Matrix decomposition

Matrix Decomposition

- Matrix decomposition (or matrix factorization) is a factorization of a matrix into a product of matrices. A common analogy for matrix decomposition is the factoring of numbers, such as the factoring of 10 into 2 x 5
- Methods
  - ✓ Linear equations solving related
    - LU decomposition
    - QR decomposition
  - ✓ Eigenvalues related
    - Eigen decomposition
    - Singular value decomposition (SVD)
- Applications

https://en.wikipedia.org/wiki/Matrix_decomposition

Gaussian elimination for linear equations $\boldsymbol{Ax} = \boldsymbol{b}$

$$\begin{cases} x+2y+z=2 \\ 3x+8y+z=12 \\ 4y+z=2 \end{cases} \qquad \begin{cases} x+2y+z=2 \\ 2y-2z=6 \\ 4y+z=2 \end{cases} \qquad \begin{cases} x+2y+z=2 \\ 2y-2z=6 \\ 5z=-10 \end{cases}$$

- Augmented matrix

$$\left(\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 3 & 8 & 1 & 12 \\ 0 & 4 & 1 & 2 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -2 & 6 \\ 0 & 4 & 1 & 2 \end{array}\right) \Rightarrow \left(\begin{array}{ccc|c} 1 & 2 & 1 & 2 \\ 0 & 2 & -2 & 6 \\ 0 & 0 & 5 & -10 \end{array}\right)$$

其中最后一个矩阵对角线上元素1, 2, 5称为主元（**pivot**）

最终得到上三角（**upper**）矩阵。

Can we summarize the process of Gaussian elimination in matrix form ?

https://zhuanlan.zhihu.com/p/36575900

LU decomposition

- $\underbrace{L_{m-1}L_{m-2}\dots L_2L_1}_{=\widetilde{L}} A = U.$

- $\widetilde{L}A = U \quad \Longleftrightarrow \quad A = LU,$ A is square matrix, L is lower triangular and U is upper triangular

- $L\underbrace{U\boldsymbol{x}}_{=\boldsymbol{y}} = \boldsymbol{b}.$ first solve ***y***, then solve ***x***, advantage?

- Example

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix} \qquad L_1A = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 2 & 4 \end{bmatrix}.$$

$$L_2(L_1A) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -2 \end{bmatrix} = U.$$

http://www.math.iit.edu/~fass/477577_Chapter_7.pdf
https://iuuk.mff.cuni.cz/~rakdver/linalg/lesson15-12.pdf
https://math.stackexchange.com/questions/266355/necessity-advantage-of-lu-decomposition-over-gaussian-elimination

LU decomposition

- Example

$$L = (L_2 L_1)^{-1} = L_1^{-1} L_2^{-1} \qquad L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix} \qquad \text{and } L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}, \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix}.$$

- Double check

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 5 \\ 4 & 6 & 8 \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 4 & 2 & 1 \end{bmatrix}. \qquad \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 3 \\ 0 & 0 & -2 \end{bmatrix} = U.$$

- Permutation (LU factorization with partial pivoting (*LUP*)
  - ✓ All square matrices can be factorized in this form, and the factorization is numerically stable

$$L_1 A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 3 \\ 0 & 2 & 3 \end{bmatrix} \qquad PL_1 A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} L_1 A = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \\ 0 & 0 & 3 \end{bmatrix}$$

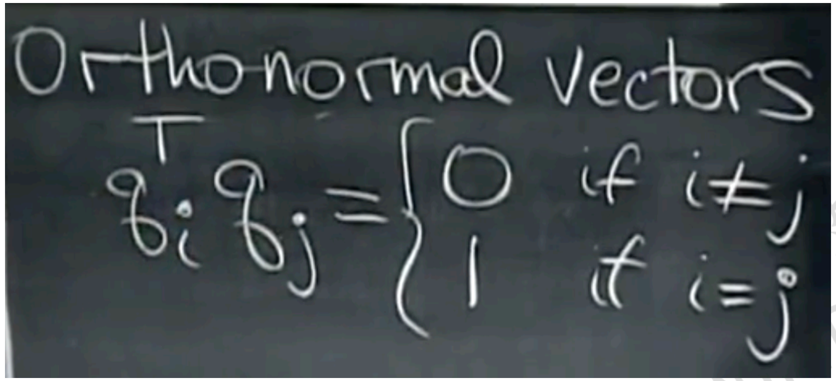
http://www.math.iit.edu/~fass/477577_Chapter_7.pdf
https://open.163.com/newview/movie/free?pid=NESCFTN3V&mid=IESCHS624

QR decomposition

$A = QR,$ Q is an orthogonal matrix, R is an upper triangular matrix

- Orthogonal matrix is a real square matrix whose columns and rows are orthonormal vectors

$$Q^{\mathrm{T}}Q = QQ^{\mathrm{T}} = I, \qquad Q^{\mathrm{T}} = Q^{-1},$$



- Gram-Schmidt process is a method for orthonormalizing a set of vectors in an inner product space

https://open.163.com/newview/movie/free?pid=NESCFTN3V&mid=YESCKDIQD
https://www.math.ucla.edu/~yanovsky/Teaching/Math151B/handouts/GramSchmidt.pdf

QR decomposition

- Example

$$\mathbf{A} = \begin{pmatrix} 1 & -1 & 4 \\ 1 & 4 & -2 \\ 1 & 4 & 2 \\ 1 & -1 & 0 \end{pmatrix}.$$

- Gram-Schmidt process

Let $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3)$, the Q-factor of $\mathbf{A}$ be $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$, and the R-factor be

$$\mathbf{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{pmatrix}.$$

The Gram-Schmidt process of computing QR decomposition

1. $r_{11} = \|\mathbf{a}_1\| = 2.^1$ $\mathbf{q}_1 = \frac{1}{\|\mathbf{a}_1\|}\mathbf{a}_1 = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 1/2 \end{pmatrix}.$

2. $r_{12} = \mathbf{q}_1^T\mathbf{a}_2 = \begin{pmatrix} 1/2 & 1/2 & 1/2 & 1/2 \end{pmatrix}\begin{pmatrix} -1 \\ 4 \\ 4 \\ -1 \end{pmatrix} = 3$

http://www.cs.nthu.edu.tw/~cherung/teaching/2008cs3331/chap4%20example.pdf

Eigen decomposition

- $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ λ is eigenvalues, **v** is eigenvector

➢ Example $A = \begin{pmatrix} 4 & 3 \\ 2 & -1 \end{pmatrix}$

That is, the **eigenvectors** are the vectors that the linear transformation **A** merely elongates or shrinks, and the amount that they elongate/shrink by is the **eigenvalue**.

$\det(\mathbf{A} - \lambda\mathbf{I}) = 0.$

$$det(A - \lambda I) = det(\begin{pmatrix} 4 & 3 \\ 2 & -1 \end{pmatrix} - \lambda\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) = det\begin{pmatrix} 4-\lambda & 3 \\ 2 & -1-\lambda \end{pmatrix} = 0$$

$$det(A - \lambda I) = (4-\lambda)(-1-\lambda) - 3 * 2 = \lambda^2 - 3\lambda - 10 = (\lambda + 2)(\lambda - 5) = 0$$

$$\begin{pmatrix} 4 & 3 \\ 2 & -1 \end{pmatrix} * \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 5\begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad \begin{pmatrix} 4v_1 + 3v_2 \\ 2v_1 - 1v_2 \end{pmatrix} = \begin{pmatrix} 5v_1 \\ 5v_2 \end{pmatrix}$$

$$\begin{pmatrix} 4 & 3 \\ 2 & -1 \end{pmatrix} * \begin{pmatrix} w_1 \\ w_2 \end{pmatrix} = -2\begin{pmatrix} w_1 \\ w_2 \end{pmatrix} \quad \begin{pmatrix} 4w_1 + 3w_2 \\ 2w_1 - 1w_2 \end{pmatrix} = \begin{pmatrix} -2w_1 \\ -2w_2 \end{pmatrix}$$

$$v = \begin{pmatrix} 3 \\ 1 \end{pmatrix} \quad w = \begin{pmatrix} -1 \\ 2 \end{pmatrix}$$

https://online.stat.psu.edu/statprogram/reviews/matrix-algebra/eigendecomposition
https://math.stackexchange.com/questions/243533/how-to-intuitively-understand-eigenvalue-and-eigenvector

Eigen decomposition

- Put together the set of eigenvectors of **A** in a matrix denoted **Q**. Each column of **Q** is an eigenvector of **A**.
- The eigenvalues are stored in a diagonal matrix (denoted **Λ**)

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v} \qquad \mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{\Lambda} \qquad \mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}$$

```
from numpy.linalg import inv
```

```
Q = np.array([[3, -1], [1, 2]])
Q_inv = inv(Q)
Λ = np.array([[5, 0], [0, -2]])
```

```
Q.dot(Λ).dot(Q_inv)
```

```
array([[ 4.,  3.],
       [ 2., -1.]])
```

Singular value decomposition (best)

The Singular Value Decomposition: Let A be any $m \times n$ matrix. Then there are orthogonal matrices U, V and a diagonal matrix Σ such that

$$A = U\Sigma V^T$$

$$A^T A = (U\Sigma V^T)^T U\Sigma V^T = V\Sigma^T U^T U\Sigma V^T = V\Sigma^T \Sigma V^T = V\Sigma^2 V^T$$

The columns of V are the eigenvectors of $A^T A$.

The non-zero singular values are the square roots of the non-zero eigenvalues of $A^T A$ or AA^T

The columns of U are the eigenvectors of AA^T

http://people.whitman.edu/~hundledr/courses/M240S16/M240/TheSVD.pdf

SVD example

$$A = \begin{pmatrix} 3 & 2 & 2 \\ 2 & 3 & -2 \end{pmatrix}. \qquad AA^T = \begin{pmatrix} 17 & 8 \\ 8 & 17 \end{pmatrix}. \qquad det(AA^T - \lambda I) = \lambda^2 - 34\lambda + 225 = (\lambda - 25)(\lambda - 9),$$

$$\sigma_1 = \sqrt{25} = 5 \text{ and } \sigma_2 = \sqrt{9} = 3.$$

$$A^T A - 25I = \begin{pmatrix} -12 & 12 & 2 \\ 12 & -12 & -2 \\ 2 & -2 & -17 \end{pmatrix} \qquad A^T A - 9I = \begin{pmatrix} 4 & 12 & 2 \\ 12 & 4 & -2 \\ 2 & -2 & -1 \end{pmatrix}$$

$$A = U\Sigma V^T = U \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{pmatrix}.$$

$$\sigma u_i = A v_i, \text{ or } u_i = \frac{1}{\sigma} A v_i. \qquad U = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix}.$$

$$A = U\Sigma V^T = \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{pmatrix} \begin{pmatrix} 5 & 0 & 0 \\ 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{18} & -1/\sqrt{18} & 4/\sqrt{18} \\ 2/3 & -2/3 & -1/3 \end{pmatrix}.$$

https://www.d.umn.edu/~mhampton/m4326svd_example.pdf

Application - Topic Modeling (NMF)

- A ~ WH

Term-Tweet Matrix

- Tweet 1
- Tweet 2
- Tweet 3

	Word 1	Word 2	Word n
Tweet 1	1	0	2
Tweet 2	0	1	0
Tweet 3	0	1	1

Specify No Themes (k)

Features Matrix

	Word 1	Word 2	Word n
Theme 1	0.5	0	1
Theme 2	0	0.5	0

Weights Matrix

	Theme 1	Theme 2
Tweet 1	1	0
Tweet 2	0	1
Tweet 3	0	1

https://medium.com/mlreview/topic-modeling-with-scikit-learn-e80d33668730

Thanks

贪心科技 GREEDY TECHNOLOGY

QR decomposition

- Solving $\boldsymbol{Ax} = \boldsymbol{b}$

$Q(Rx) = b \quad Rx = Q^T b = c,$ $\quad Rx = c$ is solved by back substitution

- Comparison between LU

Comparing flop counts for operations on $n \times n$ matrices:

$\frac{2}{3}n^3$	Gaussian elimination
$\frac{4}{3}n^3$	Householder orthogonalization
$2n^3$	Modified Gram-Schmidt
$\frac{8}{3}n^3$	Bidiagonalization
$12n^3$	Singular Value Decomposition

Three reasons to choose orthogonalization to solve square systems:

1. Flop counts exaggerate the Gaussian elimination advantage. When memory traffic and vectorization overheads are considered the **QR** factorization is comparable in efficiency.
2. Orthogonalization methods have guaranteed stability, there is no "growth factor" to worry about as in Gaussian elimination.
3. In cases of ill-conditioning, the orthogonalization methods give an added measure of reliability. **QR** with condition estimate is very dependable and, of course, SVD is unsurpassed when it comes to producing a meaningful solution to a nearly singular system.

https://math.stackexchange.com/questions/2040363/comparing-lu-or-qr-decompositions-for-solving-least-squares

The LU decomposition factorizes a matrix into a lower triangular matrix L and an upper triangular matrix U. The systems $L(Ux) = b$ and $Ux = L^{-1}b$ require fewer additions and multiplications to solve, compared with the original system $Ax = b$, though one might **require significantly more digits** in inexact arithmetic such as floating-point.

Similarly, the QR decomposition expresses A as QR with Q an orthogonal matrix and R an upper triangular matrix. The system $Q(Rx) = b$ is solved by $Rx = Q^T b = c$, and the system $Rx = c$ is solved by 'back substitution'. The number of additions and multiplications required is about twice that of using the LU solver (as you go through gram schmidt process of orthogonalization to make Q), but no more digits are required in inexact arithmetic because the QR decomposition is **numerically stable**.

https://math.stackexchange.com/questions/3744659/what-are-use-cases-of-lu-decomposition-and-qr-factorization