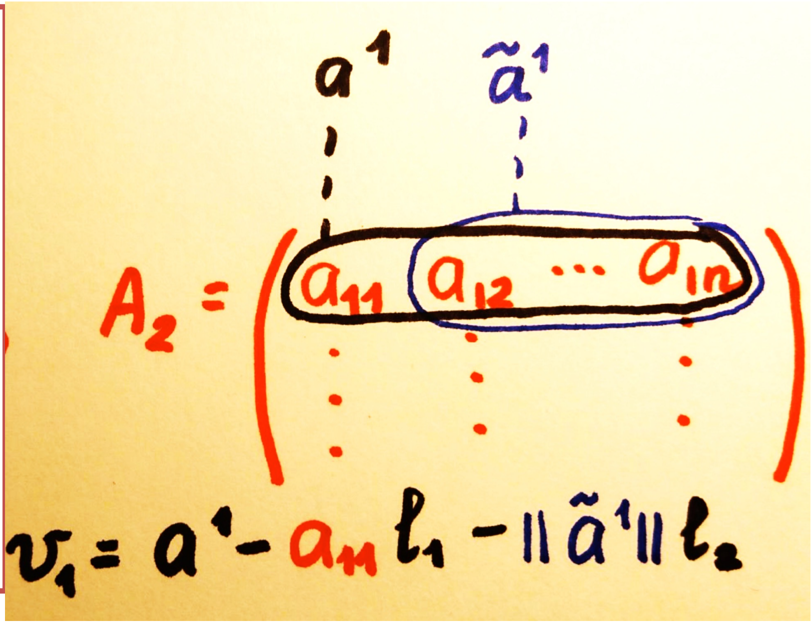
$$u_1 a_3 = a_2 - \beta_1^{-1} u_1 u_1' a_3 = a_3 - c_{13} u_1,\ c_{13} = \beta_1^{-1}(u_1', a_3).$$

І так далі.

Стовпці матриці A_2 отримуються відніманням векторів, колінеарних u_1, зі стовпців A_1.

Перетворення U_1 – хаусхолдерове віддзеркалення, і є ортогональним ($U_1'U_1$=I). З ортогональності перетворення U_1 витікає, що довжина будь-якого стовпця в A_2 дорівнює довжині відповідного стовпця в A_1.

- Перед тим, як вводити нулі в другому стовпці під діагоналлю, за допомогою V_1 одержимо нулі в першому рядку. При цьому нулі, отримані в 1 стовпці, повинні зберегтись, довжина першого рядка не повинна змінитись. Перетворення V_1 породжується матрицею A_2.
- Для цього будуємо вектор v_1, всі координати якого окрім першої та другої, дорівнюють A_2. Перша координата дорівнює нулю, друга дорівнює елементу першого рядка мінус норма вектора розмірності (n-1), координати якого – елементи першого рядка A_2, починаючи з 2-го.



$$g_1 = \frac{\|v_1\|^2}{2},\ A_3 = A_2 V_1$$

$$a_i^{2\prime} V_1 = a_i^{2\prime}(I - g_1{}^{-1} v_1 v_1') = a_i^{2\prime} - (g_1{}^{-1} a_i{}^{2\prime}, v_1) v_1';$$

з i-го рядка $a_i^{2\prime}$ матриці A_2 віднімаємо рядок, пропорційний v_1'.
Далі вводимо нулі в 2-й стовпець під діагоналлю, координати якого, починаючи з 3-ї, дорівнюють відповідним елементам 2-го стовпця. Всі побудови потрібно виконувати з матрицею розмірності (m-1)(n-1), яка одержується з A_3 відкиданням 1 стовпця та рядка. І так далі. Поки, в результаті не одержимо двухдіагональну матрицю. Для цього потрібно n перетворень U та n-2 перетворень V.

Приклад. 1 етап сингулярного розкладення

A=

1	6	11
2	7	12
3	8	13
4	9	14
5	10	15

1) Приведення до двухдіагонального виду

u_1=

8,416
2,000
3,000
4,000
5,000

$$\beta_1 = \frac{||u_1||^2}{2} = 62{,}415$$

$U_1 = I - \beta_1^{-1} u_1 u_1'$

$A_1 = A$,

$A_2 = U_1 A_1$

$U_1 a_1 = a_1 - u_1 =$

-7,416
0
0
0
0

$c_{12} = \beta_1^{-1}(u_1', a_2) = 2{,}796$, $c_{13} = \beta_1^{-1}(u_1', a_3) = 4{,}592$.

$U_1 a_2 = a_2 - 2{,}796 u_1 =$

-17,529

1,409

-0,387

-2,183

-3,949

$U_1 a_3 = a_3 - 4{,}592 u_1 =$

-27,642

2,817

-0,774

4,366

-7,957

Матриця A_2=

-7,416	-17,529	-27,642
0	1,409	2,817
0	-0,387	-0,774
0	-2,183	4,366
0	-3,949	-7,957

Далі, одержимо нулі в 1 рядку:

v_1=

0 -50,261 -27,642

$g_1 = 1645{,}124$

V_1=I-0,000608 v_1 v_1'

$A_3 = A_2 V_1$

Рядок A_3 з номером *i* одержимо так:

$a_i' V_1 = a_i' - (g_1{}^{-1} a_i{}', v_1)v_1';$

$a_1' V_1 = a_1' - 1 \cdot v_1{}'$= (-7,416 32,732 0)

$a_2' V_1 = a_2' + 0{,}090379 v_1{}'$=(0 -3,133 0,319)

$a_3' V_1 = a_3' - 0{,}024828 v_1{}'$=(0 0,861 -0,088)

$a_4' V_1 = a_4' + 0{,}006665 v_1{}'$=(0 4,856 -0.495)

$a_5' V_1 = a_5' + 0{,}254344 v_1{}'$=(0 8.850 -0.901)

$A_3=$

$$
\begin{array}{lll}
-7{,}416 & 32{,}732 & 0 \\
0 & -3{,}133 & 0{,}319 \\
0 & 0{,}861 & -0{,}088 \\
0 & 4{,}856 & -0.495 \\
0 & 8.850 & -0.901
\end{array}
$$

...

$A_4=U_2A_3=$

$$
\begin{array}{lll}
-7{,}416 & 32.732 & 0 \\
0 & 10.625 & -1.080 \\
0 & 0 & 0.000 \\
0 & 0 & 0.000 \\
0 & 0 & 0.000
\end{array}
$$

Застосунки сингулярного розкладення

Для систем виду Ax=b, A:*mxn, m≥n* виникають питання:

- чи існує розв'язок,
- чи єдиний розв'язок,
- чи має система Ax=0 ненульові розв'язки.

Від системи Ax=b переходять до USV'x=b, V'x=z, U'b=d, і досліджують систему Sz=d.

При *m≥n* підсистеми, які виникають

- $s_j z_j = d_j$, якщо j≤n, $s_{j \neq 0}$.
- $0 \cdot zj = dj$, якщо j≤n, $s_j=0$
- $0=dj$ j>n.

Обчислення визначників. det A=detU detS det V'.

- Оскільки det U=*±1*, det V=*±1*,

тоді det A=*±s1s2...sn*.

Дослідження обумовленості матриць: число обумовленості

cond A= max|s_j|/min|s_j| для матриць повного рангу. Кількість ненульових s_j відповідає рангу матриці.

Аналіз закономірностей, використання в алгоритмах машинного навчання для зниження розмірності простору. Зокрема, див. використання SVD в LSA (latent semantic analysis). https://www.youtube.com/watch?v=QUvQ7iv6c04

....
....
....
Коли розвязок існує, і рівняння сумісні?