دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

عنوان : روش های عددی محاسبه مقادیر ویژه ماتریس های سه قطری

نگارش : علیرضا محمدزاده

استاد راهنما: دکتر مهدی دهقان

مقدمه

در این پروژه به بررسی روش QL برای محاسبه ی مقادیر ویژه ی ماتریس های سه قطری اشاره می شود. در ابتدا به روش QR می پردازیم که بسیار شبیه به روش QL بوده و بعد از آن روش QL را شرح می دهیم و به راه های افزایش سرعت همگرایی این روش از جمله روش انتقال یافته می پردازیم.در نهایت برخی از کاربرد های محاسبه ی مقادیر ویژه ماتریس های سه قطری را شرح می دهیم.

الگوریتم QR

تجزیه QR

ماتریس A را در نظر می گیریم و A را به صورت حاصل ضرب دو ماتریس Q و R می نویسیم به طوری که Q متعامد بوده($Q^T . Q = Q . Q^T = I$) و R ماتریس بالا مثلثی باشد.

تکرار

در هر مرحله تجزیه ی QR ماتریس A_k را محاسبه می کنیم و ماتریس $\mathrm{A_{k+1}}$ را بدین ترتیب می سازیم $\mathrm{A_{k+1} = R_k . Q_k}$.همین طور این فرآیند ادامه می دهیم تا زمانی که ماتریس A_k بالا مثلثی بشود. درایه های قطر آن مقادیر ویژه ی ماتریس A خواهند بود.

تشابه ماتریس A با هر A_k

$\mathrm{A} = A_1 = Q_1 R_1$ آنگاه داریم $Q_1^{-1} A = R_1$ خواهیم داشت

$$A_۲ = R_۱ Q_۱ = (Q_۱^{-۱} A) Q_۱ = Q_۱^{-۱} A Q_۱$$

پس A با A_2 متشابه بوده و با همین استدلال می توان گفت کهA با هر A_k متشابه هست.

چرا QR

تجزیه ی QR برای هر ماتریس دلخواهی وجود دارد.

پایداری عددی[1] : تجزیه QR به طور کلی از نظر عددی پایدار است،به این معنی که کمتر در معرض خطاهای ناشی از گرد کردن در محاسبات است.

[1] Numerical Stability

```python
def qr_method(matrix, max_iterations=1000, tolerance=1e-10):
    A = matrix.copy()
    n = len(A)

    for iteration in range(max_iterations):
        Q, R = np.linalg.qr(A)
        A = np.dot(R, Q)

        if np.all(np.abs(np.tril(matrix, k=-1)) < tolerance):
            break

    eigenvalues = np.diagonal(A)
    return eigenvalues
```

کد پایتون روش تجزیه یQR

الگوریتم QL

الگوریتم QL، یک نسخه از الگوریتم QR، برای قطری کردن یک ماتریس هست. الگوریتم QL به خصوص برای ماتریس‌هایی که ویژگی‌های خاصی مانند سه قطری دارند، بسیار مفید است.
الگوریتم QL یک جایگزین برای الگوریتم QR است. این الگوریتم ماتریس سه قطری A را به صورت حاصلضربی از یک ماتریس متعامد Q و یک ماتریس پایین مثلثی L نشان می‌دهد.

تجزیه QL

برای محاسبه ی تجزیه ی QL ماتریس های سه قطری راه حل بهینه استفاده از صفحات چرخش[۲] هست.
برای این کار صفحات چرخش (P_{pq}) $P_{1,2}, P_{2,3}, \dots, P_{n-1,n}$را در نظر گرفته این صفحات برای صفر کردن درایه های $a_{12}, a_{23}, \dots, a_{n-1n}$ استفاده می شوند. بنابراین هر Q_s به شکل زیر از طریق حاصلضرب صفحات چرخش ساخته می شود.

[2] Plane Rotation

$$\mathbf{Q}_s^T = \mathbf{P}_1^{(s)} \cdot \mathbf{P}_2^{(s)} \cdots \mathbf{P}_{n-1}^{(s)}$$

به طوری که P_i برای صفر کردن درایه ی $a_{i,i+1}$ به کار می رود. در نظر داشته باشید که دلیل اینکه در تساوی بالا از Q^T استفاده کردیم و نه Q این بود که ماتریس L بدین شکل ساخته می شود. $(L = Q^T . A)$

صفحات چرخش در تجزیه ی QL به شکل زیر ساخته می شوند.

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \qquad c^2 + s^2 = 1$$

به طوری که c, s از رابطه ی زیر محاسبه می شوند.

$$c = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}}, \qquad s = \frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}$$

این ماتریس ها متعامد هستند در حالت ۲ در ۲ داریم:

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} c^2 + s^2 & cs - cs \\ cs - cs & c^2 + s^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

در حالت ۳ در ۳:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & s \\ 0 & -s & c \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c^2 + s^2 & 0 \\ 0 & 0 & c^2 + s^2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

در حالت کلی خواهیم داشت :

$$\begin{bmatrix} 1 & & & & & & & 0 \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & c & -s & & & \\ & & & s & c & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ 0 & & & & & & & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & & & & 0 \\ & 1 & & & & & & \\ & & \ddots & & & & & \\ & & & c & s & & & \\ & & & -s & c & & & \\ & & & & & \ddots & & \\ & & & & & & 1 & \\ 0 & & & & & & & 1 \end{bmatrix}$$

حاصلضرب سطر اول P در ستون های اول تا آخر P^T سطر زیر بوده

$$[۱,\ ۰,\ ۰,\ \dots,\ ۰]$$

که همان سطر اول ماتریس همانی است.برای سطر دوم P در ستون های اول تا آخر P^T سطر زیر را خواهیم داشت.

$$[۰,\ ۱,\ ۰,\ ۰,\ \dots,\ ۰]$$

با همین روند ادامه می دهیم و حاصضرب سطر k ام P درستون های اول تا آخر P^T سطر زیر خواهد بود.

$$[۰,\ ۰,\ \dots,\ ۰,\ \underbrace{۱}_{\text{درایه ی k ام}},\ ۰,\ \dots,\ ۰],\quad ۱ \le k \le i-۱$$

که همان سطر k ام ماتریس همانی هست.

حال اگر سطر i ام ماتریس P در ستون های اول تا آخر ماتریس P^T ضرب شود خواهیم داشت

$$[۰,\ ۰,\ \dots,\ ۰,\ \underbrace{c \times c + (-s)(-s)}_{\text{درایه ی i ام}},\ ۰,\ ۰,\ \dots,\ ۰]$$
$$= [۰\ \dots,\ ۰,\ \underbrace{۱}_{\text{درایه ی i ام}},\ ۰,\ \dots,\ ۰]$$

چون $c^2 + s^2 = 1$

برای سطر های $i < k < j$، سطر k ام ماتریس P را در ستون های اول تا آخر ماتریس P^T ضرب کنیم خواهیم داشت.

$$[۰, ۰, \dots, ۰, \underbrace{۱}_{\text{درایه‌ی } k \text{ ام}}, ۰, \dots, ۰]$$

که همان سطر k ام ماتریس همانی هست.

اگر سطر j ام ماتریس P را در ستون های اول تا آخر P^T ضرب کنیم خواهیم داشت.

$$[۰, ۰, \dots, \underbrace{s \times s + c \times c}_{\text{درایه‌ی } j \text{ ام}}, ۰, \dots, ۰]$$

$$= [۰, ۰, \dots, \underbrace{۱}_{\text{درایه‌ی } j \text{ ام}}, ۰, \dots, ۰]$$

و در نهایت برای $j < k \le n$ سطر k ام ماتریس P در ستون اول تا آخر P^T ضرب کرده و خواهیم داشت.

$$[0, 0, \ldots, 0, \underbrace{1}_{\text{ام } k \text{ درایه‌ی}}, 0, \ldots, 0]$$

که همان ماتریس همانی است.

مشاهده کردیم که PP^T ماتریس همانی شد.با روند مشابه مشاهده می شود که P^TP نیز همانی می شود.پس ماتریس P متعامد است.

از آنجایی که حاصل ضرب دو ماتریس متعامد ، متعامد است. ما توانستیم $\mathrm{Q_s}$ را به شکل زیر بسازیم به طوری که $\mathrm{Q_s}$ متعامد است.

$$\mathbf{Q}_s^T = \mathbf{P}_1^{(s)} \cdot \mathbf{P}_2^{(s)} \cdots \mathbf{P}_{n-1}^{(s)}$$

تکرار

تکرار در الگوریتم QL تقریبا شبیه QR عمل می کند. به این شکل که در هر مرحله تجزیه ی QL ماتریس A_s را محاسبه می کنیم $\mathrm{A_s = Q_s. L_s}$ و برای ماتریس $\mathrm{A_{s+1}}$ داریم که $\mathrm{A_{s+1} = L_s. Q_s}$.

تشابه ماتریس A با هر A_k

$\mathrm{A} = A_1 = Q_1L_1$ آنگاه داریم $Q_1^{-1}A = L_1$ خواهیم داشت

$$A_2 = L_1Q_1 = (Q_1^{-1}A)Q_1 = Q_1^{-1}A\,Q_1$$

پس A با A_2 متشابه بوده و با همین استدلال می توان گفت که A با هر A_k متشابه هست.

همگرایی

اگر مقدار قدر مطلق مقادیر ویژه $(|\lambda_i|)$ یکسان نباشد.الگوریتم بعد از چندین مرحله ماتریس $\mathrm{A_s}$ را تولید می کند که پایین مثلثی است و مقادیر ویژه به صورت صعودی روی قطر اصلی ظاهر می شوند.

پیچیدگی محاسباتی و زمان الگوریتم

پیچیدگی محاسباتی QL برای هر تکرار برای ماتریس های به شکل کلی $O(n^3)$ بوده اما برای ماتریس های سه قطری در هر تکرار $O(n)$ زمان می برد و این به این معنی هست که این الگوریتم برای ماتریس

های سه قطری بسیار بهینه است.همین طور این روش برای ماتریس های هسنبرگی $O(n^2)$ در هر تکرار زمان می برد.

کد پایتون روش QL

```
def ql_method(matrix,max_iteration = 1000,tolerance = 1e-10):
    n = len(matrix)
    iteration = 0
    while(iteration<max_iteration):
        Q = np.eye(n)
        L = np.eye(n)
        for k in range(n - 1, 0, -1):
            # Apply Givens rotation to eliminate subdiagonal element
            c = matrix[k-1, k-1] / np.sqrt(matrix[k-1, k-1]**2 + matrix[k, k-1]**2)
            s = matrix[k, k-1] / np.sqrt(matrix[k-1, k-1]**2 + matrix[k, k-1]**2)

            rotation_matrix = np.eye(n)
            rotation_matrix[k-1, k-1] = c
            rotation_matrix[k, k-1] = s
            rotation_matrix[k-1, k] = -s
            rotation_matrix[k, k] = c

            # Update the matrix Q
            Q = np.dot(Q, rotation_matrix.T)

        L = np.dot(Q.T,matrix)
        matrix = (np.dot(L,Q))

        if np.all(np.abs(np.tril(matrix, k=+1)) < tolerance):
            break
        iteration = iteration+1

    eigenvalues = np.diagonal(matrix)  # Extract eigenvalues
    return eigenvalues
```

کد پایتون روش QL

افزایش سرعت همگرایی

در الگوریتم QL سرعت همگرایی وابسته به $\frac{\lambda_i}{\lambda_j}$ (نرخ همگرایی) هر چند که در قبل به نابرابری λ_i و λ_j اشاره کردیم اما اگر این دو مقدار بسیار به هم نزدیک باشند سرعت همگرایی به شدت کاهش می یابد. به همین دلیل برای حل این مشکل از روش انتقال یافته بهره می بریم. به طوری که اگر k هر مقدار ثابتی باشد آنگاه $A - kI$ دارای مقادیرویژه ی $\lambda_i - k$ می باشد و نرخ همگرایی آن $\frac{\lambda_i - k}{\lambda_j - k}$ خواهد بود. یکی از مقادیر که می توانیم برای k در نظر بگیریم λ_1 (کوچکترین مقدار ویژه)

یک روش کارآمد اما اثبات نشده

استراتژی این روش به این شکل هست که مقادیر ویژه ی اولین زیر ماتریس۲در۲ را محاسبه کنیم و k را برابر با مقدار ویژه ای بگذاریم که به a_{11} نزدیک تر هست.

برای اینکه این موضوع را بهتر بیان کنیم فرض کند که $r - 1$ مقدار ویژه ی ماتریس A را پیدا کرده ایم.ماتریس A را به این شکل کاهش می دهیم که $r - 1$ سطر و ستون اول را صفر در نظر می گیریم ماتریس به شکل زیر می شود.

$$\mathbf{A} = \begin{bmatrix} 0 & & & \cdots & \cdots & & & 0 \\ & \cdots & & & & & & \\ & & 0 & & & & & \\ \vdots & & & d_r & e_r & & & \vdots \\ \vdots & & & e_r & d_{r+1} & & & \\ & & & & & \cdots & & 0 \\ & & & & & & d_{n-1} & e_{n-1} \\ 0 & & & \cdots & & 0 & e_{n-1} & d_n \end{bmatrix}$$

حال مقادیر ویژه ی ماتریس ۲در۲پیش رو را محاسبه می کنیم و مقدار ویژه ای را به عنوان مقدار ثابت k در نظر می گیریم که به d_r نزدیک تر باشد.مثال عددی زیر را در نظر بگیرید.

$$A = \begin{bmatrix} 4 & 1 & 0 & 0 \\ 1 & 3 & 2 & 0 \\ 0 & 2 & 5 & 3 \\ 0 & 0 & 3 & 6 \end{bmatrix}$$

فرض کنیم که دو مقدار ویژه ی اولیه این ماتریس را توانسته ایم محاسبه کنیم آنگاه با حذف دو سطر و دو ستون اول خواهیم داشت.

$$A = \begin{bmatrix} 5 & 3 \\ 3 & 6 \end{bmatrix}$$

مقادیر ویژه ی ماتریس بالا را محاسبه می کنیم.($\lambda_1 = 1/2(11 + \sqrt{37})$ و $\lambda_2 = 1/2(11 - \sqrt{37})$) بنابراین λ_2 به درایه ی اول نزدیکتر بوده و به عنوان مقدار ثابت k انتخاب می شود.

الگوریتم QL با انتقال های جزیی

این الگوریتم بر اساس یک لم هست:

ماتریس A را در نظر گرفته ماتریس B را به این شکل در نظر می گیریم $B = Q^T.A.Q$ به طوری که Q متعامد و B سه قطری با عناصر غیر قطری مثبت است.آنگاه ماتریس های Q و B با مشخص شدن سطر آخر Q^T مشخص می شوند.

اثبات:

معادله ی $B = Q^T.A.Q$ را در نظر گرفته که در آن B سه قطری با عناصر غیر قطری مثبت و Q متعامد است. q_i^T نشان دهنده ی iامین سطر ماتریس Q^T است.آنگاه q_i نشان دهنده ی iامین ستون ماتریس Q هست.بنابراین رابطه ی $B.Q^T = Q^T.A$ را به شکل زیر می توانیم بنویسیم

$$\begin{bmatrix} \beta_1 & \gamma_1 & & & & \\ \alpha_2 & \beta_2 & \gamma_2 & & & \\ & & & \vdots & & \\ & & & \alpha_{n-1} & \beta_{n-1} & \gamma_{n-1} \\ & & & & \alpha_n & \beta_n \end{bmatrix} \cdot \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_{n-1}^T \\ \mathbf{q}_n^T \end{bmatrix} = \begin{bmatrix} \mathbf{q}_1^T \\ \mathbf{q}_2^T \\ \vdots \\ \mathbf{q}_{n-1}^T \\ \mathbf{q}_n^T \end{bmatrix} \cdot \mathbf{A}$$

آنگاه برای nامین ماتریس تساوی زیر را داریم(تساوی ۱)

$$\alpha_n \mathbf{q}_{n-1}^T + \beta_n \mathbf{q}_n^T = \mathbf{q}_n^T \cdot \mathbf{A}$$

از آنجایی که ماتریس Q متعامد است خواهیم داشت(تساوی۲)

$$\mathbf{q}_n^T \cdot \mathbf{q}_m = \delta_{nm}$$

به طوری که

- $\delta_{nm} = 1$ if $n = m$.
- $\delta_{nm} = 0$ if $n \neq m$.

q_{n} را در دو طرف معادله ضرب می کنیم آنگاه خواهیم داشت(تساوی۳)

$$\beta_n = \mathbf{q}_n^T \cdot \mathbf{A} \cdot \mathbf{q}_n$$

از آنجایی که مقدار q_{n} مشخص است.از تساوی ۱ نتیجه می گیریم که (تساوی ۴)

$$\alpha_n \mathbf{q}_{n-1}^T = \mathbf{z}_{n-1}^T$$

به طوری که

$$\mathbf{z}_{n-1}^T \equiv \mathbf{q}_n^T \cdot \mathbf{A} - \beta_n \mathbf{q}_n^T$$

مقدار ${z^T}_{n-1}$ مشخص بوده بنابرین بنابر تساوی ۴ داریم(تساوی۵)

$$\alpha_n^2 = \mathbf{z}_{n-1}^T \mathbf{z}_{n-1},$$

$$\alpha_n = |\mathbf{z}_{n-1}|$$

و بنابر تساوی ۴ می توانیم داشته باشیم

$$\mathbf{q}_{n-1}^T = \mathbf{z}_{n-1}^T / \alpha_n$$

به کمک استقرا خواهیم داشت که اگر مقادیر $q_{\mathrm{n}}, q_{\mathrm{n-1}}, \dots, q_{\mathrm{n-j}}$ و همین طور α, β, γ تا مرحله ی $n-j$ داشته باشیم آنگاه می توانیم مقادیر را برای مرحله ی $n-(j+1)$ بدست بیاوریم.

حال از این لم در عمل بهره می بریم.فرض می کنیم که ماتریس A'_{s+1} که سه قطری است به شکل زیر

هست.(تساوی ۵)

$$A'_{s+1} = {Q'_s}^T . A'_s . Q'_s$$

به طوری که ${Q'_s}^T$ متعامد بوده و آخرین سطر آن برابر با Q_s در الگوریتم اصلی QL هست. بنابر لمی که اثبات کردیم نتیجه می گیریم که $Q'_s = Q_s$ و $A'_{s+1} = A_{s+1}$.

حال در الگوریتم اصلی QL داشتیم که $Q_s^T = P_1^{(s)} . P_2^{(s)} \dots P_{n-1}^{(s)}$ آخرین سطر Q_s^T برابر با $P_{n-1}^{(s)}$ بوده و از آنجایی که $P_{n-1}^{(s)}$ یک صفحه چرخشی است و برای صفر کردن درایه ی $(n-1, n)$ ماتریس $A_s - k_s I$ استفاده می شود.c و s به شکل زیر محاسبه می شوند.

$$c = \frac{d_n - k_s}{\sqrt{e_n^2 + (d_n - k_s)^2}} \quad , \quad s = \frac{-e_{n-1}}{\sqrt{e_n^2 + (d_n - k_s)^2}}$$

ماتریس $P_{n-1}^{(s)} . A_s . P_{n-1}^{(s)T}$ یک ماتریس سه قطری با درایه ی اضافه به شکل زیر خواهد بود.

$$\begin{bmatrix} \cdots & & & & & \\ & \times & \times & \times & & \\ & & \times & \times & \times & \mathbf{x} \\ & & & \times & \times & \times \\ & & & \mathbf{x} & \times & \times \end{bmatrix}$$

اکنون باید این ماتریس را به کمک یک ماتریس متعامد که آخرین سطرآن به این شکل هست $[0,0, \dots, 0,1]$ کاهش دهیم آنگاه آخرین سطر ${Q'_s}^T$ برابر با $P_{n-1}^{(s)}$ خواهد بود. این کار به کمک صفحات چرخش انجام می شود.برای صفر کردن درایه ی $(n-2, n)$ ماتریس را در صفحه ی $(n-2, n-1)$ دوران می دهیم.(درایه ی (n,n-2) نیز صفر خواهد شد). با این کار با یک ماتریس سه قطری و دو درایه ی اضافی $(n-3, n-1)$ و $(n-1, n-3)$ خواهیم داشت این بار در صفحه ی $(n-3, n-2)$ دوران می دهیم و به همین ترتیب ادامه می دهیم نتیجه به شکل زیر خواهد بود.

$$Q_s^T = {Q'_s}^T = {P'_1}^{(s)} . {P'_2}^{(s)} \dots {P'_{n-2}}^{(s)} . P_{n-1}^{s}$$

به طوری که P'ها دوران هایی بودند که در بالا صحبت کردیم و P_{n-1} همان صفحه ای است که در

الگوریتم اصلی بود.به کمک تساوی ۵ می توانیم تکرار های بعدی را نیز تولید کنیم.دقت کنید که اینجا ثابت k_sبه صورت جزیی از طریق پارامتر ها وارد می شوند. این الگوریتم در $O(n^2)$ عملیات قطری سازی را انجام دهد.

کاربرد ها

یادگیری ماشین

الگوریتم های متنوعی در یادگیری ماشین وجود دارند که ممکن است ماتریس سه قطری در آنها ظاهر شود و ما نیاز به محاسبه ی مقادیر ویژه ی آنها داشته باشیم از جملهPCA (Principal Component Analysis) که شامل محاسبه ی مقادیر ویژه ی ماتریس های کوواریانس می باشند.در این الگوریتم ماتریس های سه قطری ممکن است ظاهر شوند که محاسبه ی مقادیر ویژه ی آنها به فرآیند استخراج ویژگی ها و کاهش ابعاد کمک می کند.

نظریه گراف

ماتریس‌های سه قطری می‌توانند با انواع خاصی از گراف‌ها مرتبط باشند، به ویژه در مطالعه سیستم‌های خطی ناشی از تجزیه و تحلیل شبکه‌ها. محاسبات مقدار ویژه این ماتریس‌ها به درک اتصالات و مرکزیت گراف‌ها کمک می‌کنند.

مالی

ماتریس‌های سه‌قطری ممکن است در مدل‌سازی مالی ظاهر شوند، به ویژه در گسسته‌سازی معادلات دیفرانسیل جزئی برای قیمت‌گذاری گزینه‌ها. محاسبات مقدار ویژه برای درک رفتار مشتقات مالی و ارزیابی ریسک ضروری است.

منابع

Acton, F.S. (1970). Numerical Methods That Work. 1990, corrected edition. Washington: Mathematical Association of America, pp. 331–335. [1]

Wilkinson, J.H., and Reinsch, C. (1971). Linear Algebra, vol. II of Handbook for Automatic Computation. New York: Springer-Verlag. [2]

Smith, B.T., et al. (1976). Matrix Eigensystem Routines — EISPACK Guide, 2nd ed., vol. 6 of Lecture Notes in Computer Science. New York: Springer-Verlag. [3]

Stoer, J., and Bulirsch, R. (1980). Introduction to Numerical Analysis, §6.6.6. New York: Springer-Verlag. [4]