



Численные методы

Курс «Численные методы»

ВОЛКОВ Василий Михайлович, Минск, БГУ
v.volkov@tut.by

Минск, 13 ноября 2019



Лекция 6.

1.6 Итерационные методы нахождения собственных значений и собственных векторов.

Степенной метод.

$$A\mathbf{x} = \lambda\mathbf{x} \quad A \in R^{N \times N}$$

A – матрица простой структуры,

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_N|.$$

Идея степенного метода основано на том факте, что собственные векторы матрицы инвариантны относительно умножения на данную матрицу

$$A\mathbf{x}_k = \lambda_k \mathbf{x}_k$$

$$\lambda_{m \neq 1} / \lambda_1 < 1$$

Вычисление собственного вектора, соответствующего максимальному по модулю собственного значения .

Собственные векторы матрицы простой структуры линейно независимы . Следовательно, произвольный вектор \mathbf{y} можно разложить по базису данных собственных векторов.

$$\mathbf{y} = \sum_{m=1}^N \alpha_m \mathbf{x}_m \Rightarrow A\mathbf{y} = \sum_{m=1}^N \alpha_m \lambda_m \mathbf{x}_m$$

Рассмотрим последовательность

$$\mathbf{y}^{(1)} = A\mathbf{y}; \quad \mathbf{y}^{(2)} = A\mathbf{y}^{(1)}, \dots, \mathbf{y}^{(k)} = A\mathbf{y}^{(k-1)}.$$

$$\mathbf{y}^{(k)} = A\mathbf{y}^{(k-1)} = \lambda_1^k \left(\alpha_1 \mathbf{x}_1 + \sum_{m=2}^N \alpha_m \left(\frac{\lambda_m}{\lambda_1} \right)^k \mathbf{x}_m \right)$$

Алгоритм степенного метода.

1. формируем произвольный вектор $\mathbf{y}^{(0)}$
2. вычисляем вектор $\mathbf{y}^{(1)} = A\mathbf{y}^{(0)}$
3. нормируем полученный вектор $\mathbf{y}^{(1)} = \mathbf{y}^{(1)} / \|\mathbf{y}^{(1)}\|_\infty$
4. Повторяем п.п. 2,3 до тех пор, пока

$$\|\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}\| \geq \varepsilon$$

Выполнение неравенства $\|\mathbf{y}^{(k)} - \mathbf{y}^{(k-1)}\| \leq \varepsilon$

означает сходимость последовательности $\mathbf{y}^{(k)}$ к собственному вектору матрицы A , соответствующему Максимальному по модулю собственному значению λ_1 :

$$\lambda_1 = \text{sign}(y_m^{(k)} y_m^{(k-1)}) \lim_{k \rightarrow \infty} \|A\mathbf{y}^{(k)}\|_\infty / \|\mathbf{y}^{(k)}\|_\infty$$

Обобщения.

Можно ли с помощью степенного метода вычислить минимальное по модулю собственное значение?

Да, для этого достаточно воспользоваться свойством сдвига собственных значений

Скорость сходимости степенного метода определяется малостью отношения λ_2 / λ_1 . Можно ли ускорить сходимость?

В некоторых случаях это возможно с помощью Преобразования сдвига собственных значений

$$A \rightarrow A - \alpha E \quad \Rightarrow \quad \lambda(A - \alpha E) = \lambda(A) - \alpha$$

Пример: $\lambda(A) = \{2, 3, 4.5, 5, 5.25, 5.5, \}$; $\rho = \lambda_2 / \lambda_1 = 1.05 / 1.1 = 0.955$;
 $\lambda(A - 3.5E) = \{-1.5, -0.5, 1.0, 1.5, 1.75, 2.0, \}$; $\rho = \lambda_2 / \lambda_1 = 1.75 / 2.0 = 0.875$

Вычисление λ_2 ($|\lambda_2| < |\lambda_1|$)

Пусть мы вычислили максимальное по модулю собственное значение λ_1 . Для вычисления λ_2 рассмотрим последовательность

$$\mathbf{y}^{(k+1)} = A\mathbf{y}^{(k)} - \lambda_1\mathbf{y}^{(k)}, \quad \mathbf{y}^{(k+1)} = \mathbf{y}^{(k+1)} / \|\mathbf{y}^{(k+1)}\|_\infty$$

Поскольку в пределе больших k из текущего значения $\mathbf{y}^{(k+1)}$ вычитается собственный вектор, соответствующий λ_1 , то данная последовательность будет сходиться к собственному вектору, соответствующему λ_2 (по абсолютной величине следующему за λ_1). Для вычисления самого λ_2 можно использовать выражение

$$\lambda_2 = \text{sign}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}) \frac{\|A\mathbf{y}^{(k)} - \lambda_1\mathbf{y}^{(k)}\|_\infty}{\|A\mathbf{y}^{(k-1)} - \lambda_1\mathbf{y}^{(k-1)}\|_\infty},$$

Метод вращений

$$T_{km} = \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & \cos \alpha & \dots & -\sin \alpha & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \dots & \dots & \sin \alpha & \dots & \cos \alpha & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{bmatrix}.$$

Умножении матрицы T_{km} на вектор приводит к его повороту в плоскости единичных векторов \mathbf{e}_k , \mathbf{e}_m на угол α .

Свойство матрицы вращений

Матрица вращений ортогональная

$$T_{km}^{-1} = T_{km}^T$$

Преобразованием Гивенса (плоских вращений) называется преобразование подобия

$$\overline{A} \rightarrow T_{km}^T A T_{km}$$

Умножении матрицы T_{km} на вектор приводит к его повороту в плоскости единичных векторов \mathbf{e}_k , \mathbf{e}_m на угол α .

Умножении матрицы T_{km} на матрицу A справа (слева) в матрице A изменяются только k -й и m -й столбцы (строки).

Приведение матрицы к диагональному виду с помощью преобразования Гивенса

При преобразовании подобия с матрицей T_{km} можно обратить в ноль недиагональные элементы преобразуемой матрицы:

$$A^{(1)} = T_{km}^T A T_{km}, \quad a_{km}^{(1)} = a_{km} \cos 2\alpha + \frac{1}{2}(a_{mm} - a_{kk}) \sin 2\alpha.$$

$$a_{km} \cos 2\alpha + \frac{1}{2}(a_{mm} - a_{kk}) \sin 2\alpha = 0 \Rightarrow \tan 2\alpha = \frac{2a_{km}}{a_{kk} - a_{mm}} = \omega$$

$$s = \sin \alpha = \operatorname{sign}(\omega) \sqrt{\frac{\sqrt{1 + \omega^2} - 1}{2\sqrt{1 + \omega^2}}}$$

$$c = \cos \alpha = \sqrt{\frac{\sqrt{1 + \omega^2} + 1}{2\sqrt{1 + \omega^2}}}$$

Как работает метод плоских вращения?

Если применить преобразование Гивенса $\bar{A} \rightarrow T_{km}^T A T_{km}$

такого, что a_{km} — максимальный по модулю элемент A ,
а угол α — таков, что

$$\bar{a}_{km} = 0 \qquad \tan 2\alpha = \frac{2a_{km}}{a_{kk} - a_{mm}}.$$

то сумма квадратов модулей внедиагональных элементов преобразованной матрицы **уменьшается** (!!!) .

Как следствие, при многократном повторении преобразований Гивенса к матрице простой структуры преобразуемая матрица стремится к матрице, у которой внедиагональные элементы стремятся к нулю, а на диагонали остаются собственные значения исходной матрицы.

А что если матрица A не является матрицей простой структуры?

Алгоритм метода вращений

1. В матрице A находим максимальный по модулю **недиагональный** элемент a_{km} ;

2. Строим матрицу плоских вращений T_{km} :

$$T_{km} = \begin{bmatrix} 1 & 0 & \dots & \dots & \dots & \dots & 0 \\ 0 & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \dots & \dots & c & \dots & -s & \dots & \dots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \dots & \dots & s & \dots & c & \dots & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & \dots & \dots & \dots & \dots & 0 & 1 \end{bmatrix}, \quad s = \operatorname{sign}(\omega) \sqrt{\frac{\sqrt{1+\omega^2}-1}{2\sqrt{1+\omega^2}}}, \quad c = \cos \alpha = \sqrt{\frac{\sqrt{1+\omega^2}+1}{2\sqrt{1+\omega^2}}}$$
$$\omega = \frac{2a_{km}}{a_{kk} - a_{mm}}$$

3. Выполняем преобразование $A \rightarrow T_{km}^T A T_{km}$.

4. Повторяем п.п. 1-3 до тех пор, пока

$$\sum_{i \neq j} |a_{ij}|^2 > \varepsilon$$

QR-алгоритм.

Идея QR алгоритма состоит в представлении матрицы **A** в виде произведения ортогональной матрицы **Q** и верхней треугольной матрицы **R**: **A=QR**, $Q^{-1} = Q^T$.

1. Выполняем разложение $A=QR$
2. Вычисляем матрицу $A_1=RQ=Q^T QRQ=Q^T A Q$
3. Выполняем разложение $A_1=Q_1 R_1$
4. Вычисляем матрицу $A_2=R_1 Q_1=Q_1^T Q_1^T R_1 Q_1$

Повторяя процедуру имеем последовательность подобных матриц

$$A_{k+1} = R_k Q_k = Q_k^T Q_k R_k Q_k = Q_k^T A_k Q_k$$

Сходимость QR-алгоритма

Если исходная матрица A – матрица простой структуры, у которой отсутствуют комплексные (кратные) собственные значения, то поддиагональные элементы матрицы A_k стремятся к нулю и в пределе при больших k матрицы A_k стремятся к верхней треугольной матрице, на диагонали которой в силу подобия всех матриц последовательности будут находиться собственные значения исходной матрицы A . Если же среди собственных значений матрицы A имеются кратные и (или) комплексные, то в пределе последовательности будет наблюдаться сходимость A_k к блочно-треугольному виду, причем размерность диагональных блоков определяется кратностью соответствующих им собственных значений. Комплексно-сопряженным парам собственных значений соответствуют блоки 2×2 .

Преобразования Хаусхолдера

- Преобразование Хаусхолдера (преобразование отражений) выражается ортогональной матрицей

$$H = E - 2\mathbf{w}\mathbf{w}^T$$

где \mathbf{w} — произвольный вектор единичной длины $\mathbf{w}^T\mathbf{w}=1$

Несложно показать, что $H^{-1} = H^T$

Умножение матрицы H на вектор соответствует преобразованию отражения от плоскости, ортогональной вектору \mathbf{w} .

QR разложение на основе преобразования Хаусхолдера

• Если в качестве вектора w взять вектор

$$w_1 = (a_1 + \|a_1\|e_1) / \|a_1 + \|a_1\|e_1\|$$

$$\text{где } e_1 = (1, 0, 0, \dots, 0)^T, \quad a_1 = (a_{11}, a_{21}, \dots, a_{N1})^T$$

a_1 — первый столбец матрицы A .

Тогда преобразование $A_1 = H_1 A$, где $H_1 = E - w_1 w_1^T$ приводит к обнулению поддиагональных элементов первого столбца в A_1 .

Повторяя преобразование с матрицей H_2, H_3, \dots, H_{N-1}

$$H_k = E - w w^T, \quad w_k = (a_k + \|a_k\|e_k) / \|a_k + \|a_k\|e_k\|$$

$$a_m = (0, 0, \dots, a_{mm}, a_{m+1,m}, \dots, a_{Nm})$$

будем иметь

$$H_{N-1} \cdots H_2 \cdot H_1 A = R \quad \Rightarrow \quad A = H_1^T H_2^T \cdots H_{N-1}^T R = QR$$

Некоторые модификации

- 1. Сходимость QR-алгоритма определяется следующим соотношением

$$a_{nm}^{(k)} = O\left(|\lambda_m|^{-k} |\lambda_n|^k\right)$$

Как и в случае степенного метода, скорость сходимости может быть повышена на основе свойства сдвига собственных значений

- 2. Вычислительная сложность одной итерации $O(N^3)$. Эффективность алгоритма может быть повышена, если предварительно привести матрицу к почти треугольному виду (верхняя треугольная матрица с одной дополнительной ненулевой диагональю). В этом случае вид почти треугольной матрицы сохраняется в процессе выполнения QR-алгоритма и вычислительная сложность алгоритма Повышается до $O(N^2)$

Пример

```

A=pascal(4);
a1=A(:,1); e1=[1 0 0 0]';
norm_a1=norm(a1);
w=a1+e1*norm_a1;
w=w/norm(w);
H1=eye(N)-2*w*w';
A=H1*A

```

$$A = \begin{array}{c|cccc} \Gamma & 1 & 1 & 1 & | \\ | & 1 & 2 & 3 & 4 & | \\ | & 1 & 3 & 6 & 10 & | \\ L & 1 & 4 & 10 & 20 & | \end{array}$$

```

a2=[0;A(2:end,2)]; e2=[0 1 0 0]';
norm_a2=norm(a2);
w=a2+e2*norm_a2;
w=w/norm(w);
H2=eye(N)-2*w*w';
A=H2*A

```

$$H_1 A = \begin{array}{c|cccc} \Gamma & -2.0000 & -5.0000 & -10.0000 & -17.5000 & | \\ | & 0.0000 & -0.0000 & -0.6667 & -2.1667 & | \\ | & 0.0000 & 1.0000 & 2.3333 & 3.8333 & | \\ L & 0.0000 & 2.0000 & 6.3333 & 13.8333 & | \end{array}$$

```

a3=[0; 0; A(3:end,3)]; e3=[0 0 1 0]';
norm_a3=norm(a3);
w=a3+e3*norm_a3;
w=w/norm(w);
H3=eye(N)-2*w*w';
A=H3*A

```



СПАСИБО ЗА ВНИМАНИЕ!