



МГТУ имени Н.Э. Баумана

Кафедра ИУ-1 «Системы автоматического управления»

Методы вычислений

Численные методы матричных разложений



Андрей Леонидович Масленников
amas@bmstu.ru

2023 г.

Матричное разложение (декомпозиция) — это представление исходной матрицы в виде произведения других матриц, обладающих некоторым определенными свойствами

Разложения связанные с собственными или сингулярными значениями:

1. спектральное разложение;
2. разложение в каноническую форму Жордана;
3. разложение Шура;
4. QZ-разложение;
5. сингулярное разложение.

Алгоритмические разложения:

1. LU-разложение и его модификации;
2. LL-разложение и его модификации;
3. ранговая факторизация;
4. QR-разложение;
5. интерполяционное разложение;
6. полярное разложение.

Спектральное разложение:

$$\mathbf{A} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$$

- \mathbf{V} — матрица собственных векторов матрицы \mathbf{A}
- $\mathbf{\Lambda}$ — диагональная матрица с собственными значениями матрицы \mathbf{A}

Особенности:

1. только для матриц имеющих полный набор собственных значений;
2. алгоритмы требуют больших вычислительных ресурсов;
3. алгоритмы не универсальны.

решением могут быть только несколько первых собственных значений или подходят только для матриц определенного вида

Каноническая форма Жордана:

$$\mathbf{A} = \mathbf{CJC}^{-1}$$

C — матрица перехода к новому базису

J — матрица Жордана

$$\mathbf{J} = \begin{bmatrix} \lambda_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & \lambda_1 & 1 & 0 & 0 & 0 \\ 0 & 0 & \lambda_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \lambda_3 & 1 \\ 0 & 0 & 0 & 0 & 0 & \lambda_3 \end{bmatrix} = \begin{bmatrix} \mathbf{J}_1 & 0 & 0 \\ 0 & \mathbf{J}_2 & 0 \\ 0 & 0 & \mathbf{J}_3 \end{bmatrix}$$

Разложение в каноническую форму Жордана – обобщение спектрального разложения на случай кратных собственных значений матрицы A

Разложение Шура:

$$\mathbf{A} = \mathbf{UTU}^*$$

U — унитарная матрица

T — верхняя треугольная матрица

U* — эрмитово-сопряженная матрица

Особенности:

1. существует для квадратной матрицы;
2. разложение может быть не единственным.

Обобщенное разложение Шура (QZ-разложение) — это согласованное разложение двух матриц **A** и **B** следующего вида:

$$\mathbf{A} = \mathbf{QSZ}^*$$

$$\mathbf{B} = \mathbf{QTZ}^*$$

Разложение Шура:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$$

\mathbf{U} — унитарная матрица

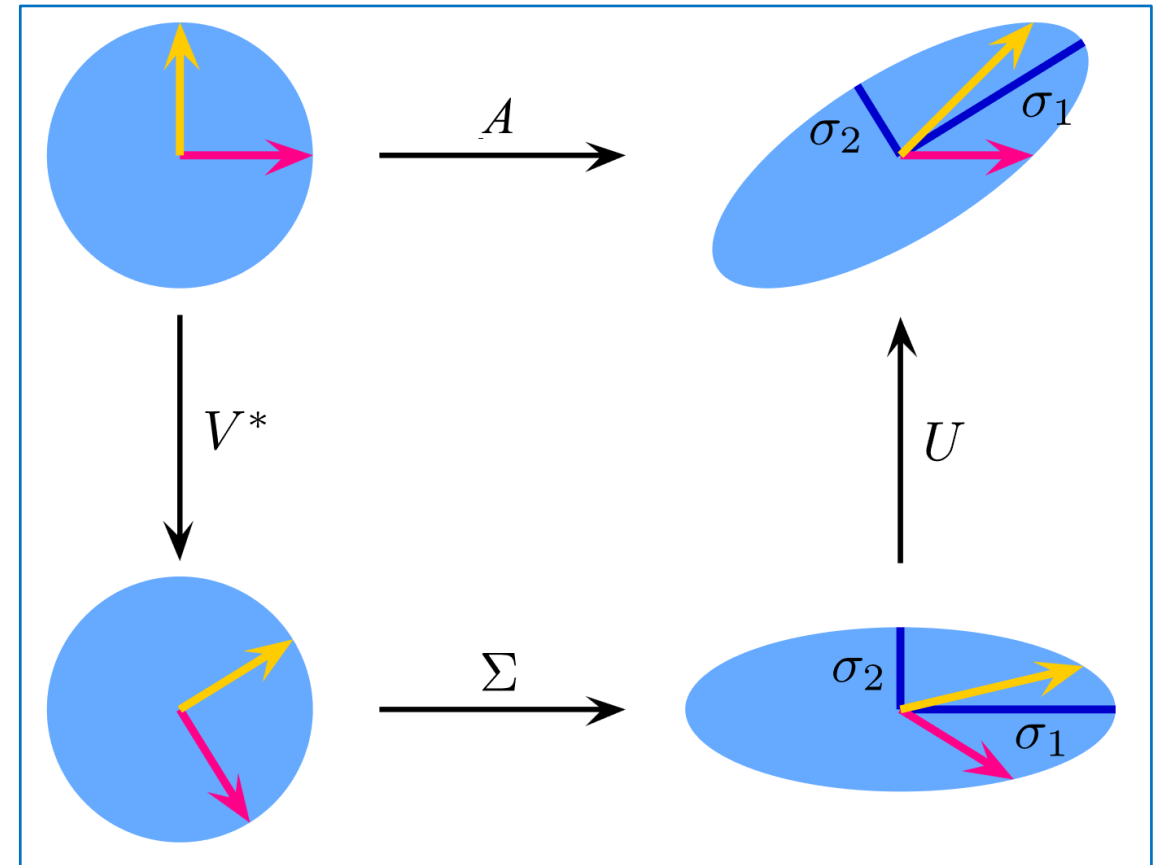
$\mathbf{\Sigma}$ — диагональная матрица с сингулярными значениями матрица \mathbf{A}

\mathbf{V}^* — эрмитово-сопряженная матрица

Особенности:

1. матрица \mathbf{U} состоит из левых сингулярных векторов матрицы $\mathbf{A}\mathbf{A}^*$;
2. матрица \mathbf{V} состоит из правых сингулярных векторов матрицы $\mathbf{A}^*\mathbf{A}$.

Сингулярное разложение (SVD) может использоваться для нахождения псевдо-обратных матриц



LU-разложение:

$$\mathbf{A} = \mathbf{L}\mathbf{U}$$

В зависимости от алгоритмов матрицы \mathbf{L} и \mathbf{U} могут обладать разными свойствами.

Алгоритмы:

1. метод Гаусса;
низкая точность для разреженных матриц, большой объем вычислений
2. алгоритм Дулиттла;
 \mathbf{L} – унитреугольная, \mathbf{U} – верхняя треугольная
3. алгоритм Кроута;
 \mathbf{L} – нижняя треугольная, \mathbf{U} – унитреугольная

Разложение существует при неравенстве нулю всех главных миноров матрицы \mathbf{A}

LUP-разложение:

$$\mathbf{P}\mathbf{A} = \mathbf{L}\mathbf{U}$$

$$\mathbf{A}\mathbf{Q} = \mathbf{L}\mathbf{U}$$

$$\mathbf{P}\mathbf{A}\mathbf{Q} = \mathbf{L}\mathbf{U}$$

\mathbf{P} — матрица перестановок по строкам

\mathbf{Q} — матрица перестановок по столбцам

LDU-разложение:

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{U}$$

\mathbf{D} — диагональная матрица

Вычисление определителя матрицы:

$$\det(\mathbf{A}) = \det(\mathbf{L}\mathbf{U}) = \det(\mathbf{L}) \det(\mathbf{U}) = \left(\prod_{i=1}^n l_{ii} \right) \left(\prod_{i=1}^n u_{ii} \right)$$

Алгоритм Дулиттла:

for $i = 1 : n$

for $j = i : n$

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$$

end

$$l_{ii} = 1$$

for $j = i + 1 : n$

$$l_{ji} = \frac{1}{u_{ii}} \left(a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki} \right)$$

end

end

Алгоритм Кроута:

for $i = 1 : n$

for $j = i : n$

$$l_{ji} = a_{ji} - \sum_{k=1}^{i-1} l_{jk} u_{ki}$$

end

$$u_{ii} = 1$$

for $j = i + 1 : n$

$$u_{ij} = \frac{1}{l_{ii}} \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} \right)$$

end

end

LL-разложение:

$$\mathbf{A} = \mathbf{L}\mathbf{L}^* = \mathbf{U}^*\mathbf{U}$$

Матрица \mathbf{L} всегда нижняя треугольная, а матрица \mathbf{U} всегда верхняя треугольная.

Алгоритмы:

1. алгоритм Холецкого;
на основе метода исключения Гаусса
2. алгоритм Холецкого-Банахевича;
вычисляется матрица \mathbf{L}
3. алгоритм Холецкого-Кроута;
вычисляется матрица \mathbf{U}

Критерием применимости алгоритмов LL-разложения является квадратичность матрицы \mathbf{A} , а условием единственности решения — ее положительная определенность

LDL-разложение:

$$\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^*$$

\mathbf{D} — диагональная матрица

\mathbf{L} — нижняя унитреугольная матрица

При вычислении LDL-разложения, в отличие от LL-разложения, не требуется вычисление квадратных корней, а условием существования разложения является отличие от нуля всех угловых миноров исходной матрицы \mathbf{A}

Алгоритм Холецкого-Банахевича:

for $i = 1 : n$

for $j = 1 : i - 1$

$$l_{ij} = \frac{1}{l_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}^* \right)$$

end

$$l_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} l_{ik} l_{ik}^*}$$

end

Алгоритм LDL-разложения:

for $i = 1 : n$

$$l_{ii} = 1$$

for $j = 1 : i - 1$

$$l_{ij} = \frac{1}{d_{jj}} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} d_{kk} l_{jk}^* \right)$$

end

$$d_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{kj} l_{kj}^* d_{kk}$$

end

Banachiewicz T. *Principes d'une nouvelle technique de la méthode des moindres carrés; Méthode de résolution numérique des équations linéaires, du calcul des déterminants et des inverses et de réduction des formes quadratiques.* Bull. Inter. Acad. Polon. Sci., Sér. A, 1938, pp. 393-404.

Пример LL-разложения:

$$\mathbf{A} = \begin{bmatrix} 1.4 & 1 & 1 \\ 1 & 0.9 & 1 \\ 1 & 1 & 1.4 \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} 1.18 & 0 & 0 \\ 0.85 & 0.43 & 0 \\ 0.85 & 0.66 & 0.5 \end{bmatrix}$$

$$i = 1, j = 0 \dots l_{10} = ???$$

$$l_{11} = \sqrt{1.4 - 0} = 1.18$$

$$l_{21} = \frac{1}{1.18}(1 - 0) = 0.85$$

$$l_{22} = \sqrt{0.9 - 0.85^2} = 0.43$$

$$l_{31} = \frac{1}{1.18}(1 - 0) = 0.85$$

$$l_{32} = \frac{1}{0.43}(1 - 0.85^2) = 0.66$$

$$l_{33} = \sqrt{1.4 - (0.85^2 + 0.66^2)} = 0.5$$

Пример LL-разложения:

$$\mathbf{A} = \begin{bmatrix} 1.4 & 1 & 2 \\ 1 & 0.9 & 1 \\ 1 & 1 & 1.4 \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} 1.18 & 0 & 0 \\ 0.85 & 0.43 & 0 \\ 0.85 & 0.66 & 0.5 \end{bmatrix}$$

Проверка:

$$\mathbf{LL}^T = \begin{bmatrix} 1.4 & 1 & 1 \\ 1 & 0.9 & 1 \\ 1 & 1 & 1.4 \end{bmatrix}$$

QR-разложение:

$$\mathbf{A} = \mathbf{QR}$$

Матрица \mathbf{Q} всегда ортогональная, а матрица \mathbf{R} всегда верхняя треугольная. Единственность разложения гарантируется при положительных диагональных элементах матрицы \mathbf{R} .

Алгоритмы:

1. классический алгоритм Грамма-Шмидта;
ортогонализация Грамма-Шмидта
2. модифицированный алгоритм Грамма-Шмидта;
более вычислительно устойчив
3. метод Хаусхолдера;
более вычислительно устойчив
4. метод Гивенса;
метод вращений

Вычисление обратной матрицы:

$$\mathbf{A}^{-1} = (\mathbf{QR})^{-1} = \mathbf{R}^{-1}\mathbf{Q}^{-1} = \mathbf{R}^{-1}\mathbf{Q}^T$$

Классический алгоритм Грамма-Шмидта:

for $i = 1 : n$

$$\mathbf{b}_i = \mathbf{a}_i - \sum_{k=1}^{i-1} \text{proj}_{\mathbf{b}_k} \mathbf{a}_i$$

end

$$\mathbf{Q} = \begin{bmatrix} \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|} & \frac{\mathbf{b}_2}{\|\mathbf{b}_2\|} & \dots & \frac{\mathbf{b}_n}{\|\mathbf{b}_n\|} \end{bmatrix}$$

$$\mathbf{R} = \mathbf{Q}^T \mathbf{A}$$

где оператор проекции $\text{proj}_{\mathbf{b}} \mathbf{a} = \frac{\mathbf{a} \cdot \mathbf{a}}{\mathbf{b} \cdot \mathbf{b}} \mathbf{b}$

Модифицированный алгоритм Грамма-Шмидта:

for $i = 1 : n$

$$\mathbf{b}_i = \mathbf{a}_i - \sum_{k=1}^{i-1} \text{proj}_{\mathbf{b}_k} \mathbf{b}_i$$

end

...

Пример алгоритма Грамма-Шмидта:

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

$$\mathbf{b}_1 = \mathbf{a}_1 - \sum_{k=1}^{1-1=0} \dots = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\mathbf{b}_2 = \mathbf{a}_2 - \sum_{k=1}^{2-1=1} \text{proj}_{\mathbf{b}_1} \mathbf{a}_2 = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \frac{2 \cdot 1 + 1 \cdot 2}{2 \cdot 2 + 1 \cdot 1} \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.6 \\ 1.2 \end{bmatrix}$$

$$\|\mathbf{b}_1\| = \sqrt{2^2 + 1^2} = 2.2361$$

$$\|\mathbf{b}_2\| = \sqrt{(-0.6)^2 + 1.2^2} = 1.3416$$

$$\mathbf{Q} = \begin{bmatrix} 0.8944 & -0.4472 \\ 0.4472 & 0.8944 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 0.8944 & -0.4472 \\ 0.4472 & 0.8944 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 2.2361 & 1.7889 \\ 0 & 1.3416 \end{bmatrix}$$

По сути, в классическом алгоритме каждый вектор \mathbf{b}_i вычисляется ортогональным ко всем предыдущим векторам, что приводит к накоплению большой вычислительной ошибки

В модифицированном алгоритме процедура ортогонализации осуществляется только относительно одного предыдущего вектора \mathbf{b}_i , а не всех

Метод Хаусхолдера:

$$\mathbf{Q} = \mathbf{I}$$

$$\mathbf{R} = \mathbf{A}$$

for $i = 1 : n - 1$

$$\mathbf{e} = \mathbf{0}$$

$$\mathbf{x} = \mathbf{a}_i$$

$$x_{1, \dots, i-1} = 0$$

$$e_i = 1$$

$$\mathbf{u} = \mathbf{x} - \|\mathbf{x}\| \cdot \mathbf{e}$$

$$\mathbf{P} = \mathbf{I} - \frac{2\mathbf{u}\mathbf{u}^T}{\|\mathbf{u}\|^2}$$

$$\mathbf{Q} = \mathbf{QP}$$

$$\mathbf{A} = \mathbf{PA}$$

$$\mathbf{R} = \mathbf{A}$$

end**Пример:**

$$i = 1$$

$$\mathbf{e} = \mathbf{0}$$

$$\mathbf{x} = \mathbf{a}_1 = \begin{bmatrix} 1.4 \\ 1 \\ 1 \end{bmatrix}$$

$$x_{1, \dots, 0} = 0$$

$$e_1 = 1$$

$$\|\mathbf{x}\| = \sqrt{1.4^2 + 1^2 + 1^2} = 1.99$$

$$\mathbf{u} = \begin{bmatrix} 1.4 \\ 1 \\ 1 \end{bmatrix} - 1.99 \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -0.59 \\ 1 \\ 1 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1.4 & 1 & 1 \\ 1 & 0.9 & 1 \\ 1 & 1 & 1.4 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 0.7035 & 0.5025 & 0.5025 \\ 0.5025 & 0.1482 & -0.8518 \\ 0.5025 & -0.8518 & 0.1482 \end{bmatrix}$$

$$\mathbf{Q} = \mathbf{IP} = \begin{bmatrix} 0.7035 & 0.5025 & 0.5025 \\ 0.5025 & 0.1482 & -0.8518 \\ 0.5025 & -0.8518 & 0.1482 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1.9900 & 1.6583 & 1.9096 \\ 0 & -0.2158 & -0.5417 \\ 0 & -0.1158 & -0.1417 \end{bmatrix}$$

$$\mathbf{R} = \mathbf{A}$$

Пример (продолжение):

$i = 2$

$$\mathbf{e} = \mathbf{0}$$

$$\mathbf{x} = \mathbf{a}_2 = \begin{bmatrix} 1.6583 \\ -0.2158 \\ -0.1158 \end{bmatrix}$$

$$x_1 = 0$$

$$\mathbf{x} = \begin{bmatrix} 0 \\ -0.2158 \\ -0.1158 \end{bmatrix}$$

$$e_2 = 1 \Rightarrow \mathbf{e} = [0 \ 1 \ 0]^T$$

$$\|\mathbf{x}\| = 0.2449$$

$$\mathbf{u} = \begin{bmatrix} 0 \\ -0.2158 \\ -0.1158 \end{bmatrix} - 0.2449 \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -0.4608 \\ -0.1158 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1.9900 & 1.6583 & 1.9096 \\ 0 & -0.2158 & -0.5417 \\ 0 & -0.1158 & -0.1417 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.8811 & -0.4729 \\ 0 & -0.4729 & 0.8811 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} 0.7035 & -0.6804 & 0.2052 \\ 0.5025 & 0.2722 & -0.8206 \\ 0.5025 & 0.6804 & 0.5334 \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} 1.9900 & 1.6583 & 1.9096 \\ 0 & 0.2449 & 0.5443 \\ 0 & 0 & 0.1313 \end{bmatrix}$$

$$\mathbf{R} = \mathbf{A}$$

Метод вращений Гивенса

$$\mathbf{Q} = \mathbf{I}$$

$$\mathbf{R} = \mathbf{A}$$

for $i = 1 : n - 1$

for $j = 1 : n$

$$\mathbf{G} = \mathbf{I}$$

$$G_{ii} = c \quad d = \sqrt{a_{ii}^2 + a_{ji}^2}$$

$$G_{ji} = c \quad c = a_{ii} / d$$

$$G_{ij} = s \quad s = a_{ij} / d$$

$$G_{jj} = -s$$

$$\mathbf{A} = \mathbf{GR}$$

$$\mathbf{R} = \mathbf{A}$$

$$\mathbf{Q} = \mathbf{QG}^T$$

end

end

Пример матрицы вращений

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Фактически алгоритмы не вычисляют непосредственно матрицу \mathbf{G} целиком, а обновляют ее значения

В основе метода Гивенса (метод вращений) лежит последовательность из нескольких поворотов Гивенса (линейных преобразований матрицы \mathbf{A})