

**Numerical Matrix Decomposition and its Modern
Applications: A Rigorous First Course**

Jun Lu

JUN.LU.LOCKY@GMAIL.COM

Abstract

In 1954, Alston S. Householder published *Principles of Numerical Analysis*, one of the first modern treatments on matrix decomposition that favored a (block) LU decomposition—the factorization of a matrix into the product of lower and upper triangular matrices. And now, matrix decomposition has become a core technology in machine learning, largely due to the development of the back propagation algorithm in fitting a neural network. The sole aim of this survey is to give a self-contained introduction to concepts and mathematical tools in numerical linear algebra and matrix analysis in order to seamlessly introduce matrix decomposition techniques and their applications in subsequent sections. However, we clearly realize our inability to cover all the useful and interesting results concerning matrix decomposition and given the paucity of scope to present this discussion, e.g., the separated analysis of the Euclidean space, Hermitian space, Hilbert space, and things in the complex domain. We refer the reader to literature in the field of linear algebra for a more detailed introduction to the related fields.

This survey is primarily a summary of purpose, significance of important matrix decomposition methods, e.g., LU, QR, and SVD, and the origin and complexity of the methods which shed light on their modern applications. Most importantly, this article presents improved procedures for most of the calculations of the decomposition algorithms which potentially reduce the complexity they induce. Again, this is a decomposition-based context, thus we will introduce the related background when it is needed and necessary. In many other textbooks on linear algebra, the principal ideas are discussed and the matrix decomposition methods serve as “byproduct”. However, we focus on the decomposition methods instead and the principal ideas serve as fundamental tools for them. The mathematical prerequisite is a first course in linear algebra. Other than this modest background, the development is self-contained, with rigorous proof provided throughout.

Keywords: Existence and computing of matrix decompositions, Complexity, Floating point operations (flops), Low-rank approximation, Pivot, LU decomposition for nonzero leading principal minors, Data distillation, CR decomposition, CUR/Skeleton decomposition, Interpolative decomposition, Biconjugate decomposition, Coordinate transformation, Hessenberg decomposition, ULV decomposition, URV decomposition, Rank decomposition, Gram-Schmidt process, Householder reflector, Givens rotation, Rank revealing decomposition, Cholesky decomposition and update/downdate, Eigenvalue problems, Alternating least squares, Randomized algorithm, Tensor decomposition, CP decomposition, Tucker decomposition, High-order SVD.

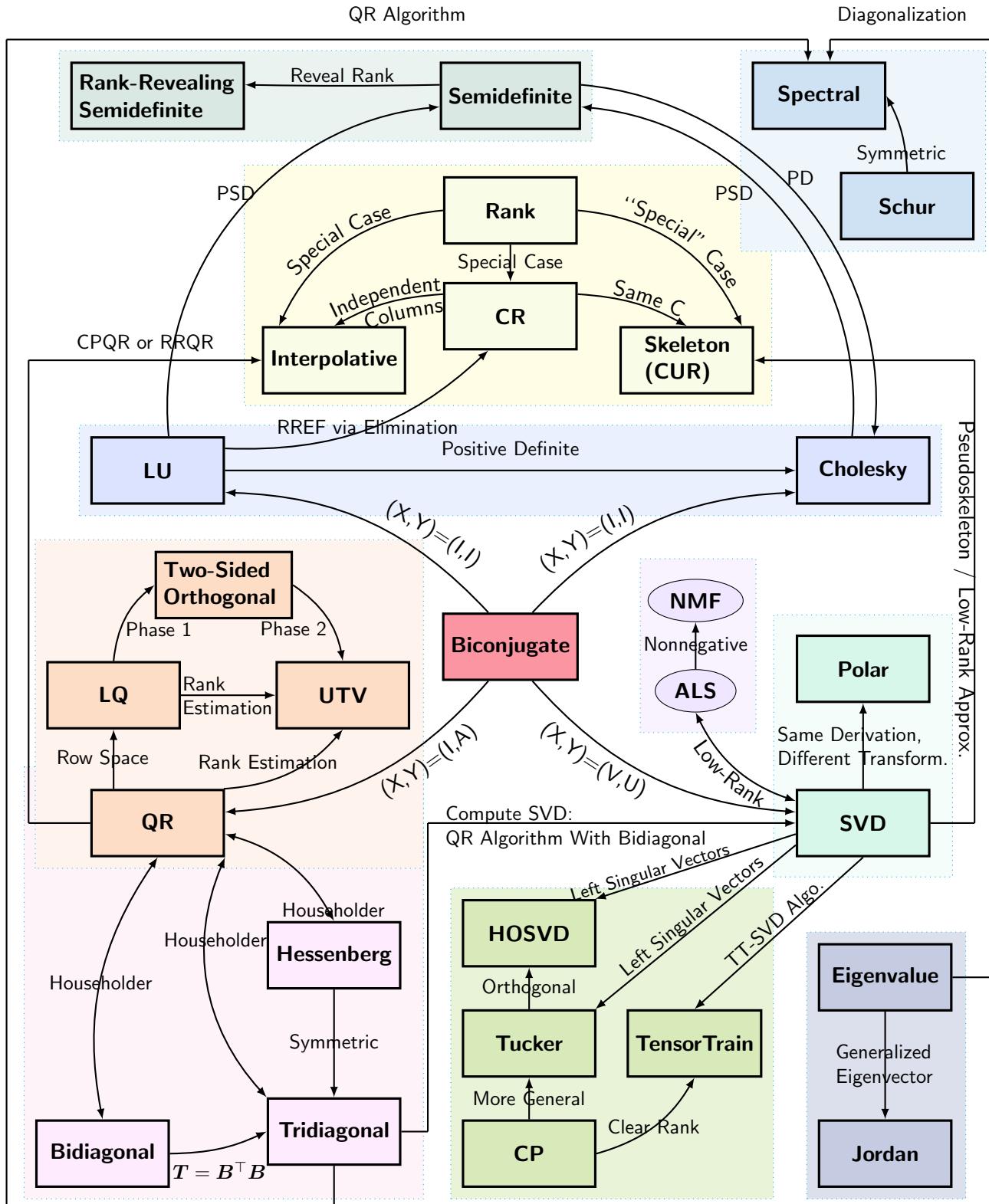


Figure 1: Matrix Decomposition World Map.

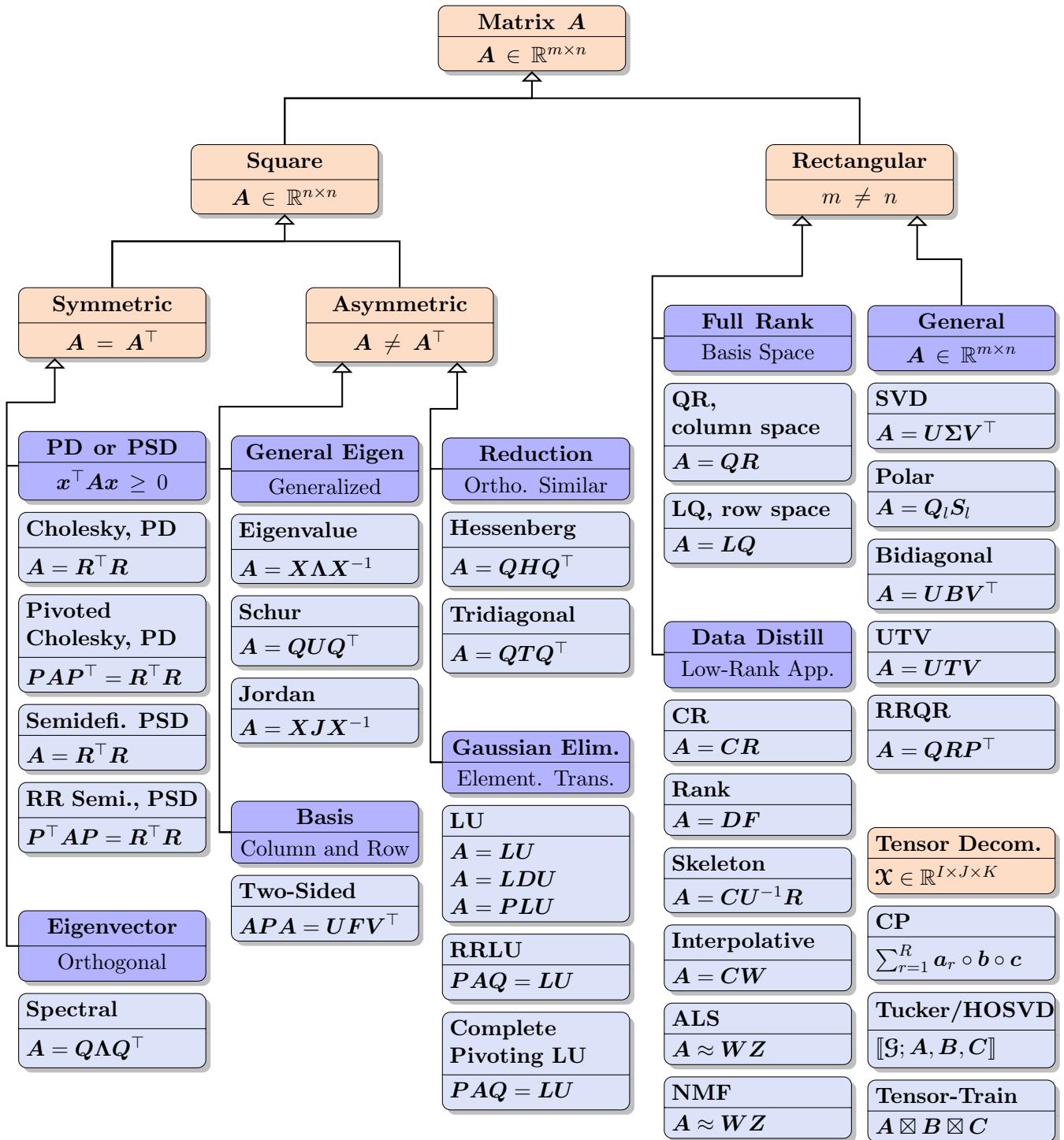


Figure 2: Matrix Decomposition World Map Under Conditions.

Contents

Introduction and Background	13
I Gaussian Elimination	26
1 LU Decomposition	29
1.1 LU Decomposition	30
1.2 Relation to Gaussian Elimination	32
1.3 Existence of the LU Decomposition without Permutation	35
1.4 Existence of the LU Decomposition with Permutation	36
1.5 Computing the LU without Pivoting Recursively: $A=LU$	38
1.5.1 Complexity of Matrix and Vector Operations	40
1.6 Computing the LU without Pivoting Element-Wise: $A=LU$	41
1.6.1 Extension to Thin Matrices	42
1.7 Computing the LU with Pivoting: $A=PLU$	43
1.8 Bandwidth Preserving in the LU Decomposition without Permutation	44
1.9 Block LU Decomposition	45
1.10 Application: Linear System via the LU Decomposition	46
1.11 Application: Computing the Inverse of Nonsingular Matrices	46
1.12 Application: Computing the Determinant via the LU Decomposition	48
1.13 Pivoting	48
1.13.1 Partial Pivoting	48
1.13.2 Complete Pivoting	51
1.13.3 Rook Pivoting	52
1.14 Rank-Revealing LU Decomposition	52
1.15 Rate of Change of L and U^*	52
2 Cholesky Decomposition	55
2.1 Cholesky Decomposition	56
2.2 Existence of the Cholesky Decomposition via Recursive Calculation	56
2.3 Sylvester's Criterion: Leading Principal Minors of PD Matrices	60
2.4 Existence of the Cholesky Decomposition via the LU Decomposition without Permutation	63
2.4.1 Diagonal Values of the Upper Triangular Matrix	63
2.4.2 Block Cholesky Decomposition	64
2.5 Existence of the Cholesky Decomposition via Induction	65

2.6	Uniqueness of the Cholesky Decomposition	66
2.7	Computing the Cholesky Decomposition Recursively	67
2.8	Computing the Cholesky Decomposition Element-Wise	68
2.9	More Properties of Positive Definite Matrices	69
2.10	Last Words on Positive Definite Matrices	70
2.11	Pivoted Cholesky Decomposition	71
2.12	Decomposition for Semidefinite Matrices	71
2.13	Application: Rank-One Update/Downdate	73
2.13.1	Rank-One Update	73
2.13.2	Rank-One Downdate	75
2.14	Application: Indefinite Rank Two Update	76
II	Triangularization, Orthogonalization and Gram-Schmidt Process	77
3	QR Decomposition	80
3.1	QR Decomposition	82
3.2	Project a Vector Onto Another Vector	82
3.3	Project a Vector Onto a Plane	83
3.4	Existence of the QR Decomposition via the Gram-Schmidt Process	83
3.5	Orthogonal vs Orthonormal	87
3.6	Properties of the QR Decomposition	89
3.7	Computing the Reduced QR Decomposition via the Gram-Schmidt Process	90
3.8	Computing the Full QR Decomposition via the Gram-Schmidt Process . .	100
3.9	Dependent Columns	100
3.10	QR with Column Pivoting: Column-Pivoted QR (CPQR)	101
3.10.1	A Simple CPQR via CGS	101
3.10.2	A Practical CPQR via CGS	103
3.10.3	A Practical CPQR via MGS	104
3.10.4	Partial Factorization for CPQR: Extra Bonus of CPQR via MGS .	105
3.11	QR with Column Pivoting: Revealing Rank One Deficiency	106
3.12	QR with Column Pivoting: Revealing Rank r Deficiency*	107
3.13	Existence of the QR Decomposition via the Householder Reflector	108
3.14	Computing the Full QR Decomposition via the Householder Reflector . .	113
3.15	Existence of the QR Decomposition via the Givens Rotation	115
3.16	Computing the Full QR Decomposition via the Givens Rotation	120
3.17	Uniqueness of the QR Decomposition	122
3.18	LQ Decomposition	123
3.19	Two-Sided Orthogonal Decomposition	124
3.20	Applications	125
3.20.1	Application: Least Squares via the Full QR Decomposition	125
3.20.2	Application: Rank-One Changes	127
3.20.3	Application: Appending or Deleting a Column	131
3.20.4	Application: Appending or Deleting a Row	135
3.20.5	Application: Reducing the Ill-Condition via the QR decomposition	137

4 UTV Decomposition: ULV and URV Decomposition	140
4.1 UTV Decomposition	141
4.2 Complete Orthogonal Decomposition	145
4.3 Applications	146
4.3.1 Application: Least Squares via ULV/URV for Rank Deficient Matrices	146
4.3.2 Application: Row Rank equals Column Rank Again via UTV	148
III Data Interpretation and Information Distillation	150
5 CR Decomposition	155
5.1 CR Decomposition	156
5.2 Existence of the CR Decomposition	156
5.3 Reduced Row Echelon Form (RREF)	157
5.4 Computing the CR Decomposition via the Gaussian Elimination	159
5.5 Rank Decomposition	164
5.6 Application: Rank and Trace of an Idempotent Matrix	165
5.7 Other Applications	166
6 Skeleton/CUR Decomposition	167
6.1 Skeleton Decomposition	168
6.2 Existence of the Skeleton Decomposition	168
6.3 Computing the Skeleton Decomposition via the Gram-Schmidt Process . .	171
6.4 Computing the Skeleton Decomposition via Modified Gram-Schmidt Process	172
6.5 Computing the Skeleton Decomposition via the Gaussian Elimination . .	173
6.6 Recover Reduced Row Echelon Form from Skeleton Decomposition	174
6.7 Randomized Algorithms	175
6.8 Pseudoskeleton Decomposition via the SVD	175
7 Interpolative Decomposition (ID)	177
7.1 Interpolative Decomposition	178
7.2 Existence of the Column Interpolative Decomposition	179
7.3 Row ID and Two-Sided ID	183
7.4 Computing the Column ID via the CPQR	185
7.5 Low-Rank Column ID via the RRQR	186
7.6 Computing the ID via Randomized Algorithm	188
IV Reduction to Hessenberg, Tridiagonal, and Bidiagonal Form	192
8 Hessenberg Decomposition	196
8.1 Hessenberg Decomposition	197
8.2 Similarity Transformation and Orthogonal Similarity Transformation . .	198
8.3 Existence of the Hessenberg Decomposition	199
8.4 Computing the Hessenberg Decomposition	202
8.5 Properties of the Hessenberg Decomposition	205

9 Tridiagonal Decomposition: Hessenberg in Symmetric Matrices	208
9.1 Tridiagonal Decomposition	209
9.2 Computing the Tridiagonal Decomposition	209
9.3 Properties of the Tridiagonal Decomposition	212
10 Bidiagonal Decomposition	214
10.1 Bidiagonal Decomposition	215
10.2 Existence of the Bidiagonal Decomposition: Golub-Kahan Bidiagonalization	215
10.3 Computing the Bidiagonal Decomposition: Golub-Kahan Bidiagonalization	220
10.4 Computing the Bidiagonal Decomposition: LHC Bidiagonalization	221
10.5 Computing the Bidiagonal Decomposition: Three-Step Bidiagonalization .	222
10.6 Connection to Tridiagonal Decomposition	224
V Eigenvalue Problem	227
11 Eigenvalue and Jordan Decomposition	229
11.1 Eigenvalue Decomposition	230
11.2 Existence of the Eigenvalue Decomposition	230
11.3 Computing the Eigenvalue Decomposition	232
11.4 Jordan Decomposition	232
11.5 Application: Computing Fibonacci Numbers	234
12 Schur Decomposition	235
12.1 Schur Decomposition	236
12.2 Existence of the Schur Decomposition	236
12.3 Other Forms of the Schur Decomposition	238
13 Spectral Decomposition (Theorem)	240
13.1 Spectral Decomposition	241
13.2 Existence of the Spectral Decomposition	241
13.3 Uniqueness of Spectral Decomposition	247
13.4 Other Forms, Connecting Eigenvalue Decomposition*	247
13.5 Skew-Symmetric Matrices and its Properties*	255
13.6 Applications	258
13.6.1 Application: Eigenvalue of Projection Matrix	258
13.6.2 Application: An Alternative Definition on PD and PSD of Matrices	259
13.6.3 Proof for Semidefinite Rank-Revealing Decomposition	261
13.6.4 Application: Cholesky Decomposition via the QR Decomposition and the Spectral Decomposition	261
13.6.5 Application: Unique Power Decomposition of Positive Definite Ma- trices	262

14 Singular Value Decomposition (SVD)	264
14.1 Singular Value Decomposition	265
14.2 Existence of the SVD	266
14.3 Properties of the SVD	269
14.3.1 Four Subspaces in SVD	269
14.3.2 SVD-Related Orthogonal Projections	270
14.3.3 Relationship between Singular Values and Determinant	271
14.3.4 Orthogonal Equivalence	271
14.3.5 SVD for QR	272
14.3.6 Interlacing Property	272
14.4 Computing the SVD	273
14.4.1 Randomized Method for Computing the SVD Approximately	273
14.5 Polar Decomposition	275
14.6 Generalized Singular Value Decomposition (GSVD)*	276
14.6.1 CS Decomposition	276
14.6.2 Generalized Singular Value Decomposition (GSVD)	277
14.7 Applications	278
14.7.1 Application: Least Squares via SVD for Rank Deficient Matrices	278
14.7.2 Application: Least Squares with Norm Ratio Method	280
14.7.3 Application: Principal Component Analysis (PCA) via the Spectral Decomposition and the SVD	281
14.7.4 Application: Low-Rank Approximation	284
15 Eigenvalue Problem	289
15.1 Background	291
15.2 Rate of Convergence	291
15.3 Eigenvalues as Optimization	293
15.4 Rayleigh Quotient	293
15.5 Power Method, Inverse Power Method, and Rayleigh Quotient Method	295
15.5.1 The Power Method	295
15.5.2 The Inverse Power Method	302
15.5.3 The Shifted Inverse Power Method	303
15.5.4 The Rayleigh Quotient Method	305
15.6 QR Algorithm	305
15.6.1 Preliminary: Power Iteration with Eigenvector Known	305
15.6.2 Preliminary: Power Iteration with Eigenvector Unknown	307
15.6.3 Preliminary: Power Iteration with Eigenvector Unknown and QR Decomposition	308
15.6.4 A Simple QR Algorithm from Power Iteration: without Shifts	309
15.6.5 A Practical QR Algorithm: with Shifts	312
15.7 Apply the Practical QR Algorithm to Tridiagonal Matrices	314
15.7.1 Explicit Shifted QR Algorithm	314
15.7.2 Implicit Shifted QR Algorithm	315
15.8 Jacobi's Method	320
15.8.1 The 2 by 2 Case	320

15.8.2	The Complete Jacobi's Method	322
15.8.3	The Cyclic-by-Row Jacobi's Method	323
15.8.4	Other Issues	324
15.9	Computing the SVD	324
15.9.1	Implicit Shifted QR Algorithm	324
15.9.2	Jacobi's SVD Method	329
15.10	Proof of Results	329
VI	Special Topics	332
16	Coordinate Transformation in Matrix Decomposition	334
16.1	An Overview of Matrix Multiplication	335
16.2	Eigenvalue Decomposition	336
16.3	Spectral Decomposition	336
16.4	SVD	337
16.5	Polar Decomposition	338
17	Alternating Least Squares	339
17.1	Preliminary: Least Squares Approximations	340
17.2	Netflix Recommender and Matrix Factorization	342
17.3	Regularization: Extension to General Matrices	347
17.4	Missing Entries	348
17.5	Vector Inner Product	350
17.6	Gradient Descent	351
17.7	Regularization: A Geometrical Interpretation	354
17.8	Stochastic Gradient Descent	356
17.9	Bias Term	358
17.10	Applications	359
17.10.1	Low-Rank Approximation	359
17.10.2	Movie Recommender	361
18	Nonnegative Matrix Factorization (NMF)	365
18.1	Nonnegative Matrix Factorization	366
18.2	NMF via Multiplicative Update	366
18.3	Regularization	367
18.4	Initialization	369
18.5	Movie Recommender Context	369
19	Biconjugate Decomposition	371
19.1	Existence of the Biconjugate Decomposition	372
19.2	Properties of the Biconjugate Decomposition	376
19.3	Connection to Well-Known Decomposition Methods	377
19.3.1	LDU Decomposition	377
19.3.2	Cholesky Decomposition	379
19.3.3	QR Decomposition	379

19.3.4 SVD	380
20 Modern Applications	382
20.1 Low-Rank Neural Networks	383
20.2 One More Step: Adding a Nonlinear Function Layer	385
VII Tensor Decomposition	388
21 Notations and Background	390
21.1 Matrices to Tensors	391
21.2 Tensor Indexing	391
21.3 Inner Product and Frobenius Norm	393
21.4 Outer Product and Rank-One Tensor	393
21.5 Diagonal and Identity Tensors	393
21.6 Matricization: Matrix Representation of a Higher-Order Tensor	394
21.7 Tensor Multiplication	396
21.8 Special Matrix Products	397
22 CP Decomposition	401
22.1 CP Decomposition	402
22.2 Computing the CP Decomposition	405
23 Tucker Decomposition	407
23.1 Tucker Decomposition	408
23.2 Computing the Tucker Decomposition	410
24 High-Order SVD (HOSVD)	413
24.1 High-Order SVD (HOSVD)	414
24.2 Computing the HOSVD	416
24.3 Properties of the HOSVD	417
24.3.1 Frobenius Norm	417
24.3.2 Low-Rank Approximation	417
25 Tensor-Train (TT) Decomposition	419
25.1 Tensor-Train (TT) Decomposition	420
25.2 Computing the TT Decomposition	421
26 Acknowledgments	424
27 Appendix	425
A Dimension of Column Space and Row Space	426
B The Fundamental Theorem of Linear Algebra	427
B.1 Find the Basis of the Four Subspaces via the CR Decomposition .	428
C The Fundamental Theorem of Linear Algebra: A Least Squares View	430
D Projection and Orthogonal Projection	432
D.1 Properties of Symmetric and Idempotent Matrices	432

D.2	Orthogonal Projection and Geometric Interpretation for LS	434
D.3	Properties of Orthogonal Projection Matrices	439
D.4	Distance Between Subspaces	441
D.5	Projection for LS with Noise Disturbance	443
E	Pseudo-Inverse	445
E.1	One-sided Inverse	445
E.2	Generalized Inverse (g-inverse)	448
E.3	Reflexive Generalized Inverse (rg-inverse)	452
E.4	Pseudo-Inverse	455
E.5	Pseudo-Inverse in SVD	460
E.6	Pseudo-Inverse in CR Decomposition and Skeleton Decomposition	462
F	Schur Complement	464
G	General Term Formula of Wedderburn Sequence	466
H	Decoding Orthogonal Matrix Multiplication	468
I	Cochran's Theorem	469
J	Taylor's Expansion	472
K	Famous Inequalities	473
L	Matrix Norm	476
L.1	Vector Norm	476
L.2	Matrix Norm	480

Introduction and Background

Matrix decomposition has become a core technology in statistics (Banerjee and Roy, 2014; Gentle, 1998), optimization (Gill et al., 2021), recommender system (Symeonidis and Zioupos, 2016), and machine learning (Goodfellow et al., 2016; Bishop, 2006), largely due to the development of back propagation algorithm in fitting a neural network. The sole aim of this survey is to give a self-contained introduction to concepts and mathematical tools in numerical linear algebra and matrix analysis in order to seamlessly introduce matrix decomposition techniques and their applications in subsequent sections. However, we clearly realize our inability to cover all the useful and interesting results concerning matrix decomposition and given the paucity of scope to present this discussion, e.g., the separated analysis of the Euclidean space, Hermitian space, and Hilbert space. We refer the reader to literature in the field of linear algebra for a more detailed introduction to the related fields. Some excellent examples include (Householder, 2006; Trefethen and Bau III, 1997; Strang, 2009; Stewart, 2000; Gentle, 2007; Higham, 2002; Quarteroni et al., 2010; Golub and Van Loan, 2013; Beck, 2017; Gallier and Quaintance, 2017; Boyd and Vandenberghe, 2018; Strang, 2019; van de Geijn and Myers, 2020; Strang, 2021). Moreover, this survey will cover the calculation and complexity of the decompositional methods with details on the cost reduction. For a compact text of only rigorous proof, one can refer to (Lu, 2022).

A matrix decomposition is a way of reducing a complex matrix into its constituent parts which are in simpler forms. The underlying principle of the decompositional approach to matrix computation is that it is not the business of the matrix algorithmists to solve particular problems, but it is an approach that can simplify more complex matrix operations which can be performed on the decomposed parts rather than on the original matrix itself. At a general level, a matrix decomposition task on matrix \mathbf{A} can be cast as

- $\mathbf{A} = \mathbf{Q}\mathbf{U}$: where \mathbf{Q} is an orthogonal matrix that contains the same column space as \mathbf{A} and \mathbf{U} is a relatively simple and sparse matrix to reconstruct \mathbf{A} .
- $\mathbf{A} = \mathbf{QTQ}^\top$: where \mathbf{Q} is orthogonal such that \mathbf{A} and \mathbf{T} are *similar matrices* that share the same properties such as same eigenvalues, sparsity. Moreover, working on \mathbf{T} is an easier task compared to that of \mathbf{A} .
- $\mathbf{A} = \mathbf{UTV}$: where \mathbf{U}, \mathbf{V} are orthogonal matrices such that the columns of \mathbf{U} and the rows of \mathbf{V} constitute an orthonormal basis of the column space and row space of \mathbf{A} respectively.
- $\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{B}} \underset{r \times n}{\mathbf{C}}$: where \mathbf{B}, \mathbf{C} are full rank matrices that can reduce the memory storage of \mathbf{A} . In practice, a low-rank approximation $\underset{m \times n}{\mathbf{A}} \approx \underset{m \times k}{\mathbf{D}} \underset{k \times n}{\mathbf{F}}$ can be employed where $k < r$ is called the *numerical rank* of the matrix such that the matrix can be stored much more inexpensively and can be multiplied rapidly with vectors or other matrices. An approximation of the form $\mathbf{A} = \mathbf{DF}$ is useful for storing the matrix \mathbf{A} more frugally (we can store \mathbf{D} and \mathbf{F} using $k(m + n)$ floats, as opposed to mn numbers for storing \mathbf{A}), for efficiently computing a matrix-vector product $\mathbf{b} = \mathbf{Ax}$ (via $\mathbf{c} = \mathbf{Fx}$ and $\mathbf{b} = \mathbf{Dc}$), for data interpretation, and much more.
- A matrix decomposition, which though is usually expensive to compute, can be reused to solve new problems involving the original matrix in different scenarios, e.g., as long

as the factorization of \mathbf{A} is obtained, it can be reused to solve the set of linear systems $\{\mathbf{b}_1 = \mathbf{Ax}_1, \mathbf{b}_2 = \mathbf{Ax}_2, \dots, \mathbf{b}_k = \mathbf{Ax}_k\}$.

- More generally, a matrix decomposition can help to understand the internal meaning of what happens when multiplied by the matrix such that each constituent has a geometrical transformation (see Section 16, p. 334).

The matrix decomposition algorithms can fall into many categories. Nonetheless, six categories hold the center and we sketch it here:

1. Factorizations arise from Gaussian elimination including the LU decomposition and its positive definite alternative - Cholesky decomposition;
2. Factorizations obtained when orthogonalizing the columns or the rows of a matrix such that the data can be explained well in an orthonormal basis;
3. Factorizations where the matrices are skeletonized such that a subset of the columns or the rows can represent the whole data in a small reconstruction error, whilst, the sparsity and nonnegativity of the matrices are kept as they are;
4. Reduction to Hessenberg, tridiagonal, or bidiagonal form, as a result, the properties of the matrices can be explored in these reduced matrices such as rank, eigenvalues, and so on;
5. Factorizations result from the computation of the eigenvalues of matrices;
6. In particular, the rest can be cast as a special kind of decompositions that involve optimization methods, high-level ideas where the category may not be straightforward to determine.

The world pictures for decomposition in Figure 1 and 2 connect each decomposition method by their internal relations and also separate different methods by the criteria or prerequisites of them. Readers will get more information about the two pictures after reading the text.

Notation and preliminaries In the rest of this section we will introduce and recap some basic knowledge about linear algebra. For the rest of the important concepts, we define and discuss them as per need for clarity. The readers with enough background in matrix analysis can skip this section. In the text, we simplify matters by considering only matrices that are real. Without special consideration, the eigenvalues of the discussed matrices are also real. We also assume throughout that $\|\cdot\| = \|\cdot\|_2$.

In all cases, scalars will be denoted in a non-bold font possibly with subscripts (e.g., a, α, α_i). We will use **boldface** lower case letters possibly with subscripts to denote vectors (e.g., $\mu, \mathbf{x}, \mathbf{x}_n, \mathbf{z}$) and **boldface** upper case letters possibly with subscripts to denote matrices (e.g., \mathbf{A}, \mathbf{L}_j). The i -th element of a vector \mathbf{z} will be denoted by \mathbf{z}_i in bold font (or z_i in the non-bold font).

The n -th element in a sequence is denoted by a superscript in parentheses, e.g., $\mathbf{A}^{(n)}$ denotes the n -th matrix in a sequence, $\mathbf{a}^{(k)}$ denote the k -th vector in a sequence.

Subarrays are formed when a subset of the indices is fixed. *The i -th row and j -th column value of matrix \mathbf{A} (entry (i, j) of \mathbf{A}) will be denoted by \mathbf{A}_{ij} if block submatrices are involved, or by a_{ij} alternatively if block submatrices are not involved.* Furthermore, it will be helpful to utilize the **Matlab-style notation**, the i -th row to the j -th row and the k -th column to the m -th column submatrix of the matrix \mathbf{A} will be denoted by $\mathbf{A}_{i:j,k:m}$. A colon is used

to indicate all elements of a dimension, e.g., $\mathbf{A}_{:,k:m}$ denotes the k -th column to the m -th column of the matrix \mathbf{A} , and $\mathbf{A}_{:,k}$ denote the k -th column of \mathbf{A} . Alternatively, the k -th column of \mathbf{A} may be denoted more compactly by \mathbf{a}_k .

When the index is not continuous, given ordered subindex sets I and J , $\mathbf{A}[I, J]$ denotes the submatrix of \mathbf{A} obtained by extracting the rows and columns of \mathbf{A} indexed by I and J , respectively; and $\mathbf{A}[:, J]$ denotes the submatrix of \mathbf{A} obtained by extracting the columns of \mathbf{A} indexed by J .

Definition 0.1: Matlab Notation

Suppose $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $I = [i_1, i_2, \dots, i_k]$ and $J = [j_1, j_2, \dots, j_l]$ are two index vectors, then $\mathbf{A}[I, J]$ denotes the $k \times l$ submatrix

$$\mathbf{A}[I, J] = \begin{bmatrix} \mathbf{A}_{i_1, j_1} & \mathbf{A}_{i_1, j_2} & \dots & \mathbf{A}_{i_1, j_l} \\ \mathbf{A}_{i_2, j_1} & \mathbf{A}_{i_2, j_2} & \dots & \mathbf{A}_{i_2, j_l} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{i_k, j_1} & \mathbf{A}_{i_k, j_2} & \dots & \mathbf{A}_{i_k, j_l} \end{bmatrix}.$$

Whilst, $\mathbf{A}[I, :]$ denotes the $k \times n$, and $\mathbf{A}[:, J]$ denotes the $m \times l$ analogously.

We note that it does not matter whether the index vectors I, J are row vectors or column vectors. It matters which axis they index (rows of \mathbf{A} or columns of \mathbf{A}). We should also notice that range of the index:

$$\begin{cases} 0 \leq \min(I) \leq \max(I) \leq m; \\ 0 \leq \min(J) \leq \max(J) \leq n. \end{cases}$$

And in all cases, vectors are formulated in a column rather than in a row. A row vector will be denoted by a transpose of a column vector such as \mathbf{a}^\top . A specific column vector with values is split by the symbol “;”, e.g., $\mathbf{x} = [1; 2; 3]$ is a column vector in \mathbb{R}^3 . Similarly, a specific row vector with values is split by the symbol “,”, e.g., $\mathbf{y} = [1, 2, 3]$ is a row vector with 3 values. Further, a column vector can be denoted by the transpose of a row vector e.g., $\mathbf{y} = [1, 2, 3]^\top$ is a column vector.

The transpose of a matrix \mathbf{A} will be denoted by \mathbf{A}^\top and its inverse will be denoted by \mathbf{A}^{-1} . We will denote the $p \times p$ identity matrix by \mathbf{I}_p . A vector or matrix of all zeros will be denoted by a **boldface** zero $\mathbf{0}$ whose size should be clear from context, or we denote $\mathbf{0}_p$ to be the vector of all zeros with p entries.

Definition 0.2: Eigenvalue

Given any vector space E and any linear map $A : E \rightarrow E$, a scalar $\lambda \in K$ is called an eigenvalue, or proper value, or characteristic value of \mathbf{A} if there is some nonzero vector $\mathbf{u} \in E$ such that

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u}.$$

Definition 0.3: Spectrum and Spectral Radius

The set of all eigenvalues of \mathbf{A} is called the spectrum of \mathbf{A} and denoted by $\Lambda(\mathbf{A})$. The largest magnitude of the eigenvalues is known as the spectral radius $\rho(\mathbf{A})$:

$$\rho(\mathbf{A}) = \max_{\lambda \in \Lambda(\mathbf{A})} |\lambda|.$$

Definition 0.4: Eigenvector

A vector $\mathbf{u} \in E$ is called an eigenvector, or proper vector, or characteristic vector of \mathbf{A} if $\mathbf{u} \neq 0$ and if there is some $\lambda \in K$ such that

$$\mathbf{A}\mathbf{u} = \lambda\mathbf{u},$$

where the scalar λ is then an eigenvalue. And we say that \mathbf{u} is an eigenvector associated with λ .

Moreover, the tuple (λ, \mathbf{u}) above is said to be an **eigenpair**. Intuitively, the above definitions mean that multiplying matrix \mathbf{A} by the vector \mathbf{u} results in a new vector that is in the same direction as \mathbf{u} , but only scaled by a factor λ . For any eigenvector \mathbf{u} , we can scale it by a scalar s such that $s\mathbf{u}$ is still an eigenvector of \mathbf{A} . That's why we call the eigenvector as an eigenvector of \mathbf{A} associated with eigenvalue λ . To avoid ambiguity, we usually assume that the eigenvector is normalized to have length 1 and the first entry is positive (or negative) since both \mathbf{u} and $-\mathbf{u}$ are eigenvectors.

In this context, we will highly use the idea about the linear independence of a set of vectors. Two equivalent definitions are given as follows.

Definition 0.5: Linearly Independent

A set of vectors $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ is called linearly independent if there is no combination can get $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \dots + x_m\mathbf{a}_m = 0$ except all x_i 's are zero. An equivalent definition is that $\mathbf{a}_1 \neq \mathbf{0}$, and for every $k > 1$, the vector \mathbf{a}_k does not belong to the span of $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{k-1}\}$.

In the study of linear algebra, every vector space has a basis and every vector is a linear combination of members of the basis. We then define the span and dimension of a subspace via the basis.

Definition 0.6: Span

If every vector \mathbf{v} in subspace \mathcal{V} can be expressed as a linear combination of $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$, then $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ is said to span \mathcal{V} .

Definition 0.7: Subspace

A nonempty subset \mathcal{V} of \mathbb{R}^n is called a subspace if $x\mathbf{a} + y\mathbf{a} \in \mathcal{V}$ for every $\mathbf{a}, \mathbf{b} \in \mathcal{V}$ and every $x, y \in \mathbb{R}$.

Definition 0.8: Basis and Dimension

A set of vectors $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ is called a basis of \mathcal{V} if they are linearly independent and span \mathcal{V} . Every basis of a given subspace has the same number of vectors, and the number of vectors in any basis is called the dimension of the subspace \mathcal{V} . By convention, the subspace $\{\mathbf{0}\}$ is said to have dimension zero. Furthermore, every subspace of nonzero dimension has a basis that is orthogonal, i.e., the basis of a subspace can be chosen orthogonal.

Definition 0.9: Column Space (Range)

If \mathbf{A} is an $m \times n$ real matrix, we define the column space (or range) of \mathbf{A} to be the set spanned by its columns:

$$\mathcal{C}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{R}^m : \exists \mathbf{x} \in \mathbb{R}^n, \mathbf{y} = \mathbf{A}\mathbf{x}\}.$$

And the row space of \mathbf{A} is the set spanned by its rows, which is equal to the column space of \mathbf{A}^\top :

$$\mathcal{C}(\mathbf{A}^\top) = \{\mathbf{x} \in \mathbb{R}^n : \exists \mathbf{y} \in \mathbb{R}^m, \mathbf{x} = \mathbf{A}^\top \mathbf{y}\}.$$

Definition 0.10: Null Space (Nullspace, Kernel)

If \mathbf{A} is an $m \times n$ real matrix, we define the null space (or kernel, or nullspace) of \mathbf{A} to be the set:

$$\mathcal{N}(\mathbf{A}) = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{A}\mathbf{y} = \mathbf{0}\}.$$

And the null space of \mathbf{A}^\top is defined as

$$\mathcal{N}(\mathbf{A}^\top) = \{\mathbf{x} \in \mathbb{R}^m : \mathbf{A}^\top \mathbf{x} = \mathbf{0}\}.$$

Both the column space of \mathbf{A} and the null space of \mathbf{A}^\top are subspaces of \mathbb{R}^n . In fact, every vector in $\mathcal{N}(\mathbf{A}^\top)$ is perpendicular to $\mathcal{C}(\mathbf{A})$ and vice versa.¹

Definition 0.11: Rank

¹. Every vector in $\mathcal{N}(\mathbf{A})$ is also perpendicular to $\mathcal{C}(\mathbf{A}^\top)$ and vice versa.

The *rank* of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is the dimension of the column space of \mathbf{A} . That is, the rank of \mathbf{A} is equal to the maximal number of linearly independent columns of \mathbf{A} , and is also the maximal number of linearly independent rows of \mathbf{A} . The matrix \mathbf{A} and its transpose \mathbf{A}^\top have the same rank. We say that \mathbf{A} has full rank, if its rank is equal to $\min\{m, n\}$. In another word, this is true if and only if either all the columns of \mathbf{A} are linearly independent, or all the rows of \mathbf{A} are linearly independent. Specifically, given a vector $\mathbf{u} \in \mathbb{R}^m$ and a vector $\mathbf{v} \in \mathbb{R}^n$, then the $m \times n$ matrix $\mathbf{u}\mathbf{v}^\top$ obtained by the outer product of vectors is of rank 1. In short, the rank of a matrix is equal to:

- number of linearly independent columns;
- number of linearly independent rows;
- and remarkably, these are always the same (see Appendix A, p. 426).

Definition 0.12: Orthogonal Complement in General

The orthogonal complement \mathcal{V}^\perp of a subspace \mathcal{V} contains every vector that is perpendicular to \mathcal{V} . That is,

$$\mathcal{V}^\perp = \{\mathbf{v} | \mathbf{v}^\top \mathbf{u} = 0, \forall \mathbf{u} \in \mathcal{V}\}.$$

The two subspaces are disjoint that span the entire space. The dimensions of \mathcal{V} and \mathcal{V}^\perp add to the dimension of the whole space. Furthermore, $(\mathcal{V}^\perp)^\perp = \mathcal{V}$.

Definition 0.13: Orthogonal Complement of Column Space

If \mathbf{A} is an $m \times n$ real matrix, the orthogonal complement of $\mathcal{C}(\mathbf{A})$, $\mathcal{C}^\perp(\mathbf{A})$ is the subspace defined as:

$$\begin{aligned} \mathcal{C}^\perp(\mathbf{A}) &= \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y}^\top \mathbf{A} \mathbf{x} = 0, \forall \mathbf{x} \in \mathbb{R}^n\} \\ &= \{\mathbf{y} \in \mathbb{R}^m : \mathbf{y}^\top \mathbf{v} = 0, \forall \mathbf{v} \in \mathcal{C}(\mathbf{A})\}. \end{aligned}$$

Then we have the four fundamental spaces for any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with rank r :

- $\mathcal{C}(\mathbf{A})$: Column space of \mathbf{A} , i.e., linear combinations of columns with dimension r ;
- $\mathcal{N}(\mathbf{A})$: Null space of \mathbf{A} , i.e., all \mathbf{x} with $\mathbf{A}\mathbf{x} = 0$ with dimension $n - r$;
- $\mathcal{C}(\mathbf{A}^\top)$: Row space of \mathbf{A} , i.e., linear combinations of rows with dimension r ;
- $\mathcal{N}(\mathbf{A}^\top)$: Left null space of \mathbf{A} , i.e., all \mathbf{y} with $\mathbf{A}^\top \mathbf{y} = 0$ with dimension $m - r$,

where r is the rank of the matrix. Furthermore, $\mathcal{N}(\mathbf{A})$ is the orthogonal complement to $\mathcal{C}(\mathbf{A}^\top)$, and $\mathcal{C}(\mathbf{A})$ is the orthogonal complement to $\mathcal{N}(\mathbf{A}^\top)$. The proof can be found in Appendix B.

Definition 0.14: Orthogonal Matrix

A real square matrix \mathbf{Q} is an orthogonal matrix if the inverse of \mathbf{Q} equals the transpose of \mathbf{Q} , that is $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ and $\mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$. In another word, suppose $\mathbf{Q} =$

$[\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$ where $\mathbf{q}_i \in \mathbb{R}^n$ for all $i \in \{1, 2, \dots, n\}$, then $\mathbf{q}_i^\top \mathbf{q}_j = \delta(i, j)$ with $\delta(i, j)$ being the Kronecker delta function. If \mathbf{Q} contains only γ of these columns with $\gamma < n$, then $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_\gamma$ still holds with \mathbf{I}_γ being the $\gamma \times \gamma$ identity matrix. But $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ will not be true. For any vector \mathbf{x} , the orthogonal matrix will preserve the length: $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$.

Definition 0.15: Permutation Matrix

A permutation matrix \mathbf{P} is a square binary matrix that has exactly one entry of 1 in each row and each column and 0's elsewhere.

Row Point That is, the permutation matrix \mathbf{P} has the rows of the identity \mathbf{I} in any order and the order decides the sequence of the row permutation. Suppose we want to permute the rows of matrix \mathbf{A} , we just multiply on the left by \mathbf{PA} .

Column Point Or, equivalently, the permutation matrix \mathbf{P} has the columns of the identity \mathbf{I} in any order and the order decides the sequence of the column permutation. And now, the column permutation of \mathbf{A} is to multiply on the right by \mathbf{AP} .

The permutation matrix \mathbf{P} can be more efficiently represented via a vector $J \in \mathbb{Z}_+^n$ of indices such that $\mathbf{P} = \mathbf{I}[:, J]$ where \mathbf{I} is the $n \times n$ identity matrix and notably, the elements in vector J sum to $1 + 2 + \dots + n = \frac{n^2+n}{2}$.

Example 0.1 (Permutation) Suppose,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} & 1 \\ & 1 \\ 1 & \end{bmatrix}.$$

The row permutation is given by

$$\mathbf{PA} = \begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \end{bmatrix},$$

where the order of the rows of \mathbf{A} appearing in \mathbf{PA} matches the order of the rows of \mathbf{I} in \mathbf{P} . And the column permutation is given by

$$\mathbf{AP} = \begin{bmatrix} 3 & 1 & 2 \\ 6 & 4 & 5 \\ 9 & 7 & 8 \end{bmatrix},$$

where the order of the columns of \mathbf{A} appearing in \mathbf{AP} matches the order of the columns of \mathbf{I} in \mathbf{P} . \square

Definition 0.16: Selection Matrix

A selection matrix \mathbf{S} is a square diagonal matrix with diagonals being 1 or 0. The 1 entries are the rows or columns that will be selected.

Row Point That is, the selection matrix \mathbf{S} has the rows of the identity \mathbf{I} if we want to select the corresponding rows, and otherwise, we mask the rows in the identity \mathbf{I} by zero. Suppose we want to select the rows of matrix \mathbf{A} , we just multiply from left by \mathbf{SA} .

Column Point Or, equivalent, the selection matrix \mathbf{S} has the columns of the identity \mathbf{I} if we want to select the corresponding columns, or otherwise, we mask the columns in the identity \mathbf{I} by zero. And now, the column selection of \mathbf{A} is to multiply from right by \mathbf{AS} .

Example 0.2 (Selection and Permutation) Suppose,

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}, \quad \text{and} \quad \mathbf{S} = \begin{bmatrix} 1 & & \\ & 0 & \\ & & 1 \end{bmatrix}.$$

The row selection is given by

$$\mathbf{SA} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 7 & 8 & 9 \end{bmatrix},$$

where the row of \mathbf{A} appearing in \mathbf{SA} matches the row entries of \mathbf{S} . And the column selection is given by

$$\mathbf{AS} = \begin{bmatrix} 1 & 0 & 3 \\ 4 & 0 & 6 \\ 7 & 0 & 9 \end{bmatrix},$$

where the columns of \mathbf{A} appearing in \mathbf{AS} matches column entries of \mathbf{S} . If now, we want to reorder the selected rows or columns in the upper left of the final matrix, we can construct a permutation as follows

$$\mathbf{P} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix},$$

such that

$$\mathbf{PSA} = \begin{bmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 0 & 0 & 0 \end{bmatrix},$$

and

$$\mathbf{ASP} = \begin{bmatrix} 1 & 3 & 0 \\ 4 & 6 & 0 \\ 7 & 9 & 0 \end{bmatrix}.$$

The trick is essential to some mathematical proofs, e.g., the properties of positive definite matrices in Lemma 2.1. \square

From an introductory course on linear algebra, we have the following remark on the equivalent claims on nonsingular matrices.

Remark 0.17: List of Equivalence of Nonsingularity for a Matrix

For a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the following claims are equivalent:

- \mathbf{A} is nonsingular;
- \mathbf{A} is invertible, i.e., \mathbf{A}^{-1} exists;
- $\mathbf{A}\mathbf{x} = \mathbf{b}$ has a unique solution $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$;
- $\mathbf{A}\mathbf{x} = \mathbf{0}$ has a unique, trivial solution: $\mathbf{x} = \mathbf{0}$;
- Columns of \mathbf{A} are linearly independent;
- Rows of \mathbf{A} are linearly independent;
- $\det(\mathbf{A}) \neq 0$;
- $\dim(\mathcal{N}(\mathbf{A})) = 0$;
- $\mathcal{N}(\mathbf{A}) = \{\mathbf{0}\}$, i.e., the null space is trivial;
- $\mathcal{C}(\mathbf{A}) = \mathcal{C}(\mathbf{A}^\top) = \mathbb{R}^n$, i.e., the column space or row space span the whole \mathbb{R}^n ;
- \mathbf{A} has full rank $r = n$;
- The reduced row echelon form is $\mathbf{R} = \mathbf{I}$;
- $\mathbf{A}^\top \mathbf{A}$ is symmetric positive definite;
- \mathbf{A} has n nonzero (positive) singular values;
- All eigenvalues are nonzero;

It will be shown important to take the above equivalence into mind, otherwise, we will easily get lost. On the other hand, the following remark also shows the equivalent claims for singular matrices.

Remark 0.18: List of Equivalence of Singularity for a Matrix

For a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with eigenpair (λ, \mathbf{u}) , the following claims are equivalent:

- $(\mathbf{A} - \lambda\mathbf{I})$ is singular;
- $(\mathbf{A} - \lambda\mathbf{I})$ is not invertible;
- $(\mathbf{A} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0}$ has nonzero $\mathbf{x} \neq \mathbf{0}$ solutions, and $\mathbf{x} = \mathbf{u}$ is one of such solutions;
- $(\mathbf{A} - \lambda\mathbf{I})$ has linearly dependent columns;
- $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$;
- $\dim(\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})) > 0$;
- Null space of $(\mathbf{A} - \lambda\mathbf{I})$ is nontrivial;
- Columns of \mathbf{A} are linearly dependent;
- Rows of \mathbf{A} are linearly dependent;
- \mathbf{A} has rank $r < n$;
- Dimension of column space = dimension of row space = $r < n$;
- $\mathbf{A}^\top \mathbf{A}$ is symmetric semidefinite;

- \mathbf{A} has $r < n$ nonzero (positive) singular values;
- Zero is an eigenvalue of \mathbf{A}

Matrix Decomposition in a Nutshell

We briefly overview the different decompositional methods that we will cover in this text as follows:

$$1. \text{ } LU. \mathbf{A} = \mathbf{L}\mathbf{U} = \begin{pmatrix} \text{lower triangular } \mathbf{L} \\ 1\text{'s on the diagonal} \end{pmatrix} \begin{pmatrix} \text{upper triangular } \mathbf{U} \\ \text{pivots on the diagonal} \end{pmatrix}$$

Requirements: \mathbf{A} has nonzero leading principal minors, i.e., no row permutations involve to reduce \mathbf{A} to upper triangular via the Gaussian elimination (Theorem 1.5, p. 31).

$$2. \text{ } LU. \mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{U} = \begin{pmatrix} \text{lower triangular } \mathbf{L} \\ 1\text{'s on the diagonal} \end{pmatrix} \begin{pmatrix} \text{pivot matrix } \mathbf{D} \\ \mathbf{D} \text{ is diagonal} \end{pmatrix} \begin{pmatrix} \text{upper triangular } \mathbf{U} \\ 1\text{'s on the diagonal} \end{pmatrix}$$

Requirements: \mathbf{A} has nonzero leading principal minors, i.e., no row permutations involves to reduce \mathbf{A} to upper triangular via the Gaussian elimination. The \mathbf{D} contains the pivots such that \mathbf{U} has 1's on the diagonal. When \mathbf{A} is symmetric, it follows that $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^\top$ (Corollary 1.2, p. 36).

$$3. \text{ } PLU. \mathbf{A} = \mathbf{P}\mathbf{L}\mathbf{U} = \begin{pmatrix} \text{permutation} \\ \text{matrix } \mathbf{P} \text{ avoids} \\ \text{zeros for} \\ \text{eliminations} \end{pmatrix} \begin{pmatrix} \text{lower triangular } \mathbf{L} \\ 1\text{'s on the diagonal} \end{pmatrix} \begin{pmatrix} \text{upper triangular } \mathbf{U} \\ \text{pivots on the diagonal} \end{pmatrix}$$

Requirements: \mathbf{A} is nonsingular, then $\mathbf{P}, \mathbf{L}, \mathbf{U}$ are nonsingular as well. \mathbf{P}^\top does the row permutation in advance such that $\mathbf{P}^\top \mathbf{A}$ has nonzero leading principle minors (Theorem 1.1, p. 30).

$$4. \text{ } RRLU. \mathbf{P}\mathbf{A}\mathbf{Q} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21}^\top & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Requirements: Any matrix with rank r such that \mathbf{P}, \mathbf{Q} are permutations, \mathbf{L}_{11} is lower triangular with 1's on the diagonal, \mathbf{U}_{11} is upper triangular with pivots on the diagonal where both of them reveal the rank of matrix \mathbf{A} (Section 1.14, p. 52).

$$5. \text{ } \text{Complete Pivoting LU}. \mathbf{P}\mathbf{A}\mathbf{Q} = \mathbf{L}\mathbf{U} = \begin{pmatrix} \text{lower triangular } \mathbf{L} \\ 1\text{'s on the diagonal} \end{pmatrix} \begin{pmatrix} \text{upper triangular } \mathbf{U} \\ \text{pivots on the diagonal} \end{pmatrix}$$

Requirements: \mathbf{A} is nonsingular such that \mathbf{P}, \mathbf{Q} are permutations, \mathbf{L} is unit lower triangular with 1's on the diagonal, \mathbf{U} is upper triangular with pivots on the diagonal (Section 1.13.2, p. 51).

$$6. \text{ } \text{Cholesky}. \mathbf{A} = \mathbf{R}^\top \mathbf{R} = (\text{lower triangular})(\text{upper triangular})$$

Requirements: \mathbf{A} is positive definite (symmetric) such that the diagonal of \mathbf{R} contains the diagonals of $\sqrt{\mathbf{D}}$ which are positive from the LU decomposition. Trivially, \mathbf{R}^\top can be set as $\mathbf{L}\sqrt{\mathbf{D}}$ from the LU decomposition. When \mathbf{A} is positive semidefinite, \mathbf{R} is upper triangular with possible zeros on the diagonal (Theorem 2.1, p. 56; Theorem 2.1, p. 72).

$$7. \text{ } \text{Pivoted Cholesky}. \mathbf{P}\mathbf{A}\mathbf{P}^\top = \mathbf{R}^\top \mathbf{R} = (\text{lower triangular})(\text{upper triangular})$$

Requirements: \mathbf{A} is positive definite (symmetric) such that \mathbf{R} is upper triangular (Section 2.11, p. 71).

$$8. \text{ } \text{Semidefinite Rank-Revealing}. \mathbf{P}^\top \mathbf{A}\mathbf{P} = \mathbf{R}^\top \mathbf{R}, \quad \text{with} \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

- Requirements:** \mathbf{A} is positive semidefinite with rank r such that after permuting by \mathbf{P} , \mathbf{R}_{11} is upper triangular with rank r (Theorem 2.2, p. 73).
9. CR . $\mathbf{A} = \mathbf{C}\mathbf{R}$ =(first independent columns of \mathbf{A})(basis for row space of \mathbf{A})
Requirements: Any matrix \mathbf{A} with rank r , \mathbf{C} contain first r linearly independent columns of \mathbf{A} , \mathbf{R} is the reduced row echelon form removing zero rows. \mathbf{R} contains an $r \times r$ identity submatrix (Theorem 5.1, p. 156).
10. *Rank.* $\mathbf{A} = \mathbf{D}\mathbf{F}$ =(column basis of \mathbf{A})(row basis of \mathbf{A})
Requirements: Any matrix \mathbf{A} with rank r , \mathbf{D} contains r columns that span the column space of \mathbf{A} , \mathbf{F} contains r rows that span the row space of \mathbf{A} (Theorem 5.1, p. 164).
11. *Skeleton.* $\mathbf{A} = \mathbf{C}\mathbf{U}^{-1}\mathbf{R}$ =(r columns of \mathbf{A})(intersection of \mathbf{C}, \mathbf{R}) $^{-1}$ (r rows of \mathbf{A})
Requirements: Any matrix \mathbf{A} with rank r such that \mathbf{C}, \mathbf{R} come directly from columns and rows of \mathbf{A} . \mathbf{U} is the mixing matrix that on the intersection of \mathbf{C}, \mathbf{R} (Theorem 6.1, p. 168).
12. *Interpolative.* $\mathbf{A} = \mathbf{C}\mathbf{W}$ =(independent columns of \mathbf{A})(columns consisting of identity matrix)
Requirements: Any matrix \mathbf{A} with rank r such that \mathbf{W} contains an $r \times r$ identity submatrix (in the sense of permutation). The elements of \mathbf{W} are no greater than 1 in absolute value (Theorem 7.1, p. 178; Theorem 7.1, p. 184).
13. *QR.* $\mathbf{A} = \mathbf{Q}\mathbf{R}$ =(orthonormal columns in \mathbf{Q})(upper triangular \mathbf{R})
Requirements: Rectangular matrix \mathbf{A} has linearly independent columns otherwise \mathbf{R} will be singular (diagonals contain at least one zero). In the full QR decomposition, \mathbf{Q} is orthogonal such that $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ (Theorem 3.1, p. 82).
14. *CPQR.* $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P}^\top$ = (orthogonal \mathbf{Q}) $\begin{pmatrix} \text{upper triangular} \\ \mathbf{R}_{11} \mid \text{full matrix} \\ \mathbf{R}_{12} \end{pmatrix}$ $\begin{pmatrix} \text{permutation} \\ \mathbf{P}^\top \end{pmatrix}$
Requirements: Any rectangular matrix \mathbf{A} with rank r such that \mathbf{R}_{11} is $r \times r$ upper triangular (Theorem 3.1, p. 101).
15. *RRQR.* $\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{P}^\top$ = (orthogonal \mathbf{Q}) $\begin{pmatrix} \text{upper triangular} \\ \mathbf{R}_{11} \mid \text{full matrix} \\ \mathbf{R}_{12} \mid \mathbf{R}_{22} \text{ small in norm} \end{pmatrix}$ $\begin{pmatrix} \text{permutation} \\ \mathbf{P}^\top \end{pmatrix}$
Requirements: Any rectangular matrix \mathbf{A} with rank r such that \mathbf{R}_{11} is $r \times r$ upper triangular and \mathbf{R}_{22} is small in norm (Section 3.12, p. 107).
16. *LQ.* $\mathbf{A} = \mathbf{L}\mathbf{Q}$ =(lower triangular \mathbf{L})(orthonormal rows in \mathbf{Q})
Requirements: Rectangular matrix \mathbf{A} has linearly independent rows otherwise \mathbf{L} will be singular (diagonals contain at least one zero). In the full LQ decomposition, \mathbf{Q} is orthogonal such that $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ (Theorem 3.1, p. 124).
17. *RPLQ.* $\mathbf{A} = \mathbf{P}^\top \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{0} \end{bmatrix} \mathbf{Q}$ = (permutation \mathbf{P}^\top) $\begin{pmatrix} \text{lower triangular} \\ \mathbf{L}_{11} \mid \text{full matrix} \\ \mathbf{L}_{21} \end{pmatrix}$ (orthogonal \mathbf{Q})
Requirements: Any rectangular matrix \mathbf{A} with rank r such that \mathbf{L}_{11} is $r \times r$ lower triangular (Section 3.18, p. 123).
18. *Two-Sided Orthogonal.* $\mathbf{A}\mathbf{P}\mathbf{A} = \mathbf{U}\mathbf{F}\mathbf{V}^\top$ = $\begin{pmatrix} \text{orthonormal} \\ \text{basis in } \mathbf{U} \end{pmatrix}$ $\begin{pmatrix} \text{Upper-left} \\ r \times r \\ \text{submatrix} \\ \text{is nonzero} \end{pmatrix}$ $\begin{pmatrix} \text{orthonormal} \\ \text{basis in } \mathbf{V} \end{pmatrix}$

Requirements: Square matrix with rank r such that first r columns of \mathbf{U} span the column space of \mathbf{A} , and the rest columns span the null space of \mathbf{A}^\top . Whilst first r columns of \mathbf{V} span the row space of \mathbf{A} and the rest columns span the null space of \mathbf{A} (Theorem 3.1, p. 125).

$$19. UTV. \mathbf{A} = \mathbf{UTV} = (\text{orthogonal } \mathbf{U}) \begin{pmatrix} \text{upper triangular } \mathbf{R} \\ \text{lower triangular } \mathbf{L} \end{pmatrix} (\text{orthogonal } \mathbf{V})$$

Requirements: Any matrix \mathbf{A} with rank r such that \mathbf{T} is lower or upper triangular with rank r (Theorem 4.1, p. 141).

$$20. \text{Hessenberg. } \mathbf{A} = \mathbf{QH}\mathbf{Q}^\top = (\text{orthogonal matrix } \mathbf{Q})(\text{Hessenberg matrix } \mathbf{H})(\mathbf{Q}^\top = \mathbf{Q}^{-1})$$

Requirements: Any square matrix \mathbf{A} such that \mathbf{A}, \mathbf{H} are orthogonal similar matrices with the same rank, trace, eigenvalues (Theorem 8.2, p. 197).

$$21. \text{Tridiagonal. } \mathbf{A} = \mathbf{QT}\mathbf{Q}^\top = (\text{orthogonal } \mathbf{Q})(\text{Symmetric tridiagonal } \mathbf{T})(\mathbf{Q}^\top = \mathbf{Q}^{-1})$$

Requirements: Any symmetric matrix \mathbf{A} such that \mathbf{A}, \mathbf{T} are orthogonal similar matrices with same rank, trace, eigenvalues (Theorem 9.1, p. 209).

$$22. \text{Bidiagonal. } \mathbf{A} = \mathbf{UBV}^\top = (\text{orthogonal matrix } \mathbf{U})(\text{Bidiagonal } \mathbf{B})(\text{orthogonal matrix } \mathbf{V})$$

Requirements: Any rectangular matrix \mathbf{A} such that \mathbf{B} is upper bidiagonal. $\mathbf{T} = \mathbf{B}^\top \mathbf{B}$ is tridiagonal if \mathbf{B} is bidiagonal such that $\mathbf{A}^\top \mathbf{A} = \mathbf{VB}^\top \mathbf{BV}^\top$ is the tridiagonal decomposition of $\mathbf{A}^\top \mathbf{A}$ (Theorem 10.2, p. 215).

$$23. \text{Eigenvalue. } \mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1} = \begin{pmatrix} \text{(eigenvectors in)} \\ \text{columns of } \mathbf{X} \end{pmatrix} \begin{pmatrix} \text{(eigenvalues in)} \\ \text{diagonal of } \Lambda \end{pmatrix} \begin{pmatrix} \text{(left eigenvectors)} \\ \text{in } \mathbf{X}^{-1} \end{pmatrix}$$

Requirements: Square matrix \mathbf{A} has n linearly independent eigenvectors (Theorem 11.1, p. 230).

$$24. \text{Schur. } \mathbf{A} = \mathbf{QU}\mathbf{Q}^\top = \begin{pmatrix} \text{(orthonormal in)} \\ \text{columns in } \mathbf{Q} \end{pmatrix} (\text{Upper triangular } \mathbf{U}) (\mathbf{Q}^\top = \mathbf{Q}^{-1})$$

Requirements: \mathbf{A} is any real matrix with real eigenvalues (Theorem 12.1, p. 236).

$$25. \text{Spectral. } \mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top = \begin{pmatrix} \text{(orthonormal in)} \\ \text{eigenvectors in } \mathbf{Q} \end{pmatrix} \begin{pmatrix} \text{(real eigenvalues in)} \\ \text{diagonal of } \Lambda \end{pmatrix} \begin{pmatrix} \text{(left eigenvectors in)} \\ \text{in } \mathbf{Q}^\top \end{pmatrix}$$

Requirements: \mathbf{A} is real and symmetric, which is a special case of the Schur decomposition (Theorem 13.1, p. 241).

$$26. \text{Jordan. } \mathbf{A} = \mathbf{X}\mathbf{J}\mathbf{X}^{-1} = \begin{pmatrix} \text{(generalized eigenvectors in)} \\ \text{columns of } \mathbf{X} \end{pmatrix} \begin{pmatrix} \text{(Jordan blocks in)} \\ \text{in } \mathbf{J} \end{pmatrix} \begin{pmatrix} \text{(left generalized eigenvectors in)} \\ \text{in } \mathbf{X}^{-1} \end{pmatrix}$$

Requirements: \mathbf{A} is any square matrix, \mathbf{J} is a Jordan form matrix which has several diagonal blocks that contains identical eigenvalues in each block (Theorem 11.3, p. 232).

$$27. \text{SVD. } \mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top = \begin{pmatrix} \text{(left singular vectors in)} \\ \text{vectors in } \mathbf{U} \end{pmatrix} \begin{pmatrix} \text{(singular values in)} \\ \text{diagonal of } \Sigma \end{pmatrix} \begin{pmatrix} \text{(right singular vectors in)} \\ \text{vectors in } \mathbf{V} \end{pmatrix}$$

Requirements: Any matrix \mathbf{A} such that the diagonals of Σ^2 contain the eigenvalues of $\mathbf{A}^\top \mathbf{A}$ or $\mathbf{A}\mathbf{A}^\top$. In the full SVD, both \mathbf{U} and \mathbf{V} are orthogonal (Theorem 14.2, p. 265).

$$28. \text{Polar. } \mathbf{A} = \mathbf{Q}_l \mathbf{S}_l = \begin{pmatrix} \text{(orthogonal matrix } \mathbf{Q}_l \text{)} \\ \text{matrix } \mathbf{Q}_l \end{pmatrix} \begin{pmatrix} \text{(positive semidefinite } \mathbf{S}_l \text{)} \\ \text{semidefinite } \mathbf{S}_l \end{pmatrix}$$

Requirements: \mathbf{A} is any square matrix. The polar decomposition indicates $\mathbf{S}_l^2 = \mathbf{A}^\top \mathbf{A}$. When \mathbf{A} is nonsingular, \mathbf{S}_l is positive definite. Moreover, there exists a right polar decomposition $\mathbf{A} = \mathbf{S}_r \mathbf{Q}_r$ such that $\mathbf{S}_r^2 = \mathbf{A}\mathbf{A}^\top$ (Theorem 14.1, p. 275).

$$29. \text{ALS. } \mathbf{A} \approx \mathbf{WZ} = (s = \text{rank}(\mathbf{W}) < \text{rank}(\mathbf{A}) = r) (s = \text{rank}(\mathbf{Z}) < \text{rank}(\mathbf{A}) = r)$$

Requirements: For any rectangular matrix \mathbf{A} , \mathbf{WZ} approximates \mathbf{A} in the sense of Frobenius norm (Section 17, p. 339).

30. *NMF.* $\mathbf{A} \approx \mathbf{WZ} = (s = \text{rank}(\mathbf{W}) < \text{rank}(\mathbf{A}) = r) \quad (s = \text{rank}(\mathbf{Z}) < \text{rank}(\mathbf{A}) = r)$

Requirements: For any rectangular matrix \mathbf{A} , \mathbf{WZ} approximates \mathbf{A} in the sense of Frobenius norm with entries of $\mathbf{A}, \mathbf{W}, \mathbf{Z}$ being nonnegative (Section 18, p. 365).

31. *Biconjugate.* $\mathbf{A} = \Phi \Omega^{-1} \Psi^\top$

Requirements: Any rectangular matrix \mathbf{A} with rank r such that Ω is $r \times r$ diagonal.

The columns of Φ, Ψ comes from the Wedderburn sequence (Theorem 19.2, p. 373; Theorem 19.6, p. 376).

32. *CP.* $\mathbf{X} \approx [\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r=1}^R \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)}$

Requirements: Any tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ such that the CP decomposition is a low-rank approximation of the original tensor (Theorem 22.1, p. 402).

33. *Tucker.* $\mathbf{X} \approx [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)}$

Requirements: Any tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ such that the Tucker decomposition makes the principal components in high-order dimensions (Theorem 23.1, p. 408).

34. *HOSVD.* $\mathbf{X} \approx [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)}$

Requirements: Any tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ such that the HOSVD makes the principal components in high-order dimensions where further the slices of \mathbf{G} are mutually orthogonal and the slices are ordered in a descending manner (Theorem 24.1, p. 414).

35. *TT.* $\mathbf{X} \approx \mathbf{G}^{(1)} \boxtimes \mathbf{G}^{(2)} \boxtimes \dots \boxtimes \mathbf{G}^{(N)}$

Requirements: Any tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ such that the TT decomposition factors the tensor into N third-order tensors (Theorem 25.1, p. 420).

Part I

Gaussian Elimination

Introduction

In linear algebra, the *Gaussian elimination* is often referred to as the *row reduction*, which is an algorithm for solving systems of linear equations such that the original full linear equation is converted into the *row echelon form* (or in some cases, an upper triangular one) whilst the computational complexity is reduced.

In the row reduction process, the Gaussian elimination performs elementary row operations that can be divided into two phases. The first phase is also known as the *forward elimination* that reduces the linear system into an upper triangular one or the row echelon form in general, where the properties of the linear system is kept into the new one such as the rank and from which we can tell whether there are solutions, the uniqueness of the solution and so on. In the second phase, the process performs a *back substitution* such that the linear system is converted into a *reduced row echelon form* and the solution is found.

In this part, we will introduce the related LU, and Cholesky decompositions. And another highly relevant decomposition, CR decomposition, will be delayed into Part III as it skeletons the matrix and compresses the matrix into a thin one whilst sparsity, nonnegativity of the matrices are kept.

Chapter 1

LU Decomposition

Contents

1.1	LU Decomposition	30
1.2	Relation to Gaussian Elimination	32
1.3	Existence of the LU Decomposition without Permutation	35
1.4	Existence of the LU Decomposition with Permutation	36
1.5	Computing the LU without Pivoting Recursively: $A=LU$	38
1.5.1	Complexity of Matrix and Vector Operations	40
1.6	Computing the LU without Pivoting Element-Wise: $A=LU$	41
1.6.1	Extension to Thin Matrices	42
1.7	Computing the LU with Pivoting: $A=PLU$	43
1.8	Bandwidth Preserving in the LU Decomposition without Permutation	44
1.9	Block LU Decomposition	45
1.10	Application: Linear System via the LU Decomposition	46
1.11	Application: Computing the Inverse of Nonsingular Matrices	46
1.12	Application: Computing the Determinant via the LU Decomposition	48
1.13	Pivoting	48
1.13.1	Partial Pivoting	48
1.13.2	Complete Pivoting	51
1.13.3	Rook Pivoting	52
1.14	Rank-Revealing LU Decomposition	52
1.15	Rate of Change of L and U^*	52

1.1. LU Decomposition

Perhaps the best known and the first matrix decomposition we should know about is the LU decomposition. We now illustrate the results in the following theorem and the proof of the existence of which will be delayed in the next sections.

Theorem 1.1: (LU Decomposition with Permutation)

Every nonsingular $n \times n$ square matrix \mathbf{A} can be factored as

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

where \mathbf{P} is a permutation matrix, \mathbf{L} is a unit lower triangular matrix (i.e., lower triangular matrix with all 1's on the diagonal), and \mathbf{U} is a nonsingular upper triangular matrix.

The LU decomposition was the decomposition of Gauss's elimination algorithm, which he sketched in 1809 ([Gauss, 1809](#)) and presented in full in 1810 ([Gauss, 1810](#)). See also the discussion by ([Stewart, 2000](#)). Note that, in the remainder of this text, we will put the decomposition-related results in the blue box. And other claims will be in a gray box. This rule will be applied for the rest of the survey without special mention.

Remark 1.2: Decomposition Notation

The above decomposition applies to any nonsingular matrix \mathbf{A} . We will see that this decomposition arises from the elimination steps in which case row operations of subtraction and exchange of two rows are allowed where the subtractions are recorded in matrix \mathbf{L} and the row exchanges are recorded in matrix \mathbf{P} . To make this row exchange explicit, the common form for the above decomposition is $\mathbf{Q}\mathbf{A} = \mathbf{L}\mathbf{U}$ where $\mathbf{Q} = \mathbf{P}^\top$ that records the exact row exchanges of the rows of \mathbf{A} . Otherwise, the \mathbf{P} would record the row exchanges of $\mathbf{L}\mathbf{U}$. In our case, we will make the decomposition to be clear for matrix \mathbf{A} rather than for $\mathbf{Q}\mathbf{A}$. For this reason, we will put the permutation matrix on the right-hand side of the equation for the remainder of the text without special mention.

Specifically, in some cases, we will not need the permutation matrix. This decomposition relies on the leading principal minors. We provide the definitions which are important for the illustration.

Definition 1.3: Principal Minors

Let \mathbf{A} be an $n \times n$ square matrix. A $k \times k$ submatrix of \mathbf{A} obtained by deleting any $n - k$ columns and the same $n - k$ rows from \mathbf{A} is called a k -th order **principal submatrix** of \mathbf{A} . The determinant of a $k \times k$ principal submatrix is called a k -th order **principal minor** of \mathbf{A} .

Under mild conditions on the selected indices for the submatrix, we may obtain a specific kind of principal minors.

Definition 1.4: Leading Principal Minors

Let \mathbf{A} be an $n \times n$ square matrix. A $k \times k$ submatrix of \mathbf{A} obtained by deleting the last $n - k$ columns and the last $n - k$ rows from \mathbf{A} is called a k -th order **leading principal submatrix** of \mathbf{A} , that is, the $k \times k$ submatrix taken from the top left corner of \mathbf{A} . The determinant of the $k \times k$ leading principal submatrix is called a k -th order **leading principal minor** of \mathbf{A} .

Given an $n \times n$ matrix \mathbf{A} with (i, j) -th entry being a_{ij} , let $\mathbf{A}_{1:k,1:k}$ denote the $k \times k$ submatrix taken from the top left corner of \mathbf{A} . That is,

$$\mathbf{A}_{1:k,1:k} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{bmatrix}.$$

Then $\Delta_k = \det(\mathbf{A}_{1:k,1:k})$ is the k -th order leading principal minor of \mathbf{A} .

Under specific conditions on the leading principal minors of matrix \mathbf{A} , the LU decomposition will not involve the permutation matrix.

Theorem 1.5: (LU Decomposition without Permutation)

For any $n \times n$ square matrix \mathbf{A} , if all the leading principal minors are nonzero, i.e., $\det(\mathbf{A}_{1:k,1:k}) \neq 0$, for all $k \in \{1, 2, \dots, n\}$, then \mathbf{A} can be factored as

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

where \mathbf{L} is a unit lower triangular matrix (i.e., lower triangular matrix with all 1's on the diagonal), and \mathbf{U} is a **nonsingular** upper triangular matrix.

Specifically, this decomposition is **unique**. See Corollary 1.1.

Remark 1.6: Other Forms of the LU Decomposition without Permutation

The leading principal minors are nonzero, in another word, means the leading principal submatrices are nonsingular.

Singular \mathbf{A} In the above theorem, we assume \mathbf{A} is nonsingular as well. The LU decomposition also exists for singular matrix \mathbf{A} . However, the matrix \mathbf{U} will be singular as well in this case. This can be shown in the following section that, if matrix \mathbf{A} is singular, some pivots will be zero, and the corresponding diagonal values of \mathbf{U} will be zero.

Singular leading principal submatrices Even if we assume matrix \mathbf{A} is nonsingular, the leading principal submatrices might be singular. Suppose further that some of the leading principal minors are zero, the LU decomposition also exists, but if so, it is again not unique.

We will discuss where this decomposition comes from in the next section. There are also generalizations of LU decomposition to non-square or singular matrices, such as rank-revealing LU decomposition. Please refer to (Pan, 2000; Miranian and Gu, 2003; Dopico et al., 2006) or we will have a short discussion in Section 1.14.

1.2. Relation to Gaussian Elimination

Solving linear system equation $\mathbf{Ax} = \mathbf{b}$ is the basic problem in linear algebra. Gaussian elimination transforms a linear system into an upper triangular one by applying simple *elementary row transformations* on the left of the linear system in $n - 1$ stages if $\mathbf{A} \in \mathbb{R}^{n \times n}$. As a result, it is much easier to solve by a backward substitution. The elementary transformation is defined rigorously as follows.

Definition 1.1: Elementary Transformation

For square matrix \mathbf{A} , the following three transformations are referred as **elementary row/column transformations**:

1. Interchanging two rows (or columns) of \mathbf{A} ;
2. Multiplying all elements of a row (or a column) of \mathbf{A} by some nonzero number;
3. Adding any row (or column) of \mathbf{A} multiplied by a nonzero number to any other row (or column);

Specifically, the elementary row transformations of \mathbf{A} are unit lower triangular to multiply \mathbf{A} on the left, and the elementary column transformations of \mathbf{A} are unit upper triangular to multiply \mathbf{A} on the right.

The Gaussian elimination is described by the third type - elementary row transformation above. Suppose the upper triangular matrix obtained by Gaussian elimination is given by $\mathbf{U} = \mathbf{E}_{n-1}\mathbf{E}_{n-2}\dots\mathbf{E}_1\mathbf{A}$, and in the k -th stage, the k -th column of $\mathbf{E}_{k-1}\mathbf{E}_{k-2}\dots\mathbf{E}_1\mathbf{A}$ is $\mathbf{x} \in \mathbb{R}^n$. Gaussian elimination will introduce zeros below the diagonal of \mathbf{x} by

$$\mathbf{E}_k = \mathbf{I} - \mathbf{z}_k \mathbf{e}_k^\top,$$

where $\mathbf{e}_k \in \mathbb{R}^n$ is the k -th unit basis vector, and $\mathbf{z}_k \in \mathbb{R}^n$ is given by

$$\mathbf{z}_k = [0, \dots, 0, z_{k+1}, \dots, z_n]^\top, \quad z_i = \frac{x_i}{x_k}, \quad \forall i \in \{k+1, \dots, n\}.$$

We realize that \mathbf{E}_k is a unit lower triangular matrix (with 1's on the diagonal) with only the k -th column of the lower submatrix being nonzero,

$$\mathbf{E}_k = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & -z_{k+1} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -z_n & 0 & \dots & 1 \end{bmatrix},$$

and multiplying on the left by \mathbf{E}_k will introduce zeros below the diagonal:

$$\mathbf{E}_k \mathbf{x} = \begin{bmatrix} 1 & \dots & 0 & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 \\ 0 & \dots & -z_{k+1} & 1 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & -z_n & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ x_{k+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

For example, we write out the Gaussian elimination steps for a 4×4 matrix. For simplicity, we assume there are no row permutations. And in the following matrix, \boxtimes represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

A Trivial Gaussian Elimination For a 4×4 Matrix

$$\begin{array}{cccc} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{E}_1} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ \mathbf{0} & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ \mathbf{0} & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{E}_2} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{E}_3} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & 0 & 0 & \textcolor{blue}{\boxtimes} \end{bmatrix}, \\ \mathbf{A} & & \mathbf{E}_1 \mathbf{A} & & \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} & & \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} & \\ & & & & & & & (1.1) \end{array}$$

where $\mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3$ are lower triangular matrices. Specifically, as discussed above, Gaussian transformation matrices \mathbf{E}_i 's are unit lower triangular matrices with 1's on the diagonal. This can be explained that for the k -th transformation \mathbf{E}_k , working on the matrix $\mathbf{E}_{k-1} \dots \mathbf{E}_1 \mathbf{A}$, the transformation subtracts multiples of the k -th row from rows $\{k+1, k+2, \dots, n\}$ to get zeros below the diagonal in the k -th column of the matrix. And never use rows $\{1, 2, \dots, k-1\}$.

For the transformation example above, at step 1, we multiply left by \mathbf{E}_1 so that multiples of the 1-st row are subtracted from rows 2, 3, 4 and the first entries of rows 2, 3, 4 are set to zero. Similar situations for step 2 and step 3. By setting $\mathbf{L} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \mathbf{E}_3^{-1}$ and letting the matrix after elimination be \mathbf{U} , ¹ we get $\mathbf{A} = \mathbf{LU}$. Thus we obtain an LU decomposition for this 4×4 matrix \mathbf{A} .

¹. The inverses of unit lower triangular matrices are also unit lower triangular matrices. And the products of unit lower triangular matrices are also unit lower triangular matrices.

Definition 1.2: Pivot

First nonzero entry in the row after each elimination step is called a **pivot**. For example, the blue crosses in Equation (1.1) are pivots.

But sometimes, it can happen that the value of A_{11} is zero. No E_1 can make the next elimination step successful. So we need to interchange the first row and the second row via a permutation matrix P_1 . This is known as the **pivoting**, or **permutation**:

Gaussian Elimination With a Permutation In the Beginning

$$\begin{array}{c}
 \left[\begin{array}{cccc} 0 & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{P_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{E_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\
 A \qquad \qquad P_1A \qquad \qquad E_1P_1A
 \end{array}$$

$$\xrightarrow{E_2} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{E_3} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & 0 & 0 & \textcolor{blue}{\boxtimes} \end{array} \right] \\
 E_2E_1P_1A \qquad \qquad E_3E_2E_1P_1A$$

By setting $L = E_1^{-1}E_2^{-1}E_3^{-1}$ and $P = P_1^{-1}$, we get $A = PLU$. Therefore we obtain a full LU decomposition with permutation for this 4×4 matrix A .

In some situations, other permutation matrices P_2, P_3, \dots will appear in between the lower triangular E_i 's. An example is shown as follows:

Gaussian Elimination With a Permutation In Between

$$\begin{array}{c}
 \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{E_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{P_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{E_2} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & 0 & 0 & \textcolor{blue}{\boxtimes} \end{array} \right] \\
 A \qquad \qquad E_1A \qquad \qquad P_1E_1A \qquad \qquad E_2P_1E_1A
 \end{array}$$

In this case, we find $U = E_2P_1E_1A$. In Section 1.4, Section 1.7 or Section 1.13.1, we will show that the permutations in-between will result in the same form $A = PLU$ where P takes account of all the permutations.

The above examples can be easily extended to any $n \times n$ matrix if we assume there are no row permutations in the process. And we will have $n - 1$ such lower triangular transformations. The k -th transformation E_k introduces zeros below the diagonal in the k -th column of A by subtracting multiples of the k -th row from rows $\{k + 1, k + 2, \dots, n\}$. Finally, by setting $L = E_1^{-1}E_2^{-1}\dots E_{n-1}^{-1}$ we obtain the LU decomposition $A = LU$ (without permutation).

1.3. Existence of the LU Decomposition without Permutation

The Gaussian elimination or Gaussian transformation shows the origin of the LU decomposition. We then prove Theorem 1.5 rigorously, i.e., the existence of the LU decomposition without permutation by induction.

Proof [of Theorem 1.5: LU Decomposition without Permutation] We will prove by induction that every $n \times n$ square matrix \mathbf{A} with nonzero leading principal minors has a decomposition $\mathbf{A} = \mathbf{L}\mathbf{U}$. The 1×1 case is trivial by setting $\mathbf{L} = 1, \mathbf{U} = \mathbf{A}$, thus, $\mathbf{A} = \mathbf{L}\mathbf{U}$.

Suppose for any $k \times k$ matrix \mathbf{A}_k with all the leading principal minors being nonzero has an LU decomposition without permutation. If we prove any $(k+1) \times (k+1)$ matrix \mathbf{A}_{k+1} can also be factored as this LU decomposition without permutation, then we complete the proof.

For any $(k+1) \times (k+1)$ matrix \mathbf{A}_{k+1} , suppose the k -th order leading principal submatrix of \mathbf{A}_{k+1} is \mathbf{A}_k with size $k \times k$. Then \mathbf{A}_k can be factored as $\mathbf{A}_k = \mathbf{L}_k \mathbf{U}_k$ with \mathbf{L}_k being a unit lower triangular matrix and \mathbf{U}_k being a nonsingular upper triangular matrix from the assumption. Write out \mathbf{A}_{k+1} as $\mathbf{A}_{k+1} = \begin{bmatrix} \mathbf{A}_k & \mathbf{b} \\ \mathbf{c}^\top & d \end{bmatrix}$. Then it admits the factorization:

$$\mathbf{A}_{k+1} = \begin{bmatrix} \mathbf{A}_k & \mathbf{b} \\ \mathbf{c}^\top & d \end{bmatrix} = \begin{bmatrix} \mathbf{L}_k & \mathbf{0} \\ \mathbf{x}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \mathbf{y} \\ \mathbf{0} & z \end{bmatrix} = \mathbf{L}_{k+1} \mathbf{U}_{k+1},$$

where $\mathbf{b} = \mathbf{L}_k \mathbf{y}$, $\mathbf{c}^\top = \mathbf{x}^\top \mathbf{U}_k$, $d = \mathbf{x}^\top \mathbf{y} + z$, $\mathbf{L}_{k+1} = \begin{bmatrix} \mathbf{L}_k & \mathbf{0} \\ \mathbf{x}^\top & 1 \end{bmatrix}$, and $\mathbf{U}_{k+1} = \begin{bmatrix} \mathbf{U}_k & \mathbf{y} \\ \mathbf{0} & z \end{bmatrix}$. From the assumption, \mathbf{L}_k and \mathbf{U}_k are nonsingular. Therefore

$$\mathbf{y} = \mathbf{L}_k^{-1} \mathbf{b}, \quad \mathbf{x}^\top = \mathbf{c}^\top \mathbf{U}_k^{-1}, \quad z = d - \mathbf{x}^\top \mathbf{y}.$$

If further, we could prove z is nonzero such that \mathbf{U}_{k+1} is nonsingular, we complete the proof.

Since all the leading principal minors of \mathbf{A}_{k+1} are nonzero, we have $\det(\mathbf{A}_{k+1}) = \text{det}(\mathbf{A}_k) \cdot \det(d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b}) = \det(\mathbf{A}_k) \cdot (d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b}) \neq 0$, where $d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b}$ is a scalar. As $\det(\mathbf{A}_k) \neq 0$ from the assumption, we obtain $d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b} \neq 0$. Substitute $\mathbf{b} = \mathbf{L}_k \mathbf{y}$ and $\mathbf{c}^\top = \mathbf{x}^\top \mathbf{U}_k$ into the formula, we have $d - \mathbf{x}^\top \mathbf{U}_k \mathbf{A}_k^{-1} \mathbf{L}_k \mathbf{y} = d - \mathbf{x}^\top \mathbf{U}_k (\mathbf{L}_k \mathbf{U}_k)^{-1} \mathbf{L}_k \mathbf{y} = d - \mathbf{x}^\top \mathbf{y} \neq 0$ which is exactly the form of $z \neq 0$. Thus we find \mathbf{L}_{k+1} with all the values on the diagonal being 1, and \mathbf{U}_{k+1} with all the values on the diagonal being nonzero which means \mathbf{L}_{k+1} and \mathbf{U}_{k+1} are nonsingular, ³ from which the result follows. ■

We further prove that if no permutation involves, the LU decomposition is unique.

-
2. By the fact that if matrix \mathbf{M} has a block formulation: $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$, then $\det(\mathbf{M}) = \det(\mathbf{A}) \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$.
 3. A triangular matrix (upper or lower) is nonsingular if and only if all the entries on its main diagonal are nonzero.

Corollary 1.1: (Uniqueness of the LU Decomposition without Permutation)

Suppose the $n \times n$ square matrix \mathbf{A} has nonzero leading principal minors. Then, the LU decomposition is unique.

Proof [of Corollary 1.1] Suppose the LU decomposition is not unique, then we can find two decompositions such that $\mathbf{A} = \mathbf{L}_1\mathbf{U}_1 = \mathbf{L}_2\mathbf{U}_2$ which implies $\mathbf{L}_2^{-1}\mathbf{L}_1 = \mathbf{U}_2\mathbf{U}_1^{-1}$. The left of the equation is a unit lower triangular matrix and the right of the equation is an upper triangular matrix. This implies both sides of the above equation are diagonal matrices. Since the inverse of a unit lower triangular matrix is also a unit lower triangular matrix, and the product of unit lower triangular matrices is also a unit lower triangular matrix, this results in that $\mathbf{L}_2^{-1}\mathbf{L}_1 = \mathbf{I}$. The equality implies that both sides are identity such that $\mathbf{L}_1 = \mathbf{L}_2$ and $\mathbf{U}_1 = \mathbf{U}_2$ and leads to a contradiction. ■

In the proof of Theorem 1.5, we have shown that the diagonal values of the upper triangular matrix are all nonzero if the leading principal minors of \mathbf{A} are all nonzero. We then can formulate this decomposition in another form if we divide each row of \mathbf{U} by each diagonal value of \mathbf{U} . This is called the *LDU decomposition*.

Corollary 1.2: (LDU Decomposition)

For any $n \times n$ square matrix \mathbf{A} , if all the leading principal minors are nonzero, i.e., $\det(\mathbf{A}_{1:k,1:k}) \neq 0$, for all $k \in \{1, 2, \dots, n\}$, then \mathbf{A} can be uniquely factored as

$$\mathbf{A} = \mathbf{LDU},$$

where \mathbf{L} is a unit lower triangular matrix, \mathbf{U} is a **unit** upper triangular matrix, and \mathbf{D} is a diagonal matrix.

The proof is trivial that from the LU decomposition of $\mathbf{A} = \mathbf{LR}$, we can find a diagonal matrix $\mathbf{D} = \text{diag}(\mathbf{R}_{11}, \mathbf{R}_{22}, \dots, \mathbf{R}_{nn})$ such that $\mathbf{D}^{-1}\mathbf{R} = \mathbf{U}$ is a unit upper triangular matrix. And the uniqueness comes from the uniqueness of the LU decomposition.

1.4. Existence of the LU Decomposition with Permutation

In Theorem 1.5, we require that matrix \mathbf{A} has nonzero leading principal minors. However, this is not necessarily. Even when the leading principal minors are zero, nonsingular matrices still have an LU decomposition, but with an additional permutation. The proof is still from induction.

Proof [of Theorem 1.1: LU Decomposition with Permutation] We note that any 1×1 nonsingular matrix has a full LU decomposition $A = PLU$ by simply setting $P = 1$, $L = 1$, $U = A$. We will show that if every $(n - 1) \times (n - 1)$ nonsingular matrix has a full LU decomposition, then this is also true for every $n \times n$ nonsingular matrix. By induction, we prove that every nonsingular matrix has a full LU decomposition.

We will formulate the proof in the following order. If \mathbf{A} is nonsingular, then its row permuted matrix \mathbf{B} is also nonsingular. And Schur complement of \mathbf{B}_{11} in \mathbf{B} is also nonsingular. Finally, we formulate the decomposition of \mathbf{A} by \mathbf{B} from this property.

We notice that at least one element in the first column of \mathbf{A} must be nonzero otherwise \mathbf{A} will be singular. We can then apply a row permutation that makes the element in entry $(1, 1)$ to be nonzero. That is, there exists a permutation \mathbf{P}_1 such that $\mathbf{B} = \mathbf{P}_1 \mathbf{A}$ in which case $\mathbf{B}_{11} \neq 0$. Since \mathbf{A} and \mathbf{P}_1 are both nonsingular and the product of nonsingular matrices is also nonsingular, then \mathbf{B} is also nonsingular.

Schur complement of \mathbf{B} is also nonsingular:

Now consider the Schur complement of \mathbf{B}_{11} in \mathbf{B} with size $(n - 1) \times (n - 1)$

$$\bar{\mathbf{B}} = \mathbf{B}_{2:n,2:n} - \frac{1}{\mathbf{B}_{11}} \mathbf{B}_{2:n,1} \mathbf{B}_{1,2:n}.$$

Suppose there is an $(n - 1)$ -vector \mathbf{x} satisfies

$$\bar{\mathbf{B}}\mathbf{x} = 0. \quad (1.2)$$

Then \mathbf{x} and $y = -\frac{1}{\mathbf{B}_{11}} \mathbf{B}_{1,2:n} \mathbf{x}$ satisfy

$$\mathbf{B} \begin{bmatrix} \mathbf{x} \\ y \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1,2:n} \\ \mathbf{B}_{2:n,1} & \mathbf{B}_{2:n,2:n} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ y \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}.$$

Since \mathbf{B} is nonsingular, \mathbf{x} and y must be zero. Hence, Equation (1.2) holds only if $\mathbf{x} = \mathbf{0}$ which means that the null space of $\bar{\mathbf{B}}$ is of dimension 0 and thus $\bar{\mathbf{B}}$ is nonsingular with size $(n - 1) \times (n - 1)$.

By the induction assumption that any $(n - 1) \times (n - 1)$ nonsingular matrix can be factorized as the full LU decomposition form

$$\bar{\mathbf{B}} = \mathbf{P}_2 \mathbf{L}_2 \mathbf{U}_2.$$

We then factor \mathbf{A} as

$$\begin{aligned} \mathbf{A} &= \mathbf{P}_1^\top \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1,2:n} \\ \mathbf{B}_{2:n,1} & \mathbf{B}_{2:n,2:n} \end{bmatrix} \\ &= \mathbf{P}_1^\top \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1,2:n} \\ \mathbf{P}_2^\top \mathbf{B}_{2:n,1} & \mathbf{P}_2^\top \mathbf{B}_{2:n,2:n} \end{bmatrix} \\ &= \mathbf{P}_1^\top \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1,2:n} \\ \mathbf{P}_2^\top \mathbf{B}_{2:n,1} & \mathbf{L}_2 \mathbf{U}_2 + \mathbf{P}_2^\top \frac{1}{\mathbf{B}_{11}} \mathbf{B}_{2:n,1} \mathbf{B}_{1,2:n} \end{bmatrix} \\ &= \mathbf{P}_1^\top \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{\mathbf{B}_{11}} \mathbf{P}_2^\top \mathbf{B}_{2:n,1} & \mathbf{L}_2 \end{bmatrix} \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1,2:n} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix}. \end{aligned}$$

Therefore, we find the full LU decomposition of $\mathbf{A} = \mathbf{PLU}$ by defining

$$\mathbf{P} = \mathbf{P}_1^\top \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\mathbf{B}_{11}} \mathbf{P}_2^\top \mathbf{B}_{2:n,1} & \mathbf{L}_2 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1,2:n} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix},$$

from which the result follows. We will formulate this process into Algorithm 4 to compute this decomposition. \blacksquare

1.5. Computing the LU without Pivoting Recursively: $\mathbf{A} = \mathbf{L}\mathbf{U}$

As a start, we stay with the most frequent and easy case without involving row exchanges

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

where \mathbf{L} is the unit lower triangular matrix and \mathbf{U} is the nonsingular upper triangular matrix. We refer to this decomposition as the LU decomposition without pivoting (or without permutation).

In Section 1.2, we mentioned the connection of the LU decomposition to the Gaussian elimination. We could find the LU decomposition of a matrix by first applying Gaussian elimination to \mathbf{A} to get \mathbf{U} , and then examine the multipliers in the Gaussian elimination process to determine the entries below the main diagonal of \mathbf{L} . We will now look at another method for finding the LU decomposition without going through the process of Gaussian elimination.

Again, we define $\mathbf{A}_{i:j,m:n}$ to be the $(j - i + 1) \times (n - m + 1)$ submatrix of \mathbf{A} with rows $i, i + 1, \dots, j$ and columns $m, m + 1, \dots, n$ of \mathbf{A} . And simply, \mathbf{A}_{ij} to be the (i, j) -th entry of \mathbf{A} .

Assume \mathbf{A} has the form $\mathbf{A} = \mathbf{L}\mathbf{U}$ of the LU decomposition. We will see, this is equivalent to assuming \mathbf{A}_{11} is not zero, which implies leading principal minors are nonzero.

From the property of lower triangular matrices and upper triangular matrices, we suppose \mathbf{A} can be factored as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{1,2:n} \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \mathbf{L}_{2:n,1} & \mathbf{L}_{2:n,2:n} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{1,2:n} \\ 0 & \mathbf{U}_{2:n,2:n} \end{bmatrix} = \mathbf{L}\mathbf{U}.$$

Writing out the product on the right-hand side of the above equation, we obtain

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{1,2:n} \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{1,2:n} \\ \mathbf{U}_{11}\mathbf{L}_{2:n,1} & \mathbf{L}_{2:n,1}\mathbf{U}_{1,2:n} + \mathbf{L}_{2:n,2:n}\mathbf{U}_{2:n,2:n} \end{bmatrix},$$

which helps us decide the values of \mathbf{L} and \mathbf{U} by

$$\left. \begin{aligned} \mathbf{U}_{11} &= \mathbf{A}_{11} \\ \mathbf{U}_{1,2:n} &= \mathbf{A}_{1,2:n} \end{aligned} \right\} \quad \text{i.e.,} \quad \mathbf{U}_{1,1:n} = \mathbf{A}_{1,1:n},$$

$$\mathbf{L}_{2:n,1} = \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1},$$

and

$$\mathbf{L}_{2:n,2:n}\mathbf{U}_{2:n,2:n} = \mathbf{A}_{2:n,2:n} - \mathbf{L}_{2:n,1}\mathbf{U}_{1,2:n} = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n}.$$

As $\mathbf{L}_{2:n,2:n} \in \mathbb{R}^{(n-1) \times (n-1)}$ is also a unit lower triangular matrix and $\mathbf{U}_{2:n,2:n} \in \mathbb{R}^{(n-1) \times (n-1)}$ is a nonsingular upper triangular matrix both of which are of size $(n - 1) \times (n - 1)$ from

the context. Let $\mathbf{A}_2 = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n}$. So we can calculate $\mathbf{L}_{2:n,2:n}$ and $\mathbf{U}_{2:n,2:n}$ by factoring \mathbf{A}_2 as

$$\mathbf{A}_2 = \mathbf{L}_{2:n,2:n} \mathbf{U}_{2:n,2:n},$$

which is an LU decomposition of a matrix with size $(n - 1) \times (n - 1)$. This suggests a recursive algorithm: to factorize a matrix of size $n \times n$, we calculate the first column of \mathbf{L} (whose first row can be calculated implicitly as 1) and the first row of \mathbf{U} (whose first column can be calculated implicitly as \mathbf{A}_{11}) leaving the other $n - 1$ columns and $n - 1$ rows of them to the next round. Continuing recursively, we arrive at a decomposition of a 1×1 matrix.

A word on the leading principal minors In this process, we only assume the elements in entry $(1, 1)$ of $\mathbf{A}, \mathbf{A}_2, \mathbf{A}_3, \dots, \mathbf{A}_n$ are nonzero. This is actually the same assumption as the leading principal minors of \mathbf{A} are all nonzero. The recursive process is formulated in Algorithm 1.

Algorithm 1 LU Decomposition without Pivoting Recursively

Require: Matrix \mathbf{A} is nonsingular and square with size $n \times n$;

- 1: Calculate the first row of \mathbf{U} : $\mathbf{U}_{1,1:n} = \mathbf{A}_{1,1:n}; \quad \triangleright 0 \text{ flops}$
- 2: Calculate the first column of \mathbf{L} : $\mathbf{L}_{11} = 1$ and $\mathbf{L}_{2:n,1} = \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1}; \quad \triangleright n - 1 \text{ flops}$
- 3: Calculate the LU decomposition

$$\mathbf{A}_2 = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n} = \mathbf{L}_{2:n,2:n} \mathbf{U}_{2:n,2:n};$$

$\triangleright 2(n - 1)^2 \text{ flops}$

Schur complement: Assume \mathbf{A}_{11} is not zero, then $\mathbf{A}_2 = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n}$ is called the Schur complement of \mathbf{A}_{11} in \mathbf{A} . More details about Schur complement can refer to Appendix F.

Operation count: The LU decomposition algorithm without pivoting is the first algorithm we have presented in this survey. It is important to assess its cost. To do so, we follow the classical route and count the number of floating-point operations (flops) that the algorithm requires. Each addition, subtraction, multiplication, division, and square root counts as one flop. Note that we have the convention that an assignment operation does not count as one flop.

Theorem 1.1: (Algorithm Complexity: LU without Pivoting Recursively)

Algorithm 1 requires $\sim (2/3)n^3$ flops to compute the LU decomposition of an $n \times n$ matrix. Note that the theorem expresses only the leading term of the flop count. And the symbol “ \sim ” has the usual asymptotic meaning

$$\lim_{n \rightarrow +\infty} \frac{\text{number of flops}}{(2/3)n^3} = 1.$$

Proof [of Theorem 1.1] The step 1 in Algorithm 1 costs 0 flops and step 2 involves $(n - 1)$ divisions.

In step 3, we can compute $\frac{1}{A_{11}}A_{2:n,1}$ firstly which costs 0 flops as it has been calculated in step 2, and the outer product with $A_{1,2:n}$ costs $(n - 1)^2$ multiplications/flops. So the computation of $\frac{1}{A_{11}}A_{2:n,1}A_{1,2:n}$ involves $(n - 1)^2$ multiplications. If we calculate $A_{2:n,1}A_{1,2:n}$ firstly, then the costs of $\frac{1}{A_{11}}A_{2:n,1}A_{1,2:n}$ is $2(n - 1)^2$ totally. So we choose the first way to do the computation. Furthermore, The subtraction of the two matrices requires $(n - 1)^2$ flops. As a result, the cost of step 3 is $2(n - 1)^2$ flops in total.

It can be shown that the costs of the final loop is $2(n - 1)^2 + (n - 1) = 2n^2 - 3n + 1$ flops. Let $f(n) = 2n^2 - 3n + 1$, the final cost can then be calculated by

$$\text{cost} = f(n) + f(n - 1) + \dots + f(1).$$

Simple calculation⁴ can show that the complexity is $(2/3)n^3 - (1/2)n^2 - (1/6)n$ flops, or $(2/3)n^3$ flops if we keep only the leading term. ■

1.5.1 Complexity of Matrix and Vector Operations

The calculation of the complexity extensively relies on the complexity of the multiplication of two matrices so that we formulate the finding in the following lemma.

Lemma 1.2: (Vector Inner Product Complexity)

Given two vectors $v, w \in \mathbb{R}^n$. The inner product of the two vectors $v^\top w$ is given by $v^\top w = v_1w_1 + v_2w_2 + \dots + v_nw_n$ which involves n scalar multiplications and $n - 1$ scalar additions. Therefore the complexity for the inner product is $2n - 1$ flops.

The matrix multiplication thus relies on the complexity of the inner product.

Lemma 1.3: (Matrix Multiplication Complexity)

For matrix $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{n \times k}$, the complexity of the multiplication $C = AB$ is $mk(2n - 1)$ flops.

Proof [of Lemma 1.3] We notice that each entry of C involves a vector inner product which requires n multiplications and $n - 1$ additions. And there are mk such entries which leads to the conclusion. ■

⁴ By the fact that $1^2 + 2^2 + \dots + n^2 = \frac{2n^3 + 3n^2 + n}{6}$ and $1 + 2 + \dots + n = \frac{n(n+1)}{2}$.

1.6. Computing the LU without Pivoting Element-Wise: $\mathbf{A}=\mathbf{LU}$

We notice that computing the LU decomposition is equivalent to solving the following equations

$$\mathbf{A}_{ij} = \sum_{s=1}^{\min(i,j)} \mathbf{L}_{is} \mathbf{U}_{sj}, \quad \forall i, j \in \{1, 2, \dots, n\}.$$

Furthermore, when $i \leq j$, the above equation can be decomposed into

$$\mathbf{A}_{ij} = \sum_{s=1}^{i-1} \mathbf{L}_{is} \mathbf{U}_{sj} + \mathbf{U}_{ij}, \quad \text{since } \mathbf{L}_{ii} = 1.$$

Suppose we know the first $k-1$ columns of \mathbf{L} and the first $k-1$ rows of \mathbf{U} , we have the following observations:

$$\begin{aligned} \mathbf{A}_{kj} &= \sum_{s=1}^{k-1} \mathbf{L}_{ks} \mathbf{U}_{sj} + \mathbf{U}_{kj}, & \text{for all } j \in \{k, k+1, \dots, n\}, & \text{since } k \leq j. \\ \mathbf{A}_{ik} &= \sum_{s=1}^{k-1} \mathbf{L}_{is} \mathbf{U}_{sk} + \mathbf{L}_{ik} \mathbf{U}_{kk}, & \text{for all } i \in \{k+1, k+2, \dots, n\}, & \text{since } i \geq k. \end{aligned}$$

Therefore, the k -th row of \mathbf{U} and k -th column of \mathbf{L} can be obtained by

$$\begin{aligned} \mathbf{U}_{kj} &= \mathbf{A}_{kj} - \sum_{s=1}^{k-1} \mathbf{L}_{is} \mathbf{U}_{sj}, & \text{for all } j \in \{k, k+1, \dots, n\}, & \text{since } k \leq j. \\ \mathbf{L}_{ik} &= (\mathbf{A}_{ik} - \sum_{s=1}^{k-1} \mathbf{L}_{is} \mathbf{U}_{sk}) / \mathbf{U}_{kk}, & \text{for all } i \in \{k+1, k+2, \dots, n\}, & \text{since } i \geq k. \end{aligned}$$

This is known as the *Doolittle's method*, the values of \mathbf{U} and \mathbf{L} can be computed element-wise, and the process is formulated in Algorithm 2. We notice that, mathematically, the Doolittle's method is equivalent to the recursive algorithm, but from different perspectives.

Algorithm 2 LU Decomposition without Pivoting Element-Wise

Require: Matrix \mathbf{A} with size $n \times n$;

- 1: Calculate first of \mathbf{R} by $\mathbf{R}_{11} = \sqrt{\mathbf{A}_{11}}$;
 - 2: **for** $k = 1$ to n **do**
 - 3: //i.e., the k -th column of \mathbf{U} and k -th row of \mathbf{L} ;
 - 4: **for** $j = k$ to n **do**
 - 5: $\mathbf{U}_{kj} = \mathbf{A}_{kj} - \sum_{s=1}^{k-1} \mathbf{L}_{is} \mathbf{U}_{sj}$;
 - 6: **end for**
 - 7: **for** $i = k+1$ to n **do**
 - 8: $\mathbf{L}_{ik} = (\mathbf{A}_{ik} - \sum_{s=1}^{k-1} \mathbf{L}_{is} \mathbf{U}_{sk}) / \mathbf{U}_{kk}$;
 - 9: **end for**
 - 10: **end for**
-

Theorem 1.1: (Algorithm Complexity: LU without Pivoting Elementwise)

Algorithm 2 requires $\sim (2/3)n^3$ flops to compute the LU decomposition of an $n \times n$ matrix.

Proof [of Theorem 1.1] The step 5 in Algorithm 2 requires $(k - 1)$ multiplications, $(k - 2)$ additions, and 1 subtractions for each loop k, j . And there are $n - k + 1$ such loop j 's, that is, $(2k - 2)(n - k + 1) = -2k^2 + (2n + 4)k - 2(n + 1)$ flops totally from step 5 for each loop k .

Similarly, the step 8 requires $(k - 1)$ multiplications, $(k - 2)$ additions, 1 subtraction, and 1 division for each loop k, j . And there are $n - k$ such loop j 's, that is, $(2k - 1)(n - k) = -2k^2 + (2n + 1)k - n$ flops totally from step 8 for each loop k .

Thus, for each loop k , the total complexity is $(2k - 2)(n - k + 1) + (2k - 1)(n - k) = -4k^2 + (4n + 5)k - (3n + 2)$ flops. Let $f(k) = -4k^2 + (4n + 5)k - (3n + 2)$, the final complexity can be computed by

$$\text{cost} = f(1) + f(2) + \dots + f(n).$$

A simple calculation can show that the complexity is $(2/3)n^3$ flops if we keep only the leading term. ■

1.6.1 Extension to Thin Matrices

We notice that the complexity of Algorithm 2 is the same as that of Algorithm 1. Doolittle's method is mathematically equivalent to the recursive algorithm. However, the Doolittle's method can be extended to compute the LU decomposition for $\mathbf{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$, i.e., a thin matrix. The LU decomposition is given by $\mathbf{A} = \mathbf{L}\mathbf{U}$ where $\mathbf{L} \in \mathbb{R}^{m \times n}$ and $\mathbf{U} \in \mathbb{R}^{n \times n}$ are upper triangular. That is, \mathbf{L} is trapezoidal: $L_{ij} = 0$ for $i < j$. The procedure is similar and is formulated in Algorithm 3 where the difference is illustrated in blue text.

Algorithm 3 Thin LU Decomposition without Pivoting Element-Wise

Require: Matrix \mathbf{A} with size $m \times n$;

- 1: Calculate first of \mathbf{R} by $R_{11} = \sqrt{A_{11}}$;
 - 2: **for** $k = 1$ to n **do**
 - 3: //i.e., the k -th column of \mathbf{U} and k -th row of \mathbf{L} ;
 - 4: **for** $j = k$ to n **do**
 - 5: $U_{kj} = A_{kj} - \sum_{s=1}^{k-1} L_{is}U_{sj}$;
 - 6: **end for**
 - 7: **for** $i = k + 1$ to m **do**
 - 8: $L_{ik} = (A_{ik} - \sum_{s=1}^{k-1} L_{is}U_{sk})/U_{kk}$;
 - 9: **end for**
 - 10: **end for**
-

Theorem 1.2: (Algorithm Complexity: LU Thin Matrix)

Algorithm 3 requires $\sim n^2(m - n/3)$ flops to compute the LU decomposition of an $n \times n$ matrix.

When $m = n$, the complexity of Algorithm 3 is $(2/3)n^3$ flops, which is the same as that of Algorithm 2.

Proof [of Theorem 1.2] The complexity of step 5 is the same as that in Algorithm 2, which is $(2k - 2)(n - k + 1) = -2k^2 + (2n + 4)k - 2(n + 1)$ flops totally from step 5 for each loop k .

The complexity of step 8 is slightly different where we replace n by m , and it requires $(2k - 1)(m - k) = -2k^2 + (2m + 1)k - m$ flops totally from step 8 for each loop k .

Thus, for each loop k , the total complexity is $(2k - 2)(n - k + 1) + (2k - 1)(m - k) = -4k^2 + (2m + 2n + 5)k - (2n + m + 2)$ flops. Let $f(k) = -4k^2 + (2m + 2n + 5)k - (2n + m + 2)$, the final complexity can be calculated by

$$\text{cost} = f(1) + f(2) + \dots + f(n).$$

A simple calculation can show that the complexity is $n^2(m - n/3)$ flops if we keep only the leading term. ■

1.7. Computing the LU with Pivoting: $\mathbf{A} = \mathbf{PLU}$

Further, we extend Algorithm 1 to the full LU decomposition with $\mathbf{A} = \mathbf{PLU}$. Note that we assume \mathbf{A}_{11} is nonzero in Algorithm 1. This is not necessarily true. We will avoid this assumption by permutation matrix. The following algorithm is just formulated from the proof of Theorem 1.1.

Algorithm 4 LU Decomposition with Pivoting

Require: matrix \mathbf{A} is nonsingular and square with size $n \times n$;

- 1: Choose permutation matrix \mathbf{P}_1 such that $\bar{\mathbf{B}} = \mathbf{P}_1 \mathbf{A}$ and $\bar{\mathbf{B}}_{11} \neq 0$; $\triangleright 0$ flops
- 2: Calculate the $\bar{\mathbf{B}}$ for next round: $\bar{\mathbf{B}} = \bar{\mathbf{B}}_{2:n,2:n} - \frac{1}{\bar{\mathbf{B}}_{11}} \bar{\mathbf{B}}_{2:n,1} \bar{\mathbf{B}}_{1,2:n} = \mathbf{P}_2 \mathbf{L}_2 \mathbf{U}_2$; $\triangleright 2(n - 1)^2 + (n - 1)$ flops
- 3: Calculate the full LU decomposition of $\mathbf{A} = \mathbf{PLU}$ with

$$\mathbf{P} = \mathbf{P}_1^\top \begin{bmatrix} 1 & 0 \\ 0 & \mathbf{P}_2 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\bar{\mathbf{B}}_{11}} \mathbf{P}_2^\top \bar{\mathbf{B}}_{2:n,1} & \mathbf{L}_2 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} \bar{\mathbf{B}}_{11} & \bar{\mathbf{B}}_{1,2:n} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix}$$

$\triangleright n - 1$ flops

Theorem 1.1: (Algorithm Complexity: LU with Pivoting)

Algorithm 4 requires $\sim (2/3)n^3$ flops to compute a full LU decomposition of an $n \times n$ nonsingular matrix.

Proof [of Theorem 1.1] The step 1 costs 0 flops as it only involves assignment operations and step 2 involves $(n - 1)^2$ multiplications and $(n - 1)^2 + (n - 1)$ divisions which costs $2(n - 1)^2 + (n - 1)$ flops to compute $\bar{\mathbf{B}} = \mathbf{B}_{2:n,2:n} - \frac{1}{\mathbf{B}_{11}}\mathbf{B}_{2:n,1}\mathbf{B}_{1,2:n}$ as shown in the proof of Theorem 1.1.

The computation of step 3 results from $\frac{1}{\mathbf{B}_{11}}\mathbf{P}_2^\top \mathbf{B}_{2:n,1}$ which costs $n - 1$ flops as the permutation does not count.

So it costs $(2n^2 - 2n)$ flops in the final loop. Let $f(n) = 2n^2 - 2n$, the final complexity can be calculated by

$$\text{cost} = f(n) + f(n - 1) + \dots + f(1).$$

Simple calculations can show that the complexity is $(2/3)n^3 - (2/3)n$ flops, or $(2/3)n^3$ flops if we keep only the leading term. ■

1.8. Bandwidth Preserving in the LU Decomposition without Permutation

For any matrix, the bandwidth of it can be defined as follows.

Definition 1.1: Matrix Bandwidth

For any matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with entry (i, j) element denoted by A_{ij} . Then \mathbf{A} has **upper bandwidth** q if $A_{ij} = 0$ for $j > i + q$, and **lower bandwidth** p if $A_{ij} = 0$ for $i > j + p$.

An example of a 6×6 matrix with upper bandwidth 2 and lower bandwidth 3 is shown as follows:

$$\begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}.$$

Then, we prove that the bandwidth after the LU decomposition without permutation is preserved.

Lemma 1.2: (Bandwidth Preserving)

For any matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with upper bandwidth q and lower bandwidth p . If \mathbf{A} has an LU decomposition $\mathbf{A} = \mathbf{L}\mathbf{U}$, then \mathbf{U} has upper bandwidth q and \mathbf{L} has lower bandwidth p .

Proof [of Lemma 1.2] Following from the computation of the LU decomposition without permutation in Section 1.5, and from the property of lower triangular matrices and upper triangular matrices, we have the decomposition for \mathbf{A} as follows

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{1,2:n} \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} & \mathbf{I}_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{1,2:n} \\ 0 & \mathbf{S} \end{bmatrix} = \mathbf{L}_1 \mathbf{U}_1,$$

where $\mathbf{S} = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n}$ is the Schur complement of \mathbf{A}_{11} in \mathbf{A} . We can name this decomposition of \mathbf{A} as the s -decomposition of \mathbf{A} . The first column of \mathbf{L}_1 and the first row of \mathbf{U}_1 have the required structure (bandwidth p and q respectively), and the Schur complement \mathbf{S} of \mathbf{A}_{11} has upper bandwidth $q-1$ and lower bandwidth $p-1$ respectively. The result follows by induction on the s -decomposition of \mathbf{S} . ■

1.9. Block LU Decomposition

Another form of the LU decomposition is to factor the matrix into block triangular matrices.

Theorem 1.1: (Block LU Decomposition without Permutation)

For any $n \times n$ square matrix \mathbf{A} , if the first m leading principal block submatrices are nonsingular, then \mathbf{A} can be factored as

$$\mathbf{A} = \mathbf{L}\mathbf{U} = \begin{bmatrix} \mathbf{I} & & & \\ \mathbf{L}_{21} & \mathbf{I} & & \\ \vdots & & \ddots & \\ \mathbf{L}_{m1} & \dots & \mathbf{L}_{m,m-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} & \dots & \mathbf{U}_{1m} \\ \mathbf{U}_{22} & & & \vdots \\ & \ddots & & \mathbf{U}_{m-1,m} \\ & & & \mathbf{U}_{mm} \end{bmatrix},$$

where $\mathbf{L}_{i,j}$'s and \mathbf{U}_{ij} 's are some block matrices.

Specifically, this decomposition is unique.

Note that the \mathbf{U} in the above theorem is not necessarily upper triangular. An example can be shown as follows:

$$\mathbf{A} = \left[\begin{array}{cc|cc} 0 & 1 & 1 & 1 \\ -1 & 2 & -1 & 2 \\ \hline 2 & 1 & 4 & 2 \\ 1 & 2 & 3 & 3 \end{array} \right] = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \hline 5 & -2 & 1 & 0 \\ 4 & -1 & 0 & 1 \end{array} \right] \left[\begin{array}{cc|cc} 0 & 1 & 1 & 1 \\ -1 & 2 & -1 & 2 \\ \hline 0 & 0 & -3 & 1 \\ 0 & 0 & -2 & 1 \end{array} \right].$$

The trivial non-block LU decomposition fails on \mathbf{A} since the entry $(1,1)$ is zero. However, the block LU decomposition exists.

1.10. Application: Linear System via the LU Decomposition

Consider the well-determined linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$ with \mathbf{A} of size $n \times n$ and nonsingular. Avoid solving the system by computing the inverse of \mathbf{A} , we solve linear equation by the LU decomposition. Suppose \mathbf{A} admits the LU decomposition $\mathbf{A} = \mathbf{P}\mathbf{L}\mathbf{U}$. The solution is given by the following algorithm.

Algorithm 5 Solving Linear Equations by LU Decomposition

Require: matrix \mathbf{A} is nonsingular and square with size $n \times n$, solve $\mathbf{A}\mathbf{x} = \mathbf{b}$;

- | | |
|---|---|
| 1: LU Decomposition: factor \mathbf{A} as $\mathbf{A} = \mathbf{P}\mathbf{L}\mathbf{U}$; | $\triangleright (2/3)n^3$ flops |
| 2: Permutation: $\mathbf{w} = \mathbf{P}^\top \mathbf{b}$; | $\triangleright 0$ flops |
| 3: Forward substitution: solve $\mathbf{L}\mathbf{v} = \mathbf{w}$; | $\triangleright 1 + 3 + \dots + (2n - 1) = n^2$ flops |
| 4: Backward substitution: solve $\mathbf{U}\mathbf{x} = \mathbf{v}$; | $\triangleright 1 + 3 + \dots + (2n - 1) = n^2$ flops |
-

The complexity of the decomposition step is $(2/3)n^3$ flops, the backward and forward substitution steps both cost $1 + 3 + \dots + (2n - 1) = n^2$ flops. Therefore, the total cost for computing the linear system via the LU factorization is $(2/3)n^3 + 2n^2$ flops. If we keep only the leading term, the Algorithm 5 costs $(2/3)n^3$ flops where the most cost comes from the LU decomposition.

Linear system via the block LU decomposition For a block LU decomposition of $\mathbf{A} = \mathbf{L}\mathbf{U}$, we need to solve $\mathbf{L}\mathbf{v} = \mathbf{w}$ and $\mathbf{U}\mathbf{x} = \mathbf{v}$. But the latter system is not triangular and requires some extra computations.

1.11. Application: Computing the Inverse of Nonsingular Matrices

By Theorem 1.1, for any nonsingular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, we have a full LU factorization $\mathbf{A} = \mathbf{P}\mathbf{L}\mathbf{U}$. Then the inverse can be obtained by solving the matrix equation

$$\mathbf{A}\mathbf{X} = \mathbf{I},$$

which contains n linear systems computation: $\mathbf{A}\mathbf{x}_i = \mathbf{e}_i$ for all $i \in \{1, 2, \dots, n\}$ where \mathbf{x}_i is the i -th column of \mathbf{X} and \mathbf{e}_i is the i -th column of \mathbf{I} (i.e., the i -th unit vector).

Theorem 1.1: (Inverse of Nonsingular Matrix by Linear System)

Computing the inverse of a nonsingular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ by n linear systems needs $\sim (2/3)n^3 + n(2n^2) = (8/3)n^3$ flops where $(2/3)n^3$ comes from the computation of the LU decomposition of \mathbf{A} .

The proof is trivial by using Algorithm 5.

However, the complexity can be reduced by taking the advantage of the structures of \mathbf{U}, \mathbf{L} . We find that the inverse of the nonsingular matrix is $\mathbf{A}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}^T$.

Theorem 1.2: (Inverse of Nonsingular Matrix by LU Factorization)

Computing the inverse of a nonsingular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ by $\mathbf{A}^{-1} = \mathbf{U}^{-1} \mathbf{L}^{-1} \mathbf{P}^T$ needs $\sim (2/3)n^3 + (4/3)n^3 = 2n^3$ flops where $(2/3)n^3$ comes from the computation of the LU decomposition of \mathbf{A} .

Proof [of Theorem 1.2] We notice the computation of $\mathbf{U}^{-1} \mathbf{L}^{-1} \mathbf{P}^T$ comes from $\mathbf{Y} = \mathbf{U}^{-1} \mathbf{L}^{-1}$. And \mathbf{U}^{-1} is an upper triangular, \mathbf{L}^{-1} is a unit lower triangular matrix. Suppose

$$\mathbf{U}^{-1} = \mathbf{Z} = \begin{bmatrix} -\mathbf{z}_1^\top & - \\ -\mathbf{z}_2^\top & - \\ \vdots & \\ -\mathbf{z}_n^\top & - \end{bmatrix} \quad \text{and} \quad \mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n],$$

are the row partitions and column partitions of \mathbf{U}^{-1} and \mathbf{U} respectively. Since both \mathbf{Z} and \mathbf{U} are upper triangular matrices. Thus we have

$$\mathbf{I} = \begin{bmatrix} \mathbf{z}_1^\top \mathbf{u}_1 = 1 & \mathbf{z}_1^\top \mathbf{u}_2 = 0 & \mathbf{z}_1^\top \mathbf{u}_3 = 0 & \mathbf{z}_1^\top \mathbf{u}_4 = 0 & \dots & \mathbf{z}_1^\top \mathbf{u}_n = 0 \\ \mathbf{z}_2^\top \mathbf{u}_1 = 0 & \mathbf{z}_2^\top \mathbf{u}_2 = 1 & \mathbf{z}_2^\top \mathbf{u}_3 = 0 & \mathbf{z}_2^\top \mathbf{u}_4 = 0 & \dots & \mathbf{z}_2^\top \mathbf{u}_n = 0 \\ \mathbf{z}_3^\top \mathbf{u}_1 = 0 & \mathbf{z}_3^\top \mathbf{u}_2 = 0 & \mathbf{z}_3^\top \mathbf{u}_3 = 1 & \mathbf{z}_3^\top \mathbf{u}_4 = 0 & \dots & \mathbf{z}_3^\top \mathbf{u}_n = 0 \\ \mathbf{z}_4^\top \mathbf{u}_1 = 0 & \mathbf{z}_4^\top \mathbf{u}_2 = 0 & \mathbf{z}_4^\top \mathbf{u}_3 = 0 & \mathbf{z}_4^\top \mathbf{u}_4 = 1 & \dots & \mathbf{z}_4^\top \mathbf{u}_n = 0 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ \mathbf{z}_n^\top \mathbf{u}_1 = 0 & \mathbf{z}_n^\top \mathbf{u}_2 = 0 & \mathbf{z}_n^\top \mathbf{u}_3 = 0 & \mathbf{z}_n^\top \mathbf{u}_4 = 0 & \dots & \mathbf{z}_n^\top \mathbf{u}_n = 1 \end{bmatrix}.$$

By $\mathbf{z}_1^\top \mathbf{u}_1 = 1$, we can compute the first component of \mathbf{z}_1 with 1 flop; By $\mathbf{z}_1^\top \mathbf{u}_2 = 0$, we can compute the second component of \mathbf{z}_1 with 3 flops as the first component is already calculated; Then we list the complexity of each inner product in an $n \times n$ matrix, where each entry (i, j) denotes the cost to calculate the (i, j) element of \mathbf{U}^{-1} :

$$\text{cost} = \begin{bmatrix} \mathbf{z}_1^\top \mathbf{u}_1 = 1 & \mathbf{z}_1^\top \mathbf{u}_2 = 3 & \mathbf{z}_1^\top \mathbf{u}_3 = 5 & \mathbf{z}_1^\top \mathbf{u}_4 = 7 & \dots & \mathbf{z}_1^\top \mathbf{u}_n = 2n - 1 \\ 0 & \mathbf{z}_2^\top \mathbf{u}_2 = 1 & \mathbf{z}_2^\top \mathbf{u}_3 = 3 & \mathbf{z}_2^\top \mathbf{u}_4 = 5 & \dots & \mathbf{z}_2^\top \mathbf{u}_n = 2n - 3 \\ 0 & 0 & \mathbf{z}_3^\top \mathbf{u}_3 = 1 & \mathbf{z}_3^\top \mathbf{u}_4 = 3 & \dots & \mathbf{z}_3^\top \mathbf{u}_n = 2n - 5 \\ 0 & 0 & 0 & \mathbf{z}_4^\top \mathbf{u}_4 = 1 & \dots & \mathbf{z}_4^\top \mathbf{u}_n = 2n - 7 \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & 0 & \dots & \mathbf{z}_n^\top \mathbf{u}_n = 1 \end{bmatrix} = \begin{bmatrix} n^2 \\ (n-1)^2 \\ (n-2)^2 \\ (n-3)^2 \\ \vdots \\ 1 \end{bmatrix},$$

which is a sum of n sets of arithmetic sequences and the sum of each sequence is shown in the last equality above. Thus the total cost to compute \mathbf{U}^{-1} is $\boxed{\frac{2n^3+3n^2+n}{6}}$ flops. Similarly, to calculate \mathbf{L}^{-1} also costs $\boxed{\frac{2n^3+3n^2+n}{6}}$ flops.

A moment of reflexion on the multiplication of $\mathbf{Y} = \mathbf{U}^{-1} \mathbf{L}^{-1}$ would reveal that:

- The entry $(1, 1)$ of \mathbf{Y}_{11} involves the computation of an inner product with n -dimension which takes $2n - 1$ flops (i.e., n multiplications and $n - 1$ additions).
- The entry $(1, 2)$ of \mathbf{Y}_{12} involves the computation of an inner product with $(n - 1)$ -dimension which takes $2n - 3$ flops.

- The process can go on, we write the flops in an $n \times n$ matrix with each entry (i, j) meaning the number of flops to calculate the (i, j) element of \mathbf{Y} :

$$\text{costs of } \mathbf{Y} = \mathbf{U}^{-1} \mathbf{L}^{-1} = \begin{bmatrix} \{2n-1 & 2n-3 & 2n-5 & 2n-7 & \dots & 1\} \\ \underbrace{2n-3}_{\color{cyan}} & \{2n-3 & 2n-5 & 2n-7 & \dots & 1\} \\ 2n-5 & \underbrace{2n-5}_{\color{magenta}} & \{2n-5 & 2n-7 & \dots & 1\} \\ 2n-7 & 2n-7 & \underbrace{2n-7}_{\color{orange}} & \{2n-7 & \dots & 1\} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ \underbrace{1}_{\color{blue}} & \underbrace{1}_{\color{pink}} & \underbrace{1}_{\color{yellow}} & \underbrace{1}_{\color{red}} & \vdots & \color{teal}1 \end{bmatrix},$$

which is a symmetric matrix and the total complexity can be calculated by the sum of several arithmetic sequences where each sequence is denoted in a different color. To make it clearer, each arithmetic sequence is also denoted by a brace in the above matrix. Simple calculations will show the complexity is $\boxed{(2/3)n^3 + (1/3)n}$.

As a result, the total cost is then $(2/3)n^3 + (1/3)n^3 + (1/3)n^3 + (2/3)n^3 = 2n^3$ flops if we keep only the leading term, where the first $(2/3)n^3$ comes from the computation of the LU decomposition of \mathbf{A} . ■

1.12. Application: Computing the Determinant via the LU Decomposition

We can find the determinant easily by using the LU decomposition. If $\mathbf{A} = \mathbf{LU}$, then $\det(\mathbf{A}) = \det(\mathbf{LU}) = \det(\mathbf{L})\det(\mathbf{U}) = \mathbf{U}_{11}\mathbf{U}_{22}\dots\mathbf{U}_{nn}$ where \mathbf{U}_{ii} is the i -th diagonal of \mathbf{U} for $i \in \{1, 2, \dots, n\}$.⁵

Further, for the LU decomposition with permutation $\mathbf{A} = \mathbf{PLU}$, $\det(\mathbf{A}) = \det(\mathbf{PLU}) = \det(\mathbf{P})\det(\mathbf{U}_{11})\det(\mathbf{U}_{22})\dots\det(\mathbf{U}_{nn})$. The determinant of a permutation matrix is either 1 or -1 because after changing rows around (which changes the sign of the determinant⁶) a permutation matrix becomes identity matrix \mathbf{I} , whose determinant is one.

1.13. Pivoting

1.13.1 Partial Pivoting

In practice, it is desirable to pivot even when it is not necessary. When dealing with a linear system via the LU decomposition as shown in Algorithm 5, if the diagonal entries of \mathbf{U} are small, it can lead to inaccurate solutions for the linear solution. Thus, it is common to pick the largest entry to be the pivot. This is known as the *partial pivoting*. For example,

-
- 5. The determinant of a lower triangular matrix (or an upper triangular matrix) is the product of the diagonal entries.
 - 6. The determinant changes sign when two rows are exchanged (sign reversal).

Partial Pivoting For a 4×4 Matrix

$$\begin{array}{c}
 \left[\begin{array}{cccc} \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{E}_1} \left[\begin{array}{cccc} \square & \square & \square & \square \\ 0 & \mathbf{2} & \square & \square \\ 0 & \mathbf{5} & \square & \square \\ 0 & \mathbf{7} & \square & \square \end{array} \right] \xrightarrow{\mathbf{P}_1} \left[\begin{array}{cccc} \square & \square & \square & \square \\ 0 & \mathbf{7} & \square & \square \\ 0 & 5 & \square & \square \\ 0 & 2 & \square & \square \end{array} \right] \xrightarrow{\mathbf{E}_2} \left[\begin{array}{cccc} \square & \square & \square & \square \\ 0 & 7 & \square & \square \\ 0 & \mathbf{0} & \square & \square \\ 0 & 0 & 0 & \square \end{array} \right], \\
 \mathbf{A} \qquad \qquad \qquad \mathbf{E}_1\mathbf{A} \qquad \qquad \qquad \mathbf{P}_1\mathbf{E}_1\mathbf{A} \qquad \qquad \qquad \mathbf{E}_2\mathbf{P}_1\mathbf{E}_1\mathbf{A}
 \end{array} \tag{1.3}$$

in which case, we pick 7 as the pivot after transformation by \mathbf{E}_1 even when it is not necessary. This interchange permutation can guarantee that no multiplier is greater than 1 in absolute value during the Gaussian elimination. A specific example is provided as follows.

Example 1.1 (Partial Pivoting) Suppose

$$\mathbf{A} = \begin{bmatrix} 2 & 10 & 5 \\ 1 & 4 & -2 \\ 6 & 8 & 4 \end{bmatrix}$$

To get the smallest possible multipliers in the first Gaussian transformation, we need to interchange the largest value in the first column to entry (1,1). The permutation matrix \mathbf{P}_1 is doing so

$$\mathbf{P}_1 = \begin{bmatrix} & & 1 \\ & 1 & \\ 1 & & \end{bmatrix}, \quad \text{such that} \quad \mathbf{U} = \mathbf{P}_1\mathbf{A} = \begin{bmatrix} 6 & 8 & 4 \\ 1 & 4 & -2 \\ 2 & 10 & 5 \end{bmatrix}.$$

Then, it follows that \mathbf{E}_1 will introduce zero below the entry (1,1),

$$\mathbf{E}_1 = \begin{bmatrix} 1 & & \\ -1/6 & 1 & \\ -1/3 & 0 & 1 \end{bmatrix}, \quad \text{and} \quad \mathbf{U} = \mathbf{E}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 6 & 8 & 4 \\ 0 & 8/3 & -8/3 \\ 0 & 22/3 & 11/3 \end{bmatrix}$$

Now, we pivot $22/3$ before $8/3$, and the permutation is given by

$$\mathbf{P}_2 = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}, \quad \text{such that} \quad \mathbf{U} = \mathbf{P}_2\mathbf{E}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 6 & 8 & 4 \\ 0 & 22/3 & 11/3 \\ 0 & 8/3 & -8/3 \end{bmatrix}.$$

Finally, the Gaussian transformation to introduce zero below entry (2,2) is given by

$$\mathbf{E}_2 = \begin{bmatrix} 1 & & \\ & 1 & \\ & -4/11 & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{U} = \mathbf{E}_2\mathbf{P}_2\mathbf{E}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 6 & 8 & 4 \\ 0 & 22/3 & 11/3 \\ 0 & 0 & -4 \end{bmatrix}.$$

And output the final \mathbf{U} . □

As discussed above, the Gaussian transformation \mathbf{E}_k in k -th step of the partial pivoting is given by

$$\mathbf{E}_k = \mathbf{I} - \mathbf{z}_k \mathbf{e}_k^\top,$$

where $\mathbf{e}_k \in \mathbb{R}^n$ is the k -th unit vector, and $\mathbf{z}_k \in \mathbb{R}^n$ is given by

$$\mathbf{z}_k = [0, \dots, 0, z_{k+1}, \dots, z_n]^\top, \quad z_i = \frac{\mathbf{U}_{ik}}{\mathbf{U}_{kk}}, \forall i \in \{k+1, \dots, n\}.$$

We realize that \mathbf{E}_k is a unit lower triangular matrix with only the k -th column of the lower submatrix being nonzero,

$$\mathbf{E}_k = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & 0 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & 1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boxtimes & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boxtimes & \mathbf{0} & \mathbf{I} \end{bmatrix}_k.$$

More generally, the procedure for computing the LU decomposition with partial pivoting of $\mathbf{A} \in \mathbb{R}^{n \times n}$ is given in Algorithm 6.

Algorithm 6 LU Decomposition with Partial Pivoting

Require: Matrix \mathbf{A} with size $n \times n$;

- 1: Let $\mathbf{U} = \mathbf{A}$;
 - 2: **for** $k = 1$ to $n - 1$ **do** ▷ i.e., get the k -th column of \mathbf{U}
 - 3: Find a row permutation matrix \mathbf{P}_k that swaps \mathbf{U}_{kk} with the largest element in $|\mathbf{U}_{k:n,k}|$;
 - 4: $\mathbf{U} = \mathbf{P}_k \mathbf{U}$;
 - 5: Determine the Gaussian transformation \mathbf{E}_k to introduce zeros below the diagonal of the k -th column of \mathbf{U} ;
 - 6: $\mathbf{U} = \mathbf{E}_k \mathbf{U}$;
 - 7: **end for**
 - 8: Output \mathbf{U} ;
-

The algorithm requires $2/3(n^3)$ flops and $(n-1) + (n-2) + \dots + 1 \sim O(n^2)$ comparisons resulting from the pivoting procedure. Upon completion, the upper triangular matrix \mathbf{U} is given by

$$\mathbf{U} = \mathbf{E}_{n-1} \mathbf{P}_{n-1} \dots \mathbf{E}_2 \mathbf{P}_2 \mathbf{E}_1 \mathbf{P}_1 \mathbf{A}.$$

Computing the final \mathbf{L} And we here show that Algorithm 6 computes the LU decomposition in the following form

$$\mathbf{A} = \mathbf{PLU},$$

where $\mathbf{P} = \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_{n-1}$ takes account of all the interchanges, \mathbf{U} is the upper triangular matrix results directly from the algorithm, \mathbf{L} is unit lower triangular with $|\mathbf{L}_{ij}| \leq 1$ for all $1 \leq i, j \leq n$. $\mathbf{L}_{k+1:n,k}$ is a permuted version of \mathbf{E}_k 's multipliers. To see this, we notice that

the permutation matrices used in the algorithm fall into a special kind of permutation matrix since we only interchange two rows of the matrix. *This implies the \mathbf{P}_k 's are symmetric and $\mathbf{P}_k^2 = \mathbf{I}$ for $k \in \{1, 2, \dots, n-1\}$.* Suppose

$$\mathbf{M}_k = (\mathbf{P}_{n-1} \dots \mathbf{P}_{k+1}) \mathbf{E}_k (\mathbf{P}_{k+1} \dots \mathbf{P}_{n-1}).$$

Then, \mathbf{U} can be written as

$$\mathbf{U} = \mathbf{M}_{n-1} \dots \mathbf{M}_2 \mathbf{M}_1 \mathbf{P}^\top \mathbf{A}.$$

To see what \mathbf{M}_k is, we realize that \mathbf{P}_{k+1} is a permutation with the upper left $k \times k$ block being an identity matrix. And thus we have

$$\begin{aligned} \mathbf{M}_k &= (\mathbf{P}_{n-1} \dots \mathbf{P}_{k+1})(\mathbf{I}_n - \mathbf{z}_k \mathbf{e}_k^\top)(\mathbf{P}_{k+1} \dots \mathbf{P}_{n-1}) \\ &= \mathbf{I}_n - (\mathbf{P}_{n-1} \dots \mathbf{P}_{k+1})(\mathbf{z}_k \mathbf{e}_k^\top)(\mathbf{P}_{k+1} \dots \mathbf{P}_{n-1}) \\ &= \mathbf{I}_n - (\mathbf{P}_{n-1} \dots \mathbf{P}_{k+1} \mathbf{z}_k)(\mathbf{e}_k^\top \mathbf{P}_{k+1} \dots \mathbf{P}_{n-1}) \\ &= \mathbf{I}_n - (\mathbf{P}_{n-1} \dots \mathbf{P}_{k+1} \mathbf{z}_k) \mathbf{e}_k^\top. \quad (\text{since } \mathbf{e}_k^\top \mathbf{P}_{k+1} \dots \mathbf{P}_{n-1} = \mathbf{e}_k^\top) \end{aligned}$$

This implies that \mathbf{M}_k is unit lower triangular with the k -th column being the permuted version of \mathbf{E}_k . And the final lower triangular \mathbf{L} is thus given by

$$\mathbf{L} = \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \dots \mathbf{M}_{n-1}^{-1}.$$

1.13.2 Complete Pivoting

In partial pivoting, when introducing zeros below the diagonal of the k -th column of \mathbf{U} , the k -th pivot is determined by scanning the current subcolumn $\mathbf{U}_{k:n,k}$. In complete pivoting, the largest absolute entry in the current submatrix $\mathbf{U}_{k:n,k:n}$ is interchanged into the entry (k, k) of \mathbf{U} . Therefore, an additional *column permutation* \mathbf{Q}_k is needed in each step. The final upper triangular matrix \mathbf{U} is obtained by

$$\mathbf{U} = \mathbf{E}_{n-1} \mathbf{P}_{n-1} \dots (\mathbf{E}_2 \mathbf{P}_2 (\mathbf{E}_1 \mathbf{P}_1 \mathbf{A} \mathbf{Q}_1) \mathbf{Q}_2) \dots \mathbf{Q}_{n-1}.$$

Similarly, the complete pivoting algorithm is formulated in Algorithm 7.

Algorithm 7 LU Decomposition with Complete Pivoting

Require: Matrix \mathbf{A} with size $n \times n$;

- 1: Let $\mathbf{U} = \mathbf{A}$;
 - 2: **for** $k = 1$ to $n-1$ **do** ▷ the value k is to get the k -th column of \mathbf{U}
 - 3: Find a row permutation matrix \mathbf{P}_k , and a column permutation \mathbf{Q}_k that swaps \mathbf{U}_{kk} with the largest element in $|\mathbf{U}_{k:n,k:n}|$, say $\mathbf{U}_{u,v} = \max |\mathbf{U}_{k:n,k:n}|$;
 - 4: $\mathbf{U} = \mathbf{P}_k \mathbf{U} \mathbf{Q}_k$;
 - 5: Determine the Gaussian transformation \mathbf{E}_k to introduce zeros below the diagonal of the k -th column of \mathbf{U} ;
 - 6: $\mathbf{U} = \mathbf{E}_k \mathbf{U}$;
 - 7: **end for**
 - 8: Output \mathbf{U} ;
-

The algorithm requires $2/3(n^3)$ flops and $(n^2 + (n-1)^2 + \dots + 1^2) \sim O(n^3)$ comparisons resulting from the pivoting procedure. Again, let $\mathbf{P} = \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_{n-1}$, $\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_{n-1}$,

$$\mathbf{M}_k = (\mathbf{P}_{n-1} \dots \mathbf{P}_{k+1}) \mathbf{E}_k (\mathbf{P}_{k+1} \dots \mathbf{P}_{n-1}), \quad \text{for all } k \in \{1, 2, \dots, n-1\}$$

and

$$\mathbf{L} = \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \dots \mathbf{M}_{n-1}^{-1}.$$

We have $\mathbf{A} = \mathbf{PLUQ}^\top$ or $\mathbf{P}^\top \mathbf{AQ} = \mathbf{LU}$ as the final decomposition.

1.13.3 Rook Pivoting

The *rook pivoting* provides an alternative to the partial and complete pivoting. Instead of choosing the largest value in $|\mathbf{U}_{k:n,k:n}|$ in the k -th step, it searches for an element of $\mathbf{U}_{k:n,k:n}$ that is maximal in both its row and column. Apparently, the rook pivoting is not unique such that we could find many entries that satisfy the criteria. For example, for a submatrix $\mathbf{U}_{k:n,k:n}$ as follows

$$\mathbf{U}_{k:n,k:n} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 7 & 3 \\ 5 & 2 & 1 & 2 \\ 2 & 1 & 2 & 1 \end{bmatrix},$$

where the 7 will be chosen by complete pivoting. And one of 5, 4, 7 will be identified as a rook pivot.

1.14. Rank-Revealing LU Decomposition

In many applications, a factorization produced by Gaussian elimination with pivoting when \mathbf{A} has rank r will reveal rank in the following form

$$\mathbf{PAQ} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21}^\top & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\mathbf{L}_{11} \in \mathbb{R}^{r \times r}$ and $\mathbf{U}_{11} \in \mathbb{R}^{r \times r}$ are nonsingular, $\mathbf{L}_{21}, \mathbf{U}_{21} \in \mathbb{R}^{r \times (n-r)}$, and \mathbf{P}, \mathbf{Q} are permutations. Gaussian elimination with rook pivoting or complete pivoting can result in such decomposition (Hwang et al., 1992; Higham, 2002).

1.15. Rate of Change of L and U*

If \mathbf{A} has a unique LU decomposition, and a small perturbation $\Delta\mathbf{A}$ such that the LU decomposition of $\mathbf{A} + \Delta\mathbf{A}$ also exists and is given by $\mathbf{A} + \Delta\mathbf{A} = (\mathbf{L} + \Delta\mathbf{L})(\mathbf{U} + \Delta\mathbf{U})$. Then $\Delta\mathbf{A}$ can be obtained by

$$\Delta\mathbf{A} = \Delta\mathbf{L} \cdot \mathbf{U} + \mathbf{L} \cdot \Delta\mathbf{U}.$$

And we have

$$\mathbf{L}^{-1} \cdot \Delta\mathbf{A} \cdot \mathbf{U}^{-1} = \mathbf{L}^{-1} \cdot \Delta\mathbf{L} + \Delta\mathbf{U} \cdot \mathbf{U}^{-1}.$$

Since both \mathbf{L} and $(\mathbf{L} + \Delta\mathbf{L})$ are unit lower triangular matrices, $\Delta\mathbf{L}$ is strictly lower triangular, i.e., lower triangular with 0's on the diagonal. And both \mathbf{U} and $\mathbf{U} + \Delta\mathbf{U}$ are upper

triangular matrices, $\Delta\mathbf{U}$ is therefore upper triangular. Thus

$$\Delta\mathbf{L} = \mathbf{L} \cdot \text{slt}(\mathbf{L}^{-1} \cdot \Delta\mathbf{A} \cdot \mathbf{U}^{-1}), \quad \Delta\mathbf{U} = \text{ut}(\mathbf{L}^{-1} \cdot \Delta\mathbf{A} \cdot \mathbf{U}^{-1})\mathbf{U},$$

where $\text{slt}(\mathbf{B})$ is the strictly lower triangular part of \mathbf{B} , and $\text{ut}(\mathbf{B})$ is the upper triangular part of \mathbf{B} . Clearly, the sensitivity, aka, the rate of change of \mathbf{L} and \mathbf{U} depends on the inverse of \mathbf{L} and \mathbf{U} . More generally, we have the following result.

Theorem 1.1: (Rate of Change of \mathbf{L} and \mathbf{U})

Suppose matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ has nonzero leading principal minors, i.e., $\det(\mathbf{A}_{1:k, 1:k}) \neq 0$, for all $k \in \{1, 2, \dots, n\}$, with LU decomposition $\mathbf{A} = \mathbf{LU}$. Let \mathbf{G} be a real $n \times n$ matrix and let $\Delta\mathbf{A} = \epsilon\mathbf{G}$, for some $\epsilon \geq 0$. If ϵ is sufficiently small such that all the leading principal minors of $\mathbf{A} + t\mathbf{G}$ are nonzero for all $|t| \leq \epsilon$. Then $\mathbf{A} + \Delta\mathbf{A}$ has the LU decomposition

$$\mathbf{A} + \Delta\mathbf{A} = (\mathbf{L} + \Delta\mathbf{L})(\mathbf{U} + \Delta\mathbf{U}),$$

with $\Delta\mathbf{L}$ and $\Delta\mathbf{U}$ satisfying

$$\begin{aligned}\Delta\mathbf{L} &= \epsilon\dot{\mathbf{L}}(0) + O(\epsilon^2), \\ \Delta\mathbf{U} &= \epsilon\dot{\mathbf{U}}(0) + O(\epsilon^2),\end{aligned}$$

where $\dot{\mathbf{L}}(0) = \dot{\mathbf{L}}(t)|_{t=0}$ and $\dot{\mathbf{U}}(0) = \dot{\mathbf{U}}(t)|_{t=0}$, and $\dot{\mathbf{L}}(t)$ and $\dot{\mathbf{U}}(t)$ are defined by the unique LU decomposition

$$\mathbf{A} + t\mathbf{G} = \mathbf{L}(t)\mathbf{U}(t), \quad |t| \leq \epsilon. \quad (1.4)$$

Proof [of Theorem 1.1] We notice that

$$\begin{aligned}\mathbf{L}(t) &= \mathbf{L} + \Delta\mathbf{L}, \\ \dot{\mathbf{L}}(t) &= \frac{\partial \mathbf{L}(t)}{\partial t}.\end{aligned} \quad (1.5)$$

By Taylor expansion, we have

$$\dot{\mathbf{L}}(\epsilon) = \frac{\partial \mathbf{L}(\epsilon)}{\partial \epsilon} \approx \dot{\mathbf{L}}(0) + \ddot{\mathbf{L}}(\epsilon)\epsilon + O(\epsilon^2).$$

Since ϵ is small enough, such that Equation (1.5) can be written as

$$\dot{\mathbf{L}}(\epsilon) = \frac{\partial \mathbf{L}(\epsilon)}{\partial \epsilon} = \frac{\Delta\mathbf{L}}{\epsilon},$$

which results in

$$\Delta\mathbf{L} = \epsilon\dot{\mathbf{L}}(0) + O(\epsilon^2).$$

Similarly, we can also obtain

$$\Delta\mathbf{U} = \epsilon\dot{\mathbf{U}}(0) + O(\epsilon^2).$$

■

This completes the proof.

Following from the Theorem 1.1 above, we can also claim that

$$\mathbf{L}\dot{\mathbf{U}}(0) + \dot{\mathbf{L}}(0)\mathbf{U} = \mathbf{G}, \quad (1.6)$$

$$\dot{\mathbf{L}}(0) = \mathbf{L} \cdot \text{slt}(\mathbf{L}^{-1}\mathbf{G}\mathbf{U}^{-1}), \quad (1.7)$$

$$\dot{\mathbf{U}}(0) = \mathbf{U}t(\mathbf{L}^{-1}\mathbf{G}\mathbf{U}^{-1})\mathbf{U}. \quad (1.8)$$

Note that $\mathbf{L}(0) = \mathbf{L}$, $\mathbf{L}(\epsilon) = \mathbf{L} + \Delta\mathbf{L}$, $\mathbf{U}(0) = \mathbf{U}$, $\mathbf{U}(\epsilon) = \mathbf{U} + \Delta\mathbf{U}$. If we differentiate Equation (1.4), and set $t = 0$, we can obtain result in Equation (1.6). Further, multiply Equation (1.6) left by \mathbf{L}^{-1} and right by \mathbf{U}^{-1} , we have

$$\dot{\mathbf{U}}(0)\mathbf{U}^{-1} + \mathbf{L}^{-1}\dot{\mathbf{L}}(0) = \mathbf{L}^{-1}\mathbf{G}\mathbf{U}^{-1},$$

where $\dot{\mathbf{L}}(0)$ is strictly lower triangular and $\dot{\mathbf{U}}(0)$ is upper triangular. Thus the results in Equation (1.7) and (1.8) can be proved.

In the above theorem, we proved that the sensitivity of $\Delta\mathbf{L}$ and $\Delta\mathbf{U}$ is bounded by $\dot{\mathbf{L}}(0)$ and $\dot{\mathbf{U}}(0)$ to some ϵ value. More results of the perturbation on the LU decomposition can be found in (Baarland, 1991; Sun, 1992b,a; Stewart, 1993, 1997; Chang, 1997; Chang and Paige, 1998; Higham, 2002; Bueno and Dopico, 2004), similar results for the QR decomposition (Stewart, 1993; Chang et al., 1997; Chang, 1997), and the Cholesky decomposition (Sun, 1992b; Stewart, 1997; Chang, 1997).

Chapter 2

Cholesky Decomposition

Contents

2.1	Cholesky Decomposition	56
2.2	Existence of the Cholesky Decomposition via Recursive Calculation	56
2.3	Sylvester's Criterion: Leading Principal Minors of PD Matrices	60
2.4	Existence of the Cholesky Decomposition via the LU Decomposition without Permutation	63
2.4.1	Diagonal Values of the Upper Triangular Matrix	63
2.4.2	Block Cholesky Decomposition	64
2.5	Existence of the Cholesky Decomposition via Induction	65
2.6	Uniqueness of the Cholesky Decomposition	66
2.7	Computing the Cholesky Decomposition Recursively	67
2.8	Computing the Cholesky Decomposition Element-Wise	68
2.9	More Properties of Positive Definite Matrices	69
2.10	Last Words on Positive Definite Matrices	70
2.11	Pivoted Cholesky Decomposition	71
2.12	Decomposition for Semidefinite Matrices	71
2.13	Application: Rank-One Update/Downdate	73
2.13.1	Rank-One Update	73
2.13.2	Rank-One Downdate	75
2.14	Application: Indefinite Rank Two Update	76

2.1. Cholesky Decomposition

Positive definiteness or positive semidefiniteness is one of the highest accolades to which a matrix can aspire. In this section, we will introduce decompositional approaches for the two special kinds of matrices and we first illustrate the most famous Cholesky decomposition as follows.

Theorem 2.1: (Cholesky Decomposition)

Every positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ can be factored as

$$\mathbf{A} = \mathbf{R}^\top \mathbf{R},$$

where $\mathbf{R} \in \mathbb{R}^{n \times n}$ is an upper triangular matrix **with positive diagonal elements**. This decomposition is known as the **Cholesky decomposition** of \mathbf{A} . \mathbf{R} is known as the **Cholesky factor** or **Cholesky triangle** of \mathbf{A} .

Alternatively, \mathbf{A} can be factored as $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$ where $\mathbf{L} = \mathbf{R}^\top$ is a lower triangular matrix **with positive diagonals**.

Specifically, the Cholesky decomposition is unique (Corollary 2.1, p. 66).

The Cholesky decomposition is named after a French military officer and mathematician, André-Louis Cholesky (1875-1918), who developed the Cholesky decomposition in his surveying work. Similar to the LU decomposition for solving linear systems, the Cholesky decomposition is further used primarily to solve positive definite linear systems. The development on the solution is similar to that of the LU decomposition in Section 1.10 (p. 46), and we shall not repeat the details.

2.2. Existence of the Cholesky Decomposition via Recursive Calculation

In this section, we will prove the existence of the Cholesky decomposition via recursive calculation. In Section 13.6.4 (p. 261), we will also prove the existence of the Cholesky decomposition via the QR decomposition and spectral decomposition.

Before showing the existence of Cholesky decomposition, we need the following definitions and lemmas.

Definition 2.1: Positive Definite and Positive Semidefinite

A matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite (PD) if $\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0$ for all nonzero $\mathbf{x} \in \mathbb{R}^n$. And a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive semidefinite (PSD) if $\mathbf{x}^\top \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$.

One of the prerequisites for the Cholesky decomposition is the definition of the above positive definiteness of a matrix. We sketch several properties of this PD matrix as follows:

Positive Definite Matrix Property 1 of 6

We will show the equivalent definition on the positive definiteness of a matrix \mathbf{A} is that \mathbf{A} only has positive eigenvalues, or on the positive semidefiniteness of a matrix \mathbf{A} is that \mathbf{A} only has nonnegative eigenvalues. The proof is provided in Section 13.6.2 (p. 259) as a consequence of the spectral theorem.

Positive Definite Matrix Property 2 of 6

Lemma 2.2: (Positive Diagonals of Positive Definite Matrices)

The diagonal elements of a positive definite matrix \mathbf{A} are all **positive**. And similarly, the diagonal elements of a positive semidefinite matrix \mathbf{B} are all **non-negative**.

Proof [of Lemma 2.2] From the definition of positive definite matrices, we have $\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0$ for all nonzero \mathbf{x} . In particular, let $\mathbf{x} = \mathbf{e}_i$ where \mathbf{e}_i is the i -th unit vector with the i -th entry being equal to 1 and other entries being equal to 0. Then,

$$\mathbf{e}_i^\top \mathbf{A} \mathbf{e}_i = a_{ii} > 0, \quad \forall i \in \{1, 2, \dots, n\},$$

where a_{ii} is the i -th diagonal component. The proof for the second part follows similarly. This completes the proof. ■

The complete pivoting in the Cholesky decomposition is simpler than that in the LU decomposition in the sense that the maximal value in a positive definite matrix is on the diagonal

Positive Definite Matrix Property 3 of 6

Lemma 2.3: (Maximal Value of Positive Definite Matrices)

The maximum element in a positive definite matrix lies on the diagonal. And this argument works similarly to positive semidefinite matrices.

Proof [of Lemma 2.3] Suppose $\mathbf{e}_i, \mathbf{e}_j$ are the i -th and j -th unit vector, and a_{ij} being the entry (i, j) of the positive definite matrix \mathbf{A} . Then, it follows that

$$(\mathbf{e}_i - \mathbf{e}_j)^\top \mathbf{A} (\mathbf{e}_i - \mathbf{e}_j) = a_{ii} + a_{jj} - 2a_{ij} > 0.$$

Therefore, either a_{ii} or a_{jj} is larger than a_{ij} . If we loop around all the entries, the result follows. When it comes to the positive semidefinite matrices, it follows that the largest element appear on the diagonal, with possibility that some non-diagonal elements are equal to the largest element. ■

Positive Definite Matrix Property 4 of 6

Lemma 2.4: (Schur Complement of Positive Definite Matrices)

For any positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, its Schur complement of \mathbf{A}_{11} is given by $\mathbf{S}_{n-1} = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{2:n,1}^\top$ which is also positive definite.

A word on the notation Note that the subscript $n-1$ of \mathbf{S}_{n-1} means it is of size $(n-1) \times (n-1)$ and it is a Schur complement of an $n \times n$ positive definite matrix. We will use this notation in the following sections.

Proof [of Lemma 2.4] For any nonzero vector $\mathbf{v} \in \mathbb{R}^{n-1}$, we can construct a vector $\mathbf{x} \in \mathbb{R}^n$ by the following equation:

$$\mathbf{x} = \begin{bmatrix} -\frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1}^\top \mathbf{v} \\ \mathbf{v} \end{bmatrix},$$

which is nonzero. Then

$$\begin{aligned} \mathbf{x}^\top \mathbf{A} \mathbf{x} &= \begin{bmatrix} -\frac{1}{\mathbf{A}_{11}} \mathbf{v}^\top \mathbf{A}_{2:n,1} & \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{2:n,1}^\top \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} \begin{bmatrix} -\frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1}^\top \mathbf{v} \\ \mathbf{v} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{\mathbf{A}_{11}} \mathbf{v}^\top \mathbf{A}_{2:n,1} & \mathbf{v}^\top \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{S}_{n-1} \mathbf{v} \end{bmatrix} \\ &= \mathbf{v}^\top \mathbf{S}_{n-1} \mathbf{v}. \end{aligned}$$

Since \mathbf{A} is positive definite, we have $\mathbf{x}^\top \mathbf{A} \mathbf{x} = \mathbf{v}^\top \mathbf{S}_{n-1} \mathbf{v} > 0$ for all nonzero \mathbf{v} . Thus, the Schur complement \mathbf{S}_{n-1} is positive definite as well. \blacksquare

The above argument can be extended to PSD matrices as well. If \mathbf{A} is PSD, then the Schur complement \mathbf{S}_{n-1} is also PSD.

A word on the Schur complement In the proof of Theorem 1.1, we have shown this Schur complement $\mathbf{S}_{n-1} = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{2:n,1}^\top$ is also nonsingular if \mathbf{A} is nonsingular and $\mathbf{A}_{11} \neq 0$. Similarly, the Schur complement of \mathbf{A}_{nn} in \mathbf{A} is $\mathbf{S}_{n-1} = \mathbf{A}_{1:n-1,1:n-1} - \frac{1}{\mathbf{A}_{nn}} \mathbf{A}_{1:n-1,n} \mathbf{A}_{1:n-1,n}^\top$ which is also positive definite if \mathbf{A} is positive definite. This property can help prove the fact that the leading principal minors of positive definite matrices are all positive. See Section 2.3 for more details.

We then prove the existence of Cholesky decomposition using these lemmas.

Proof [of Theorem 2.1: Existence of Cholesky Decomposition Recursively] For any positive definite matrix \mathbf{A} , we can write out (since \mathbf{A}_{11} is positive by Lemma 2.2)

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{2:n,1}^\top \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{\mathbf{A}_{11}} & \mathbf{0} \\ \frac{1}{\sqrt{\mathbf{A}_{11}}} \mathbf{A}_{2:n,1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \sqrt{\mathbf{A}_{11}} & \frac{1}{\sqrt{\mathbf{A}_{11}}} \mathbf{A}_{2:n,1}^\top \\ \mathbf{0} & \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{2:n,1}^\top \end{bmatrix} \\ &= \begin{bmatrix} \sqrt{\mathbf{A}_{11}} & \mathbf{0} \\ \frac{1}{\sqrt{\mathbf{A}_{11}}} \mathbf{A}_{2:n,1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{2:n,1}^\top \end{bmatrix} \begin{bmatrix} \sqrt{\mathbf{A}_{11}} & \frac{1}{\sqrt{\mathbf{A}_{11}}} \mathbf{A}_{2:n,1}^\top \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ &= \mathbf{R}_1^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{n-1} \end{bmatrix} \mathbf{R}_1.\end{aligned}$$

where

$$\mathbf{R}_1 = \begin{bmatrix} \sqrt{\mathbf{A}_{11}} & \frac{1}{\sqrt{\mathbf{A}_{11}}} \mathbf{A}_{2:n,1}^\top \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Since we proved the Schur complement \mathbf{S}_{n-1} is positive definite in Lemma 2.4, then we can factor it in the same way as

$$\mathbf{S}_{n-1} = \hat{\mathbf{R}}_2^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{n-2} \end{bmatrix} \hat{\mathbf{R}}_2.$$

Therefore, we have

$$\begin{aligned}\mathbf{A} &= \mathbf{R}_1^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_2^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{n-2} \end{bmatrix} \hat{\mathbf{R}}_2 \end{bmatrix} \mathbf{R}_1 \\ &= \mathbf{R}_1^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_2^\top \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{n-2} \end{bmatrix} \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{R}}_2 \end{bmatrix} \mathbf{R}_1 \\ &= \mathbf{R}_1^\top \mathbf{R}_2^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{n-2} \end{bmatrix} \end{bmatrix} \mathbf{R}_2 \mathbf{R}_1.\end{aligned}$$

The same formula can be recursively applied. This process gradually continues down to the bottom-right corner giving us the decomposition

$$\begin{aligned}\mathbf{A} &= \mathbf{R}_1^\top \mathbf{R}_2^\top \dots \mathbf{R}_n^\top \mathbf{R}_n \dots \mathbf{R}_2 \mathbf{R}_1 \\ &= \mathbf{R}^\top \mathbf{R},\end{aligned}$$

where $\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_n$ are upper triangular matrices with positive diagonal elements and $\mathbf{R} = \mathbf{R}_1 \mathbf{R}_2 \dots \mathbf{R}_n$ is also an upper triangular matrix with positive diagonal elements from which the result follows. \blacksquare

The process in the proof can also be used to calculate the Cholesky decomposition and compute the complexity of the algorithm. In Section 2.7, we will do the computation in a similar way but from a different point of view.

Lemma 2.5: ($\mathbf{R}^\top \mathbf{R}$ is PD)

For any upper triangular matrix \mathbf{R} with positive diagonal elements, then $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ is positive definite.

Proof [of Lemma 2.5] If an upper triangular matrix \mathbf{R} has positive diagonals, it has full column rank, and the null space of \mathbf{R} is of dimension 0 by the fundamental theorem of linear algebra (discussed in Appendix B, p. 427). As a result, $\mathbf{R}\mathbf{x} \neq \mathbf{0}$ for any nonzero vector \mathbf{x} . Thus $\mathbf{x}^\top \mathbf{A}\mathbf{x} = \|\mathbf{R}\mathbf{x}\|^2 > 0$ for any nonzero vector \mathbf{x} . ■

This corollary above works not only for the upper triangular matrices \mathbf{R} , but can be extended to any \mathbf{R} with linearly independent columns.

A word on the two claims Combine Theorem 2.1 and Lemma 2.5, we can claim that matrix \mathbf{A} is positive definite if and only if \mathbf{A} can be factored as $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ where \mathbf{R} is an upper triangular matrix with positive diagonals.

2.3. Sylvester's Criterion: Leading Principal Minors of PD Matrices

In Lemma 2.4, we proved for any positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, its Schur complement of \mathbf{A}_{11} is $\mathbf{S}_{n-1} = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{2:n,1}^\top$ and it is also positive definite. This is also true for its Schur complement of \mathbf{A}_{nn} , i.e., $\mathbf{S}'_{n-1} = \mathbf{A}_{1:n-1,1:n-1} - \frac{1}{\mathbf{A}_{nn}} \mathbf{A}_{1:n-1,n} \mathbf{A}_{1:n-1,n}^\top$ is also positive definite.

We then claim that all the leading principal minors (Definition 1.4, p. 31) of a positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ are positive. This is also known as the Sylvester's criterion (Swamy, 1973; Gilbert, 1991). Recall that these positive leading principal minors imply the existence of the LU decomposition for positive definite matrix \mathbf{A} by Theorem 1.5 (p. 31).

To show the Sylvester's criterion, we need the following lemma.

Positive Definite Matrix Property 5 of 6

Lemma 2.1: (Quadratic PD)

Let \mathbf{E} be any invertible matrix. Then \mathbf{A} is positive definite if and only if $\mathbf{E}^\top \mathbf{A}\mathbf{E}$ is also positive definite.

Proof [of Lemma 2.1] We will prove by forward implication and reverse implication separately as follows.

Forward implication Suppose \mathbf{A} is positive definite, then for any nonzero vector \mathbf{x} , we have $\mathbf{x}^\top \mathbf{E}^\top \mathbf{A}\mathbf{E}\mathbf{x} = \mathbf{y}^\top \mathbf{A}\mathbf{y} > 0$, since \mathbf{E} is invertible such that $\mathbf{E}\mathbf{x}$ is nonzero.¹ This implies $\mathbf{E}^\top \mathbf{A}\mathbf{E}$ is PD.

¹. Since the null space of \mathbf{E} is of dimension 0 and the only solution for $\mathbf{E}\mathbf{x} = \mathbf{0}$ is the trivial solution $\mathbf{x} = \mathbf{0}$.

Reverse implication Conversely, suppose $\mathbf{E}^\top \mathbf{A} \mathbf{E}$ is positive definite, for any nonzero \mathbf{x} , we have $\mathbf{x}^\top \mathbf{E}^\top \mathbf{A} \mathbf{E} \mathbf{x} > 0$. For any nonzero \mathbf{y} , there exists a nonzero \mathbf{x} such that $\mathbf{y} = \mathbf{E} \mathbf{x}$ since \mathbf{E} is invertible. This implies \mathbf{A} is PD as well. \blacksquare

We then provide the rigorous proof for Sylvester's criterion.

Positive Definite Matrix Property 6 of 6

Theorem 2.2: (Sylvester's Criterion)

The real symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite if and only if all the leading principal minors of \mathbf{A} are positive.

Proof [of Theorem 2.2] We will prove by forward implication and reverse implication separately as follows.

Forward implication: We will prove by induction for the forward implication. Suppose \mathbf{A} is positive definite. Since all the components on the diagonal of positive definite matrices are all positive (Lemma 2.2, p. 57). The case for $n = 1$ is trivial that $\det(\mathbf{A}) > 0$ if \mathbf{A} is a scalar.

Suppose all the leading principal minors for $k \times k$ matrices are all positive. If we could prove this is also true for $(k+1) \times (k+1)$ PD matrices, then we complete the proof.

For a $(k+1) \times (k+1)$ matrix with the block form $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^\top & d \end{bmatrix}$, where \mathbf{A} is a $k \times k$ submatrix. Then its Schur complement of d , $\mathbf{S}_k = \mathbf{A} - \frac{1}{d} \mathbf{b} \mathbf{b}^\top$ is also positive definite and its determinant is positive from the assumption. Therefore, $\det(\mathbf{M}) = \det(d) \det(\mathbf{A} - \frac{1}{d} \mathbf{b} \mathbf{b}^\top) = 2^k d \cdot \det(\mathbf{A} - \frac{1}{d} \mathbf{b} \mathbf{b}^\top) > 0$, which completes the proof.

Reverse implication: Conversely, suppose all the leading principal minors of $\mathbf{A} \in \mathbb{R}^{n \times n}$ are positive, i.e., leading principal submatrices are nonsingular. Suppose further the (i, j) -th entry of \mathbf{A} is denoted by a_{ij} , we realize that $a_{11} > 0$ by the assumption. Subtract multiples of the first row of \mathbf{A} to zero out the entries in the first column of \mathbf{A} below the first diagonal a_{11} . That is,

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \xrightarrow{\mathbf{E}_1 \mathbf{A}} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \dots & a_{nn} \end{bmatrix}.$$

This operation preserves the values of the principal minors of \mathbf{A} . The \mathbf{E}_1 might be mysterious to the readers. Actually, the \mathbf{E}_1 contains two steps $\mathbf{E}_1 = \mathbf{Z}_{12} \mathbf{Z}_{11}$. The first step \mathbf{Z}_{11}

-
2. By the fact that if matrix \mathbf{M} has a block formulation: $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$, then $\det(\mathbf{M}) = \det(\mathbf{D}) \det(\mathbf{A} - \mathbf{B} \mathbf{D}^{-1} \mathbf{C})$.

is to subtract the 2-nd row to the n -th row by a multiple of the first row, that is

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \xrightarrow{\mathbf{Z}_{11}\mathbf{A}} \begin{bmatrix} 1 & 0 & \dots & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{11}} & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & (a_{22} - \frac{a_{21}}{a_{11}}a_{12}) & \dots & (a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (a_{n2} - \frac{a_{n1}}{a_{11}}a_{12}) & \dots & (a_{nn} - \frac{a_{n1}}{a_{11}}a_{1n}) \end{bmatrix}, \end{aligned}$$

where we subtract the bottom-right $(n-1) \times (n-1)$ by some terms additionally. \mathbf{Z}_{12} is to add back these terms.

$$\begin{aligned} \mathbf{Z}_{11}\mathbf{A} &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & (a_{22} - \frac{a_{21}}{a_{11}}a_{12}) & \dots & (a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (a_{n2} - \frac{a_{n1}}{a_{11}}a_{12}) & \dots & (a_{nn} - \frac{a_{n1}}{a_{11}}a_{1n}) \end{bmatrix} \\ \xrightarrow{\mathbf{Z}_{12}(\mathbf{Z}_{11}\mathbf{A})} & \begin{bmatrix} 1 & 0 & \dots & 0 \\ \frac{a_{21}}{a_{11}} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_{n1}}{a_{11}} & 0 & \dots & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & (a_{22} - \frac{a_{21}}{a_{11}}a_{12}) & \dots & (a_{2n} - \frac{a_{21}}{a_{11}}a_{1n}) \\ \vdots & \vdots & \ddots & \vdots \\ 0 & (a_{n2} - \frac{a_{n1}}{a_{11}}a_{12}) & \dots & (a_{nn} - \frac{a_{n1}}{a_{11}}a_{1n}) \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \dots & a_{nn} \end{bmatrix} = \mathbf{E}_1\mathbf{A}. \end{aligned}$$

Now subtract multiples of the first column of $\mathbf{E}_1\mathbf{A}$, from the other columns of $\mathbf{E}_1\mathbf{A}$ to zero out the entries in the first row of $\mathbf{E}_1\mathbf{A}$ to the right of the first column. Since \mathbf{A} is symmetric, we can multiply on the right by \mathbf{E}_1^\top to get what we want. We then have

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \xrightarrow{\mathbf{E}_1\mathbf{A}} \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \dots & a_{nn} \end{bmatrix} \xrightarrow{\mathbf{E}_1\mathbf{A}\mathbf{E}_1^\top} \begin{bmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} & \dots & a_{nn} \end{bmatrix}.$$

This operation also preserves the values of the principal minors of \mathbf{A} . The leading principal minors of $\mathbf{E}_1\mathbf{A}\mathbf{E}_1^\top$ are exactly the same as those of \mathbf{A} .

Continue this process, we will transform \mathbf{A} into a diagonal matrix $\mathbf{E}_n \dots \mathbf{E}_1\mathbf{A}\mathbf{E}_1^\top \dots \mathbf{E}_n^\top$ whose diagonal values are exactly the same as the diagonals of \mathbf{A} and are positive. Let $\mathbf{E} = \mathbf{E}_n \dots \mathbf{E}_1$, which is an invertible matrix. Apparently, $\mathbf{E}\mathbf{A}\mathbf{E}^\top$ is PD, which implies \mathbf{A} is PD as well from Lemma 2.1. ■

2.4. Existence of the Cholesky Decomposition via the LU Decomposition without Permutation

By Theorem 2.2 on Sylvester's criterion and the existence of LU decomposition without permutation in Theorem 1.5 (p. 31), there is a unique LU decomposition for positive definite matrix $\mathbf{A} = \mathbf{L}\mathbf{U}_0$ where \mathbf{L} is a unit lower triangular matrix and \mathbf{U}_0 is an upper triangular matrix. Since *the signs of the pivots of a symmetric matrix are the same as the signs of the eigenvalues* (Strang, 2009):

$$\text{number of positive pivots} = \text{number of positive eigenvalues.}$$

And $\mathbf{A} = \mathbf{L}\mathbf{U}_0$ has the following form

$$\mathbf{A} = \mathbf{L}\mathbf{U}_0 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix}.$$

This implies that the diagonals of \mathbf{U}_0 contain the pivots of \mathbf{A} . And all the eigenvalues of PD matrices are positive (see Lemma 13.4, p. 260, which is a consequence of spectral decomposition). Thus the diagonals of \mathbf{U}_0 are positive.

Taking the diagonal of \mathbf{U}_0 out into a diagonal matrix \mathbf{D} , we can rewrite $\mathbf{U}_0 = \mathbf{DU}$ as shown in the following equation

$$\mathbf{A} = \mathbf{L}\mathbf{U}_0 = \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & 0 & \dots & 0 \\ 0 & u_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12}/u_{11} & \dots & u_{1n}/u_{11} \\ 0 & 1 & \dots & u_{2n}/u_{22} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = \mathbf{LDU},$$

where \mathbf{U} is a *unit* upper triangular matrix. By the uniqueness of the LU decomposition without permutation in Corollary 1.1 (p. 36) and the symmetry of \mathbf{A} , it follows that $\mathbf{U} = \mathbf{L}^\top$, and $\mathbf{A} = \mathbf{LDL}^\top$. Since the diagonals of \mathbf{D} are positive, we can set $\mathbf{R} = \mathbf{D}^{1/2}\mathbf{L}^\top$ where $\mathbf{D}^{1/2} = \text{diag}(\sqrt{u_{11}}, \sqrt{u_{22}}, \dots, \sqrt{u_{nn}})$ such that $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ is the Cholesky decomposition of \mathbf{A} , and \mathbf{R} is upper triangular with positive diagonals.

2.4.1 Diagonal Values of the Upper Triangular Matrix

Suppose \mathbf{A} is a PD matrix, take \mathbf{A} as a block matrix $\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$ where $\mathbf{A}_{11} \in \mathbb{R}^{k \times k}$, and its block LU decomposition is given by

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \mathbf{L}\mathbf{U}_0 = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} \\ \mathbf{0} & \mathbf{U}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}_{11}\mathbf{U}_{11} & \mathbf{L}_{11}\mathbf{U}_{12} \\ \mathbf{L}_{21}\mathbf{U}_{11} & \mathbf{L}_{21}\mathbf{U}_{12} + \mathbf{L}_{22}\mathbf{U}_{22} \end{bmatrix}. \end{aligned}$$

Then the leading principal minor (Definition 1.4, p. 31), $\Delta_k = \det(\mathbf{A}_{1:k, 1:k}) = \det(\mathbf{A}_{11})$ is given by

$$\Delta_k = \det(\mathbf{A}_{11}) = \det(\mathbf{L}_{11}\mathbf{U}_{11}) = \det(\mathbf{L}_{11})\det(\mathbf{U}_{11}).$$

We notice that \mathbf{L}_{11} is a unit lower triangular matrix and \mathbf{U}_{11} is an upper triangular matrix. By the fact that the determinant of a lower triangular matrix (or an upper triangular matrix) is the product of the diagonal entries, we obtain

$$\Delta_k = \det(\mathbf{U}_{11}) = u_{11}u_{22}\dots u_{kk},$$

i.e., the k -th leading principal minor of \mathbf{A} is the determinant of the $k \times k$ submatrix of \mathbf{U}_0 . That is also the product of the first k diagonals of \mathbf{D} (\mathbf{D} is the matrix from $\mathbf{A} = \mathbf{LDL}^\top$). Let $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_n)$, therefore, we have

$$\Delta_k = d_1d_2\dots d_k = \Delta_{k-1}d_k.$$

This gives us an alternative form of \mathbf{D} , i.e., the **squared** diagonal values of \mathbf{R} (\mathbf{R} is the Cholesky from $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$), and it is given by

$$\mathbf{D} = \text{diag}\left(\Delta_1, \frac{\Delta_2}{\Delta_1}, \dots, \frac{\Delta_n}{\Delta_{n-1}}\right),$$

where Δ_k is the k -th leading principal minor of \mathbf{A} , for all $k \in \{1, 2, \dots, n\}$. That is, the diagonal values of \mathbf{R} are given by

$$\text{diag}\left(\sqrt{\Delta_1}, \sqrt{\frac{\Delta_2}{\Delta_1}}, \dots, \sqrt{\frac{\Delta_n}{\Delta_{n-1}}}\right).$$

2.4.2 Block Cholesky Decomposition

Following from the last section, suppose \mathbf{A} is a PD matrix, take \mathbf{A} as a block matrix $\mathbf{A} = \begin{bmatrix} \mathbf{A}_k & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$ where $\mathbf{A}_k \in \mathbb{R}^{k \times k}$, and its block LU decomposition is given by

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{A}_k & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \mathbf{LU}_0 = \begin{bmatrix} \mathbf{L}_k & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \mathbf{U}_{12} \\ \mathbf{0} & \mathbf{U}_{22} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}_k \mathbf{U}_k & \mathbf{L}_{11} \mathbf{U}_{12} \\ \mathbf{L}_{21} \mathbf{U}_{11} & \mathbf{L}_{21} \mathbf{U}_{12} + \mathbf{L}_{22} \mathbf{U}_{22} \end{bmatrix}. \end{aligned}$$

where the k -th leading principal submatrix \mathbf{A}_k of \mathbf{A} also has its LU decomposition $\mathbf{A}_k = \mathbf{L}_k \mathbf{U}_k$. Then, it is trivial that the Cholesky decomposition of an $n \times n$ matrix contains $n - 1$ other Cholesky decompositions within it: $\mathbf{A}_k = \mathbf{R}_k^\top \mathbf{R}_k$, for all $k \in \{1, 2, \dots, n - 1\}$. This is particularly true that any leading principal submatrix \mathbf{A}_k of the positive definite matrix \mathbf{A} is also positive definite. This can be shown that for positive definite matrix $\mathbf{A}_{k+1} \in \mathbb{R}^{(k+1) \times (k+1)}$, and any nonzero vector $\mathbf{x}_k \in \mathbb{R}^k$ appended by a zero element $\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_k \\ 0 \end{bmatrix}$.

It follows that

$$\mathbf{x}_k^\top \mathbf{A}_k \mathbf{x}_k = \mathbf{x}_{k+1}^\top \mathbf{A}_{k+1} \mathbf{x}_{k+1} > 0,$$

and \mathbf{A}_k is positive definite. If we start from $\mathbf{A} \in \mathbb{R}^{n \times n}$, we will recursively get that \mathbf{A}_{n-1} is PD, \mathbf{A}_{n-2} is PD, And all of them admit a Cholesky decomposition.

2.5. Existence of the Cholesky Decomposition via Induction

In the last section, we proved the existence of the Cholesky decomposition via the LU decomposition without permutation. Following from the proof of the LU decomposition in Section 1.3, we realize that the existence of Cholesky decomposition can be a direct consequence of induction as well.

Proof [of Theorem 2.1: Existence of Cholesky Decomposition by Induction] We will prove by induction that every $n \times n$ positive definite matrix \mathbf{A} has a decomposition $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$. The 1×1 case is trivial by setting $R = \sqrt{A}$, thus, $A = R^2$.

Suppose for any $k \times k$ PD matrix \mathbf{A}_k has a Cholesky decomposition. If we prove any $(k+1) \times (k+1)$ PD matrix \mathbf{A}_{k+1} can also be factored as this Cholesky decomposition, then we complete the proof.

For any $(k+1) \times (k+1)$ PD matrix \mathbf{A}_{k+1} , Write out \mathbf{A}_{k+1} as

$$\mathbf{A}_{k+1} = \begin{bmatrix} \mathbf{A}_k & \mathbf{b} \\ \mathbf{b}^\top & d \end{bmatrix}.$$

We note that \mathbf{A}_k is PD. By the inductive hypothesis, it admits a Cholesky decomposition \mathbf{A}_k is given by $\mathbf{A}_k = \mathbf{R}_k^\top \mathbf{R}_k$. We can construct the upper triangular matrix

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_k & \mathbf{r} \\ 0 & s \end{bmatrix},$$

such that it follows that

$$\mathbf{R}_{k+1}^\top \mathbf{R}_{k+1} = \begin{bmatrix} \mathbf{R}_k^\top \mathbf{R}_k & \mathbf{R}_k^\top \mathbf{r} \\ \mathbf{r}^\top \mathbf{R}_k & \mathbf{r}^\top \mathbf{r} + s^2 \end{bmatrix}.$$

Therefore, if we can prove $\mathbf{R}_{k+1}^\top \mathbf{R}_{k+1} = \mathbf{A}_{k+1}$ is the Cholesky decomposition of \mathbf{A}_{k+1} (which requires the value s to be positive), then we complete the proof. That is, we need to prove

$$\begin{aligned} \mathbf{b} &= \mathbf{R}_k^\top \mathbf{r}, \\ d &= \mathbf{r}^\top \mathbf{r} + s^2. \end{aligned}$$

Since \mathbf{R}_k is nonsingular, we have a unique solution for \mathbf{r} and s that

$$\begin{aligned} \mathbf{r} &= \mathbf{R}_k^{-\top} \mathbf{b}, \\ s &= \sqrt{d - \mathbf{r}^\top \mathbf{r}} = \sqrt{d - \mathbf{b}^\top \mathbf{A}_k^{-1} \mathbf{b}}, \end{aligned}$$

since we assume s is nonnegative. However, we need to further prove that s is not only nonnegative, but also positive. Since \mathbf{A}_k is PD, from Sylvester's criterion, and the fact that if matrix \mathbf{M} has a block formulation: $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$, then $\det(\mathbf{M}) = \det(\mathbf{A}) \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$. We have

$$\det(\mathbf{A}_{k+1}) = \det(\mathbf{A}_k) \det(d - \mathbf{b}^\top \mathbf{A}_k^{-1} \mathbf{b}) = \det(\mathbf{A}_k)(d - \mathbf{b}^\top \mathbf{A}_k^{-1} \mathbf{b}) > 0.$$

Since $\det(\mathbf{A}_k) > 0$, we then obtain that $(d - \mathbf{b}^\top \mathbf{A}_k^{-1} \mathbf{b}) > 0$ and this implies $s > 0$. We complete the proof. \blacksquare

2.6. Uniqueness of the Cholesky Decomposition

Corollary 2.1: (Uniqueness of Cholesky Decomposition)

The Cholesky decomposition $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ for any positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is unique.

The uniqueness of the Cholesky decomposition can be an immediate consequence of the uniqueness of the LU decomposition without permutation. Or, an alternative rigorous proof is provided as follows.

Proof [of Corollary 2.1] Suppose the Cholesky decomposition is not unique, then we can find two decompositions such that $\mathbf{A} = \mathbf{R}_1^\top \mathbf{R}_1 = \mathbf{R}_2^\top \mathbf{R}_2$ which implies

$$\mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{R}_1^{-\top} \mathbf{R}_2^\top.$$

From the fact that the inverse of an upper triangular matrix is also an upper triangular matrix, and the product of two upper triangular matrices is also an upper triangular matrix,³ we realize that the left-side of the above equation is an upper triangular matrix and the right-side of it is a lower triangular matrix. This implies $\mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{R}_1^{-\top} \mathbf{R}_2^\top$ is a diagonal matrix, and $\mathbf{R}_1^{-\top} \mathbf{R}_2^\top = (\mathbf{R}_1^{-\top} \mathbf{R}_2^\top)^\top = \mathbf{R}_2 \mathbf{R}_1^{-1}$. Let $\Lambda = \mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{R}_2 \mathbf{R}_1^{-1}$ be the diagonal matrix. We notice that the diagonal value of Λ is the product of the corresponding diagonal values of \mathbf{R}_1 and \mathbf{R}_2^{-1} (or \mathbf{R}_2 and \mathbf{R}_1^{-1}). That is, for

$$\mathbf{R}_1 = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ 0 & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & r_{nn} \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ 0 & s_{22} & \dots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & s_{nn} \end{bmatrix},$$

we have,

$$\mathbf{R}_1 \mathbf{R}_2^{-1} = \begin{bmatrix} \frac{r_{11}}{s_{11}} & 0 & \dots & 0 \\ 0 & \frac{r_{22}}{s_{22}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{r_{nn}}{s_{nn}} \end{bmatrix} = \begin{bmatrix} \frac{s_{11}}{r_{11}} & 0 & \dots & 0 \\ 0 & \frac{s_{22}}{r_{22}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{s_{nn}}{r_{nn}} \end{bmatrix} = \mathbf{R}_2 \mathbf{R}_1^{-1}.$$

Since both \mathbf{R}_1 and \mathbf{R}_2 have positive diagonals, this implies $r_{11} = s_{11}, r_{22} = s_{22}, \dots, r_{nn} = s_{nn}$. And $\Lambda = \mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{R}_2 \mathbf{R}_1^{-1} = \mathbf{I}$. That is, $\mathbf{R}_1 = \mathbf{R}_2$ and this leads to a contradiction. The Cholesky decomposition is thus unique. ■

³. Same for lower triangular matrices: the inverse of a lower triangular matrix is also a lower triangular matrix, and the product of two lower triangular matrices is also a lower triangular matrix.

2.7. Computing the Cholesky Decomposition Recursively

Similar to computing the LU decomposition, to compute the Cholesky decomposition, we write out the equality $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{1,2:n} \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{11} & 0 \\ \mathbf{R}_{1,2:n}^\top & \mathbf{R}_{2:n,2:n}^\top \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{1,2:n} \\ 0 & \mathbf{R}_{2:n,2:n} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_{11}^2 & \mathbf{R}_{11}\mathbf{R}_{1,2:n} \\ \mathbf{R}_{11}\mathbf{R}_{1,2:n}^\top & \mathbf{R}_{1,2:n}^\top \mathbf{R}_{1,2:n} + \mathbf{R}_{2:n,2:n}^\top \mathbf{R}_{2:n,2:n} \end{bmatrix},\end{aligned}$$

which allows to determine the first row of \mathbf{R} by

$$\mathbf{R}_{11} = \sqrt{\mathbf{A}_{11}}, \quad \mathbf{R}_{1,2:n} = \frac{1}{\mathbf{R}_{11}} \mathbf{A}_{1,2:n}.$$

Let $\mathbf{A}_2 = \mathbf{R}_{2:n,2:n}^\top \mathbf{R}_{2:n,2:n}$. The equality $\mathbf{A}_{2:n,2:n} = \mathbf{R}_{1,2:n}^\top \mathbf{R}_{1,2:n} + \mathbf{R}_{2:n,2:n}^\top \mathbf{R}_{2:n,2:n}$ indicates

$$\begin{aligned}\mathbf{A}_2 &= \mathbf{R}_{2:n,2:n}^\top \mathbf{R}_{2:n,2:n} = \mathbf{A}_{2:n,2:n} - \mathbf{R}_{1,2:n}^\top \mathbf{R}_{1,2:n} \\ &= \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{1,2:n}^\top \mathbf{A}_{1,2:n} \\ &= \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n}, \quad (\mathbf{A} \text{ is symmetric})\end{aligned}$$

where \mathbf{A}_2 is the Schur complement of \mathbf{A}_{11} in \mathbf{A} of size $(n-1) \times (n-1)$. And to get $\mathbf{R}_{2:n,2:n}$ we must compute the Cholesky decomposition of matrix \mathbf{A}_2 of shape $(n-1) \times (n-1)$. Again, this is a recursive algorithm and is formulated in Algorithm 8.

Algorithm 8 Cholesky Decomposition via Recursive Algorithm

Require: Positive definite matrix \mathbf{A} with size $n \times n$;

- 1: Calculate first row of \mathbf{R} by $\mathbf{R}_{11} = \sqrt{\mathbf{A}_{11}}$, $\mathbf{R}_{1,2:n} = \frac{1}{\mathbf{R}_{11}} \mathbf{A}_{1,2:n}$; $\triangleright n$ flops
- 2: Compute the Cholesky decomposition of the $(n-1) \times (n-1)$ matrix

$$\mathbf{A}_2 = \mathbf{R}_{2:n,2:n}^\top \mathbf{R}_{2:n,2:n} = \mathbf{A}_{2:n,2:n} - \frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n};$$

$\triangleright n^2 - n$ flops

Theorem 2.1: (Algorithm Complexity: Cholesky Recursively)

Algorithm 8 requires $\sim (1/3)n^3$ flops to compute the Cholesky decomposition of an $n \times n$ positive definite matrix.

Proof [of Theorem 2.1] Step 1 takes 1 square root and $(n-1)$ divisions which take n flops totally.

For step 2, note that $\frac{1}{\mathbf{A}_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{1,2:n} = (\frac{1}{\sqrt{\mathbf{A}_{11}}} \mathbf{A}_{2:n,1})(\frac{1}{\sqrt{\mathbf{A}_{11}}} \mathbf{A}_{1,2:n}) = \mathbf{R}_{1,2:n}^\top \mathbf{R}_{1,2:n}$. If we calculate the complexity directly from the equation in step 2, we will get the same complexity

as the LU decomposition. But the symmetry of $\mathbf{R}_{1,2:n}^\top \mathbf{R}_{1,2:n}$ can be adopted, the complexity of $\mathbf{R}_{1,2:n}^\top \mathbf{R}_{1,2:n}$ reduces from $(n-1) \times (n-1)$ multiplications to $1+2+\dots+(n-1) = \frac{n^2-n}{2}$ multiplications, which is almost half of the original complexity. The cost of matrix division reduces from $(n-1) \times (n-1)$ to $1+2+\dots+(n-1) = \frac{n^2-n}{2}$ as well. So the total cost for step 2 is $n^2 - n$ flops.

Let $f(n) = n^2 - n + n = n^2$, the total complexity is

$$\text{cost} = f(n) + f(n-1) + \dots + f(1).$$

Simple calculations will show the total complexity for all the recursive steps is $\frac{2n^3+3n^2+n}{6}$ flops which is $(1/3)n^3$ flops if we keep only the leading term. ■

An important use of the Cholesky decomposition computation above is for testing whether a symmetric matrix is positive definite. The test is simply to run the algorithm above and declare the matrix positive definite if the algorithm completes without encountering any negative or zero pivots (in step 1 above) and not positive definite otherwise.

To end up this section, we provide the full pseudo code for Algorithm 8 as shown in Algorithm 9 (compare the two algorithms).

Algorithm 9 Cholesky Decomposition via Recursive Algorithm: Full Pseudo Code

Require: Positive definite matrix \mathbf{A} with size $n \times n$;

- ```

1: for $k = 1$ to n do \triangleright compute the k th row of \mathbf{R}
2: $\mathbf{R}_{kk} = \sqrt{\mathbf{A}_{kk}}$; \triangleright first element of k -th row, 1 flop
3: $\mathbf{R}_{k,k+1:n} = \frac{1}{\mathbf{R}_{kk}} \mathbf{A}_{k,k+1:n}$; \triangleright the rest elements of k -th row, $n-k$ flops
4: $\mathbf{A}_{k+1:n,k+1:n} = \mathbf{A}_{k+1:n,k+1:n} - \mathbf{R}_{k,k+1:n}^\top \mathbf{R}_{k,k+1:n}$; $\triangleright 2(1+2+\dots+(n-k))$ flops
5: end for
```
- 

## 2.8. Computing the Cholesky Decomposition Element-Wise

It is also common to compute the Cholesky decomposition via element-level equations which come from directly solving the matrix equation  $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ . We notice that the entry  $(i, j)$  of  $\mathbf{A}$  is  $\mathbf{A}_{ij} = \mathbf{R}_{:,i}^\top \mathbf{R}_{:,j} = \sum_{k=1}^i \mathbf{R}_{ki} \mathbf{R}_{kj}$  if  $i < j$ . This further implies,

$$\begin{aligned} \mathbf{A}_{ij} &= \mathbf{R}_{:,i}^\top \mathbf{R}_{:,j} = \sum_{k=1}^i \mathbf{R}_{ki} \mathbf{R}_{kj} \\ &= \sum_{k=1}^{i-1} \mathbf{R}_{ki} \mathbf{R}_{kj} + \mathbf{R}_{ii} \mathbf{R}_{ij}, \end{aligned}$$

and

$$\mathbf{R}_{ij} = (\mathbf{A}_{ij} - \sum_{k=1}^{i-1} \mathbf{R}_{ki} \mathbf{R}_{kj}) / \mathbf{R}_{ii},$$

if  $i < j$ . If we equate elements of  $\mathbf{R}$  by taking a column at a time and start with  $\mathbf{R}_{11} = \sqrt{\mathbf{A}_{11}}$ , the element-level algorithm is formulated in Algorithm 10.

**Algorithm 10** Cholesky Decomposition Element-Wise

**Require:** Positive definite matrix  $\mathbf{A}$  with size  $n \times n$ ;

- 1: Calculate first of  $\mathbf{R}$  by  $\mathbf{R}_{11} = \sqrt{\mathbf{A}_{11}}$ ;
- 2: **for**  $j = 1$  to  $n$  **do**
- 3:     **for**  $i = 1$  to  $j - 1$  **do**
- 4:          $\mathbf{R}_{ij} = (\mathbf{A}_{ij} - \sum_{k=1}^{i-1} \mathbf{R}_{ki} \mathbf{R}_{kj}) / \mathbf{R}_{ii}$ , since  $i < j$ ;
- 5:     **end for**
- 6:      $\mathbf{R}_{jj} = \sqrt{\mathbf{A}_{jj} - \sum_{k=1}^{j-1} \mathbf{R}_{kj}^2}$ ;
- 7: **end for**

**Theorem 2.1: (Algorithm Complexity: Cholesky Element-wise)**

Algorithm 10 requires  $\sim (1/3)n^3$  flops to compute the Cholesky decomposition of an  $n \times n$  positive definite matrix.

**Proof** [of Theorem 2.1] For step 4 in Algorithm 10, that is, for each  $j, i$ , step 4 involves  $(i - 1)$  multiplications,  $(i - 2)$  additions, 1 subtraction, and 1 division, which is  $2i - 1$  flops. Let  $f(k) = 2k - 1$ , the total flops required from step 4 for each loop  $j$  is given by

$$f(1) + f(2) + \dots + f(j - 1) = j^2 - 2j + 1 \text{ flops.}$$

Further, for each loop  $j$ , step 6 requires  $j - 1$  multiplications,  $j - 2$  additions, 1 subtraction, 1 square root. That is, step 6 involves  $2j - 1$  flops for each loop  $j$ . Combine the flops needed in step 4, it needs  $j^2$  flops for each loop  $j$  totally. Let  $g(k) = k^2$ , the total complexity is thus given by

$$g(1) + g(2) + \dots + g(n) = \frac{2n^3 + 3n^2 + n}{6} \text{ flops,}$$

which is  $(1/3)n^3$  flops if we keep only the leading term. ■

The complexity of Algorithm 10 is the same as that of Algorithm 8. And indeed, the ideas behind them are similar as well.

## 2.9. More Properties of Positive Definite Matrices

**Lemma 2.1: (PD Properties)**

For positive definite matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , we have the following properties

- 1). Any principal minors are positive (see Definition 1.3, p. 30, not necessarily to be the leading principal minors);
- 2). Suppose the diagonal values of  $\mathbf{A}$  are  $a_{ii}$ , for all  $i \in \{1, 2, \dots, n\}$ , then  $\det(\mathbf{A}) \leq \prod_{i=1}^n a_{ii}$ . The equality can be obtained when  $\mathbf{A}$  is a diagonal matrix.

**Proof** [of Lemma 2.1] For 1). Similar to the permutation matrix (Definition 0.15, p. 19), we can define a selection matrix. For a row selection matrix  $\mathbf{P}$ , if we select a row of the

matrix  $\mathbf{A}$  by  $\mathbf{P}\mathbf{A}$ , the corresponding diagonal of  $\mathbf{P}$  will be 1, and 0 otherwise. In this sense, the  $\mathbf{P}^\top = \mathbf{P}$  is a column selection matrix such that  $\mathbf{A}\mathbf{P}^\top$  is to select the corresponding columns. Then, any  $k \times k$  submatrix of  $\mathbf{A}$  can be obtained by  $\mathbf{P}\mathbf{A}\mathbf{P}^\top$ . For any vector  $\mathbf{x} \in \mathbb{R}^n$  where not all the corresponding  $k$  entries are zero (such that  $\mathbf{x}^\top \mathbf{P}$  is nonzero), we have

$$\mathbf{x}^\top \mathbf{P}\mathbf{A}\mathbf{P}^\top \mathbf{x} > 0.$$

Since  $(n - k)$  rows of  $\mathbf{P}$  are zero and the corresponding  $(n - k)$  columns of  $\mathbf{P}$  are zero as well, these rows and columns will not count any value for the above equation, we can just remove them away. This implies the  $k \times k$  submatrix  $\mathbf{A}_k \in \mathbb{R}^{k \times k}$  of  $\mathbf{A}$ , and its corresponding  $k \times k$  sub-selection matrix  $\mathbf{P}_k \in \mathbb{R}^{k \times k}$  (which is an identity matrix here), such that

$$\mathbf{x}_k^\top \mathbf{P}_k \mathbf{A}_k \mathbf{P}_k^\top \mathbf{x}_k > 0,$$

where  $\mathbf{x}_k \in \mathbb{R}^k$ . Since the vector  $\mathbf{x}$  is any vector in  $\mathbb{R}^n$ , the  $\mathbf{x}_k$  is thus also any vector in  $\mathbb{R}^k$ . This results in that  $\mathbf{A}_k$  is PD.

For 2). For the LU decomposition of  $\mathbf{A}$ ,  $\mathbf{A} = \mathbf{L}\mathbf{U}_0$  has the following form

$$\begin{aligned} \mathbf{A} = \mathbf{L}\mathbf{U}_0 &= \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ 0 & u_{22} & \dots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & 0 & \dots & 0 \\ 0 & u_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12}/u_{11} & \dots & u_{1n}/u_{11} \\ 0 & 1 & \dots & u_{2n}/u_{22} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & \dots & 0 \\ l_{21} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \dots & 1 \end{bmatrix} \begin{bmatrix} u_{11} & 0 & \dots & 0 \\ 0 & u_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{nn} \end{bmatrix} \begin{bmatrix} 1 & l_{21} & \dots & l_{n1} \\ 0 & 1 & \dots & l_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} = \mathbf{L}\mathbf{D}\mathbf{L}^\top. \end{aligned}$$

We have discussed in Section 2.4.1 that  $\det(\mathbf{A}) = \det(\mathbf{L})\det(\mathbf{U}_0) = \prod_{i=1}^n u_{ii} \leq \prod_{i=1}^n a_{ii}$ , where the last inequality comes from the fact that  $\mathbf{A}$  is symmetric such that

$$a_{ii} = \left( \sum_{j=1}^i l_{ij}u_{ji} \right) + u_{ii} = \left( \sum_{j=1}^i l_{ij}^2 \cdot u_{ii} \right) + u_{ii} \geq u_{ii}.$$

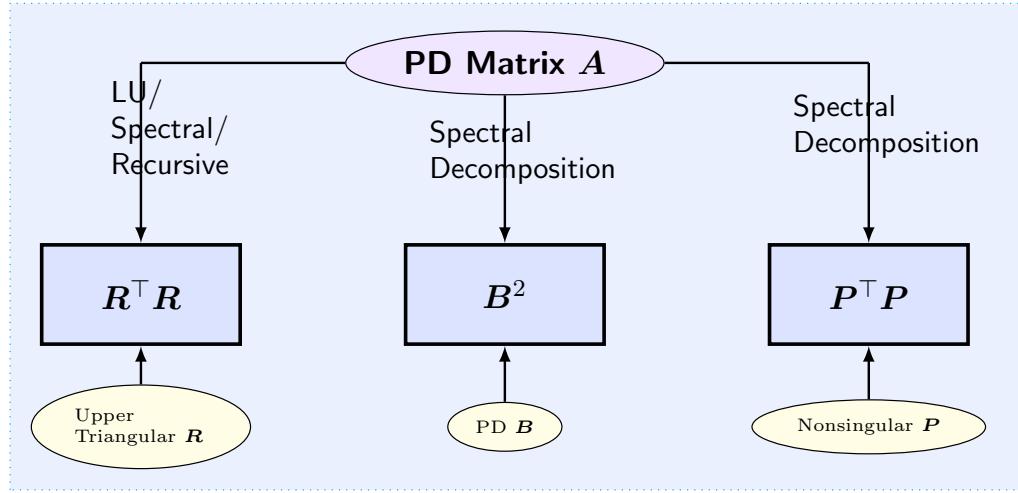
And the equality can be obtained when  $\mathbf{A}$  is a diagonal matrix. ■

## 2.10. Last Words on Positive Definite Matrices

In Section 13.6.2 (p. 259), we will prove that a matrix  $\mathbf{A}$  is PD if and only if  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$  where  $\mathbf{P}$  is nonsingular. And in Section 13.6.5 (p. 262), we will

prove that PD matrix  $\mathbf{A}$  can be uniquely factored as  $\mathbf{A} = \mathbf{B}^2$  where  $\mathbf{B}$  is also PD. The two results are both consequences of the spectral decomposition of PD matrices.

To conclude, for PD matrix  $\mathbf{A}$ , we can factor it into  $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$  where  $\mathbf{R}$  is an upper triangular matrix with positive diagonals as shown in Theorem 2.1 by Cholesky decomposition,  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$  where  $\mathbf{P}$  is nonsingular in Theorem 13.5 (p. 260), and  $\mathbf{A} = \mathbf{B}^2$  where  $\mathbf{B}$  is PD in Theorem 13.7 (p. 262). For clarity, the different factorizations of positive definite matrix  $\mathbf{A}$  are summarized in Figure 2.1.



**Figure 2.1:** Demonstration of different factorizations on positive definite matrix  $\mathbf{A}$ .

## 2.11. Pivoted Cholesky Decomposition

If  $\mathbf{P}$  is a permutation matrix and  $\mathbf{A}$  is positive definite, then  $\mathbf{P}^\top \mathbf{A} \mathbf{P}$  is said to be a diagonal permutation of  $\mathbf{A}$  (among other things, it permutes the diagonals of  $\mathbf{A}$ ). Any diagonal permutation of  $\mathbf{A}$  is positive definite and has a Cholesky factor. Such a factorization is called a pivoted Cholesky factorization. There are many ways to pivot a Cholesky decomposition, but the most common one is the complete pivoting (see Section 1.13.2, p. 51) such that

$$\mathbf{P} \mathbf{A} \mathbf{P}^\top = \mathbf{R}^\top \mathbf{R}$$

is the column-pivoted Cholesky decomposition of  $\mathbf{A}$ , where  $\mathbf{P}$  is a permutation matrix, and  $\mathbf{R}$  is upper triangular.

Following the Cholesky decomposition via recursive calculation in Algorithm 9, we notice from Lemma 2.3 that the maximal element of a PD matrix lies on the diagonal. Therefore, the complete pivoting algorithm for Cholesky decomposition can only search in the diagonals. The procedure is shown in Algorithm 11.

## 2.12. Decomposition for Semidefinite Matrices

For positive semidefinite matrices, the Cholesky decomposition also exists with slight modification.

**Algorithm 11** Cholesky Decomposition via Recursive Algorithm: Complete Pivoting

**Require:** Positive definite matrix  $\mathbf{A}$  with size  $n \times n$ ;

- 1:  $\mathbf{R} \in \mathbb{R}^{n \times n}$  is initialized with all zeros;
- 2: **for**  $k = 1$  to  $n$  **do** ▷ compute the  $k$ th row of  $\mathbf{R}$
- 3:   Search in the diagonals of  $\mathbf{A}_{k:n,k:n}$  such that  $\mathbf{A}_{vv} = \max \mathbf{A}_{k:n,k:n}$ ;
- 4:   Swap the  $k$ -th and  $v$ -th column of  $\mathbf{A}$ :  $\mathbf{A}_{:,k} \leftrightarrow \mathbf{A}_{:,v}$  by column permutation;
- 5:   Swap the  $k$ -th and  $v$ -th column of  $\mathbf{R}$ :  $\mathbf{R}_{:,k} \leftrightarrow \mathbf{R}_{:,v}$  by column permutation;
- 6:   Swap the  $k$ -th and  $v$ -th row of  $\mathbf{A}$ :  $\mathbf{A}_{k,:} \leftrightarrow \mathbf{A}_{v,:}$  by row permutation;
- 7:    $\mathbf{R}_{kk} = \sqrt{\mathbf{A}_{kk}}$ ; ▷ first element of  $k$ -th row, 1 flop
- 8:    $\mathbf{R}_{k,k+1:n} = \frac{1}{\mathbf{R}_{kk}} \mathbf{A}_{k,k+1:n}$ ; ▷ the rest elements of  $k$ -th row,  $n - k$  flops
- 9:    $\mathbf{A}_{k+1:n,k+1:n} = \mathbf{A}_{k+1:n,k+1:n} - \mathbf{R}_{k,k+1:n}^\top \mathbf{R}_{k,k+1:n}$ ; ▷  $2(1 + 2 + \dots + (n - k))$  flops
- 10: **end for**

**Theorem 2.1: (Semidefinite Decomposition)**

Every positive semidefinite matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as

$$\mathbf{A} = \mathbf{R}^\top \mathbf{R},$$

where  $\mathbf{R} \in \mathbb{R}^{n \times n}$  is an upper triangular matrix with possible **zero** diagonal elements and the factorization is **not unique** in general.

For such decomposition, the diagonal of  $\mathbf{R}$  may not display the rank of  $\mathbf{A}$  ([Higham, 2009](#)).

**Example 2.1** ([\(Higham, 2009\)](#)) Suppose

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

The semidefinite decomposition is given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = \mathbf{R}^\top \mathbf{R}.$$

$\mathbf{A}$  has rank 2, but  $\mathbf{R}$  has only one nonzero diagonal element.  $\square$

We notice that all PD matrices have full rank, and this fact permeates many of our proofs discussed above. This can be proved by the Sylvester's criterion ([Theorem 2.2, p. 61](#)) that all the leading principal minors of PD matrices are positive. Or, we can simply prove that if a PD matrix  $\mathbf{A}$  is rank deficient, this implies the null space of  $\mathbf{A}$  has positive dimension and there exists a vector  $\mathbf{x}$  in the null space of  $\mathbf{A}$  such that  $\mathbf{A}\mathbf{x} = \mathbf{0}$ , i.e.,  $\mathbf{x}^\top \mathbf{A}\mathbf{x} = 0$  and it leads to a contradiction to the definition of the PD matrices.

However, this is not necessarily true for PSD matrices where the dimension of the null space can be larger than 0. Therefore, and more generally, a rank-revealing decomposition for semidefinite decomposition is provided as follows.

**Theorem 2.2: (Semidefinite Rank-Revealing Decomposition)**

Every positive semidefinite matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with rank  $r$  can be factored as

$$\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{R}^\top \mathbf{R}, \quad \text{with} \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{R}^{n \times n},$$

where  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is an upper triangular matrix with positive diagonal elements, and  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ .

The proof for the existence of the above rank-revealing decomposition for semidefinite matrices is delayed in Section 13.6.3 (p. 261) as a consequence of the spectral decomposition (Theorem 13.1, p. 241) and the column-pivoted QR decomposition (Theorem 3.1, p. 101). Whereas, the rigorous proof for the trivial Semidefinite Decomposition Theorem 2.1 can be a direct result of the spectral decomposition and trivial QR decomposition (Theorem 3.1, p. 82).

## 2.13. Application: Rank-One Update/Downdate

Updating linear systems after low-rank modifications of the system matrix is widespread in machine learning, statistics, and many other fields. However, it is well known that this update can lead to serious instabilities in the presence of roundoff error (Seeger, 2004). If the system matrix is positive definite, it is almost always possible to use a representation based on the Cholesky decomposition which is much more numerically stable. We will shortly provide the proof for this rank one update/downdate via Cholesky decomposition in this section.

### 2.13.1 Rank-One Update

A rank-one update  $\mathbf{A}'$  of matrix  $\mathbf{A}$  by vector  $\mathbf{x}$  is of the form (Gill et al., 1974; Bojanczyk et al., 1987; Chang, 1997; Davis and Hager, 1999; Seeger, 2004; Chen et al., 2008; Davis, 2008; Higham, 2009):

$$\begin{aligned} \mathbf{A}' &= \mathbf{A} + \mathbf{v}\mathbf{v}^\top \\ \mathbf{R}'^\top \mathbf{R}' &= \mathbf{R}^\top \mathbf{R} + \mathbf{v}\mathbf{v}^\top. \end{aligned}$$

If we have already calculated the Cholesky factor  $\mathbf{R}$  of  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , then the Cholesky factor  $\mathbf{R}'$  of  $\mathbf{A}'$  can be calculated efficiently. Note that  $\mathbf{A}'$  differs from  $\mathbf{A}$  only via the symmetric rank-one matrix. Hence we can compute  $\mathbf{R}'$  from  $\mathbf{R}$  using the rank-one Cholesky update, which takes  $O(n^2)$  operations each saving from  $O(n^3)$  if we do know  $\mathbf{R}$ , the Cholesky decomposition of  $\mathbf{A}$  up front, i.e., we want to compute the Cholesky decomposition of  $\mathbf{A}'$  via that of  $\mathbf{A}$ . To see this, suppose there is a set of orthogonal matrices  $\mathbf{Q}_n \mathbf{Q}_{n-1} \dots \mathbf{Q}_1$  such that that

$$\mathbf{Q}_n \mathbf{Q}_{n-1} \dots \mathbf{Q}_1 \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{R}' \end{bmatrix}.$$

Then we find out the expression for the Cholesky factor of  $\mathbf{A}'$  by  $\mathbf{R}'$ . Specifically, multiply the left-hand side (l.h.s.,) of above equation by its transpose,

$$[\mathbf{v} \quad \mathbf{R}^\top] \mathbf{Q}_1 \dots \mathbf{Q}_{n-1} \mathbf{Q}_n \mathbf{Q}_n \mathbf{Q}_{n-1} \dots \mathbf{Q}_1 \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R} \end{bmatrix} = \mathbf{R}^\top \mathbf{R} + \mathbf{v} \mathbf{v}^\top.$$

And multiply the right-hand side (r.h.s.,) by its transpose,

$$[\mathbf{0} \quad \mathbf{R}'^\top] \begin{bmatrix} \mathbf{0} \\ \mathbf{R}' \end{bmatrix} = \mathbf{R}'^\top \mathbf{R}',$$

which agrees with the l.h.s., equation. Givens rotations are such orthogonal matrices that can transfer  $\mathbf{R}, \mathbf{v}$  into  $\mathbf{R}'$ . We will discuss the intrinsic meaning of Givens rotation shortly to prove the existence of QR decomposition in Section 3.15 (p. 115). Here, we only introduce the definition of it and write out the results directly. Feel free to skip this section for a first reading.

### Definition 2.1: $n$ -th Order Givens Rotation

A Givens rotation is represented by a matrix of the following form

$$\mathbf{G}_{kl} = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & c & & s & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & -s & & c & \\ & & & & & & 1 \\ & & & & & & & \ddots \end{bmatrix}_{n \times n},$$

where the  $(k, k), (k, l), (l, k), (l, l)$  entries are  $c, s, -s, c$  respectively, and  $s = \cos \theta$  and  $c = \sin \theta$  for some  $\theta$ .

Let  $\delta_k \in \mathbb{R}^n$  be the zero vector except that the entry  $k$  is 1. Then mathematically, the Givens rotation defined above can be denoted by

$$\mathbf{G}_{kl} = \mathbf{I} + (c - 1)(\delta_k \delta_k^\top + \delta_l \delta_l^\top) + s(\delta_k \delta_l^\top - \delta_l \delta_k^\top).$$

It can be easily verified that the  $n$ -th order Givens rotation is an orthogonal matrix and its determinant is 1. For any vector  $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top \in \mathbb{R}^n$ , we have  $\mathbf{y} = \mathbf{G}_{kl} \mathbf{x}$ , where

$$\begin{cases} y_k = c \cdot x_k + s \cdot x_l, \\ y_l = -s \cdot x_k + c \cdot x_l, \\ y_j = x_j, \quad (j \neq k, l) \end{cases}$$

That is, a Givens rotation applied to  $\mathbf{x}$  rotates two components of  $\mathbf{x}$  by some angle  $\theta$  and leaves all other components the same.

Now, suppose we have an  $(n+1)$ -th order Givens rotation indexed from 0 to  $n$ , and it is given by

$$\mathbf{G}_k = \mathbf{I} + (c_k - 1)(\delta_0 \delta_0^\top + \delta_k \delta_k^\top) + s_k(\delta_0 \delta_k^\top - \delta_k \delta_0^\top),$$

where  $c_k = \cos \theta_k$ ,  $s_k = \sin \theta_k$  for some  $\theta_k$ ,  $\mathbf{G}_k \in \mathbb{R}^{(n+1) \times (n+1)}$ ,  $\delta_k \in \mathbb{R}^{n+1}$  is a zero vector except that the  $(k+1)$ -th entry is 1.

Taking out the  $k$ -th column of the following equation

$$\begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{R}' \end{bmatrix},$$

where we let the  $k$ -th element of  $\mathbf{v}$  be  $v_k$ , and the  $k$ -th diagonal of  $\mathbf{R}$  be  $r_{kk}$ . We realize that  $\sqrt{v_k^2 + r_{kk}^2} \neq 0$ , let  $c_k = \frac{r_{kk}}{\sqrt{v_k^2 + r_{kk}^2}}$ ,  $s_k = -\frac{v_k}{\sqrt{v_k^2 + r_{kk}^2}}$ . Then,

$$\begin{cases} v_k \rightarrow c_k v_k + s_k r_{kk} = 0; \\ r_{kk} \rightarrow -s_k v_k + c_k r_{kk} = \sqrt{v_k^2 + r_{kk}^2} = r'_{kk}. \end{cases}$$

That is,  $\mathbf{G}_k$  will introduce zero value to the  $k$ -th element to  $\mathbf{v}$  and nonzero value to  $r_{kk}$ .

This finding above is essential for the rank-one update. And we obtain

$$\mathbf{G}_n \mathbf{G}_{n-1} \dots \mathbf{G}_1 \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{R}' \end{bmatrix}.$$

For each Givens rotation, it takes  $6n$  flops. And there are  $n$  such rotations, which requires  $6n^2$  flops if keeping only the leading term. The complexity to calculate the Cholesky factor of  $\mathbf{A}'$  is thus reduced from  $\frac{1}{3}n^3$  to  $6n^2$  flops if we already know the Cholesky factor of  $\mathbf{A}$  by the rank-one update. The above algorithm is essential to reduce the complexity of the posterior calculation in the Bayesian inference for Gaussian mixture model (Lu, 2021a).

### 2.13.2 Rank-One Downdate

Now suppose we have calculated the Cholesky factor of  $\mathbf{A}$ , and the  $\mathbf{A}'$  is the downdate of  $\mathbf{A}$  as follows:

$$\begin{aligned} \mathbf{A}' &= \mathbf{A} - \mathbf{v}\mathbf{v}^\top \\ \mathbf{R}'^\top \mathbf{R}' &= \mathbf{R}^\top \mathbf{R} - \mathbf{v}\mathbf{v}^\top. \end{aligned}$$

The algorithm is similar by proceeding as follows:

$$\mathbf{G}_1 \mathbf{G}_2 \dots \mathbf{G}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}' \end{bmatrix}. \quad (2.1)$$

Again,  $\mathbf{G}_k = \mathbf{I} + (c_k - 1)(\delta_0 \delta_0^\top + \delta_k \delta_k^\top) + s_k(\delta_0 \delta_k^\top - \delta_k \delta_0^\top)$ , can be constructed as follows:

Taking out the  $k$ -th column of the following equation

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}' \end{bmatrix}.$$

We realize that  $r_{kk} \neq 0$ , let  $c_k = \frac{\sqrt{r_{kk}^2 - v_k^2}}{r_{kk}}$ ,  $s_k = \frac{v_k}{r_{kk}}$ . Then,

$$\begin{cases} 0 \rightarrow s_k r_{kk} = v_k; \\ r_{kk} \rightarrow c_k r_{kk} = \sqrt{r_{kk}^2 - v_k^2} = r'_{kk}. \end{cases}$$

This requires  $r_{kk}^2 > v_k^2$  to make  $\mathbf{A}'$  to be positive definite. Otherwise,  $c_k$  above will not exist.

Again, one can check that, multiply the l.h.s., of Equation (2.1) by its transpose, we have

$$[\mathbf{0} \quad \mathbf{R}^\top] \mathbf{G}_n \dots \mathbf{G}_2 \mathbf{G}_1 \mathbf{G}_1 \mathbf{G}_2 \dots \mathbf{G}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{R} \end{bmatrix} = \mathbf{R}^\top \mathbf{R}.$$

And multiply the r.h.s., by its transpose, we have

$$[\mathbf{v} \quad \mathbf{R}'^\top] \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}' \end{bmatrix} = \mathbf{v} \mathbf{v}^\top + \mathbf{R}'^\top \mathbf{R}'.$$

This results in  $\mathbf{R}'^\top \mathbf{R}' = \mathbf{R}^\top \mathbf{R} - \mathbf{v} \mathbf{v}^\top$ .

## 2.14. Application: Indefinite Rank Two Update

Let  $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$  be the Cholesky decomposition of  $\mathbf{A}$ , (Goldfarb, 1976; Seeger, 2004) give a stable method for the indefinite rank-two update of the form

$$\mathbf{A}' = (\mathbf{I} + \mathbf{v} \mathbf{u}^\top) \mathbf{A} (\mathbf{I} + \mathbf{u} \mathbf{v}^\top).$$

Let

$$\begin{cases} \mathbf{z} = \mathbf{R}^{-\top} \mathbf{v}, \\ \mathbf{w} = \mathbf{R} \mathbf{u}, \end{cases} \rightarrow \begin{cases} \mathbf{v} = \mathbf{R}^\top \mathbf{z}, \\ \mathbf{u} = \mathbf{R}^{-1} \mathbf{w}. \end{cases}$$

And suppose the LQ decomposition <sup>4</sup> of  $\mathbf{I} + \mathbf{z} \mathbf{w}^\top$  is given by  $\mathbf{I} + \mathbf{z} \mathbf{w}^\top = \mathbf{L} \mathbf{Q}$ , where  $\mathbf{L}$  is lower triangular and  $\mathbf{Q}$  is orthogonal. Thus, we have

$$\begin{aligned} \mathbf{A}' &= (\mathbf{I} + \mathbf{v} \mathbf{u}^\top) \mathbf{A} (\mathbf{I} + \mathbf{u} \mathbf{v}^\top) \\ &= (\mathbf{I} + \mathbf{R}^\top \mathbf{z} \mathbf{w}^\top \mathbf{R}^{-\top}) \mathbf{A} (\mathbf{I} + \mathbf{R}^{-1} \mathbf{w} \mathbf{z}^\top \mathbf{R}) \\ &= \mathbf{R}^\top (\mathbf{I} + \mathbf{z} \mathbf{w}^\top) (\mathbf{I} + \mathbf{w} \mathbf{z}^\top) \mathbf{R} \\ &= \mathbf{R}^\top \mathbf{L} \mathbf{Q} \mathbf{Q}^\top \mathbf{L}^\top \mathbf{R} \\ &= \mathbf{R}^\top \mathbf{L} \mathbf{L}^\top \mathbf{R}. \end{aligned}$$

Let  $\mathbf{R}' = \mathbf{R}^\top \mathbf{L}$  which is lower triangular, we find the Cholesky decomposition of  $\mathbf{A}'$ .

<sup>4</sup> We will shortly introduce in Theorem 3.1 (p. 124).

## Part II

# Triangularization, Orthogonalization and Gram-Schmidt Process



## Introduction

Given an  $m \times l$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_l]$ , with  $m \geq l$ , the *orthonormalization* admits  $\mathbf{Q}_l = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_l]$  such that

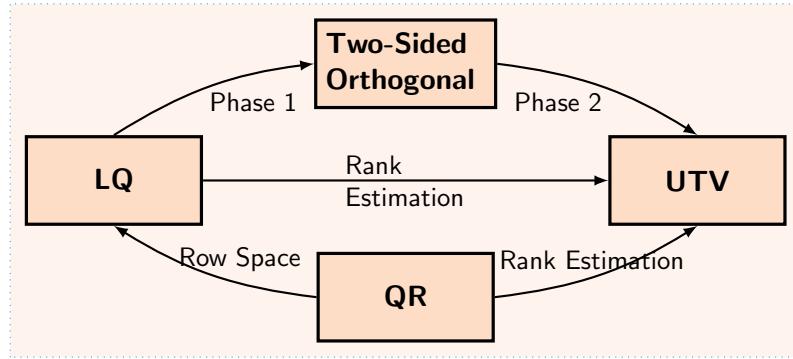
$$\text{span}([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k]) = \text{span}([\mathbf{q}_1, \mathbf{a}_2, \dots, \mathbf{a}_k]), \quad \text{for all } k \in \{1, 2, \dots, l\}.$$

Whilst, columns of  $\mathbf{Q}_l$  are mutually orthonormal:

$$\mathbf{q}_i^\top \mathbf{q}_j = \begin{cases} 1, & \text{if } i = j; \\ 0, & \text{if } i \neq j. \end{cases} \quad \text{leads to} \quad \mathbf{Q}_l^\top \mathbf{Q}_l = \mathbf{I}_l.$$

where  $\mathbf{I}_l$  is a  $l \times l$  identity matrix. When we complete  $\mathbf{Q}_l$  into  $m$  mutually orthonormal columns  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{Q}_m]$ ,  $\mathbf{Q}$  is a square matrix and it follows that  $\mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}^\top\mathbf{Q} = \mathbf{I}_m$ . Sometimes, this *column completion* is done by the Gram-Schmidt process that we will introduce in the sequel.

For the rest of this part, we will discuss several factorization methods in the sense of orthogonalization above with different focuses, e.g., in terms of column space, row space, or both. The big picture can be shown in Figure 2.2.



**Figure 2.2:** Orthogonalization World Map. See also where it's lying in the matrix decomposition world map Figure 1.

## Chapter 3

# QR Decomposition

### Contents

---

|        |                                                                               |     |
|--------|-------------------------------------------------------------------------------|-----|
| 3.1    | QR Decomposition . . . . .                                                    | 82  |
| 3.2    | Project a Vector Onto Another Vector . . . . .                                | 82  |
| 3.3    | Project a Vector Onto a Plane . . . . .                                       | 83  |
| 3.4    | Existence of the QR Decomposition via the Gram-Schmidt Process . . . . .      | 83  |
| 3.5    | Orthogonal vs Orthonormal . . . . .                                           | 87  |
| 3.6    | Properties of the QR Decomposition . . . . .                                  | 89  |
| 3.7    | Computing the Reduced QR Decomposition via the Gram-Schmidt Process . . . . . | 90  |
| 3.8    | Computing the Full QR Decomposition via the Gram-Schmidt Process . . . . .    | 100 |
| 3.9    | Dependent Columns . . . . .                                                   | 100 |
| 3.10   | QR with Column Pivoting: Column-Pivoted QR (CPQR) . .                         | 101 |
| 3.10.1 | A Simple CPQR via CGS . . . . .                                               | 101 |
| 3.10.2 | A Practical CPQR via CGS . . . . .                                            | 103 |
| 3.10.3 | A Practical CPQR via MGS . . . . .                                            | 104 |
| 3.10.4 | Partial Factorization for CPQR: Extra Bonus of CPQR via MGS                   | 105 |
| 3.11   | QR with Column Pivoting: Revealing Rank One Deficiency .                      | 106 |
| 3.12   | QR with Column Pivoting: Revealing Rank r Deficiency* .                       | 107 |
| 3.13   | Existence of the QR Decomposition via the Householder Reflector . . . . .     | 108 |
| 3.14   | Computing the Full QR Decomposition via the Householder Reflector . . . . .   | 113 |
| 3.15   | Existence of the QR Decomposition via the Givens Rotation .                   | 115 |
| 3.16   | Computing the Full QR Decomposition via the Givens Rotation                   | 120 |
| 3.17   | Uniqueness of the QR Decomposition . . . . .                                  | 122 |
| 3.18   | LQ Decomposition . . . . .                                                    | 123 |
| 3.19   | Two-Sided Orthogonal Decomposition . . . . .                                  | 124 |

|                                                                           |            |
|---------------------------------------------------------------------------|------------|
| <b>3.20 Applications . . . . .</b>                                        | <b>125</b> |
| 3.20.1 Application: Least Squares via the Full QR Decomposition . . . . . | 125        |
| 3.20.2 Application: Rank-One Changes . . . . .                            | 127        |
| 3.20.3 Application: Appending or Deleting a Column . . . . .              | 131        |
| 3.20.4 Application: Appending or Deleting a Row . . . . .                 | 135        |
| 3.20.5 Application: Reducing the Ill-Condition via the QR decomposition   | 137        |

---

### 3.1. QR Decomposition

In many applications, we are interested in the column space of a matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ . The successive spaces spanned by the columns  $\mathbf{a}_1, \mathbf{a}_2, \dots$  of  $\mathbf{A}$  are

$$\mathcal{C}([\mathbf{a}_1]) \subseteq \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2]) \subseteq \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]) \subseteq \dots,$$

where  $\mathcal{C}([\dots])$  is the subspace spanned by the vectors included in the brackets. The idea of QR decomposition is the construction of a sequence of orthonormal vectors  $\mathbf{q}_1, \mathbf{q}_2, \dots$  that span the same successive subspaces.

$$\left\{ \mathcal{C}([\mathbf{q}_1]) = \mathcal{C}([\mathbf{a}_1]) \right\} \subseteq \left\{ \mathcal{C}([\mathbf{q}_1, \mathbf{q}_2]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2]) \right\} \subseteq \left\{ \mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]) \right\} \subseteq \dots,$$

We provide the result of QR decomposition in the following theorem and we delay the discussion of its existence in the next sections.

#### Theorem 3.1: (QR Decomposition)

Every  $m \times n$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  (whether linearly independent or dependent columns) with  $m \geq n$  can be factored as

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

where

1. **Reduced:**  $\mathbf{Q}$  is  $m \times n$  with orthonormal columns and  $\mathbf{R}$  is an  $n \times n$  upper triangular matrix which is known as the **reduced QR decomposition**;
2. **Full:**  $\mathbf{Q}$  is  $m \times m$  with orthonormal columns and  $\mathbf{R}$  is an  $m \times n$  upper triangular matrix which is known as the **full QR decomposition**. If further restrict the upper triangular matrix to be a square matrix, the full QR decomposition can be denoted as

$$\mathbf{A} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_0 \\ \mathbf{0} \end{bmatrix},$$

where  $\mathbf{R}_0$  is an  $m \times m$  upper triangular matrix.

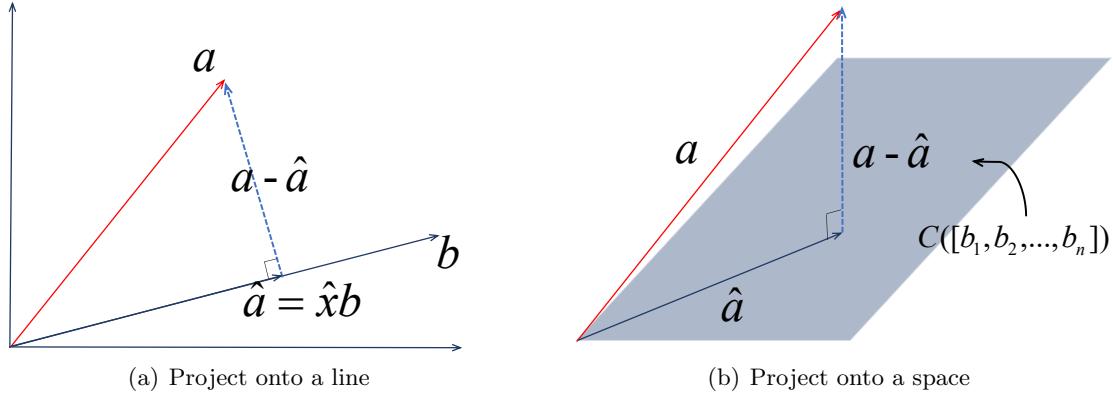
Specifically, when  $\mathbf{A}$  has full rank, i.e., has linearly independent columns,  $\mathbf{R}$  also has linearly independent columns, and  $\mathbf{R}$  is nonsingular for the *reduced* case. This implies diagonals of  $\mathbf{R}$  are nonzero. Under this condition, when we further restrict elements on the diagonal of  $\mathbf{R}$  are positive, the *reduced* QR decomposition is **unique**. The *full* QR decomposition is normally not unique since the right-most  $(m - n)$  columns in  $\mathbf{Q}$  can be in any order.

### 3.2. Project a Vector Onto Another Vector

Project a vector  $\mathbf{a}$  to a vector  $\mathbf{b}$  is to find the vector closest to  $\mathbf{a}$  on the line of  $\mathbf{b}$ . The projection vector  $\hat{\mathbf{a}}$  is some multiple of  $\mathbf{b}$ . Let  $\hat{\mathbf{a}} = \hat{x}\mathbf{b}$  and  $\mathbf{a} - \hat{\mathbf{a}}$  is perpendicular to  $\mathbf{b}$  as shown in Figure 3.1(a). We then get the following result:

**Project Vector  $\mathbf{a}$  Onto Vector  $\mathbf{b}$** 

$\mathbf{a}^\perp = \mathbf{a} - \hat{\mathbf{a}}$  is perpendicular to  $\mathbf{b}$ , so  $(\mathbf{a} - \hat{\mathbf{x}}\mathbf{b})^\top \mathbf{b} = 0$ :  $\hat{\mathbf{x}} = \frac{\mathbf{a}^\top \mathbf{b}}{\mathbf{b}^\top \mathbf{b}}$  and  $\hat{\mathbf{a}} = \frac{\mathbf{a}^\top \mathbf{b}}{\mathbf{b}^\top \mathbf{b}} \mathbf{b} = \frac{\mathbf{b}\mathbf{b}^\top}{\mathbf{b}^\top \mathbf{b}} \mathbf{a}$ .



**Figure 3.1:** Project a vector onto a line and a space.

### 3.3. Project a Vector Onto a Plane

Project a vector  $\mathbf{a}$  to a space spanned by  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$  is to find the vector closest to  $\mathbf{a}$  on the column space of  $[\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$ . The projection vector  $\hat{\mathbf{a}}$  is a combination of  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$ :  $\hat{\mathbf{a}} = \hat{x}_1 \mathbf{b}_1 + \hat{x}_2 \mathbf{b}_2 + \dots + \hat{x}_n \mathbf{b}_n$ . This is actually a least squares problem. To find the projection, we just solve the normal equation  $\mathbf{B}^\top \mathbf{B} \hat{\mathbf{x}} = \mathbf{B}^\top \mathbf{a}$  where  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$  and  $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n]$ . We refer the details of this projection view in the least squares to (Strang, 2009; Trefethen and Bau III, 1997; Yang, 2000; Golub and Van Loan, 2013; Lu, 2021d) as it is not the main interest of this survey. For each vector  $\mathbf{b}_i$ , the projection of  $\mathbf{a}$  in the direction of  $\mathbf{b}_i$  can be analogously obtained by

$$\hat{\mathbf{a}}_i = \frac{\mathbf{b}_i \mathbf{b}_i^\top}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{a}, \quad \forall i \in \{1, 2, \dots, n\}.$$

Let  $\hat{\mathbf{a}} = \sum_{i=1}^n \hat{\mathbf{a}}_i$ , this results in

$$\mathbf{a}^\perp = (\mathbf{a} - \hat{\mathbf{a}}) \perp \mathcal{C}(\mathbf{B}),$$

i.e.,  $(\mathbf{a} - \hat{\mathbf{a}})$  is perpendicular to the column space of  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$  as shown in Figure 3.1(b).

### 3.4. Existence of the QR Decomposition via the Gram-Schmidt Process

#### First View by Projection Directly

For three linearly independent vectors  $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$  and the space spanned by the three linearly independent vectors  $\mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ , i.e., the column space of the matrix  $[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ .

We intend to construct three orthogonal vectors  $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$  in which case  $\mathcal{C}([\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ . Then we divide the orthogonal vectors by their length to normalize. This process produces three mutually orthonormal vectors  $\mathbf{q}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|}$ ,  $\mathbf{q}_2 = \frac{\mathbf{b}_2}{\|\mathbf{b}_2\|}$ ,  $\mathbf{q}_3 = \frac{\mathbf{b}_3}{\|\mathbf{b}_3\|}$ .

For the first vector, we choose  $\mathbf{b}_1 = \mathbf{a}_1$  directly. The second vector  $\mathbf{b}_2$  must be perpendicular to the first one. This is actually the vector  $\mathbf{a}_2$  subtracting its projection along  $\mathbf{b}_1$ :

$$\begin{aligned}\mathbf{b}_2 &= \mathbf{a}_2 - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1} \mathbf{a}_2 = (\mathbf{I} - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1}) \mathbf{a}_2 && \text{(Projection view)} \\ &= \mathbf{a}_2 - \underbrace{\frac{\mathbf{b}_1^\top \mathbf{a}_2}{\mathbf{b}_1^\top \mathbf{b}_1} \mathbf{b}_1}_{\hat{\mathbf{a}}_2}, && \text{(Combination view)}\end{aligned}$$

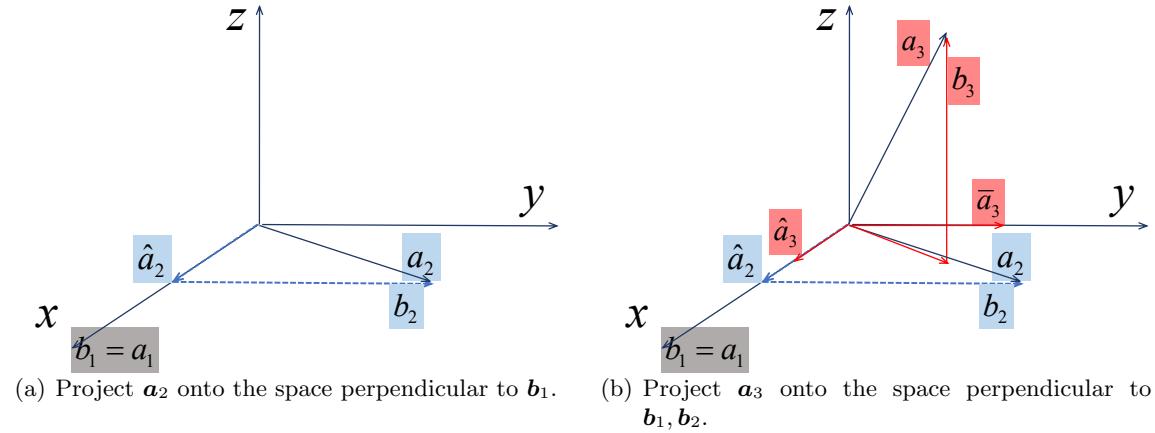
where the first equation shows  $\mathbf{b}_2$  is a multiplication of the matrix  $(\mathbf{I} - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1})$  and the vector  $\mathbf{a}_2$ , i.e., project  $\mathbf{a}_2$  onto the orthogonal complement space of  $\mathcal{C}([\mathbf{b}_1])$ . The second equality in the above equation shows  $\mathbf{a}_2$  is a combination of  $\mathbf{b}_1$  and  $\mathbf{b}_2$ . Clearly, the space spanned by  $\mathbf{b}_1, \mathbf{b}_2$  is the same space spanned by  $\mathbf{a}_1, \mathbf{a}_2$ . The situation is shown in Figure 3.2(a) in which we choose **the direction of  $\mathbf{b}_1$  as the  $x$ -axis in the Cartesian coordinate system**.  $\hat{\mathbf{a}}_2$  is the projection of  $\mathbf{a}_2$  onto line  $\mathbf{b}_1$ . It can be clearly shown that the part of  $\mathbf{a}_2$  perpendicular to  $\mathbf{b}_1$  is  $\mathbf{b}_2 = \mathbf{a}_2 - \hat{\mathbf{a}}_2$  from the figure.

For the third vector  $\mathbf{b}_3$ , it must be perpendicular to both the  $\mathbf{b}_1$  and  $\mathbf{b}_2$  which is actually the vector  $\mathbf{a}_3$  subtracting its projection along the plane spanned by  $\mathbf{b}_1$  and  $\mathbf{b}_2$

$$\begin{aligned}\mathbf{b}_3 &= \mathbf{a}_3 - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1} \mathbf{a}_3 - \frac{\mathbf{b}_2 \mathbf{b}_2^\top}{\mathbf{b}_2^\top \mathbf{b}_2} \mathbf{a}_3 = (\mathbf{I} - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1} - \frac{\mathbf{b}_2 \mathbf{b}_2^\top}{\mathbf{b}_2^\top \mathbf{b}_2}) \mathbf{a}_3 && \text{(Projection view)} \\ &= \mathbf{a}_3 - \underbrace{\frac{\mathbf{b}_1^\top \mathbf{a}_3}{\mathbf{b}_1^\top \mathbf{b}_1} \mathbf{b}_1}_{\hat{\mathbf{a}}_3} - \underbrace{\frac{\mathbf{b}_2^\top \mathbf{a}_3}{\mathbf{b}_2^\top \mathbf{b}_2} \mathbf{b}_2}, && \text{(Combination view)}\end{aligned}\tag{3.1}$$

where the first equation shows  $\mathbf{b}_3$  is a multiplication of the matrix  $(\mathbf{I} - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1} - \frac{\mathbf{b}_2 \mathbf{b}_2^\top}{\mathbf{b}_2^\top \mathbf{b}_2})$  and the vector  $\mathbf{a}_3$ , i.e., project  $\mathbf{a}_3$  onto the orthogonal complement space of  $\mathcal{C}([\mathbf{b}_1, \mathbf{b}_2])$ . The second equality in the above equation shows  $\mathbf{a}_3$  is a combination of  $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$ . We will see this property is essential in the idea of the QR decomposition. Again, it can be shown that the space spanned by  $\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$  is the same space spanned by  $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ . The situation is shown in Figure 3.2(b), in which we choose **the direction of  $\mathbf{b}_2$  as the  $y$ -axis of the Cartesian coordinate system**.  $\hat{\mathbf{a}}_3$  is the projection of  $\mathbf{a}_3$  onto line  $\mathbf{b}_1$ ,  $\bar{\mathbf{a}}_3$  is the projection of  $\mathbf{a}_3$  onto line  $\mathbf{b}_2$ . It can be shown that the part of  $\mathbf{a}_3$  perpendicular to both  $\mathbf{b}_1$  and  $\mathbf{b}_2$  is  $\mathbf{b}_3 = \mathbf{a}_3 - \hat{\mathbf{a}}_3 - \bar{\mathbf{a}}_3$  from the figure.

Finally, we normalize each vector by dividing their length which produces three orthonormal vectors  $\mathbf{q}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|}$ ,  $\mathbf{q}_2 = \frac{\mathbf{b}_2}{\|\mathbf{b}_2\|}$ ,  $\mathbf{q}_3 = \frac{\mathbf{b}_3}{\|\mathbf{b}_3\|}$ .

**Figure 3.2:** The Gram-Schmidt process.

This idea can be extended to a set of vectors rather than only three. And we call this process as *Gram-Schmidt process*. After this process, matrix  $\mathbf{A}$  will be triangularized. The method is named after Jørgen Pedersen Gram and Erhard Schmidt, but it appeared earlier in the work of Pierre-Simon Laplace in the theory of Lie group decomposition.

As we mentioned previously, the idea of the QR decomposition is the construction of a sequence of orthonormal vectors  $\mathbf{q}_1, \mathbf{q}_2, \dots$  that span the same successive subspaces.

$$\left\{ \mathcal{C}([\mathbf{q}_1]) = \mathcal{C}([\mathbf{a}_1]) \right\} \subseteq \left\{ \mathcal{C}([\mathbf{q}_1, \mathbf{q}_2]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2]) \right\} \subseteq \left\{ \mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]) \right\} \subseteq \dots,$$

This implies any  $\mathbf{a}_k$  is in the space spanned by  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k])$ .<sup>1</sup> As long as we have found these orthonormal vectors, to reconstruct  $\mathbf{a}_i$ 's from the orthonormal matrix  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$ , an upper triangular matrix  $\mathbf{R}$  is needed such that  $\mathbf{A} = \mathbf{QR}$ .

The Gram–Schmidt process is not the only algorithm for finding the QR decomposition. Several other QR decomposition algorithms exist such as Householder reflections and Givens rotations which are more reliable in the presence of round-off errors. These QR decomposition methods may also change the order in which the columns of  $\mathbf{A}$  are processed.

### Another View by Inner Product (with Projection Implicitly)

In the above direct projection view, we first find orthogonal vectors and then normalize them to unit 1. The projection relies on the finding in Section 3.2 where the projection of the vector  $\mathbf{a}$  onto the vector  $\mathbf{b}$  is given by  $\hat{\mathbf{a}} = \frac{\mathbf{a}^\top \mathbf{b}}{\mathbf{b}^\top \mathbf{b}} \mathbf{b} = \frac{\mathbf{b}^\top \mathbf{a}}{\mathbf{b}^\top \mathbf{b}} \mathbf{a}$ . Now suppose  $\mathbf{b}$  is of unit length, then

$$\hat{\mathbf{a}} = (\mathbf{a}^\top \mathbf{b}) \mathbf{b}. \quad (3.2)$$

Again, for three linearly independent vectors  $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3\}$  and the space spanned by the three linearly independent vectors  $\mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ . To abuse the notation, we will write  $\mathbf{b}_k$  in the first view as  $\mathbf{a}_k^\perp$  in this view to emphasize that  $\mathbf{a}_k^\perp$  is orthogonal to  $\{\mathbf{q}_1, \dots, \mathbf{q}_{k-1}\}$ . The process proceeds as follows:

- Compute  $\mathbf{q}_1$  of unit length so that  $\mathcal{C}([\mathbf{q}_1]) = \mathcal{C}([\mathbf{a}_1])$ :

<sup>1</sup>. And also, any  $\mathbf{q}_k$  is in the space spanned by  $\mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k])$ .

- Compute the length of  $\mathbf{a}_1$ :  $r_{11} = \|\mathbf{a}_1\|$ ;
- Sets  $\mathbf{q}_1$  to a unit vector in the direction of  $\mathbf{a}_1$ :  $\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|} = \frac{\mathbf{a}_1}{r_{11}}$ ;
- Compute  $\mathbf{q}_2$  of unit length so that  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2])$ :
  - Compute  $r_{12}$  so that  $r_{12}\mathbf{q}_1 = (\mathbf{a}_2^\top \mathbf{q}_1)\mathbf{q}_1$  equals the component of  $\mathbf{a}_2$  in the direction of  $\mathbf{q}_1$  by Equation (3.2);
  - Compute the component of  $\mathbf{a}_2$  that is orthogonal to  $\mathbf{q}_1$ :  $\mathbf{a}_2^\perp = \mathbf{a}_2 - r_{12}\mathbf{q}_1$ ;
  - Compute the length of vector  $\mathbf{a}_2^\perp$ :  $r_{22} = \|\mathbf{a}_2^\perp\|$ ;
  - Set  $\mathbf{q}_2$  to a unit vector in the direction of  $\|\mathbf{a}_2^\perp\|$ :  $\mathbf{q}_2 = \frac{\mathbf{a}_2^\perp}{\|\mathbf{a}_2^\perp\|} = \frac{\mathbf{a}_2^\perp}{r_{22}}$ ;
  - This results in:
- Compute  $\mathbf{q}_3$  of unit length so that  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ :
  - Compute  $r_{13}$  so that  $r_{13}\mathbf{q}_1 = (\mathbf{a}_3^\top \mathbf{q}_1)\mathbf{q}_1$  equals the component of  $\mathbf{a}_3$  in the direction of  $\mathbf{q}_1$  by Equation (3.2);
  - Compute  $r_{23}$  so that  $r_{23}\mathbf{q}_2 = (\mathbf{a}_3^\top \mathbf{q}_2)\mathbf{q}_2$  equals the component of  $\mathbf{a}_3$  in the direction of  $\mathbf{q}_2$  by Equation (3.2);
  - Compute the component of  $\mathbf{a}_3$  that is orthogonal to  $\mathbf{q}_1$  and  $\mathbf{q}_2$ :  $\mathbf{a}_3^\perp = \mathbf{a}_3 - r_{13}\mathbf{q}_1 - r_{23}\mathbf{q}_2$ ;
  - Compute the length of vector  $\mathbf{a}_3^\perp$ :  $r_{33} = \|\mathbf{a}_3^\perp\|$ ;
  - Set  $\mathbf{q}_3$  to a unit vector in the direction of  $\|\mathbf{a}_3^\perp\|$ :  $\mathbf{q}_3 = \frac{\mathbf{a}_3^\perp}{\|\mathbf{a}_3^\perp\|} = \frac{\mathbf{a}_3^\perp}{r_{33}}$ ;
  - This results in:

$$[\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3] = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \mathbf{q}_3] \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix}.$$

Again, this idea can be extended to a set of vectors rather than only three. The process above reveals the meaning of  $r_{ij}$  in the triangular matrix  $\mathbf{R}$  where  $r_{ij}$  represents the component of  $\mathbf{a}_j$  in the direction of  $\mathbf{q}_i$ . This matches the matrix multiplication result:

$$\mathbf{a}_j = \sum_{i=1}^j r_{ij} \mathbf{q}_j.$$

## Main Proof

Though the existence of the QR decomposition is conceptually intuitive from the two views of the Gram-Schmidt process above, the formal proof is clunky which needs inductive hypothesis. We now prove it rigorously.

**Proof** [of Theorem 3.1] We will prove by induction that every  $m \times n$  matrix  $\mathbf{A}$  with linearly independent columns admits a *reduced* QR decomposition. The *full* QR decomposition can

be done by completing the orthonormal columns in  $\mathbf{Q}$ . The  $1 \times 1$  case is trivial by setting  $\mathbf{Q} = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|}, \mathbf{R} = \|\mathbf{a}_1\|$ , thus,  $\mathbf{A} = \mathbf{a}_1 = \mathbf{Q}\mathbf{R}$ .

Suppose for any  $m \times k$  matrix  $\mathbf{A}_k$  with linearly independent columns admits a reduced QR decomposition. If we prove any  $m \times (k+1)$  matrix  $\mathbf{A}_{k+1}$  can also be factored as this reduced QR decomposition, then we complete the proof. Suppose  $\mathbf{A}_{k+1} = [\mathbf{A}_k, \mathbf{a}_{k+1}]$  where  $\mathbf{A}_k$  admits the reduced QR decomposition by inductive hypothesis

$$\mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k,$$

where  $\mathbf{Q}_k$  contains orthonormal columns  $\mathbf{Q}_k^\top \mathbf{Q}_k = \mathbf{I}_k$  and  $\mathbf{R}_k \in \mathbb{R}^{k \times k}$  is upper triangular. Also, by the induction hypothesis, if the values on the diagonal of  $\mathbf{R}_k$  are chosen to be positive, then the reduced QR decomposition of  $\mathbf{A}_k = \mathbf{Q}_k \mathbf{R}_k$  is **unique**.

Suppose further  $\mathbf{A}_{k+1}$  can be factored as

$$\mathbf{A}_{k+1} = [\mathbf{A}_k \quad \mathbf{a}_{k+1}] = [\tilde{\mathbf{Q}}_k \quad \mathbf{q}_{k+1}] \begin{bmatrix} \tilde{\mathbf{R}}_k & \mathbf{r}_k \\ & r_{k+1} \end{bmatrix}, \quad (3.3)$$

where apparently  $\mathbf{A}_k = \tilde{\mathbf{Q}}_k \tilde{\mathbf{R}}_k$  is a reduced QR decomposition of  $\mathbf{A}_k$  and if restrict the diagonal values of  $\tilde{\mathbf{R}}_k$  are positive, the factorization is **unique** which indicates  $\tilde{\mathbf{Q}}_k = \mathbf{Q}_k$  and  $\tilde{\mathbf{R}}_k = \mathbf{R}_k$ . Moreover, Equation (3.3) implies

$$\begin{aligned} \mathbf{a}_{k+1} &= \mathbf{Q}_k \mathbf{r}_k + \mathbf{q}_{k+1} r_{k+1} \\ \text{leads to } \mathbf{Q}_k^\top \mathbf{a}_{k+1} &= \mathbf{Q}_k^\top (\mathbf{Q}_k \mathbf{r}_k + \mathbf{q}_{k+1} r_{k+1}) = \mathbf{r}_k + \mathbf{Q}_k^\top \mathbf{q}_{k+1} r_{k+1}. \end{aligned}$$

Since we assume columns of  $\mathbf{Q}_k$  are orthonormal to  $\mathbf{q}_{k+1}$ , it follows that  $\mathbf{Q}_k^\top \mathbf{q}_{k+1} = \mathbf{0}$  and  $\mathbf{r}_k = \mathbf{Q}_k^\top \mathbf{a}_{k+1}$ . When  $\mathbf{Q}_k$  is fixed, the  $\mathbf{r}_k$  is **uniquely** decided.

Let  $\mathbf{a}_{k+1}^\perp = \mathbf{a}_{k+1} - \mathbf{Q}_k \mathbf{r}_k$ , then  $\mathbf{a}_{k+1}^\perp$  is orthogonal to the columns of  $\mathbf{Q}_k$ . To see this,  $\mathbf{Q}_k^\top \mathbf{a}_{k+1}^\perp = \mathbf{Q}_k^\top (\mathbf{a}_{k+1} - \mathbf{Q}_k \mathbf{r}_k) = \mathbf{0}$  as we construct  $\mathbf{r}_k$  by  $\mathbf{r}_k = \mathbf{Q}_k^\top \mathbf{a}_{k+1}$ . Since  $\mathbf{a}_{k+1}$  is linearly independent to columns of  $\mathbf{A}_k$ , which is also linearly independent to columns of  $\mathbf{Q}_k$ ,  $\mathbf{a}_{k+1}^\perp$  is thus nonzero. Therefore, let  $r_{k+1} = \|\mathbf{a}_{k+1}^\perp\|$  and  $\mathbf{q}_{k+1} = \mathbf{a}_{k+1}^\perp / r_{k+1}$ , we find the **unique** reduced QR decomposition of  $\mathbf{A}_{k+1}$  with positive diagonals in the upper triangular matrix. This completes the proof.  $\blacksquare$

### 3.5. Orthogonal vs Orthonormal

The vectors  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n \in \mathbb{R}^m$  are mutually orthogonal when their dot products  $\mathbf{q}_i^\top \mathbf{q}_j$  are zero whenever  $i \neq j$ . When each vector is divided by its length, the vectors become orthogonal unit vectors. Then the vectors  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n$  are mutually orthonormal. We put the orthonormal vectors into a matrix  $\mathbf{Q}$ .

- When  $m \neq n$ : the matrix  $\mathbf{Q}$  is easy to work with because  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I} \in \mathbb{R}^{n \times n}$ . Such  $\mathbf{Q}$  with  $m \neq n$  is sometimes referred to as a **semi-orthogonal** matrix.
- When  $m = n$ : the matrix  $\mathbf{Q}$  is square,  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$  means that  $\mathbf{Q}^\top = \mathbf{Q}^{-1}$ , i.e., the transpose of  $\mathbf{Q}$  is also the inverse of  $\mathbf{Q}$ . Then we also have  $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ , i.e.,  $\mathbf{Q}^\top$  is the **two-sided inverse** of  $\mathbf{Q}$ . We call this  $\mathbf{Q}$  an **orthogonal matrix**.<sup>2</sup>

<sup>2</sup>. Note here we use the term *orthogonal matrix* to mean the matrix  $\mathbf{Q}$  has orthonormal columns. The term *orthonormal matrix* is **not** used for historical reasons.

To see this, we have

$$\begin{bmatrix} \mathbf{q}_1^\top \\ \mathbf{q}_2^\top \\ \vdots \\ \mathbf{q}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}.$$

In other words,  $\mathbf{q}_i^\top \mathbf{q}_j = \delta_{ij}$  where  $\delta_{ij}$  is the Kronecker delta. The columns of an orthogonal matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  form an **orthonormal basis** of  $\mathbb{R}^n$ .<sup>3</sup>

Orthogonal matrices can be viewed as matrices that change the basis of other matrices. Hence they preserve the angle (inner product) between the vectors

$$\text{inner product: } \mathbf{u}^\top \mathbf{v} = (\mathbf{Q}\mathbf{u})^\top (\mathbf{Q}\mathbf{v}).$$

The above invariance of the inner products of angles between the vectors is preserved, which also relies on the invariance of their lengths:

$$\text{length: } \|\mathbf{Q}\mathbf{u}\| = \|\mathbf{u}\|.$$

In real cases, multiplied by a orthogonal matrix  $\mathbf{Q}$  will rotate (if  $\det(\mathbf{Q}) = 1$ ) or reflect (if  $\det(\mathbf{Q}) = -1$ ) the original vector space. Many decomposition algorithms will result in two orthogonal matrices, thus such rotations or reflections will happen twice. See Section 16 (p. 334) for a discussion on the coordinate transformation in matrix decomposition.

**Example 3.1 (Rotation and Reflection in Orthogonal Matrices)** *To see the rotation and reflection in orthogonal matrices, suppose*

$$\mathbf{Q}_1 = \begin{bmatrix} -1 & \\ & -1 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_2 = \begin{bmatrix} 1 & \\ & -1 \end{bmatrix},$$

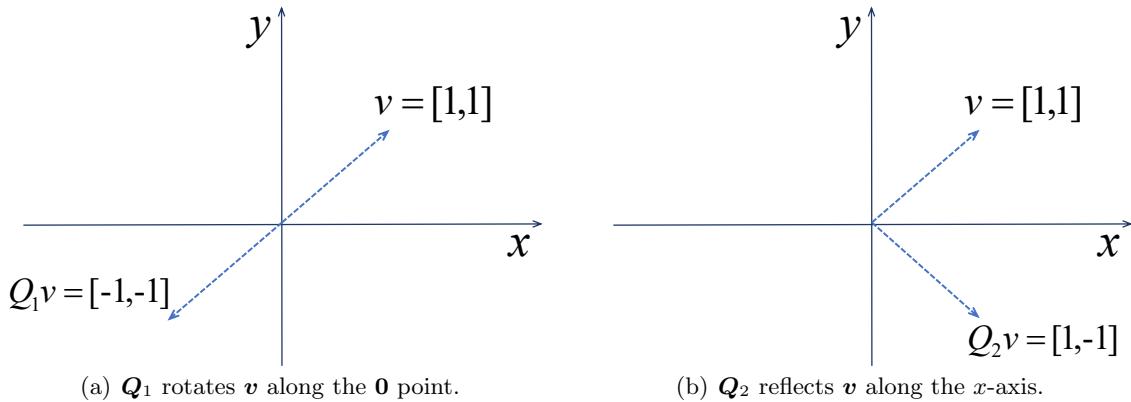
where  $\det(\mathbf{Q}_1) = 1$  and  $\det(\mathbf{Q}_2) = -1$ . For vector  $\mathbf{v} = [1, 1]^\top$ , we have

$$\mathbf{Q}_1 \mathbf{v} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_2 \mathbf{v} = \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

Thus,  $\mathbf{Q}_1$  rotates  $\mathbf{v}$  along the point  $\mathbf{0}$ , and  $\mathbf{Q}_2$  reflects  $\mathbf{v}$  along the  $x$ -axis. The illustration of the rotation and reflection is shown in Figure 3.3.  $\square$

---

3. Notice that the orthogonal matrix  $\mathbf{Q}$  contains orthonormal basis, **not** orthogonal basis.

**Figure 3.3:** Rotation and reflection in orthogonal matrices.

### 3.6. Properties of the QR Decomposition

For any matrix, we have the property:  $\mathcal{N}(\mathbf{A}^\top)$  is the orthogonal complement of the column space  $\mathcal{C}(\mathbf{A})$  in  $\mathbb{R}^m$ :  $\dim(\mathcal{N}(\mathbf{A}^\top)) + \dim(\mathcal{C}(\mathbf{A})) = m$ . This is known as the rank-nullity theorem. And the proof can be found in Appendix B (p. 427). In specific, from QR decomposition, we can find a basis for the corresponding subspaces. In singular value decomposition (SVD), we will also find the orthonormal basis for  $\mathcal{N}(\mathbf{A})$  and  $\mathcal{C}(\mathbf{A}^\top)$ .

**Lemma 3.1: (Orthonormal Basis in  $\mathbb{R}^m$ )**

For full QR decomposition of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with full rank  $n$  and  $m \geq n$ , we have the following property:

- $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A})$ ;
- $\{\mathbf{q}_{n+1}, \mathbf{q}_{n+2}, \dots, \mathbf{q}_m\}$  is an orthonormal basis of  $\mathcal{N}(\mathbf{A}^\top)$ .

**Proof** [of Lemma 3.1] From the Gram-Schmidt process, it is trivial that  $\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k\}$  is equal to  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$  for all  $k \in \{1, 2, \dots, n\}$ . Thus  $\mathcal{C}(\mathbf{A}) = \text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n\} = \text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$ , and  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$  is an orthonormal basis for the column space of  $\mathbf{A}$ . As  $\mathcal{N}(\mathbf{A}^\top) \perp \mathcal{C}(\mathbf{A})$ ,  $\dim(\mathcal{N}(\mathbf{A}^\top)) = m - \dim(\mathcal{C}(\mathbf{A})) = m - n$ . And the space spanned by  $\{\mathbf{q}_{n+1}, \mathbf{q}_{n+2}, \dots, \mathbf{q}_m\}$  is also  $\perp \mathcal{C}(\mathbf{A})$  with dimension  $m - n$ . Thus,  $\{\mathbf{q}_{n+1}, \mathbf{q}_{n+2}, \dots, \mathbf{q}_m\}$  is an orthonormal basis for  $\mathcal{N}(\mathbf{A}^\top)$ . ■

### 3.7. Computing the Reduced QR Decomposition via the Gram-Schmidt Process

We write out this form of the reduced QR Decomposition such that  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  where  $\mathbf{Q} \in \mathbb{R}^{m \times n}$  and  $\mathbf{R} \in \mathbb{R}^{n \times n}$ :

$$\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_n] = [\mathbf{q}_1 \ \mathbf{q}_2 \ \dots \ \mathbf{q}_n] \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ 0 & & \ddots & \vdots \\ & & & r_{nn} \end{bmatrix}.$$

The orthonormal matrix  $\mathbf{Q}$  can be easily calculated by the Gram-Schmidt process. To see why we have the upper triangular matrix  $\mathbf{R}$ , we write out these equations

$$\begin{aligned} \mathbf{a}_1 &= r_{11}\mathbf{q}_1 &= \sum_{i=1}^1 r_{i1}\mathbf{q}_1, \\ \mathbf{a}_2 &= r_{12}\mathbf{q}_1 + r_{22}\mathbf{q}_2 &= \sum_{i=1}^2 r_{i2}\mathbf{q}_2, \\ \mathbf{a}_3 &= r_{13}\mathbf{q}_1 + r_{23}\mathbf{q}_2 + r_{33}\mathbf{q}_3 &= \sum_{i=1}^3 r_{i3}\mathbf{q}_3, \\ &\vdots & \\ \mathbf{a}_k &= r_{1k}\mathbf{q}_1 + r_{2k}\mathbf{q}_2 + \dots + r_{kk}\mathbf{q}_k &= \sum_{i=1}^k r_{ik}\mathbf{q}_k, \\ &\vdots & \\ \mathbf{a}_n &= r_{1n}\mathbf{q}_1 + r_{2n}\mathbf{q}_2 + \dots + r_{nn}\mathbf{q}_n &= \sum_{i=1}^n r_{in}\mathbf{q}_n, \end{aligned}$$

which coincides with the second equation of Equation (3.1) and conforms to the form of an upper triangular matrix  $\mathbf{R}$ . And if we extend the idea of Equation (3.1) into the  $k$ -th term, we will get

$$\begin{aligned} \mathbf{a}_k &= \sum_{i=1}^{k-1} (\mathbf{q}_i^\top \mathbf{a}_k) \mathbf{q}_i + \mathbf{a}_k^\perp \\ &= \sum_{i=1}^{k-1} (\mathbf{q}_i^\top \mathbf{a}_k) \mathbf{q}_i + \|\mathbf{a}_k^\perp\| \cdot \mathbf{q}_k, \end{aligned}$$

which implies we can gradually orthonormalize  $\mathbf{A}$  to an orthonormal set  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$  by

$$\begin{cases} r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, \quad \forall i \in \{1, 2, \dots, k-1\}; \\ \mathbf{a}_k^\perp = \mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i; \\ r_{kk} = \|\mathbf{a}_k^\perp\|; \\ \mathbf{q}_k = \mathbf{a}_k^\perp / r_{kk}. \end{cases} \quad (3.4)$$

The procedure is formulated in Algorithm 12.

---

**Algorithm 12** Reduced QR Decomposition via Gram-Schmidt Process

---

**Require:** Matrix  $\mathbf{A}$  has linearly independent columns with size  $m \times n$  and  $m \geq n$ ;

```

1: for $k = 1$ to n do ▷ compute k -th column of \mathbf{Q}, \mathbf{R}
2: for $i = 1$ to $k-1$ do ▷ entry (i, k) of \mathbf{R} , $2m-1$ flops
3: $r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k$; ▷ all $k-1$ iterations: $(k-1)(2m-1)$ flops
4: end for ▷ $2m(k-1)$ flops
5: $\mathbf{a}_k^\perp = \mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i$; ▷ main diagonal of \mathbf{R} , $2m$ flops
6: $r_{kk} = \|\mathbf{a}_k^\perp\|$; ▷ m flops
7: $\mathbf{q}_k = \mathbf{a}_k^\perp / r_{kk}$; ▷ m flops
8: end for
9: Output $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_n]$ and \mathbf{R} with entry (i, k) being r_{ik} ;

```

---

**Theorem 3.1: (Algorithm Complexity: Reduced QR via Gram-Schmidt)**

Algorithm 12 requires  $\sim 2mn^2$  flops to compute a reduced QR decomposition of an  $m \times n$  matrix with linearly independent columns and  $m \geq n$ .

**Proof** [of Theorem 3.1] In step 3, the computation of  $r_{ik}$  is a vector inner product which requires  $m$  multiplications, and  $m-1$  additions. This makes it  $\boxed{(k-1)(2m-1)}$  flops for all the  $k-1$  iterations.

In step 5, the computation of  $r_{ik}\mathbf{q}_i$  needs  $m$  flops and there are  $k-1$  such scalar-vector multiplications, which makes it  $m(k-1)$  flops. The vector subtraction and additions require another  $m(k-1)$  flops. Thus step 5 costs  $\boxed{2m(k-1)}$  flops.

In step 6, the vector norm involves a vector inner product plus a square root that takes  $\boxed{2m}$  flops.

Step 7 costs  $\boxed{m}$  for the divisions.

Therefore, for computing the  $k$ -th column of  $\mathbf{Q}, \mathbf{R}$ , it requires  $(k-1)(2m-1) + 2m(k-1) + 2m + m = 4mk - m - k + 1$  flops. Let  $f(k) = 4mk - m - k + 1$ , the total complexity can be obtained by

$$\text{cost} = f(n) + f(n-1) + \dots + f(1).$$

Simple calculations will show the total complexity is  $2mn^2 + mn - 3m - \frac{n^2-n}{2}$  flops, or  $2mn^2$  flops if we keep only the leading term. ■

## Orthogonal Projection

We notice again from Equation (3.4), i.e., step 2 to step 6 in Algorithm 12, the first two equality imply that

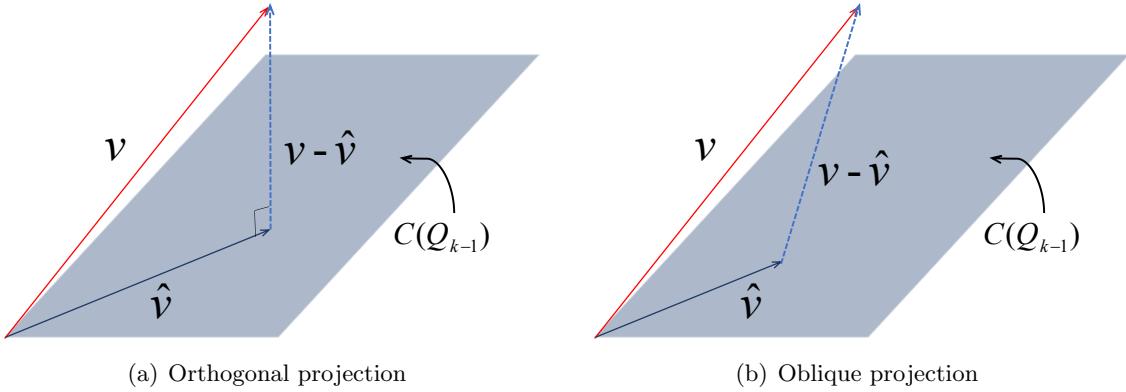
$$\left. \begin{array}{l} r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, \quad \forall i \in \{1, 2, \dots, k-1\} \\ \mathbf{a}_k^\perp = \mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i \end{array} \right\} \rightarrow \mathbf{a}_k^\perp = \mathbf{a}_k - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top \mathbf{a}_k = (\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top) \mathbf{a}_k, \quad (3.5)$$

where  $\mathbf{Q}_{k-1} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]$ . This implies  $\mathbf{q}_k$  can be obtained by

$$\mathbf{q}_k = \frac{\mathbf{a}_k^\perp}{\|\mathbf{a}_k^\perp\|} = \frac{(\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top) \mathbf{a}_k}{\|(\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top) \mathbf{a}_k\|}.$$

The matrix  $(\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top)$  in above equation is known as an *orthogonal projection matrix*<sup>4</sup> that will project  $\mathbf{a}_k$  along the column space of  $\mathbf{Q}_{k-1}$ , i.e., project a vector so that the vector is perpendicular to the column space of  $\mathbf{Q}_{k-1}$ . The net result is that the  $\mathbf{a}_k^\perp$  or  $\mathbf{q}_k$  calculated in this way will be orthogonal to the  $\mathcal{C}(\mathbf{Q}_{k-1})$ , i.e., in the null space of  $\mathbf{Q}_{k-1}^\top: \mathcal{N}(\mathbf{Q}_{k-1}^\top)$  by the fundamental theorem of linear algebra (Theorem 27.1, p. 428).

Let  $\mathbf{P}_1 = (\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top)$  and we claimed above  $\mathbf{P}_1 = (\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top)$  is an orthogonal projection matrix such that  $\mathbf{P}_1 \mathbf{v}$  will project the  $\mathbf{v}$  onto the null space of  $\mathbf{Q}_{k-1}$ . And actually, let  $\mathbf{P}_2 = \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top$ , then  $\mathbf{P}_2$  is also an orthogonal projection matrix such that  $\mathbf{P}_2 \mathbf{v}$  will project the  $\mathbf{v}$  onto the column space of  $\mathbf{Q}_{k-1}$ .



**Figure 3.4:** Demonstration of the difference between orthogonal projection and oblique projection.

But why do the matrix  $\mathbf{P}_1, \mathbf{P}_2$  can magically project a vector onto the corresponding subspaces? We will show in Lemma 7.1 that the column space of  $\mathbf{Q}_{k-1}$  is equal to the column space of  $\mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top$ :

$$\mathcal{C}(\mathbf{Q}_{k-1}) = \mathcal{C}(\mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top) = \mathcal{C}(\mathbf{P}_2).$$

<sup>4</sup> More details can be referred to Appendix D (p. 432).

Therefore, the result of  $\mathbf{P}_2\mathbf{v}$  is a linear combination of the columns of  $\mathbf{P}_2$ , which is in the column space of  $\mathbf{P}_2$  or the column space of  $\mathbf{Q}_{k-1}$ . The formal definition of a *projection matrix*  $\mathbf{P}$  is that it is idempotent  $\mathbf{P}^2 = \mathbf{P}$  such that projecting twice is equal to projecting once<sup>5</sup>. What makes the above  $\mathbf{P}_2 = \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top$  different is that the projection  $\hat{\mathbf{v}}$  of any vector  $\mathbf{v}$  is perpendicular to  $\mathbf{v} - \hat{\mathbf{v}}$ :

$$(\hat{\mathbf{v}} = \mathbf{P}_2\mathbf{v}) \perp (\mathbf{v} - \hat{\mathbf{v}}).$$

This goes to the original definition we gave above: the *orthogonal projection matrix*<sup>6</sup>. To avoid confusion, one may use the term *oblique projection matrix* in the nonorthogonal case where the difference is shown in Figure 3.4. When  $\mathbf{P}_2$  is an orthogonal projection matrix,  $\mathbf{P}_1 = \mathbf{I} - \mathbf{P}_2$  is also an orthogonal projection matrix that will project any vector onto the space perpendicular to the  $\mathcal{C}(\mathbf{Q}_{k-1})$ , i.e.,  $\mathcal{N}(\mathbf{Q}_{k-1}^\top)$ . Therefore, we conclude the two orthogonal projections:

$$\begin{cases} \mathbf{P}_1 : & \text{project onto } \mathcal{N}(\mathbf{Q}_{k-1}^\top); \\ \mathbf{P}_2 : & \text{project onto } \mathcal{C}(\mathbf{Q}_{k-1}). \end{cases}$$

The further result that is important to notice is when the columns of  $\mathbf{Q}_{k-1}$  are mutually orthonormal, we have the following decomposition:

$$\boxed{\mathbf{P}_1 = \mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top = (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^\top)(\mathbf{I} - \mathbf{q}_2\mathbf{q}_2^\top) \dots (\mathbf{I} - \mathbf{q}_{k-1}\mathbf{q}_{k-1}^\top),} \quad (3.6)$$

where  $\mathbf{Q}_{k-1} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]$  and each  $(\mathbf{I} - \mathbf{q}_i\mathbf{q}_i^\top)$  is to project a vector into the perpendicular space of  $\mathbf{q}_i$ .

### Modified Gram-Schmidt (MGS) Process

To emphasize the modified Gram-Schmidt process and make a connection to the equivalent projection in Equation (3.6), we first illustrate the lemma how the entries in the upper triangular  $\mathbf{R}$  of the QR decomposition can be obtained in an alternative way.

#### Lemma 3.2: (Modified Gram-Schmidt Process)

Suppose for  $[\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{k-1}, \mathbf{a}_k]$ , where the first  $k-1$  column are spanned by  $k-1$  orthonormal vectors  $[\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]$ :

$$\mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_i]) = \mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_i]), \quad \forall i \in \{1, 2, \dots, k-1\}.$$

Therefore,  $r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k$  is the projection of  $\mathbf{a}_k$  on the vector  $\mathbf{q}_i$ . Then it follows that

$$\begin{aligned} \mathbf{q}_i^\top \mathbf{a}_k &= \mathbf{q}_i^\top (\mathbf{a}_k - \underbrace{r_{1k}\mathbf{q}_1 - r_{2k}\mathbf{q}_2 - \dots - r_{i-1,k}\mathbf{q}_{i-1}}_{\text{orthogonal to } \mathbf{q}_i}), \quad \forall i \in \{1, 2, \dots, k-1\} \\ &= \mathbf{q}_i^\top (\mathbf{a}_k - \sum_{j=1}^{i-1} r_{jk}\mathbf{q}_j). \end{aligned}$$

5. See also Definition 27.5 (p. 434).

6. See also Definition 27.6 (p. 435).

This can be easily checked since  $\mathbf{q}_i$  is orthonormal to  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{i-1}\}$ . This observation implies another update on the  $k$ -th column of  $\mathbf{R}$ .

The lemma above reveals a second algorithm to compute the reduced QR decomposition of a matrix as shown in Algorithm 14 of which the algorithm on the left is exactly the same one as Algorithm 12 (with slight modification) to emphasize the difference.

| <b>Algorithm 13 CGS (=Algorithm 12 )</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | <b>Algorithm 14 MGS</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Require:</b> $\mathbf{A} \in \mathbb{R}^{m \times n}$ with full column rank;                                                                                                                                                                                                                                                                                                                                                                                                                          | <b>Require:</b> $\mathbf{A} \in \mathbb{R}^{m \times n}$ with full column rank;                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| <pre> 1: <b>for</b> <math>k = 1</math> to <math>n</math> <b>do</b> 2:   <math>\mathbf{a}_k^\perp = \mathbf{a}_k</math>; 3:   <b>for</b> <math>i = 1</math> to <math>k - 1</math> <b>do</b> 4:     <math>r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k</math>; 5:     <math>\mathbf{a}_k^\perp = \mathbf{a}_k^\perp - r_{ik} \mathbf{q}_i</math>; 6:   <b>end for</b> 7:   <math>r_{kk} = \ \mathbf{a}_k^\perp\ </math>; 8:   <math>\mathbf{q}_k = \mathbf{a}_k^\perp / r_{kk}</math>; 9: <b>end for</b> </pre> | <pre> 1: <b>for</b> <math>k = 1</math> to <math>n</math> <b>do</b> 2:   <math>\mathbf{a}_k^\perp = \mathbf{a}_k</math>; 3:   <b>for</b> <math>i = 1</math> to <math>k - 1</math> <b>do</b> 4:     <math>r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k^\perp</math>; 5:     <math>\mathbf{a}_k^\perp = \mathbf{a}_k^\perp - r_{ik} \mathbf{q}_i</math>; (*) 6:   <b>end for</b> 7:   <math>r_{kk} = \ \mathbf{a}_k^\perp\ </math>; 8:   <math>\mathbf{q}_k = \mathbf{a}_k^\perp / r_{kk}</math>; 9: <b>end for</b> </pre> |

The above process is named as the *modified Gram-Schmidt (MGS) process*, whereas the previous one is also known as the *classical Gram-Schmidt (CGS) process*. In theory, all CGS and MGS are equivalent in the sense that they compute exactly the same QR decompositions when exact arithmetic is employed. In practice, in the presence of round-off error, the orthonormal columns of  $\mathbf{Q}$  computed by MGS are often “more orthonormal” than those computed by CGS.

We notice that the equality (\*) in Algorithm 14 can be rewritten as (via the step 4 and step 5 of the algorithm)

$$\begin{aligned}
\mathbf{a}_k^\perp &:= \mathbf{a}_k^\perp - r_{ik} \mathbf{q}_i \\
&= \mathbf{a}_k^\perp - \mathbf{q}_i^\top \mathbf{a}_k^\perp \mathbf{q}_i \\
&= \mathbf{a}_k^\perp - \mathbf{q}_i \mathbf{q}_i^\top \mathbf{a}_k^\perp \\
&= (\mathbf{I} - \mathbf{q}_i \mathbf{q}_i^\top) \mathbf{a}_k^\perp.
\end{aligned}$$

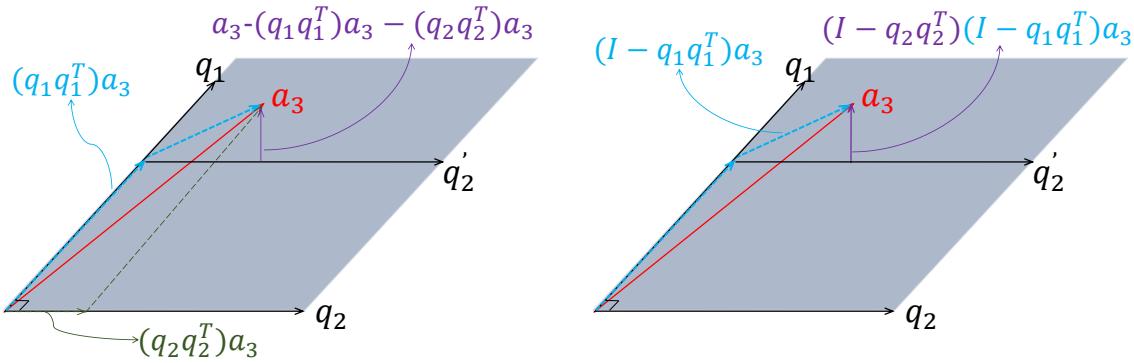
That is,  $\mathbf{a}_k^\perp$  will be updated by

$$\left\{ (\mathbf{I} - \mathbf{q}_{k-1} \mathbf{q}_{k-1}^\top) \dots [(\mathbf{I} - \mathbf{q}_2 \mathbf{q}_2^\top) ((\mathbf{I} - \mathbf{q}_1 \mathbf{q}_1^\top) \mathbf{a}_k)] \right\},$$

where the parentheses indicate the order of the computation, and which matches the orthogonal projection matrix equality in Equation (3.6) that

$$\begin{aligned}
\mathbf{P}_1 &= \mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top \\
&= (\mathbf{I} - \mathbf{q}_1 \mathbf{q}_1^\top) (\mathbf{I} - \mathbf{q}_2 \mathbf{q}_2^\top) \dots (\mathbf{I} - \mathbf{q}_{k-1} \mathbf{q}_{k-1}^\top) \\
&= \prod_{i=1}^{k-1} (\mathbf{I} - \mathbf{q}_i \mathbf{q}_i^\top),
\end{aligned}$$

where  $\mathbf{Q}_{k-1} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]$ .



(a) CGS, step 1: blue vector; step 2: green vector; step 3: purple vector.  
 (b) MGS, step 1: blue vector; step 2: purple vector.

**Figure 3.5:** CGS vs MGS in 3-dimensional space where  $\mathbf{q}'_2$  is parallel to  $\mathbf{q}_2$  so that projecting on  $\mathbf{q}_2$  is equivalent to projecting on  $\mathbf{q}'_2$ .

**What's the difference?** Taking a three-column matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$  as an example. Suppose we have computed  $\{\mathbf{q}_1, \mathbf{q}_2\}$  such that  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2\} = \text{span}\{\mathbf{a}_1, \mathbf{a}_2\}$ . And we want to proceed to compute the  $\mathbf{q}_3$ .

In the CGS, the orthogonalization of column  $\mathbf{a}_n$  against column  $\{\mathbf{q}_1, \mathbf{q}_2\}$  is performed by projecting the original column  $\mathbf{a}_3$  of  $\mathbf{A}$  onto  $\{\mathbf{q}_1, \mathbf{q}_2\}$  respectively and subtracting at once:

$$\left\{ \begin{array}{l} \mathbf{a}_3^\perp = \mathbf{a}_3 - (\mathbf{q}_1^\top \mathbf{a}_3) \mathbf{q}_1 - (\mathbf{q}_2^\top \mathbf{a}_3) \mathbf{q}_2 \\ = \mathbf{a}_3 - (\mathbf{q}_1 \mathbf{q}_1^\top) \mathbf{a}_3 - \boxed{(\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3} \\ \mathbf{q}_3 = \frac{\mathbf{a}_3^\perp}{\|\mathbf{a}_3^\perp\|}, \end{array} \right. \quad (3.7)$$

as shown in Figure 3.5(a).

In the MGS, on the other hand, the components along each  $\{\mathbf{q}_1, \mathbf{q}_2\}$  are immediately subtracted out of the rest of the column  $\mathbf{a}_3$  as soon as the  $\{\mathbf{q}_1, \mathbf{q}_2\}$  are computed. Therefore the orthogonalization of column  $\mathbf{a}_3$  against  $\{\mathbf{q}_1, \mathbf{q}_2\}$  is not performed by projecting the original column  $\mathbf{a}_3$  against  $\{\mathbf{q}_1, \mathbf{q}_2\}$  as it is in CGS, but rather against a vector obtained by subtracting from that column  $\mathbf{a}_3$  of  $\mathbf{A}$  the components in the direction of  $\mathbf{q}_1, \mathbf{q}_2$  successively. This is important because the error components of  $\mathbf{q}_i$  in  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2\}$  will be smaller (we will further discuss in the next paragraphs).

More precisely, in the MGS the orthogonalization of column  $\mathbf{a}_3$  against  $\mathbf{q}_1$  is performed by subtracting the component of  $\mathbf{q}_1$  from the vector  $\mathbf{a}_3$ :

$$\mathbf{a}_3^{(1)} = (\mathbf{I} - \mathbf{q}_1 \mathbf{q}_1^\top) \mathbf{a}_3 = \mathbf{a}_3 - (\mathbf{q}_1 \mathbf{q}_1^\top) \mathbf{a}_3,$$

where  $\mathbf{a}_3^{(1)}$  is the component of  $\mathbf{a}_3$  lies in a space perpendicular to  $\mathbf{q}_1$ . And further step is performed by

$$\begin{aligned} \mathbf{a}_3^{(2)} &= (\mathbf{I} - \mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)} = \mathbf{a}_3^{(1)} - (\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)} \\ &= \mathbf{a}_3 - (\mathbf{q}_1 \mathbf{q}_1^\top) \mathbf{a}_3 - \boxed{(\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)}} \end{aligned} \quad (3.8)$$

where  $\mathbf{a}_3^{(2)}$  is the component of  $\mathbf{a}_3^{(1)}$  lies in a space perpendicular to  $\mathbf{q}_2$  and we highlight the difference to the CGS in Equation (3.7) by blue text. This net result is that  $\mathbf{a}_3^{(2)}$  is the component of  $\mathbf{a}_3$  lies in the space perpendicular to  $\{\mathbf{q}_1, \mathbf{q}_2\}$  as shown in Figure 3.5(b).

### Main difference and catastrophic cancellation

The key difference is that the  $\mathbf{a}_3$  can in general have large components in  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2\}$  in which case one starts with large values and ends up with small values with large relative errors in them. This is known as the problem of *catastrophic cancellation*. Whereas  $\mathbf{a}_3^{(1)}$  is in the direction perpendicular to  $\mathbf{q}_1$  and has only a small “error” component in the direction of  $\mathbf{q}_1$ . Compare the boxed text in Equation (3.7) and (3.8), it is not hard to see  $(\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)}$  in Equation (3.8) is more accurate by the above argument. And thus, because of the much smaller error in this projection factor, the MGS introduces less orthogonalization error at each subtraction step than that is in the CGS. In fact, it can be shown that the final  $\mathbf{Q}$  obtained in the CGS satisfies

$$\|\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top\| \leq O(\epsilon\kappa^2(\mathbf{A})),$$

where  $\kappa(\mathbf{A})$  is a value larger than 1 determined by  $\mathbf{A}$ . Whereas, in MGS, the error satisfies

$$\|\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top\| \leq O(\epsilon\kappa(\mathbf{A})).$$

That is, the  $\mathbf{Q}$  obtained in the MGS is more orthogonal. Therefore we summarize the difference between the CGS and MGS processes for obtaining  $\mathbf{q}_k$  via the  $k$ -th column  $\mathbf{a}_k$  of  $\mathbf{A}$  and the orthonormalized vectors  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$ :

(CGS) : obtain  $\mathbf{q}_k$  by normalizing  $\mathbf{a}_k^\perp = (\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top)\mathbf{a}_k$ ;

(MGS) : obtain  $\mathbf{q}_k$  by normalizing  $\mathbf{a}_k^\perp = \left\{ (\mathbf{I} - \mathbf{q}_{k-1}\mathbf{q}_{k-1}^\top) \dots \left[ (\mathbf{I} - \mathbf{q}_2\mathbf{q}_2^\top) \left( (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^\top)\mathbf{a}_k \right) \right] \right\}$ .

### Triangular Orthogonalization in CGS and MGS

We here illustrate that in the CGS or MGS, the orthogonal matrix  $\mathbf{Q}$  is obtained via a set of triangular matrices. For simplicity, we only discuss the situation in the CGS and follow up the three-column example where  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3] \in \mathbb{R}^{3 \times 3}$ . The above discussion shows that the mutually orthonormal vectors  $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}$  can be obtained as follows:

$$\begin{cases} \mathbf{q}_1 = \frac{\mathbf{a}_1}{r_{11}}; \\ \mathbf{q}_2 = \frac{\mathbf{a}_2 - r_{12}\mathbf{q}_1}{r_{22}}; \\ \mathbf{q}_3 = \frac{\mathbf{a}_3 - r_{13}\mathbf{q}_1 - r_{23}\mathbf{q}_2}{r_{33}}. \end{cases}$$

Whilst, the three mutually orthonormal vectors can be equivalently obtained by

$$\mathbf{Q}\mathbf{R}_3\mathbf{R}_2\mathbf{R}_1 = \mathbf{A} \quad \xrightarrow{\text{leads to}} \quad \mathbf{Q} = \mathbf{A}\mathbf{R}_1^{-1}\mathbf{R}_2^{-1}\mathbf{R}_3^{-1},$$

where

$$\mathbf{R}_3 = \begin{bmatrix} 1 & 0 & r_{13} \\ 0 & 1 & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix}, \quad \mathbf{R}_2 = \begin{bmatrix} 1 & r_{12} & 0 \\ 0 & r_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_1 = \begin{bmatrix} r_{11} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

such that

$$\mathbf{R}_3 \mathbf{R}_2 \mathbf{R}_1 = \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{13} \\ 0 & 0 & r_{33} \end{bmatrix}.$$

The above procedure  $\mathbf{A}\mathbf{R}_1^{-1}\mathbf{R}_2^{-1}\mathbf{R}_3^{-1}$  will obtain the  $\{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}$  in a successive manner where  $\mathbf{A}\mathbf{R}_1^{-1}$  will get  $\mathbf{q}_1$  into the first column of  $\mathbf{Q}$ ;  $(\mathbf{A}\mathbf{R}_1^{-1})\mathbf{R}_2^{-1}$  will get  $\mathbf{q}_2$  into the second column of  $\mathbf{Q}$ ; and  $(\mathbf{A}\mathbf{R}_1^{-1}\mathbf{R}_2^{-1})\mathbf{R}_3^{-1}$  will get  $\mathbf{q}_3$  into the third column of  $\mathbf{Q}$ . This is called the *triangular orthogonalization* in the Gram-Schmidt process. The triangular orthogonalization is problematic in the sense that the condition number of a triangular matrix ( $\mathbf{R}_1^{-1}, \mathbf{R}_2^{-1}, \mathbf{R}_3^{-1}$  in the above three-column example) can be anything. And the Gram-Schmidt process have a series of them where the condition number can grow very large so that the orthogonalization is not numerical stable.

### More to go, preliminaries for Householder and Givens methods

Although, we claimed here that the MGS usually works better than the CGS in practice. An example will be given in the sequel. The MGS can still fall victim to the *catastrophic cancellation* problem. Suppose in iteration  $k$  of the MGS Algorithm 14,  $\mathbf{a}_k$  is almost in the span of  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$ . This will result in that  $\mathbf{a}_k^\perp$  has only a small component that is perpendicular to  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$ , whereas the “error” component in the  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$  will be amplified and the net result is  $\mathbf{Q}$  will be less orthonormal. As discussed above, the main disadvantage in both the CGS and MGS can be described by that the algorithms find the orthogonal matrix  $\mathbf{Q}$  via the upper triangular  $\mathbf{R}$ , i.e., if  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , one obtains  $\mathbf{Q}$  by

$$\mathbf{Q} = \mathbf{A} \underbrace{\mathbf{R}_1^{-1} \mathbf{R}_2^{-1} \dots \mathbf{R}_n^{-1}}_{\mathbf{R}^{-1}}.$$

In this case, if we can find a successive set of orthogonal matrices  $\{\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_l\}$  such that  $\mathbf{Q}_l \dots \mathbf{Q}_2 \mathbf{Q}_1 \mathbf{A}$  is triangularized, then  $\mathbf{Q} = (\mathbf{Q}_l \dots \mathbf{Q}_2 \mathbf{Q}_1)^\top$  will be “more” orthogonal than the CGS or the MGS since the condition numbers for the orthogonal matrices are all 1. We will discuss this method in Section 3.13 and 3.15 via the Householder reflectors and the Givens rotations.

### Example for MGS vs CGS

For better understanding, we will show by a  $4 \times 3$  *Lauchli matrix* where the Lauchli matrix is an  $(n+1) \times n$  rectangular matrix that has ones on the top row and the parameter  $\epsilon = \sqrt{\epsilon_{\text{mach}}}$  on the diagonal starting at entry (2, 1) i.e., on the lower subdiagonal.

**Example 3.2 (MGS vs CGS)** Let  $\epsilon = \sqrt{\epsilon_{\text{mach}}}$  and consider the QR decomposition of the following matrix by CGS and MGS:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ \epsilon & 0 & 0 \\ 0 & \epsilon & 0 \\ 0 & 0 & \epsilon \end{bmatrix} = [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3].$$

Note that we will round  $1 + \epsilon_{\text{mach}}$  to 1 whenever encountered in the calculation. The CGS proceeds as follows:

- Compute  $\mathbf{q}_1$  of unit length so that  $\mathcal{C}([\mathbf{q}_1]) = \mathcal{C}([\mathbf{a}_1])$ :
  - Compute  $r_{11}$ :  $r_{11} = \|\mathbf{a}_1\| = \sqrt{1 + \epsilon_{mach}} \approx 1$ ;
  - Compute  $\mathbf{q}_1$ :  $\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|} = \mathbf{a}_1$ ;
- Compute  $\mathbf{q}_2$  of unit length so that  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2])$ :
  - Compute  $r_{12}$ :  $r_{12} = \mathbf{a}_2^\top \mathbf{q}_1 = 1$ ;
  - Compute  $\mathbf{a}_2^\perp$ :  $\mathbf{a}_2^\perp = \mathbf{a}_2 - r_{12}\mathbf{q}_1 = \mathbf{a}_2 - \mathbf{a}_1 = [0, -\epsilon, \epsilon, 0]^\top$ ;
  - Compute  $r_{22}$ :  $r_{22} = \|\mathbf{a}_2^\perp\| = \sqrt{2\epsilon_{mach}} = \sqrt{2\epsilon}$ ;
  - Compute  $\mathbf{q}_2$ :  $\mathbf{q}_2 = \frac{\mathbf{a}_2^\perp}{\|\mathbf{a}_2^\perp\|} = \frac{\mathbf{a}_2^\perp}{r_{22}} = [0, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0]^\top$ ;
- Compute  $\mathbf{q}_3$  of unit length so that  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ :
  - Compute  $r_{13}$ :  $r_{13} = \mathbf{a}_3^\top \mathbf{q}_1 = \mathbf{a}_3^\top \mathbf{a}_1 = 1$ ;
  - Compute  $r_{23}$ :  $r_{23} = \mathbf{a}_3^\top \mathbf{q}_2 = 0$ ;
  - Compute  $\mathbf{a}_3^\perp$ :  $\mathbf{a}_3^\perp = \mathbf{a}_3 - r_{13}\mathbf{q}_1 - r_{23}\mathbf{q}_2 = [0, -\epsilon, 0, \epsilon]^\top$ ;
  - Compute  $r_{33}$ :  $r_{33} = \|\mathbf{a}_3^\perp\| = \sqrt{2\epsilon_{mach}} = \sqrt{2\epsilon}$ ;
  - Compute  $\mathbf{q}_3$ :  $\mathbf{q}_3 = \frac{\mathbf{a}_3^\perp}{\|\mathbf{a}_3^\perp\|} = \frac{\mathbf{a}_3^\perp}{r_{33}} = [0, -\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}}]^\top$ ;
- This results in

$$A = \begin{bmatrix} 1 & 0 & 0 \\ \epsilon & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & \sqrt{2}\epsilon & 0 \\ 0 & 0 & \sqrt{2}\epsilon \end{bmatrix} = Q_1 R_1$$

Whilst, the MGS proceeds as follows:

- Compute  $\mathbf{q}_1$  of unit length so that  $\mathcal{C}([\mathbf{q}_1]) = \mathcal{C}([\mathbf{a}_1])$ :
  - Compute  $r_{11}$ :  $r_{11} = \|\mathbf{a}_1\| = \sqrt{1 + \epsilon_{mach}} \approx 1$ ;
  - Compute  $\mathbf{q}_1$ :  $\mathbf{q}_1 = \frac{\mathbf{a}_1}{\|\mathbf{a}_1\|} = \mathbf{a}_1$ ;
- Compute  $\mathbf{q}_2$  of unit length so that  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2])$ :
  - Compute  $r_{12}$ :  $r_{12} = \mathbf{a}_2^\top \mathbf{q}_1 = 1$ ;
  - Compute  $\mathbf{a}_2^\perp$ :  $\mathbf{a}_2^\perp = \mathbf{a}_2 - r_{12}\mathbf{q}_1 = \mathbf{a}_2 - \mathbf{a}_1 = [0, -\epsilon, \epsilon, 0]^\top$ ;
  - Compute  $r_{22}$ :  $r_{22} = \|\mathbf{a}_2^\perp\| = \sqrt{2\epsilon_{mach}} = \sqrt{2\epsilon}$ ;
  - Compute  $\mathbf{q}_2$ :  $\mathbf{q}_2 = \frac{\mathbf{a}_2^\perp}{\|\mathbf{a}_2^\perp\|} = \frac{\mathbf{a}_2^\perp}{r_{22}} = [0, -\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}, 0]^\top$ ;
  - Till now, we are still the same as the CGS;
- Compute  $\mathbf{q}_3$  of unit length so that  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ :
  - Compute  $r_{13}$ :  $r_{13} = \mathbf{a}_3^\top \mathbf{q}_1 = \mathbf{a}_3^\top \mathbf{a}_1 = 1$ ;
  - Compute temporary  $\mathbf{a}_3^\perp = \mathbf{a}_3 - r_{13}\mathbf{q}_1 = [0, -\epsilon, 0, \epsilon]^\top$ ;
  - Compute  $r_{23}$ :  $r_{23} = \mathbf{q}_2^\top \mathbf{a}_3^\perp = \frac{\epsilon}{\sqrt{2}}$ ;

- Compute final  $\mathbf{a}_3^\perp$ :  $\mathbf{a}_3^\perp = \underbrace{\mathbf{a}_3 - r_{13}\mathbf{q}_1}_{\text{the old } \mathbf{a}_3^\perp} - r_{23}\mathbf{q}_2 = [0, -\epsilon/2, -\epsilon/2, \epsilon]^\top$ ;
- Compute  $r_{33}$ :  $r_{33} = \|\mathbf{a}_3^\perp\| = \sqrt{2\epsilon_{mach}} = \frac{\sqrt{6}}{2}\epsilon$ ;
- Compute  $\mathbf{q}_3$ :  $\mathbf{q}_3 = \frac{\mathbf{a}_3^\perp}{\|\mathbf{a}_3^\perp\|} = \frac{\mathbf{a}_3^\perp}{r_{33}} = [0, -\frac{1}{\sqrt{6}}, -\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}]^\top$ ;

• This results in

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ \epsilon & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{6}} \\ 0 & 0 & \frac{1}{\sqrt{6}} \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & \sqrt{2}\epsilon & \frac{\epsilon}{2} \\ 0 & 0 & \frac{\sqrt{6}}{2}\epsilon \end{bmatrix} = \mathbf{Q}_2 \mathbf{R}_2.$$

We notice that

$$\mathbf{Q}_1^\top \mathbf{Q}_1 = \begin{bmatrix} 1 + \epsilon_{mach} & -\frac{1}{\sqrt{2}}\epsilon & -\frac{1}{\sqrt{6}}\epsilon \\ -\frac{1}{\sqrt{2}}\epsilon & 1 & \frac{1}{2} \\ -\frac{1}{\sqrt{2}}\epsilon & \frac{1}{2} & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_2^\top \mathbf{Q}_2 = \begin{bmatrix} 1 + \epsilon_{mach} & -\frac{1}{\sqrt{2}}\epsilon & -\frac{1}{\sqrt{6}}\epsilon \\ -\frac{1}{\sqrt{2}}\epsilon & 1 & 0 \\ -\frac{1}{\sqrt{6}}\epsilon & 0 & 1 \end{bmatrix},$$

which shows that  $\mathbf{Q}_2$  is better in the sense of orthogonality.  $\square$

### Row-Wise MGS, Recursive Algorithm

The algorithms introduced above in Algorithm 13 and 14 are to compute the entries in the upper triangular matrix  $\mathbf{R}$  element-wise and column-by-column. Suppose  $\mathbf{A}$  has column partition  $\mathbf{A} = [\mathbf{a}_1, \mathbf{A}_2]$  where  $\mathbf{A}_2 = [\mathbf{a}_2, \mathbf{a}_3, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times (n-1)}$ . Notice in the CGS Algorithm 13, the first row of  $\mathbf{R}$  can be obtained by

$$\left. \begin{array}{l} r_{11} = \|\mathbf{a}_1\| \\ r_{1k} = \mathbf{q}_1^\top \mathbf{a}_k, \quad \forall k \in \{2, 3, \dots, n\}. \end{array} \right\} \xrightarrow{\text{leads to}} \left\{ \begin{array}{l} r_{11} = \|\mathbf{a}_1\| \\ \mathbf{r}_{12}^\top = \mathbf{q}_1^\top \mathbf{A}_2, \quad \mathbf{r}_{12} = [r_{12}, r_{13}, \dots, r_{1n}]. \end{array} \right.$$

Therefore, the QR decomposition of  $\mathbf{A}$  is given by

$$\mathbf{A} = [\mathbf{a}_1 \quad \mathbf{A}_2] = [\mathbf{q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} r_{11} & \mathbf{r}_{12}^\top \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} = [r_{11}\mathbf{q}_1 \quad \mathbf{q}_1 \mathbf{r}_{12}^\top + \mathbf{Q}_2 \mathbf{R}_{22}],$$

where columns of  $\mathbf{Q}_2 \in \mathbb{R}^{m \times (n-1)}$  are mutually orthonormal and  $\mathbf{R}_{22} \in \mathbb{R}^{(n-1) \times (n-1)}$  is upper triangular. And this implies  $\mathbf{Q}_2 \mathbf{R}_{22}$  is the reduced QR decomposition of  $\mathbf{A}_2 - \mathbf{q}_1 \mathbf{r}_{12}^\top$  which reveals a recursive algorithm for the reduced QR decomposition of  $\mathbf{A}$ . This is actually the same as the MGS that subtracts each component in the span of  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$  when computing column  $k$  of  $\mathbf{Q}$  (i.e., equality (\*) in Algorithm 14). The process is formulated in Algorithm 15.

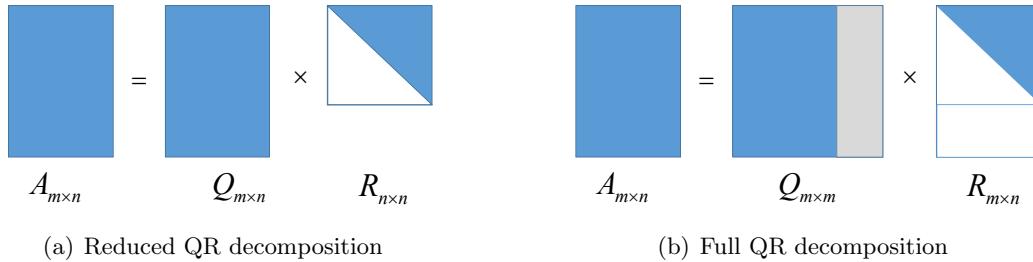
**Algorithm 15** MGS (Row-Wise and Recursively)=Algorithm 14

**Require:**  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with full column rank;

- 1: **for**  $k = 1$  to  $n$  **do** ▷ i.e., compute  $k$ -th column of  $\mathbf{Q}$  and  $k$ -th row of  $\mathbf{R}$
- 2:    $\mathbf{a}_1 = \mathbf{A}[:, 1]$ ; ▷ 1-st column of  $\mathbf{A} \in \mathbb{R}^{m \times (n-k+1)}$
- 3:    $r_{kk} = \|\mathbf{a}_1\|$ ; ▷  $\mathbf{a}_1 \in \mathbb{R}^{m \times 1}$
- 4:    $\mathbf{q}_k = \mathbf{a}_1 / r_{kk}$ ;
- 5:    $\mathbf{r}_{k2}^\top = \mathbf{q}_k^\top \mathbf{A}_2$ ; ▷  $\mathbf{A}_2 = \mathbf{A}[:, 2 : n] \in \mathbb{R}^{m \times (n-k)}$ ,  $\mathbf{r}_{k2}^\top \in \mathbb{R}^{1 \times (n-k)}$
- 6:    $\mathbf{A} = \mathbf{A}_2 - \mathbf{q}_k \mathbf{r}_{k2}^\top$ ; ▷  $\mathbf{A} \in \mathbb{R}^{m \times (n-k)}$
- 7: **end for**
- 8: Output  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_n]$  and  $\mathbf{R}$  with entry  $(i, k)$  being  $r_{ik}$ ;

### 3.8. Computing the Full QR Decomposition via the Gram-Schmidt Process

A full QR decomposition of an  $m \times n$  matrix with linearly independent columns goes further by appending additional  $m - n$  orthonormal columns to  $\mathbf{Q}$  so that it becomes an  $m \times m$  orthogonal matrix. In addition, rows of zeros are appended to  $\mathbf{R}$  so that it becomes an  $m \times n$  upper triangular matrix. We call the additional columns in  $\mathbf{Q}$  **silent columns** and additional rows in  $\mathbf{R}$  **silent rows**. The comparison between the reduced QR decomposition and the full QR decomposition is shown in Figure 3.6 where silent columns in  $\mathbf{Q}$  are denoted in gray, blank entries are zero and blue entries are elements that are not necessarily zero.



**Figure 3.6:** Comparison between the reduced and full QR decompositions.

### 3.9. Dependent Columns

Previously, we assumed matrix  $\mathbf{A}$  has linearly independent columns. However, this is not always necessary. Suppose in step  $k$  of Algorithm 12,  $\mathbf{a}_k$  is in the plane spanned by  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}$  which is equivalent to the space spanned by  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{k-1}$ , i.e., vectors  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$  are dependent. Then  $r_{kk}$  will be zero and  $\mathbf{q}_k$  does not exist because of the zero division. At this moment, we simply pick  $\mathbf{q}_k$  arbitrarily to be any normalized vector that is orthogonal to  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}])$  and continue the Gram-Schmidt process. Again, for matrix  $\mathbf{A}$  with dependent columns, we have both reduced and full QR decomposition

algorithms. We reformulate the step  $k$  in the algorithm as follows:

$$\mathbf{q}_k = \begin{cases} (\mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i) / r_{kk}, & r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, r_{kk} = \|\mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i\|, \text{ if } r_{kk} \neq 0, \\ \text{pick one in } \mathcal{C}^\perp([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]), & \text{if } r_{kk} = 0. \end{cases}$$

This idea can be further extended that when  $\mathbf{q}_k$  does not exist, we just skip the current steps. And add the silent columns in the end. In this sense, QR decomposition for a matrix with dependent columns is not unique. However, as long as you stick to a systematic process, QR decomposition for any matrix is unique.

This finding can also help to decide whether a set of vectors are linearly independent or not. Whenever  $r_{kk}$  in Algorithm 12 is zero, we report the vectors  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$  are dependent and stop the algorithm for “independent checking”.

### 3.10. QR with Column Pivoting: Column-Pivoted QR (CPQR)

Suppose  $\mathbf{A}$  has dependent columns, a column-pivoted QR (CPQR) decomposition can be found as follows.

#### Theorem 3.1: (Column-Pivoted QR Decomposition)

Every  $m \times n$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  with  $m \geq n$  and rank  $r$  can be factored as

$$\mathbf{AP} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is upper triangular,  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ ,  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  is an orthogonal matrix, and  $\mathbf{P}$  is a permutation matrix. This is also known as the **full** CPQR decomposition. Similarly, the **reduced** version is given by

$$\mathbf{AP} = \mathbf{Q}_r [\mathbf{R}_{11} \quad \mathbf{R}_{12}],$$

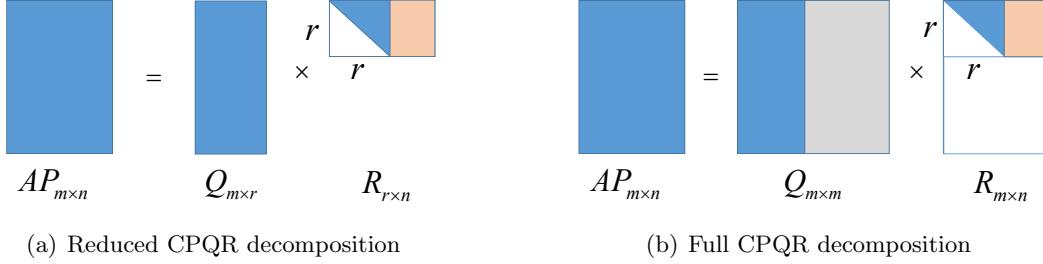
where  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is upper triangular,  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ ,  $\mathbf{Q}_r \in \mathbb{R}^{m \times r}$  contains orthonormal columns, and  $\mathbf{P}$  is a permutation matrix.

#### 3.10.1 A Simple CPQR via CGS

**A Simple CPQR via CGS** The classical Gram-Schmidt process can compute this CPQR decomposition. Following from the QR decomposition for dependent columns that when  $r_{kk} = 0$ , the column  $k$  of  $\mathbf{A}$  is dependent on the previous  $k-1$  columns. Whenever this happens, we permute this column into the last column and continue the Gram-Schmidt process. We notice that  $\mathbf{P}$  is the permutation matrix that interchanges the dependent columns into the last  $n-r$  columns. Suppose the first  $r$  columns of  $\mathbf{AP}$  are  $[\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \dots, \hat{\mathbf{a}}_r]$ , and the span of them is just the same as the span of  $\mathbf{Q}_r$  (in the reduced version), or the span of  $\mathbf{Q}_{:,r}$  (in the full version)

$$\mathcal{C}([\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \dots, \hat{\mathbf{a}}_r]) = \mathcal{C}(\mathbf{Q}_r) = \mathcal{C}(\mathbf{Q}_{:,r}).$$

And  $\mathbf{R}_{12}$  is a matrix that recovers the dependent  $n - r$  columns from the column space of  $\mathbf{Q}_r$  or column space of  $\mathbf{Q}_{:,r}$ . The comparison of reduced and full CPQR decomposition is shown in Figure 3.7 where silent columns in  $\mathbf{Q}$  are denoted in grey, blank entries are zero and blue/orange entries are elements that are not necessarily zero.



**Figure 3.7:** Comparison between the reduced and full CPQR decompositions.

---

**Algorithm 16 Simple** Reduced CPQR Decomposition via CGS

---

**Require:** Matrix  $\mathbf{A}$  with size  $m \times n$  and  $m \geq n$ ;

```

1: $cnt = 0$; \triangleright i.e., the count for the permutations
2: $\mathbf{q}_1 = \mathbf{a}_1/r_{11}, r_{11} = \|\mathbf{a}_1\|$; \triangleright i.e., the first column of \mathbf{R}_{11}
3: for $k = 2$ to n do \triangleright i.e., compute column k of \mathbf{R}_{11}
4: Set the initial value $r_{kk} = 0$;
5: while $r_{kk} == 0$ do \triangleright the column is dependent if r_{kk} is equal to 0
6: $r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, \forall i \in \{1, 2, \dots, k-1\}$; \triangleright first $k-1$ elements in column k of \mathbf{R}_{11}
7: $r_{kk} = \|\mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i\|$; \triangleright k -th elements in column k of \mathbf{R}_{11}
8: $\mathbf{q}_k = (\mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i)/r_{kk}$;
9: if $r_{kk} == 0$ then
10: $cnt = cnt + 1$;
11: Permute the column k to last column;
12: i.e., $[\mathbf{a}_k, \mathbf{a}_{k+1}, \dots, \mathbf{a}_n] \leftarrow [\mathbf{a}_{k+1}, \mathbf{a}_{k+2}, \dots, \mathbf{a}_n, \mathbf{a}_k]$;
13: end if
14: end while
15: if $k + cnt == n$ then
16: rank $r = k$;
17: for $k = r + 1$ to n do \triangleright i.e., compute column k of \mathbf{R}_{12}
18: $r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, \forall i \in \{1, 2, \dots, r\}$;
19: end for
20: end if
21: Output rank r , output $\mathbf{R}_{11}, \mathbf{R}_{12}, \mathbf{Q}_r = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_r]$. And exit the loop;
22: end for
```

---

The reduced algorithm is formulated in Algorithm 16. And it is trivial to find the last  $n - r$  columns of  $\mathbf{Q}$  in the orthogonal complement of  $\mathcal{C}(\mathbf{Q}_r)$ . Note step 6 of Algorithm 16 can be rewritten as  $\mathbf{Q}_{k-1}^\top \mathbf{a}_k$ , where  $\mathbf{Q}_{k-1} = [\mathbf{q}_1, \dots, \mathbf{q}_{k-1}]$ .

**Theorem 3.2: (Algorithm Complexity: Reduced CPQR)**

Algorithm 16 requires  $\sim 6mnr - 4mr^2$  flops to compute a CPQR decomposition of an  $m \times n$  matrix with  $m \geq n$  and rank  $r$ .

**Proof** [of Theorem 3.2] From Theorem 3.1, to compute  $\mathbf{R}_{11}$ , we just need to replace  $n$  in Theorem 3.1 by  $r$  such that the complexity of computing  $\mathbf{R}_{11}$  is  $\boxed{2mr^2}$  flops if keep only the leading term.

To compute  $\mathbf{R}_{12}$ , there are  $r \times (n-r)$  values, each taking  $2m-1$  flops ( $m$  multiplications and  $m-1$  additions) from step 18. That is  $\boxed{r(n-r)(2m-1)}$  flops.

The upper bound on steps 6, 7, 8 is set  $k-1 = r$  in these steps. This makes  $\boxed{(2m-1)r}$  flops for step 6 ( $mr$  multiplications and  $(m-1)r$  additions),  $\boxed{2m(r+1)}$  flops for step 7 ( $mr$  multiplications,  $(r-1)m$  additions,  $m$  subtractions,  $2m$  for the norm), and  $\boxed{m}$  flops for step 8. The total complexity for steps 6, 7, 8 is thus  $\boxed{4mr + 3m - r}$  flops. And there are  $n-r$  such iterations, which imply  $\boxed{(4mr + 3m - r)(n-r)}$  flops needed to find the dependent columns.

Therefore, the final complexity is

$$2mr^2 + r(n-r)(2m-1) + (4mr + 3m - r)(n-r).$$

And it is  $6mnr - 4mr^2$  flops if we keep only the leading term. ■

We notice that when  $r = n$ , the complexity for Algorithm 16 is  $2mn^2$  which agrees with the complexity of Algorithm 12.

### 3.10.2 A Practical CPQR via CGS

**A Practical CPQR via CGS** We notice that the simple CPQR algorithm pivot the first  $r$  independent columns into the first  $r$  columns of  $\mathbf{AP}$ . Let  $\mathbf{A}_1$  be the first  $r$  columns of  $\mathbf{AP}$ , and  $\mathbf{A}_2$  be the rest. Then, from the full CPQR, we have

$$[\mathbf{A}_1, \mathbf{A}_2] = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \left[ \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} \\ \mathbf{0} \end{bmatrix}, \mathbf{Q} \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{0} \end{bmatrix} \right].$$

It is not easy to see that

$$\|\mathbf{A}_2\| = \left\| \mathbf{Q} \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{0} \end{bmatrix} \right\| = \left\| \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{0} \end{bmatrix} \right\| = \|\mathbf{R}_{12}\|,$$

where the penultimate equality comes from the orthogonal equivalence under the matrix norm (Lemma 27.6, p. 480). Therefore, the norm of  $\mathbf{R}_{12}$  is decided by the norm of  $\mathbf{A}_2$ . When favoring well-conditioned CPQR,  $\mathbf{R}_{12}$  should be small in norm. And a practical CPQR decomposition is to permute columns of the matrix  $\mathbf{A}$  firstly such that the columns are ordered decreasingly in vector norm:

$$\tilde{\mathbf{A}} = \mathbf{AP}_0 = [\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_n}],$$

where  $\{j_1, j_2, \dots, j_n\}$  is a permuted index set of  $\{1, 2, \dots, n\}$  and

$$\|\mathbf{a}_{j_1}\| \geq \|\mathbf{a}_{j_2}\| \geq \dots \geq \|\mathbf{a}_{j_n}\|.$$

Then apply the “simple” reduced CPQR decomposition on  $\tilde{\mathbf{A}}$  such that  $\tilde{\mathbf{A}}\mathbf{P}_1 = \mathbf{Q}_r[\mathbf{R}_{11}, \mathbf{R}_{12}]$ . The “practical” reduced CPQR of  $\mathbf{A}$  is then recovered as

$$\mathbf{A} \underbrace{\mathbf{P}_0 \mathbf{P}_1}_{\mathbf{P}} = \mathbf{Q}_r[\mathbf{R}_{11}, \mathbf{R}_{12}].$$

When the  $l_2$  vector norm (i.e., inner product, Appendix L.1, p. 476) is applied, extra  $n(2m - 1)$  flops are required to compute the  $n$  norms of the column vectors and  $\frac{n(n-1)}{2}$  comparisons needed to determine the order of the norms.

### 3.10.3 A Practical CPQR via MGS

**A Practical CPQR via MGS** Now, based on the recursive MGS in Algorithm 15, we can also develop a practical CPQR. The algorithm is formulated in Algorithm 17 where the only difference to Algorithm 15 is highlighted in the blue text that we permute the column with the largest norm into the first column.

---

#### Algorithm 17 Practical CPQR via MGS (Row-Wise and Recursively)

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with exact rank  $r$ ;

- 1: **for**  $k = 1$  to  $n$  **do** ▷ i.e., compute  $k$ -th column of  $\mathbf{Q}$  and  $k$ -th row of  $\mathbf{R}$
  - 2:   **Find the column with largest norm in  $\mathbf{A}$ , and permute to first column;**
  - 3:    $\mathbf{a}_1 = \mathbf{A}[:, 1];$  ▷ 1-st column of  $\mathbf{A} \in \mathbb{R}^{m \times (n-k+1)}$
  - 4:    $r_{kk} = \|\mathbf{a}_1\|;$  ▷  $\mathbf{a}_1 \in \mathbb{R}^{m \times 1}$
  - 5:    $\mathbf{q}_k = \mathbf{a}_1 / r_{kk};$
  - 6:    $\mathbf{r}_{k2}^\top = \mathbf{q}_k^\top \mathbf{A}_2;$  ▷  $\mathbf{A}_2 = \mathbf{A}[:, 2:n] \in \mathbb{R}^{m \times (n-k)}$ ,  $\mathbf{r}_{k2}^\top \in \mathbb{R}^{1 \times (n-k)}$
  - 7:    $\mathbf{A} = \mathbf{A}_2 - \mathbf{q}_k \mathbf{r}_{k2}^\top;$  ▷  $\mathbf{A} \in \mathbb{R}^{m \times (n-k)}$
  - 8:   **Exit when  $r_{kk} = 0$ ;**
  - 9: **end for**
  - 10: Output  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_n]$  and  $\mathbf{R}$  with entry  $(i, k)$  being  $r_{ik};$
- 

The difference is in that, in each iteration, we need to all the norms of the columns of  $\mathbf{A}$  rather than compute the norm of it at once as that is in CGS. Suppose in iteration  $k$ , we need to compute the reduced QR decomposition of a matrix of size  $m \times (n - k + 1)$  if the original matrix  $\mathbf{A}$  is of size  $m \times n$ . That is, extra  $(n - k + 1)(2m - 1)$  flops required flops needed to do the CPQR via MGS. Let  $f(k) = (n - k + 1)(2m - 1)$ , simple calculation can show that additional complexity for CPQR via MGS is:

$$\text{extra cost} = f(1) + f(2) + \dots + f(n) \sim mn^2 \text{ flops,} \quad (3.9)$$

if only keep the leading term. This costs more than  $n(2m - 1)$  flops in the “practical” CPQR via CGS.

But do not worry, this extra cost in CPQR via MGS can be partially solved. Suppose the column partition of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ , and each squared norm of the columns are given in the vector

$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_n \end{bmatrix} = \begin{bmatrix} \|\mathbf{a}_1\|^2 \\ \|\mathbf{a}_2\|^2 \\ \vdots \\ \|\mathbf{a}_n\|^2 \end{bmatrix}.$$

Suppose further  $\mathbf{q} \in \mathbb{R}^m$  is a unit-length vector such that  $\mathbf{q}^\top \mathbf{q} = 1$  and  $\mathbf{r} \in \mathbb{R}^n$  is a trivial vector given by

$$\mathbf{r} = \mathbf{A}^\top \mathbf{q} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}. \quad (\text{similar to step 6 of above Algorithm 17})$$

Let further  $\mathbf{B} = \mathbf{A} - \mathbf{qr}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$  (similar to step 7 of above Algorithm 17). Then the squared length vector of  $\mathbf{B}$  is given by

$$\mathbf{l} = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} = \begin{bmatrix} \|\mathbf{b}_1\|^2 \\ \|\mathbf{b}_2\|^2 \\ \vdots \\ \|\mathbf{b}_n\|^2 \end{bmatrix} = \begin{bmatrix} l_1 - r_1^2 \\ l_2 - r_2^2 \\ \vdots \\ l_n - r_n^2 \end{bmatrix}.$$

This can be easily checked since  $\mathbf{b}_i = \mathbf{a}_i - r_i \mathbf{q} = \mathbf{a}_i - (\mathbf{a}_i^\top \mathbf{q}) \mathbf{q}$  such that

$$\|\mathbf{b}_i\|^2 = \|\mathbf{a}_i - r_i \mathbf{q}\|^2 = (\mathbf{a}_i - r_i \mathbf{q})^\top (\mathbf{a}_i - r_i \mathbf{q}) = l_i - r_i^2.$$

Come back to step 2 of Algorithm 17. Suppose we have computed squared norm of the original matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  (which takes  $n(2m - 1)$ , the same as that in “practical” CPQR via CGS). The squared norm of  $\mathbf{A}_2 - \mathbf{q}_1 \mathbf{r}_{12}^\top$  (suppose  $k = 1$  in step 7 of Algorithm 17) can be obtained by extra  $2(n - 1)$  flops. And for all the  $n$  iterations, the cost is  $2(n - 1) + 2(n - 2) + \dots + 2(1) = n^2 - n$  flops. This is much less than  $\sim mn^2$  in Equation (3.9).

### 3.10.4 Partial Factorization for CPQR: Extra Bonus of CPQR via MGS

**Partial Factorization for CPQR** The extra bonus for the CPQR via MGS is that we can do partial factorization at some point. We notice that at step of Algorithm 17, we permute the column with the largest norm into the first column, and step 4 of Algorithm 17 is to compute the norm into the main diagonal of the upper triangular  $\mathbf{R}$ . When  $\mathbf{A}$  has full rank  $n$ ,  $r_{kk}$  in all iterations will be positive. When  $\mathbf{A}$  has an “exact” rank  $r$ , the algorithm will stop after iteration  $r$ . However, when  $\mathbf{A}$  has an “effective” rank  $r$  with rank deficiency <sup>7</sup>, the algorithm can proceed well in the first  $r$  iterations since  $\{r_{11}, r_{22}, \dots, r_{rr}\}$

<sup>7</sup>. Effective rank, or also known as the numerical rank. Assume the  $i$ -th largest singular value of  $\mathbf{A}$  is denoted as  $\sigma_i(\mathbf{A})$ . Then if  $\sigma_r(\mathbf{A}) \gg \sigma_{r+1}(\mathbf{A}) \approx 0$ ,  $r$  is known as the numerical rank of  $\mathbf{A}$ . The singular value of matrix  $\mathbf{A}$  will be introduced in the SVD section (Section 14, p. 264). Whereas, when  $\sigma_i(\mathbf{A}) > \sigma_{r+1}(\mathbf{A}) = 0$ , it is known as having exact rank  $r$  as we have used in most of our discussions.

are relatively large in value that are far from 0. When it comes to iteration  $k = r + 1, \dots$ , the  $r_{kk}$  is small which means the column  $k$  of  $\mathbf{AP}$  has a small component in the direction of  $\mathbf{q}_k$  and is “almost” dependent on the previous  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$ . The situation is the same for the rest  $\{k + 1, k + 2, \dots\}$  columns since  $r_{kk} \geq r_{k+1,k+1} \geq \dots r_{nn}$  in iteration  $k$ . The partial factorization CPQR via MGS is formulated in Algorithm 18. This is related to the *rank-revealing QR decomposition (RRQR)* that we will introduce in the next sections. The algorithm will result in the factorization

$$\mathbf{AP} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix}, \quad (3.10)$$

where  $\mathbf{R}_{22}$  is small in norm.

However, in the “practical” CPQR with CGS (Algorithm 16 with  $\mathbf{A}$  permuted at once at the beginning of the procedure), it is wasteful when  $r \ll \min(m, n)$ . Since we only permute the column with the largest norm into the beginning of  $\mathbf{AP}$ . A value  $r_{kk}$  close to 0 only means column  $k$  of  $\mathbf{AP}$  is almost dependent on  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$ , and it does not mean the rest  $\{k + 1, k + 2, \dots\}$  columns of  $\mathbf{AP}$  are also almost dependent on  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$ .

---

**Algorithm 18** Practical and Partial CPQR via MGS (Row-Wise and Recursively)

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank deficiency;

- 1: Select a stopping criteria  $\delta$ ;
  - 2: **for**  $k = 1$  to  $n$  **do** ▷ i.e., compute  $k$ -th column of  $\mathbf{Q}$  and  $k$ -th row of  $\mathbf{R}$
  - 3:   **Find the column with largest norm in  $\mathbf{A}$ , and permute to first column;**
  - 4:    $\mathbf{a}_1 = \mathbf{A}[:, 1]$ ; ▷ 1-st column of  $\mathbf{A} \in \mathbb{R}^{m \times (n-k+1)}$
  - 5:    $r_{kk} = \|\mathbf{a}_1\|$ ; ▷  $\mathbf{a}_1 \in \mathbb{R}^{m \times 1}$
  - 6:    $\mathbf{q}_k = \mathbf{a}_1 / r_{kk}$ ;
  - 7:    $\mathbf{r}_{k2}^\top = \mathbf{q}_k^\top \mathbf{A}_2$ ; ▷  $\mathbf{A}_2 = \mathbf{A}[:, 2 : n] \in \mathbb{R}^{m \times (n-k)}$ ,  $\mathbf{r}_{k2}^\top \in \mathbb{R}^{1 \times (n-k)}$
  - 8:    $\mathbf{A} = \mathbf{A}_2 - \mathbf{q}_k \mathbf{r}_{k2}^\top$ ; ▷  $\mathbf{A} \in \mathbb{R}^{m \times (n-k)}$
  - 9:   **Exit when  $r_{kk} < \delta$ , set effective rank  $r = k$ ;**
  - 10: **end for**
  - 11: Output  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_n]$ ,  $\mathbf{R}$  with entry  $(i, k)$  being  $r_{ik}$ , and **effective rank  $r$** ;
- 

### 3.11. QR with Column Pivoting: Revealing Rank One Deficiency

We notice that Algorithm 16 is just one method to find the column permutation where  $\mathbf{A}$  is rank deficient and we interchange the first linearly independent  $r$  columns of  $\mathbf{A}$  into the first  $r$  columns of the  $\mathbf{AP}$ . If  $\mathbf{A}$  is nearly rank-one deficient and we would like to find a column permutation of  $\mathbf{A}$  such that the resulting pivotal element  $r_{nn}$  of the QR decomposition is small. This is known as the *revealing rank-one deficiency* problem.

**Theorem 3.1: (Revealing Rank One Deficiency, (Chan, 1987))**

If  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{v} \in \mathbb{R}^n$  is a unit 2-norm vector (i.e.,  $\|\mathbf{v}\| = 1$ ), then there exists a permutation  $\mathbf{P}$  such that the reduced QR decomposition

$$\mathbf{AP} = \mathbf{QR}$$

satisfies  $r_{nn} \leq \sqrt{n}\epsilon$  where  $\epsilon = \|\mathbf{Av}\|$  and  $r_{nn}$  is the  $n$ -th diagonal of  $\mathbf{R}$ . Note that  $\mathbf{Q} \in \mathbb{R}^{m \times n}$  and  $\mathbf{R} \in \mathbb{R}^{n \times n}$  in the reduced QR decomposition.

**Proof** [of Theorem 3.1] Suppose  $\mathbf{P} \in \mathbb{R}^{n \times n}$  is a permutation matrix such that if  $\mathbf{w} = \mathbf{P}^\top \mathbf{v}$  where

$$|w_n| = \max |v_i|, \quad \forall i \in \{1, 2, \dots, n\},$$

i.e., interchange the largest magnitude into the last entry such that the last component of  $\mathbf{w}$  is equal to the maximal component of  $\mathbf{v}$  in absolute value. Then we have  $|w_n| \geq 1/\sqrt{n}$ . Suppose the QR decomposition of  $\mathbf{AP}$  is  $\mathbf{AP} = \mathbf{QR}$ , then

$$\epsilon = \|\mathbf{Av}\| = \|(\mathbf{Q}^\top \mathbf{AP})(\mathbf{P}^\top \mathbf{v})\| = \|\mathbf{R}\mathbf{w}\| = \begin{bmatrix} \vdots \\ r_{nn}w_n \end{bmatrix} \geq |r_{nn}w_n| \geq |r_{nn}|/\sqrt{n},$$

where the second equality above is from the length preservation under orthogonal transformation and  $\mathbf{P}$  is orthogonal such that  $\mathbf{PP}^\top = \mathbf{I}$ . This completes the proof. ■

The following discussion is based on the existence of the singular value decomposition (SVD) which will be introduced in Section 14 (p. 264). Feel free to skip at a first reading. Suppose the SVD of  $\mathbf{A}$  is given by  $\mathbf{A} = \sum_{i=1}^n \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ , where  $\sigma_i$ 's are the singular values with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ , i.e.,  $\sigma_n$  is the smallest singular value, and  $\mathbf{u}_i$ 's,  $\mathbf{v}_i$ 's are left and right singular vectors respectively. Then, if we let  $\mathbf{v} = \mathbf{v}_n$  such that  $\mathbf{Av}_n = \sigma_n \mathbf{u}_n$ ,<sup>8</sup> we have

$$\|\mathbf{Av}\| = \sigma_n.$$

By constructing a permutation matrix  $\mathbf{P}$  such that

$$|\mathbf{P}^\top \mathbf{v}|_n = \max |\mathbf{v}_i|, \quad \forall i \in \{1, 2, \dots, n\},$$

we will find a QR decomposition of  $\mathbf{A} = \mathbf{QR}$  with a pivot  $r_{nn}$  smaller than  $\sqrt{n}\sigma_n$ . If  $\mathbf{A}$  is rank-one deficient, then  $\sigma_n$  will be close to 0 and  $r_{nn}$  is thus bounded to a small value in magnitude which is close to 0.

### 3.12. QR with Column Pivoting: Revealing Rank $r$ Deficiency\*

Following from the last section, suppose now we want to compute the reduced QR decomposition where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is nearly rank  $r$  deficient<sup>9</sup> with  $r > 1$ . Our goal now is to find a permutation  $\mathbf{P}$  such that

$$\mathbf{AP} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} \mathbf{L} & \mathbf{M} \\ \mathbf{0} & \mathbf{N} \end{bmatrix}, \quad (3.11)$$

8. We will prove that the right singular vector of  $\mathbf{A}$  is equal to the right singular vector of  $\mathbf{R}$  if the  $\mathbf{A}$  has QR decomposition  $\mathbf{A} = \mathbf{QR}$  in Lemma 14.4 (p. 272). The claim can also be applied to the singular values. So  $\mathbf{v}_n$  here is also the right singular vector of  $\mathbf{R}$ .

9. Note that rank  $r$  here does not mean the matrix has rank  $r$ , but rather it has rank  $(\min\{m, n\} - r)$ .

<sup>10</sup> where  $\mathbf{N} \in \mathbb{R}^{r \times r}$  and  $\|\mathbf{N}\|$  is small in some norm (and  $\mathbf{L} \in \mathbb{R}^{(n-r) \times (n-r)}$ ,  $\mathbf{M} \in \mathbb{R}^{(n-r) \times r}$  that can be inferred from context).

A recursive algorithm can be applied to do so. Suppose we have already isolated a small  $k \times k$  block  $\mathbf{N}_k$ , based on which, if we can isolate a small  $(k+1) \times (k+1)$  block  $\mathbf{N}_{k+1}$ , then we can find the permutation matrix recursively. To repeat, suppose we have the permutation  $\mathbf{P}_k$  such that the  $\mathbf{N}_k \in \mathbb{R}^{k \times k}$  has a small norm,

$$\mathbf{A}\mathbf{P}_k = \mathbf{Q}_k \mathbf{R}_k = \mathbf{Q}_k \begin{bmatrix} \mathbf{L}_k & \mathbf{M}_k \\ \mathbf{0} & \mathbf{N}_k \end{bmatrix}.$$

We want to find a permutation  $\mathbf{P}_{k+1}$ , such that  $\mathbf{N}_{k+1} \in \mathbb{R}^{(k+1) \times (k+1)}$  also has a small norm,

$$\boxed{\mathbf{A}\mathbf{P}_{k+1} = \mathbf{Q}_{k+1} \mathbf{R}_{k+1} = \mathbf{Q}_{k+1} \begin{bmatrix} \mathbf{L}_{k+1} & \mathbf{M}_{k+1} \\ \mathbf{0} & \mathbf{N}_{k+1} \end{bmatrix}}.$$

From the algorithm introduced in the last section, there is an  $(n-k) \times (n-k)$  permutation matrix  $\tilde{\mathbf{P}}_{k+1}$  such that  $\mathbf{L}_k \in \mathbb{R}^{(n-k) \times (n-k)}$  has the QR decomposition  $\mathbf{L}_k \tilde{\mathbf{P}}_{k+1} = \tilde{\mathbf{Q}}_{k+1} \tilde{\mathbf{L}}_k$  such that the entry  $(n-k, n-k)$  of  $\tilde{\mathbf{L}}_k$  is small. By constructing

$$\mathbf{P}_{k+1} = \mathbf{P}_k \begin{bmatrix} \tilde{\mathbf{P}}_{k+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{Q}_{k+1} = \mathbf{Q}_k \begin{bmatrix} \tilde{\mathbf{Q}}_{k+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix},$$

we have

$$\boxed{\mathbf{A}\mathbf{P}_{k+1} = \mathbf{Q}_{k+1} \begin{bmatrix} \tilde{\mathbf{L}}_k & \tilde{\mathbf{Q}}_{k+1}^\top \mathbf{M}_k \\ \mathbf{0} & \mathbf{N}_k \end{bmatrix}}.$$

We know that entry  $(n-k, n-k)$  of  $\tilde{\mathbf{L}}_k$  is small, if we can prove the last row of  $\tilde{\mathbf{Q}}_{k+1}^\top \mathbf{M}_k$  is small in norm, then we find the QR decomposition revealing rank  $k+1$  deficiency (see (Chan, 1987) for a proof). And the procedure is formulated in Algorithm 19.

### 3.13. Existence of the QR Decomposition via the Householder Reflector Householder Reflectors

We first give the formal definition of a Householder reflector and we will take a look at its properties.

#### Definition 3.1: Householder Reflector

Let  $\mathbf{u} \in \mathbb{R}^n$  be a vector of unit length (i.e.,  $\|\mathbf{u}\| = 1$ ). Then  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$  is said to be a *Householder reflector*, a.k.a., a *Householder transformation*. We call this  $\mathbf{H}$  the Householder reflector associated with the unit vector  $\mathbf{u}$  where the unit vector  $\mathbf{u}$  is also known as *Householder vector*. If a vector  $\mathbf{x}$  is multiplied by  $\mathbf{H}$ , then it is reflected in the hyperplane  $\text{span}\{\mathbf{u}\}^\perp$ .

<sup>10</sup>. To abuse the notation, we use the notation  $\mathbf{L}, \mathbf{M}, \mathbf{N}$  for clarity on the derivation. It is better to replace  $\mathbf{L}, \mathbf{M}, \mathbf{N}$  by  $\mathbf{R}_{11}, \mathbf{R}_{12}, \mathbf{R}_{22}$  to match other contexts.

**Algorithm 19** Reveal Rank  $r$  Deficiency

**Require:** Matrix  $\mathbf{A}$  with size  $m \times n$  and  $m \geq n$ , and  $\text{rank}(\mathbf{A}) = n - r$ ;

- 1: Initialize  $\mathbf{W} \in \mathbb{R}^{n \times r}$  to zero; ▷ store the singular vectors
- 2: Initial QR decomposition by  $\mathbf{A} = \mathbf{QR}$ ;
- 3: **for**  $i = n$  to  $n - r + 1$  **do**
- 4:    $\mathbf{L} \leftarrow$  leading  $i \times i$  block of  $\mathbf{R}$ ;
- 5:   Compute the singular vector  $\mathbf{v} \in \mathbb{R}^i$  corresponding to the min singular value of  $\mathbf{L}$ ;
- 6:   Compute a permutation  $\tilde{\Pi} \in \mathbb{R}^{i \times i}$  such that  $|\tilde{\Pi}^\top \mathbf{v}|_i = \max_j |\mathbf{v}_j|, \forall j \in \{1, 2, \dots, i\}$ ;
- 7:   Assign  $\begin{bmatrix} \mathbf{v} \\ \mathbf{0} \end{bmatrix}$  to the  $i$ -th column of  $\mathbf{W}$ ;
- 8:   Compute  $\mathbf{W} \leftarrow \tilde{\Pi}^\top \mathbf{W}$ , where  $\tilde{\Pi} = \begin{bmatrix} \tilde{\Pi} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$ ;
- 9:   Compute the QR decomposition:  $\mathbf{L}\tilde{\Pi} = \tilde{\mathbf{Q}}\tilde{\mathbf{L}}$ ;
- 10:    $\mathbf{P} \leftarrow \mathbf{P}\tilde{\Pi}$ ;
- 11:    $\mathbf{Q} \leftarrow \mathbf{Q} \begin{bmatrix} \tilde{\mathbf{Q}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$ ;
- 12:    $\mathbf{R} \leftarrow \begin{bmatrix} \tilde{\mathbf{L}} & \tilde{\mathbf{Q}}^\top \mathbf{B} \\ \mathbf{0} & \mathbf{C} \end{bmatrix}$ , where  $\mathbf{B} = \mathbf{R}_{1:n-i, n-i+1:n}$ , and  $\mathbf{C} = \mathbf{R}_{n-i+1:n, n-i+1:n}$ ;
- 13: **end for**

Note that if  $\|\mathbf{u}\| \neq 1$ , we can define  $\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top \mathbf{u}}$  as the Householder reflector.

From the definition of the Householder reflector, then we have the following corollary that a special kind of vectors will remain unchanged under the Householder reflector.

**Corollary 3.2: (Unreflected by Householder)**

Suppose  $\|\mathbf{u}\| = 1$ , and define the Householder reflector  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ . Then any vector  $\mathbf{v}$  that is perpendicular to  $\mathbf{u}$  is left unchanged by the Householder transformation, that is,  $\mathbf{H}\mathbf{v} = \mathbf{v}$  if  $\mathbf{u}^\top \mathbf{v} = 0$ .

The proof is trivial that  $(\mathbf{I} - 2\mathbf{u}\mathbf{u}^\top)\mathbf{v} = \mathbf{v} - 2\mathbf{u}\mathbf{u}^\top\mathbf{v} = \mathbf{v}$ .

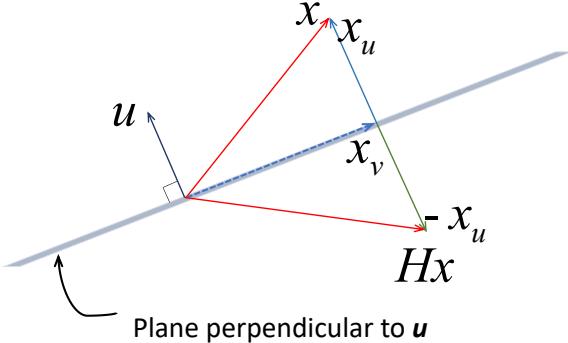
Suppose  $\mathbf{u}$  is a unit vector with  $\|\mathbf{u}\| = 1$ , and a vector  $\mathbf{v}$  is perpendicular to  $\mathbf{u}$ . Then any vector  $\mathbf{x}$  on the plane can be decomposed into two parts

$$\mathbf{x} = \mathbf{x}_v + \mathbf{x}_u,$$

where the first one  $\mathbf{x}_u$  is parallel to  $\mathbf{u}$  and the second one  $\mathbf{x}_v$  is perpendicular to  $\mathbf{u}$  (i.e., parallel to  $\mathbf{v}$ ). From Section 3.2 on the projection of a vector onto another one,  $\mathbf{x}_u$  can be computed by  $\mathbf{x}_u = \frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top \mathbf{u}}\mathbf{x} = \mathbf{u}\mathbf{u}^\top \mathbf{x}$ , i.e., the projection of  $\mathbf{x}$  onto the vector  $\mathbf{u}$ . We then transform this  $\mathbf{x}$  by the Householder reflector associated with  $\mathbf{u}$ ,

$$\mathbf{H}\mathbf{x} = (\mathbf{I} - 2\mathbf{u}\mathbf{u}^\top)(\mathbf{x}_v + \mathbf{x}_u) = \mathbf{x}_v - \mathbf{u}\mathbf{u}^\top \mathbf{x} = \mathbf{x}_v - \mathbf{x}_u,$$

**Figure 3.8:** Demonstration of the Householder reflector. The Householder reflector obtained by  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$  where  $\|\mathbf{u}\| = 1$  will reflect vector  $\mathbf{x}$  along the plane perpendicular to  $\mathbf{u}$ :  $\mathbf{x} = \mathbf{x}_v + \mathbf{x}_u \rightarrow \mathbf{x}_v - \mathbf{x}_u$ .



i.e., the Householder reflector transforms  $\mathbf{x}_v + \mathbf{x}_u$  into  $\mathbf{x}_v - \mathbf{x}_u$ . That is, the space perpendicular to  $\mathbf{u}$  acts as a mirror and any vector  $\mathbf{x}$  is reflected by the Householder reflector associated with  $\mathbf{u}$  (i.e., reflected in the hyperplane  $\text{span}\{\mathbf{u}\}^\perp$ ). The situation is shown in Figure 3.8.

The above discussion tells us how to find the reflected vector given the Householder reflector. The further question can be posed that if we know two vectors are reflected to each other up front, the next corollary tells us how to find the corresponding Householder reflector. The property is important for computing the QR decomposition if we want to reflect a column into a specific form.

### Corollary 3.3: (Finding the Householder Reflector)

Suppose  $\mathbf{x}$  is reflected to  $\mathbf{y}$  by a Householder reflector with  $\|\mathbf{x}\| = \|\mathbf{y}\|$ , then the Householder reflector is obtained by

$$\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top, \text{ where } \mathbf{u} = \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|}.$$

**Proof** [of Corollary 3.3] Write out the equation, we have

$$\begin{aligned} \mathbf{H}\mathbf{x} &= \mathbf{x} - 2\mathbf{u}\mathbf{u}^\top\mathbf{x} = \mathbf{x} - 2\frac{(\mathbf{x} - \mathbf{y})(\mathbf{x}^\top - \mathbf{y}^\top)}{(\mathbf{x} - \mathbf{y})^\top(\mathbf{x} - \mathbf{y})}\mathbf{x} \\ &= \mathbf{x} - (\mathbf{x} - \mathbf{y}) = \mathbf{y}. \end{aligned}$$

Note that the condition  $\|\mathbf{x}\| = \|\mathbf{y}\|$  is required to prove the result. ■

The Householder reflectors are useful to set a block of components of a given vector to zero. Particularly, we usually would like to set the vector  $\mathbf{a} \in \mathbb{R}^n$  to be zero except the  $i$ -th element. Then the Householder vector can be chosen to be

$$\mathbf{u} = \frac{\mathbf{a} - r\mathbf{e}_i}{\|\mathbf{a} - r\mathbf{e}_i\|}, \quad \text{where } r = \pm\|\mathbf{a}\|$$

which is a reasonable Householder vector since  $\|\mathbf{a}\| = \|r\mathbf{e}_i\| = |r|$ . We carefully notice that when  $r = \|\mathbf{a}\|$ ,  $\mathbf{a}$  is reflected to  $\|\mathbf{a}\|\mathbf{e}_i$  via the Householder reflector  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ ; otherwise when  $r = -\|\mathbf{a}\|$ ,  $\|\mathbf{a}\|$  is reflected to  $-\|\mathbf{a}\|\mathbf{e}_i$  via the Householder reflector.

Recall that in Section 3.7 (p. 90), we claimed the Householder or Givens method is to employ a set of orthogonal matrices to triangularize the matrix such that the QR decomposition is obtained and the orthogonal matrix is “more” orthogonal in this sense. The Householder reflector is such orthogonal matrix for this purpose. We not provide some more properties of the Householder reflector in the following remark.

#### Remark 3.4: Householder Properties

If  $\mathbf{H}$  is a Householder reflector, then it has the following properties:

- $\mathbf{H}\mathbf{H} = \mathbf{I}$ ;
- $\mathbf{H} = \mathbf{H}^\top$ ;
- $\mathbf{H}^\top\mathbf{H} = \mathbf{H}\mathbf{H}^\top = \mathbf{I}$  such that Householder reflector is an orthogonal matrix;
- $\mathbf{H}\mathbf{u} = -\mathbf{u}$ , if  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ .

#### Orthogonal Triangularization

To repeat, we see in the Gram-Schmidt section that QR decomposition is to use a triangular matrix to orthogonalize a matrix  $\mathbf{A}$ . The further idea is that, if we have a set of orthogonal matrices that can make  $\mathbf{A}$  to be triangular step by step, then we can also recover the QR decomposition. Specifically, if we have an orthogonal matrix  $\mathbf{Q}_1$  that can introduce zeros to the 1-st column of  $\mathbf{A}$  except the entry (1,1); and an orthogonal matrix  $\mathbf{Q}_2$  that can introduce zeros to the 2-nd column except the entries (2,1), (2,2); .... Then, we can also find the QR decomposition. For the way to introduce zeros, we could reflect the columns of the matrix to a basis vector  $\mathbf{e}_1$  whose entries are all zero except the first entry.

Let  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$  be the column partition of  $\mathbf{A}$ , and let further

$$r_1 = \|\mathbf{a}_1\|, \quad \mathbf{u}_1 = \frac{\mathbf{a}_1 - r_1\mathbf{e}_1}{\|\mathbf{a}_1 - r_1\mathbf{e}_1\|}, \quad \text{and} \quad \mathbf{H}_1 = \mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^\top, \quad (3.12)$$

where  $\mathbf{e}_1$  here is the first basis for  $\mathbb{R}^m$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^m$ . Then

$$\mathbf{H}_1\mathbf{A} = [\mathbf{H}_1\mathbf{a}_1, \mathbf{H}_1\mathbf{a}_2, \dots, \mathbf{H}_1\mathbf{a}_n] = \begin{bmatrix} r_1 & \mathbf{R}_{1,2:n} \\ \mathbf{0} & \mathbf{B}_2 \end{bmatrix}, \quad (3.13)$$

which reflects  $\mathbf{a}_1$  to  $r_1\mathbf{e}_1$  and introduces zeros below the diagonal in the 1-st column. We observe that the entries below  $r_1$  are all zero now under this specific reflection. Notice that we reflect  $\mathbf{a}_1$  to  $\|\mathbf{a}_1\|\mathbf{e}_1$  the two of which have same length, rather than reflect  $\mathbf{a}_1$  to  $\mathbf{e}_1$  directly. This is for the purpose of **numerical stability** and matches the requirement in Corollary 3.3.

**Choice of  $r_1$ :** moreover, the choice of  $r_1$  is **not unique**. For **numerical stability**, it is also desirable to choose  $r_1 = -\text{sign}(a_{11})\|\mathbf{a}_1\|$ , where  $a_{11}$  is the first component of  $\mathbf{a}_1$ . Or even,  $r_1 = \text{sign}(a_{11})\|\mathbf{a}_1\|$  is also possible as long as  $\|\mathbf{a}_1\|$  is equal to  $\|r_1\mathbf{e}_1\|$ . However, we will not cover this topic here.

We can then apply this process to  $\mathbf{B}_2$  in Equation (3.13) to make the entries below the entry (2,2) to be all zeros. Note that, we do not apply this process to the entire  $\mathbf{H}_1\mathbf{A}$  but rather the submatrix  $\mathbf{B}_2$  in it because we have already introduced zeros in the first column, and reflecting again will introduce nonzero values back and destroy what have accomplished.

Suppose  $\mathbf{B}_2 = [\mathbf{b}_2, \mathbf{b}_3, \dots, \mathbf{b}_n]$  is the column partition of  $\mathbf{B}_2$ , and let

$$r_2 = \|\mathbf{b}_2\|, \quad \mathbf{u}_2 = \frac{\mathbf{b}_2 - r_2 \mathbf{e}_1}{\|\mathbf{b}_2 - r_2 \mathbf{e}_1\|}, \quad \widetilde{\mathbf{H}}_2 = \mathbf{I} - 2\mathbf{u}_2 \mathbf{u}_2^\top, \quad \text{and} \quad \mathbf{H}_2 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{H}}_2 \end{bmatrix},$$

where now  $\mathbf{e}_1$  here is the first basis for  $\mathbb{R}^{m-1}$  and  $\mathbf{H}_2$  is also an orthogonal matrix since  $\widetilde{\mathbf{H}}_2$  is an orthogonal matrix. Then it follows that

$$\widetilde{\mathbf{H}}_2 \mathbf{B}_2 = [\mathbf{H}_2 \mathbf{b}_2, \mathbf{H}_2 \mathbf{b}_3, \dots, \mathbf{H}_2 \mathbf{b}_n] = \begin{bmatrix} r_2 & \mathbf{R}_{2,3:n} \\ \mathbf{0} & \mathbf{C}_3 \end{bmatrix},$$

and

$$\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} = [\mathbf{H}_2 \mathbf{H}_1 \mathbf{a}_1, \mathbf{H}_2 \mathbf{H}_1 \mathbf{a}_2, \dots, \mathbf{H}_2 \mathbf{H}_1 \mathbf{a}_n] = \begin{bmatrix} r_1 & \mathbf{R}_{12} & \mathbf{R}_{1,3:n} \\ 0 & r_2 & \mathbf{R}_{2,3:n} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3 \end{bmatrix}.$$

Same process can go on, and if  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , after  $n$  stages we will finally triangularize  $\mathbf{A} = (\mathbf{H}_n \mathbf{H}_{n-1} \dots \mathbf{H}_1)^{-1} \mathbf{R} = \mathbf{Q} \mathbf{R}$ . Since the  $\mathbf{H}_i$ 's are symmetric and orthogonal (Remark 3.4), we have orthogonal  $\mathbf{Q} = (\mathbf{H}_n \mathbf{H}_{n-1} \dots \mathbf{H}_1)^{-1} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_n$ .

An example of a  $5 \times 4$  matrix is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{ccc} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_1} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_2} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \\ \mathbf{A} & \mathbf{H}_1 \mathbf{A} & \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \\ \xrightarrow{\mathbf{H}_3} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{0} & \boxtimes \\ 0 & 0 & \mathbf{0} & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_4} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} & & \mathbf{H}_4 \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \end{array}$$

**A closer look at the QR factorization** The Householder algorithm is a process that makes a matrix triangular by a sequence of orthogonal matrix operations. In the Gram-Schmidt process (both CGS and MGS), we use a triangular matrix to orthogonalize the matrix. However, in the Householder algorithm, we use orthogonal matrices to triangularize. The difference between the two approaches is then summarized as follows:

- Gram-Schmidt: triangular orthogonalization;
- Householder: orthogonal triangularization.

We further notice that, in the Householder algorithm or the Givens algorithm that we will shortly see, a set of orthogonal matrices are applied so that the QR decomposition obtained is a *full* QR decomposition. Whereas, the direct QR decomposition obtained by CGS or MGS is a *reduced* one (although the silent columns or rows can be further added to find the full one).

### 3.14. Computing the Full QR Decomposition via the Householder Reflector

Since  $\mathbf{A}$  has  $n$  columns, and for every step  $i \in \{1, 2, \dots, n\}$  to introduce zeros in the  $i$ -th column below the diagonal, we operate on a submatrix of size  $(m - i + 1) \times (n - i + 1)$ . To compute the upper triangular matrix  $\mathbf{R} = \mathbf{H}_n \mathbf{H}_{n-1} \dots \mathbf{H}_1 \mathbf{A}$ , we notice that

$$\begin{aligned}\mathbf{R} &= (\mathbf{H}_n \dots (\mathbf{H}_3(\mathbf{H}_2(\mathbf{H}_1\mathbf{A})))) \\ &= \begin{bmatrix} \mathbf{I}_{n-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_n\mathbf{u}_n^\top \end{bmatrix} \cdots \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_3\mathbf{u}_3^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_2\mathbf{u}_2^\top \end{bmatrix} [\mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^\top] \mathbf{A},\end{aligned}$$

where the parentheses indicate the order of the computation, the upper-left of  $\mathbf{H}_2$  is a  $1 \times 1$  identity matrix, and it will not change the **first row** and **first column** of  $\mathbf{H}_1\mathbf{A}$  from the “triangular property”; and the upper-left of  $\mathbf{H}_3$  is a  $2 \times 2$  identity matrix which will not change the **first 2 rows** and **first 2 columns** of  $\mathbf{H}_2\mathbf{H}_2\mathbf{A}$ ; .... This property yields the step 8 in Algorithm 20 which operates only on the  $i : m$  rows and  $i + 1 : n$  column of  $\mathbf{R}$  in step  $i$  (since the  $i$ -th column takes 1 flop explicitly in step 7 though this hardly reduces the complexity). After the Householder transformation, we output the final triangular matrix  $\mathbf{R}$ , and the process is shown in Algorithm 20.

Furthermore, in Algorithm 20, to get the final orthogonal matrix  $\mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_n$ , we notice that

$$\begin{aligned}\mathbf{Q} &= (((\mathbf{H}_1 \mathbf{H}_2) \mathbf{H}_3) \dots \mathbf{H}_n) \\ &= [\mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^\top] \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_2\mathbf{u}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_3\mathbf{u}_3^\top \end{bmatrix} \cdots \begin{bmatrix} \mathbf{I}_{n-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_n\mathbf{u}_n^\top \end{bmatrix},\end{aligned}$$

where the parentheses indicate the order of the computation, the upper-left of  $\mathbf{H}_2$  is a  $1 \times 1$  identity matrix, and it will not change the **first column** of  $\mathbf{H}_1$ ; and the upper-left of  $\mathbf{H}_3$  is a  $2 \times 2$  identity matrix which will not change the **first 2 columns** of  $\mathbf{H}_1\mathbf{H}_2$ ; .... This property yields the step 14 in the algorithm.

**Algorithm 20** Full QR Decomposition via the Householder Reflector

---

**Require:** matrix  $\mathbf{A}$  with size  $m \times n$  and  $m \geq n$ ;

- 1: Initially set  $\mathbf{R} = \mathbf{A}$ ;
- 2: **for**  $i = 1$  to  $n$  **do**
- 3:    $\mathbf{a} = \mathbf{R}_{i:m,i}$ , i.e., first column of  $\mathbf{R}_{i:m,i:n} \in \mathbb{R}^{(m-i+1) \times (n-i+1)}$ ;
- 4:    $r = \|\mathbf{a}\|$ ;  $\triangleright 2(m - i + 1)$  flops;
- 5:    $\mathbf{u}_i = \mathbf{a} - r\mathbf{e}_1 \in \mathbb{R}^{m-i+1}$ ;  $\triangleright 1$  flop;
- 6:    $\mathbf{u}_i = \mathbf{u}_i / \|\mathbf{u}_i\|$ ;  $\triangleright 3(m - i + 1)$  flops;
- 7:    $\mathbf{R}_{i,i} = r$ ,  $\mathbf{R}_{i+1:m,i} = \mathbf{0}$ ;  $\triangleright 0$  flops, update first column of  $\mathbf{R}_{i:m,i:n}$
- 8:    $\mathbf{R}_{i:m,i+1:n} = \mathbf{R}_{i:m,i+1:n} - 2\mathbf{u}_i(\mathbf{u}_i^\top \mathbf{R}_{i:m,i+1:n})$ ;  $\triangleright$  update  $i + 1 : n$  columns of  $\mathbf{R}_{i:m,i:n}$
- 9: **end for**
- 10: Output  $\mathbf{R}$  as the triangular matrix;
- 11: Get  $\mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_n$ ;
- 12: Initially set  $\mathbf{Q} = \mathbf{H}_1$ ;
- 13: **for**  $i = 1$  to  $n - 1$  **do**
- 14:    $\mathbf{Q}_{1:m,i+1:m} = \mathbf{Q}_{1:m,i+1:m}(\mathbf{I} - 2\mathbf{u}_{i+1}\mathbf{u}_{i+1}^\top) = \mathbf{Q}_{1:m,i+1:m} - \mathbf{Q}_{1:m,i+1:m}2\mathbf{u}_{i+1}\mathbf{u}_{i+1}^\top$ ;
- 15: **end for**
- 16: Output  $\mathbf{Q}$  as the orthogonal matrix;

---

**Theorem 3.1: (Algorithm Complexity: QR via Householder)**

Algorithm 20 requires  $\sim 2mn^2 - \frac{2}{3}n^3$  flops to compute a full QR decomposition of an  $m \times n$  matrix with linearly independent columns and  $m \geq n$ . Further, if  $\mathbf{Q}$  is needed explicitly, additional  $\sim 4m^2n - 2mn^2$  flops are required.

**Proof** [of Theorem 3.1] For loop  $i$ ,  $\mathbf{A}_{i:m,i:n}$  is of size  $(m - i + 1) \times (n - i + 1)$ . Thus  $\mathbf{a}_1$  is in  $\mathbb{R}^{m-i+1}$ .

In step 4, to compute  $r = \|\mathbf{a}\|$  involves  $m - i + 1$  multiplications,  $m - i$  additions, and 1 square root operation which is  $\boxed{2(m - i + 1)}$  flops.

In step 5,  $\mathbf{u}_i = \mathbf{a} - r\mathbf{e}_1$  involves 1 subtraction which is  $\boxed{1}$  flop as the special structure of  $\mathbf{e}_1$ ;

In step 6, same as that in step 4, it requires  $2(m - i + 1)$  flops ( $m - i + 1$  multiplications,  $m - i$  additions, and 1 square root) to compute the norm  $\|\mathbf{u}_i\|$  and  $m - i + 1$  additional divisions which is  $\boxed{3(m - i + 1)}$  flops totally.

In step 8, suppose loop  $i = 1$ ,  $\mathbf{u}_1^\top \mathbf{R}_{1:m,2:n}$  requires  $n - 1$  times ( $m$  multiplications and  $m - 1$  additions) which is  $\boxed{(n - 1)(2m - 1)}$  flops.  $2\mathbf{u}_1$  requires  $\boxed{m}$  multiplications. Further,  $2\mathbf{u}_1(\mathbf{u}_1^\top \mathbf{R}_{1:m,2:n})$  requires  $\boxed{m(n - 1)}$  multiplications to make an  $m \times (n - 1)$  matrix. The final matrix subtraction needs  $\boxed{m(n - 1)}$  subtractions. Thus the total complexity for step 8 if loop  $i = 1$  is  $\boxed{4m(n - 1) + m - (n - 1)}$  flops. This analysis can be applied to any loop  $i$ , and the complexity of step 8 for loop  $i$  can be obtained by  $\boxed{4(m - i + 1)(n - i) + m - n + i}$  flops.

So for loop  $i$ , the total complexity from step 3 to step 8 can be defined as  $f(i)$  flops. To compute  $\mathbf{R}$ , the final complexity is

$$\text{cost} = f(1) + f(2) + \dots + f(n).$$

Simple calculations will show the sum of  $n$  loops is  $2mn^2 - \frac{2}{3}n^3$  flops if we keep only the leading terms.

To get the final orthogonal matrix  $\mathbf{Q}$ , since we have already computed  $2\mathbf{u}_{i+1}$  in step 8, this will not have additional costs. The computation of  $\mathbf{Q}_{1:m,i+1:m}2\mathbf{u}_{i+1}$  involves  $m$  times  $(m-i)$  multiplications and  $m-i-1$  additions which is  $m(2(m-i)-1)$  flops. Multiplied to  $\mathbf{u}_{i+1}^\top$  takes  $m(m-i)$  multiplications. Further, the final matrix subtraction requires  $m(m-i)$  subtractions. So in loop  $i$ , the complexity of step 14 is  $g(i) = 4m(m-i) - m = 4m^2 - 4mi - m$  flops. To compute  $\mathbf{Q}$ , the final complexity is

$$\text{cost} = g(1) + g(2) + \dots + g(n-1).$$

Simple calculation will show the sum of  $n-1$  loops is  $4m^2n - 2mn^2 - 4m^2 + mn + m$  flops, or  $4m^2n - 2mn^2$  flops if we keep only the leading terms. ■

After computing the full QR decomposition via the Householder algorithm, it is trivial to recover the reduced QR decomposition by just removing the silent columns in  $\mathbf{Q}$  and silent rows in  $\mathbf{R}$ . However, there is no direct way to compute the reduced QR decomposition without the full decomposition form.

In (Golub and Van Loan, 2013), a Householder method for rank-revealing QR decomposition is discussed, the complexity is  $4mnr - 2r^2(m+n) + 4r^3/3$  flops for rank  $r$  matrix  $\mathbf{A}$ . If  $r = n$ , the complexity agrees with the result in Theorem 3.1.

### 3.15. Existence of the QR Decomposition via the Givens Rotation

We have seen the Givens rotation can be utilized to find the rank-one update/downdate of the Cholesky decomposition in Section 2.13 (p. 73). Now let's take a look at what does the Givens rotation accomplish by specific examples. Consider the following  $2 \times 2$  orthogonal matrices

$$\mathbf{F} = \begin{bmatrix} -c & s \\ s & c \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad \mathbf{G} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where  $s = \sin \theta$  and  $c = \cos \theta$  for some  $\theta$ . The first matrix has  $\det(\mathbf{F}) = -1$  and is a special case of a Householder reflector in dimension 2 such that  $\mathbf{F} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$  where  $\mathbf{u} = \begin{bmatrix} \sqrt{\frac{1+c}{2}}, & \sqrt{\frac{1-c}{2}} \end{bmatrix}^\top$  or  $\mathbf{u} = \begin{bmatrix} -\sqrt{\frac{1+c}{2}}, & -\sqrt{\frac{1-c}{2}} \end{bmatrix}^\top$ . The latter two matrices have  $\det(\mathbf{J}) = \det(\mathbf{G}) = 1$  and effects a rotation instead of a reflection. Such a matrix is called a **Givens rotation**.

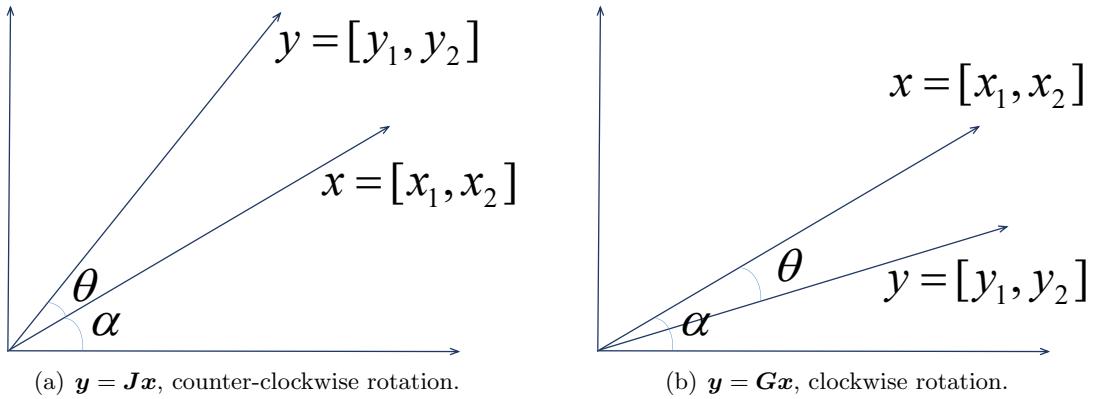
**Figure 3.9:** Demonstration of two Givens rotations.

Figure 3.9 demonstrate a rotation of  $\mathbf{x}$  under  $\mathbf{J}$ , where  $\mathbf{y} = \mathbf{J}\mathbf{x}$  such that

$$\begin{cases} y_1 = c \cdot x_1 - s \cdot x_2, \\ y_2 = s \cdot x_1 + c \cdot x_2. \end{cases}$$

We want to verify the angle between  $\mathbf{x}$  and  $\mathbf{y}$  is actually  $\theta$  (and counter-clockwise rotation) after the Givens rotation  $\mathbf{J}$  as shown in Figure 3.9(a). Firstly, we have

$$\begin{cases} \cos(\alpha) = \frac{x_1}{\sqrt{x_1^2 + x_2^2}}, \\ \sin(\alpha) = \frac{x_2}{\sqrt{x_1^2 + x_2^2}}. \end{cases} \quad \text{and} \quad \begin{cases} \cos(\theta) = c, \\ \sin(\theta) = s. \end{cases}$$

This implies  $\cos(\theta + \alpha) = \cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha)$ . If we can show  $\cos(\theta + \alpha) = \cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha)$  is equal to  $\frac{y_1}{\sqrt{y_1^2 + y_2^2}}$ , then we complete the proof.

For the former one,  $\cos(\theta + \alpha) = \cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha) = \frac{c \cdot x_1 - s \cdot x_2}{\sqrt{x_1^2 + x_2^2}}$ . For the latter one, it can be verified that  $\sqrt{y_1^2 + y_2^2} = \sqrt{x_1^2 + x_2^2}$ , and  $\frac{y_1}{\sqrt{y_1^2 + y_2^2}} = \frac{c \cdot x_1 - s \cdot x_2}{\sqrt{x_1^2 + x_2^2}}$ . This completes the proof. Similarly, we can also show that the angle between  $\mathbf{y} = \mathbf{G}\mathbf{x}$  and  $\mathbf{x}$  is also  $\theta$  in Figure 3.9(b) and the rotation is clockwise.

More generally, we define the  $n$ -th order Givens rotation as follows.

**Definition 3.1:  $n$ -th Order Givens Rotation**

An  $n \times n$  Givens rotation is represented by a matrix of the following form

$$G_{kl} = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & & c & & s \\ & & & & 1 & \\ & & & & & \ddots & \\ & & & & & & 1 \\ & & & -s & & c & \\ & & & & & & 1 \\ & & & & & & \\ & & & & & & \ddots \\ k & & & & & l & \end{bmatrix},$$

where the  $(k, k), (k, l), (l, k), (l, l)$  entries are  $c, s, -s, c$  respectively, and  $s = \cos \theta$  and  $c = \sin \theta$  for some  $\theta$ .

Let  $\delta_k \in \mathbb{R}^n$  be the zero vector where the  $k$ -th entry is 1. Then mathematically, the Givens rotation defined above can be denoted by

$$\mathbf{G}_{kl} = \mathbf{I} + (c - 1)(\boldsymbol{\delta}_k \boldsymbol{\delta}_k^\top + \boldsymbol{\delta}_l \boldsymbol{\delta}_l^\top) + s(\boldsymbol{\delta}_k \boldsymbol{\delta}_l^\top - \boldsymbol{\delta}_l \boldsymbol{\delta}_k^\top),$$

where the subscripts  $k, l$  indicate the rotation is in plane  $k$  and  $l$ .

Specifically, one can also define the  $n$ -th order Givens rotation where  $(k, k)$ ,  $(k, l)$ ,  $(l, k)$ ,  $(l, l)$  entries are  $c$ ,  $-s$ ,  $s$ ,  $c$  respectively (note the difference in the sign of  $s$ ). The ideas are the same.

It can be easily verified that the  $n$ -th order Givens rotation is an orthogonal matrix and its determinant is 1. For any vector  $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top \in \mathbb{R}^n$ , we have  $\mathbf{y} = \mathbf{G}_{kl}\mathbf{x}$ , where

$$\begin{cases} y_k = c \cdot x_k + s \cdot x_l, \\ y_l = -s \cdot x_k + c \cdot x_l, \\ y_j = x_j, \end{cases} \quad (j \neq k, l)$$

That is, a Givens rotation applied to  $\mathbf{x}$  rotates two components of  $\mathbf{x}$  by some angle  $\theta$  and leaves all other components the same. When  $\sqrt{x_k^2 + x_l^2} \neq 0$ , let  $c = \frac{x_k}{\sqrt{x_k^2 + x_l^2}}$ ,  $s = \frac{x_l}{\sqrt{x_k^2 + x_l^2}}$ . Then,

$$\begin{cases} y_k = \sqrt{x_k^2 + x_l^2}, \\ y_l = 0, \\ y_j = x_j. \end{cases} \quad (j \neq k, l)$$

This finding above is essential for the QR decomposition via the Givens rotation.

**Corollary 3.2: (Basis From Givens Rotations Forwards)**

For any vector  $\mathbf{x} \in \mathbb{R}^n$ , there exists a set of Givens rotations  $\{\mathbf{G}_{12}, \mathbf{G}_{13}, \dots, \mathbf{G}_{1n}\}$  such that  $\mathbf{G}_{1n} \dots \mathbf{G}_{13} \mathbf{G}_{12} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$  where  $\mathbf{e}_1 \in \mathbb{R}^n$  is the first unit basis in  $\mathbb{R}^n$ .

**Proof** [of Corollary 3.2] From the finding above, we can find a  $\mathbf{G}_{12}, \mathbf{G}_{13}, \mathbf{G}_{14}$  such that

$$\mathbf{G}_{12} \mathbf{x} = \left[ \sqrt{x_1^2 + x_2^2}, 0, x_3, \dots, x_n \right]^\top,$$

$$\mathbf{G}_{13} \mathbf{G}_{12} \mathbf{x} = \left[ \sqrt{x_1^2 + x_2^2 + x_3^2}, 0, 0, x_4, \dots, x_n \right]^\top,$$

and

$$\mathbf{G}_{14} \mathbf{G}_{13} \mathbf{G}_{12} \mathbf{x} = \left[ \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2}, 0, 0, 0, x_5, \dots, x_n \right]^\top.$$

Continue this process, we will obtain  $\mathbf{G}_{1n} \dots \mathbf{G}_{13} \mathbf{G}_{12} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$ . ■

**Remark 3.3: Basis From Givens Rotations Backwards**

In Corollary 3.2, we find the Givens rotation that introduces zeros from the 2-nd entry to the  $n$ -th entry (i.e., forward). Sometimes we want the reverse order, i.e., introduce zeros from the  $n$ -th entry to the 2-nd entry such that  $\mathbf{G}_{12} \mathbf{G}_{13} \dots \mathbf{G}_{1n} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$  where  $\mathbf{e}_1 \in \mathbb{R}^n$  is the first unit basis in  $\mathbb{R}^n$ .

The procedure is similar, we can find a  $\mathbf{G}_{1n}, \mathbf{G}_{1,(n-1)}, \mathbf{G}_{1,(n-2)}$  such that

$$\mathbf{G}_{1n} \mathbf{x} = \left[ \sqrt{x_1^2 + x_n^2}, x_2, x_3, \dots, x_{n-1}, 0 \right]^\top,$$

$$\mathbf{G}_{1,(n-1)} \mathbf{G}_{1n} \mathbf{x} = \left[ \sqrt{x_1^2 + x_{n-1}^2 + x_n^2}, x_2, x_3, \dots, x_{n-2}, 0, 0 \right]^\top,$$

and

$$\mathbf{G}_{1,(n-2)} \mathbf{G}_{1,(n-1)} \mathbf{G}_{1n} \mathbf{x} = \left[ \sqrt{x_1^2 + x_{n-2}^2 + x_{n-1}^2 + x_n^2}, x_2, x_3, \dots, x_{n-3}, 0, 0, 0 \right]^\top.$$

Continue this process, we will obtain  $\mathbf{G}_{12} \mathbf{G}_{13} \dots \mathbf{G}_{1n} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$ .

**An alternative form** Alternatively, there are rotations  $\{\mathbf{G}_{12}, \mathbf{G}_{23}, \dots, \mathbf{G}_{(n-1),n}\}$  such that  $\mathbf{G}_{12} \mathbf{G}_{23} \dots \mathbf{G}_{(n-1),n} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$  where  $\mathbf{e}_1 \in \mathbb{R}^n$  is the first unit basis in  $\mathbb{R}^n$  and where

$$\mathbf{G}_{(n-1),n} \mathbf{x} = \left[ x_1, x_2, \dots, x_{n-2}, \sqrt{x_{n-1}^2 + x_n^2}, 0 \right]^\top,$$

$$\mathbf{G}_{(n-2),(n-1)} \mathbf{G}_{(n-1),n} \mathbf{x} = \left[ x_1, x_2, \dots, x_{n-3}, \sqrt{x_{n-2}^2 + x_{n-1}^2 + x_n^2}, 0, 0 \right]^\top,$$

and

$$\begin{aligned} \mathbf{G}_{(n-3),(n-2)} \mathbf{G}_{(n-2),(n-1)} \mathbf{G}_{(n-1),n} \mathbf{x} = \\ \left[ x_1, x_2, \dots, x_{n-4}, \sqrt{x_{n-3}^2 + x_{n-2}^2 + x_{n-1}^2 + x_n^2}, 0, 0, 0 \right]^\top. \end{aligned}$$

Continue this process, we will obtain  $\mathbf{G}_{12} \mathbf{G}_{23} \dots \mathbf{G}_{(n-1),n} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$ .

The above backward Givens rotation basis update will be proved useful in the rank-one changes of the QR decomposition (Section 3.20.2, p. 127).

From the Corollary 3.2 above, for the way to introduce zeros, we could **rotate** the columns of the matrix to a basis vector  $\mathbf{e}_1$  whose entries are all zero except the first entry. Let  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$  be the column partition of  $\mathbf{A}$ , and let

$$\mathbf{G}_1 = \mathbf{G}_{1m} \dots \mathbf{G}_{13} \mathbf{G}_{12}, \quad (3.14)$$

where  $\mathbf{e}_1$  here is the first basis for  $\mathbb{R}^m$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^m$ . Then

$$\begin{aligned} \mathbf{G}_1 \mathbf{A} &= [\mathbf{G}_1 \mathbf{a}_1, \mathbf{G}_1 \mathbf{a}_2, \dots, \mathbf{G}_1 \mathbf{a}_n] \\ &= \begin{bmatrix} \|\mathbf{a}_1\| & \mathbf{R}_{1,2:n} \\ \mathbf{0} & \mathbf{B}_2 \end{bmatrix}, \end{aligned} \quad (3.15)$$

which rotates  $\mathbf{a}_1$  to  $\|\mathbf{a}_1\| \mathbf{e}_1$  and introduces zeros below the diagonal in the 1-st column of  $\mathbf{A}$ . And bear in mind that the  $\mathbf{G}_1$  above will sequentially change the (1-st, 2-nd), (1-st, 3-rd), (1-st, 4-th), ..., (1-st,  $m$ -th) elements in pair for any vector  $\mathbf{v} \in \mathbb{R}^m$ .

We can then apply this process to  $\mathbf{B}_2$  in Equation (3.15) to make the entries below the (2,2)-th entry to be all zeros. Suppose  $\mathbf{B}_2 = [\mathbf{b}_2, \mathbf{b}_3, \dots, \mathbf{b}_n]$  is the column partition of  $\mathbf{B}_2$ , and let

$$\mathbf{G}_2 = \mathbf{G}_{2m} \dots \mathbf{G}_{24} \mathbf{G}_{23},$$

where  $\mathbf{G}_{2n}, \dots, \mathbf{G}_{24}, \mathbf{G}_{23}$  can be implied from context. Then

$$\mathbf{G}_2 \begin{bmatrix} \mathbf{R}_{1,2:n} \\ \mathbf{B}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{12} & \mathbf{R}_{1,3:n} \\ \|\mathbf{b}_2\| & \mathbf{R}_{2,3:n} \\ \mathbf{0} & \mathbf{C}_3 \end{bmatrix},$$

and

$$\mathbf{G}_2 \mathbf{G}_1 \mathbf{A} = [\mathbf{G}_2 \mathbf{G}_1 \mathbf{a}_1, \mathbf{G}_2 \mathbf{G}_1 \mathbf{a}_2, \dots, \mathbf{G}_2 \mathbf{G}_1 \mathbf{a}_n] = \begin{bmatrix} \|\mathbf{a}_1\| & \mathbf{R}_{12} & \mathbf{R}_{1,3:n} \\ 0 & \|\mathbf{b}_2\| & \mathbf{R}_{2,3:n} \\ \mathbf{0} & \mathbf{0} & \mathbf{C}_3 \end{bmatrix}.$$

Same process can go on, and we will finally triangularize  $\mathbf{A} = (\mathbf{G}_n \mathbf{G}_{n-1} \dots \mathbf{G}_1)^{-1} \mathbf{R} = \mathbf{Q} \mathbf{R}$ . And since  $\mathbf{G}_i$ 's are orthogonal, the orthogonal  $\mathbf{Q}$  can also be obtained by  $\mathbf{Q} = (\mathbf{G}_n \mathbf{G}_{n-1} \dots \mathbf{G}_1)^{-1} = \mathbf{G}_1^\top \mathbf{G}_2^\top \dots \mathbf{G}_n^\top$ , and

$$\begin{aligned} \mathbf{G}_1^\top \mathbf{G}_2^\top \dots \mathbf{G}_n^\top &= (\mathbf{G}_n \dots \mathbf{G}_2 \mathbf{G}_1)^\top \\ &= \{(\mathbf{G}_{nm} \dots \mathbf{G}_{n,(n+1)}) \dots (\mathbf{G}_{2m} \dots \mathbf{G}_{23}) (\mathbf{G}_{1m} \dots \mathbf{G}_{12})\}^\top. \end{aligned} \quad (3.16)$$

**When will the Givens work better?** In practice, compared to the Householder algorithms, the Givens rotation algorithm works better when  $\mathbf{A}$  already has a lot of zeros below the main diagonal. Therefore, the Givens rotations can be applied to the rank-one changes of the QR decomposition as rank-one change will only introduce a small amount of nonzero values (Section 3.20.2, p. 127).

An example of a  $5 \times 4$  matrix is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

**Givens rotations in  $\mathbf{G}_1$**  For a  $5 \times 4$  example, we realize that  $\mathbf{G}_1 = \mathbf{G}_{15}\mathbf{G}_{14}\mathbf{G}_{13}\mathbf{G}_{12}$ . And the process is shown as follows:

$$\begin{array}{c}
 \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_{12}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_{13}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \\
 \mathbf{A} \qquad \qquad \qquad \mathbf{G}_{12}\mathbf{A} \qquad \qquad \qquad \mathbf{G}_{13}\mathbf{G}_{12}\mathbf{A}
 \end{array}$$
  

$$\begin{array}{c}
 \xrightarrow{\mathbf{G}_{14}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_{15}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \\
 \mathbf{G}_{14}\mathbf{G}_{13}\mathbf{G}_{12}\mathbf{A} \qquad \qquad \qquad \mathbf{G}_{15}\mathbf{G}_{14}\mathbf{G}_{13}\mathbf{G}_{12}\mathbf{A}
 \end{array}$$

**Givens rotation as a big picture** Take  $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4$  as a single matrix, we have

$$\begin{array}{c}
 \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_1} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_2} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \end{bmatrix} \\
 \mathbf{A} \qquad \qquad \qquad \mathbf{G}_1\mathbf{A} \qquad \qquad \qquad \mathbf{G}_2\mathbf{G}_1\mathbf{A}
 \end{array}$$
  

$$\begin{array}{c}
 \xrightarrow{\mathbf{G}_3} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{\boxtimes} & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_4} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{\boxtimes} & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 \end{bmatrix} \\
 \mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{A} \qquad \qquad \qquad \mathbf{G}_4\mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{A}
 \end{array}$$

**Orders to introduce the zeros** With the Givens rotations for the QR decomposition, it is flexible to choose different orders to introduce the zeros of  $\mathbf{R}$ . In our case, we introduce zeros column by column. It is also possible to introduce zeros row by row.

### 3.16. Computing the Full QR Decomposition via the Givens Rotation

The algorithm to compute the full QR decomposition via the Givens rotation is straightforward from the example shown above and is illustrated in Algorithm 21.

**Algorithm 21** Full QR Decomposition via the Givens Rotation

**Require:** matrix  $\mathbf{A}$  with size  $m \times n$  and  $m \geq n$ ;

```

1: Initially set $\mathbf{R} = \mathbf{A}$, $\mathbf{Q} = \mathbf{I}$;
2: for $i = 1$ to n do
3: for $j = i + 1$ to m do
4: Get Givens rotation $\mathbf{G}_{i,j}$ with the following parameters c, s :
5: $c = \frac{x_k}{\sqrt{x_k^2+x_l^2}}$, $s = \frac{x_l}{\sqrt{x_k^2+x_l^2}}$ where $x_k = \mathbf{R}_{i,i}$, $x_l = \mathbf{R}_{j,i}$;
6: Calculate $\mathbf{R} = \mathbf{G}_{i,j}\mathbf{R}$ in following two steps:
7: i -th row: $\mathbf{R}_{i,:} = c \cdot \mathbf{R}_{i,:} + s\mathbf{R}_{j,:}$;
8: j -th row: $\mathbf{R}_{j,:} = -s \cdot \mathbf{R}_{i,:} + c\mathbf{R}_{j,:}$;
9: end for
10: end for
11: Output \mathbf{R} as the triangular matrix;
12: for $i = 1$ to n do
13: for $j = i + 1$ to m do
14: Calculate $\mathbf{Q} = \mathbf{G}_{i,j}\mathbf{Q}$ in following two steps:
15: i -th row: $\mathbf{Q}_{i,:} = c \cdot \mathbf{Q}_{i,:} + s\mathbf{Q}_{j,:}$;
16: j -th row: $\mathbf{Q}_{j,:} = -s \cdot \mathbf{Q}_{i,:} + c\mathbf{Q}_{j,:}$;
17: end for
18: end for
19: Output $\mathbf{Q} = \mathbf{Q}^\top$ from Equation (3.16);

```

**Theorem 3.1: (Algorithm Complexity: QR via Givens)**

Algorithm 21 requires  $\sim 3mn^2 - n^3$  flops to compute a full QR decomposition of an  $m \times n$  matrix with linearly independent columns and  $m \geq n$ . Further, if  $\mathbf{Q}$  is needed explicitly, additional  $\sim 3mn^2 - n^3$  flops required.

**Proof** [of Theorem 3.1] For step 5, each iteration  $i, j$  requires 6 flops (which are 2 square operations, 1 addition, 1 square root, and 2 divisions). And there are  $m - i$  iterations for each  $i$  which means  $(m - 1) + (m - 2) + \dots + (m - n) = (mn - \frac{n^2+n}{2})$  iterations. Therefore, the complexity for all the step 5's is  $\boxed{6(mn - \frac{n^2+n}{2})}$  flops.

For each iteration  $i$ , step 7 and step 8 operate on two length- $(n - i + 1)$  vectors. The two steps take  $6(n - i + 1)$  flops for each iteration  $i$  (which are  $4(n - i + 1)$  multiplications and  $2(n - i + 1)$  additions). And for each  $i$ , there are  $(m - i)$  such operations which takes  $(m - i) \times 6(n - i + 1)$  flops for each  $i$ . Let  $f(i) = (m - i) \times 6(n - i + 1) = m(n + 1) - (m + n + 1)i + i^2$ . The total complexity for the two steps is equal to

$$\text{cost} = f(1) + f(2) + \dots + f(n),$$

or  $\boxed{3mn^2 - n^3}$  flops if keep only the leading terms.

Similarly, we can get the complexity of step 15 and 16 with  $\boxed{3mn^2 - n^3}$  flops if keep only the leading terms. ■

Same as the Householder algorithm, after computing the full QR decomposition via the Givens algorithm, it is trivial to recover the reduced QR decomposition by just removing the silent columns in  $\mathbf{Q}$  and silent rows in  $\mathbf{R}$ .

### 3.17. Uniqueness of the QR Decomposition

The results of the QR decomposition from the Gram-Schmidt process , the Householder algorithm, and the Givens algorithms are different. Even in the Householder algorithm, we have different methods to choose the sign of  $r_1$  in Equation (3.12). Thus, from this point, QR decomposition is not unique.

**Example 3.3 (Non-Uniqueness of the QR Decomposition)** Suppose the matrix  $\mathbf{A}$  is given by

$$\mathbf{A} = \begin{bmatrix} 4 & 1 \\ 3 & 2 \end{bmatrix}.$$

The QR decomposition of  $\mathbf{A}$  can be obtained by

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}_1 \mathbf{R}_1 = \begin{bmatrix} 0.8 & 0.6 \\ 0.6 & -0.8 \end{bmatrix} \begin{bmatrix} 5 & 2 \\ 0 & -1 \end{bmatrix} \\ &= \mathbf{Q}_2 \mathbf{R}_2 = \begin{bmatrix} 0.8 & -0.6 \\ 0.6 & 0.8 \end{bmatrix} \begin{bmatrix} 5 & 2 \\ 0 & 1 \end{bmatrix} \\ &= \mathbf{Q}_3 \mathbf{R}_3 = \begin{bmatrix} -0.8 & -0.6 \\ -0.6 & 0.8 \end{bmatrix} \begin{bmatrix} -5 & -2 \\ 0 & 1 \end{bmatrix} \\ &= \mathbf{Q}_4 \mathbf{R}_4 = \begin{bmatrix} -0.8 & 0.6 \\ -0.6 & -0.8 \end{bmatrix} \begin{bmatrix} -5 & -2 \\ 0 & -1 \end{bmatrix}. \end{aligned}$$

Thus the QR decomposition of  $\mathbf{A}$  is not unique. □

However, if we use just the procedure described in the Gram-Schmidt process, or systematically choose the sign in the Householder algorithm, then the decomposition is unique. The uniqueness of the *reduced* QR decomposition for full column rank matrix  $\mathbf{A}$  is assured when  $\mathbf{R}$  has positive diagonals as shown in the “main proof” of Section 3.4 by inductive analysis. We here provide another proof for the uniqueness of the *reduced* QR decomposition for matrices if the diagonal values of  $\mathbf{R}$  are positive which will shed light on the implicit Q theorem in Hessenberg decomposition (Section 8.5, p. 205) or tridiagonal decomposition (Theorem 9.3, p. 212).

#### Corollary 3.1: (Uniqueness of the reduced QR Decomposition)

Suppose matrix  $\mathbf{A}$  is an  $m \times n$  matrix with full column rank  $n$  and  $m \geq n$ . Then, the *reduced* QR decomposition is unique if the main diagonal values of  $\mathbf{R}$  are positive.

**Proof** [of Corollary 3.1] Suppose the *reduced* QR decomposition is not unique, we can complete it into a *full* QR decomposition, then we can find two such full decompositions so

that  $\mathbf{A} = \mathbf{Q}_1 \mathbf{R}_1 = \mathbf{Q}_2 \mathbf{R}_2$  which implies  $\mathbf{R}_1 = \mathbf{Q}_1^{-1} \mathbf{Q}_2 \mathbf{R}_2 = \mathbf{V} \mathbf{R}_2$  where  $\mathbf{V} = \mathbf{Q}_1^{-1} \mathbf{Q}_2$  is an orthogonal matrix. Write out the equation, we have

$$\mathbf{R}_1 = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & \dots & r_{2n} \\ 0 & \ddots & \vdots & r_{nn} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \dots & v_{mm} \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{22} & \dots & s_{2n} \\ 0 & \ddots & \vdots & s_{nn} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \end{bmatrix} = \mathbf{V} \mathbf{R}_2,$$

This implies

$$r_{11} = v_{11}s_{11}, \quad v_{21} = v_{31} = v_{41} = \dots = v_{m1} = 0.$$

Since  $\mathbf{V}$  contains mutually orthonormal columns and the first column of  $\mathbf{V}$  is of norm 1. Thus,  $v_{11} = \pm 1$ . We notice that  $r_{ii} > 0$  and  $s_{ii} > 0$  for  $i \in \{1, 2, \dots, n\}$  by assumption such that  $r_{11} > 0$  and  $s_{11} > 0$  and  $v_{11}$  can only be positive 1. Since  $\mathbf{V}$  is an orthogonal matrix, we also have

$$v_{12} = v_{13} = v_{14} = \dots = v_{1m} = 0.$$

Applying this process to the submatrices of  $\mathbf{R}_1, \mathbf{V}, \mathbf{R}_2$ , we will find the upper-left submatrix of  $\mathbf{V}$  is an identity:  $\mathbf{V}[1 : n, 1 : n] = \mathbf{I}_n$  such that  $\mathbf{R}_1 = \mathbf{R}_2$ . This implies  $\mathbf{Q}_1[:, 1 : n] = \mathbf{Q}_2[:, 1 : n]$  and leads to a contradiction such that the reduced QR decomposition is unique. ■

We notice that the uniqueness of the reduced QR decomposition shown above is from the diagonals of  $\mathbf{R}$  are positive. Therefore, if we restrict the QR decomposition that the diagonal values of  $\mathbf{R}$  are positive in Householder or Givens algorithms, then the decomposition will be unique. And actually, the Gram-Schmidt process in Algorithm 12 is such a decomposition that the diagonal values of  $\mathbf{R}$  will be positive if  $\mathbf{A}$  has full column rank. If  $\mathbf{A}$  has dependent columns, the diagonal entries of  $\mathbf{R}$  can only be nonnegative, and the factorization may not be unique.

### 3.18. LQ Decomposition

We previously proved the existence of the QR decomposition via the Gram-Schmidt process in which case we are interested in the column space of a matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ . The successive spaces spanned by the columns  $\mathbf{a}_1, \mathbf{a}_2, \dots$  of  $\mathbf{A}$  are

$$\mathcal{C}([\mathbf{a}_1]) \subseteq \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2]) \subseteq \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]) \subseteq \dots,$$

The idea of QR decomposition is the construction of a sequence of orthonormal vectors  $\mathbf{q}_1, \mathbf{q}_2, \dots$  that span the same successive subspaces:

$$\{\mathcal{C}([\mathbf{q}_1]) = \mathcal{C}([\mathbf{a}_1])\} \subseteq \{\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2])\} \subseteq \dots,$$

However, in many applications (see (Schilders, 2009)), we are also interested in the row space of a matrix  $\mathbf{B} = [\mathbf{b}_1^\top; \mathbf{b}_2^\top; \dots; \mathbf{b}_m^\top] \in \mathbb{R}^{m \times n}$ , where  $\mathbf{b}_i$  is the  $i$ -th row of  $\mathbf{B}$ . The successive spaces spanned by the rows  $\mathbf{b}_1, \mathbf{b}_2, \dots$  of  $\mathbf{B}$  are

$$\mathcal{C}([\mathbf{b}_1]) \subseteq \mathcal{C}([\mathbf{b}_1, \mathbf{b}_2]) \subseteq \mathcal{C}([\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]) \subseteq \dots.$$

The QR decomposition thus has its sibling which finds the orthogonal row space. By applying QR decomposition on  $\mathbf{B}^\top = \mathbf{Q}_0 \mathbf{R}$ , we recover the LQ decomposition of the matrix  $\mathbf{B} = \mathbf{L} \mathbf{Q}$  where  $\mathbf{Q} = \mathbf{Q}_0^\top$  and  $\mathbf{L} = \mathbf{R}^\top$ .

**Theorem 3.1: (LQ Decomposition)**

Every  $m \times n$  matrix  $\mathbf{B}$  (whether linearly independent or dependent rows) with  $n \geq m$  can be factored as

$$\mathbf{B} = \mathbf{L}\mathbf{Q},$$

where

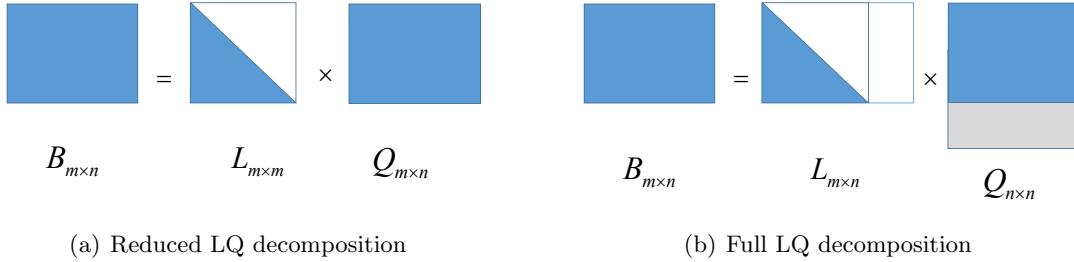
1. **Reduced:**  $\mathbf{L}$  is an  $m \times m$  lower triangular matrix and  $\mathbf{Q}$  is  $m \times n$  with orthonormal rows which is known as the **reduced LQ decomposition**;

2. **Full:**  $\mathbf{L}$  is an  $m \times n$  lower triangular matrix and  $\mathbf{Q}$  is  $n \times n$  with orthonormal rows which is known as the **full LQ decomposition**. If further restrict the lower triangular matrix to be a square matrix, the full LQ decomposition can be denoted as

$$\mathbf{B} = [\mathbf{L}_0 \quad \mathbf{0}] \mathbf{Q},$$

where  $\mathbf{L}_0$  is an  $m \times m$  square lower triangular matrix.

Similarly, a comparison between the reduced and full LQ decomposition is shown in Figure 3.10.



**Figure 3.10:** Comparison between the reduced and full LQ decomposition.

**Row-pivoted LQ (RPLQ)** Similar to the column-pivoted QR in Section 3.10, there exists a row-pivoted LQ decomposition:

$$\left\{ \begin{array}{ll} \text{Reduced RPLQ:} & \mathbf{PB} = \underbrace{\begin{bmatrix} \mathbf{L}_{11} \\ \mathbf{L}_{21} \end{bmatrix}}_{m \times r} \underbrace{\mathbf{Q}_r}_{r \times n}; \\ \text{Full RPLQ:} & \mathbf{PB} = \underbrace{\begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{0} \end{bmatrix}}_{m \times m} \underbrace{\mathbf{Q}}_{n \times n}, \end{array} \right.$$

where  $\mathbf{L}_{11} \in \mathbb{R}^{r \times r}$  is lower triangular,  $\mathbf{Q}_r$  or  $\mathbf{Q}_{1:r,:}$  spans the same row space as  $\mathbf{B}$ , and  $\mathbf{P}$  is a permutation matrix that interchange independent rows into the upper-most rows.

### 3.19. Two-Sided Orthogonal Decomposition

**Theorem 3.1: (Two-Sided Orthogonal Decomposition)**

When square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with rank  $r$ , the full CPQR, RPLQ of  $\mathbf{A}$  are given by

$$\mathbf{AP}_1 = \mathbf{Q}_1 \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{P}_2 \mathbf{A} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{0} \end{bmatrix} \mathbf{Q}_2$$

respectively. Then we would find out

$$\mathbf{APA} = \mathbf{Q}_1 \underbrace{\begin{bmatrix} \mathbf{R}_{11}\mathbf{L}_{11} + \mathbf{R}_{12}\mathbf{L}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\text{rank } r} \mathbf{Q}_2,$$

where the first  $r$  columns of  $\mathbf{Q}_1$  span the same column space of  $\mathbf{A}$ , first  $r$  rows of  $\mathbf{Q}_2$  span the same row space of  $\mathbf{A}$ , and  $\mathbf{P}$  is a permutation matrix. We name this decomposition as **two-sided orthogonal decomposition**.

This decomposition is very similar to the property of SVD:  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$  that the first  $r$  columns of  $\mathbf{U}$  span the column space of  $\mathbf{A}$  and the first  $r$  columns of  $\mathbf{V}$  span the row space of  $\mathbf{A}$  (we shall see in Lemma 14.1, p. 269). Therefore, the two-sided orthogonal decomposition can be regarded as an inexpensive alternative in this sense.

**Lemma 3.2: (Four Orthonormal Basis)**

Given the two-sided orthogonal decomposition of matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with rank  $r$ :  $\mathbf{APA} = \mathbf{UFV}^\top$ , where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$  and  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$  are the column partitions of  $\mathbf{U}$  and  $\mathbf{V}$ . Then, we have the following property:

- $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A}^\top)$ ;
- $\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}$  is an orthonormal basis of  $\mathcal{N}(\mathbf{A})$ ;
- $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A})$ ;
- $\{\mathbf{u}_{r+1}, \mathbf{u}_{r+2}, \dots, \mathbf{u}_n\}$  is an orthonormal basis of  $\mathcal{N}(\mathbf{A}^\top)$ .

## 3.20. Applications

### 3.20.1 Application: Least Squares via the Full QR Decomposition

Let's consider the overdetermined system  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the data matrix,  $\mathbf{b} \in \mathbb{R}^m$  with  $m > n$  is the observation matrix. Normally  $\mathbf{A}$  will have full column rank since the data from real work has a large chance to be unrelated. And the least squares (LS) solution is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  for minimizing  $\|\mathbf{Ax} - \mathbf{b}\|^2$ , where  $\mathbf{A}^\top \mathbf{A}$  is invertible since  $\mathbf{A}$  has full column rank and  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$ .

However, the inverse of a matrix is not easy to compute, we can then use QR decomposition to find the least squares solution as illustrated in the following theorem.

**Theorem 3.1: (LS via QR for Full Column Rank Matrix)**

Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{A} = \mathbf{QR}$  is its full QR decomposition with  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  being an orthogonal matrix,  $\mathbf{R} \in \mathbb{R}^{m \times n}$  being an upper triangular matrix appended by additional  $m - n$  zero rows, and  $\mathbf{A}$  has full column rank with  $m \geq n$ . Suppose  $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}$ , where  $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$  is the square upper triangular in  $\mathbf{R}$ ,  $\mathbf{b} \in \mathbb{R}^m$ , then the LS solution to  $\mathbf{Ax} = \mathbf{b}$  is given by

$$\mathbf{x}_{LS} = \mathbf{R}_1^{-1} \mathbf{c},$$

where  $\mathbf{c}$  is the first  $n$  components of  $\mathbf{Q}^\top \mathbf{b}$ .

**Proof** [of Theorem 3.1] Since  $\mathbf{A} = \mathbf{QR}$  is the full QR decomposition of  $\mathbf{A}$  and  $m \geq n$ , the last  $m - n$  rows of  $\mathbf{R}$  are zero as shown in Figure 3.6. Then  $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$  is the square upper triangular in  $\mathbf{R}$  and  $\mathbf{Q}^\top \mathbf{A} = \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}$ . Thus,

$$\begin{aligned} \|\mathbf{Ax} - \mathbf{b}\|^2 &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) \\ &= (\mathbf{Ax} - \mathbf{b})^\top \mathbf{Q} \mathbf{Q}^\top (\mathbf{Ax} - \mathbf{b}) \quad (\text{Since } \mathbf{Q} \text{ is an orthogonal matrix}) \\ &= \|\mathbf{Q}^\top \mathbf{Ax} - \mathbf{Q}^\top \mathbf{b}\|^2 \quad (\text{Invariant under orthogonal}) \\ &= \left\| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{x} - \mathbf{Q}^\top \mathbf{b} \right\|^2 \\ &= \|\mathbf{R}_1 \mathbf{x} - \mathbf{c}\|^2 + \|\mathbf{d}\|^2, \end{aligned}$$

where  $\mathbf{c}$  is the first  $n$  components of  $\mathbf{Q}^\top \mathbf{b}$  and  $\mathbf{d}$  is the last  $m - n$  components of  $\mathbf{Q}^\top \mathbf{b}$ . And the LS solution can be calculated by back substitution of the upper triangular system  $\mathbf{R}_1 \mathbf{x} = \mathbf{c}$ , i.e.,  $\mathbf{x}_{LS} = \mathbf{R}_1^{-1} \mathbf{c}$ .  $\blacksquare$

To verify Theorem 3.1, for the full QR decomposition of  $\mathbf{A} = \mathbf{QR}$  where  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  and  $\mathbf{R} \in \mathbb{R}^{m \times n}$ . Together with the LS solution  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ , we obtain

$$\begin{aligned} \mathbf{x}_{LS} &= (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} \\ &= (\mathbf{R}^\top \mathbf{Q}^\top \mathbf{QR})^{-1} \mathbf{R}^\top \mathbf{Q}^\top \mathbf{b} \\ &= (\mathbf{R}^\top \mathbf{R})^{-1} \mathbf{R}^\top \mathbf{Q}^\top \mathbf{b} \\ &= (\mathbf{R}_1^\top \mathbf{R}_1)^{-1} \mathbf{R}_1^\top \mathbf{Q}^\top \mathbf{b} \quad (3.17) \\ &= \mathbf{R}_1^{-1} \mathbf{R}_1^{-\top} \mathbf{R}_1^\top \mathbf{Q}^\top \mathbf{b} \\ &= \mathbf{R}_1^{-1} \mathbf{R}_1^{-\top} \mathbf{R}_1^\top \mathbf{Q}_1^\top \mathbf{b} \\ &= \mathbf{R}_1^{-1} \mathbf{Q}_1^\top \mathbf{b}, \end{aligned}$$

where  $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}$  and  $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$  is an upper triangular matrix, and  $\mathbf{Q}_1 = \mathbf{Q}_{1:m, 1:n} \in \mathbb{R}^{m \times n}$  is the first  $n$  columns of  $\mathbf{Q}$  (i.e.,  $\mathbf{Q}_1 \mathbf{R}_1$  is the reduced QR decomposition of  $\mathbf{A}$ ). Then the result of Equation (3.17) agrees with Theorem 3.1.

To conclude, using the QR decomposition, we first derive directly the least squares result which results in the argument in Theorem 3.1. Moreover, we verify the result of LS from calculus indirectly by QR decomposition as well. The two results coincide with each other. For those who are interested in LS in linear algebra, a pictorial view of least squares for full column rank  $\mathbf{A}$  in the fundamental theorem of linear algebra is provided in Appendix C or a detailed discussion in (Lu, 2021c).

### 3.20.2 Application: Rank-One Changes

We previously discussed the rank-one update/downdate of the Cholesky decomposition in Section 2.13 (p. 73). The rank-one change  $\mathbf{A}'$  of matrix  $\mathbf{A}$  in the QR decomposition is defined in a similar form:

$$\begin{aligned}\mathbf{A}' &= \mathbf{A} + \mathbf{u}\mathbf{v}^\top, \\ &\quad \downarrow \quad \downarrow \\ \mathbf{Q}'\mathbf{R}' &= \mathbf{Q}\mathbf{R} + \mathbf{u}\mathbf{v}^\top,\end{aligned}$$

where if we set  $\mathbf{A}' = \mathbf{A} - (-\mathbf{u})\mathbf{v}^\top$ , we recover the downdate form such that the update or downdate in the QR decomposition are the same. Let  $\mathbf{w} = \mathbf{Q}^\top \mathbf{u}$ , we have

$$\mathbf{A}' = \mathbf{Q}(\mathbf{R} + \mathbf{w}\mathbf{v}^\top).$$

From the second form in Remark 3.3 (p. 118) on introducing zeros backwards, there exists a set of Givens rotations  $\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n}$  such that

$$\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n}\mathbf{w} = \pm\|\mathbf{w}\|\mathbf{e}_1,$$

where  $\mathbf{G}_{(k-1),k}$  is the Givens rotation in plane  $k-1$  and  $k$  that introduces zero in the  $k$ -th entry of  $\mathbf{w}$ . Apply this rotation to  $\mathbf{R}$ , we have

$$\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n}\mathbf{R} = \mathbf{H}_0,$$

where the Givens rotations in this *reverse order* are useful to transform the upper triangular  $\mathbf{R}$  into a “simple” upper Hessenberg which is close to upper triangular matrices (see Definition 8.1, p. 197 that we will introduce in the Hessenberg decomposition). If the rotations are transforming  $\mathbf{w}$  into  $\pm\|\mathbf{w}\|\mathbf{e}_1$  from *forward order* as in Corollary 3.2 (p. 118), we will not have this upper Hessenberg  $\mathbf{H}_0$ . To see this, suppose  $\mathbf{R} \in \mathbb{R}^{5 \times 5}$ , an example is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed. The backwards rotations result in the upper Hessenberg

$\mathbf{H}_0$  which is relatively simple to handle:

$$\begin{array}{c}
 \text{Backwards} \\
 \text{(Right Way)} \\
 : 
 \end{array}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \\ 0 & 0 & 0 & 0 & \square \end{array} \right] \\
 \mathbf{R}
 \end{array}
 \xrightarrow{\mathbf{G}_{45}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \\ 0 & 0 & 0 & \square & \square \end{array} \right] \\
 \mathbf{G}_{45}\mathbf{R}
 \end{array}
 \xrightarrow{\mathbf{G}_{34}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \end{array} \right] \\
 \mathbf{G}_{34}\mathbf{G}_{45}\mathbf{R}
 \end{array}
 \\
 \xrightarrow{\mathbf{G}_{23}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \end{array} \right] \\
 \mathbf{G}_{23}\mathbf{G}_{34}\mathbf{G}_{45}\mathbf{R}
 \end{array}
 \xrightarrow{\mathbf{G}_{12}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \end{array} \right] \\
 \mathbf{G}_{12}\mathbf{G}_{23}\mathbf{G}_{34}\mathbf{G}_{45}\mathbf{R}
 \end{array}.
 \end{array}$$

And the forward rotations result in a full matrix:

$$\begin{array}{c}
 \text{Forwards} \\
 \text{(Wrong Way)} \\
 : 
 \end{array}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \\ 0 & 0 & 0 & 0 & \square \end{array} \right] \\
 \mathbf{R}
 \end{array}
 \xrightarrow{\mathbf{G}_{12}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square \\ 0 & 0 & 0 & \square & \square \end{array} \right] \\
 \mathbf{G}_{12}\mathbf{R}
 \end{array}
 \xrightarrow{\mathbf{G}_{23}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & 0 & 0 & 0 & \square \end{array} \right] \\
 \mathbf{G}_{23}\mathbf{G}_{12}\mathbf{R}
 \end{array}
 \\
 \xrightarrow{\mathbf{G}_{34}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ 0 & 0 & 0 & 0 & \square \end{array} \right] \\
 \mathbf{G}_{34}\mathbf{G}_{23}\mathbf{G}_{12}\mathbf{R}
 \end{array}
 \xrightarrow{\mathbf{G}_{45}}
 \begin{array}{c}
 \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{array} \right] \\
 \mathbf{G}_{45}\mathbf{G}_{34}\mathbf{G}_{23}\mathbf{G}_{12}\mathbf{R}
 \end{array}.
 \end{array}$$

I.e., the backward rotation will keep a lot of the zeros as they are, whereas the forward rotation will destroy these zeros. Generally, the backward rotation results in,

$$\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n}(\mathbf{R} + \mathbf{wv}^\top) = \mathbf{H}_0 \pm \|\mathbf{w}\|\mathbf{e}_1\mathbf{v}^\top = \mathbf{H},$$

which is also upper Hessenberg. Similar to triangularization via the Givens rotation in Section 3.15 (p. 115), there exists a set of rotations  $\mathbf{J}_{12}, \mathbf{J}_{23}, \dots, \mathbf{J}_{(n-1),n}$  such that

$$\mathbf{J}_{(n-1),n}\dots\mathbf{J}_{23}\mathbf{J}_{12}\mathbf{H} = \mathbf{R}',$$

is upper triangular. Following from the  $5 \times 5$  example, the triangularization is shown as follows

$$\underbrace{\mathbf{H}_0 \pm \|\mathbf{w}\| \mathbf{e}_1 \mathbf{v}^\top}_{\mathbf{H}} = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{J}_{12}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{J}_{23}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \\
 \mathbf{H} \qquad \qquad \qquad \mathbf{J}_{12}\mathbf{H} \qquad \qquad \qquad \mathbf{J}_{23}\mathbf{J}_{12}\mathbf{H} \\
 \xrightarrow{\mathbf{J}_{34}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{J}_{45}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \mathbf{0} & \boxtimes \end{bmatrix} \\
 \mathbf{J}_{34}\mathbf{J}_{23}\mathbf{J}_{12}\mathbf{H} \qquad \qquad \qquad \mathbf{J}_{45}\mathbf{J}_{34}\mathbf{J}_{23}\mathbf{J}_{12}\mathbf{H}$$

And the QR decomposition of  $\mathbf{A}'$  is thus given by

$$\mathbf{A}' = \mathbf{Q}' \mathbf{R}',$$

where

$$\left\{ \begin{array}{l} \mathbf{R}' = (\mathbf{J}_{(n-1),n} \dots \mathbf{J}_{23}\mathbf{J}_{12})(\mathbf{G}_{12}\mathbf{G}_{23} \dots \mathbf{G}_{(n-1),n})(\mathbf{R} + \mathbf{w}\mathbf{v}^\top); \\ \mathbf{Q}' = \mathbf{Q} \{(\mathbf{J}_{(n-1),n} \dots \mathbf{J}_{23}\mathbf{J}_{12})(\mathbf{G}_{12}\mathbf{G}_{23} \dots \mathbf{G}_{(n-1),n})\}^\top; \\ \text{(or)} \quad \mathbf{Q}'^\top = \{(\mathbf{J}_{(n-1),n} \dots \mathbf{J}_{23}\mathbf{J}_{12})(\mathbf{G}_{12}\mathbf{G}_{23} \dots \mathbf{G}_{(n-1),n})\} \mathbf{Q}^\top. \end{array} \right. \quad (3.18)$$

The procedure is then formulated in Algorithm 22.

### Theorem 3.2: (Algorithm Complexity: QR Rank-One Change)

Algorithm 22 requires  $\sim 8n^2$  flops to compute a full QR decomposition of an  $\mathbf{A}' \in \mathbb{R}^{n \times n}$  matrix with rank-one change to  $\mathbf{A}$  and the full QR decomposition of  $\mathbf{A}$  is known. Further, if  $\mathbf{Q}'$  is needed explicitly, additional  $\sim 12n^2$  flops are required.

**Proof** [of Theorem 3.2] It is trivial that step 1 needs  $\boxed{(*)} \cdot n(2n - 1) = 2n^2 - n$  flops to calculate  $\mathbf{w} = \mathbf{Q}^\top \mathbf{u}$ .

For step 5, each iteration  $i$  requires 6 flops (which are 2 square operations, 1 addition, 1 square root, and 2 divisions). And there are  $n - 1$  such iterations so that the complexity for all the step 5's is  $\boxed{6(n - 1)}$  flops.

For each iteration  $i$ , step 7 and step 8 operate on two length- $(n - i + 1)$  vectors. The two steps take  $6(n - i + 1)$  flops for each iteration  $i$  (which are  $4(n - i + 1)$  multiplications and  $2(n - i + 1)$  additions). Let  $f(i) = 6(n - i + 1)$ , the total complexity for the two steps is equal to

$$\text{cost} = f(1) + f(2) + \dots + f(n - 1) = \boxed{3n^2 - 3n} \text{ flops.}$$

**Algorithm 22** QR Rank-One Changes

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with QR decomposition  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ , and  $\mathbf{A}' = \mathbf{A} + \mathbf{u}\mathbf{v}^\top$ ;

- 1: Calculate  $\mathbf{w} \leftarrow \mathbf{Q}^\top \mathbf{u}$ ; ▷  $2n^2 - n$  flops
- 2: Calculate  $\mathbf{H} \leftarrow \mathbf{R}$ ;
- 3: **for**  $i = n - 1$  to 1 **do**
- 4:     Get Givens rotation  $\mathbf{G}_{i,i+1}$  with the following parameters  $c, s$ :
- 5:      $c = \frac{x_k}{\sqrt{x_k^2 + x_l^2}}, s = \frac{x_l}{\sqrt{x_k^2 + x_l^2}}$  where  $x_k = \mathbf{w}_i, x_l = \mathbf{w}_{i-1}$ ; ▷ 6 flops
- 6:     Calculate  $\mathbf{H} = \mathbf{G}_{i,i+1} \mathbf{H}$  in following two steps:
- 7:         *i*-th row:  $\mathbf{H}_{i,:} = c \cdot \mathbf{H}_{i,:} + s \mathbf{H}_{j,:}$ , where  $j = i + 1$ ; ▷  $3(n - i + 1)$  flops
- 8:         ( $i + 1$ )-th row:  $\mathbf{H}_{i+1,:} = -s \cdot \mathbf{H}_{i,:} + c \mathbf{H}_{j,:}$ , where  $j = i + 1$ ; ▷  $3(n - i + 1)$  flops
- 9:     **end for**
- 10:    Set  $\mathbf{R}' = \mathbf{H} \pm \|\mathbf{w}\| \mathbf{e}_1 \mathbf{v}^\top$ ; ▷  $\mathbf{H}, \mathbf{R}'$  are both upper Hessenberg
- 11:    **for**  $i = 1$  to  $n - 1$  **do**
- 12:         Get Givens rotation  $\mathbf{J}_{i,i+1}$  with the following parameters  $c, s$ :
- 13:          $c = \frac{x_k}{\sqrt{x_k^2 + x_l^2}}, s = \frac{x_l}{\sqrt{x_k^2 + x_l^2}}$  where  $x_k = \mathbf{H}_{i,i}, x_l = \mathbf{H}_{i+1,i}$ ;
- 14:         Calculate  $\mathbf{R}' = \mathbf{J}_{i,i+1} \mathbf{R}'$  in following two steps:
- 15:             *i*-th row:  $\mathbf{R}'_{i,:} = c \cdot \mathbf{R}'_{i,:} + s \mathbf{R}'_{j,:}$ , where  $j = i + 1$ ;
- 16:             ( $i + 1$ )-th row:  $\mathbf{R}'_{i+1,:} = -s \cdot \mathbf{R}'_{i,:} + c \mathbf{R}'_{j,:}$ , where  $j = i + 1$ ;
- 17:         **end for**
- 18:         Output  $\mathbf{R}'$ ;
- 19:         Set  $\mathbf{Q}'^\top = \mathbf{Q}^\top$ ;
- 20:         **for**  $i = n - 1$  to 1 **do** ▷ The following  $c, s$  are from step 5
- 21:             *i*-th row:  $\mathbf{Q}'^\top_{i,:} = c \cdot \mathbf{Q}'^\top_{i,:} + s \mathbf{Q}'^\top_{j,:}$ , where  $j = i + 1$ ; ▷  $6n$  flops
- 22:             ( $i + 1$ )-th row:  $\mathbf{Q}'^\top_{i+1,:} = -s \cdot \mathbf{Q}'^\top_{i,:} + c \mathbf{Q}'^\top_{j,:}$ , where  $j = i + 1$ ; ▷  $6n$  flops
- 23:         **end for**
- 24:         **for**  $i = 1$  to  $n - 1$  **do** ▷ The following  $c, s$  are from step 13
- 25:             *i*-th row:  $\mathbf{Q}'^\top_{i,:} = c \cdot \mathbf{Q}'^\top_{i,:} + s \mathbf{Q}'^\top_{j,:}$ , where  $j = i + 1$ ; ▷  $6n$  flops
- 26:             ( $i + 1$ )-th row:  $\mathbf{Q}'^\top_{i+1,:} = -s \cdot \mathbf{Q}'^\top_{i,:} + c \mathbf{Q}'^\top_{j,:}$ , where  $j = i + 1$ ; ▷  $6n$  flops
- 27:         **end for**
- 28:         Output  $\mathbf{Q}'$ ;

---

Therefore, the complexity for step 3 to step 9 is  $(*)$ .  $6(n - 1) + 3n^2 - 3n = 3n^2 + 3n - 6$  flops. Similarly, the complexity for step 11 to step 16 is again  $(*)$ .  $3n^2 + 3n - 6$  flops.

However, for each iteration  $i$ , step 21 and step 22 operate on two length- $n$  vectors. The two steps take  $6n$  flops for each iteration  $i$ , and the total complexity for step 20 to step 23 is  $(*)$ .  $6n(n - 1) = 6n^2 - 6n$  flops. Again, step 24 to step 27 take another  $(*)$ .  $6n^2 - 6n$  flops.

Therefore, we can calculate the final complexity by summing up the equations marked as  $(*)$  with  $20n^2$  flops if keep only the leading terms ( $8n^2$  flops for calculating  $\mathbf{R}'$ , and  $12n^2$  flops for calculating  $\mathbf{Q}'$ ).

Note that for each iteration  $i$ , step 7 and step 8 operate on two length- $(n - i + 1)$  vectors since  $\mathbf{R}$  is upper triangular. If we do not favor such a structure, the final complexity is  $26n^2$

flops as stated in (Golub and Van Loan, 2013). ■

The algorithm can be easily applied to a rectangular matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , or  $\mathbf{A} + \mathbf{U}\mathbf{V}^\top$  where  $\mathbf{U} \in \mathbb{R}^{m \times k}$  and  $\mathbf{V} \in \mathbb{R}^{n \times k}$ .

### 3.20.3 Application: Appending or Deleting a Column

**Deleting a column** Suppose the QR decomposition of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is given by  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  where the column partition of  $\mathbf{A}$  is  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ . Now, if we delete the  $k$ -th column of  $\mathbf{A}$  such that  $\mathbf{A}' = [\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{a}_{k+1}, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times (n-1)}$ . We want to find the QR decomposition of  $\mathbf{A}'$  efficiently. Suppose further  $\mathbf{R}$  has the following form

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{a} & \mathbf{R}_{12} \\ \mathbf{0} & r_{kk} & \mathbf{b}^\top \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \begin{matrix} k-1 \\ 1 \\ m-k \end{matrix} \begin{matrix} \\ \\ \\ k-1 & 1 & n-k \end{matrix}.$$

Apparently,

$$\mathbf{Q}^\top \mathbf{A}' = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{b}^\top \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} = \mathbf{H}$$

is upper Hessenberg. A  $6 \times 5$  example is shown as follows where  $k = 3$ :

$$\begin{array}{c} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \rightarrow \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{R} = \mathbf{Q}^\top \mathbf{A} \qquad \qquad \qquad \mathbf{H} = \mathbf{Q}^\top \mathbf{A}' \end{array}.$$

Again, for columns  $k$  to  $n-1$  of  $\mathbf{H}$ , there exists a set of rotations  $\mathbf{G}_{k,k+1}, \mathbf{G}_{k+1,k+2}, \dots, \mathbf{G}_{n-1,n}$  that could introduce zeros for the elements  $h_{k+1,k}, h_{k+2,k+1}, \dots, h_{n,n-1}$  of  $\mathbf{H}$ . The the triangular matrix  $\mathbf{R}'$  is given by

$$\mathbf{R}' = \mathbf{G}_{n-1,n} \dots \mathbf{G}_{k+1,k+2} \mathbf{G}_{k,k+1} \mathbf{Q}^\top \mathbf{A}'.$$

And the orthogonal matrix

$$\mathbf{Q}' = (\mathbf{G}_{n-1,n} \dots \mathbf{G}_{k+1,k+2} \mathbf{G}_{k,k+1} \mathbf{Q}^\top)^\top = \mathbf{Q} \mathbf{G}_{k,k+1}^\top \mathbf{G}_{k+1,k+2}^\top \dots \mathbf{G}_{n-1,n}^\top, \quad (3.19)$$

such that  $\mathbf{A}' = \mathbf{Q}'\mathbf{R}'$ . The procedure is formulated in Algorithm 23. And the  $6 \times 5$  example is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface**

indicates the value has just been changed:

$$\begin{array}{c}
 \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{k=3} \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{G_{34}} \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \textbf{0} & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{G_{45}} \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & \textbf{0} \\ 0 & 0 & 0 & 0 \end{array} \right] \\
 \mathbf{R} = \mathbf{Q}^\top \mathbf{A} \qquad \mathbf{H} = \mathbf{Q}^\top \mathbf{A}' \qquad \mathbf{G}_{34}\mathbf{H} \qquad \mathbf{G}_{45}\mathbf{G}_{34}\mathbf{H}
 \end{array}$$

---

**Algorithm 23** QR Deleting a Column

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with full QR decomposition  $\mathbf{A} = \mathbf{QR}$ , and  $\mathbf{A}' \in \mathbb{R}^{m \times (n-1)}$  by deleting column  $k$  of  $\mathbf{A}$ ;

- 1: Obtain  $\mathbf{H}$  by deleting  $k$  of  $\mathbf{R}$ , that is,  $\mathbf{H} = \mathbf{Q}^\top \mathbf{A}'$ ;
- 2: **for**  $i = k$  to  $n - 1$  **do**
- 3:   Get Givens rotation  $\mathbf{G}_{i,i+1}$  with the following parameters  $c, s$ :
- 4:    $c = \frac{x_k}{\sqrt{x_k^2 + x_l^2}}, s = \frac{x_l}{\sqrt{x_k^2 + x_l^2}}$  where  $x_k = h_{ii}, x_l = h_{i+1,i}$ ;
- 5:   Calculate  $\mathbf{H} = \mathbf{G}_{i,i+1}\mathbf{H}$  in following two steps:
- 6:    $i$ -th row:  $\mathbf{H}_{i,:} = c \cdot \mathbf{H}_{i,:} + s\mathbf{H}_{j,:}$ , where  $j = i + 1$ ;
- 7:    $(i + 1)$ -th row:  $\mathbf{H}_{i+1,:} = -s \cdot \mathbf{H}_{i,:} + c\mathbf{H}_{j,:}$ , where  $j = i + 1$ ;
- 8: **end for**
- 9: Set  $\mathbf{R}' \leftarrow \mathbf{H}$  and output  $\mathbf{R}'$ ;
- 10: Set  $\mathbf{Q}' \leftarrow \mathbf{Q}^\top$ ;
- 11: **for**  $i = k$  to  $n - 1$  **do**
- 12:    $c = \frac{x_k}{\sqrt{x_k^2 + x_l^2}}, s = \frac{x_l}{\sqrt{x_k^2 + x_l^2}}$  where  $x_k, x_l$  are from step 4;
- 13:   Calculate  $\mathbf{Q}' = \mathbf{G}_{i,i+1}\mathbf{Q}'$  in following two steps:
- 14:    $i$ -th row:  $\mathbf{Q}'_{i,:} = c \cdot \mathbf{Q}'_{i,:} + s\mathbf{Q}'_{j,:}$ , where  $j = i + 1$ ;
- 15:    $(i + 1)$ -th row:  $\mathbf{Q}'_{i+1,:} = -s \cdot \mathbf{Q}'_{i,:} + c\mathbf{Q}'_{j,:}$ , where  $j = i + 1$ ;
- 16: **end for**
- 17: Output  $\mathbf{Q}' \leftarrow \mathbf{Q}'^\top$  from Equation (3.19);

---

**Theorem 3.3: (Algorithm Complexity: QR Deleting Column)**

Algorithm 23 requires  $\sim 3n^2 - 6nk + 3k^2$  flops to compute a full QR decomposition of an  $\mathbf{A}' \in \mathbb{R}^{m \times (n-1)}$  matrix where we delete the column  $k$  of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and the full QR decomposition of  $\mathbf{A}$  is known. Further, if  $\mathbf{Q}'$  is needed explicitly, additional  $\sim 6m(n-k)$  flops required.

**Proof** [of Theorem 3.3] For step 4, each iteration  $i$  requires 6 flops (which are 2 square operations, 1 addition, 1 square root, and 2 divisions). And there are  $n - k$  such iterations so that the complexity for all the step 5's is  $\boxed{6(n - k)}$  flops.

For each iteration  $i$ , step 6 and step 7 operate on two length- $(n - i)$  vectors since  $\mathbf{H}$  is upper Hessenberg. The two steps take  $6(n - i)$  flops for each iteration  $i$  (which are  $4(n - i)$  multiplications and  $2(n - i)$  additions). Let  $f(i) = 6(n - i)$ , the total complexity for the two steps is equal to

$$\text{cost} = f(k) + f(k+1) + \dots + f(n-1) = [3n^2 - 6nk + 3k^2 + 3n - 3k] \text{ flops.}$$

Therefore, the complexity for step 2 to step 8 is  $[(*). 3n^2 - 6nk + 3k^2]$  flops if we keep only the leading terms.

However, for each iteration  $i$ , step 14 and step 15 operate on two length- $m$  vectors. The two steps take  $6m$  flops for each  $i$ , and there are  $n - k$  such iterations so that the total complexity for step 11 to step 16 is  $[(*). 6m(n - k)]$  flops. ■

Note that the number of column  $k$  takes a role in the complexity, when  $k = n$ , the complexity is 0; and when  $k = 1$ , the complexity gets its maximal value.

**Appending a column** Similarly, suppose  $\tilde{\mathbf{A}} = [\mathbf{a}_1, \mathbf{a}_k, \mathbf{w}, \mathbf{a}_{k+1}, \dots, \mathbf{a}_n]$  where we append  $\mathbf{w}$  into the  $(k + 1)$ -th column of  $\mathbf{A}$ . We can obtain

$$\mathbf{Q}^\top \tilde{\mathbf{A}} = [\mathbf{Q}^\top \mathbf{a}_1, \dots, \mathbf{Q}^\top \mathbf{a}_k, \mathbf{Q}^\top \mathbf{w}, \mathbf{Q}^\top \mathbf{a}_{k+1}, \dots, \mathbf{Q}^\top \mathbf{a}_n] = \tilde{\mathbf{H}}.$$

A set of Givens rotations  $\mathbf{J}_{m-1,m}, \mathbf{J}_{m-2,m-1}, \dots, \mathbf{J}_{k+1,k+2}$  can introduce zeros for the  $\tilde{h}_{m,k+1}, \tilde{h}_{m-1,k+1}, \dots, \tilde{h}_{k+2,k+1}$  elements of  $\tilde{\mathbf{H}}$  such that

$$\tilde{\mathbf{R}} = \mathbf{J}_{k+1,k+2} \dots \mathbf{J}_{m-2,m-1} \mathbf{J}_{m-1,m} \mathbf{Q}^\top \tilde{\mathbf{A}},$$

is upper triangular. Suppose  $\tilde{\mathbf{H}}$  is of size  $6 \times 5$  and  $k = 2$ , an example is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{c} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & 0 & \boxtimes \\ 0 & 0 & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & 0 & 0 \end{bmatrix} \xrightarrow{\mathbf{J}_{56}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & 0 & \boxtimes \\ 0 & 0 & \boxtimes & 0 & 0 \\ 0 & 0 & \mathbf{0} & 0 & 0 \end{bmatrix} \xrightarrow{\mathbf{J}_{45}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{\boxtimes} & 0 & \boxtimes \\ 0 & 0 & \mathbf{0} & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \tilde{\mathbf{H}} \qquad \mathbf{J}_{56} \tilde{\mathbf{H}} \rightarrow \tilde{h}_{63} = 0 \qquad \mathbf{J}_{45} \mathbf{J}_{56} \tilde{\mathbf{H}} \rightarrow \tilde{h}_{53} = 0 \end{array}$$

$$\xrightarrow{\mathbf{J}_{34}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \tilde{\mathbf{R}}.$$

$$\mathbf{J}_{34} \mathbf{J}_{45} \mathbf{J}_{56} \tilde{\mathbf{H}} \rightarrow \tilde{h}_{43} = 0$$

And finally, the orthogonal matrix

$$\tilde{\mathbf{Q}} = (\mathbf{J}_{k+1,k+2} \dots \mathbf{J}_{m-2,m-1} \mathbf{J}_{m-1,m} \mathbf{Q}^\top)^\top = \mathbf{Q} \mathbf{J}_{m-1,m}^\top \mathbf{J}_{m-2,m-1}^\top \dots \mathbf{J}_{k+1,k+2}^\top, \quad (3.20)$$

such that  $\tilde{\mathbf{A}} = \tilde{\mathbf{Q}} \tilde{\mathbf{R}}$ . The procedure is again formulated in Algorithm 24.

**Algorithm 24** QR Adding a Column

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with full QR decomposition  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ , and  $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times (n+1)}$  by adding column  $\mathbf{w}$  into  $(k+1)$ -th column of  $\mathbf{A}$ ;

- 1: Calculate  $\mathbf{Q}^\top \mathbf{w}$ ;
- 2: Obtain  $\tilde{\mathbf{H}}$  by inserting  $\mathbf{Q}^\top \mathbf{w}$  into  $(k+1)$ -th column of  $\mathbf{R}$ ;
- 3: **for**  $i = m-1$  to  $k+1$  **do**
- 4:     Get Givens rotation  $\mathbf{J}_{i,i+1}$  with the following parameters  $c, s$ :
- 5:      $c = \frac{x_k}{\sqrt{x_k^2+x_l^2}}, s = \frac{x_l}{\sqrt{x_k^2+x_l^2}}$  where  $x_k = \tilde{h}_{i,k+1}, x_l = \tilde{h}_{i+1,k+1}$ ;
- 6:     Calculate  $\tilde{\mathbf{H}} = \mathbf{J}_{i,i+1} \mathbf{H}$  in following two steps:
- 7:         *i*-th row:  $\tilde{\mathbf{H}}_{i,:} = c \cdot \mathbf{H}_{i,:} + s \tilde{\mathbf{H}}_{j,:}$ , where  $j = i+1$ ;
- 8:         ( $i+1$ )-th row:  $\tilde{\mathbf{H}}_{i+1,:} = -s \cdot \mathbf{H}_{i,:} + c \tilde{\mathbf{H}}_{j,:}$ , where  $j = i+1$ ;
- 9:     **end for**
- 10: Set  $\tilde{\mathbf{R}} \leftarrow \tilde{\mathbf{H}}$  and output  $\tilde{\mathbf{R}}$ ;
- 11: Set  $\tilde{\mathbf{Q}} \leftarrow \mathbf{Q}^\top$ ;
- 12: **for**  $i = m-1$  to  $k+1$  **do**
- 13:      $c = \frac{x_k}{\sqrt{x_k^2+x_l^2}}, s = \frac{x_l}{\sqrt{x_k^2+x_l^2}}$  where  $x_k, x_l$  are from step 5;
- 14:     Calculate  $\tilde{\mathbf{Q}} = \mathbf{J}_{i,i+1} \tilde{\mathbf{Q}}$  in following two steps:
- 15:         *i*-th row:  $\tilde{\mathbf{Q}}_{i,:} = c \cdot \tilde{\mathbf{Q}}_{i,:} + s \tilde{\mathbf{Q}}_{j,:}$ , where  $j = i+1$ ;
- 16:         ( $i+1$ )-th row:  $\tilde{\mathbf{Q}}_{i+1,:} = -s \cdot \tilde{\mathbf{Q}}_{i,:} + c \tilde{\mathbf{Q}}_{j,:}$ , where  $j = i+1$ ;
- 17:     **end for**
- 18: Output  $\tilde{\mathbf{Q}} \leftarrow \tilde{\mathbf{Q}}^\top$  from Equation (3.20);

---

**Theorem 3.4: (Algorithm Complexity: QR Adding Column)**

Algorithm 24 requires  $\sim 2m^2 + 6(mn + k^2 - nk - mk)$  flops to compute a full QR decomposition of an  $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times (n+1)}$  matrix where we add a column into the  $(k+1)$ -th column of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and the full QR decomposition of  $\mathbf{A}$  is known. Further, if  $\mathbf{Q}$  is needed explicitly, additional  $\sim 6m(m-k)$  flops required.

**Proof** [of Theorem 3.4] It is trivial that step 1 requires  $m(2m-1)$  flops to calculate  $\mathbf{Q}^\top \mathbf{w}$ .

For step 5, each iteration  $i$  requires 6 flops (which are 2 square operations, 1 addition, 1 square root, and 2 divisions). And there are  $m-k-1$  such iterations so that the complexity for all the step 5's is  $6(m-k-1)$  flops.

For each iteration  $i$ , step 7 and step 8 operate on two length- $(n-k+1)$  vectors since  $\tilde{\mathbf{H}}$  is upper Hessenberg. The two steps take  $6(n-k+1)$  flops for each iteration  $i$  (which are  $4(n-k+1)$  multiplications and  $2(n-k+1)$  additions). And there are  $m-k-1$  such iterations so that the complexity for all the step 7 and step 8's is  $6(n-k+1)(m-k-1)$  flops.

Therefore, the complexity for step 1 to step 9 is

$$m(2m-1) + 6(m-k-1) + 6(n-k+1)(m-k-1) = [2m^2 + 6(n-k+2)(m-k-1)],$$

or  $2m^2 + 6(mn + k^2 - nk - mk)$  flops if we keep only the leading terms.

However, for each iteration  $i$ , step 15 and step 16 operate on two length- $m$  vectors. The two steps take  $6m$  flops for each  $i$ , and there are  $m - k - 1$  such iterations so that the total complexity for step 12 to step 17 is  $6m(m - k - 1)$  flops or  $6m(m - k)$  flops if we keep only the leading terms.  $\blacksquare$

**Real world application** The method introduced above is useful for the efficient variable selection in the least squares problem via the QR decomposition. At each time we delete a column of the data matrix  $\mathbf{A}$ , and apply an  $F$ -test to see if the variable is significant or not. If not, we will delete the variable and favor a simpler model. A short review is given as follows and more details can be referred to (Lu, 2021d).

Following the setup in Section 3.20.1, let's consider the overdetermined system  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the data matrix,  $\mathbf{b} \in \mathbb{R}^m$  with  $m > n$  is the observation matrix. The LS solution is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  for minimizing  $\|\mathbf{Ax} - \mathbf{b}\|^2$ , where  $\mathbf{A}^\top \mathbf{A}$  is invertible since  $\mathbf{A}$  has full column rank and  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$ .

Suppose we delete a column of  $\mathbf{A}$  into  $\widehat{\mathbf{A}}$ , the LS solution is reduced from  $\mathbf{x}_{LS}$  to  $\widehat{\mathbf{x}}_{LS}$ . Define

$$\begin{aligned} RSS(\widehat{\mathbf{x}}_{LS}) &= \|\mathbf{b} - \widehat{\mathbf{b}}_{LS}\|^2, \quad \text{where } \widehat{\mathbf{b}}_{LS} = \widehat{\mathbf{A}}\widehat{\mathbf{x}}_{LS}, \\ RSS(\mathbf{x}_{LS}) &= \|\mathbf{b} - \mathbf{b}_{LS}\|^2, \quad \text{where } \mathbf{b}_{LS} = \mathbf{A}\mathbf{x}_{LS}, \\ \mathbf{H} &= \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top, \\ \widehat{\mathbf{H}} &= \widehat{\mathbf{A}}(\widehat{\mathbf{A}}^\top \widehat{\mathbf{A}})^{-1} \widehat{\mathbf{A}}^\top. \end{aligned}$$

Suppose the *reduced* QR decomposition of  $\mathbf{A}, \widehat{\mathbf{A}}$  are given by  $\mathbf{A} = \mathbf{QR}$ ,  $\widehat{\mathbf{A}} = \widehat{\mathbf{Q}}\widehat{\mathbf{R}}$ . Thus  $RSS(\mathbf{x}_{LS}) = \mathbf{b}^\top (\mathbf{I} - \mathbf{H})\mathbf{b} = \mathbf{b}^\top \mathbf{b} - (\mathbf{b}^\top \mathbf{Q})(\mathbf{Q}^\top \mathbf{b})$  and  $RSS(\widehat{\mathbf{x}}_{LS}) - RSS(\mathbf{x}_{LS}) = \|\mathbf{b}_{LS} - \widehat{\mathbf{b}}_{LS}\|^2 = \mathbf{b}^\top (\mathbf{H} - \widehat{\mathbf{H}})\mathbf{b} = (\mathbf{b}^\top \mathbf{Q})(\mathbf{Q}^\top \mathbf{b}) - (\mathbf{b}^\top \widehat{\mathbf{Q}})(\widehat{\mathbf{Q}}^\top \mathbf{b})$ , which are the differences of two inner products. It can be shown that  $RSS(\mathbf{x}_{LS}) \sim \sigma^2 \chi^2_{(m-n)}$  which is a Chi-square distribution and  $\sigma$  is the noise level. Under the hypothesis that the deleted column is not significant, we could conclude that

$$T = \frac{\frac{1}{n-q} (RSS(\widehat{\mathbf{x}}_{LS}) - RSS(\mathbf{x}_{LS}))}{\frac{1}{m-n} RSS(\mathbf{x}_{LS})} \sim F_{n-q, m-n},$$

which is the **test statistic for  $F$ -test** with  $q = n - 1$ . Suppose we have the data set  $(\mathbf{a}_1, b_1), (\mathbf{a}_2, b_2), \dots, (\mathbf{a}_n, b_m)$ , and we observe  $T = t$  for this specific data set. Then

$$p = P[T((\mathbf{a}_1, b_1), (\mathbf{a}_2, b_2), \dots, (\mathbf{a}_n, b_m)) \geq t] = P[F_{n-q, m-n} \geq t].$$

We reject the hypothesis if  $p < \alpha$ , for some small  $\alpha$ , say 0.05. This is called the *p-value*.

### 3.20.4 Application: Appending or Deleting a Row

**Appending a row** Suppose the full QR decomposition of  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is given by  $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \mathbf{QR}$  where  $\mathbf{A}_1 \in \mathbb{R}^{k \times n}$  and  $\mathbf{A}_2 \in \mathbb{R}^{(m-k) \times n}$ . Now, if we add a row such that

$\mathbf{A}' = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{w}^\top \\ \mathbf{A}_2 \end{bmatrix} \in \mathbb{R}^{(m+1) \times n}$ . We want to find the full QR decomposition of  $\mathbf{A}'$  efficiently.

Construct a permutation matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{m-k} \end{bmatrix} \longrightarrow \mathbf{P} \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{w}^\top \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{w}^\top \\ \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}.$$

Then,

$$\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^\top \end{bmatrix} \mathbf{P} \mathbf{A}' = \begin{bmatrix} \mathbf{w}^\top \\ \mathbf{R} \end{bmatrix} = \mathbf{H}$$

is upper Hessenberg. Similarly, a set of rotations  $\mathbf{G}_{12}, \mathbf{G}_{23}, \dots, \mathbf{G}_{n,n+1}$  can be applied to introduce zeros for the elements  $h_{21}, h_{32}, \dots, h_{n+1,n}$  of  $\mathbf{H}$ . The triangular matrix  $\mathbf{R}'$  is given by

$$\mathbf{R}' = \mathbf{G}_{n,n+1} \dots \mathbf{G}_{23} \mathbf{G}_{12} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^\top \end{bmatrix} \mathbf{P} \mathbf{A}'.$$

And the orthogonal matrix

$$\mathbf{Q}' = \left( \mathbf{G}_{n,n+1} \dots \mathbf{G}_{23} \mathbf{G}_{12} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^\top \end{bmatrix} \mathbf{P} \right)^\top = \mathbf{P}^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \mathbf{G}_{12}^\top \mathbf{G}_{23}^\top \dots \mathbf{G}_{n,n+1}^\top,$$

such that  $\mathbf{A}' = \mathbf{Q}' \mathbf{R}'$ .

**Deleting a row** Suppose  $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{w}^\top \\ \mathbf{A}_2 \end{bmatrix} \in \mathbb{R}^{m \times n}$  where  $\mathbf{A}_1 \in \mathbb{R}^{k \times n}$ ,  $\mathbf{A}_2 \in \mathbb{R}^{(m-k-1) \times n}$

with the full QR decomposition given by  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  where  $\mathbf{Q} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{R} \in \mathbb{R}^{m \times n}$ . We want to compute the full QR decomposition of  $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$  efficiently (assume  $m-1 \geq n$ ).

Analogously, we can construct a permutation matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{m-k-1} \end{bmatrix}$$

such that

$$\mathbf{P} \mathbf{A} = \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{m-k-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{w}^\top \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{w}^\top \\ \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \mathbf{P} \mathbf{Q} \mathbf{R} = \mathbf{M} \mathbf{R},$$

where  $\mathbf{M} = \mathbf{P} \mathbf{Q}$  is an orthogonal matrix. Let  $\mathbf{m}^\top$  be the first row of  $\mathbf{M}$ , and a set of givens rotations  $\mathbf{G}_{m-1,m}, \mathbf{G}_{m-2,m-1}, \dots, \mathbf{G}_{1,2}$  introducing zeros for elements  $m_m, m_{m-1}, \dots, m_2$  of  $\mathbf{m}$  respectively such that  $\mathbf{G}_{1,2} \dots \mathbf{G}_{m-2,m-1} \mathbf{G}_{m-1,m} \mathbf{m} = \alpha \mathbf{e}_1$  where  $\alpha = \pm 1$ . Therefore, we have

$$\mathbf{G}_{1,2} \dots \mathbf{G}_{m-2,m-1} \mathbf{G}_{m-1,m} \mathbf{R} = \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}_1 \end{bmatrix} \begin{matrix} 1 \\ m-1 \end{matrix},$$

which is upper Hessenberg with  $\mathbf{R}_1 \in \mathbb{R}^{(m-1) \times n}$  being upper triangular. And

$$\mathbf{M}\mathbf{G}_{m-1,m}^\top \mathbf{G}_{m-2,m-1}^\top \dots \mathbf{G}_{1,2}^\top = \begin{bmatrix} \alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_1 \end{bmatrix},$$

where  $\mathbf{Q}_1 \in \mathbb{R}^{(m-1) \times (m-1)}$  is an orthogonal matrix. The bottom-left block of the above matrix is a zero vector since  $\alpha = \pm 1$  and  $\mathbf{M}$  is orthogonal. To see this, let  $\mathbf{G} = \mathbf{G}_{m-1,m}^\top \mathbf{G}_{m-2,m-1}^\top \dots \mathbf{G}_{1,2}^\top$  with the first column being  $\mathbf{g}$  and  $\mathbf{M} = [\mathbf{m}^\top; \mathbf{m}_2^\top; \mathbf{m}_3^\top; \dots, \mathbf{m}_m^\top]$  being the row partition of  $\mathbf{M}$ . We have

$$\begin{aligned} \mathbf{m}^\top \mathbf{g} &= \pm 1 & \rightarrow & \mathbf{g} = \pm \mathbf{m}, \\ \mathbf{m}_i^\top \mathbf{m} &= 0, & \forall i \in \{2, 3, \dots, m\}. \end{aligned}$$

This results in

$$\begin{aligned} \mathbf{PA} &= \mathbf{MR} \\ &= (\mathbf{M}\mathbf{G}_{m-1,m}^\top \mathbf{G}_{m-2,m-1}^\top \dots \mathbf{G}_{1,2}^\top)(\mathbf{G}_{1,2} \dots \mathbf{G}_{m-2,m-1} \mathbf{G}_{m-1,m} \mathbf{R}) \\ &= \begin{bmatrix} \alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_1 \end{bmatrix} \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}_1 \end{bmatrix} = \begin{bmatrix} \alpha \mathbf{v}^\top \\ \mathbf{Q}_1 \mathbf{R}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{w}^\top \\ \tilde{\mathbf{A}} \end{bmatrix}. \end{aligned}$$

This implies  $\mathbf{Q}_1 \mathbf{R}_1$  is the full QR decomposition of  $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$ .

### 3.20.5 Application: Reducing the Ill-Condition via the QR decomposition

**Well-determined linear system** Consider the well-determined linear equation  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is nonsingular. The solution  $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$  exists and is unique. Now suppose the vector  $\mathbf{b}$  is perturbed by  $\delta\mathbf{b}$ , the solution is now given by

$$\mathbf{x} + \delta\mathbf{x} = \mathbf{A}^{-1}(\mathbf{b} + \delta\mathbf{b}) = \mathbf{A}^{-1}\mathbf{b} + \mathbf{A}^{-1}\delta\mathbf{b} = \mathbf{x} + \mathbf{A}^{-1}\delta\mathbf{b}.$$

That is  $\delta\mathbf{x} = \mathbf{A}^{-1}\delta\mathbf{b}$ . By properties of matrix-vector inequality of matrix 2-norm (see Appendix L.2), we have

$$\|\delta\mathbf{x}\| = \|\mathbf{A}^{-1}\delta\mathbf{b}\|_2 \leq \|\mathbf{A}^{-1}\|_2 \|\delta\mathbf{b}\|.$$

This is known as the **absolute error bound** of  $\|\delta\mathbf{x}\|$ . And if  $\|\mathbf{A}^{-1}\|_2$  is small, then small changes in  $\mathbf{b}$  (i.e.,  $\|\delta\mathbf{b}\|$  is small) will result in small changes in  $\mathbf{x}$ . However, if  $\|\mathbf{A}^{-1}\|_2$  is large, the changes in  $\mathbf{x}$  may be large as well.

Now we divide the above equation by  $\|\mathbf{x}\|$

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}^{-1}\|_2 \frac{\|\delta\mathbf{b}\|}{\|\mathbf{x}\|}.$$

By  $\|\mathbf{b}\| = \|\mathbf{Ax}\| \rightarrow \|\mathbf{x}\| \geq \frac{\|\mathbf{b}\|}{\|\mathbf{A}\|_2}$ , we have

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|} \leq \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2 \frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|}.$$

This is known as the **relative error bound** of  $\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}\|}$ . The product  $\|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2$  is called the **condition number** of  $\mathbf{A}$ , and is denoted as  $\kappa(\mathbf{A})$ :

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2.$$

Similarly, by the inequality of Frobenius norm  $\|\mathbf{b}\| = \|\mathbf{Ax}\| \rightarrow \|\mathbf{b}\| \leq \|\mathbf{A}\|_F \|\mathbf{x}\|$ , we can also define the condition number to be

$$\kappa(\mathbf{A}) = \|\mathbf{A}\|_F \|\mathbf{A}^{-1}\|_F.$$

We will only consider the first definition of the condition in this context. If the relative error in  $\mathbf{x}$  is not much larger than the relative error in  $\mathbf{b}$ , the matrix is said to be **well-conditioned**. That is, if the condition number  $\kappa(\mathbf{A})$  is small, the matrix  $\mathbf{A}$  is well-conditioned. Otherwise, the matrix is called **ill-conditioned**.

**Over-determined linear system** Now, we further consider the over-determined linear equation  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $m > n$ . Suppose  $\mathbf{A}$  has full column rank such that  $\mathbf{A}^\top \mathbf{A}$  is invertible. Then the unique least squares solution is given by

$$\mathbf{x} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b},$$

which results from the normal equation

$$(\mathbf{A}^\top \mathbf{A})\mathbf{x} = \mathbf{A}^\top \mathbf{b}.$$

It can be shown that the condition number is given by

$$\kappa(\mathbf{A}^\top \mathbf{A}) = \kappa(\mathbf{A})^2.$$

**Example 3.4 (Reducing ill-condition)** Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 + \delta & 1 \\ 1 & 1 + \delta \end{bmatrix},$$

where  $\delta$  is small. The condition number of  $\mathbf{A}$  is of order  $\delta^{-1}$ . If we use the QR decomposition  $\mathbf{A} = \mathbf{QR}$  to solve the linear equation, then

$$\kappa(\mathbf{Q}) = 1, \quad \kappa(\mathbf{A}) \rightarrow \kappa(\mathbf{Q}^\top \mathbf{A}) = \kappa(\mathbf{R}).$$

If one believes the condition number of  $\mathbf{R}$  is smaller than  $\mathbf{A}$ , then we would overcome the ill-condition problem. Specifically, two QR decomposition results of  $\mathbf{A}$  are given by

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}_1 \mathbf{R}_1 \\ &= \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} & \frac{1}{\sqrt{\delta^2+2\delta+2}} \\ \frac{1}{\sqrt{\delta^2+2\delta+2}} & -\frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \begin{bmatrix} \sqrt{\delta^2+2\delta+2} & \frac{\delta^2+\delta}{(1+\delta)\sqrt{\delta^2+2\delta+2}} + \frac{\sqrt{\delta^2+2\delta+2}}{1+\delta} \\ 0 & -\frac{\delta^2+2\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \end{aligned}$$

or

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}_2 \mathbf{R}_2 \\ &= \begin{bmatrix} q_{11} & -q_{12} \\ q_{21} & -q_{22} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix} \\ &= \begin{bmatrix} \frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} & -\frac{1}{\sqrt{\delta^2+2\delta+2}} \\ \frac{1}{\sqrt{\delta^2+2\delta+2}} & \frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \begin{bmatrix} \sqrt{\delta^2+2\delta+2} & \frac{\delta^2+\delta}{(1+\delta)\sqrt{\delta^2+2\delta+2}} + \frac{\sqrt{\delta^2+2\delta+2}}{1+\delta} \\ 0 & \frac{\delta^2+2\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \end{aligned}$$

or

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}_3 \mathbf{R}_3 \\ &= \begin{bmatrix} -q_{11} & q_{12} \\ -q_{21} & q_{22} \end{bmatrix} \begin{bmatrix} -r_{11} & -r_{12} \\ 0 & r_{22} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} & \frac{1}{\sqrt{\delta^2+2\delta+2}} \\ -\frac{1}{\sqrt{\delta^2+2\delta+2}} & -\frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \begin{bmatrix} -\sqrt{\delta^2+2\delta+2} & -\frac{\delta^2+\delta}{(1+\delta)\sqrt{\delta^2+2\delta+2}} - \frac{\sqrt{\delta^2+2\delta+2}}{1+\delta} \\ 0 & -\frac{\delta^2+2\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \end{aligned}$$

or

$$\begin{aligned} \mathbf{A} &= \mathbf{Q}_4 \mathbf{R}_4 \\ &= \begin{bmatrix} -q_{11} & -q_{12} \\ -q_{21} & -q_{22} \end{bmatrix} \begin{bmatrix} -r_{11} & -r_{12} \\ 0 & -r_{22} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} & -\frac{1}{\sqrt{\delta^2+2\delta+2}} \\ -\frac{1}{\sqrt{\delta^2+2\delta+2}} & \frac{1+\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \begin{bmatrix} -\sqrt{\delta^2+2\delta+2} & -\frac{\delta^2+\delta}{(1+\delta)\sqrt{\delta^2+2\delta+2}} - \frac{\sqrt{\delta^2+2\delta+2}}{1+\delta} \\ 0 & \frac{\delta^2+2\delta}{\sqrt{\delta^2+2\delta+2}} \end{bmatrix} \end{aligned}$$

Suppose  $\delta = 0.01$ , the condition number of  $\mathbf{A}$  is 200, and the condition number of  $\mathbf{R}_1$  or  $\mathbf{R}_2$  is 1.  $\square$

The example shown above solving linear equation via the QR decomposition can reduce the ill-condition problem, more details on the numerical stability topics can be referred to (Higham, 2002; Zhang, 2017; Golub and Van Loan, 2013; Boyd and Vandenberghe, 2018).

## Chapter 4

# UTV Decomposition: ULV and URV Decomposition

### Contents

---

|            |                                                                    |            |
|------------|--------------------------------------------------------------------|------------|
| <b>4.1</b> | <b>UTV Decomposition</b>                                           | <b>141</b> |
| <b>4.2</b> | <b>Complete Orthogonal Decomposition</b>                           | <b>145</b> |
| <b>4.3</b> | <b>Applications</b>                                                | <b>146</b> |
| 4.3.1      | Application: Least Squares via ULV/URV for Rank Deficient Matrices | 146        |
| 4.3.2      | Application: Row Rank equals Column Rank Again via UTV             | 148        |

---

## 4.1. UTV Decomposition

The UTV decomposition goes further by factoring the matrix into two orthogonal matrices  $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{V}$ , where  $\mathbf{U}, \mathbf{V}$  are orthogonal, whilst  $\mathbf{T}$  is (upper/lower) triangular.<sup>1</sup> The resulting  $\mathbf{T}$  supports rank estimation. The matrix  $\mathbf{T}$  can be lower triangular which results in the ULV decomposition, or it can be upper triangular which results in the URV decomposition. The UTV framework shares a similar form as the singular value decomposition (SVD, see Section 14, p. 264) and can be regarded as inexpensive alternatives to the SVD.

### Theorem 4.1: (Full ULV Decomposition)

Every  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$  can be factored as

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V},$$

where  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  are two orthogonal matrices, and  $\mathbf{L} \in \mathbb{R}^{r \times r}$  is a lower triangular matrix.

The existence of the ULV decomposition is from the QR and LQ decomposition.

**Proof** [of Theorem 4.1] For any rank  $r$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ , we can use a column permutation matrix  $\mathbf{P}$  (Definition 0.15, p. 19) such that the linearly independent columns of  $\mathbf{A}$  appear in the first  $r$  columns of  $\mathbf{AP}$ . Without loss of generality, we assume  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r$  are the  $r$  linearly independent columns of  $\mathbf{A}$  and

$$\mathbf{AP} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r, \mathbf{b}_{r+1}, \dots, \mathbf{b}_n].$$

Let  $\mathbf{Z} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r] \in \mathbb{R}^{m \times r}$ . Since any  $\mathbf{b}_i$  is in the column space of  $\mathbf{Z}$ , we can find a  $\mathbf{E} \in \mathbb{R}^{r \times (n-r)}$  such that

$$[\mathbf{b}_{r+1}, \mathbf{b}_{r+2}, \dots, \mathbf{b}_n] = \mathbf{ZE}.$$

That is,

$$\mathbf{AP} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r, \mathbf{b}_{r+1}, \dots, \mathbf{b}_n] = \mathbf{Z} [\mathbf{I}_r \quad \mathbf{E}],$$

where  $\mathbf{I}_r$  is an  $r \times r$  identity matrix. Moreover,  $\mathbf{Z} \in \mathbb{R}^{m \times r}$  has full column rank such that its full QR decomposition is given by  $\mathbf{Z} = \mathbf{U} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ , where  $\mathbf{R} \in \mathbb{R}^{r \times r}$  is an upper triangular matrix with full rank and  $\mathbf{U}$  is an orthogonal matrix. This implies

$$\mathbf{AP} = \mathbf{Z} [\mathbf{I}_r \quad \mathbf{E}] = \mathbf{U} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} [\mathbf{I}_r \quad \mathbf{E}] = \mathbf{U} \begin{bmatrix} \mathbf{R} & \mathbf{RE} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (4.1)$$

Since  $\mathbf{R}$  has full rank, this means  $[\mathbf{R} \quad \mathbf{RE}]$  also has full rank such that its full LQ decomposition is given by  $[\mathbf{L} \quad \mathbf{0}] \mathbf{V}_0$  where  $\mathbf{L} \in \mathbb{R}^{r \times r}$  is a lower triangular matrix and  $\mathbf{V}_0$  is an

---

1. These decompositions fall into a category known as the *double-sided orthogonal decomposition*. We will see, the UTV decomposition, complete orthogonal decomposition, and singular value decomposition are all in this notion.

orthogonal matrix. Substitute into Equation (4.1), we have

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}_0 \mathbf{P}^{-1}.$$

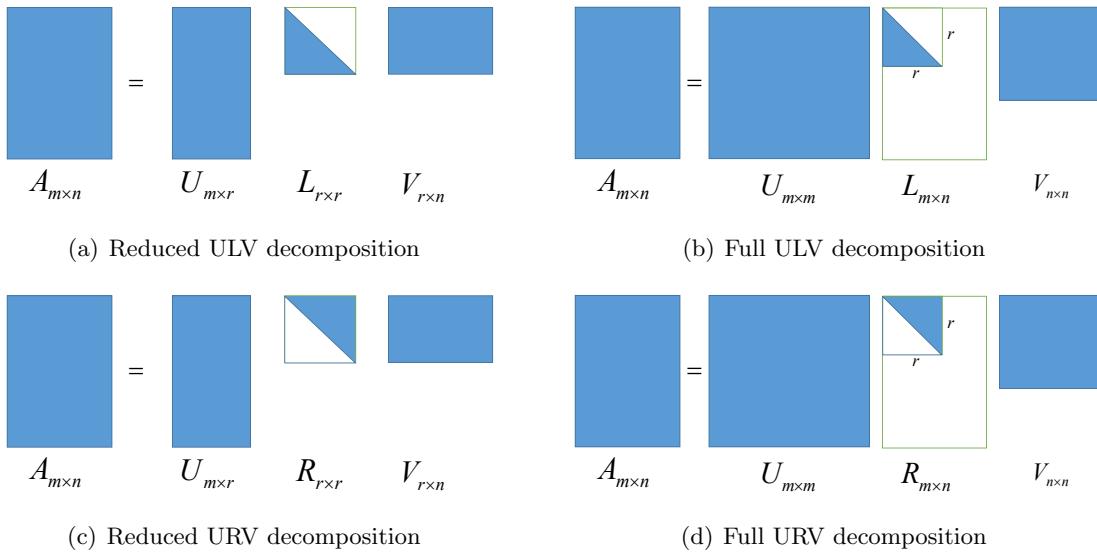
Let  $\mathbf{V} = \mathbf{V}_0 \mathbf{P}^{-1}$  which is a product of two orthogonal matrices, and is also an orthogonal matrix. This completes the proof.  $\blacksquare$

A second way to see the proof of the ULV decomposition will be discussed in the proof of Theorem 4.1 shortly via the rank-revealing QR decomposition and trivial QR decomposition.

Now suppose the ULV decomposition of matrix  $\mathbf{A}$  is

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}.$$

Let  $\mathbf{U}_0 = \mathbf{U}_{:,1:r}$  and  $\mathbf{V}_0 = \mathbf{V}_{1:r,:}$ , i.e.,  $\mathbf{U}_0$  contains only the first  $r$  columns of  $\mathbf{U}$ , and  $\mathbf{V}_0$  contains only the first  $r$  rows of  $\mathbf{V}$ . Then, we still have  $\mathbf{A} = \mathbf{U}_0 \mathbf{L} \mathbf{V}_0$ . This is known as the **reduced ULV decomposition**. The comparison between the reduced and the full ULV decomposition is shown in Figure 4.1 where white entries are zero and blue entries are not necessarily zero.



**Figure 4.1:** Comparison between the reduced and full ULV, and between the reduced and full URV.

Similarly, we can also claim the URV decomposition as follows.

**Theorem 4.2: (Full URV Decomposition)**

Every  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$  can be factored as

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V},$$

where  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  are two orthogonal matrices, and  $\mathbf{R} \in \mathbb{R}^{r \times r}$  is an upper triangular matrix.

**Proof** [of Theorem 4.2] For any rank  $r$  matrix  $\mathbf{A} = [\mathbf{a}_1^\top; \mathbf{a}_2^\top; \dots; \mathbf{a}_m^\top]$  where  $\mathbf{a}_1, \mathbf{a}_2, \dots$  are the rows of  $\mathbf{A}$ , we can again construct a row permutation matrix  $\mathbf{P}$  such that the independent rows of  $\mathbf{A}$  appear in the first  $r$  rows of  $\mathbf{PA}$ . Without loss of generality, we assume  $\mathbf{b}_1^\top, \mathbf{b}_2^\top, \dots, \mathbf{b}_r^\top$  are the  $r$  linearly independent rows of  $\mathbf{A}$  and

$$\mathbf{PA} = \begin{bmatrix} \mathbf{b}_1^\top \\ \mathbf{b}_2^\top \\ \vdots \\ \mathbf{b}_r^\top \\ \mathbf{b}_{r+1}^\top \\ \vdots \\ \mathbf{b}_m^\top \end{bmatrix}.$$

Let  $\mathbf{Z} = [\mathbf{b}_1^\top; \mathbf{b}_2^\top; \dots; \mathbf{b}_r^\top] \in \mathbb{R}^{r \times n}$ . Since any  $\mathbf{b}_i$  is in the row space of  $\mathbf{Z}$ , we can find a  $\mathbf{E} \in \mathbb{R}^{(m-r) \times r}$  such that

$$\begin{bmatrix} \mathbf{b}_{r+1}^\top \\ \mathbf{b}_{r+2}^\top \\ \vdots \\ \mathbf{b}_m^\top \end{bmatrix} = \mathbf{EZ}.$$

That is,

$$\mathbf{PA} = \begin{bmatrix} \mathbf{b}_1^\top \\ \mathbf{b}_2^\top \\ \vdots \\ \mathbf{b}_r^\top \\ \mathbf{b}_{r+1}^\top \\ \vdots \\ \mathbf{b}_m^\top \end{bmatrix} = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{E} \end{bmatrix} \mathbf{Z}$$

where  $\mathbf{I}_r$  is an  $r \times r$  identity matrix. Moreover,  $\mathbf{Z} \in \mathbb{R}^{r \times n}$  has full row rank such that its full LQ decomposition is given by  $\mathbf{Z} = [\mathbf{L}, \mathbf{0}] \mathbf{V}$ , where  $\mathbf{L} \in \mathbb{R}^{r \times r}$  is a lower triangular matrix with full rank and  $\mathbf{V}$  is an orthogonal matrix. This implies

$$\mathbf{PA} = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{E} \end{bmatrix} \mathbf{Z} = \begin{bmatrix} \mathbf{I}_r \\ \mathbf{E} \end{bmatrix} [\mathbf{L} \quad \mathbf{0}] \mathbf{V} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{EL} & \mathbf{0} \end{bmatrix} \mathbf{V}. \quad (4.2)$$

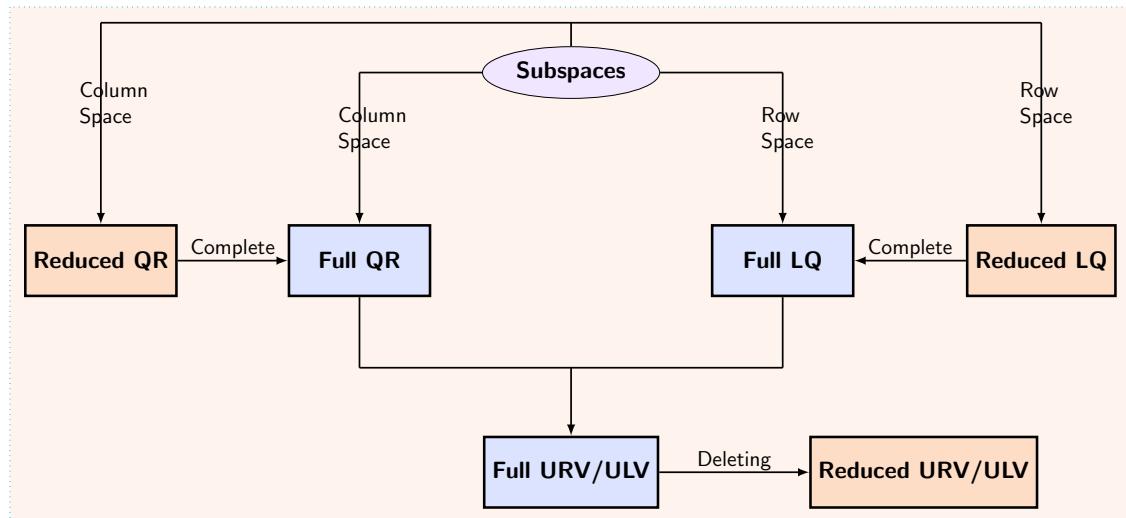
Since  $\mathbf{L}$  has full rank, this means  $\begin{bmatrix} \mathbf{L} \\ \mathbf{EL} \end{bmatrix}$  also has full rank such that its full QR decomposition is given by  $\mathbf{U}_0 \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$  where  $\mathbf{R} \in \mathbb{R}^{r \times r}$  is upper triangular and  $\mathbf{U}_0$  is an orthogonal matrix.

Substitute into Equation (4.2), we have

$$\mathbf{A} = \mathbf{P}^{-1} \mathbf{U}_0 \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}.$$

Let  $\mathbf{U} = \mathbf{P}^{-1} \mathbf{U}_0$ , which is a product of two orthogonal matrices, and is also an orthogonal matrix, from which the result follows.  $\blacksquare$

Again, there is a version of reduced URV decomposition and the difference between the full and reduced URV can be implied from the context as shown in Figure 4.1. The ULV and URV sometimes are referred to as the UTV decomposition framework (Fierro and Hansen, 1997; Golub and Van Loan, 2013).



**Figure 4.2:** Derive the ULV/URV by the QR and LR.

The relationship between the QR and ULV/URV is depicted in Figure 4.2. Furthermore, the ULV and URV decomposition can be utilized to prove an important property in linear algebra that we shall shortly see in the sequel: the row rank and column rank of any matrix are the same. Whereas, an elementary construction to prove this claim is provided in Appendix A (p. 426).

We will shortly see that the forms of ULV and URV are very close to the singular value decomposition (SVD). All of the three factor the matrix  $\mathbf{A}$  into two orthogonal matrices. Specially, there exists a set of basis for the four subspaces of  $\mathbf{A}$  in the fundamental theorem of linear algebra via the ULV and the URV. Taking ULV as an example, the first  $r$  columns of  $\mathbf{U}$  form an orthonormal basis of  $\mathcal{C}(\mathbf{A})$ , and the last  $(m - r)$  columns of  $\mathbf{U}$  form an orthonormal basis of  $\mathcal{N}(\mathbf{A}^\top)$ . The first  $r$  rows of  $\mathbf{V}$  form an orthonormal basis for the row space  $\mathcal{C}(\mathbf{A}^\top)$ , and the last  $(n - r)$  rows form an orthonormal basis for  $\mathcal{N}(\mathbf{A})$  (similar to the

two-sided orthogonal decomposition):

$$\begin{aligned}\mathcal{C}(\mathbf{A}) &= \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}, \\ \mathcal{N}(\mathbf{A}) &= \text{span}\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}, \\ \mathcal{C}(\mathbf{A}^\top) &= \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}, \\ \mathcal{N}(\mathbf{A}^\top) &= \text{span}\{\mathbf{u}_{r+1}, \mathbf{u}_{r+2}, \dots, \mathbf{u}_m\}.\end{aligned}$$

The SVD goes further than there is a connection between the two pairs of orthonormal basis, i.e., transforming from column basis into row basis, or left null space basis into right null space basis. We will get more details in the SVD section.

## 4.2. Complete Orthogonal Decomposition

What is related to the UTV decomposition is called the *complete orthogonal decomposition* which factors into two orthogonal matrices as well.

### Theorem 4.1: (Complete Orthogonal Decomposition)

Every  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$  can be factored as

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{T} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V},$$

where  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  are two orthogonal matrices, and  $\mathbf{T} \in \mathbb{R}^{r \times r}$  is an rank- $r$  matrix.

**Proof** [of Theorem 4.1] By rank-revealing QR decomposition (Theorem 3.1, p. 101),  $\mathbf{A}$  can be factored as

$$\mathbf{Q}_1^\top \mathbf{A} \mathbf{P} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is upper triangular,  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ ,  $\mathbf{Q}_1 \in \mathbb{R}^{m \times m}$  is an orthogonal matrix, and  $\mathbf{P}$  is a permutation matrix.

Then, it is not hard to find a decomposition such that

$$\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix} = \mathbf{Q}_2 \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix}, \quad (4.3)$$

where  $\mathbf{Q}_2$  is an orthogonal matrix,  $\mathbf{S}$  is an rank- $r$  matrix. The decomposition is reasonable in the sense the matrix  $\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix} \in \mathbb{R}^{n \times r}$  has rank  $r$  of which the columns stay in a subspace of  $\mathbb{R}^n$ . Nevertheless, the columns of  $\mathbf{Q}_2$  span the whole space of  $\mathbb{R}^n$ , where

we can assume the first  $r$  columns of  $\mathbf{Q}_2$  span the same space as that of  $\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix}$ . The matrix  $\begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix}$  is to transform  $\mathbf{Q}_2$  into  $\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix}$ .

Then, it follows that

$$\mathbf{Q}_1^\top \mathbf{A} \mathbf{P} \mathbf{Q}_2 = \begin{bmatrix} \mathbf{S}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Let  $\mathbf{U} = \mathbf{Q}_1$ ,  $\mathbf{V} = \mathbf{Q}_2^\top \mathbf{P}^\top$  and  $\mathbf{T} = \mathbf{S}^\top$ , we complete the proof.  $\blacksquare$

We can find that when Equation (4.3) is taken to be the reduced QR decomposition of  $\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix}$ , then the complete orthogonal decomposition reduces to the ULV decomposition.

### 4.3. Applications

#### 4.3.1 Application: Least Squares via ULV/URV for Rank Deficient Matrices

In Section 3.20.1 (p. 125), we introduced the LS via the full QR decomposition for full rank matrices. However, it often happens that the matrix may be rank-deficient. If  $\mathbf{A}$  does not have full column rank,  $\mathbf{A}^\top \mathbf{A}$  is not invertible. We can then use the ULV/URV decomposition to find the least squares solution as illustrated in the following theorem.

#### Theorem 4.1: (LS via ULV/URV for Rank Deficient Matrix)

Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r$  and  $m \geq n$ . Suppose  $\mathbf{A} = \mathbf{U} \mathbf{T} \mathbf{V}$  is its full ULV/URV decomposition with  $\mathbf{U} \in \mathbb{R}^{m \times m}$ ,  $\mathbf{V} \in \mathbb{R}^{n \times n}$  being orthogonal matrix matrices, and

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where  $\mathbf{T}_{11} \in \mathbb{R}^{r \times r}$  is a lower triangular matrix or an upper triangular matrix. Suppose  $\mathbf{b} \in \mathbb{R}^m$ , then the LS solution with minimal 2-norm to  $\mathbf{Ax} = \mathbf{b}$  is given by

$$\mathbf{x}_{LS} = \mathbf{V}^\top \begin{bmatrix} \mathbf{T}_{11}^{-1} \mathbf{c} \\ \mathbf{0} \end{bmatrix},$$

where  $\mathbf{c}$  is the first  $r$  components of  $\mathbf{U}^\top \mathbf{b}$ .

**Proof** [of Theorem 4.1] Since  $\mathbf{A} = \mathbf{QR}$  is the full QR decomposition of  $\mathbf{A}$  and  $m > n$ , the last  $m - n$  rows of  $\mathbf{R}$  are zero as shown in Figure 3.6. Then  $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$  is the square upper

triangular in  $\mathbf{R}$  and  $\mathbf{Q}^\top \mathbf{A} = \mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}$ . Thus,

$$\begin{aligned} \|\mathbf{Ax} - \mathbf{b}\|^2 &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) \\ &= (\mathbf{Ax} - \mathbf{b})^\top \mathbf{U} \mathbf{U}^\top (\mathbf{Ax} - \mathbf{b}) \quad (\text{Since } \mathbf{U} \text{ is an orthogonal matrix}) \\ &= \|\mathbf{U}^\top \mathbf{Ax} - \mathbf{U}^\top \mathbf{b}\|^2 \quad (\text{Invariant under orthogonal}) \\ &= \|\mathbf{U}^\top \mathbf{UTVx} - \mathbf{U}^\top \mathbf{b}\|^2 \\ &= \|\mathbf{TVx} - \mathbf{U}^\top \mathbf{b}\|^2 \\ &= \|\mathbf{T}_{11}\mathbf{e} - \mathbf{c}\|^2 + \|\mathbf{d}\|^2, \end{aligned}$$

where  $\mathbf{c}$  is the first  $r$  components of  $\mathbf{U}^\top \mathbf{b}$  and  $\mathbf{d}$  is the last  $m - r$  components of  $\mathbf{U}^\top \mathbf{b}$ ,  $\mathbf{e}$  is the first  $r$  components of  $\mathbf{Vx}$  and  $\mathbf{f}$  is the last  $n - r$  components of  $\mathbf{Vx}$ :

$$\mathbf{U}^\top \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \quad \mathbf{Vx} = \begin{bmatrix} \mathbf{e} \\ \mathbf{f} \end{bmatrix}$$

And the LS solution can be calculated by back/forward substitution of the upper/lower triangular system  $\mathbf{T}_{11}\mathbf{e} = \mathbf{c}$ , i.e.,  $\mathbf{e} = \mathbf{T}_{11}^{-1}\mathbf{c}$ . For  $\mathbf{x}$  to have minimal 2-norm,  $\mathbf{f}$  must be zero. That is

$$\mathbf{x}_{LS} = \mathbf{V}^\top \begin{bmatrix} \mathbf{T}_{11}^{-1}\mathbf{c} \\ \mathbf{0} \end{bmatrix}.$$

This completes the proof. ■

Moreover, we will shortly find that a similar argument can also be made via the singular value decomposition (SVD) in Section 14.7.1 since SVD shares similar form as the ULV/URV, both sandwiched by two orthogonal matrices. And SVD goes further by diagonalizing the matrix. Readers can also derive the LS solution via the rank-revealing QR decomposition.

**A word on the minimal 2-norm LS solution** For the least squares problem, the set of all minimizers

$$\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{Ax} - \mathbf{b}\| = \min\}$$

is convex (Golub and Van Loan, 2013). And if  $\mathbf{x}_1, \mathbf{x}_2 \in \mathcal{X}$  and  $\lambda \in [0, 1]$ , then

$$\|\mathbf{A}(\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2) - \mathbf{b}\| \leq \lambda\|\mathbf{Ax}_1 - \mathbf{b}\| + (1 - \lambda)\|\mathbf{Ax}_2 - \mathbf{b}\| = \min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{b}\|.$$

Thus  $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in \mathcal{X}$ . In above proof, if we do not set  $\mathbf{f} = \mathbf{0}$ , we will find more least squares solutions. However, the minimal 2-norm least squares solution is unique. For full-rank case in the previous section, the least squares solution is unique and it must have a minimal 2-norm. See also (Foster, 2003; Golub and Van Loan, 2013) for a more detailed discussion on this topic.

### 4.3.2 Application: Row Rank equals Column Rank Again via UTV

As mentioned above, the UTV framework can prove the important theorem in linear algebra that the row rank and column rank of a matrix are equal.

Notice that to apply the UTV in the proof, a slight modification on the claim of the existence of the UTV decomposition needs to be taken care of. For example, in Theorem 4.1, the assumption of the matrix  $\mathbf{A}$  is to have rank  $r$ . Since rank  $r$  already admits the fact that row rank equals column rank. A better claim here to this aim is to say matrix  $\mathbf{A}$  has column rank  $r$  in Theorem 4.1. See (Lu, 2021b) for a detailed discussion.

#### Theorem 4.2: (Row Rank Equals Column Rank)

The dimension of the column space of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is equal to the dimension of its row space, i.e., the row rank and the column rank of a matrix  $\mathbf{A}$  are equal.

**Proof [of Theorem 4.2, p. 148, A First Way]** Any  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$  can be factored as

$$\mathbf{A} = \mathbf{U}_0 \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}_0,$$

where  $\mathbf{U}_0 \in \mathbb{R}^{m \times m}$  and  $\mathbf{V}_0 \in \mathbb{R}^{n \times n}$  are two orthogonal matrices, and  $\mathbf{L} \in \mathbb{R}^{r \times r}$  is a lower triangular matrix <sup>2</sup>. Let

$$\mathbf{D} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

the row rank and column rank of  $\mathbf{D}$  are apparently the same. If we could prove the column rank of  $\mathbf{A}$  equals the column rank of  $\mathbf{D}$ , and the row rank of  $\mathbf{A}$  equals the row rank of  $\mathbf{D}$ , then we complete the proof.

Let  $\mathbf{U} = \mathbf{U}_0^\top$ ,  $\mathbf{V} = \mathbf{V}_0^\top$ , then  $\mathbf{D} = \mathbf{UAV}$ . Decompose the above idea into two steps, a moment of reflexion reveals that, if we could first prove the row rank and column rank of  $\mathbf{A}$  are equal to those of  $\mathbf{UA}$ , and then, if we further prove the row rank and column rank of  $\mathbf{UA}$  are equal to those of  $\mathbf{UAV}$ , we could also complete the proof.

**Row rank and column rank of  $\mathbf{A}$  are equal to those of  $\mathbf{UA}$**  Let  $\mathbf{B} = \mathbf{UA}$ , and let further  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  and  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$  be the column partitions of  $\mathbf{A}$  and  $\mathbf{B}$  respectively. Therefore,  $[\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n] = [\mathbf{U}\mathbf{a}_1, \mathbf{U}\mathbf{a}_2, \dots, \mathbf{U}\mathbf{a}_n]$ . If  $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \dots + x_n\mathbf{a}_n = 0$ , then we also have

$$\mathbf{U}(x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \dots + x_n\mathbf{a}_n) = x_1\mathbf{b}_1 + x_2\mathbf{b}_2 + \dots + x_n\mathbf{b}_n = 0.$$

Let  $j_1, j_2, \dots, j_r$  be distinct indices between 1 and  $n$ , if the set  $\{\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}\}$  is independent, the set  $\{\mathbf{b}_{j_1}, \mathbf{b}_{j_2}, \dots, \mathbf{b}_{j_r}\}$  must also be linearly independent. This implies

$$\dim(\mathcal{C}(\mathbf{B})) \leq \dim(\mathcal{C}(\mathbf{A})).$$

<sup>2</sup>. Instead of using the ULV decomposition, in some texts, the authors use elementary transformations  $\mathbf{E}_1, \mathbf{E}_2$  such that  $\mathbf{A} = \mathbf{E}_1 \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{E}_2$ , to prove the result.

Similarly, by  $\mathbf{A} = \mathbf{U}^\top \mathbf{B}$ , it follows that

$$\dim(\mathcal{C}(\mathbf{A})) \leq \dim(\mathcal{C}(\mathbf{B})).$$

This implies

$$\dim(\mathcal{C}(\mathbf{B})) = \dim(\mathcal{C}(\mathbf{A})).$$

Apply the process onto  $\mathbf{B}^\top$  and  $\mathbf{A}^\top$ , we have

$$\dim(\mathcal{C}(\mathbf{B}^\top)) = \dim(\mathcal{C}(\mathbf{A}^\top)).$$

This implies the row rank and column rank of  $\mathbf{A}$  and  $\mathbf{B} = \mathbf{U}\mathbf{A}$  are the same.

**Row rank and column rank of  $\mathbf{U}\mathbf{A}$  are equal to those of  $\mathbf{UAV}$**  Similarly, by applying above discussion on  $\mathbf{U}\mathbf{A}$  and  $\mathbf{UAV}$ , we can also show that the row rank and column rank of  $\mathbf{U}\mathbf{A}$  and  $\mathbf{UAV}$  are the same. This completes the proof. ■

## **Part III**

# **Data Interpretation and Information Distillation**



## Introduction

For matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r$ , there exist  $r$  linearly independent columns and rows respectively by the fundamental theorem of linear algebra (Theorem 27.1, p. 428). Then  $\mathbf{A}$  admits

$$(DI1) \quad \underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{C}} \underset{r \times n}{\mathbf{F}},$$

where  $\mathbf{C}$  contains  $r$  linearly independent columns of  $\mathbf{A}$  and  $\mathbf{F}$  is to reconstruct all the columns of  $\mathbf{A}$  since all the columns of  $\mathbf{CF}$  are combinations of the columns of  $\mathbf{C}$ . To see this, suppose  $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_n]$  is the column partition of  $\mathbf{C}$ , we have

$$\mathbf{A} = \mathbf{CF} = [\mathbf{C}\mathbf{f}_1, \mathbf{C}\mathbf{f}_2, \dots, \mathbf{C}\mathbf{f}_n],$$

where each column  $\mathbf{C}\mathbf{f}_i$  is a combination of the columns of  $\mathbf{C}$ . And the columns of  $\mathbf{C}$  are known as the *spanning columns* of  $\mathbf{A}$ . Or  $\mathbf{A}$  admits

$$(DI2) \quad \underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{D}} \underset{r \times n}{\mathbf{R}},$$

where  $\mathbf{R}$  contains  $r$  linearly independent rows of  $\mathbf{A}$  and  $\mathbf{D}$  is to reconstruct all the rows of  $\mathbf{A}$  since all the rows of  $\mathbf{DR}$  are combinations of the rows of  $\mathbf{R}$ . To see this, suppose  $\mathbf{D} = [\mathbf{d}_1^\top; \mathbf{d}_2^\top; \dots; \mathbf{d}_m^\top]$  is the row partition of  $\mathbf{D}$ , we have

$$\mathbf{A} = \mathbf{DR} = \begin{bmatrix} \mathbf{d}_1^\top \mathbf{R} \\ \mathbf{d}_2^\top \mathbf{R} \\ \vdots \\ \mathbf{d}_m^\top \mathbf{R} \end{bmatrix},$$

where each row  $\mathbf{d}_i^\top \mathbf{R}$  is a combination of the rows of  $\mathbf{R}$ . And the rows of  $\mathbf{R}$  are known as the *spanning rows* of  $\mathbf{A}$ .

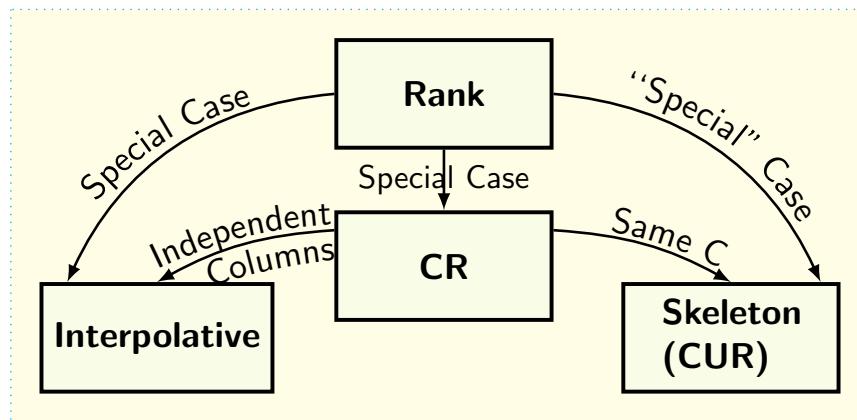
This factorization in (DI1) is similar to the QR decomposition where the first factor spans the column space of  $\mathbf{A}$  (orthogonal in the latter), whilst (DI2) is similar to the LQ decomposition. However (DI1) has several advantages, as compared to, e.g., the QR:

- It is sometimes advantageous to work with a basis that consists of a subset of the columns of  $\mathbf{A}$  itself. In this case, one typically has to give up on the requirement that the basis vectors are orthonormal;
- If  $\mathbf{A}$  is sparse and nonnegative, then  $\mathbf{C}$  shares these properties;
- The (DI1) requires less memory to store and is more efficient to calculate than the QR in general. One can see from the above representation that only  $r(m+n)$  entries have to be stored instead of  $mn$  entries of the original matrix  $\mathbf{A}$  or  $mn + \frac{(n+1)n}{2}$  entries in the *reduced* QR decomposition (Figure 3.6, p. 100);
- Finding the indices associated with the spanning columns is often helpful for the purpose of data interpretation and analysis, it can be very useful to identify a subset of the columns that distills the information in the matrix;
- Finding the least squares solution  $\mathbf{Ax} = \mathbf{b}$  can be done by calculating the least squares solution of  $\mathbf{Cx} = \mathbf{b}$  where the former one is to project  $\mathbf{b}$  into the column space of  $\mathbf{A}$  and the latter one is to project it into the column space of  $\mathbf{C}$ . This factorization

(DI1) can be regarded as the first phase of the variable selection in least squares or linear models in general. A short review of the variable selection procedure is given in Section 3.20.3 (p. 131) as an application of QR decomposition, and more details can refer to (Lu, 2021d);

- The (DI1) often preserves “the physics” of a problem in a way that the QR does not.

For the rest of this part, we will discuss several variations of the factorization above with different focuses, e.g., in terms of description, easy to interpret or not, or well-condition, and how they select the basis vectors for the column space, for the row space, or for both. The big picture can be shown in Figure 4.3.



**Figure 4.3:** Data interpretation relationship. See also where it’s lying in the matrix decomposition world map Figure 1.

### A Brief Introduction for SVD

For the analysis of the decomposition in this part, especially the interpolative decomposition, we need the SVD and Eckart-Young-Misly Theorem that we shall briefly introduce here. More details are delayed in Section 14 (p. 264). For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r$ , it admits

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top,$$

where  $\mathbf{u}_i$  and  $\mathbf{v}_i$  are left and right singular vectors respectively. The matrix  $\Sigma \in \mathbb{R}^{m \times n}$  is a rectangular diagonal matrix whose entries are singular values in descending order,  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = 0$ , along the main diagonal with only  $r$  nonzero singular values.

Given further  $1 \leq k \leq r$ , and let  $\mathbf{A}_k$  be the *truncated SVD (TSVD)* of  $\mathbf{A}$  with the largest  $k$  terms, i.e.,  $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  from SVD of  $\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  by zeroing out the  $r-k$  trailing singular values of  $\mathbf{A}$ . Then  $\mathbf{A}_k$  is the best rank- $k$  approximation to  $\mathbf{A}$  in terms of the spectral norm. That is, for any rank  $r$  matrix  $\mathbf{B}$ , we have  $\|\mathbf{A} - \mathbf{A}_k\|_2 \leq \|\mathbf{A} - \mathbf{B}\|_2$  or  $\|\mathbf{A} - \mathbf{A}_k\|_F \leq \|\mathbf{A} - \mathbf{B}\|_F$  which are measured by Frobenius norm and spectral norm

respectively. And the distance between  $\mathbf{A}, \mathbf{A}_k$  is given by

$$\|\mathbf{A} - \mathbf{A}_k\|_F = \sqrt{\sigma_{k+1}^2 + \dots + \sigma_r^2},$$

or

$$\|\mathbf{A} - \mathbf{A}_k\|_2 = \sigma_r.$$

### Subset Selection Problem

Subset selection is a method for selecting a subset of columns from a real matrix so that the subset represents the entire matrix well and is far from being rank deficient. (DI1) above is just an example of column selection with full rank  $r$  columns. However, given integer  $k < r$ , we always attempt to find  $k$  linearly independent columns that best represent the information in the matrix. The mathematical formulation of the subset selection problems is: Determine a permutation matrix  $\mathbf{P}$  such that

$$\mathbf{AP} = [\mathbf{A}_1, \mathbf{A}_2],$$

where  $\mathbf{A}_1 \in \mathbb{R}^{m \times k}$  contains  $k$  linearly independent columns of  $\mathbf{A}$  such that

1. The smallest singular value is as large as possible (similar to SVD!). That is, there exists a value  $\eta$  such that the  $k$ -th singular value of  $\mathbf{A}_1$  is bounded by that of  $\mathbf{A}$ :

$$\frac{\sigma_k(\mathbf{A})}{\eta} \leq \sigma_k(\mathbf{A}_1) \leq \sigma_k(\mathbf{A}).$$

2. The rest  $n - k$  redundant columns of  $\mathbf{A}_2$  are well represented by  $k$  columns of  $\mathbf{A}_1$ . That is

$$\min_{\mathbf{W} \in \mathbb{R}^{k \times (n-k)}} \|\mathbf{A}_1 \mathbf{W} - \mathbf{A}_2\|_2$$

is small, i.e., there exists a value  $\eta$  such that the distance is bounded by  $(k + 1)$ -th singular value of  $\mathbf{A}$ :

$$\sigma_{k+1}(\mathbf{A}) \leq \min_{\mathbf{W} \in \mathbb{R}^{k \times (n-k)}} \|\mathbf{A}_1 \mathbf{W} - \mathbf{A}_2\|_2 \leq \eta \sigma_{k+1}(\mathbf{A}).$$

# Chapter 5

# CR Decomposition

## Contents

---

|     |                                                               |     |
|-----|---------------------------------------------------------------|-----|
| 5.1 | CR Decomposition . . . . .                                    | 156 |
| 5.2 | Existence of the CR Decomposition . . . . .                   | 156 |
| 5.3 | Reduced Row Echelon Form (RREF) . . . . .                     | 157 |
| 5.4 | Computing the CR Decomposition via the Gaussian Elimination   | 159 |
| 5.5 | Rank Decomposition . . . . .                                  | 164 |
| 5.6 | Application: Rank and Trace of an Idempotent Matrix . . . . . | 165 |
| 5.7 | Other Applications . . . . .                                  | 166 |

---

## 5.1. CR Decomposition

The CR decomposition is proposed in (Strang, 2021; Strang and Moler, 2021). As usual, we firstly give the result and we will discuss the existence and the origin of this decomposition in the following sections.

### Theorem 5.1: (CR Decomposition)

Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be factored as

$$\mathbf{A}_{m \times n} = \mathbf{C}_{m \times r} \quad \mathbf{R}_{r \times n}$$

where  $\mathbf{C}$  is the first  $r$  linearly independent columns of  $\mathbf{A}$ , and  $\mathbf{R}$  is an  $r \times n$  matrix to reconstruct the columns of  $\mathbf{A}$  from columns of  $\mathbf{C}$ . In particular,  $\mathbf{R}$  is the row reduced echelon form (RREF) of  $\mathbf{A}$  without the zero rows.

The storage for the decomposition is then reduced or potentially increased from  $mn$  to  $r(m + n)$ .

## 5.2. Existence of the CR Decomposition

Since matrix  $\mathbf{A}$  is of rank  $r$ , there are some  $r$  linearly independent columns in  $\mathbf{A}$ . We then choose linearly independent columns from  $\mathbf{A}$  and put them into  $\mathbf{C}$ :

Find  $r$  linearly Independent Columns From  $\mathbf{A}$

1. If column 1 of  $\mathbf{A}$  is not zero, put it into the column of  $\mathbf{C}$ ;
2. If column 2 of  $\mathbf{A}$  is not a multiple of column 1, put it into the column of  $\mathbf{C}$ ;
3. If column 3 of  $\mathbf{A}$  is not a combination of columns 1 and 2, put it into the column of  $\mathbf{C}$ ;
4. Continue this process until we find  $r$  linearly independent columns (or all the linearly independent columns if we do not know the rank  $r$  beforehand).

When we have the  $r$  linearly independent columns from  $\mathbf{A}$ , we can prove the existence of CR decomposition by the column space view of matrix multiplication.

**Column space view of matrix multiplication** A multiplication of two matrices  $\mathbf{D} \in \mathbb{R}^{m \times k}, \mathbf{E} \in \mathbb{R}^{k \times n}$  is  $\mathbf{A} = \mathbf{DE} = \mathbf{D}[\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n] = [\mathbf{De}_1, \mathbf{De}_2, \dots, \mathbf{De}_n]$ , i.e., each column of  $\mathbf{A}$  is a combination of columns from  $\mathbf{D}$ .

**Proof** [of Theorem 5.1] As the rank of matrix  $\mathbf{A}$  is  $r$  and  $\mathbf{C}$  contains  $r$  linearly independent columns from  $\mathbf{A}$ , the column space of  $\mathbf{C}$  is equivalent to the column space of  $\mathbf{A}$ . If we take any other column  $\mathbf{a}_i$  of  $\mathbf{A}$ ,  $\mathbf{a}_i$  can be represented as a linear combination of the columns of  $\mathbf{C}$ , i.e., there exists a vector  $\mathbf{r}_i$  such that  $\mathbf{a}_i = \mathbf{Cr}_i, \forall i \in \{1, 2, \dots, n\}$ . Put these  $\mathbf{r}_i$ 's into the columns of matrix  $\mathbf{R}$ , we obtain

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] = [\mathbf{Cr}_1, \mathbf{Cr}_2, \dots, \mathbf{Cr}_n] = \mathbf{CR},$$

from which the result follows. ■

### 5.3. Reduced Row Echelon Form (RREF)

In Gaussian elimination Section 1.2, we introduced the elimination matrix (a lower triangular matrix) and permutation matrix to transform  $\mathbf{A}$  into an upper triangular form. We rewrite the Gaussian elimination for a  $4 \times 4$  square matrix, where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed:

| Gaussian Elimination for a Square Matrix                                                                                                                                                                                         |                              |                                                                                                                                                                                                                                     |                              |                                                                                                                                                                                                                                                        |                              |                                                                                                                                                                                                                               |                                                  |  |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|--|--|
| $\begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$ | $\xrightarrow{\mathbf{E}_1}$ | $\begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$ | $\xrightarrow{\mathbf{P}_1}$ | $\begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$ | $\xrightarrow{\mathbf{E}_2}$ | $\begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{\boxtimes} & \boxtimes \\ 0 & \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} \end{bmatrix}$ | $\mathbf{E}_2\mathbf{P}_1\mathbf{E}_1\mathbf{A}$ |  |  |
| $\mathbf{A}$                                                                                                                                                                                                                     |                              | $\mathbf{E}_1\mathbf{A}$                                                                                                                                                                                                            |                              | $\mathbf{P}_1\mathbf{E}_1\mathbf{A}$                                                                                                                                                                                                                   |                              |                                                                                                                                                                                                                               |                                                  |  |  |

Furthermore, the Gaussian elimination can also be applied on a rectangular matrix, we give an example for a  $4 \times 5$  matrix as follows:

| Gaussian Elimination for a Rectangular Matrix                                                                                                                                                                                                             |                              |                                                                                                                                                                                                                                                                    |                              |                                                                                                                                                                     |  |  |                                      |  |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--------------------------------------|--|--|
| $\begin{bmatrix} 2 & \boxtimes & 10 & 9 & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$ | $\xrightarrow{\mathbf{E}_1}$ | $\begin{bmatrix} 2 & \boxtimes & 10 & 9 & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{5} & \mathbf{6} & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{2} & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$ | $\xrightarrow{\mathbf{E}_2}$ | $\begin{bmatrix} 2 & \boxtimes & 10 & 9 & \boxtimes \\ 0 & 0 & \mathbf{5} & 6 & \boxtimes \\ 0 & 0 & 0 & \mathbf{3} & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ |  |  | $\mathbf{E}_2\mathbf{E}_1\mathbf{A}$ |  |  |
| $\mathbf{A}$                                                                                                                                                                                                                                              |                              | $\mathbf{E}_1\mathbf{A}$                                                                                                                                                                                                                                           |                              |                                                                                                                                                                     |  |  |                                      |  |  |

where the blue-colored numbers are **pivots** as we defined previously and we call the last matrix above **row echelon form**. Note that we get the 4-th row as a zero row in this specific example. Going further, if we subtract each row by a multiple of the next row to make the entries above the pivots to be zero:

| Reduced Row Echelon Form: Get Zero Above Pivots                                                                                                                     |                              |                                                                                                                                                                              |                              |                                                                                                                                                                    |  |  |                                                              |  |  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|--|--------------------------------------------------------------|--|--|
| $\begin{bmatrix} 2 & \boxtimes & 10 & 9 & \boxtimes \\ 0 & 0 & \mathbf{5} & 6 & \boxtimes \\ 0 & 0 & 0 & \mathbf{3} & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ | $\xrightarrow{\mathbf{E}_3}$ | $\begin{bmatrix} 2 & \boxtimes & \mathbf{0} & -3 & \boxtimes \\ 0 & 0 & \mathbf{5} & 6 & \boxtimes \\ 0 & 0 & 0 & \mathbf{3} & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ | $\xrightarrow{\mathbf{E}_4}$ | $\begin{bmatrix} 2 & \boxtimes & 0 & 0 & \boxtimes \\ 0 & 0 & \mathbf{5} & 0 & \boxtimes \\ 0 & 0 & 0 & \mathbf{3} & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$ |  |  | $\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A}$ |  |  |
| $\mathbf{E}_2\mathbf{E}_1\mathbf{A}$                                                                                                                                |                              | $\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A}$                                                                                                                             |                              |                                                                                                                                                                    |  |  |                                                              |  |  |

where  $\mathbf{E}_3$  subtracts 2 times the 2-nd row from the 1-st row, and  $\mathbf{E}_4$  adds the 3-rd row to the 1-st row and subtracts 2 times the 3-rd row from the 2-nd row. Finally, we get the full row reduced echelon form by making the pivots to be 1:

Reduced Row Echelon Form: Make The Pivots To Be 1

$$\begin{array}{c} \left[ \begin{array}{ccccc} 2 & \boxtimes & 0 & 0 & \boxtimes \\ 0 & 0 & 5 & 0 & \boxtimes \\ 0 & 0 & 0 & 3 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{\mathbf{E}_5} \left[ \begin{array}{ccccc} 1 & \boxtimes & 0 & 0 & \boxtimes \\ 0 & 0 & 1 & 0 & \boxtimes \\ 0 & 0 & 0 & 1 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right], \\ \mathbf{E}_4 \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} \qquad \qquad \qquad \mathbf{E}_5 \mathbf{E}_4 \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} \end{array}$$

where  $\mathbf{E}_5$  makes the pivots to be 1. Note here, the transformation matrix  $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_5$  are not necessarily to be lower triangular matrices as they are in LU decomposition. They can also be permutation matrices or other matrices. We call this final matrix the **reduced row echelon form** of  $\mathbf{A}$  where it has 1's as pivots and zeros above the pivots.

### Lemma 5.1: (Rank and Pivots)

The rank of  $\mathbf{A}$  is equal to the number of pivots.

### Lemma 5.2: (RREF in CR)

The reduced row echelon form of the matrix  $\mathbf{A}$  without zero rows is the matrix  $\mathbf{R}$  in the CR decomposition.

**Proof** [Informal Proof of Lemma 5.2 and Lemma 5.1] Following the steps in Gaussian elimination, the number of pivot elements indicates the number of linearly independent columns in matrix  $\mathbf{A}$ , which is on the other hand, exactly the rank of the matrix.

Following the example above, we have

$$\mathbf{E}_5 \mathbf{E}_4 \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} = \mathbf{R} \quad \longrightarrow \quad \mathbf{A} = (\mathbf{E}_5 \mathbf{E}_4 \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1)^{-1} \mathbf{R}.$$

We notice that the columns 1, 3, 4 of  $\mathbf{R}$  has only one element 1 which means we could construct a matrix  $\mathbf{C}$  (exactly the same column matrix in CR decomposition) whose first 3 columns to be equal to columns 1, 3, 4 of matrix  $\mathbf{A}$ , i.e.,  $\mathbf{C} = [\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_4]$ . Furthermore, since the last row of  $\mathbf{R}$  is all zero, the column 4 of  $\mathbf{C}$  does not account for any computation we can just ignore column 4 of  $\mathbf{C}$  and row 4 of  $\mathbf{R}$ . And this  $\mathbf{C}$  is the only matrix that can reconstruct the columns 1, 3, 4 of  $\mathbf{A}$  as the pivots of  $\mathbf{R}$  are all 1. We obtain

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

This completes the proof. ■

In short, we first compute the reduced row echelon form of matrix  $\mathbf{A}$  by  $rref(\mathbf{A})$ , Then  $\mathbf{C}$  is obtained by removing from  $\mathbf{A}$  all the non-pivot columns (which can be determined by looking for columns in  $rref(\mathbf{A})$  which do not contain a pivot). And  $\mathbf{R}$  is obtained by eliminating zero rows of  $rref(\mathbf{A})$ . And this is actually a special case of **rank decomposition** of matrix  $\mathbf{A}$ . However, CR decomposition is so special that it involves the reduced row echelon form so that we introduce it here particularly.

$\mathbf{R}$  has a remarkable form whose  $r$  columns containing the pivots form an  $r \times r$  identity matrix. Note again that we can just remove the zero rows from the row reduced echelon form to obtain this matrix  $\mathbf{R}$ . In (Strang, 2021), the authors give a specific notation for the row reduced echelon form without removing the zero rows as  $\mathbf{R}_0$ :

$$\mathbf{R}_0 = \text{rref}(\mathbf{A}) = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_r & \mathbf{F} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P},^{\text{1}}$$

where the  $n \times n$  permutation matrix  $\mathbf{P}$  puts the columns of  $r \times r$  identity matrix  $\mathbf{I}_r$  into the correct positions, matching the first  $r$  linearly independent columns of the original matrix  $\mathbf{A}$ .

Previously we proved the important theorem in linear algebra that the row rank equals the column rank of any matrix by the UTV framework (Theorem 4.2, p. 148). The CR decomposition also reveals the theorem.

**Proof [of Theorem 4.2, p. 148, A Second Way]** For CR decomposition of matrix  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , we have  $\mathbf{R} = [\mathbf{I}_r, \mathbf{F}]\mathbf{P}$ , where  $\mathbf{P}$  is an  $n \times n$  permutation to put the columns of the  $r \times r$  identity matrix  $\mathbf{I}_r$  into the correct positions as shown above. It can be easily verified that the  $r$  rows of  $\mathbf{R}$  are linearly independent of the submatrix of  $\mathbf{I}_r$  (since  $\mathbf{I}_r$  is nonsingular) such that the row rank of  $\mathbf{R}$  is  $r$ .

Firstly, from the definition of the CR decomposition, the  $r$  columns of  $\mathbf{C}$  are from  $r$  linearly independent columns of  $\mathbf{A}$ , the column rank of  $\mathbf{A}$  is  $r$ . Further,

- Since  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , all rows of  $\mathbf{A}$  are combinations of the rows of  $\mathbf{R}$ . That is, the row rank of  $\mathbf{A}$  is no larger than the row rank of  $\mathbf{R}$ ;
  - From  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , we also have  $(\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{C}\mathbf{R} = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{A}$ , that is  $\mathbf{R} = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{A}$ .  $\mathbf{C}^\top \mathbf{C}$  is nonsingular since it has full column rank  $r$ . Then all rows of  $\mathbf{R}$  are also combinations of the rows of  $\mathbf{A}$ . That is, the row rank of  $\mathbf{R}$  is no larger than the row rank of  $\mathbf{A}$ ;
  - By “sandwiching”, the row rank of  $\mathbf{A}$  is equal to the row rank of  $\mathbf{R}$  which is  $r$ .

Therefore, both the row rank and column rank of  $\mathbf{A}$  are equal to  $r$  from which the result follows. ■

In the proof above, we use CR decomposition to show that the row rank of a matrix is equal to its column rank. An elementary proof without using CR decomposition or Gaussian elimination is provided in Appendix A (p. 426). Moreover, we also discuss the special form of pseudo-inverse from CR decomposition in Appendix E (p. 445).

## 5.4. Computing the CR Decomposition via the Gaussian Elimination

The central step to compute CR decomposition is to find the row reduced echelon form of the matrix  $\mathbf{A}$ . Suppose  $\mathbf{A}$  is of size  $m \times n$ :

- Get row echelon form (REF):
  - A. Use row 1 of  $\mathbf{A}$  to make the values below  $\mathbf{A}_{11}$  to be 0 in the first column (permutation involved if  $\mathbf{A}_{11} = 0$ ), and put the result into  $\mathbf{R}$ ,  $(2(m-1)n + (m-1)$  flops);

---

1. Permutation matrix  $\mathbf{P}$  in the right side of a matrix is to permute the column of that matrix.

- B. Use row 2 of the result matrix  $\mathbf{R}$  to make the values under  $\mathbf{R}_{22}$  to be 0 (permutation involved if  $\mathbf{R}_{22} = 0$ ),  $(2(m - 2)(n - 1) + (m - 2)$  flops);  
C. Continue this process until we get the row echelon form.  
• Get row reduced echelon form (RREF):  
D. Use the last row to make the values above this last pivot to be zero (ignore if the row is all zero);  
E. Use the penultimate row to make the values above this second last pivot to be zero (ignore if the row is all zero);  
F. Continue this process until we get the row reduced echelon form and divide each row to make the pivots to be 1. Note that there are  $m - 1$  such steps if  $m \leq n$  and  $n - 1$  such steps if  $m > n$ ;

This process is formulated in Algorithm 25.

---

**Algorithm 25** CR Decomposition via the Gaussian Elimination

---

**Require:** rank- $r$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  with size  $m \times n$ ;

- 1: Initially set  $\mathbf{R} = \mathbf{A}$ ;
  - 2: Set  $\mathbf{b}_1 = \mathbf{a}_1$ , (Suppose the first column is nonzero);  $\triangleright 0$  flops
  - 3: **for**  $k = 1$  to  $m - 1$  **do**
  - 4:     Use row  $k$  of  $\mathbf{R}$  to make the values under  $\mathbf{R}_{kk}$  to be 0 (permutation involved if  $\mathbf{R}_{kk} = 0$ );
  - 5: **end for**
  - 6: **for** ( $k = m$  to 2 if  $m \leq n$ ), or ( $k = n$  to 2 if  $m > n$ ) **do**
  - 7:     Use row  $k$  to make the values above this last pivots to be zero (ignore if the row is all zero);
  - 8: **end for**
  - 9: Make the pivot to be 1;
  - 10: Select the rows from  $\mathbf{A}$  corresponding to the rows of  $\mathbf{U}$  into  $\mathbf{R}$ .
- 

**Theorem 5.1: (Algorithm Complexity: CR via the Gaussian Elimination)**

Algorithm 25 requires

$$\text{cost} = \begin{cases} \sim 2m^2n - m^3 \text{ flops,} & \text{if } m \leq n; \\ \sim mn^2 \text{ flops,} & \text{if } m > n. \end{cases}$$

to compute the CR decomposition of an  $m \times n$  matrix.

**Proof** [of Theorem 5.1] Procedure A in the above discussion needs  $m - 1$  divisions to get the multipliers and it takes  $m - 1$  times (rows)  $n$  multiplications with the multipliers. To make the values under  $\mathbf{A}_{11}$  to be zero, it involves  $m - 1$  times  $n$  subtractions. As a result, procedure A costs  $2(m - 1)n + (m - 1)$  flops.

Procedure B needs  $m - 2$  divisions to get the multipliers and it takes  $m - 2$  times (rows)  $n - 1$  multiplications with the multipliers. To make the values under  $\mathbf{R}_{22}$  to

be zero, it involves  $m - 2$  times  $n - 1$  subtractions. As a result, procedure B costs  $\boxed{2(m-2)(n-1) + (m-2)}$  flops.

The procedure can go on, and we can thus summarize the costs for each loop of step 4 in the following table:

| $k$      | Get multipliers        | Multipliers multiply each row | Rows subtraction |
|----------|------------------------|-------------------------------|------------------|
| 1        | $2 : m = m - 1$ rows   | $(m-1)(n)$                    | $(m-1)(n)$       |
| 2        | $3 : m = m - 2$ rows   | $(m-2)(n-1)$                  | $(m-2)(n-1)$     |
| 3        | $4 : m = m - 3$ rows   | $(m-3)(n-2)$                  | $(m-3)(n-2)$     |
| $\vdots$ | $\vdots$               | $\vdots$                      | $\vdots$         |
| $k$      | $k+1 : m = m - k$ rows | $(m-k)(n-k+1)$                | $(m-k)(n-k+1)$   |
| $\vdots$ | $\vdots$               | $\vdots$                      | $\vdots$         |
| $m-1$    | $m : m = 1$ row        | $(1)(n-m+2)$                  | $(1)(n-m+2)$     |

We notice that the  $n-m+2$  in the last row of the above table may not be positive, so that we separate it into two cases to discuss the complexity of computing the row echelon form (REF).

**Get REF, case 1:**  $m \leq n$ , the procedure can go on until the loop  $m-1$ . Thus in step 4 to get the row echelon form, we need

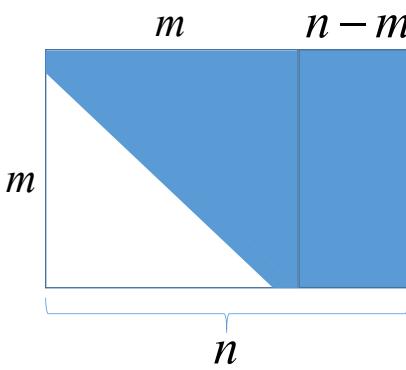
$$\sum_{i=1}^{m-1} [2(m-i)(n-i+1) + (m-i)] = \sum_{i=1}^{m-1} [2i^2 - (2m+2n+3)i + (2mn+3m)],$$

or  $\boxed{m^2n - \frac{1}{3}m^3}$  flops if keep only the leading terms.

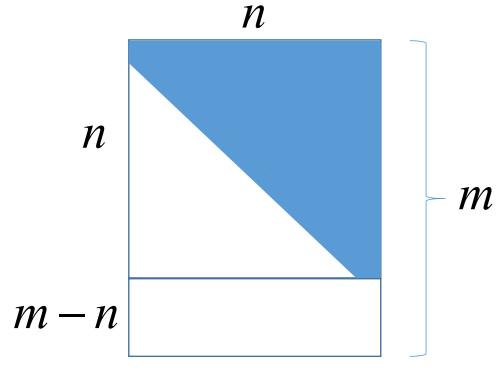
**Get REF, case 2:**  $m > n$ , the procedure should stop at loop  $n$ , otherwise,  $n-k+1$  will be zero if  $k=n+1$ . Thus, in step 4 to get the row echelon form, we need

$$\sum_{i=1}^n [2(m-i)(n-i+1) + (m-i)] = \sum_{i=1}^n [2i^2 - (2m+2n+3)i + (2mn+3m)],$$

or  $\boxed{mn^2 - \frac{1}{3}n^3}$  flops if keep only the leading terms.



(a) RREF for  $m \leq n$



(b) RREF for  $m > n$

**Figure 5.1:** Get RREF from row echelon form.

To get row reduced echelon form (RREF) from row echelon form (REF), we can also separate it into two cases,  $m \leq n$  and  $m > n$  as shown in Figure 5.1 where blank entries indicate zeros, blue entries indicate values that are not necessarily zero, pivots are on the diagonal in the “worst” case.

### Get RREF, case 1: $m \leq n$

Procedure D needs  $m-1$  divisions to get the multipliers and it takes  $m-1$  times  $n-m+1$  multiplications with the multipliers. Then to get the zeros above the pivots, it involves  $m-1$  times  $n-m+1$  subtractions. As a results, procedure D costs  $2(m-1)(n-m+1) + m-1$  flops.

Procedure E needs  $m-2$  divisions to get the multipliers and it takes  $m-2$  times  $n-m+2$  multiplications with the multipliers. Then to get the zeros above the pivots, it involves  $m-2$  times  $n-m+2$  subtractions. As a results, procedure E costs  $2(m-2)(n-m+2) + m-2$  flops.

The procedure can go on, and we can thus again summarize the cost for each loop in the following table:

| $k$      | Get multipliers      | Multipliers multiply each row | Rows subtraction |
|----------|----------------------|-------------------------------|------------------|
| $m$      | $1 : m-1 = m-1$ rows | $(m-1)(n-m+1)$                | $(m-1)(n-m+1)$   |
| $m-1$    | $1 : m-2 = m-2$ rows | $(m-2)(n-m+2)$                | $(m-2)(n-m+2)$   |
| $m-2$    | $1 : m-3 = m-3$ rows | $(m-3)(n-m+3)$                | $(m-3)(n-m+3)$   |
| $\vdots$ | $\vdots$             | $\vdots$                      | $\vdots$         |
| $k$      | $1 : k-1 = k-1$ rows | $(k-1)(n-k+1)$                | $(k-1)(n-k+1)$   |
| $\vdots$ | $\vdots$             | $\vdots$                      | $\vdots$         |
| 2        | $1 : 1 = 1$ row      | $(1)(n-2+1)$                  | $(1)(n-2+1)$     |

In the second loop to get the row reduced echelon form from the row echelon form, we need

$$\begin{aligned} \sum_{k=2}^m [2(k-1)(n-k+1) + k-1] &= \sum_{k=2}^m [2(k-1)(n-k+1) + k-1] \\ &= \sum_{i=1}^{m-1} [2(i)(n-i) + i], \quad (i = k-1) \\ &= \sum_{i=1}^{m-1} [-2i^2 + (2n+1)i], \end{aligned}$$

or  $-\frac{2}{3}m^3 + m^2n$  flops if we keep only the leading terms.

### Get RREF, case 2: $m > n$

Procedure D needs  $n-1$  divisions to get the multipliers and it takes  $n-1$  times 1 multiplications with the multipliers. Then to get the zeros above the pivots, it involves  $n-1$  times 1 subtractions. As a result, procedure D costs  $2(n-1)(1) + n-1$  flops.

Procedure E needs  $n-2$  divisions to get the multipliers and it takes  $n-2$  times 2 multiplications with the multipliers. Then to get the zeros above the pivots, it involves  $n-2$  times 2 subtractions. As a result, procedure E costs  $2(n-2)(2) + n-2$  flops.

The procedure can go on, and we can thus again summarize the cost for each loop in the following table:

| $k$      | Get multipliers          | Multipliers multiply each row | Rows subtraction     |
|----------|--------------------------|-------------------------------|----------------------|
| $n$      | $1 : n - 1 = n - 1$ rows | $(n - 1)(1)$                  | $(n - 1)(1)$         |
| $n - 1$  | $1 : n - 2 = n - 2$ rows | $(n - 2)(2)$                  | $(n - 2)(2)$         |
| $n - 2$  | $1 : n - 3 = n - 3$ rows | $(n - 3)(3)$                  | $(n - 3)(3)$         |
| $\vdots$ | $\vdots$                 | $\vdots$                      | $\vdots$             |
| $k$      | $1 : k - 1 = k - 1$ rows | $(k - 1)(n - k + 1)$          | $(k - 1)(n - k + 1)$ |
| $\vdots$ | $\vdots$                 | $\vdots$                      | $\vdots$             |
| 2        | $1 : 1 = 1$ row          | $(1)(n - 2 + 1)$              | $(1)(n - 2 + 1)$     |

In the second loop to get the row reduced echelon form from the row echelon form, we need

$$\begin{aligned} \sum_{k=2}^n [2(k-1)(n-k+1) + k-1] &= \sum_{k=2}^n [2(k-1)(n-k+1) + k-1] \\ &= \sum_{i=1}^{n-1} [2(i)(n-i) + i], \quad (i = k-1) \\ &= \sum_{i=1}^{n-1} [-2i^2 + (2n+1)i], \end{aligned}$$

or  $\boxed{\frac{1}{3}n^3}$  flops if we keep only the leading terms.

**Total cost:** Step 9 involves  $mn$  flops (divisions) at most to make the pivots to be 1's. So the total cost is:

- the total cost for  $m \leq n$  is then  $m^2n - \frac{1}{3}m^3 - \frac{2}{3}m^3 + m^2n = \boxed{2m^2n - m^3}$  flops if we keep only the leading terms.
- the total cost for  $m > n$  is then  $mn^2 - \frac{1}{3}n^3 + \frac{1}{3}n^3 = \boxed{mn^2}$  flops if we keep only the leading term.

To conclude, the total cost is

$$\text{cost} = \begin{cases} 2m^2n - m^3 \text{ flops,} & \text{if } m \leq n; \\ mn^2 \text{ flops,} & \text{if } m > n. \end{cases}$$

And this completes the proof. ■

In CUR decomposition, we will use the Gram-Schmidt process to find the linearly independent columns of  $\mathbf{A}$  which shares similar complexity with this Gaussian elimination for row reduced echelon form. But the Gram-Schmidt process is more clear from the perspective of equations to be computed, i.e., we can have specific mathematical forms for the entries of the matrices.

## 5.5. Rank Decomposition

We previously mentioned that the CR decomposition is a special case of rank decomposition. Formally, we prove the existence of the rank decomposition rigorously in the following theorem.

### Theorem 5.1: (Rank Decomposition)

Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be factored as

$$\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{D}} \underset{r \times n}{\mathbf{F}},$$

where  $\mathbf{D} \in \mathbb{R}^{m \times r}$  has rank  $r$ , and  $\mathbf{F} \in \mathbb{R}^{r \times n}$  also has rank  $r$ , i.e.,  $\mathbf{D}, \mathbf{F}$  have full rank  $r$ .

The storage for the decomposition is then reduced or potentially increased from  $mn$  to  $r(m + n)$ .

**Proof** [of Theorem 5.1] By ULV decomposition in Theorem 4.1 (p. 141), we can decompose  $\mathbf{A}$  by

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}.$$

Let  $\mathbf{U}_0 = \mathbf{U}_{:,1:r}$  and  $\mathbf{V}_0 = \mathbf{V}_{1:r,:}$ , i.e.,  $\mathbf{U}_0$  contains only the first  $r$  columns of  $\mathbf{U}$ , and  $\mathbf{V}_0$  contains only the first  $r$  rows of  $\mathbf{V}$ . Then, we still have  $\mathbf{A} = \mathbf{U}_0 \mathbf{L} \mathbf{V}_0$  where  $\mathbf{U}_0 \in \mathbb{R}^{m \times r}$  and  $\mathbf{V}_0 \in \mathbb{R}^{r \times n}$ . This is also known as the reduced ULV decomposition as shown in Figure 4.1. Let  $\{\mathbf{D} = \mathbf{U}_0 \mathbf{L}$  and  $\mathbf{F} = \mathbf{V}_0\}$ , or  $\{\mathbf{D} = \mathbf{U}_0$  and  $\mathbf{F} = \mathbf{L} \mathbf{V}_0\}$ , we find such rank decomposition. ■

The rank decomposition is not unique. Even by elementary transformations, we have

$$\mathbf{A} = \mathbf{E}_1 \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{E}_2,$$

where  $\mathbf{E}_1 \in \mathbb{R}^{m \times m}$ ,  $\mathbf{E}_2 \in \mathbb{R}^{n \times n}$  represent elementary row and column operations,  $\mathbf{Z} \in \mathbb{R}^{r \times r}$ . The transformation is rather general, and there are dozens of these  $\mathbf{E}_1, \mathbf{E}_2, \mathbf{Z}$ . Similar construction on this decomposition as shown in the above proof, we can recover another rank decomposition.

Analogously, we can find such  $\mathbf{D}, \mathbf{F}$  by SVD, URV, CR, CUR, and many other decompositional algorithms. However, we may connect the different rank decompositions by the following lemma.

### Lemma 5.2: (Connection Between Rank Decompositions)

For any two rank decompositions of  $\mathbf{A} = \mathbf{D}_1 \mathbf{F}_1 = \mathbf{D}_2 \mathbf{F}_2$ , there exists a nonsingular matrix  $\mathbf{P}$  such that

$$\mathbf{D}_1 = \mathbf{D}_2 \mathbf{P} \quad \text{and} \quad \mathbf{F}_1 = \mathbf{P}^{-1} \mathbf{F}_2.$$

**Proof** [of Lemma 5.2] Since  $\mathbf{D}_1 \mathbf{F}_1 = \mathbf{D}_2 \mathbf{F}_2$ , we have  $\mathbf{D}_1 \mathbf{F}_1 \mathbf{F}_1^\top = \mathbf{D}_2 \mathbf{F}_2 \mathbf{F}_1^\top$ . It is trivial that  $\text{rank}(\mathbf{F}_1 \mathbf{F}_1^\top) = \text{rank}(\mathbf{F}_1) = r$  such that  $\mathbf{F}_1 \mathbf{F}_1^\top$  is a square matrix with full rank and

thus is nonsingular. This implies  $\mathbf{D}_1 = \mathbf{D}_2 \mathbf{F}_2 \mathbf{F}_1^\top (\mathbf{F}_1 \mathbf{F}_1^\top)^{-1}$ . Let  $\mathbf{P} = \mathbf{F}_2 \mathbf{F}_1^\top (\mathbf{F}_1 \mathbf{F}_1^\top)^{-1}$ , we have  $\mathbf{D}_1 = \mathbf{D}_2 \mathbf{P}$  and  $\mathbf{F}_1 = \mathbf{P}^{-1} \mathbf{F}_2$ .  $\blacksquare$

To end up this section, we write another form of the rank decomposition that will be useful to make connection to the tensor decomposition (Theorem 22.1, p. 402).

**Theorem 5.3: (Rank Decomposition, An Alternative Form)**

Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be factored as

$$\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{D}} \underset{n \times r}{\mathbf{E}^\top},$$

where  $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_r] \in \mathbb{R}^{m \times r}$  has rank  $r$ , and  $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_r] \in \mathbb{R}^{n \times r}$  also has rank  $r$ , i.e.,  $\mathbf{D}, \mathbf{E}$  have full rank  $r$ . Equivalently, the rank decomposition can be written as

$$\mathbf{A} = \sum_{i=1}^r \mathbf{d}_i \mathbf{e}_i^\top.$$

## 5.6. Application: Rank and Trace of an Idempotent Matrix

The CR decomposition is quite useful to prove the rank of an idempotent matrix. See also the orthogonal projection in Appendix D.2.

**Lemma 5.1: (Rank and Trace of an Idempotent Matrix)**

For any  $n \times n$  idempotent matrix  $\mathbf{A}$  (i.e.,  $\mathbf{A}^2 = \mathbf{A}$ ), the rank of  $\mathbf{A}$  equals the trace of  $\mathbf{A}$ .

**Proof** [of Lemma 5.1] Any  $n \times n$  rank- $r$  matrix  $\mathbf{A}$  has CR decomposition  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , where  $\mathbf{C} \in \mathbb{R}^{n \times r}$  and  $\mathbf{R} \in \mathbb{R}^{r \times n}$  with  $\mathbf{C}, \mathbf{R}$  having full rank  $r$ . Then,

$$\begin{aligned} \mathbf{A}^2 &= \mathbf{A}, \\ \mathbf{C}\mathbf{R}\mathbf{C}\mathbf{R} &= \mathbf{C}\mathbf{R}, \\ \mathbf{R}\mathbf{C}\mathbf{R} &= \mathbf{R}, \\ \mathbf{R}\mathbf{C} &= \mathbf{I}_r, \end{aligned}$$

where  $\mathbf{I}_r$  is an  $r \times r$  identity matrix. Thus

$$\text{trace}(\mathbf{A}) = \text{trace}(\mathbf{C}\mathbf{R}) = \text{trace}(\mathbf{R}\mathbf{C}) = \text{trace}(\mathbf{I}_r) = r,$$

which equals the rank of  $\mathbf{A}$ . The equality above is from the invariant of cyclic permutation of trace.  $\blacksquare$

## 5.7. Other Applications

The CR decomposition or rank decomposition is essential for the proof of many important theorems, such as the existence of pseudo-inverse in Lemma 27.19 (p. 456), finding the basis of the four subspaces in the fundamental theorem of linear algebra in Appendix B.1 (p. 428). The CR factorization can also be used for data interpretation or to solve computational problems, such as least squares where a reduced linear system can be considered to remove redundant variables. Readers will find the usage of the CR decomposition throughout the text.

## Chapter 6

# Skeleton/CUR Decomposition

### Contents

---

|     |                                                                                  |     |
|-----|----------------------------------------------------------------------------------|-----|
| 6.1 | Skeleton Decomposition . . . . .                                                 | 168 |
| 6.2 | Existence of the Skeleton Decomposition . . . . .                                | 168 |
| 6.3 | Computing the Skeleton Decomposition via the Gram-Schmidt Process . . . . .      | 171 |
| 6.4 | Computing the Skeleton Decomposition via Modified Gram-Schmidt Process . . . . . | 172 |
| 6.5 | Computing the Skeleton Decomposition via the Gaussian Elimination . . . . .      | 173 |
| 6.6 | Recover Reduced Row Echelon Form from Skeleton Decomposition . . . . .           | 174 |
| 6.7 | Randomized Algorithms . . . . .                                                  | 175 |
| 6.8 | Pseudoskeleton Decomposition via the SVD . . . . .                               | 175 |

---

## 6.1. Skeleton Decomposition

### Theorem 6.1: (Skeleton Decomposition)

Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be factored as

$$\mathbf{A}_{m \times n} = \mathbf{C}_{m \times r} \quad \mathbf{U}_{r \times r}^{-1} \quad \mathbf{R}_{r \times n},$$

where  $\mathbf{C}$  is some  $r$  linearly independent columns of  $\mathbf{A}$ ,  $\mathbf{R}$  is some  $r$  linearly independent rows of  $\mathbf{A}$  and  $\mathbf{U}$  is the nonsingular submatrix on the intersection.

- The storage for the decomposition is then reduced or potentially increased from  $mn$  floats to  $r(m + n) + r^2$  floats.
- Or further, if we only record the position of the indices, it requires  $mr$ ,  $nr$  floats for storing  $\mathbf{C}$ ,  $\mathbf{R}$  respectively and extra  $2r$  integers to remember the position of each column of  $\mathbf{C}$  in that of  $\mathbf{A}$  and each row of  $\mathbf{R}$  in that of  $\mathbf{A}$  (i.e., construct  $\mathbf{U}$  from  $\mathbf{C}$ ,  $\mathbf{R}$ ).

Skeleton decomposition is also known as the *CUR decomposition* follows from the notation in the decomposition. The illustration of skeleton decomposition is shown in Figure 6.1 where the **yellow** vectors denote the linearly independent columns of  $\mathbf{A}$  and **green** vectors denote the linearly independent rows of  $\mathbf{A}$ . Specifically, if  $I$ ,  $J$  index vectors both with size  $r$  that contain the indices of rows and columns selected from  $\mathbf{A}$  into  $\mathbf{R}$  and  $\mathbf{C}$  respectively,  $\mathbf{U}$  can be denoted as  $\mathbf{U} = \mathbf{A}[I, J]$  (see Definition 0.1, p. 15).

$$\mathbf{A}_{m \times n} = \mathbf{C}_{m \times r} \quad \mathbf{U}_{r \times r}^{-1} \quad \mathbf{R}_{r \times n}$$

**Figure 6.1:** Demonstration of skeleton decomposition of a matrix.

## 6.2. Existence of the Skeleton Decomposition

In Corollary 4.2 (p. 148), we proved the row rank and the column rank of a matrix are equal. In another word, we can also claim that the dimension of the column space and the dimension of the row space are equal. This property is essential for the existence of the skeleton decomposition.

We are then ready to prove the existence of the skeleton decomposition. The proof is rather elementary.

**Proof** [of Theorem 6.1] The proof relies on the existence of such nonsingular matrix  $\mathbf{U}$  which is central to this decomposition method.

**Existence of such nonsingular matrix  $\mathbf{U}$**  Since matrix  $\mathbf{A}$  is rank- $r$ , we can pick  $r$  columns from  $\mathbf{A}$  so that they are linearly independent. Suppose we put the specific  $r$  independent columns  $\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots, \mathbf{a}_{ir}$  into the columns of an  $m \times r$  matrix  $\mathbf{N} = [\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots, \mathbf{a}_{ir}] \in \mathbb{R}^{m \times r}$ . The dimension of the column space of  $\mathbf{N}$  is  $r$  so that the dimension of the row space of  $\mathbf{N}$  is also  $r$  by Corollary 4.2 (p. 148). Again, we can pick  $r$  linearly independent rows  $\mathbf{n}_{j1}^\top, \mathbf{n}_{j2}^\top, \dots, \mathbf{n}_{jr}^\top$  from  $\mathbf{N}$  and put the specific  $r$  rows into rows of an  $r \times r$  matrix  $\mathbf{U} = [\mathbf{n}_{j1}^\top; \mathbf{n}_{j2}^\top; \dots; \mathbf{n}_{jr}^\top] \in \mathbb{R}^{r \times r}$ . Using Corollary 4.2 (p. 148) again, the dimension of the column space of  $\mathbf{U}$  is also  $r$  which means there are the  $r$  linearly independent columns from  $\mathbf{U}$ . So  $\mathbf{U}$  is such a nonsingular matrix with size  $r \times r$ .

**Main proof** As long as we find the nonsingular  $r \times r$  matrix  $\mathbf{U}$  inside  $\mathbf{A}$ , we can find the existence of the skeleton decomposition as follows.

Suppose  $\mathbf{U} = \mathbf{A}[I, J]$  where  $I, J$  are index vectors of size  $r$ . Since  $\mathbf{U}$  is a nonsingular matrix, the columns of  $\mathbf{U}$  are linearly independent. Thus the columns of matrix  $\mathbf{C}$  based on the columns of  $\mathbf{U}$  are also linearly independent (i.e., select the  $r$  columns of  $\mathbf{A}$  with the same entries of the matrix  $\mathbf{U}$ ). Here  $\mathbf{C}$  is equal to the  $\mathbf{N}$  we construct above and  $\mathbf{C} = \mathbf{A}[:, J]$ .

As the rank of the matrix  $\mathbf{A}$  is  $r$ , if we take any other column  $\mathbf{a}_i$  of  $\mathbf{A}$ ,  $\mathbf{a}_i$  can be represented as a linear combination of the columns of  $\mathbf{C}$ , i.e., there exists a vector  $\mathbf{x}$  such that  $\mathbf{a}_i = \mathbf{Cx}$ , for all  $i \in \{1, 2, \dots, n\}$ . Let  $r$  rows of  $\mathbf{a}_i$  corresponding to the row entries of  $\mathbf{U}$  be  $\mathbf{r}_i \in \mathbb{R}^r$  for all  $i \in \{1, 2, \dots, n\}$  (i.e.,  $\mathbf{r}_i$  contains  $r$  entries of  $\mathbf{a}_i$ ). That is, select the  $r$  entries of  $\mathbf{a}_i$ 's corresponding to the entries of  $\mathbf{U}$  as follows:

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n} \quad \longrightarrow \quad \mathbf{A}[I, :] = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n] \in \mathbb{R}^{r \times n}.$$

Since  $\mathbf{a}_i = \mathbf{Cx}$ ,  $\mathbf{U}$  is a submatrix inside  $\mathbf{C}$ , and  $\mathbf{r}_i$  is a subvector inside  $\mathbf{a}_i$ , we have  $\mathbf{r}_i = \mathbf{Ux}$  which is equivalent to  $\mathbf{x} = \mathbf{U}^{-1}\mathbf{r}_i$ . Thus for every  $i$ , we have  $\mathbf{a}_i = \mathbf{CU}^{-1}\mathbf{r}_i$ . Combine the  $n$  columns of such  $\mathbf{r}_i$  into  $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n]$ , we obtain

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] = \mathbf{CU}^{-1}\mathbf{R},$$

from which the result follows.

In short, we first find  $r$  linearly independent columns of  $\mathbf{A}$  into  $\mathbf{C} \in \mathbb{R}^{m \times r}$ . From  $\mathbf{C}$ , we find an  $r \times r$  nonsingular submatrix  $\mathbf{U}$ . The  $r$  rows of  $\mathbf{A}$  corresponding to entries of  $\mathbf{U}$  can help to reconstruct the columns of  $\mathbf{A}$ . Again, the situation is shown in Figure 6.1. ■

In case  $\mathbf{A}$  is square and invertible, we have skeleton decomposition  $\mathbf{A} = \mathbf{CU}^{-1}\mathbf{R}$  where  $\mathbf{C} = \mathbf{R} = \mathbf{U} = \mathbf{A}$  such that the decomposition reduces to  $\mathbf{A} = \mathbf{AA}^{-1}\mathbf{A}$ .

**CR decomposition vs skeleton decomposition** We note that CR decomposition and skeleton decomposition share a similar form. Even for the symbols used  $\mathbf{A} = \mathbf{CR}$  for the CR decomposition and  $\mathbf{A} = \mathbf{CU}^{-1}\mathbf{R}$  for the skeleton decomposition.

Both in the CR decomposition and the skeleton decomposition, we can<sup>1</sup> select the first  $r$  independent columns to obtain the matrix  $\mathbf{C}$  (the symbol for both the CR decomposi-

---

<sup>1</sup> Here, we highlight that we can, but not necessarily, as we will see in the randomized algorithm for the skeleton decomposition.

tion and the skeleton decomposition). So  $\mathbf{C}$ 's in the CR decomposition and the skeleton decomposition are exactly the same. On the contrary,  $\mathbf{R}$  in the CR decomposition is the reduced row echelon form without the zero rows, whereas  $\mathbf{R}$  in the skeleton decomposition is exactly some rows from  $\mathbf{A}$  so that  $\mathbf{R}$ 's have different meanings in the two decompositional methods. We will formally show that  $\mathbf{U}^{-1}\mathbf{R}$  in skeleton decomposition is the row reduced echelon form without zero rows in Theorem 6.1.

#### A word on the uniqueness of CR decomposition and skeleton decomposition

As mentioned above, both in the CR decomposition and the skeleton decomposition, we select the first  $r$  linearly independent columns to obtain the matrix  $\mathbf{C}$ . In this sense, the CR and skeleton decompositions have a unique form. However, if we select the last  $r$  linearly independent columns, we will get a different CR decomposition or skeleton decomposition. We will not discuss this situation here as it is not the main interest of this text.

To repeat, in the above proof for the existence of the skeleton decomposition, we first find the  $r$  linearly independent columns of  $\mathbf{A}$  into the matrix  $\mathbf{C}$ . From  $\mathbf{C}$ , we find an  $r \times r$  nonsingular submatrix  $\mathbf{U}$ . From the submatrix  $\mathbf{U}$ , we finally find the final row submatrix  $\mathbf{R} \in \mathbb{R}^{r \times n}$ . A further question can be posed that if matrix  $\mathbf{A}$  has rank  $r$ , matrix  $\mathbf{C}$  contains  $r$  linearly independent columns, and matrix  $\mathbf{R}$  contains  $r$  linearly independent rows, then whether the  $r \times r$  “intersection” of  $\mathbf{C}$  and  $\mathbf{R}$  is invertible or not <sup>2</sup>.

#### Corollary 6.1: (Nonsingular Intersection)

If matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has rank  $r$ , matrix  $\mathbf{C}$  contains  $r$  linearly independent columns, and matrix  $\mathbf{R}$  contains  $r$  linearly independent rows, then the  $r \times r$  “intersection” matrix  $\mathbf{U}$  of  $\mathbf{C}$  and  $\mathbf{R}$  is invertible.

**Proof** [of Corollary 6.1] If  $I, J$  are the indices of rows and columns selected from  $\mathbf{A}$  into  $\mathbf{R}$  and  $\mathbf{C}$  respectively, then,  $\mathbf{R}$  can be denoted as  $\mathbf{R} = \mathbf{A}[I, :]$ ,  $\mathbf{C}$  can be represented as  $\mathbf{C} = \mathbf{A}[:, J]$ , and  $\mathbf{U}$  can be denoted as  $\mathbf{U} = \mathbf{A}[I, J]$ .

Since  $\mathbf{C}$  contains  $r$  linearly independent columns of  $\mathbf{A}$ , any column  $\mathbf{a}_i$  of  $\mathbf{A}$  can be represented as  $\mathbf{a}_i = \mathbf{C}\mathbf{x}_i = \mathbf{A}[:, J]\mathbf{x}_i$  for all  $i \in \{1, 2, \dots, n\}$ . This implies the  $r$  entries of  $\mathbf{a}_i$  corresponding to the  $I$  indices can be represented by the columns of  $\mathbf{U}$  such that  $\mathbf{a}_i[I] = \mathbf{U}\mathbf{x}_i \in \mathbb{R}^r$  for all  $i \in \{1, 2, \dots, n\}$ , i.e.,

$$\mathbf{a}_i = \mathbf{C}\mathbf{x}_i = \mathbf{A}[:, J]\mathbf{x}_i \in \mathbb{R}^m \quad \longrightarrow \quad \mathbf{a}_i[I] = \mathbf{A}[I, J]\mathbf{x}_i = \mathbf{U}\mathbf{x}_i \in \mathbb{R}^r.$$

Since  $\mathbf{R}$  contains  $r$  linearly independent rows of  $\mathbf{A}$ , the row rank and column rank of  $\mathbf{R}$  are equal to  $r$ . Combining the facts above, the  $r$  columns of  $\mathbf{R}$  corresponding to indices  $J$  (i.e., the  $r$  columns of  $\mathbf{U}$ ) are linearly independent.

Again, by applying Corollary 4.2 (p. 148), the dimension of the row space of  $\mathbf{U}$  is also equal to  $r$  which means there are the  $r$  linearly independent rows from  $\mathbf{U}$ , and  $\mathbf{U}$  is invertible. ■

---

<sup>2</sup>. We thank Gilbert Strang for raising this interesting question.

### 6.3. Computing the Skeleton Decomposition via the Gram-Schmidt Process

In Section 3.9, we have discussed how to tackle dependent columns when orthogonalizing the matrix. We can thus use the Gram-Schmidt process to select  $r$  linearly independent columns from  $\mathbf{A}$  resulting in  $\mathbf{C}$ . And use the Gram-Schmidt process to select  $r$  linearly independent columns from  $\mathbf{C}^\top$  which results in  $\mathbf{U}$  and  $\mathbf{R}$ . The process is shown in Algorithm 26.

---

**Algorithm 26** Skeleton Decomposition via Gram-Schmidt Process

---

**Require:** Rank- $r$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  with size  $m \times n$ ;

```

1: Initially set column count $ck = 0$ and row count $rk = 0$;
2: Set $\mathbf{q}_1 = \mathbf{a}_1/r_{11}, r_{11} = \|\mathbf{a}_1\|$; ▷ Suppose the first column is nonzero and $r_{11} \neq 0$
3: for $k = 2$ to n do
4: $\mathbf{q}_k = (\mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i)/r_{kk}, r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, r_{kk} = \|\mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i\|, \forall i \in \{1, \dots, k-1\}$;
5: if $r_{kk} \neq 0$ then
6: Select the k -th column of \mathbf{A} into ck -th column of \mathbf{C} ;
7: $ck = ck + 1$;
8: end if
9: end for
10: $\mathbf{C} = [\mathbf{c}_1^\top; \mathbf{c}_2^\top; \dots; \mathbf{c}_m^\top]$, where \mathbf{c}_i^\top is the i -th row of \mathbf{C} ;
11: Set $\mathbf{q}_1 = \mathbf{c}_1/r_{11}, r_{11} = \|\mathbf{c}_1\|$; ▷ Suppose the first row is nonzero and $r_{11} \neq 0$
12: for $k = 2$ to m do
13: $\mathbf{q}_k = (\mathbf{c}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i)/r_{kk}, r_{ik} = \mathbf{q}_i^\top \mathbf{c}_k, r_{kk} = \|\mathbf{c}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i\|, \forall i \in \{1, \dots, k-1\}$;
14: if $r_{kk} \neq 0$ then
15: Select the k -th row of \mathbf{C} into rk -th row of \mathbf{U} ;
16: $rk = rk + 1$;
17: end if
18: end for
19: Select the rows from \mathbf{A} corresponding to the rows of \mathbf{U} into \mathbf{R} .

```

---

**Theorem 6.1: (Algorithm Complexity: Skeleton via Gram-Schmidt)**

Algorithm 26 requires  $\sim 2(mn^2 + rm^2 + r^3)$  flops to compute a skeleton decomposition of an  $m \times n$  matrix. We will show this algorithm is not the best way to get the skeleton decomposition.

**Proof** [of Theorem 6.1] This is just applying Theorem 3.1 twice to get matrix  $\mathbf{U}$  (to get  $\mathbf{C}$ , it costs  $2mn^2$  flops. And to get  $\mathbf{U}$ , it costs  $2rm^2$  flops.) and applying Theorem 1.2 to compute the inverse of  $\mathbf{U}$  ( $2r^3$  flops). ■

Note that, choosing the first linearly independent columns from  $\mathbf{A}$  will make the inverse of  $\mathbf{U}$  not stable. A maxvol procedure is used in (Goreinov et al., 2010). We will introduce a pseudoskeleton decomposition to overcome this problem at the end of the section.

## 6.4. Computing the Skeleton Decomposition via Modified Gram-Schmidt Process

In the Gram-Schmidt process, we want to find orthonormal vectors. However, this orthogonal property is not useful for skeleton decomposition. We just need to decide whether the vectors are linearly independent or not. We can thus use the Gram-Schmidt process with slight modification to select  $r$  linearly independent columns from  $\mathbf{A}$  to get  $\mathbf{C}$ . And use the Gram-Schmidt process (again with slight modification) to select  $r$  linearly independent columns from  $\mathbf{C}^\top$  to get  $\mathbf{U}$  and  $\mathbf{R}$ .

For a set of  $n$  vectors  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$  from the columns of  $\mathbf{A}$ , if we want to project a vector  $\mathbf{a}_k$  onto the space perpendicular to the space spanned by  $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_{k-1}$  (suppose the  $k-1$  vectors are linearly independent), we can write out this projection from Equation (3.1)

$$\begin{aligned}\mathbf{b}_k &= \mathbf{a}_k - \left( \frac{\mathbf{b}_1^\top \mathbf{a}_k}{\mathbf{b}_1^\top \mathbf{b}_1} \mathbf{b}_1 + \frac{\mathbf{b}_2^\top \mathbf{a}_k}{\mathbf{b}_2^\top \mathbf{b}_2} \mathbf{b}_2 + \dots + \frac{\mathbf{b}_{k-1}^\top \mathbf{a}_k}{\mathbf{b}_{k-1}^\top \mathbf{b}_{k-1}} \mathbf{b}_{k-1} \right) \\ &= \mathbf{a}_k - \sum_{i=1}^{k-1} \frac{\mathbf{b}_i^\top \mathbf{a}_k}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{b}_i,\end{aligned}$$

where  $\mathbf{b}_k$  is the projection of  $\mathbf{a}_k$  onto the space perpendicular to the space spanned by  $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_{k-1}\}$ ,  $\mathbf{b}_{k-1}$  is the projection of  $\mathbf{a}_{k-1}$  onto the space perpendicular to the space spanned by  $\{\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_{k-2}\}$ , and so on.

If  $\mathbf{b}_k$  is nonzero,  $\mathbf{a}_k$  is linearly independent of  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{k-1}$  which will be chosen into the column of  $\mathbf{C}$ . Similarly, we can choose linearly independent rows from  $\mathbf{C}$  to produce  $\mathbf{U}$ . The process is shown in Algorithm 27.

### Theorem 6.1: (Algorithm Complexity: Skeleton via Modified Gram-Schmidt)

Algorithm 27 requires  $\sim 2(mn^2 + rm^2 + r^3)$  flops to compute a skeleton decomposition of an  $m \times n$  matrix.

**Proof** [of Theorem 6.1] For the last loop  $n$  in step 4, the  $\frac{\mathbf{b}_i^\top \mathbf{a}_n}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{b}_i$  involves:

- a.  $m$  multiplications and  $m-1$  additions for the calculation of  $\mathbf{b}_i^\top \mathbf{a}_n$ ;
- b.  $m$  multiplications for the calculation of  $\mathbf{b}_i^\top \mathbf{a}_n * \mathbf{b}_i$ ;
- c.  $m$  multiplications and  $m-1$  additions for the calculation of  $\mathbf{b}_i^\top \mathbf{b}_i$ ;
- d. 1 division;

There are  $n-1$  such terms to compute procedure  $a, b, d$  above. However, for procedure  $c$ , we did the calculation of  $\mathbf{b}_i^\top \mathbf{b}_i$  for  $i \in \{1, 2, \dots, n-2\}$  in the previous loops. As a result, we need  $\boxed{3m(n-1) + 2m - 1}$  to compute  $\frac{\mathbf{b}_i^\top \mathbf{a}_n}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{b}_i$  for  $i \in \{1, 2, \dots, n-1\}$  where the last  $2m-1$  flops is to compute  $\mathbf{b}_{n-1}^\top \mathbf{b}_{n-1}$ . The final  $\mathbf{a}_k - \sum_{i=1}^{k-1} \frac{\mathbf{b}_i^\top \mathbf{a}_k}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{b}_i$  then takes  $(n-2)m$  additions and  $m$  subtractions which cost  $\boxed{(n-1)m}$  flops. So the total cost of step 4 in the last loop  $n$  is  $4m(n-1) + 2m - 1$  flops. Let  $f(i) = 4m(i-1) + 2m - 1$ , the total cost for step 3 to step 9 can be obtained by

$$\text{cost} = f(2) + f(3) + \dots + f(n).$$

**Algorithm 27** Skeleton Decomposition via Modified Gram-Schmidt Process

---

**Require:** Rank- $r$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  with size  $m \times n$ ;

- 1: Initially set column count  $ck = 0$  and row count  $rk = 0$ ;
- 2: Set  $\mathbf{b}_1 = \mathbf{a}_1$ ; ▷ Suppose the first column is nonzero, 0 flops
- 3: **for**  $k = 2$  to  $n$  **do**
- 4:    $\mathbf{b}_k = \mathbf{a}_k - \sum_{i=1}^{k-1} \frac{\mathbf{b}_i^\top \mathbf{a}_k}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{b}_i$ , (Skip the zero terms  $\mathbf{b}_i$ );
- 5:   **if**  $\mathbf{b}_k \neq 0$  **then**
- 6:     Select the  $k$ -th column of  $\mathbf{A}$  into the  $ck$ -th column of  $\mathbf{C}$ ;
- 7:      $ck = ck + 1$ ;
- 8:   **end if**
- 9: **end for**
- 10:  $\mathbf{C} = [\mathbf{c}_1^\top; \mathbf{c}_2^\top; \dots; \mathbf{c}_m^\top]$ , where  $\mathbf{c}_i^\top$  is the  $i$ -th row of  $\mathbf{C}$ ;
- 11: Set  $\mathbf{b}_1 = \mathbf{c}_1$ , (Suppose the first row is nonzero);
- 12: **for**  $k = 2$  to  $m$  **do**
- 13:    $\mathbf{b}_k = \mathbf{c}_k - \sum_{i=1}^{k-1} \frac{\mathbf{b}_i^\top \mathbf{c}_k}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{b}_i$ , (Skip the zero terms  $\mathbf{b}_i$ );
- 14:   **if**  $\mathbf{b}_k \neq 0$  **then**
- 15:     Select the  $k$ -th row of  $\mathbf{C}$  into the  $rk$ -th row of  $\mathbf{U}$ ;
- 16:      $rk = rk + 1$ ;
- 17:   **end if**
- 18: **end for**
- 19: Select the rows from  $\mathbf{A}$  corresponding to the rows of  $\mathbf{U}$  into  $\mathbf{R}$ .

---

Simple calculations can show this loop takes  $4m\frac{n^2-n}{2} + (2m-1)(n-1)$  flops, or  $\boxed{2mn^2}$  flops if keep only the leading term.

Similarly, for step 13, the total cost for the second loop is  $4r\frac{m^2-m}{2} + (2r-1)(m-1)$  flops, or  $\boxed{2rm^2}$  flops if keep only the leading term.

As a result, the total cost is  $2(mn^2 + rm^2 + r^3)$  flops if keep only the leading term, where  $2r^3$  results from the computation of the inverse of  $\mathbf{U}$  by Theorem 1.2 (p. 47). ■

The complexity of Algorithm 27 is the same as that of Algorithm 26 as they are mathematically equivalent. However, the meaning from Algorithm 27 is much more clear.

## 6.5. Computing the Skeleton Decomposition via the Gaussian Elimination

In Algorithm 25, we discussed the method to get the row reduced echelon form by Gaussian elimination. And the columns containing the pivots are the ones that are linearly independent. We thus can use this row reduced echelon form to find the skeleton decomposition.

**Algorithm 28** Skeleton Decomposition via Gaussian Elimination

**Require:** Rank- $r$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  with size  $m \times n$ ;

- 1: Use Algorithm 25 to get the row reduced echelon form of  $\mathbf{A}$  and choose the columns containing pivots from  $\mathbf{A}$  into  $\mathbf{C}$ ;
- 2: Use Algorithm 25 to get the row reduced echelon form of  $\mathbf{C}^\top$  and choose the columns containing pivots from  $\mathbf{C}^\top$  into the columns of  $\mathbf{U}^\top$ ;
- 3: Select the rows from  $\mathbf{A}$  corresponding to the rows of  $\mathbf{U}$  into  $\mathbf{R}$ .

**Theorem 6.1: (Algorithm Complexity: Skeleton via Guassian Elimination)**

Algorithm 28 requires

$$\text{cost} = \begin{cases} (2m^2n - m^3) + (2r^2m - r^3) + 2r^3 = 2m^2n - m^3 + 2r^2m + r^3, & \text{if } m \leq n; \\ (mn^2) + (2r^2m - r^3) + 2r^3 = mn^2 + 2r^2m + r^3, & \text{if } m > n. \end{cases}$$

flops to compute a skeleton decomposition of an  $m \times n$  matrix. The cost of  $2r^3$  above is again from the calculation of the inverse of  $\mathbf{U}$ .

The proof of the above theorem is trivial since  $r \leq m$  by Theorem 5.1.

## 6.6. Recover Reduced Row Echelon Form from Skeleton Decomposition

In this section, we formally claim that  $\mathbf{U}^{-1}\mathbf{R}$  is the row reduced echelon form without zero rows.

**Theorem 6.1: (Recover RREF from Skeleton Decomposition)**

Suppose we have skeleton decomposition via Algorithm 26, Algorithm 27, or Algorithm 28 that  $\mathbf{A} = \mathbf{C}\mathbf{U}^{-1}\mathbf{R}$ , then  $\mathbf{U}^{-1}\mathbf{R}$  is the row reduced echelon form without zero rows.

**Proof** [of Theorem 6.1] Without loss of generality, we assume matrix  $\mathbf{A} \in \mathbb{R}^{4 \times 5}$  and the linearly independent columns of matrix  $\mathbf{A}$  are the columns 1, 3, 5, which means column 2 is a multiple  $m_1$  of column 1, column 4 is a linear combination of column 1 and 3: column 4 =  $n_1$  column 1 +  $n_2$  column 2.

Then we use  $\mathbf{U}$  to recover the column 1, we will have  $\mathbf{x}_1 = \mathbf{U}^{-1}\mathbf{r}_1 = [1; 0; 0]$ ;

When we use  $\mathbf{U}$  to recover the column 2, we will have  $\mathbf{x}_2 = \mathbf{U}^{-1}\mathbf{r}_2 = [m_1; 0; 0]$ ;

When we use  $\mathbf{U}$  to recover column 3, we will have  $\mathbf{x}_3 = \mathbf{U}^{-1}\mathbf{r}_3 = [0; 1; 0]$ ;

When we use  $\mathbf{U}$  to recover column 4, we will have  $\mathbf{x}_4 = \mathbf{U}^{-1}\mathbf{r}_4 = [n_1; n_2; 0]$ ;

When we use  $\mathbf{U}$  to recover column 5, we will have  $\mathbf{x}_5 = \mathbf{U}^{-1}\mathbf{r}_5 = [0; 0; 1]$ .

This clearly gives this row reduced echelon form without zero rows:

$$\mathbf{X} = \begin{bmatrix} 1 & m_1 & 0 & n_1 & 0 \\ 0 & 0 & 1 & n_2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

where we find the pivots are all 1's, the entries above the pivots are all 0's. Now we go back to matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , Algorithm 26, Algorithm 27, or Algorithm 28 find the first linearly independent columns from left to right as columns  $k, l, m, n, \dots$  where  $k \leq l \leq m \leq n \leq \dots$ . We will get similar results. Pivots are in 1: row 1, column  $k$ ; 2: row 2, column  $l$ ; 3: row 3, column  $m$ ; 4: row 4, column  $n$ , ....

The entries above the pivots are 0's for sure.

For columns  $k+1, k+2, \dots, l-1$ , they are combinations of columns  $k$ , so the entries below row 1, columns  $k+1, k+2, \dots, l-1$  are zero.

For columns  $l+1, l+2, \dots, m-1$ , they are combinations of columns  $k, l$ , so the entries below row 2, columns  $l+1, l+2, \dots, m-1$  are zero.

The process can go on and it produces the reduced row echelon form without zero rows. ■

## 6.7. Randomized Algorithms

The computing methods for skeleton decomposition described here are choosing the first linearly independent columns from  $\mathbf{A}$  into  $\mathbf{C}$  same as that in the CR decomposition. However, this is not necessarily. Randomized algorithms are discussed in (Mahoney and Drineas, 2009; Boutsidis et al., 2009; Drineas et al., 2012; Kishore Kumar and Schneider, 2017) which do not choose the linearly independent columns from left to right.

In (Goreinov et al., 1997; Goreinov and Tyrtyshnikov, 2001), an approximation of skeleton decomposition has been developed which is called *pseudoskeleton approximation*. The  $k$  columns in  $\mathbf{C}$  and  $k$  rows in  $\mathbf{R}$  with  $k < r$  were chosen such that their intersection  $\mathbf{U}_{k \times k}$  has maximum volume (i.e., the maximum determinant among all  $k \times k$  submatrices of  $\mathbf{A}$ ).

## 6.8. Pseudoskeleton Decomposition via the SVD

We will discuss singular value decomposition (SVD) extensively in Section 14. But now, let's just assume we have the background of SVD, and we will show how to use the SVD to approximate the skeleton decomposition. Feel free to skip this section for a first reading.

For matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , we want to construct an approximation of  $\mathbf{A}$  with rank  $\gamma \leq \min(m, n)$  in the form of skeleton decomposition. That is  $\mathbf{A}$  is approximated by  $\mathbf{A} \approx \mathbf{C}\mathbf{G}\mathbf{R}$  where  $\mathbf{C}$  and  $\mathbf{R}$  contain  $\gamma$  selected columns and rows respectively and  $\mathbf{G} = \mathbf{U}^{-1}$  with  $\mathbf{U}$  being the intersection of  $\mathbf{C}$  and  $\mathbf{R}$ . Specifically, if  $I, J$  are the indices of the selected rows and columns respectively,  $\mathbf{U} = \mathbf{A}[I, J]$ . Note here the  $\gamma$  is not necessarily equal to the rank  $r$  of  $\mathbf{A}$ .

Instead of choosing the  $r$  linearly independent columns from  $\mathbf{A}$  (as shown in skeleton decomposition), we choose  $k$  random columns into matrix  $\mathbf{C}$  with  $k > r$  or even  $k = \min(m, n)$  given by the indices  $J$  from the matrix  $\mathbf{A}$ . That is  $\mathbf{C} = \mathbf{A}[:, J] \in \mathbb{R}^{m \times k}$ . Now we select  $k$  rows from  $\mathbf{A}$  by the indices  $I$  from the matrix  $\mathbf{A}$  into matrix  $\mathbf{R} = \mathbf{A}[I, :]$  such that the volume of the intersection matrix  $\mathbf{U} = \mathbf{A}[I, J]$  is maximized, i.e.,  $\det(\mathbf{U})$  is maximized given the  $\mathbf{C}$  chosen randomly. Note here  $\mathbf{C}$  is randomly chosen, but  $\mathbf{R}$  is not randomly chosen. Now the decomposition becomes

$$\mathbf{A} = \mathbf{C}_{m \times k} \mathbf{U}_{k \times k}^{-1} \mathbf{R}_{k \times n}.$$

Again, the inverse of  $\mathbf{U}_{k \times k}$  is not stable due to the random choice. To overcome this, we decompose  $\mathbf{U}_{k \times k}$  by full SVD (refer to Section 14, p. 264 for the difference between the reduced SVD and full SVD):

$$\mathbf{U}_{k \times k} = \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^\top,$$

where  $\mathbf{U}_k, \mathbf{V}_k \in \mathbb{R}^{k \times k}$  are orthogonal matrices,  $\boldsymbol{\Sigma}_k$  is a diagonal matrix containing  $k$  singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k$  with zeros allowed. Now we choose  $\gamma$  singular values that are greater than some value  $\epsilon$  and truncate the  $\mathbf{U}_k, \mathbf{V}_k, \boldsymbol{\Sigma}_k$  according to the  $\gamma$  selected singular values such that  $\mathbf{U}_{k \times k}$  is approximated by a rank- $\gamma$  matrix  $\mathbf{U}_{k \times k} \approx \mathbf{U}_\gamma \boldsymbol{\Sigma}_\gamma \mathbf{V}_\gamma^\top$ , where  $\mathbf{U}_\gamma, \mathbf{V}_\gamma \in \mathbb{R}^{k \times \gamma}$ , and  $\boldsymbol{\Sigma}_\gamma \in \mathbb{R}^{\gamma \times \gamma}$ . Therefore, the pseudoinverse of  $\mathbf{U}_{k \times k}$  is <sup>3</sup>

$$\mathbf{U}^+ = (\mathbf{U}_\gamma \boldsymbol{\Sigma}_\gamma \mathbf{V}_\gamma^\top)^{-1} = \mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1} \mathbf{U}_\gamma^\top.$$

As a result, the matrix  $\mathbf{A}$  is approximated by a rank- $\gamma$  matrix

$$\begin{aligned} \mathbf{A} &\approx \mathbf{C} \mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1} \mathbf{U}_\gamma^\top \mathbf{R} \\ &= \mathbf{C}_2 \mathbf{R}_2, \quad (\text{Let } \mathbf{C}_2 = \mathbf{C} \mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1/2} \text{ and } \mathbf{R}_2 = \boldsymbol{\Sigma}_\gamma^{-1/2} \mathbf{U}_\gamma^\top \mathbf{R}) \end{aligned} \tag{6.1}$$

where  $\mathbf{C}_2$  and  $\mathbf{R}_2$  are rank- $\gamma$  matrices. Please refer to (Goreinov et al., 1997; Kishore Kumar and Schneider, 2017) for further details about the choice of how to choose such  $\epsilon$  systematically. In the method described above, we only choose  $\mathbf{C}$  randomly. Algorithms introduced in (Zhu and Lin, 2011) choose both  $\mathbf{C}$  and  $\mathbf{R}$  randomly and more stable results are obtained.

---

<sup>3</sup>. See Section 14.7.1 (p. 278) and Appendix E (p.445) for the detailed discussion of pseudoinverse.

## Chapter 7

# Interpolative Decomposition (ID)

### Contents

---

|     |                                                               |     |
|-----|---------------------------------------------------------------|-----|
| 7.1 | Interpolative Decomposition . . . . .                         | 178 |
| 7.2 | Existence of the Column Interpolative Decomposition . . . . . | 179 |
| 7.3 | Row ID and Two-Sided ID . . . . .                             | 183 |
| 7.4 | Computing the Column ID via the CPQR . . . . .                | 185 |
| 7.5 | Low-Rank Column ID via the RRQR . . . . .                     | 186 |
| 7.6 | Computing the ID via Randomized Algorithm . . . . .           | 188 |

---

## 7.1. Interpolative Decomposition

The column interpolative decomposition (ID) factors a matrix as the product of two matrices, one of which contains selected columns from the original matrix, and the other of which has a subset of columns consisting of the identity matrix and all its values are no greater than 1 in absolute value. Formally, we have the following theorem describing the details of the column ID.

**Theorem 7.1: (Column Interpolative Decomposition)**

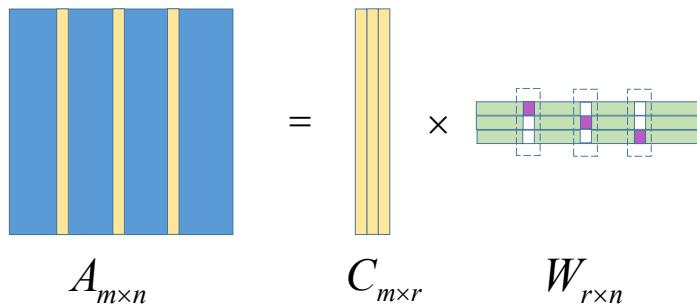
Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be factored as

$$\mathbf{A}_{m \times n} = \mathbf{C}_{m \times r} \mathbf{W}_{r \times n},$$

where  $\mathbf{C} \in \mathbb{R}^{m \times r}$  is some  $r$  linearly independent columns of  $\mathbf{A}$ ,  $\mathbf{W} \in \mathbb{R}^{r \times n}$  is the matrix to reconstruct  $\mathbf{A}$  which contains an  $r \times r$  identity submatrix (under a mild column permutation). Specifically entries in  $\mathbf{W}$  have values no larger than 1 in magnitude:

$$\max |w_{ij}| \leq 1, \forall i \in [1, r], j \in [1, n].$$

The storage for the decomposition is then reduced or potentially increased from  $mn$  floats to  $mr$ ,  $(n - r)r$  floats for storing  $\mathbf{C}, \mathbf{W}$  respectively and extra  $r$  integers are required to remember the position of each column of  $\mathbf{C}$  in that of  $\mathbf{A}$ .



**Figure 7.1:** Demonstration of the column ID of a matrix where the **yellow** vector denotes the linearly independent columns of  $\mathbf{A}$ , white entries denote zero, and **purple** entries denote one.

The illustration of the column ID is shown in Figure 7.1 where the **yellow** vectors denote the linearly independent columns of  $\mathbf{A}$  and the **purple** vectors in  $\mathbf{W}$  form an  $r \times r$  identity submatrix. The positions of the **purple** vectors inside  $\mathbf{W}$  are exactly the same as the positions of the corresponding **yellow** vectors inside  $\mathbf{A}$ . The column ID is very similar to the CR decomposition (Theorem 5.1, p. 156), both select  $r$  linearly independent columns into the first factor and the second factor contains an  $r \times r$  identity submatrix. The difference is in that the CR decomposition will exactly choose the first  $r$  linearly independent columns

into the first factor and the identity submatrix appears in the pivots (Definition 1.2, p. 34). And more importantly, the second factor in the CR decomposition comes from the RREF (Lemma 5.2, p. 158). Therefore, the column ID can also be utilized in the applications of the CR decomposition, say proving the fact of rank equals trace in idempotent matrices (Lemma 5.1, p. 165), and proving the elementary theorem in linear algebra that column rank equals row rank of a matrix (Corollary 4.2, p. 148). Moreover, the column ID is also a special case of rank decomposition (Theorem 5.1, p. 164) and is apparently not unique. The connection between different column IDs is given by Lemma 5.2 (p. 164).

**Notations that will be extensively used in the sequel** Following again the Matlab-style notation, if  $J_s$  is an index vector with size  $r$  that contains the indices of columns selected from  $\mathbf{A}$  into  $\mathbf{C}$ , then  $\mathbf{C}$  can be denoted as  $\mathbf{C} = \mathbf{A}[:, J_s]$  (Definition 0.1, p. 15). The matrix  $\mathbf{C}$  contains “skeleton” columns of  $\mathbf{A}$ , hence the subscript  $s$  in  $J_s$ . From the “skeleton” index vector  $J_s$ , the  $r \times r$  identity matrix inside  $\mathbf{W}$  can be recovered by

$$\mathbf{W}[:, J_s] = \mathbf{I}_r \in \mathbb{R}^{r \times r}.$$

Suppose further we put the remaining indices of  $\mathbf{A}$  into an index vector  $J_r$  where

$$J_s \cap J_r = \emptyset \quad \text{and} \quad J_s \cup J_r = \{1, 2, \dots, n\}.$$

The remaining  $n - r$  columns in  $\mathbf{W}$  consists of an  $r \times (n - r)$  *expansion matrix* since the matrix contains *expansion coefficients* to reconstruct the columns of  $\mathbf{A}$  from  $\mathbf{C}$ :

$$\mathbf{E} = \mathbf{W}[:, J_r] \in \mathbb{R}^{r \times (n-r)},$$

where the entries of  $\mathbf{E}$  are known as the *expansion coefficients*. Moreover, let  $\mathbf{P} \in \mathbb{R}^{n \times n}$  be a (column) permutation matrix (Definition 0.15, p. 19) defined by  $\mathbf{P} = \mathbf{I}_n[:, (J_s, J_r)]$  so that

$$\mathbf{AP} = \mathbf{A}[:, (J_s, J_r)] = [\mathbf{C}, \mathbf{A}[:, J_r]],$$

and

$$\mathbf{WP} = \mathbf{W}[:, (J_s, J_r)] = [\mathbf{I}_r, \mathbf{E}] \quad \xrightarrow{\text{leads to}} \quad \mathbf{W} = [\mathbf{I}_r, \mathbf{E}] \mathbf{P}^\top. \quad (7.1)$$

## 7.2. Existence of the Column Interpolative Decomposition

**Cramer’s rule** The proof of the existence of the column ID relies on the Cramer’s rule that we shall shortly discuss here. Consider a system of  $n$  linear equations for  $n$  unknowns, represented in matrix multiplication form as follows :

$$\mathbf{M}\mathbf{x} = \mathbf{l},$$

where  $\mathbf{M} \in \mathbb{R}^{n \times n}$  is nonsingular and  $\mathbf{x}, \mathbf{l} \in \mathbb{R}^n$ . Then the theorem states that in this case, the system has a unique solution, whose individual values for the unknowns are given by:

$$x_i = \frac{\det(\mathbf{M}_i)}{\det(\mathbf{M})}, \quad \text{for all } i \in \{1, 2, \dots, n\},$$

where  $\mathbf{M}_i$  is the matrix formed by replacing the  $i$ -th column of  $\mathbf{M}$  with the column vector  $\mathbf{l}$ . In full generality, the Cramer's rule considers the matrix equation

$$\mathbf{M}\mathbf{X} = \mathbf{L},$$

where  $\mathbf{M} \in \mathbb{R}^{n \times n}$  is nonsingular and  $\mathbf{X}, \mathbf{L} \in \mathbb{R}^{n \times m}$ . Let  $I = [i_1, i_2, \dots, i_k]$  and  $J = [j_1, j_2, \dots, j_k]$  be two index vectors where  $1 \leq i_1 \leq i_2 \leq \dots \leq i_k \leq n$  and  $1 \leq j_1 \leq j_2 \leq \dots \leq j_k \leq n$ . Then  $\mathbf{X}[I, J]$  is a  $k \times k$  submatrix of  $\mathbf{X}$ . Let further  $\mathbf{M}_L(I, J)$  be the  $n \times n$  matrix formed by replacing the  $i_s$  column of  $\mathbf{M}$  by  $j_s$  column of  $\mathbf{L}$  for all  $s \in \{1, 2, \dots, k\}$ . Then

$$\det(\mathbf{X}[I, J]) = \frac{\det(\mathbf{M}_L(I, J))}{\det(\mathbf{M})}.$$

When  $I, J$  are of size 1, it follows that

$$x_{ij} = \frac{\det(\mathbf{M}_L(i, j))}{\det(\mathbf{M})}. \quad (7.2)$$

Now we are ready to prove the existence of the column ID.

**Proof** [of Theorem 7.1] We have mentioned above the proof relies on the Cramer's rule. If we can show the entries of  $\mathbf{W}$  can be denoted by the Cramer's rule equality in Equation (7.2) and the numerator is smaller than the denominator, then we can complete the proof. However, we notice that the matrix in the denominator of Equation (7.2) is a square matrix. Here comes the trick.

**Step 1: column ID for full row rank matrix** For a start, we first consider the full row rank matrix  $\mathbf{A}$  (which implies  $r = m$ ,  $m \leq n$ , and  $\mathbf{A} \in \mathbb{R}^{r \times n}$  such that the matrix  $\mathbf{C} \in \mathbb{R}^{r \times r}$  is a square matrix in the column ID  $\mathbf{A} = \mathbf{CW}$  that we want). Determine the “skeleton” index vector  $J_s$  by

$$J_s = \arg \max_J \{|\det(\mathbf{A}[:, J])| : J \text{ is a subset of } \{1, 2, \dots, n\} \text{ with size } r = m\}, \quad (7.3)$$

i.e.,  $J_s$  is the index vector that is determined by maximizing the magnitude of the determinant of  $\mathbf{A}[:, J]$ . As we have discussed in the last section, there exists a (column) permutation matrix such that

$$\mathbf{AP} = [\mathbf{A}[:, J_s] \quad \mathbf{A}[:, J_r]].$$

Since  $\mathbf{C} = \mathbf{A}[:, J_s]$  has full column rank  $r = m$ , it is then nonsingular. The above equation can be rewritten as

$$\begin{aligned} \mathbf{A} &= [\mathbf{A}[:, J_s] \quad \mathbf{A}[:, J_r]] \mathbf{P}^\top \\ &= \mathbf{A}[:, J_s] \left[ \mathbf{I}_r \quad \mathbf{A}[:, J_s]^{-1} \mathbf{A}[:, J_r] \right] \mathbf{P}^\top, \\ &= \mathbf{C} \underbrace{[\mathbf{I}_r \quad \mathbf{C}^{-1} \mathbf{A}[:, J_r]]}_{\mathbf{W}} \mathbf{P}^\top \end{aligned}$$

where the matrix  $\mathbf{W}$  is given by  $[\mathbf{I}_r \quad \mathbf{C}^{-1} \mathbf{A}[:, J_r]] \mathbf{P}^\top = [\mathbf{I}_r \quad \mathbf{E}] \mathbf{P}^\top$  by Equation (7.1). To prove the claim that the magnitude of  $\mathbf{W}$  is no larger than 1 is equivalent to proving that entries in  $\mathbf{E} = \mathbf{C}^{-1} \mathbf{A}[:, J_r] \in \mathbb{R}^{r \times (n-r)}$  are no greater than 1 in absolute value.

Define the index vector  $[j_1, j_2, \dots, j_n]$  as a permutation of  $[1, 2, \dots, n]$  such that

$$[j_1, j_2, \dots, j_n] = [1, 2, \dots, n] \mathbf{P} = [J_s, J_r].^{\textcolor{blue}{1}}$$

Thus, it follows from  $\mathbf{C}\mathbf{E} = \mathbf{A}[:, J_r]$  that

$$\underbrace{[\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}] \mathbf{E}}_{=C=\mathbf{A}[:, J_s]} = \underbrace{[\mathbf{a}_{j_{r+1}}, \mathbf{a}_{j_{r+2}}, \dots, \mathbf{a}_{j_n}]}_{=\mathbf{A}[:, J_r]:=B},$$

where  $\mathbf{a}_i$  is the  $i$ -th column of  $\mathbf{A}$  and let  $\mathbf{B} = \mathbf{A}[:, J_r]$ . Therefore, by Cramer's rule in Equation (7.2), we have

$$\mathbf{E}_{kl} = \frac{\det(\mathbf{C}_B(k, l))}{\det(\mathbf{C})}, \quad (7.4)$$

where  $\mathbf{E}_{kl}$  is the entry  $(k, l)$  of  $\mathbf{E}$  and  $\mathbf{C}_B(k, l)$  is the  $r \times r$  matrix formed by replacing the  $k$ -th column of  $\mathbf{C}$  by the  $l$ -th column of  $\mathbf{B}$ . For example,

$$\begin{aligned} \mathbf{E}_{11} &= \frac{\det([\mathbf{a}_{j_{r+1}}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}, & \mathbf{E}_{12} &= \frac{\det([\mathbf{a}_{j_{r+2}}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}, \\ \mathbf{E}_{21} &= \frac{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_{r+1}}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}, & \mathbf{E}_{22} &= \frac{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_{r+2}}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}. \end{aligned}$$

Since  $J_s$  is chosen to maximize the magnitude of  $\det(\mathbf{C})$  in Equation (7.3), it follows that

$$|\mathbf{E}_{kl}| \leq 1, \quad \text{for all } k \in \{1, 2, \dots, r\}, l \in \{1, 2, \dots, n-r\}.$$

**Step 2: apply to general matrices** To summarize what we have proved above and to abuse the notation. For any matrix  $\mathbf{F} \in \mathbb{R}^{r \times n}$  with full rank  $r \leq n$ , the column ID exists that  $\mathbf{F} = \mathbf{C}_0 \mathbf{W}$  where the values in  $\mathbf{W}$  are no greater than 1 in absolute value.

Apply the finding to the full general matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r \leq \{m, n\}$ , it is trivial that the matrix  $\mathbf{A}$  admits a rank decomposition (Theorem 5.1, p. 164):

$$\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{D}} \underset{r \times n}{\mathbf{F}},$$

where  $\mathbf{D}, \mathbf{F}$  have full column rank  $r$  and full row rank  $r$  respectively. For the column ID of  $\mathbf{F} = \mathbf{C}_0 \mathbf{W}$  where  $\mathbf{C}_0 = \mathbf{F}[:, J_s]$  contains  $r$  linearly independent columns of  $\mathbf{F}$ . We notice by  $\mathbf{A} = \mathbf{DF}$  such that

$$\mathbf{A}[:, J_s] = \mathbf{DF}[:, J_s],$$

i.e., the columns indexed by  $J_s$  of  $(\mathbf{DF})$  can be obtained by  $\mathbf{DF}[:, J_s]$  which in turn are the columns of  $\mathbf{A}$  indexed by  $J_s$ . This makes

$$\underbrace{\mathbf{A}[:, J_s]}_C = \underbrace{\mathbf{DF}[:, J_s]}_{DC_0},$$

And

$$\mathbf{A} = \mathbf{DF} = \mathbf{DC}_0 \mathbf{W} = \underbrace{\mathbf{DF}[:, J_s]}_C \mathbf{W} = \mathbf{CW}.$$

---

<sup>1</sup>. Note here  $[j_1, j_2, \dots, j_n]$ ,  $[1, 2, \dots, n]$ ,  $J_s$ , and  $J_r$  are row vectors.

■

This completes the proof.

The above proof reveals an intuitive way to compute the optimal column ID of matrix  $\mathbf{A}$  as shown in Algorithm 29. However, any algorithm that is guaranteed to find such an optimally-conditioned factorization must have combinatorial complexity (Martinsson, 2019). In the next sections, we will consider alternative ways to find a relatively well-conditioned factorization.

---

**Algorithm 29** An *Intuitive* Method to Compute the Column ID

---

**Require:** Rank- $r$  matrix  $\mathbf{A}$  with size  $m \times n$ ;

- 1: Compute the rank decomposition  $\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{D}} \underset{r \times n}{\mathbf{F}}$  (Theorem 5.1, p. 164) such as from UTV (Section 4, p. 140);
- 2: Compute column ID of  $\mathbf{F}$ :  $\mathbf{F} = \mathbf{F}[:, J_s] \mathbf{W} = \tilde{\mathbf{C}} \mathbf{W}$ :

$$\begin{aligned} 2.1. & \left\{ \begin{array}{l} J_s = \arg \max_J \{ |\det(\mathbf{F}[:, J])| : J \text{ is a subset of } \{1, 2, \dots, n\} \text{ with size } r \}; \\ J_r = \{1, 2, \dots, n\}/J_s; \end{array} \right. \\ 2.2. & \left\{ \begin{array}{l} \tilde{\mathbf{C}} = \mathbf{F}[:, J_s]; \\ \mathbf{M} = \mathbf{F}[:, J_r]; \end{array} \right. \end{aligned}$$

2.3.  $\mathbf{FP} = \mathbf{F}[:, (J_s, J_r)]$  to obtain  $\mathbf{P}$ ;

$$2.4. \mathbf{E}_{kl} = \frac{\det(\tilde{\mathbf{C}}_{\mathbf{M}}(k, l))}{\det(\tilde{\mathbf{C}})}, \quad \text{for all } k \in [1, r], l \in [1, n - r] \quad (\text{Equation (7.4)});$$

2.5.  $\mathbf{W} = [\mathbf{I}_r, \mathbf{E}] \mathbf{P}^\top$  (Equation (7.1)).

- 3:  $\mathbf{C} = \mathbf{A}[:, J_s]$ ;

- 4: Output the column ID  $\mathbf{A} = \mathbf{CW}$ ;
- 

**Example 7.1 (Compute the Column ID)** For matrix

$$\mathbf{A} = \begin{bmatrix} 56 & 41 & 30 \\ 32 & 23 & 18 \\ 80 & 59 & 42 \end{bmatrix}$$

with rank 2. The trivial process for computing the column ID of  $\mathbf{A}$  is shown as follows. We first find a rank decomposition

$$\mathbf{A} = \mathbf{DF} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 56 & 41 & 30 \\ 32 & 23 & 18 \end{bmatrix}.$$

Since rank  $r = 2$ ,  $J_s$  is one of  $[1, 2]$ ,  $[0, 2]$ ,  $[0, 1]$  where the absolute determinant of  $\mathbf{F}[:, J_s]$  are 48, 48, 24 respectively. We proceed by choosing  $J_s = [0, 2]$ :

$$\tilde{\mathbf{C}} = \mathbf{F}[:, J_s] = \begin{bmatrix} 56 & 30 \\ 32 & 18 \end{bmatrix}, \quad \mathbf{M} = \mathbf{F}[:, J_r] = \begin{bmatrix} 41 \\ 23 \end{bmatrix}.$$

And

$$\mathbf{F}\mathbf{P} = \mathbf{F}[:, (J_s, J_r)] = \mathbf{F}[:, (0, 2, 1)] \quad \xrightarrow{\text{leads to}} \quad \mathbf{P} = \begin{bmatrix} 1 & & \\ & 1 & \\ & & 1 \end{bmatrix}.$$

In this example,  $\mathbf{E} \in \mathbb{R}^{2 \times 1}$ :

$$\begin{aligned} \mathbf{E}_{11} &= \det \begin{pmatrix} 41 & 30 \\ 23 & 18 \end{pmatrix} \Big/ \det \begin{pmatrix} 56 & 30 \\ 32 & 18 \end{pmatrix} = 1; \\ \mathbf{E}_{21} &= \det \begin{pmatrix} 56 & 41 \\ 32 & 23 \end{pmatrix} \Big/ \det \begin{pmatrix} 56 & 30 \\ 32 & 18 \end{pmatrix} = -\frac{1}{2}. \end{aligned}$$

This makes

$$\mathbf{E} = \begin{bmatrix} 1 \\ -\frac{1}{2} \end{bmatrix} \quad \xrightarrow{\text{leads to}} \quad \mathbf{W} = [\mathbf{I}_2, \mathbf{E}] \mathbf{P}^\top = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}.$$

The final selected columns are

$$\mathbf{C} = \mathbf{A}[:, J_s] = \begin{bmatrix} 56 & 30 \\ 32 & 18 \\ 80 & 42 \end{bmatrix}.$$

The net result is given by

$$\mathbf{A} = \mathbf{C}\mathbf{W} = \begin{bmatrix} 56 & 30 \\ 32 & 18 \\ 80 & 42 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix},$$

where entries of  $\mathbf{W}$  are no greater than 1 in absolute value as we want.  $\square$

To end up this section, we discuss where comes the non-uniqueness of the column ID.

### Remark 7.1: Non-uniqueness of the Column ID

In the above specific example 7.1, we notice the determinant for  $\mathbf{F}[:, (1, 2)]$  and  $\mathbf{F}[:, (0, 2)]$  both get maximal absolute determinant. Therefore, both of them can result in a column ID of  $\mathbf{A}$ . Whilst, we only select  $J_s$  from  $[1, 2], [0, 2], [0, 1]$ . When the  $J_s$  is fixed from the maximal absolute determinant search, any permute of it can also be selected, e.g.,  $J_s = [0, 2]$  or  $J_s = [2, 0]$  are both good. The two choice on the selection of the column index search yield the non-uniqueness of the column ID.

### 7.3. Row ID and Two-Sided ID

We term the decomposition above as column ID. This is no coincidence since it has its siblings:

**Theorem 7.1: (The Whole Interpolative Decomposition)**

Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be factored as

$$\begin{aligned} \text{Column ID: } \mathbf{A}_{m \times n} &= \boxed{\mathbf{C}}_{m \times r} \quad \boxed{\mathbf{W}}_{r \times n}; \\ \text{Row ID: } &= \boxed{\mathbf{Z}}_{m \times r} \quad \boxed{\mathbf{R}}_{r \times n}; \\ \text{Two-Sided ID: } &= \boxed{\mathbf{Z}}_{m \times r} \quad \boxed{\mathbf{U}}_{r \times r} \quad \boxed{\mathbf{W}}_{r \times n}, \end{aligned}$$

where

- $\mathbf{C} = \mathbf{A}[:, J_s] \in \mathbb{R}^{m \times r}$  is some  $r$  linearly independent columns of  $\mathbf{A}$ ,  $\mathbf{W} \in \mathbb{R}^{r \times n}$  is the matrix to reconstruct  $\mathbf{A}$  which contains an  $r \times r$  identity submatrix (under a mild column permutation):  $\mathbf{W}[:, J_s] = \mathbf{I}_r$ ;
- $\mathbf{R} = \mathbf{A}[I_s, :] \in \mathbb{R}^{r \times n}$  is some  $r$  linearly independent rows of  $\mathbf{R}$ ,  $\mathbf{Z} \in \mathbb{R}^{m \times r}$  is the matrix to reconstruct  $\mathbf{A}$  which contains an  $r \times r$  identity submatrix (under a mild row permutation):  $\mathbf{Z}[I_s, :] = \mathbf{I}_r$ ;
- Entries in  $\mathbf{W}, \mathbf{Z}$  have values no larger than 1 in magnitude:  $\max |w_{ij}| \leq 1$  and  $\max |z_{ij}| \leq 1$ ;
- $\mathbf{U} = \mathbf{A}[I_s, J_s] \in \mathbb{R}^{r \times r}$  is the nonsingular submatrix on the intersection of  $\mathbf{C}, \mathbf{R}$ ;
- The three matrices  $\mathbf{C}, \mathbf{R}, \mathbf{U}$  in the  $\boxed{\text{boxed}}$  texts share same notation as the skeleton decomposition (Theorem 6.1, p. 168) where they even have same meanings such that the three matrices make the skeleton decomposition of  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{C}\mathbf{U}^{-1}\mathbf{R}$ .

The proof of the row ID is just similar to that of the column ID. Suppose the column ID of  $\mathbf{A}^\top$  is given by  $\mathbf{A}^\top = \mathbf{C}_0 \mathbf{W}_0$  where  $\mathbf{C}_0$  contains  $r$  linearly independent columns of  $\mathbf{A}^\top$  (i.e.,  $r$  linearly independent rows of  $\mathbf{A}$ ). Let  $\mathbf{R} = \mathbf{C}_0, \mathbf{Z} = \mathbf{W}_0$ , the row ID is obtained by  $\mathbf{A} = \mathbf{Z}\mathbf{R}$ .

For the two-sided ID, recall from the skeleton decomposition (Theorem 6.1, p. 168). When  $\mathbf{U}$  is the intersection of  $\mathbf{C}, \mathbf{R}$ , it follows that  $\mathbf{A} = \mathbf{C}\mathbf{U}^{-1}\mathbf{R}$ . Thus  $\mathbf{C}\mathbf{U}^{-1} = \mathbf{Z}$  by the row ID. And this implies  $\mathbf{C} = \mathbf{Z}\mathbf{U}$ . By column ID, it follows that  $\mathbf{A} = \mathbf{C}\mathbf{W} = \mathbf{Z}\mathbf{U}\mathbf{W}$  which proves the existence of the two-sided ID.

**Data storage** For the data storage of each ID, we summarize as follows

- *Column ID.* It requires  $mr$  and  $(n - r)r$  floats to store  $\mathbf{C}$  and  $\mathbf{W}$  respectively, and  $r$  integers to store the indices of the selected columns in  $\mathbf{A}$ ;
- *Row ID.* It requires  $nr$  and  $(m - r)r$  floats to store  $\mathbf{R}$  and  $\mathbf{Z}$  respectively, and  $r$  integers to store the indices of the selected rows in  $\mathbf{A}$ ;
- *Two-Sided ID.* It requires  $(m - r)r$ ,  $(n - r)r$ , and  $r^2$  floats to store  $\mathbf{Z}, \mathbf{W}$ , and  $\mathbf{U}$  respectively. And extra  $2r$  integers are required to store the indices of the selected rows and columns in  $\mathbf{A}$ .

**Further reduction on the storage for two-sided ID for sparse matrix  $\mathbf{A}$**  Suppose the column ID of  $\mathbf{A} = \mathbf{C}\mathbf{W}$  where  $\mathbf{C} = \mathbf{A}[:, J_s]$  and a good spanning rows index  $I_s$  set of

$\mathbf{C}$  could be found:

$$\mathbf{A}[I_s, :] = \mathbf{C}[I_s, :] \mathbf{W}.$$

We observe that  $\mathbf{C}[I_s, :] = \mathbf{A}[I_s, J_s] \in \mathbb{R}^{r \times r}$  which is nonsingular (since full rank  $r$  in the sense of both row rank and column rank). It follows that

$$\mathbf{W} = (\mathbf{A}[I_s, J_s])^{-1} \mathbf{A}[I_s, :].$$

Therefore, there is no need to store the matrix  $\mathbf{W}$  explicitly. We only need to store  $\mathbf{A}[I_s, :]$  and  $(\mathbf{A}[I_s, J_s])^{-1}$ . Or when we can compute the inverse of  $\mathbf{A}[I_s, J_s]$  on the fly, it only requires  $r$  integers to store  $J_s$  and recover  $\mathbf{A}[I_s, J_s]$  from  $\mathbf{A}[I_s, :]$ . The storage of  $\mathbf{A}[I_s, :]$  is cheap if  $\mathbf{A}$  is sparse.

#### 7.4. Computing the Column ID via the CPQR

The method used in the proof of the last section can be utilized to compute the “optimal” column ID. However, any algorithm that is guaranteed to find such an optimally-conditioned factorization must have combinatorial complexity. An inexpensive alternative is to favor that  $\mathbf{W}$  is small in norm rather than bounding each entry in modulus by one. Recall the column-pivoted QR decomposition (CPQR, Theorem 3.1, p. 101) such that for matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , the *reduced* CPQR is given by:

$$\mathbf{AP} = \mathbf{Q}_r [\mathbf{R}_{11} \quad \mathbf{R}_{12}] = [\mathbf{Q}_r \mathbf{R}_{11} \quad \mathbf{Q}_r \mathbf{R}_{12}],$$

where  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is upper triangular,  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ ,  $\mathbf{Q}_r \in \mathbb{R}^{m \times r}$  has orthonormal columns, and  $\mathbf{P}$  is a permutation matrix. The complexity of the CRPQ is  $O(mnr)$  flops (Theorem 3.2, p. 103).  $\mathbf{AP}$  is permuting the  $r$  linearly independent columns into the first  $r$  columns of  $\mathbf{AP}$ :

$$\mathbf{AP} = \mathbf{A}[:, (J_s, J_r)] = [\mathbf{Q}_r \mathbf{R}_{11} \quad \mathbf{Q}_r \mathbf{R}_{12}].$$

In the “practical” CPQR via CGS<sup>2</sup> we introduced in Section 3.10.2 (p. 103),  $\mathbf{Q}_r \mathbf{R}_{11}$  contains the  $r$  independent columns of  $\mathbf{R}$  with the largest norm and  $\mathbf{Q}_r \mathbf{R}_{12}$  is small in norm. This is important to our aim that the column ID is *well-conditioned* in that entries of  $\mathbf{W}$  are small in magnitude. Therefore  $\mathbf{Q}_r \mathbf{R}_{11}$  contains  $r$  linearly independent columns of  $\mathbf{A}$ . Let  $\mathbf{C} = \mathbf{Q}_r \mathbf{R}_{11}$ , and solve the linear equation  $\mathbf{CE} = \mathbf{Q}_r \mathbf{R}_{12}$ , the column ID then follows:

$$\mathbf{A} = \mathbf{C} \underbrace{[\mathbf{I}_r, \mathbf{E}]\mathbf{P}^\top}_{\mathbf{W}}.$$

**Calculation of the linear system** The linear system  $\mathbf{CE} = \mathbf{Q}_r \mathbf{R}_{12}$  is well defined and is not in a sense of least squares as we have shown in Section 3.20.1 (p. 125), since every column of  $\mathbf{Q}_r \mathbf{R}_{12}$  is in the column space of  $\mathbf{C}$  (i.e., the column space of  $\mathbf{A}$ ). The solution can be obtained via the *normal equation*<sup>3</sup>:  $\mathbf{E} = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{Q}_r \mathbf{R}_{12}$ . An extra cost for computing the column ID thus comes from the calculation of the  $\mathbf{E}$ . The calculation

---

2. As a recap, CGS is short for the classical Gram-Schmidt process.

3. We shall briefly discuss in Definition 17.2 (p. 341).

of  $(\mathbf{C}^\top \mathbf{C})^{-1}$  takes  $r^2(2m - 1) + 2r^3$  flops where  $2r^3$  comes from the computation of the inverse of an  $r \times r$  matrix (Theorem 1.2, p. 47). Let's see what's left:

$$\text{step 2: } \mathbf{E} = \underbrace{(\mathbf{C}^\top \mathbf{C})^{-1}}_{r \times r} \quad \underbrace{\mathbf{C}^\top}_{r \times m} \quad \underbrace{\mathbf{Q}_r}_{m \times r} \quad \underbrace{\mathbf{R}_{12}}_{r \times (n-r)}.$$

Since  $r < m$ , the  $\mathbf{C}^\top \mathbf{Q}_r$  should be considered as the next step to make a smaller  $r \times r$  matrix which takes  $(2m - 1)r^2$  flops:

$$\text{step 3: } \mathbf{E} = \underbrace{(\mathbf{C}^\top \mathbf{C})^{-1}}_{r \times r} \quad \underbrace{\mathbf{C}^\top \mathbf{Q}_r}_{r \times r} \quad \underbrace{\mathbf{R}_{12}}_{r \times (n-r)}.$$

When  $r < (n - r)$ , the calculation of  $(\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{Q}_r$  in the above equation should be done firstly which makes the rest complexity be  $(2r - 1)r^2 + (2r - 1)r(n - r)$ . Otherwise,  $\mathbf{C}^\top \mathbf{Q}_r \mathbf{R}_{12}$  should be done firstly, which makes the rest complexity be  $2(2r - 1)r(n - r)$ . To conclude, the final complexity from normal equation is summarized as follows:

$$\text{cost} = \begin{cases} \{r^2(2m - 1) + 2r^3 + (2m - 1)r^2\} + (2r - 1)r^2 + (2r - 1)r(n - r), & r < (n - r); \\ \{r^2(2m - 1) + 2r^3 + (2m - 1)r^2\} + 2(2r - 1)r(n - r), & r \geq (n - r). \end{cases}$$

The normal equation is just a trivial way to solve the above linear equation. Other iterative methods can be employed such as the gradient descent (Section 17.6, p. 351).

### Partial factorization via the CPQR

The complexity of the algorithms for computing column ID of a matrix via the CPQR relies on the complexity of the CPQR decomposition. When the matrix has “exact” rank  $r$ , the complexity of the CPQR is  $O(mr^2)$  flops for matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . However, when it comes to the matrix  $\mathbf{A}$  being rank deficient, the complexity will become  $O(mn^2)$ . The partial factorization CPQR via MGS<sup>4</sup> (Section 3.10.4, p. 105) can be employed to attack this problem where the partial CPQR decomposition is given by

$$\mathbf{AP} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix},$$

where  $\mathbf{R}_{22}$  is small in norm. Such a partial factorization can either take a rank  $k$  set up front, or a tolerance  $\delta$  such that whenever  $r_{kk} < \delta$  in the upper triangular matrix  $\mathbf{R}$  of the CPQR decomposition, we stop. This is similar to what we will introduce in the low-rank column ID via the rank-revealing QR (RRQR) decomposition. But in the partial CPQR, the norm on  $\mathbf{R}_{22}$  is not guaranteed to be minimal which is assured in RRQR.

### 7.5. Low-Rank Column ID via the RRQR

An approximate rank  $\gamma$  ID of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the approximate factorization:

$$\begin{aligned} \mathbf{A} &= \underbrace{\mathbf{C}}_{m \times \gamma} \quad \underbrace{\mathbf{W}}_{\gamma \times n}; \\ \mathbf{AP} &= \quad \mathbf{C} \quad [\mathbf{I}, \mathbf{E}], \end{aligned}$$

---

<sup>4</sup> As a recap, MGS is short for the modified Gram-Schmidt process.

where the partial column skeleton  $\mathbf{C} \in \mathbb{R}^{m \times \gamma}$  is given by a subset of the columns of  $\mathbf{A}$ ,  $\gamma$  is known as the *numerical rank*, the entries of  $\mathbf{E}$  are known as the *expansion coefficients* as we have shown previously, and  $\mathbf{W}$  is well-conditioned in a sense that we will make precise shortly.

The low-rank approximation of column ID has been studied extensively in the context of rank-revealing QR decomposition (Voronin and Martinsson, 2017; Martinsson, 2019; Martinsson and Tropp; Halko et al., 2011). By rank-revealing QR (RRQR) (Equation (3.11), p. 107), there exists a permutation  $\mathbf{P}$  such that the linearly independent columns of  $\mathbf{A}$  can be permuted in the left and

$$\begin{aligned}\mathbf{AP} &= \mathbf{QR} = [\mathbf{Q}_1 \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{L} & \mathbf{M} \\ \mathbf{0} & \mathbf{N} \end{bmatrix} \\ &= [\mathbf{Q}_1 \mathbf{L} \quad \mathbf{Q}_1 \mathbf{M} + \mathbf{Q}_2 \mathbf{N}] \\ &= \mathbf{Q}_1 \mathbf{L} [\mathbf{I}_\gamma \quad \mathbf{Y}],\end{aligned}$$

where  $\mathbf{N} \in \mathbb{R}^{(n-\gamma) \times (n-\gamma)}$  and  $\|\mathbf{N}\|$  is small in some norm. And  $\mathbf{Y}$  is defined to be the solution of the linear system  $(\mathbf{Q}_1 \mathbf{L}) \mathbf{Y} = (\mathbf{Q}_1 \mathbf{M} + \mathbf{Q}_2 \mathbf{N})$ . We observe that  $\mathbf{Q}_1 \mathbf{L}$  is the first  $\gamma$  columns of  $\mathbf{AP}$ . Let  $\mathbf{C} = \mathbf{Q}_1 \mathbf{L}$ ,  $\mathbf{E} = \mathbf{Y}$ , then the low-rank column ID approximation can be found by solving the linear system

$$(\mathbf{Q}_1 \mathbf{L}) \mathbf{E} = (\mathbf{Q}_1 \mathbf{M} + \mathbf{Q}_2 \mathbf{N}). \quad (7.5)$$

Since  $\mathbf{N}$  is small in norm, it can be approximated to find the solution of

$$\frac{\mathbf{L}}{\gamma \times \gamma} \frac{\mathbf{E}}{\gamma \times (n-\gamma)} = \frac{\mathbf{M}}{(n-\gamma) \times (n-\gamma)}.$$

The problem is well defined since  $\mathbf{L}$  is nonsingular (upper triangular with nonzero diagonals). If  $\mathbf{L}$  is singular, then one can show that  $\mathbf{A}$  must necessarily have rank  $\gamma'$  less than  $\gamma$ , and the bottom  $\gamma - \gamma'$  rows in the above linear system consist of all zeros, so there exists a solution in this case as well.

The approximation error of the column ID obtained via RRQR with pivoting is the same as that of the RRQR:

$$\|\mathbf{A} - \mathbf{CW}\| = \|\mathbf{Q}_2 \mathbf{N}\|,$$

where  $\mathbf{W} = [\mathbf{I}_\gamma, \mathbf{E}] \mathbf{P}^\top$ .

### Remark 7.1: Condition of the Linear System

Unfortunately,  $\mathbf{L}$  in the linear Equation (7.5) is typically quite ill-conditioned. However, the linear system in Equation (7.5) still has a solution  $\mathbf{E}$  whose entries are of moderate size. Informally, the directions where  $\mathbf{L}$  and  $\mathbf{M}$  point in are “lined up” (Cheng et al., 2005). We shall not give the details.

## 7.6. Computing the ID via Randomized Algorithm

Suppose matrix  $\mathbf{A}$  admits the rank decomposition:

$$\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{D}} \underset{r \times n}{\mathbf{F}}.$$

Upon finding the rank decomposition, suppose further that the row ID of  $\mathbf{D}$  is obtained by

$$\text{row ID of } \mathbf{D}: \quad \underset{m \times r}{\mathbf{D}} = \underset{m \times r}{\mathbf{Z}} \underset{r \times r}{\mathbf{R}_0} = \mathbf{Z}\mathbf{D}[I_s, :].$$

Similar to the proof of the column ID in Section 7.2, we observe that  $Z\mathbf{A}[I_s, :]$  is automatically a row ID of  $\mathbf{A}$ :  $\mathbf{A} = Z\mathbf{A}[I_s, :]$ . This can be shown as follows:

$$\begin{aligned} Z\mathbf{A}[I_s, :] &= Z(\mathbf{D}[I_s, :]\mathbf{F}) && (\text{since } \mathbf{A} = \mathbf{DF}) \\ &= \mathbf{DF} && (\text{since } \mathbf{D} = Z\mathbf{D}[I_s, :]) \\ &= \mathbf{A}. \end{aligned} \tag{7.6}$$

We notice that  $\mathbf{D}$  spans the same column space as  $\mathbf{A}$ :  $\mathcal{C}(\mathbf{D}) = \mathcal{C}(\mathbf{A})$ . The finding above tells us that as long as we find an  $m \times r$  matrix  $\mathbf{D}$  spanning the same column space of  $\mathbf{A}$ , a row ID can be applied to  $\mathbf{D}$  to find the row ID of  $\mathbf{A}$ . This reveals the randomized algorithm for computing row ID of  $\mathbf{A}$ . To show this, we need a few facts that will be useful.

### Lemma 7.1: (Subspace of $\mathbf{A}^\top \mathbf{A}$ and $\mathbf{A} \mathbf{A}^\top$ )

For matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , we have

- The column space of  $\mathbf{A}^\top \mathbf{A}$  is equal to the column space of  $\mathbf{A}^\top$  (i.e., row space of  $\mathbf{A}$ ):  $\mathcal{C}(\mathbf{A}^\top \mathbf{A}) = \mathcal{C}(\mathbf{A}^\top)$ ;
- The column space of  $\mathbf{A} \mathbf{A}^\top$  is equal to the column space of  $\mathbf{A}$ :  $\mathcal{C}(\mathbf{A} \mathbf{A}^\top) = \mathcal{C}(\mathbf{A})$ .

**Proof** [of Lemma 7.1] Let  $\mathbf{x} \in \mathcal{N}(\mathbf{A})$ , we have

$$\mathbf{Ax} = \mathbf{0} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}^\top \mathbf{Ax} = \mathbf{0},$$

i.e.,  $\mathbf{x} \in \mathcal{N}(\mathbf{A}) \xrightarrow{\text{leads to}} \mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ , therefore  $\mathcal{N}(\mathbf{A}) \subseteq \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ . Further, let  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ , we have

$$\mathbf{A}^\top \mathbf{Ax} = \mathbf{0} \xrightarrow{\text{leads to}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} = 0 \xrightarrow{\text{leads to}} \|\mathbf{Ax}\|^2 = 0 \xrightarrow{\text{leads to}} \mathbf{Ax} = \mathbf{0},$$

i.e.,  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A}) \xrightarrow{\text{leads to}} \mathbf{x} \in \mathcal{N}(\mathbf{A})$ , therefore  $\mathcal{N}(\mathbf{A}^\top \mathbf{A}) \subseteq \mathcal{N}(\mathbf{A})$ .

As a result, by “sandwiching”, it follows that

$$\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^\top \mathbf{A}).$$

By the fundamental theorem of linear algebra in Appendix B (p. 427),

$$\mathcal{C}(\mathbf{A}^\top) = \mathcal{C}(\mathbf{A}^\top \mathbf{A}).$$

The second half of the lemma is just the same when applying the process on  $A^+$ .

The above lemma tells us that if a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r$ , a matrix  $\mathbf{B} \in \mathbb{R}^{m \times k}$  with  $k > r$  and  $\mathcal{C}(\mathbf{B}) = \mathcal{C}(\mathbf{A})$ . Then a moment of reflexion reveals  $\mathbf{A}\mathbf{A}^\top\mathbf{B}$  spans the same column space as  $\mathbf{A}$ :

$$\mathcal{C}(AA^\top B) = \mathcal{C}(A), \quad \text{if} \quad \mathcal{C}(B) = \mathcal{C}(A). \quad (7.7)$$

Further, by Lemma 27.1 (p. 426), for any matrix  $A \in \mathbb{R}^{m \times n}$ , suppose that  $\{g_1, g_2, \dots, g_r\}$  is a set of vectors in  $\mathbb{R}^n$  which forms a basis for the row space, then  $\{Ag_1, Ag_2, \dots, Ag_r\}$  is a basis for the column space of  $A$ :

$$\mathcal{C}(AG) = \mathcal{C}(A). \quad (7.8)$$

Then, the above matrix  $\mathbf{B}$  in Equation (7.7) can be constructed by  $\mathbf{AG}$  where the columns of  $\mathbf{G}$  contain row basis of  $\mathbf{A}$ :

$$\mathcal{C}(AA^\top AG) = \mathcal{C}(A). \quad (7.9)$$

**Algorithm 30** A Randomized Method to Compute the Row ID

**Require:** Rank- $r$  matrix  $A$  with size  $m \times n$ ;

- 1: Decide the over-sampling parameter  $k$  (e.g.,  $k = 10$ )  $\rightarrow$  let  $z = r + k$ ;
  - 2: Decide the iteration number:  $\eta$  (e.g.,  $\eta = 0, 1$  or  $2$ );
  - 3: Generate  $r + k$  Gaussian random vectors in  $\mathbb{R}^n$  into columns of matrix  $\mathbf{G} \in \mathbb{R}^{n \times (r+k)}$ ;  
 ▷ i.e., probably contain the row basis of  $\mathbf{A}$
  - 4: Initialize  $\mathbf{D} = \mathbf{AG} \in \mathbb{R}^{m \times (r+k)}$ ;  $\quad \quad \quad$  ▷ i.e., probably  $\mathcal{C}(\mathbf{D}) = \mathcal{C}(\mathbf{A})$ ,  $(2n - 1)mz$  flops
  - 5: **for**  $i = 1$  to  $\eta$  **do**
  - 6:    $\mathbf{D} = \mathbf{AA}^\top \mathbf{D}$ ;  $\quad \quad \quad$  ▷  $(2m - 1)nz + (2n - 1)mz$  flops
  - 7: **end for**
  - 8: Calc. the row ID of small matrix  $\mathbf{D}$ :  $\mathbf{D} = \mathbf{ZR}_0 = \mathbf{ZD}[I_s, :]$ ; ▷  $O(mz^2)$  flops by CPQR
  - 9: Output row ID of  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{ZA}[I_s, :]$ ;

Normally, we could simply stop at  $\mathbf{AG}$  since  $\mathcal{C}(\mathbf{AG}) = \mathcal{C}(\mathbf{A})$ , and the row ID of  $\mathbf{AG}$  would reveal the row ID of  $\mathbf{A}$  by Equation (7.6). Computing the row ID of  $\mathbf{AG}$  is a relatively simpler task compared to that of  $\mathbf{A}$  directly since  $\mathbf{AG} \in \mathbb{R}^{m \times r}$  and  $r \ll n$ .

However, in some situations, when matrix  $\mathbf{A}$  is not exactly rank- $r$ , i.e., some singular values  $\sigma_k$  (with  $k > r$ ) of  $\mathbf{A}$  may be small enough to be regarded as zero (truncated). We will introduce the SVD in Section 14 (p. 264) such that any matrix  $\mathbf{A}$  admits factorization  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$  where  $\mathbf{U}, \mathbf{V}$  are orthogonal, and roughly speaking  $\Sigma$  is a diagonal matrix containing singular values with the number of nonzero singular values being the rank of  $\mathbf{A}$ . The columns of  $\mathbf{U}$  contain the column basis of  $\mathbf{A}$  and the columns of  $\mathbf{V}$  contain the row basis of  $\mathbf{A}$  (Lemma 14.3.1, p. 269). Therefore,  $\mathbf{A}\mathbf{A}^\top\mathbf{A}\mathbf{G}$  results in

$$AA^\top AG = (U\Sigma V^\top)(V\Sigma U^\top)(U\Sigma V^\top)G = U\Sigma^3 V^\top G.$$

where  $\Sigma^3$  will **shrink the small singular values towards 0**. This is reasonable in that they do not count in the *numerical rank*. For matrices whose singular values decay slowly, we should consider the process twice or third times:

$$AA^\top(AA^\top AG) = U\Sigma^5V^\top G.$$

The procedure to compute the row ID of a matrix is shown in Algorithm 30 where a parameter  $k$  is used to assure that there is a high probability that  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_r, \mathbf{g}_{r+1}, \dots, \mathbf{g}_{r+k}]$  contains the row basis of  $\mathbf{A}$  (The choice  $k = 10$  is often good) and each  $\mathbf{g}_i$  are generated by *random Gaussian vector*. Iteration parameter  $\eta$  is usually picked as 1 or 2 to improve accuracy (to shrink the small singular values). The complexity of the algorithm is shown in the comment of each step which makes it

$$\boxed{O(mn(r+k))} \quad \text{flops.} \quad (7.10)$$

### Remark 7.2: A Word on the Source of Row Basis

In step 4 of Algorithm 30, we desire  $\mathbf{D}$  contains linearly independent vectors that can span the column space of  $\mathbf{A}$ , i.e., find a column basis. We just show one way that transform row basis matrix  $\mathbf{G}$  to the column basis matrix  $\mathbf{D} = \mathbf{AG}$ , and this is known as the *random projection method*. The matrix  $\mathbf{G}$  formed by these columns is expected to be very close to  $\mathbf{A}$  in a sense that the basis of the range of  $\mathbf{D}$  covers the range of  $\mathbf{A}$  well. The probability of failure is negligible. However, further question can be posed, what if  $\mathbf{G}$  does not contain enough row basis vectors? The column basis matrix  $\mathbf{D}$  can be chosen in different ways: by *subsampling of the input matrix* directly (in the sense of row basis or column basis that we will show in next paragraph). The orthonormal basis consisting of  $r$  linearly independent vectors can be obtained using exact methods since the size of  $\mathbf{D}$  or  $\mathbf{G}$  is very small. These techniques are relatively insensitive to the quality of randomness and produce high accurate results.

**Subsampling method for constructing basis** We have shown in Gram-Schmidt process (Section 3.2, p. 82), the projection of a vector  $\mathbf{a}$  onto  $\mathbf{b}$  results in the projection  $\hat{\mathbf{a}} = \frac{\mathbf{a}^\top \mathbf{b}}{\mathbf{b}^\top \mathbf{b}} \mathbf{b}$  which is in the direction of  $\mathbf{b}$  since it is a multiple of  $\mathbf{b}$ . The component along  $\mathbf{b}$  thus can be obtained by  $\mathbf{a} - \hat{\mathbf{a}} = \mathbf{a} - \frac{\mathbf{a}^\top \mathbf{b}}{\mathbf{b}^\top \mathbf{b}} \mathbf{b}$  which is  $\mathbf{0}$  if  $\mathbf{a}$  and  $\mathbf{b}$  are dependent (i.e., in the same direction). When  $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ , the complexity is  $O(n)$  flops. Therefore, when we generate the  $\mathbf{g}_i$ 's vector in Algorithm 30, we can project it onto each row of  $\mathbf{A}$  to check it is in the row space of  $\mathbf{A}$ , and this costs  $O(mn)$  flops. And there are  $\sim (r+k)$  such generation, which makes it  $\boxed{O(mn(r+k))}$  flops to assure  $\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_{r+k}\}$  can span the row space of  $\mathbf{A}$ . This is the same order of cost compared to the total cost of the algorithm in Equation (7.10) when  $n \leq m$ . Analogously, we can directly generate the columns of  $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_{r+k}]$  via Gaussian random vector and check if each  $\mathbf{d}_i$  is in the column space of  $\mathbf{A}$ . This costs  $\boxed{O(mn(r+k))}$  that again is equal to the cost of the algorithm in Equation (7.10). So the complexity of the algorithm would not blow up.

**Compute the column ID by randomized method** Analogously, the above procedure on row ID can be easily applied to compute the column ID of matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  by calculating the row ID of  $\mathbf{A}^\top$ . Or from the proof of the column ID. If  $\mathbf{A}$  admits a rank decomposition (Theorem 5.1, p. 164)

$$\mathbf{A}_{m \times n} = \mathbf{D}_{m \times r} \mathbf{F}_{r \times n},$$

**Algorithm 31** A Randomized Method to Compute the Column ID

**Require:** Rank- $r$  matrix  $A$  with size  $m \times n$ ;

- 1: Decide the over-sampling parameter  $k$  (e.g.,  $k = 10$ )  $\rightarrow$  let  $z = r + k$ ;
  - 2: Decide the iteration number:  $\eta$  (e.g.,  $\eta = 0, 1$  or  $2$ );
  - 3: Generate  $r + k$  Gaussian random vectors in  $\mathbb{R}^m$  into columns of matrix  $\mathbf{G} \in \mathbb{R}^{m \times (r+k)}$ ;  
 $\triangleright$  i.e., probably contain the column basis of  $\mathbf{A}$
  - 4: Initialize  $\mathbf{F} = \mathbf{A}^\top \mathbf{G} \in \mathbb{R}^{n \times (r+k)}$ ;  $\triangleright$  i.e., probably  $\mathcal{C}(\mathbf{F}) = \mathcal{C}(\mathbf{A}^\top)$ ,  $(2m - 1)nz$  flops
  - 5: **for**  $i = 1$  to  $\eta$  **do**
  - 6:    $\mathbf{F} = \mathbf{A}^\top \mathbf{A}\mathbf{F}$ ;  $\triangleright (2m - 1)nz + (2n - 1)mz$  flops
  - 7: **end for**
  - 8: Calc. the **column** ID of small  $\mathbf{F}$ :  $\mathbf{F} = \mathbf{C}_0 \mathbf{W} = \mathbf{F}[:, J_s] \mathbf{W}$ ;  $\triangleright O(mz^2)$  flops by CPQR
  - 9: Output column ID of  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{A}[:, J_s] \mathbf{W}$ ;

where  $\mathbf{D}, \mathbf{F}$  have full column rank  $r$  and full row rank  $r$  respectively. Upon finding the rank decomposition, suppose further that the column ID of  $\mathbf{F}$  is obtained by

$$\text{column ID of } \mathbf{F}: \quad \begin{matrix} \mathbf{F} \\ r \times n \end{matrix} = \begin{matrix} \mathbf{C}_0 \\ r \times r \end{matrix} \quad \begin{matrix} \mathbf{W} \\ r \times n \end{matrix} = \mathbf{F}[:, J_s] \mathbf{W}.$$

We observe that  $\mathbf{A}[:, J_s]\mathbf{W}$  is automatically a column ID of  $\mathbf{A}$ :  $\mathbf{A} = \mathbf{A}[:, J_s]\mathbf{W}$ . This can be shown as follows:

$$\begin{aligned}
 \mathbf{A}[:, J_s] \mathbf{W} &= (\mathbf{D}\mathbf{F}[:, J_s]) \mathbf{W} && \text{(since } \mathbf{A} = \mathbf{D}\mathbf{F} \text{)} \\
 &= \mathbf{D}\mathbf{F} && \text{(since } \mathbf{F} = \mathbf{F}[:, J_s]\mathbf{W} \text{)} \\
 &= \mathbf{A}.
 \end{aligned} \tag{7.11}$$

We notice that  $\mathbf{F}$  spans the same row space as  $\mathbf{A}$ :  $\mathcal{C}(\mathbf{F}^\top) = \mathcal{C}(\mathbf{A}^\top)$ . The finding above tells us that as long as we find an  $r \times n$  matrix  $\mathbf{D}$  spanning the same row space of  $\mathbf{A}$ , a column ID can be applied on  $\mathbf{F}$  to find the column ID of  $\mathbf{A}$ . This reveals the randomized algorithm for computing column ID of  $\mathbf{A}$ . This is actually the second part of the proof of column ID in Section 7.2.

We may further notice that suppose columns of  $\mathbf{G}$  contain the column basis of  $\mathbf{A}$ ,  $\mathbf{A}^\top \mathbf{G}$  will contain row basis of  $\mathbf{A}$  by Lemma 27.1 (p. 426) again.  $\mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{G}$  will span the row space of  $\mathbf{A}$  where  $\mathbf{A}^\top \mathbf{A}$  on the left can help shrink small singular values:

$$\mathbf{A}^\top \mathbf{A} \mathbf{A}^\top \mathbf{G} = (\mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^\top) (\mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^\top) (\mathbf{V} \boldsymbol{\Sigma} \mathbf{U}^\top) \mathbf{G} = \mathbf{V} \boldsymbol{\Sigma}^3 \mathbf{U}^\top \mathbf{G}.$$

The randomized procedure to compute column ID is formulated in Algorithm 31 where the difference compared to row ID is made in blue texts.

# Part IV

## Reduction to Hessenberg, Tridiagonal, and Bidiagonal Form



## Introduction

In real applications, we often want to factor matrix  $\mathbf{A}$  into two orthogonal matrices  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$  where  $\Lambda$  is diagonal or upper triangular, e.g., eigen analysis via the Schur decomposition and principal component analysis (PCA) via the spectral decomposition. This can be computed via a sequence of *orthogonal similarity transformations*:

$$\underbrace{\mathbf{Q}_k^\top \dots \mathbf{Q}_2^\top \mathbf{Q}_1^\top}_{\mathbf{Q}^\top} \mathbf{A} \underbrace{\mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_k}_{\mathbf{Q}}$$

which converges to  $\Lambda$ . However, this will always be very hard to handle in practice, for example, via the Householder reflectors. Following from the QR decomposition via the Householder example,<sup>5</sup> the sequence of orthogonal similarity transformation can be constructed via the Householder reflectors:

$$\begin{array}{ccc} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\mathbf{H}_1 \times} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\times \mathbf{H}_1^\top} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \end{array} \right], \\ \mathbf{A} & & \mathbf{H}_1 \mathbf{A} & & \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \end{array}$$

where the left Householder introduces zeros in the first column below the main diagonal (see Section 3.13), and unfortunately the right Householder will destroy the zeros introduced by the left Householder.

However, if we are less ambitious to modify the algorithms into two phases, where the first phase transforms into a Hessenberg matrix (Definition 8.1, p. 197) or a tridiagonal matrix (Definition 9.1, p. 209). And if we find a second phase algorithm to transform the results from the first one to the goal we want to find, then we complete the algorithm:

$$\begin{array}{ccc} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\mathbf{H}_1 \times} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\times \mathbf{H}_1^\top} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \end{array} \right], \\ \mathbf{A} & & \mathbf{H}_1 \mathbf{A} & & \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \end{array}$$

where the left Householder will not influence the first row, and the right Householder will not influence the first column. A phase 2<sup>6</sup> algorithm to find the triangular matrix is shown as follows:

$$\begin{array}{ccc} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \mathbf{0} & \square & \square & \square & \square \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \square & \square & \square \end{array} \right] & \xrightarrow{\text{Phase 2}} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square & \square & \square \\ \mathbf{0} & \mathbf{0} & \square & \square & \square & \square \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \square & \square & \square \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \square & \square \end{array} \right] \\ \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \mathbf{H}_3^\top & & \Lambda \end{array}$$

<sup>5</sup>. where  $\square$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

<sup>6</sup>. which is usually an iterative algorithm.

From the discussion above, to compute the spectral decomposition, Schur decomposition, or singular value decomposition (SVD), we usually reach a compromise to calculate the Hessenberg, tridiagonal, or bidiagonal form in the first phase and leave the second phase to finish the rest (Van Zee et al., 2012, 2014; Trefethen and Bau III, 1997).

## Chapter 8

# Hessenberg Decomposition

### Contents

---

|     |                                                                              |     |
|-----|------------------------------------------------------------------------------|-----|
| 8.1 | Hessenberg Decomposition . . . . .                                           | 197 |
| 8.2 | Similarity Transformation and Orthogonal Similarity Transformation . . . . . | 198 |
| 8.3 | Existence of the Hessenberg Decomposition . . . . .                          | 199 |
| 8.4 | Computing the Hessenberg Decomposition . . . . .                             | 202 |
| 8.5 | Properties of the Hessenberg Decomposition . . . . .                         | 205 |

---

## 8.1. Hessenberg Decomposition

We firstly give the rigorous definition of the upper Hessenberg matrix.

### Definition 8.1: Upper Hessenberg Matrix

An *upper Hessenberg matrix* is a square matrix where all the entries below the first diagonal (i.e., the ones below the *main diagonal*) (a.k.a., *lower subdiagonal*) are zeros. Similarly, a lower Hessenberg matrix is a square matrix where all the entries above the first diagonal (i.e., the ones above the main diagonal) are zeros.

The definition of the upper Hessenberg can also be extended to rectangular matrices, and the form can be implied from the context.

In matrix language, for any matrix  $\mathbf{H} \in \mathbb{R}^{n \times n}$ , and the entry  $(i, j)$  denoted by  $h_{ij}$  for all  $i, j \in \{1, 2, \dots, n\}$ . Then  $\mathbf{H}$  with  $h_{ij} = 0$  for all  $i \geq j + 2$  is known as an Hessenberg matrix.

Let  $i$  denote the smallest positive integer for which  $h_{i+1,i} = 0$  where  $i \in \{1, 2, \dots, n-1\}$ , then  $\mathbf{H}$  is *unreduced* if  $i = n - 1$ .

Take a  $5 \times 5$  matrix as an example, the lower triangular below the lower sub-diagonal are zero in the upper Hessenberg matrix:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \quad \text{or} \quad \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] . \\ \text{possibly unreduced} \qquad \qquad \qquad \text{reduced} \end{array}$$

Then we have the following Hessenberg decomposition:

### Theorem 8.2: (Hessenberg Decomposition)

Every  $n \times n$  square matrix  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{Q} \mathbf{H} \mathbf{Q}^\top \quad \text{or} \quad \mathbf{H} = \mathbf{Q}^\top \mathbf{A} \mathbf{Q},$$

where  $\mathbf{H}$  is an upper Hessenberg matrix, and  $\mathbf{Q}$  is an orthogonal matrix.

It's not hard to find that a lower Hessenberg decomposition of  $\mathbf{A}^\top$  is given by  $\mathbf{A}^\top = \mathbf{Q} \mathbf{H}^\top \mathbf{Q}^\top$  if  $\mathbf{A}$  has the Hessenberg decomposition  $\mathbf{A} = \mathbf{Q} \mathbf{H} \mathbf{Q}^\top$ . The Hessenberg decomposition shares a similar form as the QR decomposition in that they both reduce a matrix into a sparse form where the lower parts of both are zero.

### Remark 8.3: Why Hessenberg Decomposition

We will see that the zeros introduced into  $\mathbf{H}$  from  $\mathbf{A}$  is accomplished by the left orthogonal matrix  $\mathbf{Q}$  (same as the QR decomposition) and the right orthogonal matrix

$\mathbf{Q}^\top$  here does not transform the matrix into any better or simple form. Then why do we want the Hessenberg decomposition rather than just a QR decomposition which has a simpler structure in that it even has zeros in the lower sub-diagonal? The answer is given in the previous section that the Hessenberg decomposition is usually used by other algorithms as a phase 1 step to find a decomposition that factor the matrix into two orthogonal matrices, e.g., SVD, UTV, and so on. And if we employ an aggressive algorithm that even favors zeros in the lower sub-diagonal (again, as in the QR decomposition), the right orthogonal transform  $\mathbf{Q}^\top$  will destroy the zeros that can be seen very shortly.

On the other hand, the form  $\mathbf{A} = \mathbf{Q}\mathbf{H}\mathbf{Q}^\top$  on  $\mathbf{H}$  is known as the *orthogonal similarity transformation* (Definition 8.1, p. 198) on  $\mathbf{A}$  such that the eigenvalues, rank and trace of  $\mathbf{A}$  and  $\mathbf{H}$  are the same (Lemma 8.2, p. 198). Then if we want to study the properties of  $\mathbf{A}$ , exploration on  $\mathbf{H}$  can be a relatively simpler task than that on the original matrix  $\mathbf{A}$ .

## 8.2. Similarity Transformation and Orthogonal Similarity Transformation

As mentioned previously, the Hessenberg decomposition introduced in this section, the tridiagonal decomposition in the next section, the Schur decomposition (Theorem 12.1, p. 236), and the spectral decomposition (Theorem 13.1, p. 241) share a similar form that transforms the matrix into a similar matrix. We now give the rigorous definition of similar matrices and similarity transformations.

### Definition 8.1: Similar Matrices and Similarity Transformation

$\mathbf{A}$  and  $\mathbf{B}$  are called *similar matrices* if there exists a nonsingular matrix  $\mathbf{P}$  such that  $\mathbf{B} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}$ .

In words, for any nonsingular matrix  $\mathbf{P}$ , the matrices  $\mathbf{A}$  and  $\mathbf{P}\mathbf{A}\mathbf{P}^{-1}$  are similar matrices. And in this sense, given the nonsingular matrix  $\mathbf{P}$ ,  $\mathbf{P}\mathbf{A}\mathbf{P}^{-1}$  is called a *similarity transformation* applied to matrix  $\mathbf{A}$ .

Moreover, when  $\mathbf{P}$  is orthogonal, then  $\mathbf{P}\mathbf{A}\mathbf{P}^\top$  is also known as the *orthogonal similarity transformation* of  $\mathbf{A}$ . The orthogonal similarity transformation is important in the sense that the condition number of the transformed matrix  $\mathbf{P}\mathbf{A}\mathbf{P}^\top$  is not worse than that of the original matrix  $\mathbf{A}$ .

The difference between the similarity transformation and orthogonal similarity transformation is partly explained in the sense of coordinate transformation (Section 16, p. 334). Now we prove the important properties of similar matrices in the following lemma.

### Lemma 8.2: (Eigenvalue, Trace and Rank of Similar Matrices)

Any eigenvalue of  $\mathbf{A}$  is also an eigenvalue of  $\mathbf{PAP}^{-1}$ . The converse is also true that any eigenvalue of  $\mathbf{PAP}^{-1}$  is also an eigenvalue of  $\mathbf{A}$ . I.e.,  $\Lambda(\mathbf{A}) = \Lambda(\mathbf{B})$ , where  $\Lambda(\mathbf{X})$  is the spectrum of matrix  $\mathbf{X}$  (Definition 0.3, p. 16).

And also the trace and rank of  $\mathbf{A}$  are equal to those of matrix  $\mathbf{PAP}^{-1}$  for any nonsingular matrix  $\mathbf{P}$ .

**Proof** [of Lemma 8.2] For any eigenvalue  $\lambda$  of  $\mathbf{A}$ , we have  $\mathbf{Ax} = \lambda\mathbf{x}$ . Then  $\lambda\mathbf{Px} = \mathbf{PAP}^{-1}\mathbf{Px}$  such that  $\mathbf{Px}$  is an eigenvector of  $\mathbf{PAP}^{-1}$  corresponding to  $\lambda$ .

Similarly, for any eigenvalue  $\lambda$  of  $\mathbf{PAP}^{-1}$ , we have  $\mathbf{PAP}^{-1}\mathbf{x} = \lambda\mathbf{x}$ . Then  $\mathbf{AP}^{-1}\mathbf{x} = \lambda\mathbf{P}^{-1}\mathbf{x}$  such that  $\mathbf{P}^{-1}\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  corresponding to  $\lambda$ .

For the trace of  $\mathbf{PAP}^{-1}$ , we have  $\text{trace}(\mathbf{PAP}^{-1}) = \text{trace}(\mathbf{AP}^{-1}\mathbf{P}) = \text{trace}(\mathbf{A})$ , where the first equality comes from the fact that trace of a product is invariant under cyclical permutations of the factors:

$$\text{trace}(\mathbf{ABC}) = \text{trace}(\mathbf{BCA}) = \text{trace}(\mathbf{CAB}),$$

if all  $\mathbf{ABC}$ ,  $\mathbf{BCA}$ , and  $\mathbf{CAB}$  exist.

For the rank of  $\mathbf{PAP}^{-1}$ , we separate it into two claims as follows.

**Rank claim 1:  $\text{rank}(\mathbf{ZA}) = \text{rank}(\mathbf{A})$  if  $\mathbf{Z}$  is nonsingular** We will first show that  $\text{rank}(\mathbf{ZA}) = \text{rank}(\mathbf{A})$  if  $\mathbf{Z}$  is nonsingular. For any vector  $\mathbf{n}$  in the null space of  $\mathbf{A}$ , that is  $\mathbf{An} = \mathbf{0}$ . Thus,  $\mathbf{ZAn} = \mathbf{0}$ , that is,  $\mathbf{n}$  is also in the null space of  $\mathbf{ZA}$ . And this implies  $\mathcal{N}(\mathbf{A}) \subseteq \mathcal{N}(\mathbf{ZA})$ .

Conversely, for any vector  $\mathbf{m}$  in the null space of  $\mathbf{ZA}$ , that is  $\mathbf{ZAm} = \mathbf{0}$ , we have  $\mathbf{Am} = \mathbf{Z}^{-1}\mathbf{0} = \mathbf{0}$ . That is,  $\mathbf{m}$  is also in the null space of  $\mathbf{A}$ . And this indicates  $\mathcal{N}(\mathbf{ZA}) \subseteq \mathcal{N}(\mathbf{A})$ .

By “sandwiching”, the above two arguments imply

$$\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{ZA}) \quad \longrightarrow \quad \text{rank}(\mathbf{ZA}) = \text{rank}(\mathbf{A}).$$

**Rank claim 2:  $\text{rank}(\mathbf{AZ}) = \text{rank}(\mathbf{A})$  if  $\mathbf{Z}$  is nonsingular** We notice that the row rank is equal to the column rank of any matrix (Corollary 4.2, p. 148). Then  $\text{rank}(\mathbf{AZ}) = \text{rank}(\mathbf{Z}^\top \mathbf{A}^\top)$ . Since  $\mathbf{Z}^\top$  is nonsingular, by claim 1, we have  $\text{rank}(\mathbf{Z}^\top \mathbf{A}^\top) = \text{rank}(\mathbf{A}^\top) = \text{rank}(\mathbf{A})$  where the last equality is again from the fact that the row rank is equal to the column rank of any matrix. This results in  $\text{rank}(\mathbf{AZ}) = \text{rank}(\mathbf{A})$  as claimed.

Since  $\mathbf{P}, \mathbf{P}^{-1}$  are nonsingular, we then have  $\text{rank}(\mathbf{PAP}^{-1}) = \text{rank}(\mathbf{AP}^{-1}) = \text{rank}(\mathbf{A})$  where the first equality is from claim 1 and the second equality is from claim 2. We complete the proof.  $\blacksquare$

The lemma above will be proved very useful in the sequel (see Lemma 27.3, p. 433 to prove the trace and rank of symmetric idempotent matrices).

### 8.3. Existence of the Hessenberg Decomposition

We will prove that any  $n \times n$  matrix can be reduced to Hessenberg form via a sequence of Householder transformations that are applied from the left and the right to the matrix. Previously, we utilized a Householder reflector to triangularize matrices and introduce zeros below the diagonal to obtain the QR decomposition. A similar approach can be applied to

introduce zeros below the subdiagonal. To see this, you have to revisit the ideas behind the Householder reflector in Definition 3.1 (p. 108).

Before introducing the mathematical construction of such decomposition, we emphasize the following remark which will be very useful in the finding of the decomposition.

**Remark 8.1: Left and Right Multiplied by a Matrix with Block Identity**

For square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , and a matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{n-k} \end{bmatrix},$$

where  $\mathbf{I}_k$  is a  $k \times k$  identity matrix. Then  $\mathbf{BA}$  will not change the first  $k$  rows of  $\mathbf{A}$ , and  $\mathbf{AB}$  will not change the first  $k$  columns of  $\mathbf{A}$ .

The proof of this remark is trivial.

**First Step: Introduce Zeros for the First Column**

Let  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  be the column partitions of  $\mathbf{A}$ , and each  $\mathbf{a}_i \in \mathbb{R}^n$ . Suppose  $\bar{\mathbf{a}}_1, \bar{\mathbf{a}}_2, \dots, \bar{\mathbf{a}}_n \in \mathbb{R}^{n-1}$  are vectors removing the first component in  $\mathbf{a}_i$ 's. Let

$$r_1 = \|\bar{\mathbf{a}}_1\|, \quad \mathbf{u}_1 = \frac{\bar{\mathbf{a}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{a}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \widetilde{\mathbf{H}}_1 = \mathbf{I} - 2\mathbf{u}_1 \mathbf{u}_1^\top \in \mathbb{R}^{(n-1) \times (n-1)},$$

where  $\mathbf{e}_1$  here is the first basis for  $\mathbb{R}^{n-1}$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^{n-1}$ . To introduce zeros below the sub-diagonal and operate on the submatrix  $\mathbf{A}_{2:n, 1:n}$ , we append the Householder reflector into

$$\mathbf{H}_1 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{H}}_1 \end{bmatrix},$$

in which case,  $\mathbf{H}_1 \mathbf{A}$  will introduce zeros in the first column of  $\mathbf{A}$  below entry (2,1). The first row of  $\mathbf{A}$  will not be affected at all and kept unchanged by Remark 8.1. And we can easily verify that both  $\mathbf{H}_1$  and  $\widetilde{\mathbf{H}}_1$  are orthogonal matrices and they are symmetric (from the definition of Householder reflector). To have the form in Theorem 8.2, we multiply  $\mathbf{H}_1 \mathbf{A}$  on the right by  $\mathbf{H}_1^\top$  which results in  $\mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top$ . The  $\mathbf{H}_1^\top$  on the right will not change the first column of  $\mathbf{H}_1 \mathbf{A}$  and thus keep the zeros introduced in the first column.

An example of a  $5 \times 5$  matrix is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{c} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{H}_1 \times} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ \boldsymbol{0} & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ \boldsymbol{0} & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ \boldsymbol{0} & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\times \mathbf{H}_1^\top} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boldsymbol{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \\ \mathbf{A} \qquad \qquad \mathbf{H}_1 \mathbf{A} \qquad \qquad \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \end{array}$$

### Second Step: Introduce Zeros for the Second Column

Let  $\mathbf{B} = \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top$ , where the entries in the first column below entry (2,1) are all zeros. And the goal is to introduce zeros in the second column below entry (3,2). Let  $\mathbf{B}_2 = \mathbf{B}_{2:n,2:n} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{n-1}]$ . Suppose again  $\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \dots, \bar{\mathbf{b}}_{n-1} \in \mathbb{R}^{n-2}$  are vectors removing the first component in  $\mathbf{b}_i$ 's. We can again construct a Householder reflector

$$r_1 = \|\bar{\mathbf{b}}_1\|, \quad \mathbf{u}_2 = \frac{\bar{\mathbf{b}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{b}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \widetilde{\mathbf{H}}_2 = \mathbf{I} - 2\mathbf{u}_2 \mathbf{u}_2^\top \in \mathbb{R}^{(n-2) \times (n-2)}, \quad (8.1)$$

where  $\mathbf{e}_1$  now is the first basis for  $\mathbb{R}^{n-2}$ . To introduce zeros below the sub-diagonal and operate on the submatrix  $\mathbf{B}_{3:n,1:n}$ , we append the Householder reflector into

$$\mathbf{H}_2 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{H}}_2 \end{bmatrix},$$

where  $\mathbf{I}_2$  is a  $2 \times 2$  identity matrix. We can see that  $\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top$  will not change the first two rows of  $\mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top$ , and as the Householder cannot reflect a zero vector such that the zeros in the first column will be kept. Again, putting  $\mathbf{H}_2^\top$  on the right of  $\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top$  will not change the first 2 columns so that the zeros will be kept.

Following the example of a  $5 \times 5$  matrix, the second step is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{ccc} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_2 \times} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ 0 & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ 0 & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} & \xrightarrow{\times \mathbf{H}_2^\top} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ 0 & 0 & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ 0 & 0 & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} \\ \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top & & \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top & & \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \end{array}$$

Same process can go on, and there are  $n - 2$  such steps. We will finally triangularize by

$$\mathbf{H} = \mathbf{H}_{n-2} \mathbf{H}_{n-3} \dots \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \dots \mathbf{H}_{n-2}^\top.$$

And since  $\mathbf{H}_i$ 's are symmetric and orthogonal, the above equation can be simply reduced to

$$\mathbf{H} = \mathbf{H}_{n-2} \mathbf{H}_{n-3} \dots \mathbf{H}_1 \mathbf{A} \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_{n-2}.$$

Note here only  $n - 2$  such stages exist rather than  $n - 1$  or  $n$ . We will verify this number of steps by the example below. The example of a  $5 \times 5$  matrix as a whole is shown as follows where again  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

#### A Complete Example of Hessenberg Decomposition

$$\begin{array}{c}
\left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{H}_1 \times} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \end{array} \right] \xrightarrow{\times \mathbf{H}_1^\top} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \end{array} \right] \\
\mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \\
\\
\xrightarrow{\mathbf{H}_2 \times} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square & \square \end{array} \right] \xrightarrow{\times \mathbf{H}_2^\top} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square & \square \end{array} \right] \\
\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \qquad \qquad \qquad \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \\
\\
\xrightarrow{\mathbf{H}_3 \times} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square & \square \\ 0 & 0 & 0 & \square & \square & \square \end{array} \right] \xrightarrow{\times \mathbf{H}_3^\top} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square & \square \\ 0 & 0 & \square & \square & \square & \square \\ 0 & 0 & 0 & \square & \square & \square \end{array} \right] \\
\mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \qquad \qquad \qquad \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \mathbf{H}_3^\top
\end{array}$$

where we find

- when multiplying by  $\mathbf{H}_1$  on the left, we operate on a  $(5 - 1) \times (5 - 1 + 1)$  submatrix;
- when multiplying by  $\mathbf{H}_2$  on the left, we operate on a  $(5 - 2) \times (5 - 2 + 1)$  submatrix;
- and when multiplying by  $\mathbf{H}_3$  on the left, we operate on a  $(5 - 3) \times (5 - 3 + 1)$  submatrix.

Similarly,

- when multiplying by  $\mathbf{H}_1^\top$  on the right, we operate on a  $5 \times (5 - 1)$  submatrix;
- when multiplying by  $\mathbf{H}_2^\top$  on the right, we operate on a  $5 \times (5 - 2)$  submatrix;
- and when multiplying by  $\mathbf{H}_3^\top$  on the right, we operate on a  $5 \times (5 - 3)$  submatrix.

The above two findings are important for the computation and reduction in the complexity of the Hessenberg decomposition.

#### 8.4. Computing the Hessenberg Decomposition

Matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , and for left multiplied by  $\mathbf{H}_i$ 's in each step  $i \in \{1, 2, \dots, n - 2\}$ , we operate on a submatrix of size  $(n - i) \times (n - i + 1)$ . Furthermore, if we take the first column of this  $(n - i) \times (n - i + 1)$  matrix, we can set the first column by the reflected vector to obtain the Householder reflector such that we can operate on an  $(n - i) \times (n - i)$  submatrix instead.

For multiplied by  $\mathbf{H}_i^\top$ 's on the right in each step  $i \in \{1, 2, \dots, n - 2\}$ , we operate on a submatrix of size  $n \times (n - i)$ . After the Householder transformation, we get the final triangular matrix  $\mathbf{R}$ , and the process is shown in Algorithm 32.

Similar to computing the QR decomposition via Householder reflectors, in Algorithm 32, to compute  $\mathbf{H} = \mathbf{H}_{n-2}\mathbf{H}_{n-3}\dots\mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top\mathbf{H}_2^\top\dots\mathbf{H}_{n-2}^\top$ , we write out the equation

$$\begin{aligned}\mathbf{H} &= (\mathbf{H}_{n-2}\dots(\mathbf{H}_3(\mathbf{H}_2(\mathbf{H}_1\mathbf{A}\mathbf{H}_1)\mathbf{H}_2)\mathbf{H}_3)\dots\mathbf{H}_{n-2}) \\ &= \begin{bmatrix} \mathbf{I}_{n-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_{n-2}\mathbf{u}_{n-2}^\top \end{bmatrix} \dots \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_3\mathbf{u}_3^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_2\mathbf{u}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^\top \end{bmatrix} \\ &\quad \mathbf{A} \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_2\mathbf{u}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_3\mathbf{u}_3^\top \end{bmatrix} \dots \begin{bmatrix} \mathbf{I}_{n-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_{n-2}\mathbf{u}_{n-2}^\top \end{bmatrix},\end{aligned}$$

where the different colors for the parentheses indicate the computational order, and

- the upper-left of  $\mathbf{H}_1$  is a  $1 \times 1$  identity matrix, multiplying on the left will not change the **first row** of  $\mathbf{A}$ , then multiplying on the right will not change the **first column** of  $\mathbf{H}_1\mathbf{A}$ ;
- the upper-left of  $\mathbf{H}_2$  is a  $2 \times 2$  identity matrix, multiplying on the left will not change the **first 2 rows** of  $\mathbf{H}_1\mathbf{A}\mathbf{H}_1$ , then multiplying on the right will not change the **first 2 columns** of  $\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{H}_1$ ;
- the upper-left of  $\mathbf{H}_3$  is a  $3 \times 3$  identity matrix, multiplying on the left will not change the **first 3 rows** of  $\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{H}_1\mathbf{H}_2$ , then multiplying on the right will not change the **first 3 columns** of  $\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{H}_1\mathbf{H}_2$ ;
- the process can go on, and this property yields the step 8 and 9 in the algorithm.

Similarly, to get the final orthogonal matrix  $\mathbf{Q} = \mathbf{H}_1\mathbf{H}_2\dots\mathbf{H}_{n-2}$ , we write out the equation:

$$\begin{aligned}\mathbf{Q} &= \mathbf{H}_1\mathbf{H}_2\mathbf{H}_3\dots\mathbf{H}_{n-2} \\ &= \begin{bmatrix} \mathbf{I}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_2\mathbf{u}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_3\mathbf{u}_3^\top \end{bmatrix} \dots \begin{bmatrix} \mathbf{I}_{n-2} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} - 2\mathbf{u}_{n-2}\mathbf{u}_{n-2}^\top \end{bmatrix},\end{aligned}$$

where the upper-left of  $\mathbf{H}_1$  is a  $1 \times 1$  identity matrix, the upper-left of  $\mathbf{H}_2$  is a  $2 \times 2$  identity matrix, and it will not change the first 2 columns of  $\mathbf{H}_1$ ; and the upper-left of  $\mathbf{H}_3$  is a  $3 \times 3$  identity matrix which will not modify the first 3 columns of  $\mathbf{H}_1\mathbf{H}_2$ ; .... This property yields the step 15 in the algorithm.

### Theorem 8.1: (Algorithm Complexity: Hessenberg via Householder)

Algorithm 32 requires  $\sim \frac{10}{3}n^3$  flops to compute a Hessenberg decomposition of an  $n \times n$  square matrix. Further, if  $\mathbf{Q}$  is needed explicitly, additional  $\sim 2n^3$  flops required.

**Proof** [of Theorem 8.1] We separate the proof into obtaining the Hessenberg matrix and the orthogonal matrix.

**To obtain the Hessenberg matrix:** For loop  $i$ ,  $\mathbf{H}_{i+1:n,i:n} \in \mathbb{R}^{(n-i) \times (n-i+1)}$ . Thus  $\mathbf{a}$  is in  $\mathbb{R}^{n-i}$ .

In step 4, to compute  $r = \|\mathbf{a}\|$  involves  $n - i$  multiplications,  $n - i - 1$  additions, and 1 square root operation which is  $\boxed{2(n - i)}$  flops.

---

**Algorithm 32** Hessenberg Decomposition via the Householder Reflector

---

**Require:** matrix  $\mathbf{A}$  with size  $n \times n$ ;

- 1: Initially set  $\mathbf{H} = \mathbf{A}$ ; (**note**  $\mathbf{H}$  is the Hessenberg,  $\mathbf{H}_i$ 's are Householders)
- 2: **for**  $i = 1$  to  $n - 2$  **do**
- 3:    $\mathbf{a} = \mathbf{H}_{i+1:n,i}$ , i.e., first column of  $\mathbf{H}_{i+1:n,i:n} \in \mathbb{R}^{(n-i) \times (n-i+1)}$ ;
- 4:    $r = \|\mathbf{a}\|$ ;  $\triangleright 2(n-i)$  flops
- 5:    $\mathbf{u}_i = \mathbf{a} - r\mathbf{e}_1$ ;  $\triangleright 1$  flop
- 6:    $\mathbf{u}_i = \mathbf{u}_i / \|\mathbf{u}_i\|$ ;  $\triangleright 3(n-i)$  flops
- 7:    $\mathbf{H}_{i+1,i} = r$ ,  $\mathbf{H}_{i+2:n,i} = \mathbf{0}$ , i.e., set the value of first column of  $\mathbf{H}_{i+1:n,i:n}$ ;  $\triangleright 0$  flops
- 8:   Left: set the value of columns 2 to  $n$  of  $\mathbf{H}_{i+1:n,i:n}$ ,

$$\begin{aligned}\mathbf{H}_{i+1:n,i+1:n} &= (\mathbf{I} - 2\mathbf{u}_i\mathbf{u}_i^\top)\mathbf{H}_{i+1:n,i+1:n} \\ &= \mathbf{H}_{i+1:n,i+1:n} - 2\mathbf{u}_i\mathbf{u}_i^\top\mathbf{H}_{i+1:n,i+1:n} \in \mathbb{R}^{(n-i) \times (n-i)} \\ &\quad (4(n-i)^2 \text{ flops});\end{aligned}$$

- 9:   Right:

$$\begin{aligned}\mathbf{H}_{1:n,i+1:n} &= \mathbf{H}_{1:n,i+1:n}(\mathbf{I} - 2\mathbf{u}_i\mathbf{u}_i^\top) \\ &= \mathbf{H}_{1:n,i+1:n} - \mathbf{H}_{1:n,i+1:n}2\mathbf{u}_i\mathbf{u}_i^\top \in \mathbb{R}^{n \times (n-i)} \\ &\quad (4n(n-i) - n \text{ flops});\end{aligned}$$

- 10: **end for**

- 11: Output  $\mathbf{H}$  as the upper Hessenberg matrix;
- 12: Get  $\mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_{n-2}$ , where  $\mathbf{H}_i$ 's are Householder reflectors.
- 13: Initially set  $\mathbf{Q} = \mathbf{H}_1$ ;
- 14: **for**  $i = 1$  to  $n - 3$  **do**
- 15:   Compute  $\mathbf{Q}$ :

$$\begin{aligned}\mathbf{Q}_{1:n,i+2:n} &= \mathbf{Q}_{1:n,i+2:n}\mathbf{H}_{i+1} \\ &= \mathbf{Q}_{1:n,i+2:n}(\mathbf{I} - 2\mathbf{u}_{i+1}\mathbf{u}_{i+1}^\top) \\ &= \mathbf{Q}_{1:n,i+2:n} - \mathbf{Q}_{1:n,i+2:n}2\mathbf{u}_{i+1}\mathbf{u}_{i+1}^\top \in \mathbb{R}^{n \times (n-i-1)} \\ &\quad (4n(n-i-1) - n \text{ flops});\end{aligned}$$

- 16: **end for**

- 17: Output  $\mathbf{Q}$  as the orthogonal matrix.
- 

In step 5,  $\mathbf{u}_i = \mathbf{a} - r\mathbf{e}_1$  requires 1 subtraction which is  $\boxed{1}$  flop as the special structure of  $\mathbf{e}_1$ ;

In step 6, same to step 4, it requires  $2(n-i)$  flops to compute the norm and  $(n-i)$  additional divisions which is  $\boxed{3(n-i)}$  flops.

In step 8, suppose in loop  $i$ ,  $\mathbf{u}_i^\top \mathbf{H}_{i+1:n,i+1:n}$  requires  $n - i$  times  $(n - i)$  multiplications and  $n - i - 1$  additions which is  $\boxed{(n-i)(2(n-i)-1)}$  flops.  $2\mathbf{u}_i$  requires  $\boxed{n-i}$  multiplications. Further  $2\mathbf{u}_i(\mathbf{u}_i^\top \mathbf{H}_{i+1:n,i+1:n})$  requires  $\boxed{(n-i)^2}$  multiplications to make an

$(n - i) \times (n - i)$  matrix. The final matrix subtraction needs  $(n - i)^2$  subtractions. Thus the total complexity for step 7 if loop  $i$  is  $4(n - i)^2$  flops for each iteration  $i$ .

In step 9, the computation of  $2\mathbf{u}_i$  needs 0 flops since it has already been calculated in step 8. Similarly,  $\mathbf{H}_{1:n,i+1:n}2\mathbf{u}_i$  involves  $n$  times  $(n - i)$  multiplications and  $n - i - 1$  additions which is  $n(2(n - i) - 1)$  flops.  $\mathbf{H}_{1:n,i+1:n}2\mathbf{u}_i\mathbf{u}_i^\top$  takes  $n(n - i)$  multiplications to make an  $n \times (n - i)$  matrix. The final matrix subtraction needs additional  $n(n - i)$  subtractions.

This makes the complexity of step 9 to be  $4n(n - i) - n$  flops for each iteration  $i$ .

So for loop  $i$  the total complexity is  $4i^2 - (12n + 4)i + (8n^2 + 3n + 2)$  flops. Let  $f(i) = 4i^2 - (12n + 4)i + (8n^2 + 3n + 2)$ , the complexity for step 2 to step 10 can be obtained by

$$\text{cost} = f(1) + f(2) + \dots + f(n - 2).$$

Simple calculation will show the sum of  $n - 2$  loops is  $\frac{10}{3}n^3$  flops if we keep only the leading term.

**To obtain the orthogonal matrix:** For the additional computation of  $\mathbf{Q}$  in step 15, the situation is similar to step 9. the computation of  $2\mathbf{u}_{i+1}$  needs 0 flops since it has already been calculated in step 8.  $\mathbf{Q}_{1:n,i+2:n}2\mathbf{u}_{i+1}$  involves  $n$  times  $(n - i - 1)$  multiplications and  $n - i - 2$  additions which is  $n(2(n - i - 1) - 1)$  flops.  $\mathbf{Q}_{1:n,i+2:n}2\mathbf{u}_{i+1}\mathbf{u}_{i+1}^\top$  takes  $n(n - i - 1)$  to make an  $n \times (n - i - 1)$  matrix. The final matrix subtraction needs additional  $n(n - i - 1)$  subtractions. This makes the complexity of step 15 to be  $4n(n - i - 1) - n$  flops for each iteration  $i$ . Let  $g(i) = 4n(n - i - 1) - n$ , the complexity for step 14 to step 16 is given by

$$\text{cost} = g(1) + g(2) + \dots + g(n - 3).$$

Trivial calculations can show that the complexity is  $2n^3$  flops if we keep only the leading term.  $\blacksquare$

## 8.5. Properties of the Hessenberg Decomposition

The Hessenberg decomposition is not unique in the different ways to construct the Householder reflectors (say Equation (8.1), p. 201). However, under mild conditions, we can claim a similar structure in different decompositions.

### Theorem 8.1: (Implicit Q Theorem for Hessenberg Decomposition)

Suppose two Hessenberg decompositions of matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  are given by  $\mathbf{A} = \mathbf{U}\mathbf{H}\mathbf{U}^\top = \mathbf{V}\mathbf{G}\mathbf{V}^\top$  where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$  and  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$  are the column partitions of  $\mathbf{U}, \mathbf{V}$ . Suppose further that  $k$  is the smallest positive integer for which  $h_{k+1,k} = 0$  where  $h_{ij}$  is the entry  $(i, j)$  of  $\mathbf{H}$ . Then

- If  $\mathbf{u}_1 = \mathbf{v}_1$ , then  $\mathbf{u}_i = \pm \mathbf{v}_i$  and  $|h_{i,i-1}| = |g_{i,i-1}|$  for  $i \in \{2, 3, \dots, k\}$ .

- When  $k = n - 1$ , the Hessenberg matrix  $\mathbf{H}$  is known as *unreduced* (Definition 8.1, p. 197). However, if  $k < n - 1$ , then  $g_{k+1,k} = 0$ .

**Proof** [of Theorem 8.1] Define the orthogonal matrix  $\mathbf{Q} = \mathbf{V}^\top \mathbf{U}$  and we have

$$\left. \begin{aligned} \mathbf{G}\mathbf{Q} &= \mathbf{V}^\top \mathbf{A}\mathbf{V}\mathbf{V}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{A}\mathbf{U} \\ \mathbf{Q}\mathbf{H} &= \mathbf{V}^\top \mathbf{U}\mathbf{U}^\top \mathbf{A}\mathbf{U} = \mathbf{V}^\top \mathbf{A}\mathbf{U} \end{aligned} \right\} \quad \xrightarrow{\text{leads to}} \quad \mathbf{G}\mathbf{Q} = \mathbf{Q}\mathbf{H},$$

the  $(i - 1)$ -th column of each can be represented as

$$\mathbf{G}\mathbf{q}_{i-1} = \mathbf{Q}\mathbf{h}_{i-1},$$

where  $\mathbf{q}_{i-1}$  and  $\mathbf{h}_{i-1}$  are the  $(i - 1)$ -th column of  $\mathbf{Q}$  and  $\mathbf{H}$  respectively. Since  $h_{l,i-1} = 0$  for  $l \geq i + 1$  (by the definition of upper Hessenberg matrices),  $\mathbf{Q}\mathbf{h}_{i-1}$  can be represented as

$$\mathbf{Q}\mathbf{h}_{i-1} = \sum_{j=1}^i h_{j,i-1} \mathbf{q}_j = h_{i,i-1} \mathbf{q}_i + \sum_{j=1}^{i-1} h_{j,i-1} \mathbf{q}_j.$$

Combine the two findings above, it follows that

$$h_{i,i-1} \mathbf{q}_i = \mathbf{G}\mathbf{q}_{i-1} - \sum_{j=1}^{i-1} h_{j,i-1} \mathbf{q}_j.$$

A moment of reflexion reveals that  $[\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k]$  is upper triangular. And since  $\mathbf{Q}$  is orthogonal, it must be diagonal and each value on the diagonal is in  $\{-1, 1\}$  for  $i \in \{2, \dots, k\}$ . Then,  $\mathbf{q}_1 = \mathbf{e}_1$  and  $\mathbf{q}_i = \pm \mathbf{e}_i$  for  $i \in \{2, \dots, k\}$ . Further, since  $\mathbf{q}_i = \mathbf{V}^\top \mathbf{u}_i$  and  $h_{i,i-1} = \mathbf{q}_i^\top (\mathbf{G}\mathbf{q}_{i-1} - \sum_{j=1}^{i-1} h_{j,i-1} \mathbf{q}_j) = \mathbf{q}_i^\top \mathbf{G}\mathbf{q}_{i-1}$ . For  $i \in \{2, \dots, k\}$ ,  $\mathbf{q}_i^\top \mathbf{G}\mathbf{q}_{i-1}$  is just  $\pm g_{i,i-1}$ . It follows that

$$\begin{aligned} |h_{i,i-1}| &= |g_{i,i-1}|, & \forall i \in \{2, \dots, k\}, \\ \mathbf{u}_i &= \pm \mathbf{v}_i, & \forall i \in \{2, \dots, k\}. \end{aligned}$$

This proves the first part. For the second part, if  $k < n - 1$ ,

$$\begin{aligned} g_{k+1,k} &= \mathbf{e}_{k+1}^\top \mathbf{G}\mathbf{e}_k = \pm \mathbf{e}_{k+1}^\top \underbrace{\mathbf{G}\mathbf{Q}}_{\mathbf{Q}\mathbf{H}} \mathbf{e}_k = \pm \mathbf{e}_{k+1}^\top \underbrace{\mathbf{Q}\mathbf{H}\mathbf{e}_k}_{\text{k-th column of } \mathbf{Q}\mathbf{H}} \\ &= \pm \mathbf{e}_{k+1}^\top \mathbf{Q}\mathbf{h}_k = \pm \mathbf{e}_{k+1}^\top \sum_j^{k+1} h_{jk} \mathbf{q}_j = \pm \mathbf{e}_{k+1}^\top \sum_j^k h_{jk} \mathbf{q}_j = 0, \end{aligned}$$

where the penultimate equality is from the assumption that  $h_{k+1,k} = 0$ . This completes the proof.  $\blacksquare$

We observe from the above theorem, when two Hessenberg decompositions of matrix  $\mathbf{A}$  are both unreduced and have the same first column in the orthogonal matrices, then the Hessenberg matrices  $\mathbf{H}, \mathbf{G}$  are similar matrices such that  $\mathbf{H} = \mathbf{D}\mathbf{G}\mathbf{D}^{-1}$  where  $\mathbf{D} = \text{diag}(\pm 1, \pm 1, \dots, \pm 1)$ . Moreover, and most importantly, if we restrict the elements in the lower sub-diagonal of the Hessenberg matrix  $\mathbf{H}$  to be positive (if possible), then the Hessenberg decomposition  $\mathbf{A} = \mathbf{Q}\mathbf{H}\mathbf{Q}^\top$  is uniquely determined by  $\mathbf{A}$  and the first column of  $\mathbf{Q}$ . This is similar to what we have claimed on the uniqueness of the QR decomposition (Corollary 3.1, p. 122).

The next finding involves a Krylov matrix defined as follows:

**Definition 8.2: Krylov Matrix**

Given matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , a vector  $\mathbf{q} \in \mathbb{R}^n$ , and a scalar  $k$ , the *Krylov matrix* is defined to be

$$\mathbf{K}(\mathbf{A}, \mathbf{q}, k) = [\mathbf{q} \quad \mathbf{A}\mathbf{q} \quad \dots \quad \mathbf{A}^{k-1}\mathbf{q}] \in \mathbb{R}^{n \times n}.$$

**Theorem 8.3: (Reduced Hessenberg)**

Suppose there exists an orthogonal matrix  $\mathbf{Q}$  such that  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as  $\mathbf{A} = \mathbf{Q}\mathbf{H}\mathbf{Q}^\top$ . Then  $\mathbf{Q}^\top\mathbf{A}\mathbf{Q} = \mathbf{H}$  is an unreduced upper Hessenberg matrix if and only if  $\mathbf{R} = \mathbf{Q}^\top\mathbf{K}(\mathbf{A}, \mathbf{q}_1, n)$  is nonsingular and upper triangular where  $\mathbf{q}_1$  is the first column of  $\mathbf{Q}$ .

If  $\mathbf{R}$  is singular and  $k$  is the smallest index so that  $r_{kk} = 0$ , then  $k$  is also the smallest index that  $h_{k,k-1} = 0$ .

**Proof** [of Theorem 8.3] We prove by forward implication and converse implication separately as follows:

**Forward implication** Suppose  $\mathbf{H}$  is unreduced, write out the following matrix

$$\mathbf{R} = \mathbf{Q}^\top\mathbf{K}(\mathbf{A}, \mathbf{q}_1, n) = [\mathbf{e}_1, \mathbf{H}\mathbf{e}_1, \dots, \mathbf{H}^{n-1}\mathbf{e}_1],$$

where  $\mathbf{R}$  is upper triangular with  $r_{11} = 1$  obviously. Observe that  $r_{ii} = h_{21}h_{32} \dots h_{i,i-1}$  for  $i \in \{2, 3, \dots, n\}$ . When  $\mathbf{H}$  is unreduced,  $\mathbf{R}$  is nonsingular as well.

**Converse implication** Now suppose  $\mathbf{R}$  is upper triangular and nonsingular, we observe that  $\mathbf{r}_{k+1} = \mathbf{H}\mathbf{r}_k$  such that the  $(k+2 : n)$ -th rows of  $\mathbf{H}$  are zero and  $h_{k+1,k} \neq 0$  for  $k \in \{1, 2, \dots, n-1\}$ . Then  $\mathbf{H}$  is unreduced.

If  $\mathbf{R}$  is singular and  $k$  is the smallest index so that  $r_{kk} = 0$ , then

$$\left. \begin{array}{l} r_{k-1,k-1} = h_{21}h_{32} \dots h_{k-1,k-2} \\ r_{kk} = h_{21}h_{32} \dots h_{k-1,k-2}h_{k,k-1} \end{array} \right\} \neq 0 \quad \text{leads to} \quad h_{k,k-1} = 0,$$

from which the result follows. ■

## Chapter 9

# Tridiagonal Decomposition: Hessenberg in Symmetric Matrices

### Contents

---

|     |                                                       |     |
|-----|-------------------------------------------------------|-----|
| 9.1 | Tridiagonal Decomposition . . . . .                   | 209 |
| 9.2 | Computing the Tridiagonal Decomposition . . . . .     | 209 |
| 9.3 | Properties of the Tridiagonal Decomposition . . . . . | 212 |

---

## 9.1. Tridiagonal Decomposition

We firstly give the formal definition of the tridiagonal matrix.

### Definition 9.1: Tridiagonal Matrix

A tridiagonal matrix is a square matrix where all the entries below the lower sub-diagonal and the entries above the upper sub-diagonal are zeros. I.e., the tridiagonal matrix is a *band matrix*.

The definition of the tridiagonal matrix can also be extended to rectangular matrices, and the form can be implied from the context.

In matrix language, for any matrix  $\mathbf{T} \in \mathbb{R}^{n \times n}$ , and the entry  $(i, j)$  denoted by  $t_{ij}$  for all  $i, j \in \{1, 2, \dots, n\}$ . Then  $\mathbf{T}$  with  $t_{ij} = 0$  for all  $i \geq j + 2$  and  $i \leq j - 2$  is known as a tridiagonal matrix.

Let  $i$  denote the smallest positive integer for which  $h_{i+1,i} = 0$  where  $i \in \{1, 2, \dots, n-1\}$ , then  $\mathbf{T}$  is *unreduced* if  $i = n-1$ .

Take a  $5 \times 5$  matrix as an example, the lower triangular below the lower sub-diagonal and upper triangular above the upper sub-diagonal are zero in the tridiagonal matrix:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \\ \text{possibly unreduced} \end{array} \quad \begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \textcolor{blue}{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \\ \text{reduced} \end{array} .$$

Obviously, a tridiagonal matrix is a special case of an upper Hessenberg matrix. Then we have the following tridiagonal decomposition:

### Theorem 9.2: (Tridiagonal Decomposition)

Every  $n \times n$  symmetric matrix  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{Q} \mathbf{T} \mathbf{Q}^\top \quad \text{or} \quad \mathbf{T} = \mathbf{Q}^\top \mathbf{A} \mathbf{Q},$$

where  $\mathbf{T}$  is a *symmetric* tridiagonal matrix, and  $\mathbf{Q}$  is an orthogonal matrix.

The existence of the tridiagonal matrix is trivial by applying the Hessenberg decomposition to symmetric matrix  $\mathbf{A}$ .

## 9.2. Computing the Tridiagonal Decomposition

Since zeros are now introduced in rows as well as columns by symmetry, additional arithmetic can be avoided by ignoring these additional zeros.

An example of a  $5 \times 5$  matrix is shown as follows where  $\boxtimes$  or a letter represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{c}
\begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{H}_1 \times} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{a} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\times \mathbf{H}_1^\top} \begin{bmatrix} \boxtimes & \mathbf{a} & 0 & 0 & 0 \\ a & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \\
\mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top
\end{array}$$
  

$$\begin{array}{c}
\mathbf{H}_2 \times \begin{bmatrix} \boxtimes & a & 0 & 0 & 0 \\ a & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{b} & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\times \mathbf{H}_2^\top} \begin{bmatrix} \boxtimes & a & 0 & 0 & 0 \\ a & \boxtimes & \mathbf{b} & 0 & 0 \\ 0 & b & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \\
\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \qquad \qquad \qquad \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top
\end{array}$$
  

$$\begin{array}{c}
\mathbf{H}_3 \times \begin{bmatrix} \boxtimes & a & 0 & 0 & 0 \\ a & \boxtimes & b & 0 & 0 \\ 0 & b & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & c & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\times \mathbf{H}_3^\top} \begin{bmatrix} \boxtimes & a & 0 & 0 & 0 \\ a & \boxtimes & b & 0 & 0 \\ 0 & b & \boxtimes & \mathbf{c} & 0 \\ 0 & 0 & c & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \\
\mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \qquad \qquad \qquad \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \mathbf{H}_3^\top
\end{array}$$

### Theorem 9.1: (Algorithm Complexity: Tridiagonalization via Householder)

Algorithm 33 requires  $\sim \frac{4}{3}n^3$  flops to compute a Tridiagonal decomposition of an  $n \times n$  symmetric matrix. Further, if  $\mathbf{Q}$  is needed explicitly, additional  $\sim 2n^3$  flops required.

**Proof** [of Theorem 9.1] The complexity of step 9 needs close scrutiny. It follows from the left Householder and right Householder updates. Since we have computed the first column of  $\mathbf{T}_{i+1:n,i:n}$  and the first row of  $\mathbf{T}_{i:n,i+1:n}$  explicitly, the left and right Householder updates have the following form (where the red-colored text is the difference from the one in computing Hessenberg decomposition for non-symmetric matrices):

#### Where Does the Step 9 Come From: A Splitting Way

Left: update columns 2 to  $n$  of  $\mathbf{T}_{i+1:n,i:n}$ , i.e., working on  $\mathbf{T}_{i+1:n,i+1:n} \in \mathbb{R}^{(n-i) \times (n-i)}$ :

$$\begin{aligned}
\mathbf{T}_{i+1:n,i+1:n} &= (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top) \mathbf{T}_{i+1:n,i+1:n} \\
&= \mathbf{T}_{i+1:n,i+1:n} - 2\mathbf{u}_i \mathbf{u}_i^\top \mathbf{T}_{i+1:n,i+1:n} \in \mathbb{R}^{(n-i) \times (n-i)} \\
&\quad (4(n-i)^2 \text{ flops});
\end{aligned}$$

---

**Algorithm 33** Tridiagonal Decomposition via the Householder Reflector

---

**Require:** matrix  $\mathbf{A}$  with size  $n \times n$ ;

- 1: Initially set  $\mathbf{T} = \mathbf{A}$ ;
- 2: **for**  $i = 1$  to  $n - 2$  **do**
- 3:    $\mathbf{a} = \mathbf{T}_{i+1:n,i}$ , i.e., first column of  $\mathbf{T}_{i+1:n,i:n} \in \mathbb{R}^{(n-i) \times (n-i+1)}$ ;
- 4:    $r = \|\mathbf{a}\|$ ;  $\triangleright 2(n-i)$  flops
- 5:    $\mathbf{u}_i = \mathbf{a} - r\mathbf{e}_1$ ;  $\triangleright 1$  flop
- 6:    $\mathbf{u}_i = \mathbf{u}_i / \|\mathbf{u}_i\|$ ;  $\triangleright 3(n-i)$  flops
- 7:    $\mathbf{T}_{i+1,i} = r$ ,  $\mathbf{T}_{i+2:n,i} = \mathbf{0}$ , i.e., set the value of first column of  $\mathbf{T}_{i+1:n,i:n}$ ;  $\triangleright 0$  flops
- 8:    $\mathbf{T}_{i,i+1} = r$ ,  $\mathbf{T}_{i,i+2:n} = \mathbf{0}$ , i.e., set the value of first row of  $\mathbf{T}_{i:n,i+1:n}$ ;  $\triangleright 0$  flops
- 9: Left and Right: let  $\mathbf{Z} = \mathbf{T}_{i+1:n,i+1:n} \in \mathbb{R}^{(n-i) \times (n-i)}$ ,

$$\mathbf{Z} = \mathbf{H}_i \mathbf{Z} \mathbf{H}_i \quad (\mathbf{H}_i \text{ is the } i\text{-th Householder reflector})$$

$$\begin{aligned} &= (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top) \mathbf{Z} (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top) \\ &= \mathbf{Z} - \mathbf{Z} 2\mathbf{u}_i \mathbf{u}_i^\top - 2\mathbf{u}_i \mathbf{u}_i^\top \mathbf{Z} + 2\mathbf{u}_i \mathbf{u}_i^\top \mathbf{Z} 2\mathbf{u}_i \mathbf{u}_i^\top \end{aligned}$$

$\triangleright 4(n-i)^2$  flops

- 10: **end for**
- 11: Output  $\mathbf{T}$  as the tridiagonal matrix;
- 12: Get  $\mathbf{Q} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_{n-2}$ , where  $\mathbf{H}_i$ 's are Householder reflectors.
- 13: Initially set  $\mathbf{Q} = \mathbf{H}_1$ ;
- 14: **for**  $i = 1$  to  $n - 3$  **do**
- 15:   Compute  $\mathbf{Q}$ :

$$\begin{aligned} \mathbf{Q}_{1:n,i+2:n} &= \mathbf{Q}_{1:n,i+2:n} \mathbf{H}_{i+1} \\ &= \mathbf{Q}_{1:n,i+2:n} (\mathbf{I} - 2\mathbf{u}_{i+1} \mathbf{u}_{i+1}^\top) \\ &= \mathbf{Q}_{1:n,i+2:n} - \mathbf{Q}_{1:n,i+2:n} 2\mathbf{u}_{i+1} \mathbf{u}_{i+1}^\top \in \mathbb{R}^{n \times (n-i-1)} \end{aligned}$$

$\triangleright 4n(n-i-1) - n$  flops

- 16: **end for**
  - 17: Output  $\mathbf{Q}$  as the orthogonal matrix.
- 

Right: update rows 2 to  $n$  of  $\mathbf{T}_{i:n,i+1:n}$ , i.e., working on  $\mathbf{T}_{i+1:n,i+1:n} \in \mathbb{R}^{(n-i) \times (n-i)}$ :

$$\begin{aligned} \mathbf{T}_{i+1:n,i+1:n} &= \mathbf{T}_{i+1:n,i+1:n} (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top) \\ &= \mathbf{T}_{i+1:n,i+1:n} - \mathbf{T}_{i+1:n,i+1:n} 2\mathbf{u}_i \mathbf{u}_i^\top \in \mathbb{R}^{n \times (n-i)} \\ &\quad (4n(n-i) - n \text{ flops}); \end{aligned}$$

Since the two updates now working on the same submatrix  $\mathbf{T}_{i+1:n,i+1:n}$ , we can put them together which results in the step 9 of the algorithm. Let  $\mathbf{Z} = \mathbf{T}_{i+1:n,i+1:n} \in \mathbb{R}^{(n-i) \times (n-i)}$ .

A delve into the reduction of the complexity in step 9 goes as follows:

$$\begin{aligned}
& \mathbf{Z} \leftarrow \mathbf{H}_i \mathbf{Z} \mathbf{H}_i \quad (\mathbf{H}_i \text{ is the } i\text{-th Householder reflector}) \\
& = (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top) \mathbf{Z} (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top) \\
& = (\mathbf{Z} - 2\mathbf{u}_i \underbrace{\mathbf{u}_i^\top \mathbf{Z}}_{\mathbf{y}^\top} (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top)) \\
& = (\mathbf{Z} - 2\mathbf{u}_i \mathbf{y}^\top) (\mathbf{I} - 2\mathbf{u}_i \mathbf{u}_i^\top) \quad (\text{let } \mathbf{y}^\top = \mathbf{u}_i^\top \mathbf{Z} \rightarrow \mathbf{y} = \mathbf{Z} \mathbf{u}_i) \\
& = \mathbf{Z} - 2\mathbf{u}_i \mathbf{y}^\top - 2\mathbf{Z} \mathbf{u}_i \mathbf{u}_i^\top + 4\beta \mathbf{u}_i \mathbf{u}_i^\top \quad (\text{let } \beta = \mathbf{y}^\top \mathbf{u}_i) \\
& = \mathbf{Z} - (2\mathbf{u}_i \mathbf{y}^\top + 2\beta \mathbf{u}_i \mathbf{u}_i^\top) - (2 \underbrace{\mathbf{Z} \mathbf{u}_i \mathbf{u}_i^\top}_{\mathbf{y}} - 2\beta \mathbf{u}_i \mathbf{u}_i^\top) \\
& = \mathbf{Z} - (2\mathbf{u}_i \mathbf{y}^\top + 2\beta \mathbf{u}_i \mathbf{u}_i^\top) - (2\mathbf{y} \mathbf{u}_i^\top - 2\beta \mathbf{u}_i \mathbf{u}_i^\top) \quad (\mathbf{y} = \mathbf{Z} \mathbf{u}_i) \\
& = \mathbf{Z} - \{\mathbf{u}_i \underbrace{(2\mathbf{y}^\top + 2\beta \mathbf{u}_i^\top)}_{\mathbf{x}^\top} + \underbrace{(2\mathbf{y}^\top + 2\beta \mathbf{u}_i^\top)^\top}_{\mathbf{x}} \mathbf{u}_i^\top\} \\
& = \mathbf{Z} - \{\mathbf{u}_i \mathbf{x}^\top + (\mathbf{u}_i \mathbf{x}^\top)^\top\} \quad (\text{let } \mathbf{x} = (2\mathbf{y} + 2\beta \mathbf{u}_i))
\end{aligned}$$

The costs come from

- $\mathbf{y} = \mathbf{Z} \mathbf{u}_i$ :  $[2(n-i)-1](n-i) = 2(n-i)^2 - (n-i)$  flops from Lemma 1.3 (on the complexity of a matrix multiplication);
- $\beta = \mathbf{y}^\top \mathbf{u}_i$ :  $2(n-i) - 1$  flops;
- $\mathbf{x} = (2\mathbf{y} + 2\beta \mathbf{u}_i)$ :  $(n-i) + 1 + (n-i) = 2(n-i) + 1$  flops;
- $\mathbf{u}_i \mathbf{x}^\top$ :  $(n-i)^2$  flops;
- $(\mathbf{u}_i \mathbf{x}^\top)^\top$ : 0 flops;
- $\mathbf{u}_i \mathbf{x}^\top + (\mathbf{u}_i \mathbf{x}^\top)^\top$ :  $1 + 2 + \dots + (n-i) = \frac{(n-i)^2 + (n-i)}{2}$  additions since it results in a symmetric matrix;
- $\underbrace{\mathbf{Z}}_{\substack{\text{symmetric}}} - \underbrace{\{\mathbf{u}_i \mathbf{x}^\top + (\mathbf{u}_i \mathbf{x}^\top)^\top\}}_{\substack{\text{symmetric}}}$ :  $1 + 2 + \dots + (n-i) = \frac{(n-i)^2 + (n-i)}{2}$  subtractions since both matrices are symmetric;

If we keep only the leading terms of step 4 to step 9, the total complexity for loop  $i$  is given by  $f(i) = 4(n-i)^2$  flops. By summation, we find the final cost

$$\text{cost} = f(1) + f(2) + \dots + f(n-2) = \frac{4}{3}n^3 \text{ flops.}$$

This completes the proof. ■

### 9.3. Properties of the Tridiagonal Decomposition

Similarly, the tridiagonal decomposition is not unique. However, and most importantly, if we restrict the elements in the lower sub-diagonal of the tridiagonal matrix  $\mathbf{T}$  to be positive (if possible), then the tridiagonal decomposition  $\mathbf{A} = \mathbf{Q} \mathbf{T} \mathbf{Q}^\top$  is uniquely determined by  $\mathbf{A}$  and the first column of  $\mathbf{Q}$ .

**Theorem 9.1: (Implicit Q Theorem for Tridiagonal)**

Suppose two Tridiagonal decompositions of symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  are given by  $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^\top = \mathbf{V}\mathbf{G}\mathbf{V}^\top$  where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n]$  and  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$  are the column partitions of  $\mathbf{U}, \mathbf{V}$ . Suppose further that  $k$  is the smallest positive integer for which  $t_{k+1,k} = 0$  where  $t_{ij}$  is the entry  $(i, j)$  of  $\mathbf{T}$ . Then

- if  $\mathbf{u}_1 = \mathbf{v}_1$ , then  $\mathbf{u}_i = \pm \mathbf{v}_i$  and  $|t_{i,i-1}| = |g_{i,i-1}|$  for  $i \in \{2, 3, \dots, k\}$ .
- When  $k = n - 1$ , the tridiagonal matrix  $\mathbf{T}$  is known as unreduced. However, if  $k < n - 1$ , then  $g_{k+1,k} = 0$ .

From the above theorem, we observe that if we restrict the elements in the lower sub-diagonal of the tridiagonal matrix  $\mathbf{T}$  to be positive (if possible), i.e., *unreduced*, then the tridiagonal decomposition  $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^\top$  is uniquely determined by  $\mathbf{A}$  and the first column of  $\mathbf{Q}$ . This again is similar to what we have claimed on the uniqueness of the QR decomposition (Corollary 3.1, p. 122).

Similarly, a reduced tridiagonal decomposition can be obtained from the implication of the Krylov matrix (Definition 8.2, p. 207).

**Theorem 9.2: (Reduced Tridiagonal)**

Suppose there exists an orthogonal matrix  $\mathbf{Q}$  such that  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as  $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^\top$ . Then  $\mathbf{Q}^\top \mathbf{A} \mathbf{Q} = \mathbf{T}$  is an unreduced tridiagonal matrix if and only if  $\mathbf{R} = \mathbf{Q}^\top \mathbf{K}(\mathbf{A}, \mathbf{q}_1, n)$  is nonsingular and upper triangular where  $\mathbf{q}_1$  is the first column of  $\mathbf{Q}$ .

If  $\mathbf{R}$  is singular and  $k$  is the smallest index so that  $r_{kk} = 0$ , then  $k$  is also the smallest index that  $t_{k,k-1} = 0$ .

The proofs of the above two theorems are the same as those in Theorem 8.1 and 8.3 (p. 205 and p. 207).

# Chapter 10

## Bidiagonal Decomposition

### Contents

---

|      |                                                                                    |     |
|------|------------------------------------------------------------------------------------|-----|
| 10.1 | Bidiagonal Decomposition . . . . .                                                 | 215 |
| 10.2 | Existence of the Bidiagonal Decomposition: Golub-Kahan Bidiagonalization . . . . . | 215 |
| 10.3 | Computing the Bidiagonal Decomposition: Golub-Kahan Bidiagonalization . . . . .    | 220 |
| 10.4 | Computing the Bidiagonal Decomposition: LHC Bidiagonalization . . . . .            | 221 |
| 10.5 | Computing the Bidiagonal Decomposition: Three-Step Bidiagonalization . . . . .     | 222 |
| 10.6 | Connection to Tridiagonal Decomposition . . . . .                                  | 224 |

---

### 10.1. Bidiagonal Decomposition

We firstly give the rigorous definition of the upper Bidiagonal matrix as follows:

**Definition 10.1: Upper Bidiagonal Matrix**

An upper bidiagonal matrix is a square matrix which is a banded matrix with non-zero entries along the *main diagonal* and the *upper subdiagonal* (i.e., the ones above the main diagonal). This means there are exactly two nonzero diagonals in the matrix.

Furthermore, when the diagonal below the main diagonal has the non-zero entries, the matrix is lower bidiagonal.

The definition of bidigonal matrices can also be extended to rectangular matrices, and the form can be implied from the context.

Take a  $7 \times 5$  matrix as an example, the lower triangular below the main diagonal and the upper triangular above the upper subdiagonal are zero in the upper bidiagonal matrix:

$$\begin{bmatrix} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & \boxtimes & 0 \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Then we have the following bidiagonal decomposition:

**Theorem 10.2: (Bidiagonal Decomposition)**

Every  $m \times n$  matrix  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{U}\mathbf{B}\mathbf{V}^\top \quad \text{or} \quad \mathbf{B} = \mathbf{U}^\top \mathbf{A}\mathbf{V},$$

where  $\mathbf{B}$  is an upper bidiagonal matrix, and  $\mathbf{U}, \mathbf{V}$  are orthogonal matrices.

We will see the bidiagonalization resembles the form of a singular value decomposition where the only difference is the values of  $\mathbf{B}$  in bidiagonal form has nonzero entries on the upper sub-diagonal such that it will be shown to play an important role in the calculation of the singular value decomposition.

### 10.2. Existence of the Bidiagonal Decomposition: Golub-Kahan Bidiagonalization

Previously, we utilized a Householder reflector to triangularize matrices and introduce zeros below the diagonal to obtain the QR decomposition, and introduce zeros below the sub-diagonal to obtain the Hessenberg decomposition. A similar approach can be employed to find the bidiagonal decomposition. To see this, you have to recap the ideas behind the Householder reflector in Definition 3.1 (p. 108).

### First Step 1.1: Introduce Zeros for the First Column

Let  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  be the column partitions of  $\mathbf{A}$ , and each  $\mathbf{a}_i \in \mathbb{R}^m$ . We can construct the Householder reflector as follows:

$$r_1 = \|\mathbf{a}_1\|, \quad \mathbf{u}_1 = \frac{\mathbf{a}_1 - r_1 \mathbf{e}_1}{\|\mathbf{a}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \mathbf{H}_1 = \mathbf{I} - 2\mathbf{u}_1 \mathbf{u}_1^\top \in \mathbb{R}^{m \times m},$$

where  $\mathbf{e}_1$  here is the first basis for  $\mathbb{R}^m$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^m$ . In this case,  $\mathbf{H}_1 \mathbf{A}$  will introduce zeros in the first column of  $\mathbf{A}$  below entry (1,1), i.e., reflect  $\mathbf{a}_1$  to  $r_1 \mathbf{e}_1$ . We can easily verify that both  $\mathbf{H}_1$  is a symmetric and orthogonal matrix (from the definition of Householder reflector).

An example of a  $7 \times 5$  matrix is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{c} \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{H}_1 \times} \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \end{array}.$$

Till now, this is exactly what we have done in the QR decomposition via the Householder reflector in Section 3.13 (p. 108). Going further, to introduce zeros above the upper sub-diagonal of  $\mathbf{H}_1 \mathbf{A}$  is equivalent to introducing zeros below the lower subdiagonal of  $(\mathbf{H}_1 \mathbf{A})^\top$ .

### First Step 1.2: Introduce Zeros for the First Row

Now suppose we are looking at the transpose of  $\mathbf{H}_1 \mathbf{A}$ , that is  $(\mathbf{H}_1 \mathbf{A})^\top = \mathbf{A}^\top \mathbf{H}_1^\top \in \mathbb{R}^{n \times m}$  and the column partition is given by  $\mathbf{A}^\top \mathbf{H}_1^\top = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_m]$  where each  $\mathbf{z}_i \in \mathbb{R}^n$ . Suppose  $\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \dots, \bar{\mathbf{z}}_m \in \mathbb{R}^{n-1}$  are vectors removing the first component in  $\mathbf{z}_i$ 's. Let

$$r_1 = \|\bar{\mathbf{z}}_1\|, \quad \mathbf{v}_1 = \frac{\bar{\mathbf{z}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{z}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \tilde{\mathbf{L}}_1 = \mathbf{I} - 2\mathbf{v}_1 \mathbf{v}_1^\top \in \mathbb{R}^{(n-1) \times (n-1)},$$

where  $\mathbf{e}_1$  now is the first basis for  $\mathbb{R}^{n-1}$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^{n-1}$ . To introduce zeros below the sub-diagonal and operate on the submatrix  $(\mathbf{A}^\top \mathbf{H}_1^\top)_{2:n, 1:m}$ , we append the Householder reflector into

$$\mathbf{L}_1 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{L}}_1 \end{bmatrix},$$

in which case,  $\mathbf{L}_1 (\mathbf{A}^\top \mathbf{H}_1^\top)$  will introduce zeros in the first column of  $(\mathbf{A}^\top \mathbf{H}_1^\top)$  below entry (2,1), i.e., reflect  $\bar{\mathbf{z}}_1$  to  $r_1 \mathbf{e}_1$ . The first row of  $(\mathbf{A}^\top \mathbf{H}_1^\top)$  will not be affected at all and kept unchanged by Remark 8.1 (p. 200) such that the zeros introduced in Step 1.1 will be kept. And we can easily verify that both  $\mathbf{L}_1$  and  $\tilde{\mathbf{L}}_1$  are orthogonal matrices and they are symmetric (from the definition of Householder reflector).

Come back to the original untransposed matrix  $\mathbf{H}_1 \mathbf{A}$ , multiply on the right by  $\mathbf{L}_1^\top$  is to introduce zeros in the first row to the right of entry (1,2). Again, following the example

above, a  $7 \times 5$  matrix is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{c} \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{L}_1 \times} \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \end{array} \right] \xrightarrow{(\cdot)^\top} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{A}^\top \mathbf{H}_1^\top \qquad \qquad \mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \qquad \qquad \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \end{array}$$

In short,  $\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top$  finishes the first step to introduce zeros for the first column and the first row of  $\mathbf{A}$ .

### Second Step 2.1: Introduce Zeros for the Second Column

Let  $\mathbf{B} = \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top$ , where the entries in the first column below entry (1,1) are all zeros and the entries in the first row to the right of entry (1,2) are all zeros as well. And the goal is to introduce zeros in the second column below entry (2,2). Let  $\mathbf{B}_2 = \mathbf{B}_{2:m,2:n} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{n-1}] \in \mathbb{R}^{(m-1) \times (n-1)}$ . We can again construct a Householder reflector

$$r_1 = \|\mathbf{b}_1\|, \quad \mathbf{u}_2 = \frac{\mathbf{b}_1 - r_1 \mathbf{e}_1}{\|\mathbf{b}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \widetilde{\mathbf{H}}_2 = \mathbf{I} - 2\mathbf{u}_2 \mathbf{u}_2^\top \in \mathbb{R}^{(m-1) \times (m-1)},$$

where  $\mathbf{e}_1$  now is the first basis for  $\mathbb{R}^{m-1}$  i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^{m-1}$ . To introduce zeros below the main diagonal and operate on the submatrix  $\mathbf{B}_{2:m,2:n}$ , we append the Householder reflector into

$$\mathbf{H}_2 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{H}}_2 \end{bmatrix},$$

in which case, we can see that  $\mathbf{H}_2(\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top)$  will not change the first row of  $(\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top)$  by Remark 8.1 (p. 200), and as the Householder cannot reflect a zero vector such that the zeros in the first column will be kept as well.

Following the above example, a  $7 \times 5$  matrix is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{H}_2 \times} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \\ 0 & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} & \textbf{\boxtimes} \end{array} \right] \\ \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \qquad \qquad \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \end{array}$$

### Second Step 2.2: Introduce Zeros for the Second Row

Same as step 1.2), now suppose we are looking at the transpose of  $\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top$ , that is  $(\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top)^\top = \mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top \in \mathbb{R}^{n \times m}$  and the column partition is given by  $\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$  where each  $\mathbf{x}_i \in \mathbb{R}^n$ . Suppose  $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_m \in \mathbb{R}^{n-2}$  are vectors removing the first two components in  $\mathbf{x}_i$ 's. Construct the Householder reflector as follows:

$$r_1 = \|\bar{\mathbf{x}}_1\|, \quad \mathbf{v}_2 = \frac{\bar{\mathbf{x}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{x}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \tilde{\mathbf{L}}_2 = \mathbf{I} - 2\mathbf{v}_2 \mathbf{v}_2^\top \in \mathbb{R}^{(n-2) \times (n-2)},$$

where  $\mathbf{e}_1$  now is the first basis for  $\mathbb{R}^{n-2}$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^{n-2}$ . To introduce zeros below the sub-diagonal and operate on the submatrix  $(\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1 \mathbf{H}_2)_{3:n, 1:m}$ , we append the Householder reflector into

$$\mathbf{L}_1 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{L}}_2 \end{bmatrix},$$

where  $\mathbf{I}_2$  is a  $2 \times 2$  identity matrix. In this case,  $\mathbf{L}_2(\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top)$  will introduce zeros in the second column of  $(\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top)$  below entry (3,2). The first two rows of  $(\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top)$  will not be affected at all and kept unchanged by Remark 8.1 (p. 200). **Further, the first column of it will be kept unchanged as well.** And we can easily verify that both  $\mathbf{L}_1$  and  $\tilde{\mathbf{L}}_1$  are orthogonal matrices and they are symmetric (from the definition of Householder reflector).

Come back to the original *untransposed* matrix  $\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top$ , multiply on the right by  $\mathbf{L}_2^\top$  is to introduce zeros in the second row to the right of entry (2,3). Following the above example, a  $7 \times 5$  matrix is shown as follows where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

$$\begin{array}{c} \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{L}_2 \times} \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} \\ 0 & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} \\ 0 & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} & \boldsymbol{\boxtimes} \end{array} \right] \xrightarrow{(\cdot)^\top} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right]. \\ \mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top \qquad \qquad \mathbf{L}_2 \mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top \qquad \qquad \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \mathbf{L}_2^\top \end{array}$$

In short,  $\mathbf{H}_2(\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top) \mathbf{L}_2^\top$  finish the second step to introduce zeros for the second column and the second row of  $\mathbf{A}$ . Same process can go on, and we shall notice that there are  $n$  such  $\mathbf{H}_i$  Householder reflectors on the left and  $n-2$  such  $\mathbf{L}_i$  Householder reflectors on the right (suppose  $m > n$  for simplicity). The interleaved Householder factorization is known as the *Golub-Kahan Bidiagonalization* (Golub and Kahan, 1965). We will finally bidiagonalize

$$\mathbf{B} = \mathbf{H}_n \mathbf{H}_{n-1} \dots \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \mathbf{L}_2^\top \dots \mathbf{L}_{n-2}^\top.$$

And since the  $\mathbf{H}_i$ 's and  $\mathbf{L}_i$ 's are symmetric and orthogonal, we have

$$\mathbf{B} = \mathbf{H}_n \mathbf{H}_{n-1} \dots \mathbf{H}_1 \mathbf{A} \mathbf{L}_1 \mathbf{L}_2 \dots \mathbf{L}_{n-2}.$$

A full example of a  $7 \times 5$  matrix is shown as follows where again  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed.

### A Complete Example of Golub-Kahan Bidiagonalization

$$\begin{array}{c}
 \begin{array}{ccccc}
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes
 \end{array} \xrightarrow{\mathbf{H}_1 \times} \begin{array}{ccccc}
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes
 \end{array} \xrightarrow{\times \mathbf{L}_1^\top} \begin{array}{ccccc}
 \boxtimes & \boxtimes & 0 & 0 & 0 \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes
 \end{array} \\
 \mathbf{A} \qquad \qquad \mathbf{H}_1\mathbf{A} \qquad \qquad \mathbf{H}_1\mathbf{A}\mathbf{L}\mathbf{L}_1^\top
 \end{array} \\
 \begin{array}{cc}
 \xrightarrow{\mathbf{H}_2 \times} \begin{array}{ccccc}
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes
 \end{array} \xrightarrow{\times \mathbf{L}_2^\top} \begin{array}{ccccc}
 \boxtimes & \boxtimes & 0 & 0 & 0 \\
 0 & \boxtimes & \boxtimes & 0 & 0 \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes
 \end{array} \\
 \mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{L}\mathbf{L}_1^\top \qquad \qquad \mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{L}_1^\top\mathbf{L}_2^\top
 \end{array} \\
 \begin{array}{cc}
 \xrightarrow{\mathbf{H}_3 \times} \begin{array}{ccccc}
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes
 \end{array} \xrightarrow{\times \mathbf{L}_3^\top} \begin{array}{ccccc}
 \boxtimes & \boxtimes & 0 & 0 & 0 \\
 \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\
 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes
 \end{array} \\
 \mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{L}_1^\top\mathbf{L}_2^\top \qquad \qquad \mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{L}_1^\top\mathbf{L}_2^\top\mathbf{L}_3^\top
 \end{array} \\
 \begin{array}{cc}
 \xrightarrow{\mathbf{H}_4 \times} \begin{array}{ccccc}
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes \\
 0 & 0 & 0 & 0 & \boxtimes \\
 0 & 0 & 0 & 0 & \boxtimes \\
 0 & 0 & 0 & 0 & \boxtimes
 \end{array} \xrightarrow{\mathbf{H}_5 \times} \begin{array}{ccccc}
 \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\
 0 & 0 & 0 & \boxtimes & \boxtimes \\
 0 & 0 & 0 & 0 & \boxtimes \\
 0 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0 & 0 & 0
 \end{array} \\
 \mathbf{H}_4\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{L}_1^\top\mathbf{L}_2\mathbf{L}_3^\top \qquad \qquad \mathbf{H}_5\mathbf{H}_4\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{L}_1^\top\mathbf{L}_2\mathbf{L}_3^\top
 \end{array} .
 \end{array}$$

We present in a way where a right Householder reflector  $\mathbf{L}_i$  follows from a left one  $\mathbf{H}_i$ . However, a trivial error that might be employed is that we do the left ones altogether, and the right ones follow. That is, a bidiagonal decomposition is a combination of a QR decomposition and a Hessenberg decomposition. Nevertheless, this is problematic, the right Householder reflector  $\mathbf{L}_1$  will destroy the zeros introduced by the left ones. Therefore, the left and right reflectors need to be employed in an interleaved manner to introduce back the zeros.

### 10.3. Computing the Bidiagonal Decomposition: Golub-Kahan Bidiagonalization

---

**Algorithm 34** Golub-Kahan Bidiagonal Decomposition

---

**Require:** matrix  $\mathbf{A}$  with size  $m \times n$  with  $m > n$ ;

```

1: Initially set $\mathbf{B} = \mathbf{A}^\top$;
2: for $i = 1$ to $n - 2$ do
3: // do the left Householder reflector;
4: if $i \leq n - 2$ then
5: $\mathbf{B} = \mathbf{B}^\top$ such that $\mathbf{B} \in \mathbb{R}^{m \times n}$;
6: end if
7: $\mathbf{a} = \mathbf{B}_{i:m,i}$, i.e., first column of $\mathbf{B}_{i:m,i:n} \in \mathbb{R}^{(m-i+1) \times (n-i+1)}$;
8: $r = \|\mathbf{a}\|$; ▷ $2(m - i + 1)$ flops
9: $\mathbf{u}_i = \mathbf{a} - r\mathbf{e}_1 \in \mathbb{R}^{m-i+1}$; ▷ 1 flop
10: $\mathbf{u}_i = \mathbf{u}_i / \|\mathbf{u}_i\|$; ▷ $3(m - i + 1)$ flops
11: $\mathbf{B}_{i,i} = r$, $\mathbf{B}_{i+1:m,i} = \mathbf{0}$; ▷ update the first column of $\mathbf{B}_{i:m,i:n}$, 0 flops
12: $\mathbf{B}_{i:m,i+1:n} = \mathbf{B}_{i:m,i+1:n} - 2\mathbf{u}_i(\mathbf{u}_i^\top \mathbf{B}_{i:m,i+1:n})$; ▷ update the rest columns of $\mathbf{B}_{i:m,i:n}$,
 $4(m - i + 1)(n - i) + (m - n + 1)$ flops
13: if $i \leq n - 2$ then // do the right Householder reflector;
14: $\mathbf{B} = \mathbf{B}^\top$ such that $\mathbf{B} \in \mathbb{R}^{n \times m}$;
15: $\mathbf{z} = \mathbf{B}_{i+1:n,i}$, i.e., first column of $\mathbf{B}_{i+1:n,i:m} \in \mathbb{R}^{(n-i) \times (m-i+1)}$;
16: $s = \|\mathbf{z}\|$; ▷ $2(n - i)$ flops
17: $\mathbf{v}_i = \mathbf{z} - s\mathbf{e}_1 \in \mathbb{R}^{n-i}$; ▷ 1 flop
18: $\mathbf{v}_i = \mathbf{v}_i / \|\mathbf{v}_i\|$; ▷ $3(n - i)$ flops
19: $\mathbf{B}_{i+1,i} = r$, $\mathbf{B}_{i+2:n,i} = \mathbf{0}$; ▷ update the first column of $\mathbf{B}_{i+1:n,i:m}$, 0 flops
20: update columns $i + 1$ to m of $\mathbf{B}_{i+1:n,i:m}$:
```

$$\begin{aligned}
\mathbf{B}_{i+1:n,i+1:m} &= (\mathbf{I} - 2\mathbf{v}_i\mathbf{v}_i^\top)\mathbf{B}_{i+1:n,i+1:m} \\
&= \mathbf{B}_{i+1:n,i+1:m} - 2\mathbf{v}_i(\mathbf{v}_i^\top \mathbf{B}_{i+1:n,i+1:m}) \in \mathbb{R}^{(n-i) \times (m-i)} \\
&\quad (4(n - i)(m - i) + (n - m)) \text{ flops}
\end{aligned}$$

```

21: end if
22: end for
23: Output \mathbf{B} as the bidiagonal matrix;
24: Get $\mathbf{U} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_n$;
25: Initially set $\mathbf{U} = \mathbf{H}_1$;
26: for $i = 1$ to $n - 1$ do
27: $\mathbf{U}_{1:m,i+1:m} = \mathbf{U}_{1:m,i+1:m}(\mathbf{I} - 2\mathbf{u}_{i+1}\mathbf{u}_{i+1}^\top) = \mathbf{U}_{1:m,i+1:m} - \mathbf{U}_{1:m,i+1:m}2\mathbf{u}_{i+1}\mathbf{u}_{i+1}^\top$.
28: end for
29: Initially set $\mathbf{V} = \mathbf{L}_1$;
30: for $i = 1$ to $n - 3$ do
31: $\mathbf{V}_{1:n,i+2:n} = \mathbf{V}_{1:n,i+2:n}(\mathbf{I} - 2\mathbf{v}_{i+1}\mathbf{v}_{i+1}^\top) = \mathbf{V}_{1:n,i+2:n} - \mathbf{V}_{1:n,i+2:n}2\mathbf{v}_{i+1}\mathbf{v}_{i+1}^\top \in \mathbb{R}^{n \times (n-i-1)}$;
32: end for
33: Output \mathbf{U} , \mathbf{V} as the orthogonal matrix;
```

---

From the QR decomposition and Hessenberg decomposition via the Householder reflector, it is trivial to obtain the procedure formulated in Algorithm 34 where the red-colored text is the difference between the right Householder reflectors in bidiagonal decomposition and the reflectors in Hessenberg decomposition (Algorithm 32).

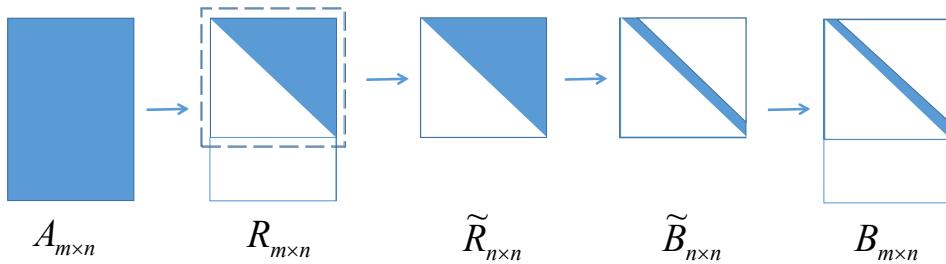
**Theorem 10.1: (Algorithm Complexity: Golub-Kahan Bidiagonalization)**

Algorithm 34 requires  $\sim 4mn^2 - \frac{4}{3}n^3$  flops to compute a bidiagonal decomposition of an  $m \times n$  matrix with  $m > n$ . Further, if  $\mathbf{U}, \mathbf{V}$  are needed explicitly, additional  $\sim 4m^2n - 2mn^2 + 2n^3$  flops are required.

The proof is trivial that the procedure shown above requires twice the complexity of QR decomposition via the Householder reflector as it resembles two Householder QR decomposition interleaved with one operating on the  $m \times n$  matrix  $\mathbf{A}$  and the other on the  $n \times m$  matrix  $\mathbf{A}^\top$ . The complexity to obtain the orthogonal matrix  $\mathbf{U}$  is  $4m^2n - 2mn^2$  flops which is the same as that in QR decomposition. And similarly, the complexity to obtain the orthogonal matrix  $\mathbf{V}$  is  $2n^3$  flops which is the same as that in Hessenberg decomposition.

#### 10.4. Computing the Bidiagonal Decomposition: LHC Bidiagonalization

We mentioned in the previous section that the left Householder reflectors and the right reflectors are applied in an interleaved manner, otherwise, zeros introduced by the left reflectors will be destroyed. Nevertheless, when  $m \gg n$ , we can extract the square triangular matrix (i.e., the QR decomposition) and apply the Golub-Kahan diagonalization on the square  $n \times n$  matrix. This is known as the *Lawson-Hanson-Chan (LHC) bidiagonalization* (Lawson and Hanson, 1995; Chan, 1982) and the procedure is shown in Figure 10.1.



**Figure 10.1:** Demonstration of LHC-bidiagonalization of a matrix

The LHC bidiagonalization starts by computing the QR decomposition  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ . Then follows by applying the Golub-Kahan process such that  $\tilde{\mathbf{R}} = \tilde{\mathbf{U}}\tilde{\mathbf{B}}\mathbf{V}^\top$  where  $\tilde{\mathbf{R}}$  is the square  $n \times n$  triangular submatrix inside  $\mathbf{R}$ . Append  $\tilde{\mathbf{U}}$  into

$$\mathbf{U}_0 = \begin{bmatrix} \tilde{\mathbf{U}} \\ \mathbf{I}_{m-n} \end{bmatrix},$$

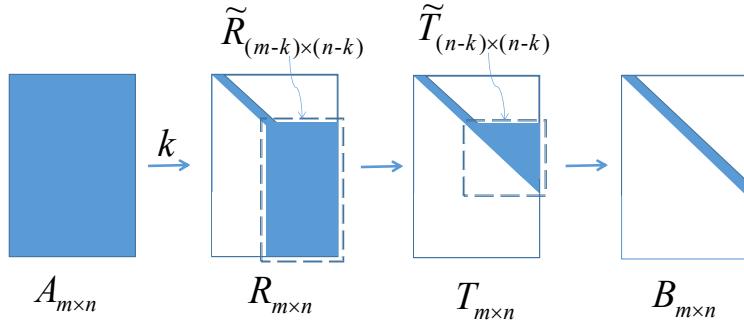
which results in  $\mathbf{R} = \mathbf{U}_0 \mathbf{B} \mathbf{V}^\top$  and  $\mathbf{A} = \mathbf{Q} \mathbf{U}_0 \mathbf{B} \mathbf{V}^\top$ . Let  $\mathbf{U} = \mathbf{Q} \mathbf{U}_0$ , we obtain the bidiagonal decomposition. The QR decomposition requires  $2mn^2 - \frac{2}{3}n^3$  flops and the Golub-Kahan process now requires  $\frac{8}{3}n^3$  (operating on an  $n \times n$  submatrix). Thus the total complexity to obtain the bidiagonal matrix  $\mathbf{B}$  is

$$\text{LHC bidiagonalization: } \sim 2mn^2 + 2n^3 \text{ flops.}$$

The LHC process creates zeros and then destroys them again in the lower triangle of the upper  $n \times n$  square of  $\mathbf{R}$ , but the zeros in the lower  $(m-n) \times n$  rectangular matrix of  $\mathbf{R}$  will be kept. Thus when  $m-n$  is large enough (or  $m \gg n$ ), there is a net gain. Simple calculations will show the LHC bidiagonalization costs less when  $m > \frac{5}{3}n$  compared to the Golub-Kahan bidiagonalization.

### 10.5. Computing the Bidiagonal Decomposition: Three-Step Bidiagonalization

The LHC procedure is advantageous only when  $m > \frac{5}{3}n$ . A further trick is to apply the QR decomposition not at the beginning of the computation, but at a suitable point in the middle ([Trefethen and Bau III, 1997](#)). In particular, the procedure is shown in Figure 10.2 where we apply the first  $k$  steps of left and right Householder reflectors as in the Golub-Kahan process leaving the bottom-right  $(m-k) \times (n-k)$  submatrix “unreflected”. Then follow up the same LHC process on the submatrix to obtain the final bidiagonal decomposition. By doing so, the complexity reduces when  $n < m < 2n$ .



**Figure 10.2:** Demonstration of Three-Step bidiagonalization of a matrix

The complexity of the Three-Step bidiagonalization can be decomposed into three ones. The complexity of  $k$  loops in Algorithm 34 can be shown to be

$$\text{Step 1: } f_1 = 8mnk - (4m + 4n)k^2 + \frac{8}{3}k^3 \text{ flops,}$$

which makes  $4mn^2 - \frac{8}{3}n^3$  flops when  $k = n$ . The complexity of the QR decomposition via the Householder reflector for  $\tilde{\mathbf{R}} \in \mathbb{R}^{(m-k) \times (n-k)}$  is

$$\text{Step 2: } f_2 = 2(m-k)(n-k)^2 - \frac{2}{3}(n-k)^3 \text{ flops.}$$

And the complexity of the Golub-Kahan diagonalization for  $\tilde{T} \in \mathbb{R}^{(n-k) \times (n-k)}$  is

$$\text{Step 3: } f_3 = \frac{8}{3}(n-k)^3 \text{ flops.}$$

Thus, the total complexity of the three steps is given by

$$g(k) = f_1 + f_2 + f_3 = -\frac{4}{3}k^3 + (6n-2m)k^2 + (4mn-8n^2)k + 2mn^2 + 2n^3.$$

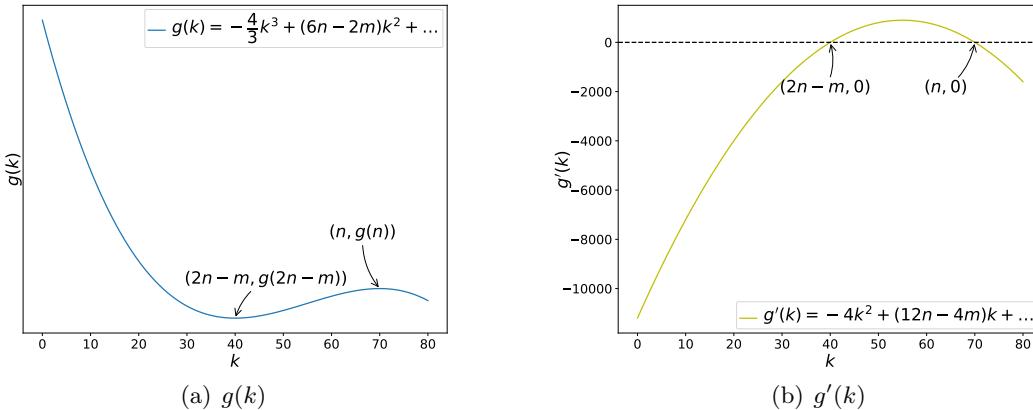
The problem now becomes finding a  $k < n$  such that  $g(k)$  is minimized. Taking the gradient of above function, we obtain

$$g'(k) = -4k^2 + (12n-4m)k + (4mn-8n^2),$$

of which the root is  $k = n$  or  $2n-m$ . We notice that  $0 < k < n$  such that when  $2n-m > 0$ , the optimal value appears in one of  $\{0, n, 2n-m\}$ . Trivial calculation shows that the optimal value for  $k$  is  $k = 2n-m$  and the final complexity is now reduced to

$$g(2n-m) = 2mn^2 + 2m^2n - \frac{2}{3}m^3 - \frac{2}{3}n^3 \text{ flops,} \quad \text{when } m < 2n.$$

An example of the complexity of the Three-Step method when  $n = 70, m = 100$  is shown in Figure 10.3 where the roots of the gradient are found to be  $2n-m = 40$  and  $n = 70$  such that  $g'(40) = g'(70) = 0$ . In this specific case, the function  $g(k)$  is decreasing when  $k \in (0, 2n-m]$  and increasing when  $k \in (2n-m, n]$ .

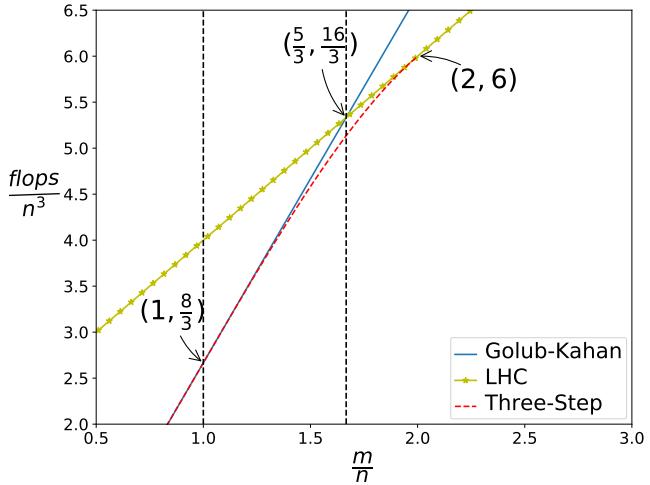


**Figure 10.3:** An example of the complexity when  $n = 70, m = 100$ .

To conclude, the costs of the three methods are shown as follows:

$$\begin{cases} \text{Golub-Kahan: } \sim 4mn^2 - \frac{4}{3}n^3 \text{ flops,} \\ \text{LHC: } \sim 2mn^2 + 2n^3 \text{ flops,} \\ \text{Three-Step: } \sim 2mn^2 + 2m^2n - \frac{2}{3}m^3 - \frac{2}{3}n^3 \text{ flops.} \end{cases}$$

**Figure 10.4:** Comparison of the complexity among the three bidiagonal methods. When  $m > 2n$ , LHC is preferred; when  $n < m < 2n$ , the Three-Step method is preferred though the improvement is small enough.



When  $m > 2n$ , LHC is preferred; when  $n < m < 2n$ , the Three-Step method is preferred though the improvement is small enough as shown in Figure 10.4 where the operation counts for the three methods are plotted as a function of  $\frac{m}{n}$ .

Notice that the complexity discussed here does not involve the extra computation of  $\mathbf{U}, \mathbf{V}$ . We shall not discuss the issue for simplicity.

## 10.6. Connection to Tridiagonal Decomposition

We first illustrate the connection by the following lemma that reveals how to construct a tridiagonal matrix from a bidiagonal one.

### Lemma 10.1: (Construct Tridiagonal From Bidiagonal)

Suppose  $\mathbf{B} \in \mathbb{R}^{n \times n}$  is upper bidiagonal, then  $\mathbf{T}_1 = \mathbf{B}^\top \mathbf{B}$  and  $\mathbf{T}_2 = \mathbf{B}\mathbf{B}^\top$  are symmetric triangular matrices.

**Proof** [of Lemma 10.1] Suppose  $\mathbf{B}$  has the following form

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & 0 & 0 & \dots \\ 0 & b_{22} & b_{23} & 0 & \dots \\ 0 & 0 & b_{33} & b_{34} & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ \dots & \dots & \dots & \dots & b_{nn} \end{bmatrix}.$$

Then  $\mathbf{T}_1 = \mathbf{B}^\top \mathbf{B}$  is given by

$$\mathbf{T}_1 = \mathbf{B}^\top \mathbf{B} =$$

$$\begin{bmatrix} b_{11} & 0 & 0 & 0 & \dots \\ b_{12} & b_{22} & 0 & 0 & \dots \\ 0 & b_{23} & b_{33} & 0 & \dots \\ 0 & 0 & b_{34} & \ddots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & 0 & 0 & \dots \\ 0 & b_{22} & b_{23} & 0 & \dots \\ 0 & 0 & b_{33} & b_{34} & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ \dots & \dots & \dots & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} b_{11}^2 & b_{11}b_{12} & 0 & \dots \\ b_{11}b_{12} & b_{12}^2 + b_{22}^2 & b_{22}b_{23} & \dots \\ 0 & b_{22}b_{23} & b_{23}^2 + b_{33}^2 & \dots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix},$$

which is symmetric and tridiagonal as claimed. Similarly, we can prove  $\mathbf{T}_2 = \mathbf{B}\mathbf{B}^\top$  is also symmetric and tridiagonal:

$$\mathbf{T}_2 = \mathbf{B}\mathbf{B}^\top =$$

$$\begin{bmatrix} b_{11} & b_{12} & 0 & 0 & \dots \\ 0 & b_{22} & b_{23} & 0 & \dots \\ 0 & 0 & b_{33} & b_{34} & \dots \\ \vdots & \vdots & \vdots & \ddots & \ddots \\ \dots & \dots & \dots & \dots & b_{nn} \end{bmatrix} \begin{bmatrix} b_{11} & 0 & 0 & 0 & \dots \\ b_{12} & b_{22} & 0 & 0 & \dots \\ 0 & b_{23} & b_{33} & 0 & \dots \\ 0 & 0 & b_{34} & \vdots & \ddots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} = \begin{bmatrix} b_{11}^2 + b_{12}^2 & b_{12}b_{22} & 0 & \dots \\ b_{12}b_{22} & b_{22}^2 + b_{23}^2 & b_{23}b_{33} & \dots \\ 0 & b_{23}b_{33} & b_{33}^2 + b_{34}^2 & \dots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix},$$

■

The lemma above reveals an important property. Suppose  $\mathbf{A} = \mathbf{U}\mathbf{B}\mathbf{V}^\top$  is the bidiagonal decomposition of  $\mathbf{A}$ , then the symmetric matrix  $\mathbf{A}\mathbf{A}^\top$  has a tridiagonal decomposition

$$\mathbf{A}\mathbf{A}^\top = \mathbf{U}\mathbf{B}\mathbf{V}^\top \mathbf{V}\mathbf{B}^\top \mathbf{U}^\top = \mathbf{U}\mathbf{B}\mathbf{B}^\top \mathbf{U}^\top.$$

And the symmetric matrix  $\mathbf{A}^\top \mathbf{A}$  has a tridiagonal decomposition

$$\mathbf{A}^\top \mathbf{A} = \mathbf{V}\mathbf{B}^\top \mathbf{U}^\top \mathbf{U}\mathbf{B}\mathbf{V}^\top = \mathbf{V}\mathbf{B}^\top \mathbf{B}\mathbf{V}^\top.$$

As a final result in this section, we state a theorem giving the tridiagonal decomposition of a symmetric matrix with special eigenvalues.

### Theorem 10.2: (Tridiagonal Decomposition for Nonnegative Eigenvalues)

Suppose  $n \times n$  symmetric matrix  $\mathbf{A}$  has nonnegative eigenvalues, then there exists a matrix  $\mathbf{Z}$  such that

$$\mathbf{A} = \mathbf{Z}\mathbf{Z}^\top.$$

Moreover, the tridiagonal decomposition of  $\mathbf{A}$  can be reduced to a problem to find the bidiagonal decomposition of  $\mathbf{Z} = \mathbf{U}\mathbf{B}\mathbf{V}^\top$  such that the tridiagonal decomposition of  $\mathbf{A}$  is given by

$$\mathbf{A} = \mathbf{Z}\mathbf{Z}^\top = \mathbf{U}\mathbf{B}\mathbf{B}^\top \mathbf{U}^\top.$$

**Proof** [of Theorem 10.2] The eigenvectors of symmetric matrices can be chosen to be orthogonal (Lemma 13.2, p. 242) such that symmetric matrix  $\mathbf{A}$  can be decomposed into

$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$  (spectral theorem 13.1, p. 241) where  $\Lambda$  is a diagonal matrix containing the eigenvalues of  $\mathbf{A}$ . When eigenvalues are nonnegative,  $\Lambda$  can be factored as  $\Lambda = \Lambda^{1/2}\Lambda^{1/2}$ . Let  $\mathbf{Z} = \mathbf{Q}\Lambda^{1/2}$ ,  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{Z}\mathbf{Z}^\top$ . Thus, combining our findings yields the result.  $\blacksquare$

# Part V

# Eigenvalue Problem



## Chapter 11

# Eigenvalue and Jordan Decomposition

### Contents

---

|      |                                                     |     |
|------|-----------------------------------------------------|-----|
| 11.1 | Eigenvalue Decomposition . . . . .                  | 230 |
| 11.2 | Existence of the Eigenvalue Decomposition . . . . . | 230 |
| 11.3 | Computing the Eigenvalue Decomposition . . . . .    | 232 |
| 11.4 | Jordan Decomposition . . . . .                      | 232 |
| 11.5 | Application: Computing Fibonacci Numbers . . . . .  | 234 |

---

### 11.1. Eigenvalue Decomposition

**Theorem 11.1: (Eigenvalue Decomposition)**

Any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with linearly independent eigenvectors can be factored as

$$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1},$$

where  $\mathbf{X}$  contains the eigenvectors of  $\mathbf{A}$  as columns, and  $\Lambda$  is a diagonal matrix  $\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  and  $\lambda_1, \lambda_2, \dots, \lambda_n$  are eigenvalues of  $\mathbf{A}$ .

Eigenvalue decomposition is also known as to diagonalize the matrix  $\mathbf{A}$ . When no eigenvalues of  $\mathbf{A}$  are repeated, the eigenvectors are sure to be linearly independent. Then  $\mathbf{A}$  can be diagonalized. Note here without  $n$  linearly independent eigenvectors, we cannot diagonalize. In Section 13.4 (p. 247), we will further discuss conditions under which the matrix has linearly independent eigenvectors.

### 11.2. Existence of the Eigenvalue Decomposition

**Proof** [of Theorem 11.1] Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  as the linearly independent eigenvectors of  $\mathbf{A}$ . Clearly, we have

$$\mathbf{A}\mathbf{x}_1 = \lambda_1\mathbf{x}_1, \quad \mathbf{A}\mathbf{x}_2 = \lambda_2\mathbf{x}_2, \quad \dots, \quad \mathbf{A}\mathbf{x}_n = \lambda_n\mathbf{x}_n.$$

In the matrix form,

$$\mathbf{AX} = [\mathbf{Ax}_1, \mathbf{Ax}_2, \dots, \mathbf{Ax}_n] = [\lambda_1\mathbf{x}_1, \lambda_2\mathbf{x}_2, \dots, \lambda_n\mathbf{x}_n] = \mathbf{X}\Lambda.$$

Since we assume the eigenvectors are linearly independent, then  $\mathbf{X}$  has full rank and is invertible. We obtain

$$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}.$$

This completes the proof. ■

We will discuss some similar forms of eigenvalue decomposition in the spectral decomposition section, where the matrix  $\mathbf{A}$  is required to be symmetric, and the  $\mathbf{X}$  is not only nonsingular but also orthogonal. Or, the matrix  $\mathbf{A}$  is required to be a *simple matrix*, that is, the algebraic multiplicity and geometric multiplicity are the same for  $\mathbf{A}$ , and  $\mathbf{X}$  will be a trivial nonsingular matrix that may not contain the eigenvectors of  $\mathbf{A}$ . The decomposition also has a geometric meaning, which we will discuss in Section 16 (p. 334).

A matrix decomposition in the form of  $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$  has a nice property that we can compute the  $m$ -th power efficiently.

**Remark 11.1:  $m$ -th Power**

The  $m$ -th power of  $\mathbf{A}$  is  $\mathbf{A}^m = \mathbf{X}\Lambda^m\mathbf{X}^{-1}$  if the matrix  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$ .

We notice that we require  $\mathbf{A}$  have linearly independent eigenvectors to prove the existence of the eigenvalue decomposition. Under specific conditions, the requirement is intrinsically satisfied.

### Lemma 11.2: (Different Eigenvalues)

Suppose the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  of  $\mathbf{A} \in \mathbb{R}^{n \times n}$  are all different. Then the corresponding eigenvectors are automatically independent. In another word, any square matrix with different eigenvalues can be diagonalized.

**Proof** [of Lemma 11.2] Suppose the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  are all different, and the eigenvectors  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  are dependent. That is, there exists a nonzero vector  $\mathbf{c} = [c_1, c_2, \dots, c_{n-1}]^\top$  such that

$$\mathbf{x}_n = \sum_{i=1}^{n-1} c_i \mathbf{x}_i.$$

Then we have

$$\begin{aligned} \mathbf{A}\mathbf{x}_n &= \mathbf{A}\left(\sum_{i=1}^{n-1} c_i \mathbf{x}_i\right) \\ &= c_1 \lambda_1 \mathbf{x}_1 + c_2 \lambda_2 \mathbf{x}_2 + \dots + c_{n-1} \lambda_{n-1} \mathbf{x}_{n-1}. \end{aligned}$$

and

$$\begin{aligned} \mathbf{A}\mathbf{x}_n &= \lambda_n \mathbf{x}_n \\ &= \lambda_n (c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \dots + c_{n-1} \mathbf{x}_{n-1}). \end{aligned}$$

Combine above two equations, we have

$$\sum_{i=1}^{n-1} (\lambda_n - \lambda_i) c_i \mathbf{x}_i = \mathbf{0}.$$

This leads to a contradiction since  $\lambda_n \neq \lambda_i$  for all  $i \in \{1, 2, \dots, n-1\}$ , from which the result follows. ■

### Remark 11.3: Limitation of Eigenvalue Decomposition

The limitation of eigenvalue decomposition is that:

- The eigenvectors in  $\mathbf{X}$  are usually not orthogonal and there are not always enough eigenvectors (i.e., some eigenvalues are equal).
- To compute the eigenvalues and eigenvectors  $\mathbf{Ax} = \lambda\mathbf{x}$  requires  $\mathbf{A}$  to be square. Rectangular matrices cannot be diagonalized by eigenvalue decomposition.

### 11.3. Computing the Eigenvalue Decomposition

The computation of eigenvalues and eigenvectors involves the computation of a polynomial. Some algorithms like Rayleigh quotient iterations can solve this problem and we will further discuss algorithms to find the eigenvalues and eigenvectors of a matrix in Section 15 (p. 289).

As long as we have the eigenvalues and eigenvectors, we just need to compute the inverse of the nonsingular matrix  $\mathbf{X}$ . As shown in Theorem 1.2 (p. 47), the complexity is  $2n^3$  flops.

### 11.4. Jordan Decomposition

In eigenvalue decomposition, we suppose matrix  $\mathbf{A}$  has  $n$  linearly independent eigenvectors. However, this is not necessarily true for all square matrices. We introduce further a generalized version of eigenvalue decomposition which is called the Jordan decomposition named after Camille Jordan ([Jordan, 1870](#)).

We first introduce the definition of Jordan blocks and Jordan form for the further description of Jordan decomposition.

#### Definition 11.1: Jordan Block

An  $m \times m$  upper triangular matrix  $B(\lambda, m)$  is called a Jordan block provided all  $m$  diagonal elements are the same eigenvalue  $\lambda$  and all upper sub-diagonal elements are all ones:

$$B(\lambda, m) = \begin{bmatrix} \lambda & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \lambda & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & \lambda & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & \lambda \end{bmatrix}_{m \times m}$$

#### Definition 11.2: Jordan Form

Given an  $n \times n$  matrix  $\mathbf{A}$ , a Jordan form  $\mathbf{J}$  for  $\mathbf{A}$  is a block diagonal matrix defined as

$$\mathbf{J} = \text{diag}(B(\lambda_1, m_1), B(\lambda_2, m_2), \dots, B(\lambda_k, m_k))$$

where  $\lambda_1, \lambda_2, \dots, \lambda_k$  are eigenvalues of  $\mathbf{A}$  (duplicates possible) and  $m_1 + m_2 + \dots + m_k = n$ .

Then, the Jordan decomposition follows:

#### Theorem 11.3: (Jordan Decomposition)

Any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as

$$\mathbf{A} = \mathbf{X} \mathbf{J} \mathbf{X}^{-1},$$

where  $\mathbf{X}$  is a nonsingular matrix containing the generalized eigenvectors of  $\mathbf{A}$  as columns, and  $\mathbf{J}$  is a Jordan form matrix  $\text{diag}(\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_k)$  where

$$\mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \lambda_i & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & \lambda_i & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_i & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda_i & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & \lambda_i \end{bmatrix}_{m_i \times m_i}$$

is an  $m_i \times m_i$  square matrix with  $m_i$  being the number of repetitions of eigenvalue  $\lambda_i$  and  $m_1 + m_2 + \dots + m_k = n$ .  $\mathbf{J}_i$ 's are referred to as Jordan blocks.

Further, nonsingular matrix  $\mathbf{X}$  is called the **matrix of generalized eigenvectors** of  $\mathbf{A}$ .

As an example, a Jordan form can have the following structure:

$$\begin{aligned} \mathbf{J} &= \text{diag}(B(\lambda_1, m_1), B(\lambda_2, m_2), \dots, B(\lambda_k, m_k)) \\ &= \left[ \begin{array}{c|ccccc} \begin{bmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 1 \\ 0 & 0 & \lambda_1 \end{bmatrix} & & & & & \\ \hline & \begin{bmatrix} \lambda_2 \end{bmatrix} & & & & \\ & & \begin{bmatrix} \lambda_3 & 1 \\ 0 & \lambda_3 \end{bmatrix} & & & \\ & & & \ddots & & \\ & & & & \begin{bmatrix} \lambda_k & 1 \\ 0 & \lambda_k \end{bmatrix} & \end{array} \right]. \end{aligned}$$

**Decoding a Jordan Decomposition:** Note that zeros can appear on the upper sub-diagonal of  $\mathbf{J}$  and in each block, the first column is always a diagonal containing only eigenvalues of  $\mathbf{A}$ . Take out one block to decode, without loss of generality, we take out the first block  $\mathbf{J}_1$ . We shall show the columns  $1, 2, \dots, m_1$  of  $\mathbf{AX} = \mathbf{XJ}$  with  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ :

$$\begin{aligned} \mathbf{Ax}_1 &= \lambda_1 \mathbf{x}_1 \\ \mathbf{Ax}_2 &= \lambda_1 \mathbf{x}_2 + \mathbf{x}_1 \\ &\vdots = \vdots \\ \mathbf{Ax}_{m_1} &= \lambda_1 \mathbf{x}_{m_1} + \mathbf{x}_{m_1-1}. \end{aligned}$$

For more details about Jordan decomposition, please refer to (Gohberg and Goldberg, 1996; Hales and Passi, 1999).

The Jordan decomposition is not particularly interesting in practice as it is extremely sensitive to perturbation. Even with the smallest random change to a matrix , the matrix can be made diagonalizable (van de Geijn and Myers, 2020). As a result, there is no practical mathematical software library or tool that computes it. And the proof takes dozens of pages to discuss. For this reason, we leave the proof to interesting readers.

### 11.5. Application: Computing Fibonacci Numbers

We use eigenvalue decomposition to compute the Fibonacci number. This example is drawn from (Strang, 2009). Every new Fibonacci number  $F_{k+2}$  is the sum of the two previous Fibonacci numbers  $F_{k+1} + F_k$ . The sequence is 0, 1, 1, 2, 3, 5, 8, .... Now, the problem is what is the number of  $F_{100}$ ?

Let  $\mathbf{u}_k = \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix}$ . Then  $\mathbf{u}_{k+1} = \begin{bmatrix} F_{k+2} \\ F_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{u}_k$  by the rule that  $F_{k+2} = F_{k+1} + F_k$  and  $F_{k+1} = F_{k+1}$ .

Let  $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ , we then have the equation that is  $\mathbf{u}_{100} = \mathbf{A}^{100} \mathbf{u}_0$  where  $\mathbf{u}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ .

We will see in Lemma 12.1 that  $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$  where  $\lambda$  is the eigenvalue of  $\mathbf{A}$ . Simple calculation shows that  $\det(\mathbf{A} - \lambda \mathbf{I}) = \lambda^2 - \lambda + 1 = 0$  and

$$\lambda_1 = \frac{1 + \sqrt{5}}{2}, \quad \lambda_2 = \frac{1 - \sqrt{5}}{2}.$$

The corresponding eigenvectors are

$$\mathbf{x}_1 = \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} \lambda_2 \\ 1 \end{bmatrix}.$$

By Remark 11.1,  $\mathbf{A}^{100} = \mathbf{X} \Lambda^{100} \mathbf{X}^{-1} = \mathbf{X} \begin{bmatrix} \lambda_1^{100} & 0 \\ 0 & \lambda_2^{100} \end{bmatrix} \mathbf{X}^{-1}$  where  $\mathbf{X}^{-1}$  can be easily calculated as  $\mathbf{X}^{-1} = \begin{bmatrix} \frac{1}{\lambda_1 - \lambda_2} & \frac{-\lambda_2}{\lambda_1 - \lambda_2} \\ -\frac{1}{\lambda_1 - \lambda_2} & \frac{\lambda_1}{\lambda_1 - \lambda_2} \end{bmatrix} = \begin{bmatrix} \frac{\sqrt{5}}{5} & \frac{5 - \sqrt{5}}{10} \\ -\frac{\sqrt{5}}{5} & \frac{5 + \sqrt{5}}{10} \end{bmatrix}$ . We notice that  $\mathbf{u}_{100} = \mathbf{A}^{100} \mathbf{u}_0$  is just the first column of  $\mathbf{A}^{100}$  which is

$$\mathbf{u}_{100} = \begin{bmatrix} F_{101} \\ F_{100} \end{bmatrix} = \begin{bmatrix} \frac{\lambda_1^{101} - \lambda_2^{101}}{\lambda_1 - \lambda_2} \\ \frac{\lambda_1^{100} - \lambda_2^{100}}{\lambda_1 - \lambda_2} \end{bmatrix}.$$

Check by calculation, we have  $F_{100} = 3.542248481792631e + 20$ . Or more generally,

$$\mathbf{u}_K = \begin{bmatrix} F_{K+1} \\ F_K \end{bmatrix} = \begin{bmatrix} \frac{\lambda_1^{K+1} - \lambda_2^{K+1}}{\lambda_1 - \lambda_2} \\ \frac{\lambda_1^K - \lambda_2^K}{\lambda_1 - \lambda_2} \end{bmatrix},$$

where the general form of  $F_K$  is given by  $F_K = \frac{\lambda_1^K - \lambda_2^K}{\lambda_1 - \lambda_2}$ .

# Chapter 12

## Schur Decomposition

### Contents

---

|      |                                                  |     |
|------|--------------------------------------------------|-----|
| 12.1 | Schur Decomposition . . . . .                    | 236 |
| 12.2 | Existence of the Schur Decomposition . . . . .   | 236 |
| 12.3 | Other Forms of the Schur Decomposition . . . . . | 238 |

---

## 12.1. Schur Decomposition

### Theorem 12.1: (Schur Decomposition)

Any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with real eigenvalues can be factored as

$$\mathbf{A} = \mathbf{Q}\mathbf{U}\mathbf{Q}^\top,$$

where  $\mathbf{Q}$  is an orthogonal matrix, and  $\mathbf{U}$  is an upper triangular matrix. That is, all square matrix  $\mathbf{A}$  with real eigenvalues can be triangularized.

**A close look at Schur decomposition** The first column of  $\mathbf{A}\mathbf{Q}$  and  $\mathbf{Q}\mathbf{U}$  are  $\mathbf{A}\mathbf{q}_1$  and  $\mathbf{U}_{11}\mathbf{q}_1$ . Then,  $\mathbf{U}_{11}, \mathbf{q}_1$  are eigenvalue and eigenvector of  $\mathbf{A}$ . But other columns of  $\mathbf{Q}$  need not be eigenvectors of  $\mathbf{A}$ .

**Schur decomposition for symmetric matrices** Symmetric matrix  $\mathbf{A} = \mathbf{A}^\top$  leads to  $\mathbf{Q}\mathbf{U}\mathbf{Q}^\top = \mathbf{Q}\mathbf{U}^\top\mathbf{Q}^\top$ . Then  $\mathbf{U}$  is a diagonal matrix. And this diagonal matrix actually contains eigenvalues of  $\mathbf{A}$ . All the columns of  $\mathbf{Q}$  are eigenvectors of  $\mathbf{A}$ . We conclude that all symmetric matrices are diagonalizable even with repeated eigenvalues.

## 12.2. Existence of the Schur Decomposition

To prove Theorem 12.1, we need to use the following lemmas.

### Lemma 12.1: (Determinant Intermezzo)

We have the following properties for determinant of matrices:

- The determinant of multiplication of two matrices is  $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$ ;
- The determinant of the transpose is  $\det(\mathbf{A}^\top) = \det(\mathbf{A})$ ;
- Suppose matrix  $\mathbf{A}$  has eigenvalue  $\lambda$ , then  $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ ;
- Determinant of any identity matrix is 1;
- Determinant of an orthogonal matrix  $\mathbf{Q}$ :

$$\det(\mathbf{Q}) = \det(\mathbf{Q}^\top) = \pm 1, \quad \text{since } \det(\mathbf{Q}^\top)\det(\mathbf{Q}) = \det(\mathbf{Q}^\top\mathbf{Q}) = \det(\mathbf{I}) = 1;$$

- Any square matrix  $\mathbf{A}$ , we then have an orthogonal matrix  $\mathbf{Q}$ :

$$\det(\mathbf{A}) = \det(\mathbf{Q}^\top)\det(\mathbf{A})\det(\mathbf{Q}) = \det(\mathbf{Q}^\top\mathbf{A}\mathbf{Q});$$

### Lemma 12.2: (Submatrix with Same Eigenvalue)

Suppose square matrix  $\mathbf{A}_{k+1} \in \mathbb{R}^{(k+1) \times (k+1)}$  has real eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_{k+1}$ . Then we can construct a  $k \times k$  matrix  $\mathbf{A}_k$  with eigenvalues  $\lambda_2, \lambda_3, \dots, \lambda_{k+1}$  by

$$\mathbf{A}_k = \begin{bmatrix} -\mathbf{p}_2^\top & - \\ -\mathbf{p}_3^\top & - \\ \vdots & \\ -\mathbf{p}_{k+1}^\top & - \end{bmatrix} \mathbf{A}_{k+1} \begin{bmatrix} \mathbf{p}_2 & \mathbf{p}_3 & \dots & \mathbf{p}_{k+1} \end{bmatrix},$$

where  $\mathbf{p}_1$  is a eigenvector of  $\mathbf{A}_{k+1}$  with norm 1 corresponding to eigenvalue  $\lambda_1$ , and  $\mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_{k+1}$  are any orthonormal vectors orthogonal to  $\mathbf{p}_1$ .

**Proof** [of Lemma 12.2] Let  $\mathbf{P}_{k+1} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{k+1}]$ . Then  $\mathbf{P}_{k+1}^\top \mathbf{P}_{k+1} = \mathbf{I}$ , and

$$\mathbf{P}_{k+1}^\top \mathbf{A}_{k+1} \mathbf{P}_{k+1} = \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \end{bmatrix}.$$

For any eigenvalue  $\lambda = \{\lambda_2, \lambda_3, \dots, \lambda_{k+1}\}$ , by Lemma 12.1, we have

$$\begin{aligned} \det(\mathbf{A}_{k+1} - \lambda \mathbf{I}) &= \det(\mathbf{P}_{k+1}^\top (\mathbf{A}_{k+1} - \lambda \mathbf{I}) \mathbf{P}_{k+1}) \\ &= \det(\mathbf{P}_{k+1}^\top \mathbf{A}_{k+1} \mathbf{P}_{k+1} - \lambda \mathbf{P}_{k+1}^\top \mathbf{P}_{k+1}) \\ &= \det \left( \begin{bmatrix} \lambda_1 - \lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k - \lambda \mathbf{I} \end{bmatrix} \right) \\ &= (\lambda_1 - \lambda) \det(\mathbf{A}_k - \lambda \mathbf{I}). \end{aligned}$$

Where the last equality is from the fact that if matrix  $\mathbf{M}$  has a block formulation:  $\mathbf{M} = \begin{bmatrix} \mathbf{E} & \mathbf{F} \\ \mathbf{G} & \mathbf{H} \end{bmatrix}$ , then  $\det(\mathbf{M}) = \det(\mathbf{E}) \det(\mathbf{H} - \mathbf{G}\mathbf{E}^{-1}\mathbf{F})$ . Since  $\lambda$  is an eigenvalue of  $\mathbf{A}$  and  $\lambda \neq \lambda_1$ , then  $\det(\mathbf{A}_{k+1} - \lambda \mathbf{I}) = (\lambda_1 - \lambda) \det(\mathbf{A}_k - \lambda \mathbf{I}) = 0$  means  $\lambda$  is also an eigenvalue of  $\mathbf{A}_k$ .  $\blacksquare$

We then prove the existence of the Schur decomposition by induction.

**Proof [of Theorem 12.1: Existence of Schur Decomposition]** We note that the theorem is trivial when  $n = 1$  by setting  $Q = 1$  and  $U = A$ . Suppose the theorem is true for  $n = k$  for some  $k \geq 1$ . If we prove the theorem is also true for  $n = k + 1$ , then we complete the proof.

Suppose for  $n = k$ , the theorem is true for  $\mathbf{A}_k = \mathbf{Q}_k \mathbf{U}_k \mathbf{Q}_k^\top$ .

Suppose further  $\mathbf{P}_{k+1}$  contains orthogonal vectors  $\mathbf{P}_{k+1} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{k+1}]$  as constructed in Lemma 12.2 where  $\mathbf{p}_1$  is an eigenvector of  $\mathbf{A}_{k+1}$  corresponding to eigenvalue  $\lambda_1$  and its norm is 1,  $\mathbf{p}_2, \dots, \mathbf{p}_{k+1}$  are orthonormal to  $\mathbf{p}_1$ . Let the other  $k$  eigenvalues of  $\mathbf{A}_{k+1}$  be  $\lambda_2, \lambda_3, \dots, \lambda_{k+1}$ . Since we suppose for  $n = k$ , the theorem is true, we can find a matrix  $\mathbf{A}_k$  with eigenvalues  $\lambda_2, \lambda_3, \dots, \lambda_{k+1}$ . So we have the following property by Lemma 12.2:

$$\mathbf{P}_{k+1}^\top \mathbf{A}_{k+1} \mathbf{P}_{k+1} = \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \end{bmatrix} \quad \text{and} \quad \mathbf{A}_{k+1} \mathbf{P}_{k+1} = \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \end{bmatrix}.$$

Let  $\mathbf{Q}_{k+1} = \mathbf{P}_{k+1} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix}$ . Then, it follows that

$$\begin{aligned}
\mathbf{A}_{k+1}\mathbf{Q}_{k+1} &= \mathbf{A}_{k+1}\mathbf{P}_{k+1} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \\
&= \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \\
&= \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \mathbf{Q}_k \end{bmatrix} \\
&= \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \mathbf{U}_k \end{bmatrix} \quad (\text{By the assumption for } n = k) \\
&= \mathbf{P}_{k+1} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_k \end{bmatrix} \\
&= \mathbf{Q}_{k+1} \mathbf{U}_{k+1}. \quad (\mathbf{U}_{k+1} = \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_k \end{bmatrix})
\end{aligned}$$

We then have  $\mathbf{A}_{k+1} = \mathbf{Q}_{k+1} \mathbf{U}_{k+1} \mathbf{Q}_{k+1}^\top$ , where  $\mathbf{U}_{k+1}$  is an upper triangular matrix, and  $\mathbf{Q}_{k+1}$  is an orthogonal matrix since  $\mathbf{P}_{k+1}$  and  $\begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix}$  are both orthogonal matrices and this orthogonal form is decoded in Appendix H (p. 468). ■

### 12.3. Other Forms of the Schur Decomposition

From the proof of the Schur decomposition, we obtain the upper triangular matrix  $\mathbf{U}_{k+1}$  by appending the eigenvalue  $\lambda_1$  to  $\mathbf{U}_k$ . From this process, the values on the diagonal are always eigenvalues. Therefore, we can decompose the upper triangular into two parts.

#### Corollary 12.1: (Form 2 of Schur Decomposition)

Any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with real eigenvalues can be factored as

$$\mathbf{Q}^\top \mathbf{A} \mathbf{Q} = \mathbf{\Lambda} + \mathbf{T}, \quad \text{or} \quad \mathbf{A} = \mathbf{Q}(\mathbf{\Lambda} + \mathbf{T})\mathbf{Q}^\top,$$

where  $\mathbf{Q}$  is an orthogonal matrix,  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  is a diagonal matrix containing the eigenvalues of  $\mathbf{A}$ , and  $\mathbf{T}$  is a *strictly upper triangular* matrix (with zeros on the diagonal).

A strictly upper triangular matrix is an upper triangular matrix having 0's along the diagonal as well as the lower portion. Another proof for this decomposition is that  $\mathbf{A}$  and  $\mathbf{U}$  (where  $\mathbf{U} = \mathbf{Q}^\top \mathbf{A} \mathbf{Q}$ ) are similar matrices so that they have the same eigenvalues (Lemma 8.2, p. 198). And the eigenvalues of any upper triangular matrices are on the diagonal. To see this, for any upper triangular matrix  $\mathbf{R} \in \mathbb{R}^{n \times n}$  where the diagonal values are  $r_{ii}$  for all  $i \in \{1, 2, \dots, n\}$ . We have

$$\mathbf{R} \mathbf{e}_i = r_{ii} \mathbf{e}_i,$$

where  $\mathbf{e}_i$  is the  $i$ -th basis vector in  $\mathbb{R}^n$ , i.e.,  $\mathbf{e}_i$  is the  $i$ -th column of the  $n \times n$  identity matrix  $\mathbf{I}_n$ . So we can decompose  $\mathbf{U}$  into  $\mathbf{\Lambda}$  and  $\mathbf{T}$ . Apparently, the first part of Lemma 8.2 (p. 198) can prove the existence of the second form of the Schur decomposition.

A final observation on the second form of the Schur decomposition is shown as follows. From  $\mathbf{AQ} = \mathbf{Q}(\mathbf{\Lambda} + \mathbf{T})$ , it follows that

$$\mathbf{Aq}_k = \lambda_k \mathbf{q}_k + \sum_{i=1}^{k-1} t_{ik} \mathbf{q}_i,$$

where  $t_{ik}$  is the  $(i, k)$ -th entry of  $\mathbf{T}$ . The form is quite close to the eigenvalue decomposition. Nevertheless, the columns become orthonormal bases and the orthonormal bases are correlated.

## Chapter 13

# Spectral Decomposition (Theorem)

### Contents

---

|        |                                                                                                          |     |
|--------|----------------------------------------------------------------------------------------------------------|-----|
| 13.1   | Spectral Decomposition . . . . .                                                                         | 241 |
| 13.2   | Existence of the Spectral Decomposition . . . . .                                                        | 241 |
| 13.3   | Uniqueness of Spectral Decomposition . . . . .                                                           | 247 |
| 13.4   | Other Forms, Connecting Eigenvalue Decomposition* . . . . .                                              | 247 |
| 13.5   | Skew-Symmetric Matrices and its Properties* . . . . .                                                    | 255 |
| 13.6   | Applications . . . . .                                                                                   | 258 |
| 13.6.1 | Application: Eigenvalue of Projection Matrix . . . . .                                                   | 258 |
| 13.6.2 | Application: An Alternative Definition on PD and PSD of Matrices                                         | 259 |
| 13.6.3 | Proof for Semidefinite Rank-Revealing Decomposition . . . . .                                            | 261 |
| 13.6.4 | Application: Cholesky Decomposition via the QR Decomposition<br>and the Spectral Decomposition . . . . . | 261 |
| 13.6.5 | Application: Unique Power Decomposition of Positive Definite<br>Matrices . . . . .                       | 262 |

---

### 13.1. Spectral Decomposition

**Theorem 13.1: (Spectral Decomposition)**

A real matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is symmetric if and only if there exists an orthogonal matrix  $\mathbf{Q}$  and a diagonal matrix  $\Lambda$  such that

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top,$$

where the columns of  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$  are eigenvectors of  $\mathbf{A}$  and are mutually orthonormal, and the entries of  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  are the corresponding eigenvalues of  $\mathbf{A}$ , which are real. And the rank of  $\mathbf{A}$  is the number of nonzero eigenvalues. This is known as the **spectral decomposition** or **spectral theorem** of real symmetric matrix  $\mathbf{A}$ . Specifically, we have the following properties:

1. A symmetric matrix has only **real eigenvalues**;
2. The eigenvectors are orthogonal such that they can be chosen **orthonormal** by normalization;
3. The rank of  $\mathbf{A}$  is the number of nonzero eigenvalues;
4. If the eigenvalues are distinct, the eigenvectors are unique as well.

The above decomposition is called the spectral decomposition for real symmetric matrices and is often known as the *spectral theorem*.

**Spectral theorem vs eigenvalue decomposition** In the eigenvalue decomposition, we require the matrix  $\mathbf{A}$  to be square and the eigenvectors to be linearly independent. Whereas in the spectral theorem, any symmetric matrix can be diagonalized, and the eigenvectors are chosen to be orthonormal.

**A word on the spectral decomposition** In Lemma 8.2 (p. 198), we proved that the eigenvalues of similar matrices are the same. From the spectral decomposition, we notice that  $\mathbf{A}$  and  $\Lambda$  are similar matrices such that their eigenvalues are the same. For any diagonal matrices, the eigenvalues are the diagonal components.<sup>1</sup> To see this, we realize that

$$\Lambda \mathbf{e}_i = \lambda_i \mathbf{e}_i,$$

where  $\mathbf{e}_i$  is the  $i$ -th basis vector. Therefore, the matrix  $\Lambda$  contains the eigenvalues of  $\mathbf{A}$ .

### 13.2. Existence of the Spectral Decomposition

We prove the theorem in several steps.

---

<sup>1</sup> Actually, we have shown in the last section that the diagonal values for triangular matrices are the eigenvalues of it.

## Symmetric Matrix Property 1 of 4

**Lemma 13.1: (Real Eigenvalues)**

The eigenvalues of any symmetric matrix are all real.

**Proof** [of Lemma 13.1] Suppose eigenvalue  $\lambda$  is a complex number  $\lambda = a + ib$  where  $a, b$  are real. Its complex conjugate is  $\bar{\lambda} = a - ib$ . Same for complex eigenvector  $\mathbf{x} = \mathbf{c} + i\mathbf{d}$  and its complex conjugate  $\bar{\mathbf{x}} = \mathbf{c} - i\mathbf{d}$  where  $\mathbf{c}, \mathbf{d}$  are real vectors. We then have the following property

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}\bar{\mathbf{x}} = \bar{\lambda}\bar{\mathbf{x}} \quad \xrightarrow{\text{transpose to}} \quad \bar{\mathbf{x}}^\top \mathbf{A} = \bar{\lambda}\bar{\mathbf{x}}^\top.$$

We take the dot product of the first equation with  $\bar{\mathbf{x}}$  and the last equation with  $\mathbf{x}$ :

$$\bar{\mathbf{x}}^\top \mathbf{A}\mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}, \quad \text{and} \quad \bar{\mathbf{x}}^\top \mathbf{A}\bar{\mathbf{x}} = \bar{\lambda}\bar{\mathbf{x}}^\top \bar{\mathbf{x}}.$$

Then we have the equality  $\lambda\bar{\mathbf{x}}^\top \mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}$ . Since  $\bar{\mathbf{x}}^\top \mathbf{x} = (\mathbf{c} - i\mathbf{d})^\top (\mathbf{c} + i\mathbf{d}) = \mathbf{c}^\top \mathbf{c} + \mathbf{d}^\top \mathbf{d}$  is a real number. Therefore the imaginary part of  $\lambda$  is zero and  $\lambda$  is real. ■

## Symmetric Matrix Property 2 of 4

**Lemma 13.2: (Orthogonal Eigenvectors)**

The eigenvectors corresponding to distinct eigenvalues of any symmetric matrix are orthogonal so that we can normalize eigenvectors to make them orthonormal since  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$  leads to  $\mathbf{A}\frac{\mathbf{x}}{\|\mathbf{x}\|} = \lambda\frac{\mathbf{x}}{\|\mathbf{x}\|}$  which corresponds to the same eigenvalue.

**Proof** [of Lemma 13.2] Suppose eigenvalues  $\lambda_1, \lambda_2$  correspond to eigenvectors  $\mathbf{x}_1, \mathbf{x}_2$  so that  $\mathbf{A}\mathbf{x}_1 = \lambda_1\mathbf{x}_1$  and  $\mathbf{A}\mathbf{x}_2 = \lambda_2\mathbf{x}_2$ . We have the following equality:

$$\mathbf{A}\mathbf{x}_1 = \lambda_1\mathbf{x}_1 \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_1^\top \mathbf{A} = \lambda_1\mathbf{x}_1^\top \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_1^\top \mathbf{A}\mathbf{x}_2 = \lambda_1\mathbf{x}_1^\top \mathbf{x}_2,$$

and

$$\mathbf{A}\mathbf{x}_2 = \lambda_2\mathbf{x}_2 \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_2^\top \mathbf{A} = \lambda_2\mathbf{x}_2^\top \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_2^\top \mathbf{A}\mathbf{x}_2 = \lambda_2\mathbf{x}_2^\top \mathbf{x}_2,$$

which implies  $\lambda_1\mathbf{x}_1^\top \mathbf{x}_2 = \lambda_2\mathbf{x}_2^\top \mathbf{x}_2$ . Since eigenvalues  $\lambda_1 \neq \lambda_2$ , the eigenvectors are orthogonal. ■

In the above Lemma 13.2, we prove that the eigenvectors corresponding to distinct eigenvalues of symmetric matrices are orthogonal. More generally, we prove the important theorem that eigenvectors corresponding to distinct eigenvalues of any matrix are linearly independent.

**Theorem 13.3: (Independent Eigenvector Theorem)**

If a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  has  $k$  distinct eigenvalues, then any set of  $k$  corresponding eigenvectors are linearly independent.

**Proof** [of Theorem 13.3] We will prove by induction. Firstly, we will prove that any two eigenvectors corresponding to distinct eigenvalues are linearly independent. Suppose  $\mathbf{v}_1, \mathbf{v}_2$  correspond to distinct eigenvalues  $\lambda_1$  and  $\lambda_2$  respectively. Suppose further there exists a nonzero vector  $\mathbf{x} = [x_1, x_2] \neq \mathbf{0}$  that

$$x_1\mathbf{v}_1 + x_2\mathbf{v}_2 = \mathbf{0}. \quad (13.1)$$

That is,  $\mathbf{v}_1, \mathbf{v}_2$  are linearly independent. Multiply Equation (13.1) on the left by  $\mathbf{A}$ , we get

$$x_1\lambda_1\mathbf{v}_1 + x_2\lambda_2\mathbf{v}_2 = \mathbf{0}. \quad (13.2)$$

Multiply Equation (13.1) on the left by  $\lambda_2$ , we get

$$x_1\lambda_2\mathbf{v}_1 + x_2\lambda_2\mathbf{v}_2 = \mathbf{0}. \quad (13.3)$$

Subtract Equation (13.2) from Equation (13.3) to find

$$x_1(\lambda_2 - \lambda_1)\mathbf{v}_1 = \mathbf{0}.$$

Since  $\lambda_2 \neq \lambda_1$ ,  $\mathbf{v}_1 \neq \mathbf{0}$ , we must have  $x_1 = 0$ . From Equation (13.1),  $\mathbf{v}_2 \neq \mathbf{0}$ , we must also have  $x_2 = 0$  which arrives at a contradiction. Thus  $\mathbf{v}_1, \mathbf{v}_2$  are linearly independent.

Now, suppose any  $j < k$  eigenvectors are linearly independent, if we could prove that any  $j+1$  eigenvectors are also linearly independent, we finish the proof. Suppose  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j$  are linearly independent and  $\mathbf{v}_{j+1}$  is dependent on the first  $j$  eigenvectors. That is, there exists a nonzero vector  $\mathbf{x} = [x_1, x_2, \dots, x_j] \neq \mathbf{0}$  that

$$\mathbf{v}_{j+1} = x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + \dots + x_j\mathbf{v}_j. \quad (13.4)$$

Suppose the  $j+1$  eigenvectors correspond to distinct eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_j, \lambda_{j+1}$ . Multiply Equation (13.4) on the left by  $\mathbf{A}$ , we get

$$\lambda_{j+1}\mathbf{v}_{j+1} = x_1\lambda_1\mathbf{v}_1 + x_2\lambda_2\mathbf{v}_2 + \dots + x_j\lambda_j\mathbf{v}_j. \quad (13.5)$$

Multiply Equation (13.4) on the left by  $\lambda_{j+1}$ , we get

$$\lambda_{j+1}\mathbf{v}_{j+1} = x_1\lambda_{j+1}\mathbf{v}_1 + x_2\lambda_{j+1}\mathbf{v}_2 + \dots + x_j\lambda_{j+1}\mathbf{v}_j. \quad (13.6)$$

Subtract Equation (13.6) from Equation (13.5), we find

$$x_1(\lambda_{j+1} - \lambda_1)\mathbf{v}_1 + x_2(\lambda_{j+1} - \lambda_2)\mathbf{v}_2 + \dots + x_j(\lambda_{j+1} - \lambda_j)\mathbf{v}_j = \mathbf{0}.$$

From assumption,  $\lambda_{j+1} \neq \lambda_i$  for all  $i \in \{1, 2, \dots, j\}$ , and  $\mathbf{v}_i \neq \mathbf{0}$  for all  $i \in \{1, 2, \dots, j\}$ . We must have  $x_1 = x_2 = \dots = x_j = 0$  which leads to a contradiction. Then  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j, \mathbf{v}_{j+1}$  are linearly independent. This completes the proof.  $\blacksquare$

A direct consequence of above theorem is as follows:

**Corollary 13.4: (Independent Eigenvector Theorem, CNT.)**

If a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  has  $n$  distinct eigenvalues, then any set of  $n$  corresponding eigenvectors form a basis for  $\mathbb{R}^n$ .

Symmetric Matrix Property 3 of 4

**Lemma 13.5: (Orthonormal Eigenvectors for Duplicate Eigenvalue)**

If  $\mathbf{A}$  has a duplicate eigenvalue  $\lambda_i$  with multiplicity  $k \geq 2$ , then there exist  $k$  orthonormal eigenvectors corresponding to  $\lambda_i$ .

**Proof** [of Lemma 13.5] We note that there is at least one eigenvector  $\mathbf{x}_{i1}$  corresponding to  $\lambda_i$ . And for such eigenvector  $\mathbf{x}_{i1}$ , we can always find additional  $n - 1$  orthonormal vectors  $\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n$  so that  $\{\mathbf{x}_{i1}, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n\}$  forms an orthonormal basis in  $\mathbb{R}^n$ . Put the  $\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n$  into matrix  $\mathbf{Y}_1$  and  $\{\mathbf{x}_{i1}, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n\}$  into matrix  $\mathbf{P}_1$

$$\mathbf{Y}_1 = [\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n] \quad \text{and} \quad \mathbf{P}_1 = [\mathbf{x}_{i1}, \mathbf{Y}_1].$$

We then have

$$\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 = \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 \end{bmatrix}.$$

As a result,  $\mathbf{A}$  and  $\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1$  are similar matrices such that they have the same eigenvalues since  $\mathbf{P}_1$  is nonsingular (even orthogonal here, see Lemma 8.2, p. 198). We obtain

$$\det(\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 - \lambda \mathbf{I}_n) = ^2(\lambda_i - \lambda) \det(\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 - \lambda \mathbf{I}_{n-1}).$$

If  $\lambda_i$  has multiplicity  $k \geq 2$ , then the term  $(\lambda_i - \lambda)$  occurs  $k$  times in the polynomial from the determinant  $\det(\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 - \lambda \mathbf{I}_n)$ , i.e., the term occurs  $k - 1$  times in the polynomial from  $\det(\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 - \lambda \mathbf{I}_{n-1})$ . In another word,  $\det(\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 - \lambda_i \mathbf{I}_{n-1}) = 0$  and  $\lambda_i$  is an eigenvalue of  $\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1$ .

Let  $\mathbf{B} = \mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1$ . Since  $\det(\mathbf{B} - \lambda_i \mathbf{I}_{n-1}) = 0$ , the null space of  $\mathbf{B} - \lambda_i \mathbf{I}_{n-1}$  is not none. Suppose  $(\mathbf{B} - \lambda_i \mathbf{I}_{n-1})\mathbf{n} = \mathbf{0}$ , i.e.,  $\mathbf{B}\mathbf{n} = \lambda_i \mathbf{n}$  and  $\mathbf{n}$  is an eigenvector of  $\mathbf{B}$ .

From  $\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 = \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$ , we have  $\mathbf{A} \mathbf{P}_1 \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} = \mathbf{P}_1 \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix}$ , where  $z$  is any scalar.

From the left side of this equation, we have

$$\begin{aligned} \mathbf{A} \mathbf{P}_1 \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} &= [\lambda_i \mathbf{x}_{i1}, \mathbf{A} \mathbf{Y}_1] \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} \\ &= \lambda_i z \mathbf{x}_{i1} + \mathbf{A} \mathbf{Y}_1 \mathbf{n}. \end{aligned} \tag{13.7}$$

---

2. By the fact that if matrix  $\mathbf{M}$  has a block formulation:  $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$ , then  $\det(\mathbf{M}) = \det(\mathbf{A}) \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ .

And from the right side of the equation, we have

$$\begin{aligned}
 \mathbf{P}_1 \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} &= [\mathbf{x}_{i1} \quad \mathbf{Y}_1] \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} \\
 &= [\lambda_i \mathbf{x}_{i1} \quad \mathbf{Y}_1 \mathbf{B}] \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} \\
 &= \lambda_i z \mathbf{x}_{i1} + \mathbf{Y}_1 \mathbf{B} \mathbf{n} \\
 &= \lambda_i z \mathbf{x}_{i1} + \lambda_i \mathbf{Y}_1 \mathbf{n}. \quad (\text{Since } \mathbf{B} \mathbf{n} = \lambda_i \mathbf{n})
 \end{aligned} \tag{13.8}$$

Combine Equation (13.8) and Equation (13.7), we obtain

$$\mathbf{A} \mathbf{Y}_1 \mathbf{n} = \lambda_i \mathbf{Y}_1 \mathbf{n},$$

which means  $\mathbf{Y}_1 \mathbf{n}$  is an eigenvector of  $\mathbf{A}$  corresponding to the eigenvalue  $\lambda_i$  (same eigenvalue corresponding to  $\mathbf{x}_{i1}$ ). Since  $\mathbf{Y}_1 \mathbf{n}$  is a combination of  $\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n$  which are orthonormal to  $\mathbf{x}_{i1}$ , the  $\mathbf{Y}_1 \mathbf{n}$  can be chosen to be orthonormal to  $\mathbf{x}_{i1}$ .

To conclude, if we have one eigenvector  $\mathbf{x}_{i1}$  corresponding to  $\lambda_i$  whose multiplicity is  $k \geq 2$ , we could construct the second eigenvector by choosing one vector from the null space of  $(\mathbf{B} - \lambda_i \mathbf{I}_{n-1})$  constructed above. Suppose now, we have constructed the second eigenvector  $\mathbf{x}_{i2}$  which is orthonormal to  $\mathbf{x}_{i1}$ . For such eigenvectors  $\mathbf{x}_{i1}, \mathbf{x}_{i2}$ , we can always find additional  $n-2$  orthonormal vectors  $\mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n$  so that  $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n\}$  forms an orthonormal basis in  $\mathbb{R}^n$ . Put the  $\mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n$  into matrix  $\mathbf{Y}_2$  and  $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n\}$  into matrix  $\mathbf{P}_2$ :

$$\mathbf{Y}_2 = [\mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n] \quad \text{and} \quad \mathbf{P}_2 = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{Y}_1].$$

We then have

$$\mathbf{P}_2^\top \mathbf{A} \mathbf{P}_2 = \begin{bmatrix} \lambda_i & 0 & \mathbf{0} \\ 0 & \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Y}_2^\top \mathbf{A} \mathbf{Y}_2 \end{bmatrix} = \begin{bmatrix} \lambda_i & 0 & \mathbf{0} \\ 0 & \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C} \end{bmatrix},$$

where  $\mathbf{C} = \mathbf{Y}_2^\top \mathbf{A} \mathbf{Y}_2$  such that  $\det(\mathbf{P}_2^\top \mathbf{A} \mathbf{P}_2 - \lambda \mathbf{I}_n) = (\lambda_i - \lambda)^2 \det(\mathbf{C} - \lambda \mathbf{I}_{n-2})$ . If the multiplicity of  $\lambda_i$  is  $k \geq 3$ ,  $\det(\mathbf{C} - \lambda_i \mathbf{I}_{n-2}) = 0$  and the null space of  $\mathbf{C} - \lambda_i \mathbf{I}_{n-2}$  is not none so that we can still find a vector from null space of  $\mathbf{C} - \lambda_i \mathbf{I}_{n-2}$  and  $\mathbf{C} \mathbf{n} = \lambda_i \mathbf{n}$ . Now

we can construct a vector  $\begin{bmatrix} z_1 \\ z_2 \\ \mathbf{n} \end{bmatrix} \in \mathbb{R}^n$ , where  $z_1, z_2$  are any scalar values, such that

$$\mathbf{A} \mathbf{P}_2 \begin{bmatrix} z_1 \\ z_2 \\ \mathbf{n} \end{bmatrix} = \mathbf{P}_2 \begin{bmatrix} \lambda_i & 0 & \mathbf{0} \\ 0 & \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{C} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ \mathbf{n} \end{bmatrix}.$$

Similarly, from the left side of the above equation, we will get  $\lambda_i z_1 \mathbf{x}_{i1} + \lambda_i z_2 \mathbf{x}_{i2} + \mathbf{A} \mathbf{Y}_2 \mathbf{n}$ . From the right side of the above equation, we will get  $\lambda_i z_1 \mathbf{x}_{i1} + \lambda_i z_2 \mathbf{x}_{i2} + \lambda_i \mathbf{Y}_2 \mathbf{n}$ . As a result,

$$\mathbf{A} \mathbf{Y}_2 \mathbf{n} = \lambda_i \mathbf{Y}_2 \mathbf{n},$$

where  $\mathbf{Y}_2\mathbf{n}$  is an eigenvector of  $\mathbf{A}$  and orthogonal to  $\mathbf{x}_{i1}, \mathbf{x}_{i2}$ . And it is easy to construct the eigenvector to be orthonormal to the first two.

The process can go on, and finally, we will find  $k$  orthonormal eigenvectors corresponding to  $\lambda_i$ .

Actually, the dimension of the null space of  $\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 - \lambda_i \mathbf{I}_n$  is equal to the multiplicity  $k$ . It also follows that if the multiplicity of  $\lambda_i$  is  $k$ , there cannot be more than  $k$  orthogonal eigenvectors corresponding to  $\lambda_i$ . Otherwise, it will come to the conclusion that we could find more than  $n$  orthogonal eigenvectors which leads to a contradiction. ■

The proof of the existence of the spectral decomposition is trivial from the lemmas above. Also, we can use Schur decomposition to prove the existence of it.

**Proof [of Theorem 13.1: Existence of Spectral Decomposition]** From the Schur decomposition in Theorem 12.1 (p. 236), symmetric matrix  $\mathbf{A} = \mathbf{A}^\top$  leads to  $\mathbf{Q}\mathbf{U}\mathbf{Q}^\top = \mathbf{Q}\mathbf{U}^\top\mathbf{Q}^\top$ . Then  $\mathbf{U}$  is a diagonal matrix. And this diagonal matrix actually contains eigenvalues of  $\mathbf{A}$ . All the columns of  $\mathbf{Q}$  are eigenvectors of  $\mathbf{A}$ . We conclude that all symmetric matrices are diagonalizable even with repeated eigenvalues. ■

For any matrix multiplication, we have the rank of the multiplication result no larger than the rank of the inputs. However, the symmetric matrix  $\mathbf{A}^\top \mathbf{A}$  is rather special in that the rank of  $\mathbf{A}^\top \mathbf{A}$  is equal to that of  $\mathbf{A}$  which will be used in the proof of singular value decomposition in the next section.

### Lemma 13.6: (Rank of $\mathbf{AB}$ )

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{n \times k}$ , then the matrix multiplication  $\mathbf{AB} \in \mathbb{R}^{m \times k}$  has  $\text{rank}(\mathbf{AB}) \leq \min(\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B}))$ .

**Proof [of Lemma 13.6]** For matrix multiplication  $\mathbf{AB}$ , we have

- All rows of  $\mathbf{AB}$  are combinations of rows of  $\mathbf{B}$ , the row space of  $\mathbf{AB}$  is a subset of the row space of  $\mathbf{B}$ . Thus  $\text{rank}(\mathbf{AB}) \leq \text{rank}(\mathbf{B})$ .
- All columns of  $\mathbf{AB}$  are combinations of columns of  $\mathbf{A}$ , the column space of  $\mathbf{AB}$  is a subset of the column space of  $\mathbf{A}$ . Thus  $\text{rank}(\mathbf{AB}) \leq \text{rank}(\mathbf{A})$ .

Therefore,  $\text{rank}(\mathbf{AB}) \leq \min(\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B}))$ . ■

### Symmetric Matrix Property 4 of 4

### Lemma 13.7: (Rank of Symmetric Matrices)

If  $\mathbf{A}$  is an  $n \times n$  real symmetric matrix, then  $\text{rank}(\mathbf{A}) =$  the total number of nonzero eigenvalues of  $\mathbf{A}$ . In particular,  $\mathbf{A}$  has full rank if and only if  $\mathbf{A}$  is nonsingular. Further,  $\mathcal{C}(\mathbf{A})$  is the linear space spanned by the eigenvectors of  $\mathbf{A}$  that correspond to nonzero eigenvalues.

**Proof** [of Lemma 13.7] For any symmetric matrix  $\mathbf{A}$ , we have  $\mathbf{A}$ , in spectral form, as  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$  and also  $\Lambda = \mathbf{Q}^\top\mathbf{A}\mathbf{Q}$ . Since we have shown in Lemma 13.6 that the rank of the multiplication  $\text{rank}(\mathbf{AB}) \leq \min(\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B}))$ .

- From  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , we have  $\text{rank}(\mathbf{A}) \leq \text{rank}(\mathbf{Q}\Lambda) \leq \text{rank}(\Lambda)$ ;
- From  $\Lambda = \mathbf{Q}^\top\mathbf{A}\mathbf{Q}$ , we have  $\text{rank}(\Lambda) \leq \text{rank}(\mathbf{Q}^\top\mathbf{A}) \leq \text{rank}(\mathbf{A})$ ,

The inequalities above give us a contradiction. And thus  $\text{rank}(\mathbf{A}) = \text{rank}(\Lambda)$  which is the total number of nonzero eigenvalues.

Since  $\mathbf{A}$  is nonsingular if and only if all of its eigenvalues are nonzero,  $\mathbf{A}$  has full rank if and only if  $\mathbf{A}$  is nonsingular.  $\blacksquare$

Similar to the eigenvalue decomposition, we can compute the  $m$ -th power of matrix  $\mathbf{A}$  via the spectral decomposition more efficiently.

#### Remark 13.8: $m$ -th Power

The  $m$ -th power of  $\mathbf{A}$  is  $\mathbf{A}^m = \mathbf{Q}\Lambda^m\mathbf{Q}^\top$  if the matrix  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ .

### 13.3. Uniqueness of Spectral Decomposition

Clearly, the spectral decomposition is not unique essentially because of the multiplicity of eigenvalues. One can imagine that eigenvalue  $\lambda_i$  and  $\lambda_j$  are the same for some  $1 \leq i, j \leq n$ , and interchange the corresponding eigenvectors in  $\mathbf{Q}$  will have the same results but the decompositions are different. But the *eigenspaces* (i.e., the null space  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$  for eigenvalue  $\lambda_i$ ) corresponding to each eigenvalue are fixed. So there is a unique decomposition in terms of eigenspaces and then any orthonormal basis of these eigenspaces can be chosen.

### 13.4. Other Forms, Connecting Eigenvalue Decomposition\*

In this section, we discuss other forms of the spectral decomposition under different conditions.

#### Definition 13.1: Characteristic Polynomial

For any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , the **characteristic polynomial**  $\det(\mathbf{A} - \lambda\mathbf{I})$  is given by

$$\begin{aligned}\det(\lambda\mathbf{I} - \mathbf{A}) &= \lambda^n - \gamma_{n-1}\lambda^{n-1} + \dots + \gamma_1\lambda + \gamma_0 \\ &= (\lambda - \lambda_1)^{k_1}(\lambda - \lambda_2)^{k_2} \dots (\lambda - \lambda_m)^{k_m},\end{aligned}$$

where  $\lambda_1, \lambda_2, \dots, \lambda_m$  are the distinct roots of  $\det(\lambda\mathbf{I} - \mathbf{A})$  and also the eigenvalues of  $\mathbf{A}$ , and  $k_1 + k_2 + \dots + k_m = n$ , i.e.,  $\det(\lambda\mathbf{I} - \mathbf{A})$  is a polynomial of degree  $n$  for any matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  (see proof of Lemma 13.5, p. 244).

An important multiplicity arises from the characteristic polynomial of a matrix is then defined as follows:

**Definition 13.2: Algebraic Multiplicity and Geometric Multiplicity**

Given the characteristic polynomial of matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ :

$$\det(\lambda\mathbf{I} - \mathbf{A}) = (\lambda - \lambda_1)^{k_1}(\lambda - \lambda_2)^{k_2} \dots (\lambda - \lambda_m)^{k_m}.$$

The integer  $k_i$  is called the **algebraic multiplicity** of the eigenvalue  $\lambda_i$ , i.e., the algebraic multiplicity of eigenvalue  $\lambda_i$  is equal to the multiplicity of the corresponding root of the characteristic polynomial.

The **eigenspace associated to eigenvalue**  $\lambda_i$  is defined by the null space of  $(\mathbf{A} - \lambda_i\mathbf{I})$ , i.e.,  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$ .

And the dimension of the eigenspace associated to  $\lambda_i$ ,  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$ , is called the **geometric multiplicity** of  $\lambda_i$ .

In short, we denote the algebraic multiplicity of  $\lambda_i$  by  $alg(\lambda_i)$ , and its geometric multiplicity by  $geo(\lambda_i)$ .

**Remark 13.3: Geometric Multiplicity**

Note that for matrix  $\mathbf{A}$  and the eigenspace  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$ , the dimension of the eigenspace is also the number of linearly independent eigenvectors of  $\mathbf{A}$  associated to  $\lambda_i$ , namely a basis for the eigenspace. This implies that while there are an infinite number of eigenvectors associated with each eigenvalue  $\lambda_i$ , the fact that they form a subspace (provided the zero vector is added) means that they can be described by a finite number of vectors.

By definition, the sum of the algebraic multiplicities is equal to  $n$ , but the sum of the geometric multiplicities can be strictly smaller.

**Corollary 13.4: (Multiplicity in Similar Matrices)**

Similar matrices have same algebraic multiplicities and geometric multiplicities.

**Proof** [of Corollary 13.4] In Lemma 8.2 (p. 198), we proved that the eigenvalues of similar matrices are the same, therefore, the algebraic multiplicities of similar matrices are the same as well.

Suppose  $\mathbf{A}$  and  $\mathbf{B} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}$  are similar matrices where  $\mathbf{P}$  is nonsingular. And the geometric multiplicity of an eigenvalue of  $\mathbf{A}$ , say  $\lambda$ , is  $k$ . Then there exists a set of orthogonal vectors  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$  that are the basis for the eigenspace  $\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})$  such that  $\mathbf{A}\mathbf{v}_i = \lambda\mathbf{v}_i$  for all  $i \in \{1, 2, \dots, k\}$ . Then,  $\mathbf{w}_i = \mathbf{P}\mathbf{v}_i$ 's are the eigenvectors of  $\mathbf{B}$  associated with eigenvalue  $\lambda$ . Further,  $\mathbf{w}_i$ 's are linearly independent since  $\mathbf{P}$  is nonsingular. Thus, the dimension of the eigenspace  $\mathcal{N}(\mathbf{B} - \lambda\mathbf{I})$  is at least  $k$ , that is,  $\dim(\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})) \leq \dim(\mathcal{N}(\mathbf{B} - \lambda\mathbf{I}))$ .

Similarly, there exists a set of orthogonal vectors  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k$  that are the bases for the eigenspace  $\mathcal{N}(\mathbf{B} - \lambda\mathbf{I})$ , then  $\mathbf{v}_i = \mathbf{P}^{-1}\mathbf{w}_i$  for all  $i \in \{1, 2, \dots, k\}$  are the eigenvectors of  $\mathbf{A}$  associated to  $\lambda$ . This will result in  $\dim(\mathcal{N}(\mathbf{B} - \lambda\mathbf{I})) \leq \dim(\mathcal{N}(\mathbf{A} - \lambda\mathbf{I}))$ .

Therefore, by “sandwiching”, we get  $\dim(\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})) = \dim(\mathcal{N}(\mathbf{B} - \lambda\mathbf{I}))$ , which is the equality of the geometric multiplicities, and the claim follows. ■

**Lemma 13.5: (Bounded Geometric Multiplicity)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , its geometric multiplicity is bounded by algebraic multiplicity for any eigenvalue  $\lambda_i$ :

$$\text{geo}(\lambda_i) \leq \text{alg}(\lambda_i).$$

**Proof** [of Lemma 13.5] If we can find a similar matrix  $\mathbf{B}$  of  $\mathbf{A}$  that has a specific form of the characteristic polynomial, then we complete the proof.

Suppose  $\mathbf{P}_1 = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$  contains the eigenvectors of  $\mathbf{A}$  associated with  $\lambda_i$  which are linearly independent. That is, the  $k$  vectors are bases for the eigenspace  $\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})$  and the geometric multiplicity associated with  $\lambda_i$  is  $k$ . We can expand it to  $n$  linearly independent vectors such that

$$\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2] = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k, \mathbf{v}_{k+1}, \dots, \mathbf{v}_n],$$

where  $\mathbf{P}$  is nonsingular. Then  $\mathbf{AP} = [\lambda_i \mathbf{P}_1, \mathbf{AP}_2]$ .

Construct a matrix  $\mathbf{B} = \begin{bmatrix} \lambda_i \mathbf{I}_k & \mathbf{C} \\ \mathbf{0} & \mathbf{D} \end{bmatrix}$  where  $\mathbf{AP}_2 = \mathbf{P}_1 \mathbf{C} + \mathbf{P}_2 \mathbf{D}$ , then  $\mathbf{P}^{-1} \mathbf{AP} = \mathbf{B}$  such that  $\mathbf{A}$  and  $\mathbf{B}$  are similar matrices. We can always find such  $\mathbf{C}, \mathbf{D}$  that satisfy the above condition, since  $\mathbf{v}_i$ 's are linearly independent with spanning the whole space  $\mathbb{R}^n$ , and any column of  $\mathbf{AP}_2$  is in the column space of  $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ . Therefore,

$$\begin{aligned} \det(\mathbf{A} - \lambda\mathbf{I}) &= \det(\mathbf{P}^{-1}) \det(\mathbf{A} - \lambda\mathbf{I}) \det(\mathbf{P}) && (\det(\mathbf{P}^{-1}) = 1 / \det(\mathbf{P})) \\ &= \det(\mathbf{P}^{-1}(\mathbf{A} - \lambda\mathbf{I})\mathbf{P}) && (\det(\mathbf{A}) \det(\mathbf{B}) = \det(\mathbf{AB})) \\ &= \det(\mathbf{B} - \lambda\mathbf{I}) \\ &= \det\left(\begin{bmatrix} (\lambda_i - \lambda)\mathbf{I}_k & \mathbf{C} \\ \mathbf{0} & \mathbf{D} - \lambda\mathbf{I} \end{bmatrix}\right) \\ &= (\lambda_i - \lambda)^k \det(\mathbf{D} - \lambda\mathbf{I}), \end{aligned}$$

where the last equality is from the fact that if matrix  $\mathbf{M}$  has a block formulation:  $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}$ , then  $\det(\mathbf{M}) = \det(\mathbf{A}) \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ . This implies

$$\text{geo}(\lambda_i) \leq \text{alg}(\lambda_i).$$

And we complete the proof. ■

Following from the proof of Lemma 13.5, we notice that the algebraic multiplicity and geometric multiplicity are the same for symmetric matrices. We call these matrices simple matrices.

**Definition 13.6: Simple Matrix**

When the algebraic multiplicity and geometric multiplicity are the same for a matrix, we call it a simple matrix.

**Definition 13.7: Diagonalizable**

A matrix  $\mathbf{A}$  is diagonalizable if there exists a nonsingular matrix  $\mathbf{P}$  and a diagonal matrix  $\mathbf{D}$  such that  $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ .

Eigenvalue decomposition in Theorem 11.1 and spectral decomposition in Theorem 13.1 are such kinds of matrices that are diagonalizable.

**Lemma 13.8: (Simple Matrices are Diagonalizable)**

A matrix is a simple matrix if and only if it is diagonalizable.

**Proof** [of Lemma 13.8] We will show by forward implication and backward implication separately as follows.

**Forward implication** Suppose that  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is a simple matrix, such that the algebraic and geometric multiplicities for each eigenvalue are equal. For a specific eigenvalue  $\lambda_i$ , let  $\{\mathbf{v}_1^i, \mathbf{v}_2^i, \dots, \mathbf{v}_{k_i}^i\}$  be a basis for the eigenspace  $\mathcal{N}(\mathbf{A} - \lambda_i \mathbf{I})$ , that is,  $\{\mathbf{v}_1^i, \mathbf{v}_2^i, \dots, \mathbf{v}_{k_i}^i\}$  is a set of linearly independent eigenvectors of  $\mathbf{A}$  associated to  $\lambda_i$ , where  $k_i$  is the algebraic or geometric multiplicity associated to  $\lambda_i$ :  $alg(\lambda_i) = geo(\lambda_i) = k_i$ . Suppose there are  $m$  distinct eigenvalues, since  $k_1 + k_2 + \dots + k_m = n$ , the set of eigenvectors consists of the union of  $n$  vectors. Suppose there is a set of  $x_j$ 's such that

$$\mathbf{z} = \sum_{j=1}^{k_1} x_j^1 \mathbf{v}_j^1 + \sum_{j=1}^{k_2} x_j^2 \mathbf{v}_j^2 + \dots + \sum_{j=1}^{k_m} x_j^m \mathbf{v}_j^m = \mathbf{0}. \quad (13.9)$$

Let  $\mathbf{w}^i = \sum_{j=1}^{k_i} x_j^i \mathbf{v}_j^i$ . Then  $\mathbf{w}^i$  is either an eigenvector associated to  $\lambda_i$ , or it is a zero vector. That is  $\mathbf{z} = \sum_{i=1}^m \mathbf{w}^i$  is a sum of either zero vector or an eigenvector associated with different eigenvalues of  $\mathbf{A}$ . Since eigenvectors associated with different eigenvalues are linearly independent. We must have  $\mathbf{w}^i = \mathbf{0}$  for all  $i \in \{1, 2, \dots, m\}$ . That is

$$\mathbf{w}^i = \sum_{j=1}^{k_i} x_j^i \mathbf{v}_j^i = \mathbf{0}, \quad \text{for all } i \in \{1, 2, \dots, m\}.$$

Since we assume the eigenvectors  $\mathbf{v}_j^i$ 's associated to  $\lambda_i$  are linearly independent, we must have  $x_j^i = 0$  for all  $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, k_i\}$ . Thus, the  $n$  vectors are linearly independent:

$$\{\mathbf{v}_1^1, \mathbf{v}_2^1, \dots, \mathbf{v}_{k_1}^1\}, \{\mathbf{v}_1^2, \mathbf{v}_2^2, \dots, \mathbf{v}_{k_2}^2\}, \dots, \{\mathbf{v}_1^m, \mathbf{v}_2^m, \dots, \mathbf{v}_{k_m}^m\}.$$

By eigenvalue decomposition in Theorem 11.1, matrix  $\mathbf{A}$  can be diagonalizable.

**Backward implication** Suppose  $\mathbf{A}$  is diagonalizable. That is, there exists a nonsingular matrix  $\mathbf{P}$  and a diagonal matrix  $\mathbf{D}$  such that  $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ .  $\mathbf{A}$  and  $\mathbf{D}$  are similar matrices such that they have the same eigenvalues (Lemma 8.2, p. 198), same algebraic multiplicities, and geometric multiplicities (Corollary 13.4, p. 248). It can be easily verified that a diagonal matrix has equal algebraic multiplicity and geometric multiplicity such that  $\mathbf{A}$  is a simple matrix. ■

### Remark 13.9: Equivalence on Diagonalization

From Theorem 13.3 that any eigenvectors corresponding to different eigenvalues are linearly independent, and Remark 13.3 that the geometric multiplicity is the dimension of the eigenspace. We realize, if the geometric multiplicity is equal to the algebraic multiplicity, the eigenspace can span the whole space  $\mathbb{R}^n$  if matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . So the above Lemma is equivalent to claim that if the eigenspace can span the whole space  $\mathbb{R}^n$ , then  $\mathbf{A}$  can be diagonalizable.

### Corollary 13.10

A square matrix  $\mathbf{A}$  with linearly independent eigenvectors is a simple matrix. Or if  $\mathbf{A}$  is symmetric, it is also a simple matrix.

From the eigenvalue decomposition in Theorem 11.1 (p. 230) and the spectral decomposition in Theorem 13.1 (p. 241), the proof is trivial for the corollary.

Now we are ready to show the second form of the spectral decomposition.

### Theorem 13.11: (Spectral Decomposition: The Second Form)

A simple matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as a sum of a set of idempotent matrices

$$\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{A}_i,$$

where  $\lambda_i$  for all  $i \in \{1, 2, \dots, n\}$  are eigenvalues of  $\mathbf{A}$  (duplicate possible), and also known as the **spectral values** of  $\mathbf{A}$ . Specifically, we have the following properties:

1. Idempotent:  $\mathbf{A}_i^2 = \mathbf{A}_i$  for all  $i \in \{1, 2, \dots, n\}$ ;
2. Orthogonal:  $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$  for all  $i \neq j$ ;
3. Additivity:  $\sum_{i=1}^n \mathbf{A}_i = \mathbf{I}_n$ ;
4. Rank-Additivity:  $\text{rank}(\mathbf{A}_1) + \text{rank}(\mathbf{A}_2) + \dots + \text{rank}(\mathbf{A}_n) = n$ .

**Proof** [of Theorem 13.11] Since  $\mathbf{A}$  is a simple matrix, from Lemma 13.8, there exists a nonsingular matrix  $\mathbf{P}$  and a diagonal matrix  $\mathbf{\Lambda}$  such that  $\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$  where  $\mathbf{\Lambda} =$

$\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ , and  $\lambda_i$ 's are eigenvalues of  $\mathbf{A}$  and columns of  $\mathbf{P}$  are eigenvectors of  $\mathbf{A}$ . Suppose

$$\mathbf{P} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_n] \quad \text{and} \quad \mathbf{P}^{-1} = \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \vdots \\ \mathbf{w}_n^\top \end{bmatrix}$$

are the column and row partitions of  $\mathbf{P}$  and  $\mathbf{P}^{-1}$  respectively. Then, we have

$$\mathbf{A} = \mathbf{P} \Lambda \mathbf{P}^{-1} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_n] \Lambda \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \vdots \\ \mathbf{w}_n^\top \end{bmatrix} = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{w}_i^\top.$$

Let  $\mathbf{A}_i = \mathbf{v}_i \mathbf{w}_i^\top$ , we have  $\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{A}_i$ . We realize that  $\mathbf{P}^{-1} \mathbf{P} = \mathbf{I}$  such that

$$\begin{cases} \mathbf{w}_i^\top \mathbf{v}_j = 1, & \text{if } i = j. \\ \mathbf{w}_i^\top \mathbf{v}_j = 0, & \text{if } i \neq j. \end{cases}$$

Therefore,

$$\mathbf{A}_i \mathbf{A}_j = \mathbf{v}_i \mathbf{w}_i^\top \mathbf{v}_j \mathbf{w}_j^\top = \begin{cases} \mathbf{v}_i \mathbf{w}_i^\top = \mathbf{A}_i, & \text{if } i = j. \\ \mathbf{0}, & \text{if } i \neq j. \end{cases}$$

This implies the idempotency and orthogonality of  $\mathbf{A}_i$ 's. We also notice that  $\sum_{i=1}^n \mathbf{A}_i = \mathbf{P} \mathbf{P}^{-1} = \mathbf{I}$ , that is the additivity of  $\mathbf{A}_i$ 's. The rank-additivity of the  $\mathbf{A}_i$ 's is trivial since  $\text{rank}(\mathbf{A}_i) = 1$  for all  $i \in \{1, 2, \dots, n\}$ .  $\blacksquare$

The decomposition is highly related to the Cochran's theorem in Appendix I and its application in the distribution theory of linear models (Lu, 2021d).

### Theorem 13.12: (Spectral Decomposition: The Third Form)

A simple matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with  $k$  distinct eigenvalues can be factored as a sum of a set of idempotent matrices

$$\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i,$$

where  $\lambda_i$  for all  $i \in \{1, 2, \dots, k\}$  are the distinct eigenvalues of  $\mathbf{A}$ , and also known as the spectral values of  $\mathbf{A}$ . Specifically, we have the following properties:

1. Idempotent:  $\mathbf{A}_i^2 = \mathbf{A}_i$  for all  $i \in \{1, 2, \dots, k\}$ ;
2. Orthogonal:  $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$  for all  $i \neq j$ ;
3. Additivity:  $\sum_{i=1}^k \mathbf{A}_i = \mathbf{I}_n$ ;
4. Rank-Additivity:  $\text{rank}(\mathbf{A}_1) + \text{rank}(\mathbf{A}_2) + \dots + \text{rank}(\mathbf{A}_k) = n$ .

**Proof** [of Theorem 13.12] From Theorem 13.11, we can decompose  $\mathbf{A}$  by  $\mathbf{A} = \sum_{j=1}^n \beta_j \mathbf{B}_j$ . Without loss of generality, the eigenvalues  $\beta_i$ 's are ordered such that  $\beta_1 \leq \beta_2 \leq \dots \leq \beta_n$  where

duplicate is possible. Let  $\lambda_i$ 's be the distinct eigenvalues, and  $\mathbf{A}_i$  be the sum of the  $\mathbf{B}_j$ 's associated with  $\lambda_i$ . Suppose the multiplicity of  $\lambda_i$  is  $m_i$ , and the  $\mathbf{B}_j$ 's associated to  $\lambda_i$  can be denoted as  $\{\mathbf{B}_1^i, \mathbf{B}_2^i, \dots, \mathbf{B}_{m_i}^i\}$ . Then  $\mathbf{A}_i$  can be denoted as  $\mathbf{A}_i = \sum_{j=1}^{m_i} \mathbf{B}_j^i$ . Apparently  $\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i$ .

**Idempotency**  $\mathbf{A}_i^2 = (\mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i)(\mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i) = \mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i = \mathbf{A}_i$  from the idempotency and orthogonality of  $\mathbf{B}_j^i$ 's.

**Orthogonality**  $\mathbf{A}_i \mathbf{A}_j = (\mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i)(\mathbf{B}_1^j + \mathbf{B}_2^j + \dots + \mathbf{B}_{m_j}^j) = \mathbf{0}$  from the orthogonality of the  $\mathbf{B}_j^i$ 's.

**Additivity** It is trivial that  $\sum_{i=1}^k \mathbf{A}_i = \mathbf{I}_n$ .

**Rank-Additivity**  $\text{rank}(\mathbf{A}_i) = \text{rank}(\sum_{j=1}^{m_i} \mathbf{B}_j^i) = m_i$  such that  $\text{rank}(\mathbf{A}_1) + \text{rank}(\mathbf{A}_2) + \dots + \text{rank}(\mathbf{A}_k) = m_1 + m_2 + \dots + m_k = n$ .  $\blacksquare$

### Theorem 13.13: (Spectral Decomposition: Backward Implication)

If a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with  $k$  distinct eigenvalues can be factored as a sum of a set of idempotent matrices

$$\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i,$$

where  $\lambda_i$  for all  $i \in \{1, 2, \dots, k\}$  are the distinct eigenvalues of  $\mathbf{A}$ , and

1. Idempotent:  $\mathbf{A}_i^2 = \mathbf{A}_i$  for all  $i \in \{1, 2, \dots, k\}$ ;
2. Orthogonal:  $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$  for all  $i \neq j$ ;
3. Additivity:  $\sum_{i=1}^k \mathbf{A}_i = \mathbf{I}_n$ ;
4. Rank-Additivity:  $\text{rank}(\mathbf{A}_1) + \text{rank}(\mathbf{A}_2) + \dots + \text{rank}(\mathbf{A}_k) = n$ .

Then, the matrix  $\mathbf{A}$  is a simple matrix.

**Proof** [of Corollary 13.13] Suppose  $\text{rank}(\mathbf{A}_i) = r_i$  for all  $i \in \{1, 2, \dots, k\}$ . By ULV decomposition in Theorem 4.1,  $\mathbf{A}_i$  can be factored as

$$\mathbf{A}_i = \mathbf{U}_i \begin{bmatrix} \mathbf{L}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}_i,$$

where  $\mathbf{L}_i \in \mathbb{R}^{r_i \times r_i}$ ,  $\mathbf{U}_i \in \mathbb{R}^{n \times n}$  and  $\mathbf{V}_i \in \mathbb{R}^{n \times n}$  are orthogonal matrices. Let

$$\mathbf{X}_i = \mathbf{U}_i \begin{bmatrix} \mathbf{L}_i \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{Y}_i = \begin{bmatrix} \mathbf{Y}_i \\ \mathbf{Z}_i \end{bmatrix},$$

where  $\mathbf{X}_i \in \mathbb{R}^{n \times r_i}$ , and  $\mathbf{Y}_i \in \mathbb{R}^{r_i \times n}$  is the first  $r_i$  rows of  $\mathbf{V}_i$ . Then, we have

$$\mathbf{A}_i = \mathbf{X}_i \mathbf{Y}_i.$$

This can be seen as a **reduced** ULV decomposition of  $\mathbf{A}_i$ . Appending the  $\mathbf{X}_i$ 's and  $\mathbf{Y}_i$ 's into  $\mathbf{X}$  and  $\mathbf{Y}$ ,

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k], \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_k \end{bmatrix},$$

where  $\mathbf{X} \in \mathbb{R}^{n \times n}$  and  $\mathbf{Y} \in \mathbb{R}^{n \times n}$  (from rank-additivity). By block matrix multiplication and the additivity of  $\mathbf{A}_i$ 's, we have

$$\mathbf{XY} = \sum_{i=1}^k \mathbf{X}_i \mathbf{Y}_i = \sum_{i=1}^k \mathbf{A}_i = \mathbf{I}.$$

Therefore  $\mathbf{Y}$  is the inverse of  $\mathbf{X}$ , and

$$\mathbf{YX} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_k \end{bmatrix} [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k] = \begin{bmatrix} \mathbf{Y}_1 \mathbf{X}_1 & \mathbf{Y}_1 \mathbf{X}_2 & \dots & \mathbf{Y}_1 \mathbf{X}_k \\ \mathbf{Y}_2 \mathbf{X}_1 & \mathbf{Y}_2 \mathbf{X}_2 & \dots & \mathbf{Y}_2 \mathbf{X}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}_k \mathbf{X}_1 & \mathbf{Y}_k \mathbf{X}_2 & \dots & \mathbf{Y}_k \mathbf{X}_k \end{bmatrix} = \mathbf{I},$$

such that

$$\mathbf{Y}_i \mathbf{X}_j = \begin{cases} \mathbf{I}_{r_i}, & \text{if } i = j; \\ \mathbf{0}, & \text{if } i \neq j. \end{cases}$$

This implies

$$\mathbf{A}_i \mathbf{X}_j = \begin{cases} \mathbf{X}_i, & \text{if } i = j; \\ \mathbf{0}, & \text{if } i \neq j, \end{cases} \quad \text{and} \quad \mathbf{AX}_i = \lambda_i \mathbf{X}_i.$$

Finally, we have

$$\mathbf{AX} = \mathbf{A}[\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k] = [\lambda_1 \mathbf{X}_1, \lambda_2 \mathbf{X}_2, \dots, \lambda_k \mathbf{X}_k] = \mathbf{X}\Lambda,$$

where

$$\Lambda = \begin{bmatrix} \lambda_1 \mathbf{I}_{r_1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \lambda_2 \mathbf{I}_{r_2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \lambda_k \mathbf{I}_{r_k} \end{bmatrix}$$

is a diagonal matrix. This implies  $\mathbf{A}$  can be diagonalized and from Lemma 13.8,  $\mathbf{A}$  is a simple matrix.  $\blacksquare$

#### Corollary 13.14: (Forward and Backward Spectral)

Combine Theorem 13.12 and Theorem 13.13, we can claim that matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is a simple matrix with  $k$  distinct eigenvalues if and only if it can be factored as a sum of a

set of idempotent matrices

$$\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i,$$

where  $\lambda_i$  for all  $i \in \{1, 2, \dots, k\}$  are the distinct eigenvalues of  $\mathbf{A}$ , and

1. Idempotent:  $\mathbf{A}_i^2 = \mathbf{A}_i$  for all  $i \in \{1, 2, \dots, k\}$ ;
2. Orthogonal:  $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$  for all  $i \neq j$ ;
3. Additivity:  $\sum_{i=1}^k \mathbf{A}_i = \mathbf{I}_n$ ;
4. Rank-Additivity:  $\text{rank}(\mathbf{A}_1) + \text{rank}(\mathbf{A}_2) + \dots + \text{rank}(\mathbf{A}_k) = n$ .

### 13.5. Skew-Symmetric Matrices and its Properties\*

We have introduced the spectral decomposition for symmetric matrices. A special kind of matrices that's related to symmetric is called the skew-symmetric matrices.

#### Definition 13.1: Skew-Symmetric Matrix

If matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  have the following property, then it is known as a **skew-symmetric matrix**:

$$\mathbf{A}^\top = -\mathbf{A}.$$

Note that under this definition, for the diagonal values  $a_{ii}$  for all  $i \in \{1, 2, \dots, n\}$ , we have  $a_{ii} = -a_{ii}$  which implies all the diagonal components are 0.

We have proved in Lemma 13.1 that all the eigenvalues of symmetric matrices are real. Similarly, we could show that all the eigenvalues of skew-symmetric matrices are imaginary.

#### Lemma 13.2: (Imaginary Eigenvalues)

The eigenvalues of any skew-symmetric matrix are all imaginary or zero.

**Proof** [of Lemma 13.2] Suppose eigenvalue  $\lambda$  is a complex number  $\lambda = a + ib$  where  $a, b$  are real. Its complex conjugate is  $\bar{\lambda} = a - ib$ . Same for complex eigenvector  $\mathbf{x} = \mathbf{c} + i\mathbf{d}$  and its complex conjugate  $\bar{\mathbf{x}} = \mathbf{c} - i\mathbf{d}$  where  $\mathbf{c}, \mathbf{d}$  are real vectors. We then have the following property

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}\bar{\mathbf{x}} = \bar{\lambda}\bar{\mathbf{x}} \quad \xrightarrow{\text{transpose to}} \quad \bar{\mathbf{x}}^\top \mathbf{A}^\top = \bar{\lambda}\bar{\mathbf{x}}^\top.$$

We take the dot product of the first equation with  $\bar{\mathbf{x}}$  and the last equation with  $\mathbf{x}$ :

$$\bar{\mathbf{x}}^\top \mathbf{A}\mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}, \quad \text{and} \quad \bar{\mathbf{x}}^\top \mathbf{A}^\top \mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}.$$

Then we have the equality  $-\lambda\bar{\mathbf{x}}^\top \mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}$  (since  $\mathbf{A}^\top = -\mathbf{A}$ ). Since  $\bar{\mathbf{x}}^\top \mathbf{x} = (\mathbf{c} - i\mathbf{d})^\top (\mathbf{c} + i\mathbf{d}) = \mathbf{c}^\top \mathbf{c} + \mathbf{d}^\top \mathbf{d}$  is a real number. Therefore the real part of  $\lambda$  is zero and  $\lambda$  is either imaginary or zero. ■

**Lemma 13.3: (Odd Skew-Symmetric Determinant)**

For skew-symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , if  $n$  is odd, then  $\det(\mathbf{A}) = 0$ .

**Proof** [of Lemma 13.3] When  $n$  is odd, we have

$$\det(\mathbf{A}) = \det(\mathbf{A}^\top) = \det(-\mathbf{A}) = (-1)^n \det(\mathbf{A}) = -\det(\mathbf{A}).$$

This implies  $\det(\mathbf{A}) = 0$ . ■

**Theorem 13.4: (Block-Diagonalization of Skew-Symmetric Matrices)**

A real skew-symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as

$$\mathbf{A} = \mathbf{Z} \mathbf{D} \mathbf{Z}^\top,$$

where  $\mathbf{Z}$  is an  $n \times n$  nonsingular matrix, and  $\mathbf{D}$  is a block-diagonal matrix with the following form

$$\mathbf{D} = \text{diag}\left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, 0, \dots, 0\right).$$

**Proof** [of Theorem 13.4] We will prove by recursive calculation. As usual, we will denote the entry  $(i, j)$  of matrix  $\mathbf{A}$  by  $\mathbf{A}_{ij}$ .

**Case 1).** Suppose the first row of  $\mathbf{A}$  is nonzero, we notice that  $\mathbf{EAE}^\top$  is skew-symmetric if  $\mathbf{A}$  is skew-symmetric for any matrix  $\mathbf{E}$ . This will make both the diagonals of  $\mathbf{A}$  and  $\mathbf{EAE}^\top$  are zeros, and the upper-left  $2 \times 2$  submatrix of  $\mathbf{EAE}^\top$  has the following form

$$(\mathbf{EAE}^\top)_{1:2,1:2} = \begin{bmatrix} 0 & x \\ -x & 0 \end{bmatrix}.$$

Since we suppose the first row of  $\mathbf{A}$  is nonzero, there exists a permutation matrix  $\mathbf{P}$  (Definition 0.15, p. 19), such that we will exchange the nonzero value, say  $a$ , in the first row to the second column of  $\mathbf{PAP}^\top$ . And as discussed above, the upper-left  $2 \times 2$  submatrix of  $\mathbf{PAP}^\top$  has the following form

$$(\mathbf{PAP}^\top)_{1:2,1:2} = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}.$$

Construct a nonsingular matrix  $\mathbf{M} = \begin{bmatrix} 1/a & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n-1} \end{bmatrix}$  such that the upper left  $2 \times 2$  submatrix of  $\mathbf{MPAP}^\top \mathbf{M}^\top$  has the following form

$$(\mathbf{MPAP}^\top \mathbf{M}^\top)_{1:2,1:2} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Now we finish diagonalizing the upper-left  $2 \times 2$  block. Suppose now  $(\mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top)$  above has a nonzero value, say  $b$ , in the first row with entry  $(1, j)$  for some  $j > 2$ , we can construct a nonsingular matrix  $\mathbf{L} = \mathbf{I} - b \cdot \mathbf{E}_{j2}$  where  $\mathbf{E}_{2j}$  is an all-zero matrix except the entry  $(2, j)$  is 1, such that  $(\mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top)$  will introduce 0 for the entry with value  $b$ .

### A Trivial Example

For example, suppose  $\mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top$  is a  $3 \times 3$  matrix with the following value

$$\mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top = \begin{bmatrix} 0 & 1 & b \\ -1 & 0 & \times \\ \times & \times & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{L} = \mathbf{I} - b \cdot \mathbf{E}_{j2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -b & 1 \end{bmatrix},$$

where  $j = 3$  for this specific example. This results in

$$\mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -b & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & \color{blue}{b} \\ -1 & 0 & \times \\ \times & \times & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & \color{blue}{0} \\ -1 & 0 & \times \\ \times & \times & 0 \end{bmatrix}.$$

Similarly, if the second row of  $\mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top$  contains a nonzero value, say  $c$ , we could construct a nonsingular matrix  $\mathbf{K} = \mathbf{I} + c \cdot \mathbf{E}_{j1}$  such that  $\mathbf{K} \mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top \mathbf{K}^\top$  will introduce 0 for the entry with value  $c$ .

### A Trivial Example

For example, suppose  $\mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top$  is a  $3 \times 3$  matrix with the following value

$$\mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & c \\ \times & \times & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{K} = \mathbf{I} + c \cdot \mathbf{E}_{j1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c & 0 & 1 \end{bmatrix},$$

where  $j = 3$  for this specific example. This results in

$$\mathbf{K} \mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top \mathbf{K}^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & \color{blue}{c} \\ \times & \times & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & \color{blue}{0} \\ \times & \times & 0 \end{bmatrix}.$$

Since we have shown that  $\mathbf{K} \mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top \mathbf{K}^\top$  is also skew-symmetric, then, it is actually

$$\mathbf{K} \mathbf{L} \mathbf{M} \mathbf{P} \mathbf{A} \mathbf{P}^\top \mathbf{M}^\top \mathbf{L}^\top \mathbf{K}^\top = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & \color{blue}{0} \\ \color{red}{0} & \color{red}{0} & 0 \end{bmatrix},$$

so that we do not need to tackle the first 2 columns of the above equation.

Apply this process for the bottom-right  $(n-2) \times (n-2)$  submatrix, we will complete the proof.

**Case 2).** Suppose the first row of  $\mathbf{A}$  is zero, a permutation matrix to put the first row into the last row and apply the process in case 1 to finish the proof. ■

From the block-diagonalization of skew-symmetric matrices above, we could easily find that the rank of a skew-symmetric matrix is even. And we could prove the determinant of skew-symmetric with even order is nonnegative as follows.

**Lemma 13.5: (Even Skew-Symmetric Determinant)**

For skew-symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , if  $n$  is even, then  $\det(\mathbf{A}) \geq 0$ .

**Proof** [of Lemma 13.5] By Theorem 13.4, we could block-diagonalize  $\mathbf{A} = \mathbf{ZDZ}^\top$  such that

$$\det(\mathbf{A}) = \det(\mathbf{ZDZ}^\top) = \det(\mathbf{Z})^2 \det(\mathbf{D}) \geq 0.$$

This completes the proof. ■

## 13.6. Applications

### 13.6.1 Application: Eigenvalue of Projection Matrix

In Section 3.20.1 (p. 125), we introduced the QR decomposition can be applied to solve the least squares problem, where we consider the overdetermined system  $\mathbf{Ax} = \mathbf{b}$  with  $\mathbf{A} \in \mathbb{R}^{m \times n}$  being the data matrix,  $\mathbf{b} \in \mathbb{R}^m$  with  $m > n$  being the observation matrix. Normally  $\mathbf{A}$  will have full column rank since the data from real work has a large chance to be unrelated. And the least squares solution is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  for minimizing  $\|\mathbf{Ax} - \mathbf{b}\|^2$ , where  $\mathbf{A}^\top \mathbf{A}$  is invertible since  $\mathbf{A}$  has full column rank and  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$ . The recovered observation matrix is then  $\hat{\mathbf{b}} = \mathbf{Ax}_{LS} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ .  $\mathbf{b}$  may not be in the column space of  $\mathbf{A}$ , but the recovered  $\hat{\mathbf{b}}$  is in this column space. We then define such matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  to be a projection matrix <sup>3</sup>, i.e., projecting  $\mathbf{b}$  onto the column space of  $\mathbf{A}$ . Or, it is also known as hat matrix, since we put a hat on  $\mathbf{b}$ . It can be easily verified the projection matrix is symmetric and idempotent (i.e.,  $\mathbf{H}^2 = \mathbf{H}$ ).

**Remark 13.1: Column Space of Projection Matrix**

We notice that the hat matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  is to project any vector in  $\mathbb{R}^m$  into the column space of  $\mathbf{A}$ . That is,  $\mathbf{Hy} \in \mathcal{C}(\mathbf{A})$ . Notice again  $\mathbf{Hy}$  is the nothing but a combination of the columns of  $\mathbf{H}$ , thus  $\mathcal{C}(\mathbf{H}) = \mathcal{C}(\mathbf{A})$ .

In general, for any projection matrix  $\mathbf{H}$  to project vector onto subspace  $\mathcal{V}$ , then  $\mathcal{C}(\mathbf{H}) = \mathcal{V}$ . More formally, in a mathematical language, this property can be proved by SVD.

---

<sup>3</sup>. A detailed analysis of orthogonal projection is provided in Appendix D (p. 432).

We now show that for any projection matrix, it has specific eigenvalues. See Appendix D.2 for a detailed discussion on the orthogonal projection.

**Proposition 13.2: (Eigenvalue of Projection Matrix)**

The only possible eigenvalues of a projection matrix are 0 and 1.

**Proof** [of Proposition 13.2] Since  $\mathbf{H}$  is symmetric, we have spectral decomposition  $\mathbf{H} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . From the idempotent property, we have

$$\begin{aligned} (\mathbf{Q}\Lambda\mathbf{Q}^\top)^2 &= \mathbf{Q}\Lambda\mathbf{Q}^\top \\ \mathbf{Q}\Lambda^2\mathbf{Q}^\top &= \mathbf{Q}\Lambda\mathbf{Q}^\top \\ \Lambda^2 &= \Lambda \\ \lambda_i^2 &= \lambda_i, \end{aligned}$$

Therefore, the only possible eigenvalues for  $\mathbf{H}$  are 0 and 1. ■

This property of the projection matrix is important for the analysis of distribution theory for linear models. See (Lu, 2021d) for more details. Following from the eigenvalue of the projection matrix, it can also give rise to the perpendicular projection  $\mathbf{I} - \mathbf{H}$ .

**Proposition 13.3: (Project onto  $\mathcal{V}^\perp$ )**

Let  $\mathcal{V}$  be a subspace and  $\mathbf{H}$  be a projection onto  $\mathcal{V}$ . Then  $\mathbf{I} - \mathbf{H}$  is the projection matrix onto  $\mathcal{V}^\perp$ .

**Proof** [of Proposition 13.3] First,  $(\mathbf{I} - \mathbf{H})$  is symmetric,  $(\mathbf{I} - \mathbf{H})^\top = \mathbf{I} - \mathbf{H}^\top = \mathbf{I} - \mathbf{H}$  since  $\mathbf{H}$  is symmatrix. And

$$(\mathbf{I} - \mathbf{H})^2 = \mathbf{I}^2 - \mathbf{IH} - \mathbf{HI} + \mathbf{H}^2 = \mathbf{I} - \mathbf{H}.$$

Thus  $\mathbf{I} - \mathbf{H}$  is a projection matrix. By spectral theorem again, let  $\mathbf{H} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . Then  $\mathbf{I} - \mathbf{H} = \mathbf{QQ}^\top - \mathbf{Q}\Lambda\mathbf{Q}^\top = \mathbf{Q}(\mathbf{I} - \Lambda)\mathbf{Q}^\top$ . Hence the column space of  $\mathbf{I} - \mathbf{H}$  is spanned by the eigenvectors of  $\mathbf{H}$  corresponding to the zero eigenvalues of  $\mathbf{H}$  (by Proposition 13.2, p. 259), which coincides with  $\mathcal{V}^\perp$ . ■

Again, for a detailed analysis of the origin of the projection matrix and results behind the projection matrix, we highly recommend the readers refer to Appendix D.2 although it is not the main interest of matrix decomposition results.

### 13.6.2 Application: An Alternative Definition on PD and PSD of Matrices

In Definition 2.1 (p. 56), we defined the positive definite matrices and positive semidefinite matrices by the quadratic form of the matrices. We here prove that a symmetric matrix is positive definite if and only if all eigenvalues are positive.

**Lemma 13.4: (Eigenvalues of PD and PSD Matrices)**

A matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive definite (PD) if and only if  $\mathbf{A}$  has only positive eigenvalues. And a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive semidefinite (PSD) if and only if  $\mathbf{A}$  has only nonnegative eigenvalues.

**Proof** [of Lemma 13.4] We will prove by forward implication and reverse implication separately as follows.

**Forward implication:** Suppose  $\mathbf{A}$  is PD, then for any eigenvalue  $\lambda$  and its corresponding eigenvector  $\mathbf{v}$  of  $\mathbf{A}$ , we have  $\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$ . Thus

$$\mathbf{v}^\top \mathbf{A}\mathbf{v} = \lambda \|\mathbf{v}\|^2 > 0.$$

This implies  $\lambda > 0$ .

**Reverse implication:** Conversely, suppose the eigenvalues are positive. By spectral decomposition of  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . If  $\mathbf{x}$  is a nonzero vector, let  $\mathbf{y} = \mathbf{Q}^\top \mathbf{x}$ , we have

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \mathbf{x}^\top (\mathbf{Q}\Lambda\mathbf{Q}^\top)\mathbf{x} = (\mathbf{x}^\top \mathbf{Q})\Lambda(\mathbf{Q}^\top \mathbf{x}) = \mathbf{y}^\top \Lambda \mathbf{y} = \sum_{i=1}^n \lambda_i y_i^2 > 0.$$

That is,  $\mathbf{A}$  is PD.

Analogously, we can prove the second part of the claim. ■

**Theorem 13.5: (Nonsingular Factor of PSD and PD Matrices)**

A real symmetric matrix  $\mathbf{A}$  is PSD if and only if  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ , and is PD if and only if  $\mathbf{P}$  is nonsingular.

**Proof** [of Theorem 13.5] For the first part, we will prove by forward implication and reverse implication separately as follows.

**Forward implication:** Suppose  $\mathbf{A}$  is PSD, its spectral decomposition is given by  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . Since eigenvalues of PSD matrices are nonnegative, we can decompose  $\Lambda = \Lambda^{1/2}\Lambda^{1/2}$ . Let  $\mathbf{P} = \Lambda^{1/2}\mathbf{Q}^\top$ , we can decompose  $\mathbf{A}$  by  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ .

**Reverse implication:** If  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ , then all eigenvalues of  $\mathbf{A}$  are nonnegative since for any eigenvalues  $\lambda$  and its corresponding eigenvector  $\mathbf{v}$  of  $\mathbf{A}$ , we have

$$\lambda = \frac{\mathbf{v}^\top \mathbf{A}\mathbf{v}}{\mathbf{v}^\top \mathbf{v}} = \frac{\mathbf{v}^\top \mathbf{P}^\top \mathbf{P}\mathbf{v}}{\mathbf{v}^\top \mathbf{v}} = \frac{\|\mathbf{P}\mathbf{v}\|^2}{\|\mathbf{v}\|^2} \geq 0.$$

This implies  $\mathbf{A}$  is PSD by Lemma 13.4.

Similarly, we can prove the second part for PD matrices where the positive definiteness will result in the nonsingular  $\mathbf{P}$  and the nonsingular  $\mathbf{P}$  will result in the positiveness of the

eigenvalues.<sup>4</sup>

■

### 13.6.3 Proof for Semidefinite Rank-Revealing Decomposition

In this section, we provide a proof for Theorem 2.2 (p. 73), the existence of the rank-revealing decomposition for positive semidefinite matrix.

**Proof** [of Theorem 2.2] The proof is a consequence of the nonsingular factor of PSD matrices (Theorem 13.5, p. 260) and the existence of column-pivoted QR decomposition (Theorem 3.1, p. 101).

By Theorem 13.5, the nonsingular factor of PSD matrix  $\mathbf{A}$  is given by  $\mathbf{A} = \mathbf{Z}^\top \mathbf{Z}$ , where  $\mathbf{Z} = \Lambda^{1/2} \mathbf{Q}^\top$  and  $\mathbf{A} = \mathbf{Q} \Lambda \mathbf{Q}^\top$  is the spectral decomposition of  $\mathbf{A}$ .

By Lemma 13.7, the rank of matrix  $\mathbf{A}$  is the number of nonzero eigenvalues (here the number of positive eigenvalues since  $\mathbf{A}$  is PSD). Therefore only  $r$  components in  $\Lambda^{1/2}$  are nonzero, and  $\mathbf{Z} = \Lambda^{1/2} \mathbf{Q}^\top$  contains only  $r$  independent columns, i.e.,  $\mathbf{Z}$  is of rank  $r$ . By column-pivoted QR decomposition, we have

$$\mathbf{ZP} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where  $\mathbf{P}$  is a permutation matrix,  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is upper triangular with positive diagonals, and  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ . Therefore

$$\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{P}^\top \mathbf{Z}^\top \mathbf{ZP} = \begin{bmatrix} \mathbf{R}_{11}^\top & \mathbf{0} \\ \mathbf{R}_{12}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Let

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

we find the rank-revealing decomposition for semidefinite matrix  $\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{R}^\top \mathbf{R}$ . ■

This decomposition is produced by using complete pivoting, which at each stage permutes the largest diagonal element in the active submatrix into the pivot position. The procedure is similar to the partial pivoting discussed in Section 1.13.1 (p. 48).

### 13.6.4 Application: Cholesky Decomposition via the QR Decomposition and the Spectral Decomposition

In this section, we provide another proof for the existence of the Cholesky decomposition.

#### Theorem 13.6: (Cholesky Decomposition: A Simpler Version of Theorem 2.1)

Every positive definite matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as

$$\mathbf{A} = \mathbf{R}^\top \mathbf{R},$$

<sup>4</sup>. See also wiki page: [https://en.wikipedia.org/wiki/Sylvester's\\_criterion](https://en.wikipedia.org/wiki/Sylvester's_criterion).

where  $\mathbf{R}$  is an upper triangular matrix with positive diagonals.

**Proof** [of Theorem 13.6] From Theorem 13.5, the PD matrix  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$  where  $\mathbf{P}$  is a nonsingular matrix. Then, the QR decomposition of  $\mathbf{P}$  is given by  $\mathbf{P} = \mathbf{QR}$ . This implies

$$\mathbf{A} = \mathbf{P}^\top \mathbf{P} = \mathbf{R}^\top \mathbf{Q}^\top \mathbf{Q} \mathbf{R} = \mathbf{R}^\top \mathbf{R},$$

where we notice that the form is very similar to the Cholesky decomposition except that we do not claim the  $\mathbf{R}$  has only positive diagonal values. From Algorithm 12, the existence of QR decomposition via the Gram-Schmidt process, we realize that the diagonals of  $\mathbf{R}$  are nonnegative, and if  $\mathbf{P}$  is nonsingular, the diagonals of  $\mathbf{R}$  are also positive. ■

The proof for the above theorem is a consequence of the existence of both the QR decomposition and the spectral decomposition. Thus, the existence of Cholesky decomposition can be proved via the QR decomposition and the spectral decomposition in this sense.

### 13.6.5 Application: Unique Power Decomposition of Positive Definite Matrices

#### Theorem 13.7: (Unique Power Decomposition of PD Matrices)

Any  $n \times n$  positive matrix  $\mathbf{A}$  can be **uniquely** factored as a product of a positive definite matrix  $\mathbf{B}$  such that  $\mathbf{A} = \mathbf{B}^2$ .

**Proof** [of Theorem 13.7] We first prove that there exists such positive definite matrix  $\mathbf{B}$  so that  $\mathbf{A} = \mathbf{B}^2$ .

**Existence** Since  $\mathbf{A}$  is PD which is also symmetric, the spectral decomposition of  $\mathbf{A}$  is given by  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . Since eigenvalues of PD matrices are positive by Lemma 13.4, the square root of  $\Lambda$  exists. We can define  $\mathbf{B} = \mathbf{Q}\Lambda^{1/2}\mathbf{Q}^\top$  such that  $\mathbf{A} = \mathbf{B}^2$  where  $\mathbf{B}$  is apparently PD.

**Uniqueness** Suppose such factorization is not unique, then there exist two of this decomposition such that

$$\mathbf{A} = \mathbf{B}_1^2 = \mathbf{B}_2^2,$$

where  $\mathbf{B}_1$  and  $\mathbf{B}_2$  are both PD. The spectral decompositions of them are given by

$$\mathbf{B}_1 = \mathbf{Q}_1 \Lambda_1 \mathbf{Q}_1^\top, \quad \text{and} \quad \mathbf{B}_2 = \mathbf{Q}_2 \Lambda_2 \mathbf{Q}_2^\top.$$

We notice that  $\Lambda_1^2$  and  $\Lambda_2^2$  contains the eigenvalues of  $\mathbf{A}$ , and both eigenvalues of  $\mathbf{B}_1$  and  $\mathbf{B}_2$  contained in  $\Lambda_1$  and  $\Lambda_2$  are positive (since  $\mathbf{B}_1$  and  $\mathbf{B}_2$  are both PD). Without loss of generality, we suppose  $\Lambda_1 = \Lambda_2 = \Lambda^{1/2}$ , and  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  such that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ . By  $\mathbf{B}_1^2 = \mathbf{B}_2^2$ , we have

$$\mathbf{Q}_1 \Lambda \mathbf{Q}_1^\top = \mathbf{Q}_2 \Lambda \mathbf{Q}_2^\top \quad \text{leads to} \quad \mathbf{Q}_2^\top \mathbf{Q}_1 \Lambda = \Lambda \mathbf{Q}_2^\top \mathbf{Q}_1.$$

Let  $\mathbf{Z} = \mathbf{Q}_2^\top \mathbf{Q}_1$ , this implies  $\Lambda$  and  $\mathbf{Z}$  commute, and  $\mathbf{Z}$  must be a block diagonal matrix whose partitioning conforms to the block structure of  $\Lambda$ . This results in  $\Lambda^{1/2} = \mathbf{Z} \Lambda^{1/2} \mathbf{Z}^\top$  and

$$\mathbf{B}_2 = \mathbf{Q}_2 \Lambda^{1/2} \mathbf{Q}_2^\top = \mathbf{Q}_2 \mathbf{Q}_2^\top \mathbf{Q}_1 \Lambda^{1/2} \mathbf{Q}_1^\top \mathbf{Q}_2 \mathbf{Q}_2^\top = \mathbf{B}_1.$$

■

This completes the proof.

Similarly, we could prove the unique decomposition of PSD matrix  $\mathbf{A} = \mathbf{B}^2$  where  $\mathbf{B}$  is PSD. A more detailed discussion on this topic can be referred to (Koeber and Schäfer, 2006).

**Decomposition for PD matrices** To conclude, for PD matrix  $\mathbf{A}$ , we can factor it into  $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$  where  $\mathbf{R}$  is an upper triangular matrix with positive diagonals as shown in Theorem 2.1 by Cholesky decomposition,  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$  where  $\mathbf{P}$  is nonsingular in Theorem 13.5, and  $\mathbf{A} = \mathbf{B}^2$  where  $\mathbf{B}$  is PD in Theorem 13.7.

## Chapter 14

# Singular Value Decomposition (SVD)

### Contents

---

|             |                                                                                            |            |
|-------------|--------------------------------------------------------------------------------------------|------------|
| <b>14.1</b> | <b>Singular Value Decomposition</b>                                                        | <b>265</b> |
| <b>14.2</b> | <b>Existence of the SVD</b>                                                                | <b>266</b> |
| <b>14.3</b> | <b>Properties of the SVD</b>                                                               | <b>269</b> |
| 14.3.1      | Four Subspaces in SVD                                                                      | 269        |
| 14.3.2      | SVD-Related Orthogonal Projections                                                         | 270        |
| 14.3.3      | Relationship between Singular Values and Determinant                                       | 271        |
| 14.3.4      | Orthogonal Equivalence                                                                     | 271        |
| 14.3.5      | SVD for QR                                                                                 | 272        |
| 14.3.6      | Interlacing Property                                                                       | 272        |
| <b>14.4</b> | <b>Computing the SVD</b>                                                                   | <b>273</b> |
| 14.4.1      | Randomized Method for Computing the SVD Approximately                                      | 273        |
| <b>14.5</b> | <b>Polar Decomposition</b>                                                                 | <b>275</b> |
| <b>14.6</b> | <b>Generalized Singular Value Decomposition (GSVD)*</b>                                    | <b>276</b> |
| 14.6.1      | CS Decomposition                                                                           | 276        |
| 14.6.2      | Generalized Singular Value Decomposition (GSVD)                                            | 277        |
| <b>14.7</b> | <b>Applications</b>                                                                        | <b>278</b> |
| 14.7.1      | Application: Least Squares via SVD for Rank Deficient Matrices                             | 278        |
| 14.7.2      | Application: Least Squares with Norm Ratio Method                                          | 280        |
| 14.7.3      | Application: Principal Component Analysis (PCA) via the Spectral Decomposition and the SVD | 281        |
| 14.7.4      | Application: Low-Rank Approximation                                                        | 284        |

---

### 14.1. Singular Value Decomposition

In eigenvalue decomposition, we factor the matrix into a diagonal matrix. However, this is not always true. If  $\mathbf{A}$  does not have linearly independent eigenvectors, such diagonalization does not exist. The singular value decomposition (SVD) fills this gap. Instead of factoring the matrix into an eigenvector matrix, SVD gives rise to two orthogonal matrices. We provide the result of SVD in the following theorem and we will discuss the existence of SVD in the next sections.

#### Theorem 14.1: (Reduced SVD for Rectangular Matrices)

For every real  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$ , then matrix  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top,$$

where  $\Sigma \in \mathbb{R}^{r \times r}$  is a diagonal matrix  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$  with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$  and

- $\sigma_i$ 's are the nonzero **singular values** of  $\mathbf{A}$ , in the meantime, they are the (positive) square roots of the nonzero **eigenvalues** of  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A} \mathbf{A}^\top$ .
- Columns of  $\mathbf{U} \in \mathbb{R}^{m \times r}$  contain the  $r$  eigenvectors of  $\mathbf{A} \mathbf{A}^\top$  corresponding to the  $r$  nonzero eigenvalues of  $\mathbf{A} \mathbf{A}^\top$ .
- Columns of  $\mathbf{V} \in \mathbb{R}^{n \times r}$  contain the  $r$  eigenvectors of  $\mathbf{A}^\top \mathbf{A}$  corresponding to the  $r$  nonzero eigenvalues of  $\mathbf{A}^\top \mathbf{A}$ .
- Moreover, the columns of  $\mathbf{U}$  and  $\mathbf{V}$  are called the **left and right singular vectors** of  $\mathbf{A}$ , respectively.
- Further, the columns of  $\mathbf{U}$  and  $\mathbf{V}$  are orthonormal (by Spectral Theorem 13.1, p. 241).

In particular, we can write out the matrix decomposition by the sum of outer products of vectors  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ , which is a sum of  $r$  rank-one matrices.

If we append additional  $m - r$  silent columns that are orthonormal to the  $r$  eigenvectors of  $\mathbf{A} \mathbf{A}^\top$ , just like the silent columns in the QR decomposition, we will have an orthogonal matrix  $\mathbf{U} \in \mathbb{R}^{m \times m}$ . Similar situation for the columns of  $\mathbf{V}$ . We then illustrate the full SVD for matrices in the following theorem. We formulate the difference between reduced and full SVD in the blue text.

#### Theorem 14.2: (Full SVD for Rectangular Matrices)

For every real  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$ , then matrix  $\mathbf{A}$  can be factored as

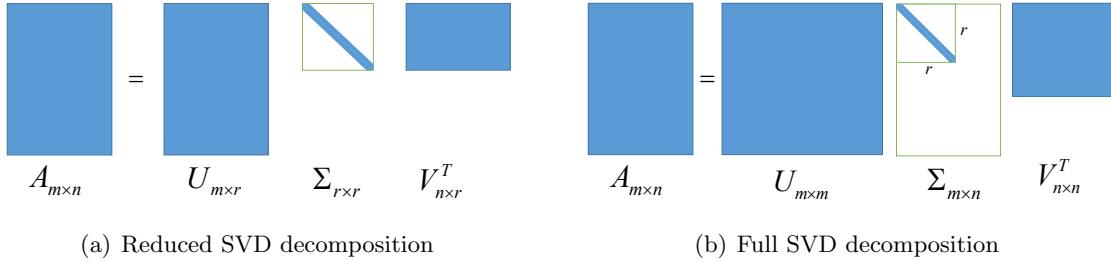
$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top,$$

where the left-upper side of  $\Sigma \in \mathbb{R}^{m \times n}$  is a diagonal matrix, that is  $\Sigma = \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$  where  $\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{r \times r}$  with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r$  and

- $\sigma_i$ 's are the nonzero **singular values** of matrix  $\mathbf{A}$ , in the meantime, they are the (positive) square roots of the nonzero **eigenvalues** of  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A} \mathbf{A}^\top$ .
- $\mathbf{U} \in \mathbb{R}^{m \times m}$  contains the  $r$  eigenvectors of  $\mathbf{A} \mathbf{A}^\top$  corresponding to the  $r$  nonzero eigenvalues of  $\mathbf{A} \mathbf{A}^\top$  and  $m - r$  extra orthonormal vectors from  $\mathcal{N}(\mathbf{A}^\top)$ .
- $\mathbf{V} \in \mathbb{R}^{n \times n}$  contains the  $r$  eigenvectors of  $\mathbf{A}^\top \mathbf{A}$  corresponding to the  $r$  nonzero eigenvalues of  $\mathbf{A}^\top \mathbf{A}$  and  $n - r$  extra orthonormal vectors from  $\mathcal{N}(\mathbf{A})$ .
- Moreover, the columns of  $\mathbf{U}$  and  $\mathbf{V}$  are called the **left and right singular vectors** of  $\mathbf{A}$ , respectively.
- Further, the columns of  $\mathbf{U}$  and  $\mathbf{V}$  are orthonormal (by Spectral Theorem 13.1, p. 241), and  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal matrices.

In particular, we can write the matrix decomposition by the sum of outer products of vectors  $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^\top = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ , which is a sum of  $r$  rank-one matrices.

The comparison between the reduced and the full SVD is shown in Figure 14.1 where white entries are zero and blue entries are not necessarily zero.



**Figure 14.1:** Comparison between the reduced and full SVD.

## 14.2. Existence of the SVD

To prove the existence of the SVD, we need to use the following lemmas. We mentioned that the singular values are the square roots of the eigenvalues of  $\mathbf{A}^\top \mathbf{A}$ . While, negative values do not have square roots such that the eigenvalues must be nonnegative.

### Lemma 14.1: (Nonnegative Eigenvalues of $\mathbf{A}^\top \mathbf{A}$ )

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{A}^\top \mathbf{A}$  has nonnegative eigenvalues.

**Proof** [of Lemma 14.1] For eigenvalue and its corresponding eigenvector  $\lambda, \mathbf{x}$  of  $\mathbf{A}^\top \mathbf{A}$ , we have

$$\mathbf{A}^\top \mathbf{A} \mathbf{x} = \lambda \mathbf{x} \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} = \lambda \mathbf{x}^\top \mathbf{x}.$$

Since  $\mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} = \|\mathbf{A} \mathbf{x}\|^2 \geq 0$  and  $\mathbf{x}^\top \mathbf{x} \geq 0$ . We then have  $\lambda \geq 0$ . ■

Since  $\mathbf{A}^\top \mathbf{A}$  has nonnegative eigenvalues, we then can define the singular value  $\sigma \geq 0$  of  $\mathbf{A}$  such that  $\sigma^2$  is the eigenvalue of  $\mathbf{A}^\top \mathbf{A}$ , i.e.,  $\boxed{\mathbf{A}^\top \mathbf{A} \mathbf{v} = \sigma^2 \mathbf{v}}$ . This is essential to the existence of the SVD.

We have shown in Lemma 13.6 (p. 246) that  $\text{rank}(\mathbf{AB}) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\}$ . However, the symmetric matrix  $\mathbf{A}^\top \mathbf{A}$  is rather special in that the rank of  $\mathbf{A}^\top \mathbf{A}$  is equal to  $\text{rank}(\mathbf{A})$ . And we now prove it.

**Lemma 14.2: (Rank of  $\mathbf{A}^\top \mathbf{A}$ )**

$\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A}$  have same rank.

**Proof** [of Lemma 14.2] Let  $\mathbf{x} \in \mathcal{N}(\mathbf{A})$ , we have

$$\mathbf{Ax} = \mathbf{0} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}^\top \mathbf{Ax} = \mathbf{0},$$

i.e.,  $\mathbf{x} \in \mathcal{N}(\mathbf{A})$  leads to  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ , therefore  $\mathcal{N}(\mathbf{A}) \subseteq \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ .

Further, let  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ , we have

$$\mathbf{A}^\top \mathbf{Ax} = \mathbf{0} \xrightarrow{\text{leads to}} \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} = 0 \xrightarrow{\text{leads to}} \|\mathbf{Ax}\|^2 = 0 \xrightarrow{\text{leads to}} \mathbf{Ax} = \mathbf{0},$$

i.e.,  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$  leads to  $\mathbf{x} \in \mathcal{N}(\mathbf{A})$ , therefore  $\mathcal{N}(\mathbf{A}^\top \mathbf{A}) \subseteq \mathcal{N}(\mathbf{A})$ .

As a result, by “sandwiching”, it follows that

$$\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^\top \mathbf{A}) \quad \text{and} \quad \dim(\mathcal{N}(\mathbf{A})) = \dim(\mathcal{N}(\mathbf{A}^\top \mathbf{A})).$$

By the fundamental theorem of linear algebra in Appendix B (p. 427),  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A}$  have the same rank. ■

Apply the observation to  $\mathbf{A}^\top$ , we can also prove that  $\mathbf{AA}^\top$  and  $\mathbf{A}$  have the same rank:

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{AA}^\top).$$

In the form of the SVD, we claimed the matrix  $\mathbf{A}$  is a sum of  $r$  rank-one matrices where  $r$  is the number of nonzero singular values. And the number of nonzero singular values is actually the rank of the matrix.

**Lemma 14.3: (The Number of Nonzero Singular Values Equals the Rank)**

The number of nonzero singular values of matrix  $\mathbf{A}$  equals the rank of  $\mathbf{A}$ .

**Proof** [of Lemma 14.3] The rank of any symmetric matrix (here  $\mathbf{A}^\top \mathbf{A}$ ) equals the number of nonzero eigenvalues (with repetitions) by Lemma 13.7 (p. 246). So the number of nonzero singular values equals the rank of  $\mathbf{A}^\top \mathbf{A}$ . By Lemma 14.2, the number of nonzero singular values equals the rank of  $\mathbf{A}$ . ■

We are now ready to prove the existence of the SVD.

**Proof [of Theorem 14.1: Existence of the SVD]** Since  $\mathbf{A}^\top \mathbf{A}$  is a symmetric matrix, by Spectral Theorem 13.1 (p. 241) and Lemma 14.1, there exists an orthogonal matrix  $\mathbf{V}$  such that

$$\boxed{\mathbf{A}^\top \mathbf{A} = \mathbf{V} \boldsymbol{\Sigma}^2 \mathbf{V}^\top},$$

where  $\boldsymbol{\Sigma}$  is a diagonal matrix containing the singular values of  $\mathbf{A}$ , i.e.,  $\boldsymbol{\Sigma}^2$  contains the eigenvalues of  $\mathbf{A}^\top \mathbf{A}$ . Specifically,  $\boldsymbol{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$  and  $\{\sigma_1^2, \sigma_2^2, \dots, \sigma_r^2\}$  are the nonzero eigenvalues of  $\mathbf{A}^\top \mathbf{A}$  with  $r$  being the rank of  $\mathbf{A}$ . I.e.,  $\{\sigma_1, \dots, \sigma_r\}$  are the singular values of  $\mathbf{A}$ . In this case,  $\mathbf{V} \in \mathbb{R}^{n \times r}$ . Now we are into the central part.

Start from  $\boxed{\mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i}$ ,  $\forall i \in \{1, 2, \dots, r\}$ , i.e., the eigenvector  $\mathbf{v}_i$  of  $\mathbf{A}^\top \mathbf{A}$  corresponding to  $\sigma_i^2$ :

1. Multiply both sides by  $\mathbf{v}_i^\top$ :

$$\mathbf{v}_i^\top \mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i^\top \mathbf{v}_i \quad \xrightarrow{\text{leads to}} \quad \|\mathbf{A} \mathbf{v}_i\|^2 = \sigma_i^2 \quad \xrightarrow{\text{leads to}} \quad \|\mathbf{A} \mathbf{v}_i\| = \sigma_i$$

2. Multiply both sides by  $\mathbf{A}$ :

$$\mathbf{A} \mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \sigma_i^2 \mathbf{A} \mathbf{v}_i \quad \xrightarrow{\text{leads to}} \quad \mathbf{A} \mathbf{A}^\top \frac{\mathbf{A} \mathbf{v}_i}{\sigma_i} = \sigma_i^2 \frac{\mathbf{A} \mathbf{v}_i}{\sigma_i} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A} \mathbf{A}^\top \mathbf{u}_i = \sigma_i^2 \mathbf{u}_i$$

where we notice that this form can find the eigenvector of  $\mathbf{A} \mathbf{A}^\top$  corresponding to  $\sigma_i^2$  which is  $\mathbf{A} \mathbf{v}_i$ . Since the length of  $\mathbf{A} \mathbf{v}_i$  is  $\sigma_i$ , we then define  $\mathbf{u}_i = \frac{\mathbf{A} \mathbf{v}_i}{\sigma_i}$  with norm 1.

These  $\mathbf{u}_i$ 's are orthogonal because  $(\mathbf{A} \mathbf{v}_i)^\top (\mathbf{A} \mathbf{v}_j) = \mathbf{v}_i^\top \mathbf{A}^\top \mathbf{A} \mathbf{v}_j = \sigma_j^2 \mathbf{v}_i^\top \mathbf{v}_j = 0$ . That is

$$\boxed{\mathbf{A} \mathbf{A}^\top = \mathbf{U} \boldsymbol{\Sigma}^2 \mathbf{U}^\top}.$$

Since  $\boxed{\mathbf{A} \mathbf{v}_i = \sigma_i \mathbf{u}_i}$ , we have

$$[\mathbf{A} \mathbf{v}_1, \mathbf{A} \mathbf{v}_2, \dots, \mathbf{A} \mathbf{v}_r] = [\sigma_1 \mathbf{u}_1, \sigma_2 \mathbf{u}_2, \dots, \sigma_r \mathbf{u}_r] \quad \xrightarrow{\text{leads to}} \quad \mathbf{A} \mathbf{V} = \mathbf{U} \boldsymbol{\Sigma},$$

which completes the proof. ■

By appending silent columns in  $\mathbf{U}$  and  $\mathbf{V}$ , we can easily find the full SVD. A byproduct of the above proof is that the spectral decomposition of  $\mathbf{A}^\top \mathbf{A} = \mathbf{V} \boldsymbol{\Sigma}^2 \mathbf{V}^\top$  will result in the spectral decomposition of  $\mathbf{A} \mathbf{A}^\top = \mathbf{U} \boldsymbol{\Sigma}^2 \mathbf{U}^\top$  with the same eigenvalues.

#### Corollary 14.4: (Eigenvalues of $\mathbf{A}^\top \mathbf{A}$ and $\mathbf{A} \mathbf{A}^\top$ )

The nonzero eigenvalues of  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A} \mathbf{A}^\top$  are the same.

We have shown in Lemma 14.1 that the eigenvalues of  $\mathbf{A}^\top \mathbf{A}$  are nonnegative, such that the eigenvalues of  $\mathbf{A} \mathbf{A}^\top$  are nonnegative as well.

#### Corollary 14.5: (Nonnegative Eigenvalues of $\mathbf{A}^\top \mathbf{A}$ and $\mathbf{A} \mathbf{A}^\top$ )

The eigenvalues of  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A} \mathbf{A}^\top$  are nonnegative.

The existence of the SVD is important for defining the effective rank of a matrix.

### Definition 14.6: Effective Rank vs Exact Rank

*Effective rank*, or also known as the *numerical rank*. Following from Lemma 14.3, the number of nonzero singular values is equal to the rank of a matrix. Assume the  $i$ -th largest singular value of  $\mathbf{A}$  is denoted as  $\sigma_i(\mathbf{A})$ . Then if  $\sigma_r(\mathbf{A}) \gg \sigma_{r+1}(\mathbf{A}) \approx 0$ ,  $r$  is known as the numerical rank of  $\mathbf{A}$ . Whereas, when  $\sigma_i(\mathbf{A}) > \sigma_{r+1}(\mathbf{A}) = 0$ , it is known as having *exact rank*  $r$  as we have used in most of our discussions.

## 14.3. Properties of the SVD

### 14.3.1 Four Subspaces in SVD

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , we have the following property:

- $\mathcal{N}(\mathbf{A})$  is the orthogonal complement of the row space  $\mathcal{C}(\mathbf{A}^\top)$  in  $\mathbb{R}^n$ :  $\dim(\mathcal{N}(\mathbf{A})) + \dim(\mathcal{C}(\mathbf{A}^\top)) = n$ ;
- $\mathcal{N}(\mathbf{A}^\top)$  is the orthogonal complement of the column space  $\mathcal{C}(\mathbf{A})$  in  $\mathbb{R}^m$ :  $\dim(\mathcal{N}(\mathbf{A}^\top)) + \dim(\mathcal{C}(\mathbf{A})) = m$ ;

This is called the fundamental theorem of linear algebra and is also known as the rank-nullity theorem. And the proof can be found in Appendix B. Furthermore, we find the basis for the four subspaces via the CR decomposition in Appendix B.1. In specific, from the SVD, we can find an orthonormal basis for each subspace.

### Lemma 14.1: (Four Orthonormal Basis)

Given the full SVD of matrix  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$  and  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$  are the column partitions of  $\mathbf{U}$  and  $\mathbf{V}$ . Then, we have the following property:

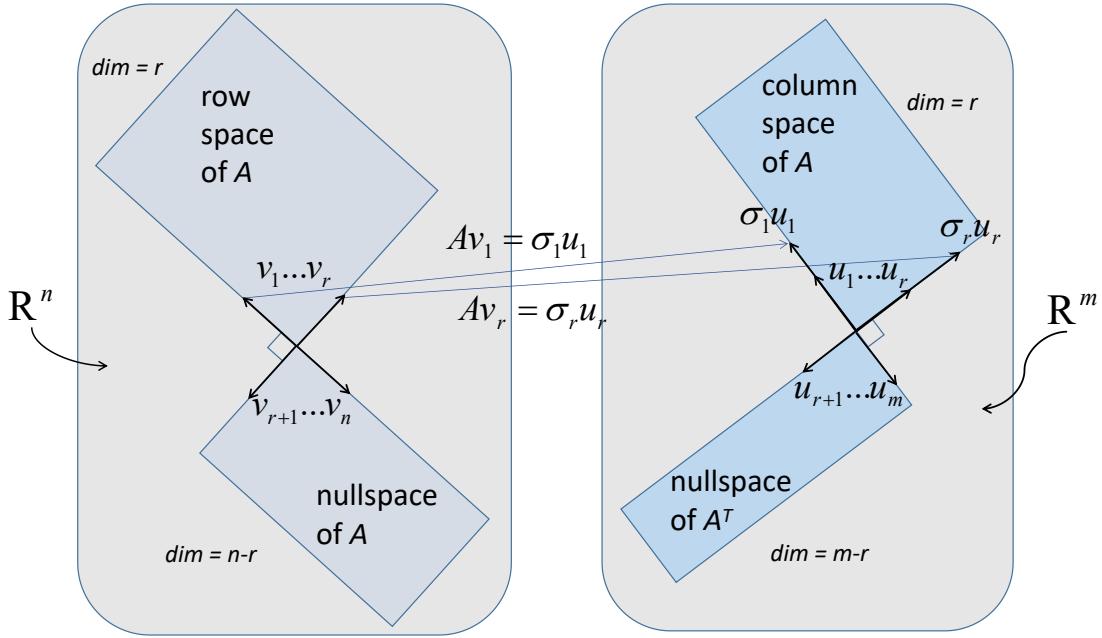
- $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A}^\top)$ ;
- $\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}$  is an orthonormal basis of  $\mathcal{N}(\mathbf{A})$ ;
- $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A})$ ;
- $\{\mathbf{u}_{r+1}, \mathbf{u}_{r+2}, \dots, \mathbf{u}_m\}$  is an orthonormal basis of  $\mathcal{N}(\mathbf{A}^\top)$ .

The relationship of the four subspaces is demonstrated in Figure 14.2 where  $\mathbf{A}$  transfer the row basis  $\mathbf{v}_i$  into column basis  $\mathbf{u}_i$  by  $\sigma_i \mathbf{u}_i = \mathbf{A} \mathbf{v}_i$  for all  $i \in \{1, 2, \dots, r\}$ .

**Proof** [of Lemma 14.1] From Lemma 13.7, for symmetric matrix  $\mathbf{A}^\top \mathbf{A}$ ,  $\mathcal{C}(\mathbf{A}^\top \mathbf{A})$  is spanned by the eigenvectors, thus  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A}^\top \mathbf{A})$ .

Since,

1.  $\mathbf{A}^\top \mathbf{A}$  is symmetric, then the row space of  $\mathbf{A}^\top \mathbf{A}$  equals the column space of  $\mathbf{A}^\top \mathbf{A}$ .
2. All rows of  $\mathbf{A}^\top \mathbf{A}$  are the combinations of the rows of  $\mathbf{A}$ , so the row space of  $\mathbf{A}^\top \mathbf{A} \subseteq$  the row space of  $\mathbf{A}$ , i.e.,  $\mathcal{C}(\mathbf{A}^\top \mathbf{A}) \subseteq \mathcal{C}(\mathbf{A}^\top)$ .
3. Since  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$  by Lemma 14.2, we then have



**Figure 14.2:** Orthonormal bases that diagonalize  $\mathbf{A}$  from SVD.

The row space of  $\mathbf{A}^\top \mathbf{A} =$  the column space of  $\mathbf{A}^\top \mathbf{A} =$  the row space of  $\mathbf{A}$ , i.e.,  $\mathcal{C}(\mathbf{A}^\top \mathbf{A}) = \mathcal{C}(\mathbf{A}^\top)$ . Thus  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A}^\top)$ .

Further, the space spanned by  $\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}$  is an orthogonal complement to the space spanned by  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$ , so  $\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}$  is an orthonormal basis of  $\mathcal{N}(\mathbf{A})$ .

If we apply this process to  $\mathbf{A}\mathbf{A}^\top$ , we will prove the rest claims in the lemma. Also, we can see that  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}$  is a basis for the column space of  $\mathbf{A}$  by Lemma 27.1<sup>1</sup>, since  $\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}$ ,  $\forall i \in \{1, 2, \dots, r\}$ . ■

### 14.3.2 SVD-Related Orthogonal Projections

Following from the four subspaces in SVD, there are several important orthogonal projections associated with the SVD. A detailed discussion on the orthogonal projection is provided in Appendix D (p. 432). In words, the orthogonal projection matrices are matrices that are symmetric and idempotent. The projection matrix will project any vector onto the column space of it. The idempotency has a geometrical meaning such that projecting twice is equivalent to projecting once. The symmetry also has a geometrical meaning such that the distance between the original vector and the projected vector (which is in the column space of the projection matrix) is minimal. Suppose  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$  is the SVD of  $\mathbf{A}$

1. For any matrix  $\mathbf{A}$ , let  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r\}$  be a set of vectors in  $\mathbb{R}^n$  which forms a basis for the row space, then  $\{\mathbf{A}\mathbf{r}_1, \mathbf{A}\mathbf{r}_2, \dots, \mathbf{A}\mathbf{r}_r\}$  is a basis for the column space of  $\mathbf{A}$ .

with rank  $r$ . If we have the following column partitions

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_r & \mathbf{U}_m \\ m \times r & m \times (m-r) \end{bmatrix}, \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_r & \mathbf{V}_n \\ n \times r & n \times (n-r) \end{bmatrix},$$

where  $\mathbf{U}_r$  and  $\mathbf{V}_r$  are the first  $r$  columns of  $\mathbf{U}$  and  $\mathbf{V}$ . Then, the four orthogonal projections can be obtained by

$$\begin{aligned} \mathbf{V}_r \mathbf{V}_r^\top &= \text{projection onto } \mathcal{C}(\mathbf{A}^\top), \\ \mathbf{V}_n \mathbf{V}_n^\top &= \text{projection onto } \mathcal{N}(\mathbf{A}), \\ \mathbf{U}_r \mathbf{U}_r^\top &= \text{projection onto } \mathcal{C}(\mathbf{A}), \\ \mathbf{U}_m \mathbf{U}_m^\top &= \text{projection onto } \mathcal{N}(\mathbf{A}^\top). \end{aligned}$$

#### 14.3.3 Relationship between Singular Values and Determinant

Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a square matrix and the singular value decomposition of  $\mathbf{A}$  is given by  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , it follows that

$$|\det(\mathbf{A})| = |\det(\mathbf{U}\Sigma\mathbf{V}^\top)| = |\det(\Sigma)| = \sigma_1 \sigma_2 \dots \sigma_n.$$

If all the singular values  $\sigma_i$  are nonzero, then  $\det(\mathbf{A}) \neq 0$ . That is,  $\mathbf{A}$  is **nonsingular**. If there is at least one singular value such that  $\sigma_i = 0$ , then  $\det(\mathbf{A}) = 0$ , and  $\mathbf{A}$  does not have full rank, and is not invertible. Then the matrix is called **singular**. This is why  $\sigma_i$ 's are known as the singular values.

#### 14.3.4 Orthogonal Equivalence

We have defined in Definition 8.1 (p. 198) that  $\mathbf{A}$  and  $\mathbf{PAP}^{-1}$  are similar matrices for any nonsingular matrix  $\mathbf{P}$  and square matrix  $\mathbf{A}$ . The orthogonal equivalence is defined in a similar way for rectangular matrices.

##### Definition 14.2: Orthogonal Equivalent Matrices

For any orthogonal matrices  $\mathbf{U}$  and  $\mathbf{V}$ , the matrices  $\mathbf{A}$  and  $\mathbf{UAV}$  are called *orthogonal equivalent matrices*. Or *unitary equivalent* in complex domain when  $\mathbf{U}$  and  $\mathbf{V}$  are unitary.

Then, we have the following property for orthogonal equivalent matrices.

##### Lemma 14.3: (Orthogonal Equivalent Matrices)

For any orthogonal equivalent matrices  $\mathbf{A}$  and  $\mathbf{B}$ , then singular values are the same.

**Proof** [of Lemma 14.3] Since  $\mathbf{A}$  and  $\mathbf{B}$  are orthogonal equivalent, there exist orthogonal matrices that  $\mathbf{B} = \mathbf{UAV}$ . We then have

$$\mathbf{BB}^\top = (\mathbf{UAV})(\mathbf{V}^\top \mathbf{A}^\top \mathbf{U}^\top) = \mathbf{UAA}^\top \mathbf{U}^\top.$$

This implies  $\mathbf{B}\mathbf{B}^\top$  and  $\mathbf{A}\mathbf{A}^\top$  are similar matrices. By Lemma 8.2 (p. 198), the eigenvalues of similar matrices are the same, which proves the singular values of  $\mathbf{A}$  and  $\mathbf{B}$  are the same. ■

### 14.3.5 SVD for QR

#### Lemma 14.4: (Orthogonal Equivalent Matrices)

Suppose the full QR decomposition for matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $m \geq n$  is given by  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  where  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  is orthogonal and  $\mathbf{R} \in \mathbb{R}^{m \times n}$  is upper triangular. Then  $\mathbf{A}$  and  $\mathbf{R}$  have the same singular values and right singular vectors.

**Proof** [of Lemma 14.4] We notice that  $\mathbf{A}^\top \mathbf{A} = \mathbf{R}^\top \mathbf{R}$  such that  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{R}^\top \mathbf{R}$  have the same eigenvalues and eigenvectors, i.e.,  $\mathbf{A}$  and  $\mathbf{R}$  have the same singular values and right singular vectors (i.e., the eigenvectors of  $\mathbf{A}^\top \mathbf{A}$  or  $\mathbf{R}^\top \mathbf{R}$ ). ■

The above lemma is important show the existence and properties of the rank-revealing QR decomposition (Section 3.11, p. 106). And it implies that an SVD of a matrix can be constructed by the QR decomposition of itself. Suppose the QR decomposition of  $\mathbf{A}$  is given by  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  and the SVD of  $\mathbf{R}$  is given by  $\mathbf{R} = \mathbf{U}_0 \Sigma \mathbf{V}^\top$ . Therefore, the SVD of  $\mathbf{A}$  can be obtained by

$$\mathbf{A} = \underbrace{\mathbf{Q}\mathbf{U}_0}_{\mathbf{U}} \Sigma \mathbf{V}^\top.$$

### 14.3.6 Interlacing Property

The interlacing property of SVD is from that of the symmetric matrix. We provide the theorem directly and the proof can be found in ([Wilkinson, 1971](#)) and further discussed in ([Golub and Van Loan, 2013](#)) (Theorem 8.1.7, p. 443).

#### Theorem 14.5: (Interlacing Property for Symmetric Matrix)

Suppose  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is symmetric, and  $\mathbf{A}_r$  is defined as the upper left  $r \times r$  submatrix of  $\mathbf{A}$  such that  $\mathbf{A}_r = \mathbf{A}[1:r, 1:r]$ . Define  $\lambda_i(\mathbf{B})$  the  $i$ -th largest eigenvalue of matrix  $\mathbf{B}$ . Let  $k = r + 1$  and  $\mathbf{B}_k = \mathbf{A}_k$ , then we have

$$\lambda_{r+1}(\mathbf{B}_k) \leq \lambda_r(\mathbf{A}_r) \leq \lambda_r(\mathbf{B}_k) \leq \dots \leq \lambda_2(\mathbf{B}_k) \leq \lambda_1(\mathbf{A}_r) \leq \lambda_1(\mathbf{B}_k).$$

The interlacing property of for singular value can be derived directly from that of the symmetric matrices:

**Theorem 14.6: (Interlacing Property for Singular Values)**

Suppose  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$  with  $m \geq n$ , and  $\mathbf{A}_r = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r]$ . Define  $\sigma_i(\mathbf{B})$  as the  $i$ -th largest singular value of matrix  $\mathbf{B}$ .

$\mathbf{A}$  is symmetric, and  $\mathbf{A}_r$  is defined as the upper left  $r \times r$  submatrix of  $\mathbf{A}$  such that  $\mathbf{A}_r = \mathbf{A}[1:r, 1:r]$ . Define  $\lambda_i(\mathbf{B})$  the  $i$ -th largest eigenvalue of matrix  $\mathbf{B}$ . Let  $k = r+1$  and  $\mathbf{B}_k = \mathbf{A}_k$ , then we have

$$\sigma_k(\mathbf{B}_k) \leq \sigma_r(\mathbf{A}_r) \leq \sigma_r(\mathbf{B}_k) \leq \dots \leq \sigma_2(\mathbf{B}_k) \leq \sigma_1(\mathbf{A}_r) \leq \sigma_1(\mathbf{B}_k).$$

**14.4. Computing the SVD**

Suppose again we have an oracle algorithm to compute the eigenvalues and eigenvectors of  $\mathbf{A}^\top \mathbf{A}$  which costs  $f(m, n)$  flops. Then the computation of SVD is trivial from the steps shown above. The algorithm is shown in Algorithm 35.

**Algorithm 35** A Simple SVD

**Require:** Rank- $r$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  with size  $m \times n$ ;

- 1: Initially get  $\mathbf{A}^\top \mathbf{A} \mathbf{x}_i = \sigma_i^2 \mathbf{x}_i \quad \forall i \in \{1, 2, \dots, r\}; \quad \triangleright f(m, n) \text{ flops}$
- 2: Normalize each eigenvectors  $\mathbf{v}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|}; \quad \triangleright r \times 3n = 3nr \text{ flops}$
- 3: Normalize each eigenvectors  $\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}; \quad \triangleright r(m(2n-1) + m) \text{ flops}$

By completing  $\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}$  into a full orthogonal basis via Gram-Schmidt, we have

$$\begin{aligned} \mathbf{v}_i^\top \mathbf{v}_i &= 1, \quad \forall r+1 \leq i \leq n, \\ \mathbf{v}_i^\top \mathbf{v}_j &= 0, \quad \forall 1 \leq i \leq n, r+1 \leq j \leq n. \end{aligned}$$

And similarly  $\{\mathbf{u}_{r+1}, \mathbf{u}_{r+2}, \dots, \mathbf{u}_m\}$  into a full orthogonal basis via Gram-Schmidt, we have

$$\begin{aligned} \mathbf{u}_i^\top \mathbf{u}_i &= 1, \quad \forall r+1 \leq i \leq m, \\ \mathbf{u}_i^\top \mathbf{u}_j &= 0, \quad \forall 1 \leq i \leq m, r+1 \leq j \leq m. \end{aligned}$$

The detailed analysis of computational complexity for SVD is complicated. It is typically computed numerically by a two-step procedure. In the first step, the matrix is reduced to a bidiagonal matrix. This takes  $O(mn^2)$  flops (Section 10, p. 214) and the second step is to compute the SVD of the bidiagonal matrix which takes  $O(n)$  iterations with each involving  $O(n)$  flops. Thus, the overall cost is  $O(mn^2)$  flops. For those who are interested in the computation of SVD, please refer to Section 15 or ([Trefethen and Bau III, 1997](#); [Golub and Van Loan, 2013](#); [Kishore Kumar and Schneider, 2017](#)) for more details.

**14.4.1 Randomized Method for Computing the SVD Approximately**

Suppose now the matrix  $\mathbf{A}$  admits rank decomposition (Theorem 5.1, p. 164):

$$\mathbf{A}_{m \times n} = \mathbf{D}_{m \times r} \mathbf{F}_{r \times n},$$

where the columns of  $\mathbf{D}$  span the same column space of  $\mathbf{A}$ :  $\mathcal{C}(\mathbf{D}) = \mathcal{C}(\mathbf{F})$ . If we orthogonalize the columns of  $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_r]$  into  $\mathbf{Q}_r = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_r]$  such that

$$\text{span}([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k]) = \text{span}([\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_k]), \quad \text{for } k \in \{1, 2, \dots, r\}.$$

This can be done by the *reduced* QR decomposition via the Gram-Schmidt process where the complexity is  $O(mr^2)$  (Section 3.7, p. 90) if  $\mathbf{D} \in \mathbb{R}^{m \times r}$  (where we complete the  $\mathbf{Q}_r$  into an orthogonal one  $\mathbf{Q} = [\mathbf{Q}_r, \mathbf{Q}_2]$  such that  $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$  for further analysis, but note that  $\mathbf{Q}_2$  is not needed for the final algorithm!). One can show that if the SVD of the  $m \times n$  matrix  $\mathbf{E} = \mathbf{Q}^\top \mathbf{A} \in \mathbb{R}^{m \times n}$  is given by  $\mathbf{E} = \tilde{\mathbf{U}} \Sigma \mathbf{V}^\top$ , the SVD of  $\mathbf{A}$  can be obtained by

$$\mathbf{A} = \mathbf{Q} \tilde{\mathbf{E}} = \underbrace{(\mathbf{Q} \tilde{\mathbf{U}})}_{\mathbf{U}} \Sigma \mathbf{V}^\top. \quad (14.1)$$

Now let's decompose the above finding into

$$\begin{aligned} \tilde{\mathbf{E}} &= \begin{bmatrix} \mathbf{Q}_r^\top \mathbf{A} \\ \mathbf{Q}_2^\top \mathbf{A} \end{bmatrix} = \tilde{\mathbf{U}} \Sigma \mathbf{V}^\top = \begin{bmatrix} \tilde{\mathbf{U}}_r & \tilde{\mathbf{U}}_{12} \\ \tilde{\mathbf{U}}_{21} & \tilde{\mathbf{U}}_{22} \end{bmatrix} \begin{bmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_r^\top \\ \mathbf{V}_2^\top \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{U}}_r \Sigma_r \\ \tilde{\mathbf{U}}_{21} \Sigma_r \end{bmatrix} \mathbf{V}_r^\top \\ &\xrightarrow{\text{leads to}} \begin{cases} \mathbf{Q}_r^\top \mathbf{A} = \tilde{\mathbf{U}}_r \Sigma_r \mathbf{V}_r^\top; \\ \mathbf{Q}_2^\top \mathbf{A} = \tilde{\mathbf{U}}_{21} \Sigma_r \mathbf{V}_r^\top. \end{cases} \end{aligned}$$

By Equation (14.1), we have

$$\tilde{\mathbf{U}} = \mathbf{Q}^\top \mathbf{U} = \begin{bmatrix} \mathbf{Q}_r^\top \\ \mathbf{Q}_2^\top \end{bmatrix} \begin{bmatrix} \mathbf{U}_r & \mathbf{U}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_r^\top \mathbf{U}_r & \mathbf{Q}_r^\top \mathbf{U}_2 \\ \mathbf{Q}_2^\top \mathbf{U}_r & \mathbf{Q}_2^\top \mathbf{U}_2 \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{U}}_r & \tilde{\mathbf{U}}_{12} \\ \tilde{\mathbf{U}}_{21} & \tilde{\mathbf{U}}_{22} \end{bmatrix}.$$

Since  $\mathcal{C}(\mathbf{Q}_r) = \mathcal{C}(\mathbf{A}) = \mathcal{C}(\mathbf{U}_r)$ , it follows that  $\mathbf{Q}_2^\top \mathbf{U}_r = \mathbf{0}$  since  $\mathbf{Q}_2$  lies in the orthogonal complement of  $\mathbf{Q}_r$  (which is also the orthogonal complement of  $\mathbf{U}_r$ ). Therefore,

$$\tilde{\mathbf{E}} = \begin{bmatrix} \mathbf{Q}_r^\top \mathbf{A} \\ \mathbf{0} \end{bmatrix}$$

By Equation (14.1) again,

$$\mathbf{A} = \mathbf{Q} \tilde{\mathbf{E}} = [\mathbf{Q}_r \quad \mathbf{Q}_2] \begin{bmatrix} \mathbf{Q}_r^\top \mathbf{A} \\ \mathbf{0} \end{bmatrix} = \mathbf{Q}_r \mathbf{Q}_r^\top \mathbf{A}.$$

We carefully notice that  $\mathbf{Q}_r^\top \mathbf{A} = \tilde{\mathbf{U}}_r \Sigma_r \mathbf{V}_r^\top$  above is the SVD of  $\mathbf{Q}_r^\top \mathbf{A} \in \mathbb{R}^{r \times n}$ .

To conclude the observation, suppose we find a matrix  $\mathbf{D} \in \mathbb{R}^{m \times r}$  that spans the column space of  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . The *reduced* QR decomposition of  $\mathbf{D} = \mathbf{Q}_r \mathbf{R}$  such that  $\mathbf{Q}_r$  also span the column space of  $\mathbf{A}$ . Calculate the SVD of the small matrix  $\mathbf{E} = \mathbf{Q}_r^\top \mathbf{A} \in \mathbb{R}^{r \times n}$  which costs  $O(nr^2)$  flops:  $\mathbf{E} = \mathbf{Q}_r^\top \mathbf{A} = \tilde{\mathbf{U}}_r \Sigma_r \mathbf{V}_r^\top$ . The *reduced* SVD of the large matrix can be obtained by

$$\mathbf{A} = \mathbf{Q}_r (\mathbf{Q}_r^\top \mathbf{A}) = \underbrace{\mathbf{Q}_r \tilde{\mathbf{U}}_r}_{\mathbf{U}_r} \Sigma_r \mathbf{V}_r^\top.$$

By completing the matrices  $\mathbf{U}_r \in \mathbb{R}^{m \times r}$ ,  $\mathbf{V}_r \in \mathbb{R}^{n \times r}$  into full orthogonal matrices, we find the *full* SVD of  $\mathbf{A}$ .

Further, similar to the randomized algorithm for interpolative decomposition in Section 7.6 (p. 188), by Lemma 27.1 (p. 426), for any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , suppose that  $\{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_r\}$  is a set of vectors in  $\mathbb{R}^n$  which forms a basis for the row space, then  $\{\mathbf{A}\mathbf{g}_1, \mathbf{A}\mathbf{g}_2, \dots, \mathbf{A}\mathbf{g}_r\}$  is a basis for the column space of  $\mathbf{A}$ :

$$\mathcal{C}(\mathbf{AG}) = \mathcal{C}(\mathbf{A}). \quad (14.2)$$

And a small integer  $k$  (say  $k = 10$ ) should be picked to over-sample such that there is a high probability that  $\mathbf{G} = [\mathbf{g}_1, \dots, \mathbf{g}_r, \mathbf{g}_{r+1}, \dots, \mathbf{g}_{r+k}]$  contains the row basis of  $\mathbf{A}$ . Other methods to make sure  $\mathbf{G}$  can span the row basis of  $\mathbf{A}$  or  $\mathbf{AG}$  can span the column space of  $\mathbf{A}$  are discussed in Remark 7.2 (p. 190). Again, The choice  $k = 10$  is often good. The procedure is formulated in Algorithm 36 where the end result costs  $O(mn(r + k))$  flops compared to the original  $O(mn^2)$  flops. We notice that the leading term of  $O(mn(r + k))$  flops in the algorithms comes from step 6 to do the matrix multiplication  $\mathbf{Q}_r^\top \mathbf{A}$ . A structured choice on the random matrix  $\mathbf{G}$  can reduce to  $O(mn \log(r + k))$  flops (Martinsson, 2019; Ailon and Chazelle, 2006). We shall not give the details.

---

**Algorithm 36** A Randomized Method to Compute the SVD

---

**Require:** Rank- $r$  matrix  $\mathbf{A}$  with size  $m \times n$ ;

- 1: Decide the over-sampling parameter  $k$  (e.g.,  $k = 10$ ), and let  $z = r + k$ ;
  - 2: Decide the iteration number:  $\eta$  (e.g.,  $\eta = 0, 1$  or  $2$ );
  - 3: Generate  $r + k$  Gaussian random vectors in  $\mathbb{R}^n$  into columns of matrix  $\mathbf{G} \in \mathbb{R}^{n \times (r+k)}$ ;▷ i.e., probably contain the row basis of  $\mathbf{A}$
  - 4: Initialize  $\mathbf{D} = \mathbf{AG} \in \mathbb{R}^{m \times (r+k)}$ ; ▷ probably  $\mathcal{C}(\mathbf{D}) = \mathcal{C}(\mathbf{A})$ ,  $m(2n - 1)(r + k)$  flops
  - 5: Compute full QR decomposition  $\mathbf{D} = \underbrace{\mathbf{Q}_r}_{m \times (r+k)} \mathbf{R}$ ; ▷  $O(mz^2)$  flops
  - 6: Form matrix  $\mathbf{E} = \mathbf{Q}_r^\top \mathbf{A} \in \mathbb{R}^{(r+k) \times n}$ ; ▷  $nz(2m - 1)$  flops
  - 7: Compute the SVD of the small matrix  $\mathbf{E}$ :  $\mathbf{E} = \mathbf{U}_0 \Sigma \mathbf{V}^\top$ ; ▷  $O(nz^2)$  flops
  - 8: Form  $\mathbf{U} = \mathbf{Q}_r \mathbf{U}_0$  such that the *reduced* SVD of  $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^\top$ ; ▷  $mz(2z - 1)$  flops
- 

## 14.5. Polar Decomposition

### Theorem 14.1: (Polar Decomposition)

For every real  $n \times n$  square matrix  $\mathbf{A}$  with rank  $r$ , then matrix  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{Q}_l \mathbf{S},$$

where  $\mathbf{Q}_l$  is an orthogonal matrix, and  $\mathbf{S}$  is a positive semidefinite matrix. And this form is called the **left polar decomposition**. Also matrix  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{S} \mathbf{Q}_r,$$

where  $\mathbf{Q}_r$  is an orthogonal matrix, and  $\mathbf{S}$  is a positive semidefinite matrix. And this form is called the **right polar decomposition**.

Specially, the left and right polar decomposition of a square matrix  $\mathbf{A}$  is **unique**.

Since every  $n \times n$  square matrix  $\mathbf{A}$  has full SVD  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where both  $\mathbf{U}$  and  $\mathbf{V}$  are  $n \times n$  orthogonal matrix. We then have  $\mathbf{A} = (\mathbf{U}\mathbf{V}^\top)(\mathbf{V}\Sigma\mathbf{V}^\top) = \mathbf{Q}_l\mathbf{S}$  where it can be easily verified that  $\mathbf{Q}_l = \mathbf{U}\mathbf{V}^\top$  is an orthogonal matrix and  $\mathbf{S} = \mathbf{V}\Sigma\mathbf{V}^\top$  is a symmetric matrix. We notice that the singular values in  $\Sigma$  are nonnegative, such that  $\mathbf{S} = \mathbf{V}\Sigma\mathbf{V}^\top = \mathbf{V}\Sigma^{1/2}\Sigma^{1/2\top}\mathbf{V}^\top$  showing that  $\mathbf{S}$  is PSD.

Similarly, we have  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{U}^\top\mathbf{U}\mathbf{V}^\top = (\mathbf{U}\Sigma\mathbf{U}^\top)(\mathbf{U}\mathbf{V}^\top) = \mathbf{S}\mathbf{Q}_r$ . And  $\mathbf{S} = \mathbf{U}\Sigma\mathbf{U}^\top = \mathbf{U}\Sigma^{1/2}\Sigma^{1/2\top}\mathbf{U}^\top$  such that  $\mathbf{S}$  is PSD as well.

For the uniqueness of the right polar decomposition, we suppose the decomposition is not unique, and two of the decompositions are given by

$$\mathbf{A} = \mathbf{S}_1\mathbf{Q}_1 = \mathbf{S}_2\mathbf{Q}_2,$$

such that

$$\mathbf{S}_1 = \mathbf{S}_2\mathbf{Q}_2\mathbf{Q}_1^\top.$$

Since  $\mathbf{S}_1$  and  $\mathbf{S}_2$  are symmetric, we have

$$\mathbf{S}_1^2 = \mathbf{S}_1\mathbf{S}_1^\top = \mathbf{S}_2\mathbf{Q}_2\mathbf{Q}_1^\top\mathbf{Q}_1\mathbf{Q}_2^\top\mathbf{S}_2 = \mathbf{S}_2^2.$$

This implies  $\mathbf{S}_1 = \mathbf{S}_2$ , and the decomposition is unique (Theorem 13.7, p. 262). Similarly, the uniqueness of the left polar decomposition can be implied from the context.

### Corollary 14.2: (Full Rank Polar Decomposition)

When  $\mathbf{A} \in \mathbb{R}^{n \times n}$  has full rank, then the  $\mathbf{S}$  in both the left and right polar decomposition above is a symmetric positive definite matrix.

## 14.6. Generalized Singular Value Decomposition (GSVD)\*

Following from (Golub and Van Loan, 2013), we here give a short review for generalized singular value decomposition (GSVD). See also (Zhang, 2017; Bai and Demmel, 1993; Zha, 1989; Golub and Van Loan, 2013; Paige and Saunders, 1981) for a more detailed discussion on GSVD.

### 14.6.1 CS Decomposition

#### Theorem 14.1: (CS Decomposition)

Suppose

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} \begin{matrix} m_1 \\ m_2 \end{matrix}, \quad \begin{matrix} n_1 \\ n_2 \end{matrix},$$

is an orthogonal matrix with  $m_1 \geq n_1$  and  $m_1 \geq m_2$ . Define the nonnegative integer  $p$  and  $q$  by  $p = \max\{0, n_1 - m_2\}$ , and  $q = \max\{0, m_2 - n_1\}$ . There exist orthogonal matrices  $\mathbf{U}_1 \in \mathbb{R}^{m_1 \times m_1}$ ,  $\mathbf{U}_2 \in \mathbb{R}^{m_2 \times m_2}$ ,  $\mathbf{V}_1 \in \mathbb{R}^{n_1 \times n_1}$ , and  $\mathbf{V}_2 \in \mathbb{R}^{n_2 \times n_2}$  such that if

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix}, \quad \text{and} \quad \mathbf{V} = \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 \end{bmatrix},$$

then

$$\mathbf{U}^\top \mathbf{Q} \mathbf{V} = \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 & 0 \\ 0 & \mathbf{C} & \mathbf{S} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{I} \\ 0 & \mathbf{S} & -\mathbf{C} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{I} & 0 \end{bmatrix} \begin{matrix} p \\ n_1 - p \\ m_1 - n_1 \\ n_1 - p \\ q \end{matrix},$$

$$\begin{matrix} p & n_1 - p & n_1 - p & q & m_1 - n_1 \end{matrix}$$

where

$$\mathbf{C} = \text{diag}(\cos(\theta_{p+1}), \dots, \cos(\theta_{n_1})) = \text{diag}(c_{p+1}, \dots, c_{n_1}),$$

$$\mathbf{S} = \text{diag}(\sin(\theta_{p+1}), \dots, \sin(\theta_{n_1})) = \text{diag}(s_{p+1}, \dots, s_{n_1}),$$

and  $0 \leq \theta_{p+1} \leq \dots \leq \theta_{n_1} \leq \pi/2$ .

The proof can be found in (Paige and Saunders, 1981) and a thin version of CS decomposition is provided in (Golub and Van Loan, 2013).

#### 14.6.2 Generalized Singular Value Decomposition (GSVD)

##### Theorem 14.2: (Generalized Singular Value Decomposition)

Assume that  $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}$  and  $\mathbf{B} \in \mathbb{R}^{m_2 \times n_1}$  with  $m_1 \geq n_1$  and

$$r = \text{rank}(\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}).$$

Then there exist orthogonal matrices  $\mathbf{U}_1 \in \mathbb{R}^{m_1 \times m_1}$  and  $\mathbf{U}_2 \in \mathbb{R}^{m_2 \times m_2}$ , and invertible matrix  $\mathbf{X} \in \mathbb{R}^{n_1 \times n_1}$  such that

$$\mathbf{U}_1^\top \mathbf{A} \mathbf{X} = \mathbf{D}_A = \begin{bmatrix} \mathbf{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_A & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{matrix} p \\ r-p \\ m_1-r \end{matrix}$$

$$\begin{matrix} p & r-p & n_1-r \end{matrix}$$

$$\mathbf{U}_2^\top \mathbf{B} \mathbf{X} = \mathbf{D}_B = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_B & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{matrix} p \\ r-p \\ m_2-r \end{matrix},$$

$$\begin{matrix} p & r-p & n_1-r \end{matrix}$$

where  $p = \max\{r - m_2, 0\}$ ,  $\mathbf{S}_A = \text{diag}(\alpha_{p+1}, \dots, \alpha_r)$ , and  $\mathbf{S}_B = \text{diag}(\beta_{p+1}, \dots, \beta_r)$ , and

$$\alpha_i^2 + \beta_i^2 = 1, \quad \forall i \in \{p+1, \dots, r\}.$$

**Proof** [of Theorem 14.2] Suppose the SVD of  $\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}$  is given by

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{11} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} \begin{bmatrix} \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P}^\top,$$

where  $\Sigma_r \in \mathbb{R}^{r \times r}$  is nonsingular,  $\mathbf{Q}_{11} \in \mathbb{R}^{m_1 \times r}$ , and  $\mathbf{Q}_{21} \in \mathbb{R}^{m_2 \times r}$ . Apply CS decomposition on  $\mathbf{Q}$ , there exist orthogonal matrices  $\mathbf{U}_1 \in \mathbb{R}^{m_1 \times m_1}$ ,  $\mathbf{U}_2 \in \mathbb{R}^{m_2 \times m_2}$ , and  $\mathbf{V}_1 \in \mathbb{R}^{r \times r}$  such that

$$\begin{bmatrix} \mathbf{U}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix}^\top \begin{bmatrix} \mathbf{Q}_{11} \\ \mathbf{Q}_{21} \end{bmatrix} \mathbf{V}_1 = \begin{bmatrix} \mathbf{D}_A^r \\ \mathbf{D}_B^r \end{bmatrix},$$

where  $\mathbf{D}_A^r$  and  $\mathbf{D}_B^r$  are first  $r$  columns of  $\mathbf{D}_A$  and  $\mathbf{D}_B$  (note that row permutation needed here). It then follows that

$$\begin{aligned} \begin{bmatrix} \mathbf{U}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix}^\top \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix} \mathbf{P} &= \begin{bmatrix} \mathbf{D}_A^r & \mathbf{U}_1 \mathbf{Q}_{12} \\ \mathbf{D}_B^r & \mathbf{U}_2 \mathbf{Q}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^\top \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{D}_A^r & \mathbf{0} \\ \mathbf{D}_B^r & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^\top \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_1-r} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{D}_A \\ \mathbf{D}_B \end{bmatrix} \begin{bmatrix} \mathbf{V}_1^\top \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_1-r} \end{bmatrix}. \end{aligned}$$

By setting

$$\mathbf{X} = \mathbf{P} \begin{bmatrix} \mathbf{V}_1^\top \Sigma_r & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{n_1-r} \end{bmatrix}^{-1},$$

we complete the proof. ■

Note that if  $\mathbf{B} = \mathbf{I}_{n_1}$ , and we set  $\mathbf{X} = \mathbf{U}_2$ , then we obtain the SVD of  $\mathbf{A}$ .

## 14.7. Applications

### 14.7.1 Application: Least Squares via SVD for Rank Deficient Matrices

The least squares problem is described in Section 3.20.1 (p. 125). As a recap, let's consider the overdetermined system  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the data matrix,  $\mathbf{b} \in \mathbb{R}^m$  with  $m > n$  is the observation matrix. Normally  $\mathbf{A}$  will have full column rank since the data from real work has a large chance to be unrelated. And the least squares (LS) solution is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  for minimizing  $\|\mathbf{Ax} - \mathbf{b}\|^2$ , where  $\mathbf{A}^\top \mathbf{A}$  is invertible since  $\mathbf{A}$  has full column rank and  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$ .

However, if  $\mathbf{A}$  does not have full column rank,  $\mathbf{A}^\top \mathbf{A}$  is not invertible. We thus can use the SVD decomposition of  $\mathbf{A}$  to solve the LS problem. And we illustrate this in the following theorem.

**Theorem 14.1: (LS via SVD for Rank Deficient Matrix)**

Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$  is its full SVD decomposition with  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  being orthogonal matrices and  $\text{rank}(\mathbf{A}) = r$ . Suppose  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$ ,  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$  and  $\mathbf{b} \in \mathbb{R}^m$ , then the LS solution with the minimal 2-norm to  $\mathbf{Ax} = \mathbf{b}$  is given by

$$\mathbf{x}_{LS} = \sum_{i=1}^r \frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \mathbf{v}_i = \mathbf{V}\Sigma^+\mathbf{U}^\top \mathbf{b}, \quad (14.3)$$

where the upper-left side of  $\Sigma^+ \in \mathbb{R}^{n \times m}$  is a diagonal matrix,  $\Sigma^+ = \begin{bmatrix} \Sigma_1^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$  where  $\Sigma_1^+ = \text{diag}(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r})$ .

**Proof** [of Theorem 14.1] Write out the loss to be minimized

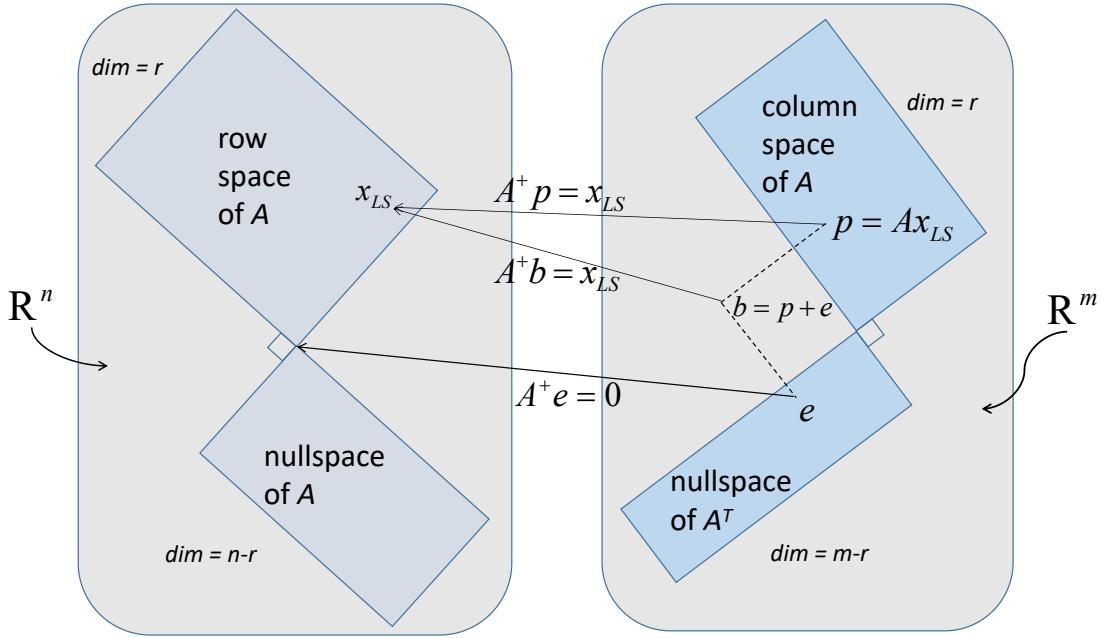
$$\begin{aligned} \|\mathbf{Ax} - \mathbf{b}\|^2 &= (\mathbf{Ax} - \mathbf{b})^\top (\mathbf{Ax} - \mathbf{b}) \\ &= (\mathbf{Ax} - \mathbf{b})^\top \mathbf{U}\mathbf{U}^\top (\mathbf{Ax} - \mathbf{b}) \quad (\text{Since } \mathbf{U} \text{ is an orthogonal matrix}) \\ &= \|\mathbf{U}^\top \mathbf{Ax} - \mathbf{U}^\top \mathbf{b}\|^2 \quad (\text{Invariant under orthogonal}) \\ &= \|\mathbf{U}^\top \mathbf{AVV}^\top \mathbf{x} - \mathbf{U}^\top \mathbf{b}\|^2 \quad (\text{Since } \mathbf{V} \text{ is an orthogonal matrix}) \\ &= \|\Sigma\alpha - \mathbf{U}^\top \mathbf{b}\|^2 \quad (\text{Let } \alpha = \mathbf{V}^\top \mathbf{x}) \\ &= \sum_{i=1}^r (\sigma_i \alpha_i - \mathbf{u}_i^\top \mathbf{b})^2 + \sum_{i=r+1}^m (\mathbf{u}_i^\top \mathbf{b})^2. \quad (\text{Since } \sigma_{r+1} = \sigma_{r+2} = \dots = \sigma_m = 0) \end{aligned}$$

Since  $\mathbf{x}$  only appears in  $\alpha$ , we just need to set  $\alpha_i = \frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i}$  for all  $i \in \{1, 2, \dots, r\}$  to minimize the above equation. For any value of  $\alpha_{r+1}, \alpha_{r+2}, \dots, \alpha_n$ , it won't change the result. From the regularization point of view (or here, we want the minimal 2-norm) we can set them to be 0. This gives us the LS solution via SVD:

$$\mathbf{x}_{LS} = \sum_{i=1}^r \frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \mathbf{v}_i = \mathbf{V}\Sigma^+\mathbf{U}^\top \mathbf{b} = \mathbf{A}^+\mathbf{b},$$

where  $\mathbf{A}^+ = \mathbf{V}\Sigma^+\mathbf{U}^\top \in \mathbb{R}^{n \times m}$  is known as the **pseudo-inverse** of  $\mathbf{A}$ . Please refer to Appendix E (p. 445) for a detailed discussion about pseudo-inverse where we also prove that the column space of  $\mathbf{A}^+$  is equal to the row space of  $\mathbf{A}$ , and the row space of  $\mathbf{A}^+$  is equal to the column space of  $\mathbf{A}$ . ■

Let  $\mathbf{A}^+ = \mathbf{V}\Sigma^+\mathbf{U}^\top$  be the pseudo-inverse of  $\mathbf{A}$ . The pseudo-inverse  $\mathbf{A}^+$  agrees with  $\mathbf{A}^{-1}$  when  $\mathbf{A}$  is invertible. The solution of least squares is to make the error  $\mathbf{b} - \mathbf{Ax}$  as small as possible concerning the mean square error. Since  $\mathbf{Ax}$  can never leave the column space of  $\mathbf{A}$ , we should choose the closest point to  $\mathbf{b}$  in the column space (Strang, 1993). This point is the projection  $\mathbf{p}$  of  $\mathbf{b}$ . Then the error vector  $\mathbf{e} = \mathbf{b} - \mathbf{p}$  has minimal length. In another word, the best combination  $\mathbf{p} = \mathbf{Ax}_{LS}$  is the projection of  $\mathbf{b}$  onto the column



**Figure 14.3:**  $A^+$ : Pseudo-inverse of  $A$ .

space. The error  $e$  is perpendicular to the column space. Therefore,  $e = b - Ax_{LS}$  is in the null space of  $A^\top$ :

$$A^\top(b - Ax_{LS}) = \mathbf{0} \quad \text{or} \quad A^\top b = A^\top Ax_{LS},$$

which is also known as the normal equation of least squares. The relationship between  $e$  and  $p$  is shown in Figure 14.3 where  $b$  is split into  $p + e$ . Since  $e$  is in  $\mathcal{N}(A^\top)$  and perpendicular to  $\mathcal{C}(A)$ , and we have shown in Section 14.3,  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}$  is an orthonormal basis of  $\mathcal{C}(A)$ , then the first  $r$  components of  $\mathbf{U}^\top e$  are all zeros. Therefore,  $A^\top e = \mathbf{V}\Sigma^+\mathbf{U}^\top e = \mathbf{0}$ . Moreover,  $x_{LS} = A^+b = A^+(p + e) = A^+p$ .

Further, we have also shown in Section 14.3,  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(A^\top)$ ,  $x_{LS} = \sum_{i=1}^r \frac{\mathbf{u}_i^\top b}{\sigma_i} \mathbf{v}_i$  thus is in the row space of  $A$ , i.e., it cannot be split into a combination of two components that are in row space of  $A$  and null space of  $A$  respectively.

Apart from this LS solution from SVD, in practice, a direct solution of the normal equations can lead to numerical difficulties when  $\mathbf{X}^\top \mathbf{X}$  is close to singular. In particular, when two or more of the columns in  $\mathbf{X}^\top \mathbf{X}$  are co-linear, the resulting parameter values can have a large magnitude. Such near degeneracies will not be uncommon when dealing with real data sets. The resulting numerical difficulties can be addressed using the SVD as well (Bishop, 2006).

#### 14.7.2 Application: Least Squares with Norm Ratio Method

We first define the Frobenius norm as follows and a detailed discussion on matrix norm can be found in Appendix L (p. 476).

**Definition 14.2: Frobenius Norm**

The Frobenius norm of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is defined as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1, j=1}^{m, n} (\mathbf{A}_{ij})^2} = \sqrt{\text{tr}(\mathbf{A}\mathbf{A}^\top)} = \sqrt{\text{tr}(\mathbf{A}^\top\mathbf{A})} = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_r^2}.$$

Following the setup in the last section. Let  $\mathbf{A}_k \in \mathbb{R}^{m \times n}$  be the rank- $k$  approximation to the original  $m \times n$  matrix  $\mathbf{A}$ . Define the *Frobenius norm ratio* (Zhang, 2017) as

$$\nu(k) = \frac{\|\mathbf{A}_k\|_F}{\|\mathbf{A}\|_F} = \frac{\sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_k^2}}{\sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_h^2}}, \quad h = \min\{m, n\},$$

where  $\mathbf{A}_k$  is the truncated SVD of  $\mathbf{A}$  with the largest  $k$  terms, i.e.,  $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  from SVD of  $\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ . We choose the minimum integer  $k$  satisfying

$$\nu(k) \geq \alpha$$

as the *effective rank estimate*  $\hat{r}$ , where  $\alpha$  is the threshold with maximum value to be 1, and it is usually chosen to be  $\alpha = 0.997$ . After the effective rank  $\hat{r}$  has been determined, we replace the  $\hat{r}$  in Equation (14.3),

$$\hat{\mathbf{x}}_{LS} = \sum_{i=1}^{\hat{r}} \frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \mathbf{v}_i,$$

which can be regarded as an approximation to the LS solution  $\mathbf{x}_{LS}$ . And this solution is the LS solution of the linear equation  $\mathbf{A}_{\hat{r}} = \mathbf{b}$ , where

$$\mathbf{A}_{\hat{r}} = \sum_{i=1}^{\hat{r}} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top.$$

This filtering method introduced above is useful when matrix  $\mathbf{A}$  is noisy.

#### 14.7.3 Application: Principal Component Analysis (PCA) via the Spectral Decomposition and the SVD

Given a data set of  $n$  observations  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$  where  $\mathbf{x}_i \in \mathbb{R}^p$  for all  $i \in \{1, 2, \dots, n\}$ . Our goal is to project the data onto a low-dimensional space, say  $m < p$ . Define the sample mean vector and sample covariance matrix

$$\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \quad \text{and} \quad \mathbf{S} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top.$$

where the  $n - 1$  term in the covariance matrix is to make it to be an unbiased consistent estimator of the covariance (Lu, 2021d). Or the covariance matrix can also be defined as  $\mathbf{S} = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$  which is also a consistent estimator of covariance matrix <sup>2</sup>.

Each data point  $\mathbf{x}_i$  is then projected onto a scalar value by  $\mathbf{u}_1$  such that  $\mathbf{u}_1^\top \mathbf{x}_i$ . The mean of the projected data is obtained by  $E[\mathbf{u}_1^\top \mathbf{x}_i] = \mathbf{u}_1^\top \bar{\mathbf{x}}$ , and the variance of the projected data is given by

$$\begin{aligned}\text{Cov}[\mathbf{u}_1^\top \mathbf{x}_i] &= \frac{1}{n-1} \sum_{i=1}^n (\mathbf{u}_1^\top \mathbf{x}_i - \mathbf{u}_1^\top \bar{\mathbf{x}})^2 = \frac{1}{n-1} \sum_{i=1}^n \mathbf{u}_1^\top (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{u}_1 \\ &= \mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1.\end{aligned}$$

We want to maximize the projected variance  $\mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1$  with respect to  $\mathbf{u}_1$  where we must constrain  $\|\mathbf{u}_1\|$  to prevent  $\|\mathbf{u}_1\| \rightarrow \infty$  by setting it to be  $\mathbf{u}_1^\top \mathbf{u}_1 = 1$ . By Lagrange multiplier (see (Bishop, 2006; Boyd et al., 2004)), we have

$$\mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1 + \lambda_1(1 - \mathbf{u}_1^\top \mathbf{u}_1).$$

Trivial calculation will lead to

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \quad \xrightarrow{\text{leads to}} \quad \mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1 = \lambda_1.$$

That is,  $\mathbf{u}_1$  is an eigenvector of  $\mathbf{S}$  corresponding to eigenvalue  $\lambda_1$ . And the maximum variance projection  $\mathbf{u}_1$  is corresponding to the largest eigenvalues of  $\mathbf{S}$ . The eigenvector is known as the *first principal axis*.

Define the other principal axes by decremental eigenvalues until we have  $m$  such principal components bring about the dimension reduction. This is known as the *maximum variance formulation* of PCA (Hotelling, 1933; Bishop, 2006; Shlens, 2014). A *minimum-error formulation* of PCA is discussed in (Pearson, 1901; Bishop, 2006).

**PCA via the spectral decomposition** Now let's assume the data are centered such that  $\bar{\mathbf{x}}$  is zero, or we can set  $\mathbf{x}_i = \mathbf{x}_i - \bar{\mathbf{x}}$  to centralize the data. Let the data matrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$  contain the data observations as rows. The covariance matrix is given by

$$\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1},$$

which is a symmetric matrix, and its spectral decomposition is given by

$$\mathbf{S} = \mathbf{U} \Lambda \mathbf{U}^\top, \tag{14.4}$$

where  $\mathbf{U}$  is an orthogonal matrix of eigenvectors (columns of  $\mathbf{U}$  are eigenvectors of  $\mathbf{S}$ ), and  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$  is a diagonal matrix with eigenvalues (ordered such that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ ). The eigenvectors are called *principal axes* of the data, and they *decorrelate* the the covariance matrix. Projections of the data on the principal axes are called the *principal components*. The  $i$ -th principal component is given by the  $i$ -th column of  $\mathbf{XU}$ . If we want to reduce the dimension from  $p$  to  $m$ , we just select the first  $m$  columns of  $\mathbf{XU}$ .

---

2. Consistency: An estimator  $\theta_n$  of  $\theta$  constructed on the basis of a sample of size  $n$  is said to be consistent if  $\theta_n \xrightarrow{P} \theta$  as  $n \rightarrow \infty$ .

**PCA via the SVD** If the SVD of  $\mathbf{X}$  is given by  $\mathbf{X} = \mathbf{P}\Sigma\mathbf{Q}^\top$ , then the covariance matrix can be written as

$$\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1} = \mathbf{Q} \frac{\Sigma^2}{n-1} \mathbf{Q}^\top, \quad (14.5)$$

where  $\mathbf{Q} \in \mathbb{R}^{p \times p}$  is an orthogonal matrix and contains the right singular vectors of  $\mathbf{X}$ , and the upper left of  $\Sigma$  is a diagonal matrix containing the singular values  $\text{diag}(\sigma_1, \sigma_2, \dots)$  with  $\sigma_1 \geq \sigma_2 \geq \dots$ . The number of singular values is equal to  $\min\{n, p\}$  which will not be larger than  $p$  and some of which are zeros.

The above Equation (14.5) compared with Equation (14.4) implies Equation (14.5) is also a spectral decomposition of  $\mathbf{S}$ , since the eigenvalues in  $\Lambda$  and singular values in  $\Sigma$  are ordered in a descending way and the uniqueness of the spectral decomposition in terms of the eigenspaces (Section 13.3, p. 247).

This results in the right singular vectors  $\mathbf{Q}$  are also the principal axes which decorrelate the covariance matrix, and the singular values are related to the eigenvalues of the covariance matrix via  $\lambda_i = \frac{\sigma_i^2}{n-1}$ . To reduce the dimensionality of the data from  $p$  to  $m$ , we should select largest  $m$  singular values and the corresponding right singular vectors. This is also related to the truncated SVD (TSVD)  $\mathbf{X}_m = \sum_{i=1}^m \sigma_i \mathbf{p}_i \mathbf{q}_i^\top$  as will be shown in the next section, where  $\mathbf{p}_i$ 's and  $\mathbf{q}_i$ 's are the columns of  $\mathbf{P}$  and  $\mathbf{Q}$ .

**A byproduct of PCA via the SVD for high-dimensional data** For a principle axis  $\mathbf{u}_i$  of  $\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1}$ , we have

$$\frac{\mathbf{X}^\top \mathbf{X}}{n-1} \mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

Left multiply by  $\mathbf{X}$ , we obtain

$$\frac{\mathbf{X} \mathbf{X}^\top}{n-1} (\mathbf{X} \mathbf{u}_i) = \lambda_i (\mathbf{X} \mathbf{u}_i),$$

which implies  $\lambda_i$  is also an eigenvalue of  $\frac{\mathbf{X} \mathbf{X}^\top}{n-1} \in \mathbb{R}^{n \times n}$ , and the corresponding eigenvector is  $\mathbf{X} \mathbf{u}_i$ . This is also stated in the proof of Theorem 14.1, the existence of the SVD. If  $p \gg n$ , instead of finding the eigenvector of  $\mathbf{S}$ , i.e., the principle axes of  $\mathbf{S}$ , we can find the eigenvector of  $\frac{\mathbf{X} \mathbf{X}^\top}{n-1}$ . This reduces the complexity from  $O(p^3)$  to  $O(n^3)$ . Suppose now, the eigenvector of  $\frac{\mathbf{X} \mathbf{X}^\top}{n-1}$  is  $\mathbf{v}_i$  corresponding to nonzero eigenvalue  $\lambda_i$ ,

$$\frac{\mathbf{X} \mathbf{X}^\top}{n-1} \mathbf{v}_i = \lambda_i \mathbf{v}_i.$$

Left multiply by  $\mathbf{X}^\top$ , we obtain

$$\frac{\mathbf{X}^\top \mathbf{X}}{n-1} (\mathbf{X}^\top \mathbf{v}_i) = \mathbf{S} (\mathbf{X}^\top \mathbf{v}_i) = \lambda_i (\mathbf{X}^\top \mathbf{v}_i),$$

i.e., the eigenvector  $\mathbf{u}_i$  of  $\mathbf{S}$ , is proportional to  $\mathbf{X}^\top \mathbf{v}_i$ , where  $\mathbf{v}_i$  is the eigenvector of  $\frac{\mathbf{X} \mathbf{X}^\top}{n-1}$  corresponding to the same eigenvalue  $\lambda_i$ . A further normalization step is needed to make  $\|\mathbf{u}_i\| = 1$ .

#### 14.7.4 Application: Low-Rank Approximation

For a low-rank approximation problem, there are basically two types related due to the interplay of rank and error: *fixed-precision approximation problem* and *fixed-rank approximation problem*. In the fixed-precision approximation problem, for a given matrix  $\mathbf{A}$  and a given tolerance  $\epsilon$ , one wants to find a matrix  $\mathbf{B}$  with rank  $r = r(\epsilon)$  such that  $\|\mathbf{A} - \mathbf{B}\| \leq \epsilon$  in an appropriate matrix norm. On the contrary, in the fixed-rank approximation problem, one looks for a matrix  $\mathbf{B}$  with fixed rank  $k$  and an error  $\|\mathbf{A} - \mathbf{B}\|$  as small as possible. In this section, we will consider the latter. Some excellent examples can also be found in (Kishore Kumar and Schneider, 2017; Martinsson, 2019).

Suppose we want to approximate matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r$  by a rank  $k < r$  matrix  $\mathbf{B}$ . The approximation is measured by spectral norm:

$$\mathbf{B} = \arg \min_{\mathbf{B}} \|\mathbf{A} - \mathbf{B}\|_2,$$

where the spectral norm is defined as follows:

##### Definition 14.3: Spectral Norm

The spectral norm of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is defined as

$$\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Ax}\|_2}{\|\mathbf{x}\|_2} = \max_{\mathbf{u} \in \mathbb{R}^n : \|\mathbf{u}\|_2=1} \|\mathbf{Ax}\|_2,$$

which is also the maximal singular value of  $\mathbf{A}$ , i.e.,  $\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$ .

Then, we can recover the best rank- $k$  approximation by the following theorem.

##### Theorem 14.4: (Eckart-Young-Misky Theorem w.r.t. Spectral Norm)

Given matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $1 \leq k \leq \text{rank}(\mathbf{A}) = r$ , and let  $\mathbf{A}_k$  be the truncated SVD (TSVD) of  $\mathbf{A}$  with the largest  $k$  terms, i.e.,  $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  from SVD of  $\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  by zeroing out the  $r - k$  trailing singular values of  $\mathbf{A}$ . Then  $\mathbf{A}_k$  is the best rank- $k$  approximation to  $\mathbf{A}$  in terms of the spectral norm.

**Proof** [of Theorem 14.4] We need to show for any matrix  $\mathbf{B}$ , if  $\text{rank}(\mathbf{B}) = k$ , then  $\|\mathbf{A} - \mathbf{B}\|_2 \geq \|\mathbf{A} - \mathbf{A}_k\|_2$ .

Since  $\text{rank}(\mathbf{B}) = k$ , then  $\dim(\mathcal{N}(\mathbf{B})) = n - k$ . As a result, any  $k + 1$  basis in  $\mathbb{R}^n$  intersects  $\mathcal{N}(\mathbf{B})$ . As shown in Lemma 14.1,  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A}^\top) \subset \mathbb{R}^n$ , so that we can choose the first  $k + 1$   $\mathbf{v}_i$ 's as a  $k + 1$  basis for  $\mathbb{R}^n$ . Let  $\mathbf{V}_{k+1} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k+1}]$ , then there is a vector  $\mathbf{x}$  that

$$\mathbf{x} \in \mathcal{N}(\mathbf{B}) \cap \mathcal{C}(\mathbf{V}_{k+1}), \quad s.t. \quad \|\mathbf{x}\|_2 = 1.$$



**Figure 14.4:** A gray flag image to be compressed.

That is  $\mathbf{x} = \sum_{i=1}^{k+1} a_i \mathbf{v}_i$ , and  $\|\sum_{i=1}^{k+1} a_i \mathbf{v}_i\|_2 = \sum_{i=1}^{k+1} a_i^2 = 1$ . Thus,

$$\begin{aligned}
 \|\mathbf{A} - \mathbf{B}\|_2^2 &\geq \|(\mathbf{A} - \mathbf{B})\mathbf{x}\|_2^2 \cdot \|\mathbf{x}\|_2^2, & (\text{From defintion of spectral norm}) \\
 &= \|\mathbf{Ax}\|_2^2, & (\mathbf{x} \text{ in null space of } \mathbf{B}) \\
 &= \sum_{i=1}^{k+1} \sigma_i^2 (\mathbf{v}_i^\top \mathbf{x})^2, & (\mathbf{x} \text{ orthogonal to } \mathbf{v}_{k+2}, \dots, \mathbf{v}_r) \\
 &\geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} (\mathbf{v}_i^\top \mathbf{x})^2, & (\sigma_{k+1} \leq \sigma_k \leq \dots \leq \sigma_1) \\
 &\geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} a_i^2, & (\mathbf{v}_i^\top \mathbf{x} = a_i) \\
 &= \sigma_{k+1}^2.
 \end{aligned}$$

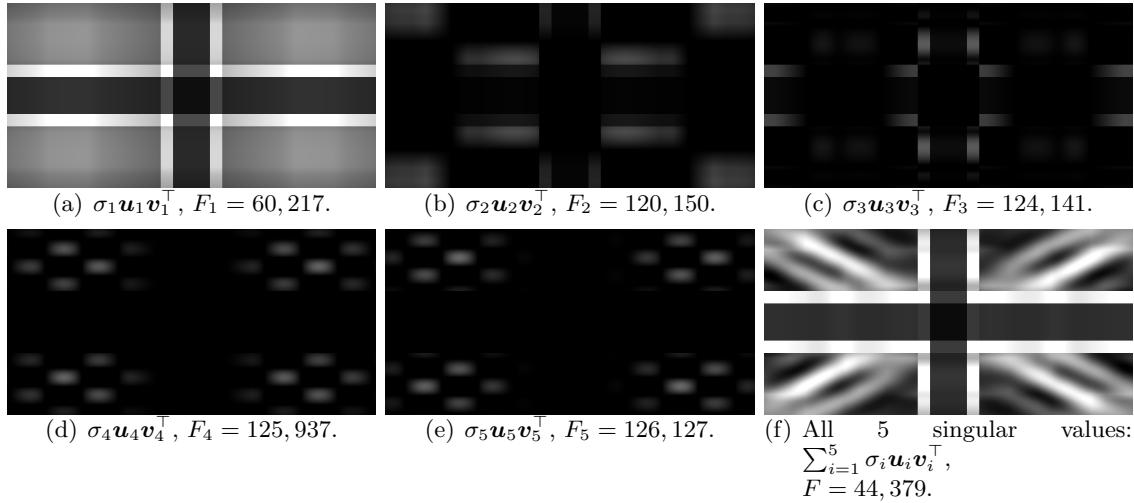
It is trivial that  $\|\mathbf{A} - \mathbf{A}_k\|_2^2 = \|\sum_{i=k+1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top\|_2^2 = \sigma_{k+1}^2$ . Thus,  $\|\mathbf{A} - \mathbf{A}_k\|_2 \leq \|\mathbf{A} - \mathbf{B}\|_2$ , which completes the proof.  $\blacksquare$

Moreover, readers can prove that  $\mathbf{A}_k$  is the best rank- $k$  approximation to  $\mathbf{A}$  in terms of the Frobenius norm. The minimal error is given by the Euclidean norm of the singular values that have been zeroed out in the process  $\|\mathbf{A} - \mathbf{A}_k\|_F = \sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \dots + \sigma_r^2}$ .

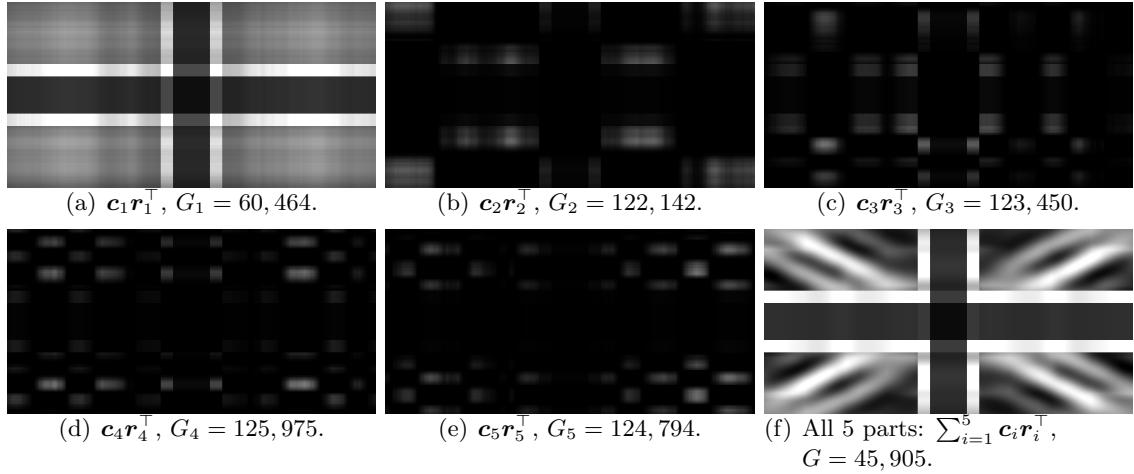
SVD gives the best approximation of a matrix. As mentioned in (Stewart, 1998; Kishore Kumar and Schneider, 2017), *the singular value decomposition is the creme de la creme of rank-reducing decompositions — the decomposition that all others try to beat*. And also *The SVD is the climax of this linear algebra course* in (Strang, 2009).

Figure 14.4 shows an example of a gray image to be compressed. The size of the image is  $600 \times 1200$  with a rank of 402.

In Figure 14.5, we approximate the image into a rank-5 matrix by truncated SVD:  $\mathbf{A} \approx \sum_{i=1}^5 \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ . It is known that the singular values contain the spectrum information with higher singular values containing lower-frequency information. And low-frequency contains more useful information (Leondes, 1995). We find that the image,  $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top$ , reconstructed by the first singular value  $\sigma_1$ , first left singular vector  $\mathbf{u}_1$ , and first right singular vector  $\mathbf{v}_1$  is very close to the original flag image, and second to the fifth image reconstructed by



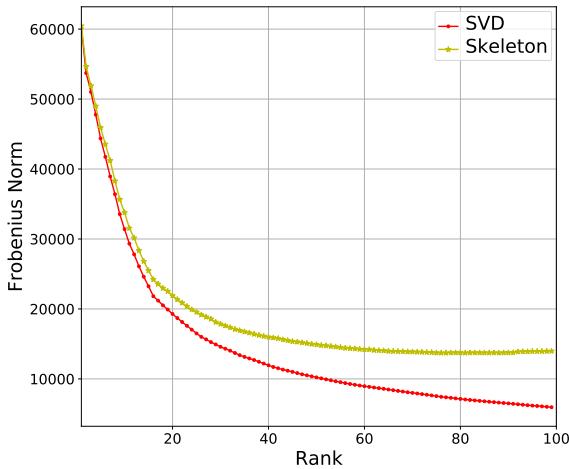
**Figure 14.5:** Image compression for gray flag image into a rank-5 matrix via the SVD, and decompose into 5 parts where  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_5$ , i.e.,  $F_1 \leq F_2 \leq \dots \leq F_5$  with  $F_i = \|\sigma_i \mathbf{u}_i \mathbf{v}_i^\top - \mathbf{A}\|_F$  for  $i \in \{1, 2, \dots, 5\}$ . And reconstruct images by single singular value and its corresponding left and right singular vectors.



**Figure 14.6:** Image compression for gray flag image into a rank-5 matrix via the Pseudoskeleton decomposition, and decompose into 5 parts where  $G_i = \|\mathbf{c}_i \mathbf{r}_i^\top - \mathbf{A}\|_F$  for  $i \in \{1, 2, \dots, 5\}$  and  $G_1 \leq G_2 \leq \dots \leq G_5$ . And reconstruct images by  $\mathbf{c}_i \mathbf{r}_i^\top$ .

the corresponding singular values and singular vectors contain more details of the flag to reconstruct it.

Similar results can be observed for the low-rank approximation via the pseudoskeleton decomposition (Section 6.8, p. 175). In Equation (6.1) (p. 176), we derived the low-rank approximation by  $\mathbf{A} \approx \mathbf{C}_2 \mathbf{R}_2$  where  $\mathbf{C}_2 \in \mathbb{R}^{m \times \gamma}$ ,  $\mathbf{R}_2 \in \mathbb{R}^{\gamma \times n}$  if  $\mathbf{A} \in \mathbb{R}^{m \times n}$  such that  $\mathbf{C}_2$



**Figure 14.7:** Comparison of reconstruction errors measured by Frobenius norm between the SVD and the pseudoskeleton approximation.

and  $\mathbf{R}_2$  are rank- $\gamma$  matrices. Suppose  $\gamma = 5$ , and

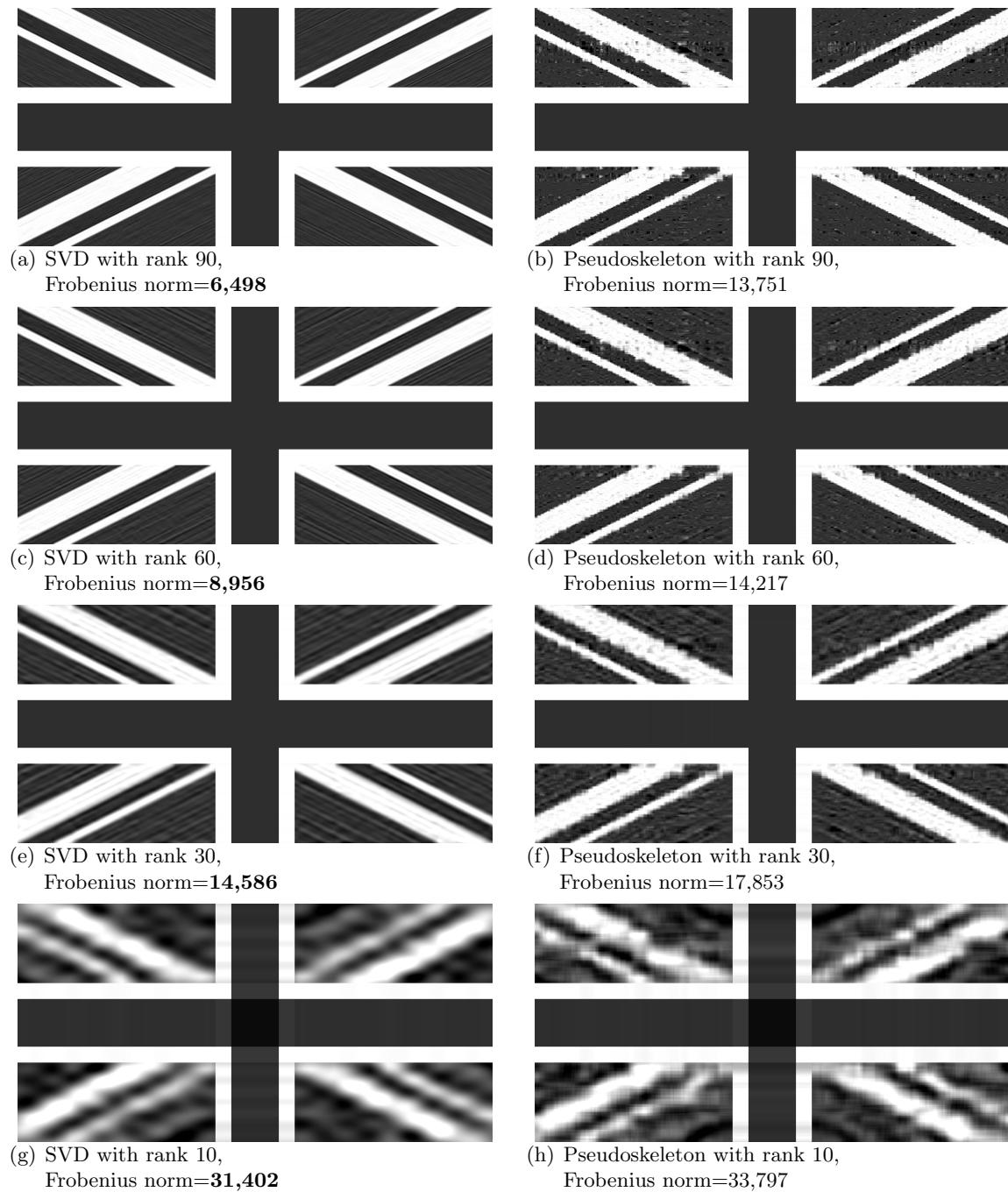
$$\mathbf{C}_2 = [\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_5], \quad \text{and} \quad \mathbf{R}_2 = \begin{bmatrix} \mathbf{r}_1^\top \\ \mathbf{r}_2^\top \\ \vdots \\ \mathbf{r}_5^\top \end{bmatrix},$$

are the column and row partitions of  $\mathbf{C}_2$ ,  $\mathbf{R}_2$  respectively. Then  $\mathbf{A}$  can be approximated by  $\sum_{i=1}^5 \mathbf{c}_i \mathbf{r}_i^\top$ . The partitions are ordered such that

$$\underbrace{\|\mathbf{c}_1 \mathbf{r}_1^\top - \mathbf{A}\|_F}_{G_1} \leq \underbrace{\|\mathbf{c}_2 \mathbf{r}_2^\top - \mathbf{A}\|_F}_{G_2} \leq \dots \leq \underbrace{\|\mathbf{c}_5 \mathbf{r}_5^\top - \mathbf{A}\|_F}_{G_5}.$$

We observe that  $\mathbf{c}_1 \mathbf{r}_1^\top$  works similarly to that of  $\sigma_1 \mathbf{u}_1 \mathbf{v}^\top$  where the reconstruction errors measured by the Frobenius norm are very close (60,464 in the pseudoskeleton case compared to that of 60,217 in the SVD case). This is partly because the pseudoskeleton decomposition relies on the SVD (Section 6.8, p. 175) such that  $\mathbf{c}_1 \mathbf{r}_1^\top$  internally has the largest “singular value” in the sense.

We finally compare low-rank approximation between the SVD and the pseudoskeleton with different ranks. Figure 14.8 shows the difference of each compression with ranks of 90, 60, 30, 10. We observe that the SVD does well with ranks of 90, 60, 30. The pseudoskeleton-approximation compresses well in the black horizontal and vertical lines in the image. But it performs poorly in the details of the flag. Figure 14.7 shows the comparison of the reconstruction errors between the SVD and the pseudoskeleton approximation measured by Frobenius norm ranging from rank 1 to 100 where we find in all cases, the truncated SVD does better in terms of Frobenius norm. Similar results can be observed when applied to the spectral norm.



**Figure 14.8:** Image compression for gray flag image with different ranks.

# Chapter 15

# Eigenvalue Problem

## Contents

---

|        |                                                                            |     |
|--------|----------------------------------------------------------------------------|-----|
| 15.1   | Background                                                                 | 291 |
| 15.2   | Rate of Convergence                                                        | 291 |
| 15.3   | Eigenvalues as Optimization                                                | 293 |
| 15.4   | Rayleigh Quotient                                                          | 293 |
| 15.5   | Power Method, Inverse Power Method, and Rayleigh Quotient Method           | 295 |
| 15.5.1 | The Power Method                                                           | 295 |
| 15.5.2 | The Inverse Power Method                                                   | 302 |
| 15.5.3 | The Shifted Inverse Power Method                                           | 303 |
| 15.5.4 | The Rayleigh Quotient Method                                               | 305 |
| 15.6   | QR Algorithm                                                               | 305 |
| 15.6.1 | Preliminary: Power Iteration with Eigenvector Known                        | 305 |
| 15.6.2 | Preliminary: Power Iteration with Eigenvector Unknown                      | 307 |
| 15.6.3 | Preliminary: Power Iteration with Eigenvector Unknown and QR Decomposition | 308 |
| 15.6.4 | A Simple QR Algorithm from Power Iteration: without Shifts                 | 309 |
| 15.6.5 | A Practical QR Algorithm: with Shifts                                      | 312 |
| 15.7   | Apply the Practical QR Algorithm to Tridiagonal Matrices                   | 314 |
| 15.7.1 | Explicit Shifted QR Algorithm                                              | 314 |
| 15.7.2 | Implicit Shifted QR Algorithm                                              | 315 |
| 15.8   | Jacobi's Method                                                            | 320 |
| 15.8.1 | The 2 by 2 Case                                                            | 320 |
| 15.8.2 | The Complete Jacobi's Method                                               | 322 |
| 15.8.3 | The Cyclic-by-Row Jacobi's Method                                          | 323 |
| 15.8.4 | Other Issues                                                               | 324 |
| 15.9   | Computing the SVD                                                          | 324 |
| 15.9.1 | Implicit Shifted QR Algorithm                                              | 324 |
| 15.9.2 | Jacobi's SVD Method                                                        | 329 |



### 15.1. Background

The decompositional methods discussed in the above sections are all related to the eigenvalues and eigenvectors of matrices. Thus, the eigenvalue problem merits independent consideration. In this section, we present some classical eigenvalue algorithms that will be often useful for computing eigenvalue-related decompositions. And we simplify the discussion by considering only matrices that are real and symmetric. When  $\mathbf{A}$  is real and symmetric, it can be factored into (by spectral theorem, Theorem 13.1, p. 241)

$$\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top = \mathbf{Q}\Lambda\mathbf{Q}^{-1},$$

where the columns of  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$  are eigenvectors of  $\mathbf{A}$  whilst mutually orthonormal, the entries of  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  are the corresponding eigenvalues of  $\mathbf{A}$ , which are real. We suppose further the eigenvalues are ordered in magnitude such that

$$|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n| \geq 0,$$

where in some discussions the equality will not be satisfied with special considerations. One thing to note further is that since  $\mathbf{A}$  is diagonalizable, the eigenvectors  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$  span the whole space  $\mathbb{R}^n$  such that any vector  $\mathbf{v}$  in space  $\mathbb{R}^n$  can be written as a combination of the eigenvectors

$$\mathbf{v} = x_1\mathbf{q}_1 + x_2\mathbf{q}_2 + \dots + x_n\mathbf{q}_n, \quad \text{for all } \mathbf{v} \in \mathbb{R}^n.$$

### 15.2. Rate of Convergence

Before we talk about specific algorithms to compute the eigenvalues, we need criteria to evaluate how fast the algorithms will be as most of them are iterative methods. We define the convergence of a sequence as follows. Note that the  $k$ -th element in a sequence is denoted by a superscript in parentheses, e.g.,  $\mathbf{A}^{(k)}$  denotes the  $k$ -th matrix in a sequence, and  $\mathbf{a}^{(k)}$  denote the  $k$ -th vector in a sequence.

#### Definition 15.1: Convergence of a Sequence

Let  $\alpha^{(1)}, \alpha^{(2)}, \dots \in \mathbb{R}$  be an infinite sequence of scalars. Then  $\alpha^{(k)}$  is said to converge to  $\alpha^*$  if

$$\lim_{k \rightarrow \infty} |\alpha^{(k)} - \alpha^*| = 0.$$

Similarly, let  $\boldsymbol{\alpha}^{(1)}, \boldsymbol{\alpha}^{(2)}, \dots \in \mathbb{R}^n$  be an infinite sequence of vectors. Then  $\boldsymbol{\alpha}^{(k)}$  is said to converge to  $\boldsymbol{\alpha}^*$  if

$$\lim_{k \rightarrow \infty} \|\boldsymbol{\alpha}^{(k)} - \boldsymbol{\alpha}^*\| = 0.$$

The convergence of a sequence of vectors or matrices rely on the norm. One should note that by the equivalence of vector norms (Theorem 27.4, p. 480), if a sequence of vectors converges in one norm, then it converges in all norms. This will be proved important for the analysis of the convergence of eigenvectors in the sequel.

**Definition 15.2: Linear Convergence**

A sequence  $\alpha^{(k)}$  with limit  $\alpha^*$  is *linearly convergence* if there exists a constant  $c \in (0, 1)$  such that

$$|\alpha^{(k+1)} - \alpha^*| \leq c|\alpha^{(k)} - \alpha^*|.$$

In other words, the *linearly convergent sequence* has the following property:

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|} = c \in (0, 1).$$

For example, the sequence  $\alpha^{(k)} = 4 + (1/4)^k$  converges linearly to  $\alpha^* = 4$  since

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|} = \frac{1}{4} \in (0, 1).$$

**Definition 15.3: Superlinear Convergence**

A sequence  $\alpha^{(k)}$  with limit  $\alpha^*$  is *superlinearly convergence* if there exists a constant  $c_k > 0$  with  $c_k \rightarrow 0$  such that

$$|\alpha^{(k+1)} - \alpha^*| \leq c_k |\alpha^{(k)} - \alpha^*|.$$

In other words, the *superlinearly convergent sequence* has the following property:

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|} = 0.$$

For example, the sequence  $\alpha^{(k)} = 4 + \left(\frac{1}{k+4}\right)^{k+3}$  converges superlinearly to  $\alpha^* = 4$  since

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|} = \left(\frac{k+4}{k+5}\right)^{k+3} \frac{1}{k+5} = 0.$$

**Definition 15.4: Quadratic Convergence**

A sequence  $\alpha^{(k)}$  with limit  $\alpha^*$  is *quadratically convergence* if there exists a constant  $c > 0$  such that

$$|\alpha^{(k+1)} - \alpha^*| \leq c|\alpha^{(k)} - \alpha^*|^2.$$

In other words, the *quadratically convergent sequence* has the following property:

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|^2} = c.$$

For example, the sequence  $\alpha^{(k)} = 4 + (1/4)^{2^k}$  converges quadratically to  $\alpha^* = 4$  since

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|^2} = 1.$$

### Definition 15.5: Cubic Convergence

A sequence  $\alpha^{(k)}$  with limit  $\alpha^*$  is *cubically convergence* if there exists a constant  $c > 0$  such that

$$|\alpha^{(k+1)} - \alpha^*| \leq c|\alpha^{(k)} - \alpha^*|^3.$$

In other words, the *cubically convergent sequence* has the following property:

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|^3} = c.$$

For example, the sequence  $\alpha^{(k)} = 4 + (1/4)^{3^k}$  converges cubically to  $\alpha^* = 4$  since

$$\lim_{k \rightarrow \infty} \frac{|\alpha^{(k+1)} - \alpha^*|}{|\alpha^{(k)} - \alpha^*|^3} = 1.$$

### 15.3. Eigenvalues as Optimization

For real and symmetric matrix  $\mathbf{A}$ , we consider the following constrained optimization

$$\max_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^\top \mathbf{A} \mathbf{x}, \quad s.t., \quad \|\mathbf{x}\|_2 = 1.$$

By forming the Lagrangian, the optimization can be transformed into

$$\mathcal{L}(\mathbf{x}, \lambda) = \mathbf{x}^\top \mathbf{A} \mathbf{x} - \lambda \mathbf{x}^\top \mathbf{x},$$

where  $\lambda$  is called the *Lagrange multiplier*. To find the solution, the gradient of the Lagrangian has to be zero at  $\mathbf{x}^*$ :

$$\Delta_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \lambda) = 2\mathbf{A}\mathbf{x} - 2\lambda\mathbf{x} = \mathbf{0}.$$

This implies  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$  and shows that the optimal points indicating  $\lambda$  and  $\mathbf{x}^*$  are eigenvalue and eigenvector of  $\mathbf{A}$ .

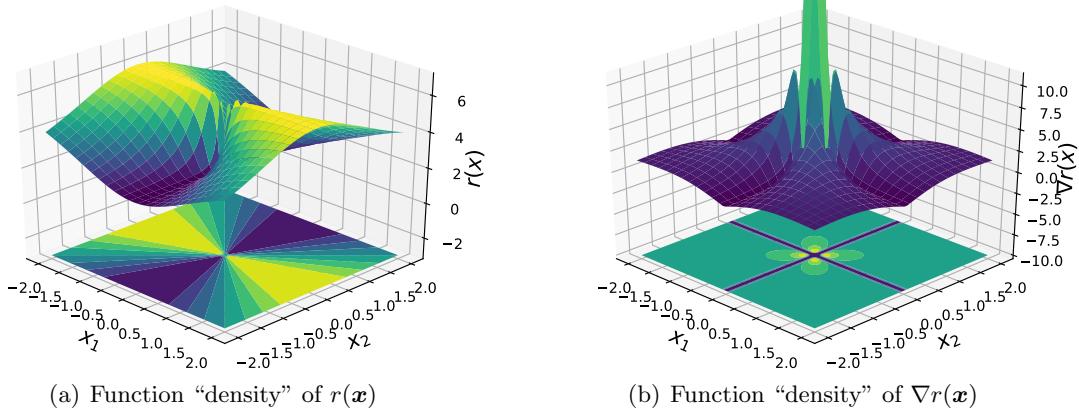
### 15.4. Rayleigh Quotient

The Rayleigh quotient of a vector  $\mathbf{x} \in \mathbb{R}^n$  associated with the matrix  $\mathbf{A}$  is the scalar given by the quadratic form:

$$r(\mathbf{x}) = \frac{\mathbf{x}^\top \mathbf{A} \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} = \left( \frac{\mathbf{x}}{\|\mathbf{x}\|} \right)^\top \mathbf{A} \left( \frac{\mathbf{x}}{\|\mathbf{x}\|} \right),$$

where  $\left(\frac{\mathbf{x}}{\|\mathbf{x}\|}\right)$  is a normalized vector. When  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}$ ,  $r(\mathbf{x})$  is the eigenvalue  $\lambda$  associated with  $\mathbf{x}$ . To see this, suppose  $\mathbf{Ax} = \lambda\mathbf{x}$ , it follows that

$$\mathbf{x}^\top \mathbf{Ax} = \lambda \mathbf{x}^\top \mathbf{x} \quad \xrightarrow{\text{leads to}} \quad \lambda = \frac{\mathbf{x}^\top \mathbf{Ax}}{\mathbf{x}^\top \mathbf{x}} = r(\mathbf{x}).$$



**Figure 15.1:** Function “density” and contour plots (blue=low, yellow=high) where in Figure 15.1(a), the upper graph is the “density”, and the lower one is the projection of it (i.e., contour). The example is drawn by setting  $\mathbf{A} = \begin{bmatrix} 6 & 0 \\ 0 & 2 \end{bmatrix}$  where the input  $\mathbf{x} = [x_1, x_2]^\top$  lies in  $\mathbb{R}^2$ . The eigenvalues are 6 and 2, where the corresponding eigenvectors lie in the lines of  $z \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix}$  and  $z \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix}$  with  $z \in (-\infty, \infty)$  being a scalar.

Whereas, the most important property of the Rayleigh quotient is that when  $\mathbf{x}$  is not an eigenvector of  $\mathbf{A}$ , the scalar  $r(\mathbf{x})$  acts most like an eigenvalue in the sense that the squared norm  $\|\mathbf{Ax} - r(\mathbf{x})\mathbf{x}\|^2$  is minimized. To see this, suppose we want to find  $\lambda$  such that  $\|\mathbf{Ax} - \lambda\mathbf{x}\|^2$  is minimized. Write out the equation with respect to  $\lambda$

$$\|\mathbf{Ax} - r(\mathbf{x})\mathbf{x}\|^2 = \mathbf{x}^\top \mathbf{x} \lambda^2 - 2\mathbf{x}^\top \mathbf{Ax} \lambda + \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax}.$$

Since  $\mathbf{x}^\top \mathbf{x} \geq 0$ , the above equation is minimized by setting the gradient to be zero which leads to  $\lambda = r(\mathbf{x})$ , i.e., the Rayleigh quotient of  $\mathbf{x}$ . Therefore, the Rayleigh quotient is a natural eigenvalue estimate to consider if  $\mathbf{x}$  is close to, but not necessarily equal to, an eigenvector. To see this, it is reasonable to take the vector  $\mathbf{x} \in \mathbb{R}^n$  as an input variable, and  $r(\mathbf{x})$  as an output. Let  $a = \mathbf{x}^\top \mathbf{Ax}$ ,  $b = \mathbf{x}^\top \mathbf{x}$ , the gradient of  $r(\mathbf{x})$  with respect to  $\mathbf{x}$  is given by

$$\Delta r(\mathbf{x}) = \frac{a'b - ab'}{b^2} = \frac{\mathbf{x}^\top \mathbf{x} (\mathbf{A} + \mathbf{A}^\top) \mathbf{x} - 2\mathbf{x}^\top \mathbf{A} \mathbf{x} \mathbf{x}}{\|\mathbf{x}\|^4}. \quad (\mathbf{A} \text{ is any square matrix})$$

If we further restrict  $\mathbf{A}$  to be symmetric as we will consider mostly in this section, the gradient of the Rayleigh quotient reduces to

$$\nabla r(\mathbf{x}) = \frac{2\mathbf{x}^\top \mathbf{Ax} \mathbf{x} - 2\mathbf{x}^\top \mathbf{A} \mathbf{x} \mathbf{x}}{\|\mathbf{x}\|^4} = \frac{2}{\|\mathbf{x}\|^2} (\mathbf{Ax} - r(\mathbf{x})\mathbf{x}). \quad (\mathbf{A} \text{ is symmetric})$$

We observe that the gradient is a zero vector if and only if  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  in which case  $r(\mathbf{x})$  is the corresponding eigenvalue. This finding has a geometric meaning, when viewing  $\mathbf{x}$  as the input variable, the *stationary points* (also known as the *saddle points*) of function  $r(\mathbf{x})$  are eigenvectors of  $\mathbf{A}$  and the function output is the corresponding eigenvalues. An example of these stationary points is shown in Figure 15.1 where the  $r(\mathbf{x})$  and  $\nabla r(\mathbf{x})$  are drawn from the matrix  $\mathbf{A} = \begin{bmatrix} 6 & 0 \\ 0 & 2 \end{bmatrix}$ .

### 15.5. Power Method, Inverse Power Method, and Rayleigh Quotient Method

We start computing the eigenvalues by several partial methods, which compute the extremal eigenvalues of  $\mathbf{A}$ , i.e., the one having maximum and minimum magnitude.

#### 15.5.1 The Power Method

The power method will produce a sequence  $\mathbf{v}^{(k)}$  that converges linearly to an eigenvector corresponding to the largest eigenvalue of  $\mathbf{A}$ . To obtain the corresponding eigenvalue, the Rayleigh quotient can be utilized afterwards. The first attempt to find the eigenvalue of  $\mathbf{A}$  will be using the value of the largest eigenvalue (in magnitude)  $\lambda_1$ . We will see in this and the sections to follow, this “theoretical” algorithms will shed light on the convergence analysis of the algorithms.

---

**Algorithm 37** Power Iteration (A Theoretical but Impossible One: For Convergence Analysis Only)

---

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

- 1:  $\mathbf{v}^{(0)} = \text{some vector with } \|\mathbf{v}^{(0)}\| = 1;$
  - 2: **for**  $k = 1, 2, \dots$  **do**
  - 3:      $\mathbf{w} = \mathbf{A}\mathbf{v}^{(k-1)};$
  - 4:      $\mathbf{v}^{(k)} = \mathbf{w}/\lambda_1;$
  - 5:      $\lambda^{(k)} = \frac{(\mathbf{v}^{(k)})^\top \mathbf{A} \mathbf{v}^{(k)}}{(\mathbf{v}^{(k)})^\top \mathbf{v}^{(k)}};$  ▷ i.e., Rayleigh quotient
  - 6: **end for**
- 

Suppose the eigenvalue  $\lambda_1$  with largest magnitude is given up front, Algorithm 37 provide an iterative way to find the corresponding eigenvector. Write  $\mathbf{v}^{(0)}$  as a linear combination of the orthonormal eigenvectors  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n\}$  (since they span the whole space  $\mathbb{R}^n$ ):

$$\mathbf{v}^{(0)} = x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2 + \dots + x_n \mathbf{q}_n = \mathbf{Q} \mathbf{x} \quad (15.1)$$

leads to  $\mathbf{x} = \mathbf{Q}^{-1} \mathbf{v}^{(0)},$

where  $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top$ . Since matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric, the  $k$ -th power of  $\mathbf{A}$  is given by

$$\mathbf{A}^k = \mathbf{Q} \Lambda^k \mathbf{Q}^{-1} \quad \text{leads to} \quad \begin{cases} \mathbf{A}^k \mathbf{Q} = \mathbf{Q} \Lambda^k; \\ \mathbf{A}^k \mathbf{q}_i = \lambda_i^k \mathbf{q}_i, \end{cases} \quad (15.2)$$

where  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^{-1}$  is the spectral decomposition of  $\mathbf{A}$  (Theorem 13.1, p. 241; Remark 13.8, p. 247). Clearly, we can obtain the  $k$ -th element  $\mathbf{v}^{(k)}$  in the sequence by

$$\begin{aligned}\mathbf{v}^{(k)} &= \mathbf{A}\mathbf{v}^{(k-1)}/\lambda_1 = \mathbf{A}^k\mathbf{v}^{(0)}/\lambda_1^k \\ &= 1/\lambda_1^k(x_1\lambda_1^k\mathbf{q}_1 + x_2\lambda_2^k\mathbf{q}_2 + \dots + x_n\lambda_n^k\mathbf{q}_n) \quad (\text{Equation (15.1) and (15.2)}) \\ &= x_1\mathbf{q}_1 + x_2\left(\frac{\lambda_2}{\lambda_1}\right)^k\mathbf{q}_2 + \dots + x_n\left(\frac{\lambda_n}{\lambda_1}\right)^k\mathbf{q}_n.\end{aligned}$$

Then, it follows that

$$\begin{aligned}\mathbf{v}^{(k)} - x_1\mathbf{q}_1 &= \mathbf{A}^k\mathbf{v}^{(0)}/\lambda_1^k - x_1\mathbf{q}_1 \\ &= (\mathbf{Q}\Lambda^k\mathbf{Q}^{-1})\mathbf{v}^{(0)}/\lambda_1^k - x_1\mathbf{q}_1 \\ &\stackrel{x=\mathbf{Q}^{-1}\mathbf{v}^{(0)}}{=} \mathbf{Q} \begin{bmatrix} 1 & & & \\ & \left(\frac{\lambda_2}{\lambda_1}\right)^k & & \\ & & \ddots & \\ & & & \left(\frac{\lambda_n}{\lambda_1}\right)^k \end{bmatrix} \mathbf{x} - x_1\mathbf{q}_1 = \mathbf{Q} \begin{bmatrix} 0 & & & \\ & \left(\frac{\lambda_2}{\lambda_1}\right)^k & & \\ & & \ddots & \\ & & & \left(\frac{\lambda_n}{\lambda_1}\right)^k \end{bmatrix} \mathbf{x}.\end{aligned}$$

Therefore,

$$\mathbf{Q}^{-1}(\mathbf{v}^{(k)} - x_1\mathbf{q}_1) = \begin{bmatrix} 0 & & & \\ & \left(\frac{\lambda_2}{\lambda_1}\right)^k & & \\ & & \ddots & \\ & & & \left(\frac{\lambda_n}{\lambda_1}\right)^k \end{bmatrix} \mathbf{x},$$

and

$$\mathbf{Q}^{-1}(\mathbf{v}^{(k+1)} - x_1\mathbf{q}_1) = \begin{bmatrix} 0 & & & \\ & \left(\frac{\lambda_2}{\lambda_1}\right)^{k+1} & & \\ & & \ddots & \\ & & & \left(\frac{\lambda_n}{\lambda_1}\right)^{k+1} \end{bmatrix} \mathbf{x} = \begin{bmatrix} 0 & & & \\ & \left(\frac{\lambda_2}{\lambda_1}\right) & & \\ & & \ddots & \\ & & & \left(\frac{\lambda_n}{\lambda_1}\right) \end{bmatrix} \mathbf{Q}^{-1}(\mathbf{v}^{(k)} - x_1\mathbf{q}_1).$$

Define  $\|\mathbf{B}\|_{\mathbf{Q}^{-1}}$  by  $\|\mathbf{Q}^{-1}\mathbf{B}\|_2$  for matrix  $\mathbf{B}$ , this results in

$$\begin{aligned}\|\mathbf{v}^{(k+1)} - x_1\mathbf{q}_1\|_{\mathbf{Q}^{-1}} &= \left\| \mathbf{Q}^{-1}(\mathbf{v}^{(k+1)} - x_1\mathbf{q}_1) \right\|_2 \\ &\leq \left| \frac{\lambda_2}{\lambda_1} \right| \cdot \left\| \mathbf{Q}^{-1}(\mathbf{v}^{(k)} - x_1\mathbf{q}_1) \right\|_2 = \left| \frac{\lambda_2}{\lambda_1} \right| \cdot \left\| \mathbf{v}^{(k)} - x_1\mathbf{q}_1 \right\|_{\mathbf{Q}^{-1}},\end{aligned} \tag{15.3}$$

where the inequality is from the matrix-vector inequality (Remark 27.10, p. 482). The above deduction shows that if we can prove that  $\|\mathbf{v}\|_{\mathbf{X}^{-1}} = \|\mathbf{X}^{-1}\mathbf{v}\|_2$  is a reasonable vector norm that satisfies the three criteria of vector norm (Definition 27.1, p. 476), then by the equivalence of vector norms (Theorem 27.4, p. 480), we prove  $\mathbf{v}^{(k)}$  in Algorithm 37 converges to  $x_1\mathbf{q}_1$  linearly (Definition 15.2, p. 292). This is actually the case.

However, the problem of Algorithm 37 is that the  $\lambda_1$  is unknown to us making it impractical for us to compute the eigenvector. To tame this problem, we scale it to be of unit length at each iteration. The algorithm is then formulated in Algorithm 38 where the difference is shown in blue text.

---

**Algorithm 38** Power Iteration (A Practical One, Compare to Algorithm 37)

---

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

- 1:  $\mathbf{v}^{(0)}$  = some vector with  $\|\mathbf{v}^{(0)}\| = 1$ ;
  - 2: **for**  $k = 1, 2, \dots$  **do**
  - 3:    $\mathbf{w} = \mathbf{A}\mathbf{v}^{(k-1)}$ ;
  - 4:    $\mathbf{v}^{(k)} = \mathbf{w}/\|\mathbf{w}\|$ ;
  - 5:    $\lambda^{(k)} = (\mathbf{v}^{(k)})^\top \mathbf{A}\mathbf{v}^{(k)}$ ; ▷ i.e., Rayleigh quotient
  - 6: **end for**
- 

**Convergence Analysis** Clearly,  $\mathbf{v}^{(k)}$  is still a multiple of  $\mathbf{A}^k \mathbf{v}^{(0)}$  such that  $\mathbf{v}^{(k)} = c_k \mathbf{A}^k \mathbf{v}^{(0)} = \frac{\mathbf{A}^k \mathbf{v}^{(0)}}{\|\mathbf{A}^k \mathbf{v}^{(0)}\|}$ . We have

$$\begin{aligned} \mathbf{v}^{(k)} &= c_k \mathbf{A}^k \mathbf{v}^{(0)} \\ &= c_k (x_1 \lambda_1^k \mathbf{q}_1 + x_2 \lambda_2^k \mathbf{q}_2 + \dots + x_n \lambda_n^k \mathbf{q}_n) \\ &= c_k \lambda_1^k \left( x_1 \mathbf{q}_1 + \underbrace{x_2 \left( \frac{\lambda_2}{\lambda_1} \right)^k \mathbf{q}_2 + \dots + x_n \left( \frac{\lambda_n}{\lambda_1} \right)^k \mathbf{q}_n}_{\mathbf{y}^{(k)}} \right). \end{aligned} \quad (15.4)$$

The sequence  $\mathbf{y}^{(k)}$  in the above equation vanishes as  $k \rightarrow \infty$  if  $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0$ . Further, if  $x_1 \neq 0$ , then as  $k \rightarrow \infty$ , the vector sequence  $\mathbf{v}^{(k)}$  converges to  $\pm \mathbf{q}_1$  where the sign is decided by  $\lambda_1^k x_1$ .

Now since we know  $\mathbf{v}^{(k)}$  converges to  $\pm \mathbf{q}_1$  such that  $c_k \lambda_1^k x_1 \rightarrow \pm 1$ . Following from the discussion in (Quarteroni et al., 2010), we here provide an improved result on the convergence of the eigenvector  $\mathbf{v}^{(k)}$  with a lower bound such that a form in the linear convergence can be obtained (Definition 15.2, p. 292). This observation proceeds the following theorem.

**Theorem 15.1: (Convergence of Power Iteration: Eigenvector)**

Suppose  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric. Suppose further that the following two conditions are met

- $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0$ , i.e.,  $\lambda_1$  has algebraic multiplicity being equal to 1;
- $\mathbf{q}_1^\top \mathbf{v}^{(0)} \neq 0$ , i.e., the initial guess  $\mathbf{v}^{(0)}$  has a component in the direction of the eigenvector  $\mathbf{q}_1$  associated with the eigenvalue  $\lambda_1$ .

Then there exists a constant  $c$  such that

$$\|\tilde{\mathbf{v}}^{(k)} - \mathbf{q}_1\| \leq c \cdot \left| \frac{\lambda_2}{\lambda_1} \right|^k,$$

where the sequence  $\tilde{\mathbf{v}}^{(k)} = \frac{\mathbf{v}^{(k)}}{c_k \lambda_1^k x_1} = \mathbf{q}_1 + \sum_{i=2}^n \frac{x_i}{x_1} \left( \frac{\lambda_i}{\lambda_1} \right)^k \mathbf{q}_i$ .

**Proof** [of Theorem 15.1] We notice that  $\|\tilde{\mathbf{v}}^{(k)} - \mathbf{q}_1\| = \left\| \sum_{i=2}^n \frac{x_i}{x_1} \left( \frac{\lambda_i}{\lambda_1} \right)^k \mathbf{q}_i \right\|$ . Let  $\mathbf{z} = \left[ 0, \frac{x_2}{x_1} \left( \frac{\lambda_2}{\lambda_1} \right)^k, \frac{x_3}{x_1} \left( \frac{\lambda_3}{\lambda_1} \right)^k, \dots, \frac{x_n}{x_1} \left( \frac{\lambda_n}{\lambda_1} \right)^k \right]$ ,  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3, \dots, \mathbf{q}_n]$ , it follows that

$$\begin{aligned} \|\tilde{\mathbf{v}}^{(k)} - \mathbf{q}_1\| &= \|\mathbf{Q}\mathbf{z}\| \stackrel{*}{=} \|\mathbf{z}\| \\ &= \left( \sum_{i=2}^n \left( \frac{x_i}{x_1} \right)^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} \right)^{1/2} \leq \left| \frac{\lambda_2}{\lambda_1} \right|^k \left( \sum_{i=2}^n \left( \frac{x_i}{x_1} \right)^2 \right)^{1/2}, \end{aligned}$$

where the inequality comes from the Matrix-vector product inequality (Remark 27.10, p. 482, or simply by the assumption  $|\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$ ) and the equality  $(*)$  comes from the length is preserved under orthogonal transformation (similar to Lemma 27.6, p. 480).

Let  $c = \left( \sum_{i=2}^n \left( \frac{x_i}{x_1} \right)^2 \right)^{1/2}$ , the result follows. Note in the discussion of (Quarteroni et al., 2010), the equality  $(*)$  is replace by a inequality from Matrix-vector product inequality (Remark 27.10, p. 482 such that  $\|\mathbf{Q}\mathbf{z}\| \leq \|\mathbf{Q}\| \cdot \|\mathbf{z}\|$ .) This has a problem that we cannot find the lower bound on the above equation. The above equation also tells us (by the assumption  $|\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$ ):

$$\begin{aligned} \|\tilde{\mathbf{v}}^{(k)} - \mathbf{q}_1\| &= \|\mathbf{Q}\mathbf{z}\| \stackrel{*}{=} \|\mathbf{z}\| \\ &= \left( \sum_{i=2}^n \left( \frac{x_i}{x_1} \right)^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} \right)^{1/2} \geq \left| \frac{\lambda_n}{\lambda_1} \right|^k \left( \sum_{i=2}^n \left( \frac{x_i}{x_1} \right)^2 \right)^{1/2}. \end{aligned}$$

Therefore, a bound on the convergence is given by

$$\left. \begin{aligned} \|\tilde{\mathbf{v}}^{(k+1)} - \mathbf{q}_1\| &\leq c \cdot \left| \frac{\lambda_2}{\lambda_1} \right|^{k+1} \\ \|\tilde{\mathbf{v}}^{(k)} - \mathbf{q}_1\| &\geq c \cdot \left| \frac{\lambda_n}{\lambda_1} \right|^k \end{aligned} \right\} \quad \xrightarrow{\text{leads to}} \quad \frac{\|\tilde{\mathbf{v}}^{(k+1)} - \mathbf{q}_1\|}{\|\tilde{\mathbf{v}}^{(k)} - \mathbf{q}_1\|} \leq \left| \frac{\lambda_2^{k+1}}{\lambda_1 \cdot \lambda_n^k} \right|.$$

However, since we do not know the value of  $\lambda_n, \lambda_2, \lambda_1$ , it happens that the above bound can explode. Only when  $|\lambda_1| \gg |\lambda_2|$ , it will be tight.  $\blacksquare$

Going further and following from the discussion in (Golub and Van Loan, 2013), we here provide an improved result on the convergence of the eigenvalue  $\lambda^{(k)}$  with a tighter bound such that the sequence  $\lambda^{(k)}$  converges to  $\lambda_1$  **quadratically** with respect to the ratio  $\left| \frac{\lambda_2}{\lambda_1} \right|$ .

**Theorem 15.2: (Convergence of Power Iteration: Eigenvalue)**

(Under the same condition as Theorem 15.1) Suppose  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric. Suppose further that the following two conditions are met

- $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0$ , i.e.,  $\lambda_1$  has algebraic multiplicity being equal to 1;
- $\mathbf{q}_1^\top \mathbf{v}^{(0)} \neq 0$ , i.e., the initial guess  $\mathbf{v}^{(0)}$  has a component in the direction of the eigenvector  $\mathbf{q}_1$  associated with the eigenvalue  $\lambda_1$ .

Then define  $\theta_k \in [0, \pi/2]$  by

$$c_k = \cos \theta_k = |\mathbf{q}_1^\top \mathbf{v}^{(k)}|.$$

The  $c_k$  is well defined since  $\|\mathbf{q}_1\| = \|\mathbf{v}^{(k)}\| = 1 \rightarrow 0 < |\mathbf{q}_1^\top \mathbf{v}^{(k)}| \leq 1$ . Then the sequence  $s_k = \sin \theta_k, t_k = \tan \theta_k$  follows

- Convergence of  $s_k$ :

$$\begin{aligned} - |s_k| &\leq \textcolor{blue}{s_0} \left| \frac{\lambda_2}{\lambda_1} \right|^k; \\ - \text{ or } |s_k| &\leq \textcolor{blue}{t_0} \left| \frac{\lambda_2}{\lambda_1} \right|^k; \end{aligned}$$

- Convergence of  $\lambda^{(k)}$ :

$$\begin{aligned} - |\lambda^{(k)} - \lambda_1| &\leq \max_{2 \leq i \leq n} |\lambda_1 - \lambda_i| \cdot \textcolor{blue}{s_0^2} \left( \frac{\lambda_i}{\lambda_1} \right)^{2k}; \\ - \text{ or } |\lambda^{(k)} - \lambda_1| &\leq \max_{2 \leq i \leq n} |\lambda_1 - \lambda_i| \cdot \textcolor{blue}{t_0^2} \left( \frac{\lambda_i}{\lambda_1} \right)^{2k}. \end{aligned}$$

**Proof** [of Theorem 15.2] Since  $s_k^2 = 1 - c_k^2 = 1 - (\mathbf{q}_1^\top \mathbf{v}^{(k)})^2 = 1 - \left( \frac{\mathbf{q}_1^\top \mathbf{A}^k \mathbf{v}^{(0)}}{\|\mathbf{A}^k \mathbf{v}^{(0)}\|} \right)^2$  by Equation (15.4) where  $\mathbf{v}^{(0)}$  can be written as the linear combination of the orthonormal eigenvectors:

$$\mathbf{v}^{(0)} = x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2 + \dots + x_n \mathbf{q}_n = \mathbf{Q} \mathbf{x} \quad \text{leads to} \quad \mathbf{x} = \mathbf{Q}^{-1} \mathbf{v}^{(0)}.$$

Since  $\|\mathbf{x}\|^2 = \|\mathbf{Q}^{-1} \mathbf{v}^{(0)}\|^2$  and  $\mathbf{v}^{(0)}$  is of length 1. By equivalence under orthogonal (similar to Lemma 27.6, p. 480), this indicates

$$x_1^2 + x_2^2 + \dots + x_n^2 = \|\mathbf{x}\|^2 = 1.$$

And we have also shown in Equation (15.4) that

$$\mathbf{A}^k \mathbf{v}^{(0)} = x_1 \lambda_1^k \mathbf{q}_1 + x_2 \lambda_2^k \mathbf{q}_2 + \dots + x_n \lambda_n^k \mathbf{q}_n.$$

The above findings imply that

$$\begin{aligned}
s_k^2 &= 1 - \left( \frac{\mathbf{q}_1^\top \mathbf{A}^k \mathbf{v}^{(0)}}{\|\mathbf{A}^k \mathbf{v}^{(0)}\|} \right)^2 = 1 - \left( \frac{x_1^2 \lambda_1^{2k}}{\sum_{i=1}^n x_i^2 \lambda_i^{2k}} \right) \\
&= \frac{\sum_{i=2}^n x_i^2 \lambda_i^{2k}}{\sum_{i=1}^n x_i^2 \lambda_i^{2k}} \leq \frac{\sum_{i=2}^n x_i^2 \lambda_i^{2k}}{x_1^2 \lambda_1^{2k}} \\
&= \frac{1}{x_1^2} \sum_{i=2}^n x_i^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} \leq \frac{1}{x_1^2} \left( \sum_{i=2}^n x_i^2 \right) \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} \\
&= \frac{1 - x_1^2}{x_1^2} \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} = t_0^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k},
\end{aligned} \tag{15.5}$$

where the last equality follows from the definition of  $c_k$ , when  $k = 0$ , we have  $c_0 = |\mathbf{q}_1^\top \mathbf{v}^{(0)}| = |x_1| \neq 0$ . Therefore, the first result follows:

$$|s_k| \leq t_0 \left| \frac{\lambda_2}{\lambda_1} \right|^k.$$

The above result is exactly what have done in (Golub and Van Loan, 2013), however, the blue text above is a loose bound since  $0 < x_1^2 \leq 1$  under our assumption. An improved result is shown as follows (where the difference is made into blue text):

$$\begin{aligned}
s_k^2 &= 1 - \left( \frac{\mathbf{q}_1^\top \mathbf{A}^k \mathbf{v}^{(0)}}{\|\mathbf{A}^k \mathbf{v}^{(0)}\|} \right)^2 = 1 - \left( \frac{x_1^2 \lambda_1^{2k}}{\sum_{i=1}^n x_i^2 \lambda_i^{2k}} \right) \\
&= \frac{\sum_{i=2}^n x_i^2 \lambda_i^{2k}}{\sum_{i=1}^n x_i^2 \lambda_i^{2k}} \leq \frac{\sum_{i=2}^n x_i^2 \lambda_i^{2k}}{\sum_{i=1}^n x_i^2 \lambda_1^{2k}} = \frac{\sum_{i=2}^n x_i^2 \lambda_i^{2k}}{\lambda_1^{2k}} \\
&= \sum_{i=2}^n x_i^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} \leq \left( \sum_{i=2}^n x_i^2 \right) \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} \\
&= (1 - x_1^2) \left( \frac{\lambda_i}{\lambda_1} \right)^{2k} = s_0^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k}.
\end{aligned} \tag{15.6}$$

Therefore, the result follows that

$$|s_k| \leq s_0 \left| \frac{\lambda_2}{\lambda_1} \right|^k.$$

Further, since  $\mathbf{v}^{(k)} = \frac{\mathbf{A}^k \mathbf{v}^{(0)}}{\|\mathbf{A}^k \mathbf{v}^{(0)}\|}$  from Equation (15.4), we have

$$\begin{aligned}
\lambda^{(k)} &= \mathbf{v}^{(k)\top} \mathbf{A} \mathbf{v}^{(k)} = \left( \frac{\mathbf{A}^k \mathbf{v}^{(0)}}{\|\mathbf{A}^k \mathbf{v}^{(0)}\|} \right)^\top \mathbf{A} \left( \frac{\mathbf{A}^k \mathbf{v}^{(0)}}{\|\mathbf{A}^k \mathbf{v}^{(0)}\|} \right) \\
&= \frac{\mathbf{v}^{(0)\top} \mathbf{A}^{2k+1} \mathbf{v}^{(0)}}{\mathbf{v}^{(0)\top} \mathbf{A}^{2k} \mathbf{v}^{(0)}} = \frac{\sum_{i=1}^n x_i^2 \lambda_i^{2k+1}}{\sum_{i=1}^n x_i^2 \lambda_i^{2k}}.
\end{aligned}$$

Therefore,

$$\begin{aligned} |\lambda^{(k)} - \lambda_1| &= \left| \frac{\sum_{i=2}^n x_i^2 \lambda_i^{2k} (\lambda_i - \lambda_1)}{\sum_{i=1}^n x_i^2 \lambda_i^{2k}} \right| \leq \max_{2 \leq i \leq n} |\lambda_1 - \lambda_i| \left( \frac{\sum_{i=2}^n x_i^2 \lambda_i^{2k}}{\sum_{i=1}^n x_i^2 \lambda_i^{2k}} \right) \\ &\leq \max_{2 \leq i \leq n} |\lambda_1 - \lambda_i| \cdot s_0^2 \left( \frac{\lambda_i}{\lambda_1} \right)^{2k}, \end{aligned}$$

where the last inequality comes from Equation (15.5) or Equation (15.6) shown above. ■

From the above deduction, we conclude the following convergence result:

**Theorem 15.3: (Convergence of Power Iteration)**

(Under the same condition as Theorem 15.1) Suppose  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric. Suppose further that the following two conditions are met

- $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0$ , i.e.,  $\lambda_1$  has algebraic multiplicity being equal to 1;
- $\mathbf{q}_1^\top \mathbf{v}^{(0)} \neq 0$ , i.e., the initial guess  $\mathbf{v}^{(0)}$  has a component in the direction of the eigenvector  $\mathbf{q}_1$  associated to the eigenvalue  $\lambda_1$ .

Then the iterates of Algorithm 38 satisfy

$$\|\mathbf{v}^{(k)} - (\pm \mathbf{q}_1)\| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right), \quad |\lambda^{(k)} - \lambda_1| = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)$$

as  $k \rightarrow \infty$ .

The  $\pm$  sign of  $\mathbf{q}_1$  means that, one of  $\mathbf{q}_1$  and  $-\mathbf{q}_1$  is to be taken into the result. And this shows  $\mathbf{v}^{(k)}$  converges to  $\pm \mathbf{q}_1$  linearly in Algorithm 38.

**Eigenvalue Assumptions** The assumption  $\mathbf{q}_1^\top \mathbf{v}^{(0)} \neq 0$  in the above algorithm is to assume the initial guess of the eigenvector has a component in the direction of the eigenvector  $\mathbf{q}_1$  we want to find. Otherwise, it will not converge to the  $\pm \mathbf{q}_1$ .

We also carefully notice that, the equality sign does not appear in the assumption  $|\lambda_1| > |\lambda_2|$ , in which case  $\lambda_1$  is also known as the *dominant* eigenvalue of matrix  $\mathbf{A}$ . Nevertheless, when  $\{|\lambda_1| = |\lambda_2| > |\lambda_3| \geq |\lambda_4| \geq \dots, \mathbf{q}_1^\top \mathbf{v}^{(0)} \neq 0, \text{ and } \mathbf{q}_2^\top \mathbf{v}^{(0)} \neq 0\}$ ,  $\mathbf{v}^{(k)}$  will converge to a multiple of  $x_1 \mathbf{q}_1 \pm x_2 \mathbf{q}_2$ , i.e., lies in the subspace spanned by  $\{\mathbf{q}_1, \mathbf{q}_2\}$ . To see this,

1.  $\lambda_1 = \lambda_2$ , i.e., the two dominant eigenvalues are coincident. Equation (15.4) shows  $\mathbf{v}^{(k)}, \lambda^{(k)}$  converges to

$$\begin{cases} \mathbf{v}^{(k)} \xrightarrow{k \rightarrow \infty} \boldsymbol{\beta}_1 = \frac{x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2}{\|x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2\|} \in \text{span}\{\mathbf{q}_1, \mathbf{q}_2\} & \text{i.e., } \mathbf{A}\boldsymbol{\beta}_1 = \lambda_1 \boldsymbol{\beta}_1, \\ \lambda^{(k)} \xrightarrow{k \rightarrow \infty} \boldsymbol{\beta}_1^\top \mathbf{A} \boldsymbol{\beta}_1 = \lambda_1. \end{cases}$$

Therefore, the vector sequence  $\mathbf{v}^{(k)}$  still converges to an eigenvector of  $\mathbf{A}$ , which lies in the space spanned by  $\{\mathbf{q}_1, \mathbf{q}_2\}$ , and the scalar sequence  $\lambda^{(k)}$  still converges to  $\lambda_1 = \lambda_2$ . We carefully notice that when  $\lambda_1, \lambda_2$ , any vector in  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2\}$  will be an eigenvector of  $\mathbf{A}$ .

2.  $\lambda_1 = -\lambda_2$ , i.e., the two dominant eigenvalues are opposite. Equation (15.4) shows  $\mathbf{v}^{(k)}, \lambda^{(k)}$  converges to

$$\mathbf{v}^{(k)} \xrightarrow{k \rightarrow \infty} \boldsymbol{\beta}_2 = \frac{\lambda_1^k x_1 \mathbf{q}_1 + \lambda_2^k x_2 \mathbf{q}_2}{\|\lambda_1^k x_1 \mathbf{q}_1 + \lambda_2^k x_2 \mathbf{q}_2\|} \in \text{span}\{\mathbf{q}_1, \mathbf{q}_2\}.$$

Then, we have

$$\begin{cases} \mathbf{A}\boldsymbol{\beta}_2 = \lambda_1 \frac{x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2}{\|x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2\|}, & \text{when } k \text{ is odd;} \\ \mathbf{A}\boldsymbol{\beta}_2 = \lambda_1 \frac{x_1 \mathbf{q}_1 - x_2 \mathbf{q}_2}{\|x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2\|}, & \text{when } k \text{ is even.} \end{cases}$$

Therefore,  $\mathbf{v}^{(k)}$  will not converge to an eigenvector of  $\mathbf{A}$ , neither  $\lambda^{(k)}$  do. In this case, we observe that  $\mathbf{Ax} = \lambda x \rightarrow \mathbf{A}^2 x = \lambda^2 x$  such that the eigenvalues of  $\mathbf{A}^2$  will be nonnegative and  $\lambda_1^2 = \lambda_2^2$  are the same eigenvalue of  $\mathbf{A}^2$  if they are eigenvalues of  $\mathbf{A}$ . Then apply the power method on  $\mathbf{A}^2$  will lead to the convergence to the eigenvector and eigenvalue of  $\mathbf{A}^2$  which is the same as analyzed in case (1).

3.  $\lambda_1 = \bar{\lambda}_2$ , i.e., the two dominant eigenvalues are complex conjugate. The power method is not convergent (Wilkinson, 1971; Quarteroni et al., 2010) and we shall not give the details.

**What if  $\mathbf{q}_1^\top \mathbf{v}^{(0)} = 0$ ?** All the convergence results are made under the assumption that  $\mathbf{q}_1^\top \mathbf{v}^{(0)} \neq 0$ . But since we do not know  $\mathbf{q}_1$  up front, the requirement is not able to fulfill. When it happens that  $\mathbf{q}_1^\top \mathbf{v}^{(0)} = 0$ , from Equation (15.4) again, the vector sequence  $\mathbf{v}^{(k)}$  converges to  $\mathbf{q}_2$  such that  $\lambda^{(k)} \rightarrow \lambda_2$ . Therefore, the requirement on the initial guess will not harm the convergence of the power method, but the net result will be different.

**What do we start from the “theoretical” one?** In the theoretical power method Algorithm 37, we assume the  $\lambda_1$  is known, and shows the vector sequence converges linearly to the eigenvector in Equation (15.3). This result matches the convergence of the “practical” power method Algorithm 38 as shown in Theorem 15.3. The “theoretical” one thus can be employed to find a first analysis on the power method, and we shall shortly find its counterpart in the *inverse power method* in the next section.

### 15.5.2 The Inverse Power Method

The *power method* homes in on an eigenvector associated with the largest eigenvalue (in magnitude). Nevertheless, the *inverse power method* homes in on an eigenvector associated with the smallest eigenvalue (in magnitude). To see this, we first provide the lemma about the eigenpair of the inverse of matrices as follows.

#### Lemma 15.4: (Eigenpair of Inverse Matrix)

Suppose matrix  $\mathbf{A}^{n \times n}$  is nonsingular, and  $(\lambda, \mathbf{x})$  is an eigenpair of  $\mathbf{A}$ . Then  $(1/\lambda, \mathbf{x})$  is an eigenpair of  $\mathbf{A}^{-1}$ .

Furthermore, we assume previously that the matrix  $\mathbf{A}$  is real and symmetric. We have shown in Section 13 that the rank of the real and symmetric matrix is the number of nonzero eigenvalues. The inverse power method involves the inverse of the eigenvalues such that we suppose further the matrix involved in this section is nonsingular, i.e., all the eigenvalues are nonzero and the matrix is invertible.

---

**Algorithm 39** Inverse Power Iteration (Compare to Algorithm 38)

---

**Require:** nonsingular matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

- 1:  $\mathbf{v}^{(0)}$  = some vector with  $\|\mathbf{v}^{(0)}\| = 1$ ;
  - 2: **for**  $k = 1, 2, \dots$  **do**
  - 3:    $\mathbf{w} = \mathbf{A}^{-1}\mathbf{v}^{(k-1)}$ ;
  - 4:    $\mathbf{v}^{(k)} = \mathbf{w}/\|\mathbf{w}\|$ ;
  - 5:    $\lambda^{(k)} = (\mathbf{v}^{(k)})^\top \mathbf{A} \mathbf{v}^{(k)}$ ; ▷ i.e., Rayleigh quotient
  - 6: **end for**
- 

The idea behind the algorithm is that an eigenvector associated with the smallest eigenvalue (in magnitude) of  $\mathbf{A}$  is an eigenvector associated with the largest eigenvalue (in magnitude) of  $\mathbf{A}^{-1}$  by Lemma 15.4. From the power method, it is trivial that the sequence  $\mathbf{v}^{(k)}$  in Algorithm 39 converges linearly to the eigenvector of  $\mathbf{A}$  corresponding to the smallest eigenvalue  $\lambda_n$  (or converges linearly to the eigenvector of  $\mathbf{A}^{-1}$  corresponding to the largest eigenvalue  $1/\lambda_n$ ).

To abuse the analysis of the convergence of the inverse power method, we can carefully employ a theoretical (but impossible) way as shown in Algorithm 37 where we assume  $\lambda_1$  is known and the convergence result is shown in Equation (15.3). Come back to the inverse power method, suppose  $\lambda_n$  is known and a “theoretical” inverse power method is induced, the convergence is thus given by (analogous to Equation (15.3)):

$$\begin{aligned} \|\mathbf{v}^{(k+1)} - x_n \mathbf{q}_n\|_{\mathbf{Q}^{-1}} &= \|\mathbf{Q}^{-1}(\mathbf{v}^{(k+1)} - x_n \mathbf{q}_n)\|_2 \\ &\leq \left| \frac{\lambda_n}{\lambda_{n-1}} \right| \cdot \|\mathbf{Q}^{-1}(\mathbf{v}^{(k)} - x_n \mathbf{q}_n)\|_2 = \left| \frac{\lambda_n}{\lambda_{n-1}} \right| \cdot \|\mathbf{v}^{(k)} - x_n \mathbf{q}_n\|_{\mathbf{Q}^{-1}}, \end{aligned} \quad (15.7)$$

where we use the fact that if the spectral decomposition of  $\mathbf{A}$  is  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{A}^{-1}$ , then the spectral decomposition of  $\mathbf{A}^{-1}$  is  $\mathbf{A}^{-1} = \mathbf{Q}\Lambda^{-1}\mathbf{Q}^{-1}$ . Since  $\lambda_n, \lambda_{n-1}$  are the smallest two eigenvalues of  $\mathbf{A}$  (in magnitude). It is possible that the two eigenvalues are very close to each other and thus the bound  $\left| \frac{\lambda_n}{\lambda_{n-1}} \right|$  is close to 1.

### 15.5.3 The Shifted Inverse Power Method

Now we suppose that we know that one of the eigenvalues of  $\mathbf{A}$  is close to a value  $\mu \in \mathbb{R}$ . For any value  $\mu$  that is not an eigenvalue of  $\mathbf{A}$ ,  $\mathbf{A} - \mu\mathbf{I}$  is nonsingular even if  $\mathbf{A}$  is singular. The matrix  $\mathbf{A} - \mu\mathbf{I}$  is referred to as the matrix  $\mathbf{A}$  that has been “shifted” by  $\mu$ , and  $\mu$  is called a *shift*.

The following lemma reveals that a shifted version of the inverse power method can be employed to find the eigenvector associated with the eigenvalue that is closest to  $\mu$ .

**Lemma 15.5: (Eigenpair of Shifted Matrix)**

Suppose  $(\lambda, \mathbf{x})$  is an eigenpair of  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $\mu \in \mathbb{R}$  is not an eigenvalue of  $\mathbf{A}$ . Then  $(\lambda - \mu, \mathbf{x})$  is an eigenpair of  $\mathbf{A} - \mu\mathbf{I}$ .

Notice that the eigenvectors of  $\mathbf{A} - \mu\mathbf{I}$  are the same as those of  $\mathbf{A}$  since  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x} \rightarrow (\mathbf{A} - \mu\mathbf{I})\mathbf{x} = (\lambda - \mu)\mathbf{x}$ .

**Algorithm 40** Shifted Inverse Power Iteration (Compare to Algorithm 39)

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric,  $\mu$  is not an eigenvalue of  $\mathbf{A}$ ;

- 1:  $\mathbf{v}^{(0)}$  = some vector with  $\|\mathbf{v}^{(0)}\| = 1$ ;
- 2: **for**  $k = 1, 2, \dots$  **do**
- 3:    $\mathbf{w} = (\mathbf{A} - \mu\mathbf{I})^{-1}\mathbf{v}^{(k-1)}$ ;
- 4:    $\mathbf{v}^{(k)} = \mathbf{w}/\|\mathbf{w}\|$ ;
- 5:    $\lambda^{(k)} = (\mathbf{v}^{(k)})^\top \mathbf{A} \mathbf{v}^{(k)}$ ; ▷ i.e., Rayleigh quotient
- 6: **end for**

The procedure is again formulated in Algorithm 40. To abuse the analysis of convergence again, suppose  $\mu$  is closest to the smallest eigenvalue  $\lambda_n$  in magnitude, the convergence is (again, analogous to Equation (15.3)):

$$\begin{aligned} \|\mathbf{v}^{(k+1)} - x_n \mathbf{q}_n\|_{\mathbf{Q}^{-1}} &= \|\mathbf{Q}^{-1}(\mathbf{v}^{(k+1)} - x_n \mathbf{q}_n)\|_2 \\ &\leq \left| \frac{\lambda_n - \mu}{\lambda_{n-1} - \mu} \right| \cdot \|\mathbf{Q}^{-1}(\mathbf{v}^{(k)} - x_n \mathbf{q}_n)\|_2 = \left| \frac{\lambda_n - \mu}{\lambda_{n-1} - \mu} \right| \cdot \|\mathbf{v}^{(k)} - x_n \mathbf{q}_n\|_{\mathbf{Q}^{-1}}. \end{aligned} \quad (15.8)$$

When  $\mu$  is close to  $\lambda_n$ , the bound  $\left| \frac{\lambda_n - \mu}{\lambda_{n-1} - \mu} \right|$  is small such that the convergence is faster than the (naive) inverse power method (see Equation (15.7)), although it is still linearly convergent.

The formal convergence results is shown in the following theorem (one can follow the deduction as in Theorem 15.3 to obtain the result).

**Theorem 15.6: (Convergence of Shifted Inverse Power Iteration)**

Suppose  $\lambda_J$  is the closest eigenvalue to  $\mu$  and  $\lambda_K$  is the second closest. Moreover,  $\mathbf{q}_J^\top \mathbf{v}^{(0)} \neq 0$ . Then the iterates of Algorithm 40 satisfy

$$\|\mathbf{v}^{(k)} - (\pm \mathbf{q}_J)\| = O\left(\left| \frac{\lambda_J - \mu}{\lambda_K - \mu} \right|^k\right), \quad |\lambda^{(k)} - \lambda_J| = O\left(\left| \frac{\lambda_J - \mu}{\lambda_K - \mu} \right|^{2k}\right)$$

as  $k \rightarrow \infty$ .

And this shows  $\mathbf{v}^{(k)}$  converges to  $\pm \mathbf{q}_J$  linearly for Algorithm 40.

### 15.5.4 The Rayleigh Quotient Method

We know that the *inverse power iteration* converges to the eigenvector corresponding to the smallest eigenvalue of  $\mathbf{A}$ , and the *shifted inverse power iteration* converges to the eigenvector corresponding to the eigenvalue closest to  $\mu$  with possible faster convergence. Fortunately, both of the two methods are *inverse power iteration* in some sense. If we can combine the ideas behind the two algorithms, i.e., use the Rayleigh quotient of the estimated eigenvector in each iteration as the estimate of the eigenvalue, we can get a faster algorithm.

---

#### Algorithm 41 Rayleigh Quotient Iteration (Compare to Algorithm 40)

---

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

- 1:  $\mathbf{v}^{(0)}$  = some vector with  $\|\mathbf{v}^{(0)}\| = 1$ ;
  - 2:  $\lambda^{(0)} = (\mathbf{v}^{(0)})^\top \mathbf{A} \mathbf{v}^{(0)}$ ; ▷ i.e., Rayleigh quotient
  - 3: **for**  $k = 1, 2, \dots$  **do**
  - 4:    $\mathbf{w} = (\mathbf{A} - \lambda^{(k-1)} \mathbf{I})^{-1} \mathbf{v}^{(k-1)}$ ;
  - 5:    $\mathbf{v}^{(k)} = \mathbf{w} / \|\mathbf{w}\|$ ;
  - 6:    $\lambda^{(k)} = (\mathbf{v}^{(k)})^\top \mathbf{A} \mathbf{v}^{(k)}$ ; ▷ i.e., Rayleigh quotient
  - 7: **end for**
- 

The Rayleigh quotient iteration finds an eigenvector, but with which eigenvalue it is associated is not clear from the start.

## 15.6. QR Algorithm

The QR algorithm for computing the eigenvalues and eigenvectors of matrices has been named as one of the ten most important algorithms of the twentieth century ([Dongarra and Sullivan, 2000](#); [Cipra, 2000](#)) which is published by John G. F. Francis in the work of ([Francis, 1961, 1962](#)) that is quoted as one of the jewels of numerical analysis ([Trefethen and Bau III, 1997](#)). We will introduce the QR algorithm from the most simple case to a shifted version with implicit calculation.

The QR algorithm goes further by *simultaneously* calculating the eigenvalues of a given matrix  $\mathbf{A}$ . The central idea is to reduce matrix  $\mathbf{A}$  by a set of *similarity transformations* (Definition 8.1, p. 198) from which the eigenvalues are kept unmodified (Lemma 8.2, p. 198) and in the meantime the eigenvalue is somewhat easier to calculate. The net result turns to be simple to handle, all we do in the QR algorithm is take a QR decomposition, multiply the computed factors  $\mathbf{Q}$  and  $\mathbf{R}$  together in the reverse order  $\mathbf{R}\mathbf{Q}$ , and repeat the procedure. Hence the name *QR algorithm*.

### 15.6.1 Preliminary: Power Iteration with Eigenvector Known

We firstly show how it evolves from the power iteration algorithm. Now, let's put a subscript for all the  $\mathbf{v}^{(k)}$ 's to emphasize it converges to the eigenvector associated with  $\lambda_i$ : e.g.,  $\mathbf{v}_i^{(k)}$  will be proven to converge to  $\lambda_i$ . And suppose further that we know the normalized eigenvector  $\mathbf{q}_1$  associated with  $\lambda_1$  up front. A second initial vector,  $\mathbf{v}_2^{(0)}$ , does not have a component in the direction of  $\mathbf{q}_1$ . This can be met if it is orthogonal to  $\mathbf{q}_1$  which is constructed by

projection introduced in the Gram-Schmidt process (Section 3.2, p. 82):

$$\mathbf{v}_2^{(k+1)} \leftarrow \mathbf{v}_2^{(k)} - \mathbf{q}_1^\top \mathbf{v}_2^{(k)} \mathbf{q}_1$$

such that  $\mathbf{q}_1^\top \mathbf{v}_2^{(k+1)} = 0$  (recall  $\mathbf{q}_1^\top \mathbf{v}_2^{(k)} \mathbf{q}_1$  above is the component of  $\mathbf{v}_2^{(k)}$  in the direction of  $\mathbf{q}_1$  since  $\mathbf{q}_1$  has unit length). With  $\mathbf{q}_1$  known beforehand, we consider the method in Algorithm 42.

---

**Algorithm 42** Power Iteration ( $\mathbf{q}_1$  is Known Up Front)

---

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

- 1:  $\mathbf{q}_1$  is known up front;
  - 2:  $\mathbf{v}_2^{(0)} =$  some vector in  $\mathbb{R}^n$ ;
  - 3:  $\mathbf{v}_2^{(0)} = \mathbf{v}_2^{(0)} - \mathbf{q}_1^\top \mathbf{v}_2^{(0)} \mathbf{q}_1$ ; ▷ i.e., project along  $\mathbf{q}_1$ :  $\mathbf{q}_1^\top \mathbf{v}_2^{(0)} = 0$
  - 4:  $\mathbf{v}_2^{(0)} = \mathbf{v}_2^{(0)} / \|\mathbf{v}_2^{(0)}\|$ ; ▷ normalize to have length one
  - 5: **for**  $k = 1, 2, \dots$  **do**
  - 6:    $\mathbf{v}_2^{(k)} = \mathbf{A}\mathbf{v}_2^{(k-1)}$ ;
  - 7:    $\mathbf{v}_2^{(k)} = \mathbf{v}_2^{(k)} - \mathbf{q}_1^\top \mathbf{v}_2^{(k)} \mathbf{q}_1$ ; ▷ make sure  $\mathbf{v}_2^{(k)}$  is orthogonal to  $\mathbf{q}_1$
  - 8:    $\mathbf{v}_2^{(k)} = \mathbf{v}_2^{(k)} / \|\mathbf{v}_2^{(k)}\|$ ;
  - 9:    $\lambda_2^{(k)} = (\mathbf{v}_2^{(k)})^\top \mathbf{A}\mathbf{v}_2^{(k)}$ ; ▷ i.e., Rayleigh quotient
  - 10: **end for**
- 

Write again  $\mathbf{v}_2^{(0)}$  as a linear combination of the orthonormal eigenvectors  $\mathbf{q}_i$ :

$$\mathbf{v}_2^{(0)} = x_1 \mathbf{q}_1 + x_2 \mathbf{q}_2 + \dots + x_n \mathbf{q}_n. \quad (15.9)$$

Since in step 3 of above algorithm, we project along the  $\mathbf{v}_2^{(0)}$  along  $\mathbf{q}_1$ , then the component  $x_1$  of Equation (15.9) is equal to 0. Similarly,  $\mathbf{v}_2^{(k)}$  is a multiple of  $\mathbf{A}^k \mathbf{v}_2^{(0)}$  such that  $\mathbf{v}_2^{(k)} = c_{2k} \mathbf{A}^k \mathbf{v}_2^{(0)} = \frac{\mathbf{A}^k \mathbf{v}_2^{(0)}}{\|\mathbf{A}^k \mathbf{v}_2^{(0)}\|}$ . We have

$$\begin{aligned} \mathbf{v}_2^{(k)} &= c_{2k} \mathbf{A}^k \mathbf{v}_2^{(0)} \\ &= c_{2k} (x_1 \lambda_1^k \mathbf{q}_1 + x_2 \lambda_2^k \mathbf{q}_2 + x_3 \lambda_3^k \mathbf{q}_3 + \dots + x_n \lambda_n^k \mathbf{q}_n) \\ &= c_{2k} (x_2 \lambda_2^k \mathbf{q}_2 + x_3 \lambda_3^k \mathbf{q}_3 + \dots + x_n \lambda_n^k \mathbf{q}_n) \\ &= c_{2k} \lambda_2^k \left( x_2 \mathbf{q}_2 + x_3 \left( \frac{\lambda_3}{\lambda_2} \right)^k \mathbf{q}_3 + \dots + x_n \left( \frac{\lambda_n}{\lambda_2} \right)^k \mathbf{q}_n \right). \end{aligned}$$

Therefore, following from Theorem 15.3, if we assume  $|\lambda_2| > |\lambda_3| \geq |\lambda_3| \geq \dots \geq |\lambda_n| \geq 0$  and  $\mathbf{q}_2^\top \mathbf{v}_2^{(0)} \neq 0$ ,  $\mathbf{v}_2^{(k)}$  will converge linearly to  $\pm \mathbf{q}_2$ , and so  $\lambda_2^{(k)}$  will converge to  $\lambda_2$  by Algorithm 42.

Analogously, when  $\{|\lambda_2| = |\lambda_3| > |\lambda_4| \geq |\lambda_5| \dots, \mathbf{q}_1^\top \mathbf{v}^{(0)} \neq 0, \text{ and } \mathbf{q}_2^\top \mathbf{v}^{(0)} \neq 0\}$ ,  $\mathbf{v}^{(k)}$  will converge to a multiple of  $x_2 \mathbf{q}_2 \pm x_3 \mathbf{q}_3$ , i.e., lies in the space spanned by  $\{\mathbf{q}_2, \mathbf{q}_3\}$ .

### 15.6.2 Preliminary: Power Iteration with Eigenvector Unknown

However, the method presented in Algorithm 42 is not practical since we usually do not know  $\mathbf{q}_1$  up front. But since we have a trivial power iteration that can compute the  $\mathbf{v}_1^{(0)}$  converging to  $\pm \mathbf{q}_1$  (Algorithm 38, p. 297), a simultaneous algorithms can be constructed. In lieu of project  $\mathbf{v}_2^{(k)}$  along  $\mathbf{q}_1$ , one can project it along  $\mathbf{v}_1^{(k)}$  instead. Therefore, a method to find both  $\mathbf{q}_1, \mathbf{q}_2$  simultaneously can be constructed in Algorithm 43.

---

**Algorithm 43** Power Iteration ( $\mathbf{q}_1$  is Unknown Up Front, Compare to Algorithm 42)

---

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

```

1: $\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}$ = two random vectors in \mathbb{R}^n ;
2: $\mathbf{v}_2^{(0)} = \mathbf{v}_2^{(0)} - \mathbf{v}_1^{(0)\top} \mathbf{v}_2^{(0)} \mathbf{v}_1^{(0)}$; ▷ i.e., project along $\mathbf{v}_1^{(0)}$: $\mathbf{v}_1^{(0)\top} \mathbf{v}_2^{(0)} = 0$
3: $\mathbf{v}_1^{(0)} = \mathbf{v}_1^{(0)} / \|\mathbf{v}_1^{(0)}\|$; ▷ normalize to have length one
4: $\mathbf{v}_2^{(0)} = \mathbf{v}_2^{(0)} / \|\mathbf{v}_2^{(0)}\|$; ▷ normalize to have length one
5: for $k = 1, 2, \dots$ do
6: // update $\mathbf{v}_1^{(k)}$
7: $\mathbf{v}_1^{(k)} = \mathbf{A}\mathbf{v}_1^{(k-1)}$;
8: $\mathbf{v}_1^{(k)} = \mathbf{v}_1^{(k)} / \|\mathbf{v}_1^{(k)}\|$;
9: // update $\mathbf{v}_2^{(k)}$
10: $\mathbf{v}_2^{(k)} = \mathbf{A}\mathbf{v}_2^{(k-1)}$;
11: $\mathbf{v}_2^{(k)} = \mathbf{v}_2^{(k)} - \mathbf{v}_1^{(k)\top} \mathbf{v}_2^{(k)} \mathbf{v}_1^{(k)}$; ▷ make sure $\mathbf{v}_2^{(k)}$ is orthogonal to $\mathbf{v}_1^{(k)}$
12: $\mathbf{v}_2^{(k)} = \mathbf{v}_2^{(k)} / \|\mathbf{v}_2^{(k)}\|$;
13: // compute the corresponding eigenvalues
14: $\lambda_1^{(k)} = (\mathbf{v}_1^{(k)})^\top \mathbf{A} \mathbf{v}_1^{(k)}$; ▷ i.e., Rayleigh quotient
15: $\lambda_2^{(k)} = (\mathbf{v}_2^{(k)})^\top \mathbf{A} \mathbf{v}_2^{(k)}$; ▷ i.e., Rayleigh quotient
16: end for

```

---

Combine the findings above, the iterates of Algorithm 43 satisfies that (when  $\mathbf{q}_1^\top \mathbf{v}_1^{(0)} \neq 0$  and  $\mathbf{q}_2^\top \mathbf{v}_2^{(0)} \neq 0$ )

- If  $|\lambda_1| > |\lambda_2|$ , the vector sequence  $\mathbf{v}_1^{(k)}$  will converge linearly to  $\pm \mathbf{q}_1$  at a rate of  $|\frac{\lambda_2}{\lambda_1}|$ ;
- If  $|\lambda_1| > |\lambda_2| > |\lambda_3|$ , the vector sequence  $\mathbf{v}_2^{(k)}$  will converge linearly to  $\pm \mathbf{q}_2$  at a rate of  $|\frac{\lambda_3}{\lambda_2}|$ ;
- If  $|\lambda_1| = |\lambda_2| > |\lambda_3|$ , the vector sequence  $\mathbf{v}_1^{(k)}$  will converge to a multiple of  $x_1 \mathbf{q}_1 \pm x_2 \mathbf{q}_2$ , and the vector sequence  $\mathbf{v}_2^{(k)}$  will converge linearly to  $\pm \mathbf{q}_2$ . I.e., the span of  $\{\mathbf{q}_1, \mathbf{q}_2\}$  can be approximated by the span of  $\{\mathbf{v}_1^{(k)}, \mathbf{v}_2^{(k)}\}$ .

### 15.6.3 Preliminary: Power Iteration with Eigenvector Unknown and QR Decomposition

We carefully notice that step 2 to step 4 Algorithm 43 is equivalent to apply a QR decomposition on a  $\mathbb{R}^{n \times 2}$  matrix  $[\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}]$ , that is

$$\underbrace{[\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}]}_{\widehat{\mathbf{V}}^{(0)}}, \mathbf{R} \leftarrow QR([\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)})],$$

where  $QR(\mathbf{A})$  is defined as the function to obtain the QR decomposition of matrix  $\mathbf{A}$ . Moreover, step 6 to step 12 of Algorithm 43 can also be rephrased as an QR decomposition on the  $n \times 2$  matrix  $\mathbf{A}[\mathbf{v}_1^{(k-1)}, \mathbf{v}_2^{(k-1)}]$ . A further simplification on the form is to obtain the eigenvalues  $\lambda_1^{(k)}, \lambda_2^{(k)}$  as follows:

$$\underbrace{\begin{bmatrix} \lambda_1^{(k)} & 0 \\ 0 & \lambda_2^{(k)} \end{bmatrix}}_{\mathbf{A}^{(k)}} = \underbrace{\begin{bmatrix} \mathbf{v}_1^{(k)} & \mathbf{v}_2^{(k)} \end{bmatrix}^\top}_{\widehat{\mathbf{V}}^{(k)\top}} \mathbf{A} \underbrace{\begin{bmatrix} \mathbf{v}_1^{(k)} & \mathbf{v}_2^{(k)} \end{bmatrix}}_{\widehat{\mathbf{V}}^{(k)}}$$

---

#### Algorithm 44 Power Iteration (On 2 Vectors, Equivalent to Algorithm 43)

---

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

- 1:  $\widehat{\mathbf{V}}^{(0)} = [\mathbf{v}_1^{(0)}, \mathbf{v}_2^{(0)}] \in \mathbb{R}^{n \times 2}$  = two random vectors in  $\mathbb{R}^n$ ;
  - 2:  $\widehat{\mathbf{V}}^{(0)}, \mathbf{R} = QR(\widehat{\mathbf{V}}^{(0)})$ ;
  - 3: **for**  $k = 1, 2, \dots$  **do**
  - 4:    $\widehat{\mathbf{V}}^{(k)}, \mathbf{R} = QR(\mathbf{A}\widehat{\mathbf{V}}^{(k-1)})$ ;
  - 5:    $\widehat{\mathbf{A}}^{(k)} = \widehat{\mathbf{V}}^{(k)\top} \mathbf{A} \widehat{\mathbf{V}}^{(k)}$ ; ▷ compute the corresponding eigenvalues
  - 6: **end for**
- 

Note that the widehat above the matrices in Algorithm 44 will be proven useful to differentiate the ones in the QR algorithms. With hindsight, it is natural to extent the algorithm not only for two vectors, but for  $p \leq n$  vectors. The full algorithm is formulated in Algorithm 45.

---

#### Algorithm 45 Power Iteration (On $p$ Vectors, Compare to Algorithm 44)

---

**Require:** matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  that is real and symmetric;

- 1:  $\widehat{\mathbf{V}}^{(0)}$  = random matrix in  $\mathbb{R}^{n \times p}$ ;
  - 2:  $\widehat{\mathbf{V}}^{(0)}, \mathbf{R} = QR(\widehat{\mathbf{V}}^{(0)})$ ;
  - 3: **for**  $k = 1, 2, \dots$  **do**
  - 4:    $\widehat{\mathbf{V}}^{(k)}, \mathbf{R} = QR(\mathbf{A}\widehat{\mathbf{V}}^{(k-1)})$ ;
  - 5:    $\widehat{\mathbf{A}}^{(k)} = \widehat{\mathbf{V}}^{(k)\top} \mathbf{A} \widehat{\mathbf{V}}^{(k)}$ ; ▷ compute the corresponding eigenvalues
  - 6: **end for**
- 

Again, we observe that, when  $\mathbf{q}_i^\top \mathbf{v}_i^{(0)} \neq 0$  for all  $i \in \{1, 2, \dots, p\}$ , <sup>1</sup> the iterates of Algorithm 45 satisfies that

<sup>1</sup>. That is, the initial guess  $\mathbf{v}_i^{(0)}$  is not orthogonal to eigenvector  $\mathbf{q}_i$ , and has a component in the direction of the eigenvector.

1. If  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_p| > |\lambda_{p+1}| \geq |\lambda_{p+2}| \geq \dots$ , then each column  $i$  of  $\widehat{\mathbf{V}}^{(k)}$  (i.e.,  $\mathbf{v}_i^{(k)}$ ) will converge linearly to  $\pm \mathbf{q}_i$  where the rate of removing from the component in the direction of  $\mathbf{q}_j$  is recorded as  $|\frac{\lambda_i}{\lambda_j}|$  ( $0 < i \leq p$  and  $p < j \leq n$ );
2. If some of the eigenvalues have equal magnitude, then the subspace spanned by the corresponding columns of  $\widehat{\mathbf{V}}^{(k)}$  will approximate the subspace spanned by the corresponding eigenvectors associated with those eigenvalues;
3. If  $p = n$ , and  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ , then we will find all the eigenvectors of  $\mathbf{A}$ .

#### 15.6.4 A Simple QR Algorithm from Power Iteration: without Shifts

So far, we have transferred power iteration into an algorithm finding all the eigenvectors (under mild conditions) which employs QR decomposition.

Now, let's consider the power iteration with slight modification in Algorithm 46 that cares the  $\mathbf{R}$  from the QR decomposition as a sequence of vectors, and  $\widehat{\mathbf{A}}^{(0)}$  initialized to be matrix  $\mathbf{A}$  (where we ignored the value indexed by 0 previously). Compare the two Algorithm 46 and 47.

---

#### Algorithm 46 Power Iteration

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric;

- 1: Same as Algorithm 45;
  - 2:  $\widehat{\mathbf{A}}^{(0)} = \mathbf{A}$ ;
  - 3:  $\widehat{\mathbf{V}}^{(0)} = \mathbf{I}_n$ ;  $\triangleright$  initial eigenvector guess
  - 4:  $\widehat{\mathbf{R}}^{(0)} = \mathbf{I}_n$ ;  $\triangleright$  compensate the sequence
  - 5: **for**  $k = 1, 2, \dots$  **do**
  - 6:    $\widehat{\mathbf{V}}^{(k)}, \widehat{\mathbf{R}}^{(k)} = QR(\mathbf{A}\widehat{\mathbf{V}}^{(k-1)})$ ;
  - 7:    $\widehat{\mathbf{A}}^{(k)} = \widehat{\mathbf{V}}^{(k)\top} \mathbf{A} \widehat{\mathbf{V}}^{(k)}$ ;
  - 8: **end for**
- 

---

#### Algorithm 47 Simple QR Algorithm

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric;

- 1:  $\mathbf{A}^{(0)} = \mathbf{A}$ ;
  - 2:  $\mathbf{V}^{(0)} = \mathbf{I}_n$ ;  $\triangleright$  initial eigenvector guess
  - 3:  $\mathbf{R}^{(0)} = \mathbf{I}_n$ ;  $\triangleright$  compensate the sequence
  - 4:  $\mathbf{Q}^{(0)} = \mathbf{I}_n$ ;  $\triangleright$  compensate the sequence
  - 5: **for**  $k = 1, 2, \dots$  **do**
  - 6:    $\mathbf{Q}^{(k)}, \mathbf{R}^{(k)} = QR(\mathbf{A}^{(k-1)})$ ;
  - 7:    $\mathbf{A}^{(k)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)}$ ;
  - 8:    $\mathbf{V}^{(k)} = \mathbf{V}^{(k-1)} \mathbf{Q}^{(k)}$ ;
  - 9: **end for**
- 

We firstly show the equivalence of the two algorithms by the following lemma.

**Lemma 15.1: (QR Algorithm from Power Iteration)**

We can show Algorithm 46 and Algorithm 47 are equivalent in the sense that for all iterates  $k \in \{0, 1, 2, \dots\}$ , we have

$$\begin{cases} \widehat{\mathbf{A}}^{(k)} = \mathbf{A}^{(k)}; & \text{(diagonals that will converge to eigenvalues)} \\ \widehat{\mathbf{R}}^{(k)} = \mathbf{R}^{(k)}; \\ \widehat{\mathbf{V}}^{(k)} = \mathbf{V}^{(k)}. & \text{(columns that will converge to eigenvectors)} \end{cases}$$

For clarity, the proof is delayed in Section 15.10. You might well ask: how could we find the reduction of Algorithm 46 to Algorithm 47. The answer is unglamorous! It was by trial and error.

**What's in the QR algorithm** All we do in Algorithm 47 is to take a QR decomposition, multiply the computed factors  $\mathbf{Q}$  and  $\mathbf{R}$  together in the reverse order  $\mathbf{R}\mathbf{Q}$ , and repeat. For convergence to diagonal form to be useful for finding eigenvalues, of course, the operations from  $\mathbf{A}^{(k-1)}$  to  $\mathbf{A}^{(k)}$  should be similarity transformations (Definition 8.1, p. 198). By Algorithm 47, it can be shown that

#### Simple QR Algorithm Property 1

$$(SQR\ 1) \quad \mathbf{A}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{A}^{(k-1)} \mathbf{Q}^{(k)}, \quad (15.10)$$

since  $\mathbf{A}^{(k)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)} = (\mathbf{Q}^{(k)\top} \mathbf{A}^{(k-1)}) \mathbf{Q}^{(k)}$ . Therefore,  $\mathbf{A}^{(k)}$  is a *similarity transformation*<sup>2</sup> of  $\mathbf{A}^{(k-1)}$ , which is also similar transformation from  $\mathbf{A}$ . This results in that  $\mathbf{A}^{(k)}$  and  $\mathbf{A}$  have *same eigenvalues, trace, and rank* (Lemma 8.2, p. 198). Going to the root of the induction in Equation (15.10), it follows that

#### Simple QR Algorithm Property 2

$$\begin{aligned} (SQR\ 2) \quad \mathbf{A}^{(k)} &= \mathbf{Q}^{(k)\top} \mathbf{A}^{(k-1)} \mathbf{Q}^{(k)} \\ &= \mathbf{Q}^{(k)\top} \left( \mathbf{Q}^{(k-1)\top} \mathbf{A}^{(k-2)} \mathbf{Q}^{(k-1)} \right) \mathbf{Q}^{(k)} \\ &= \dots \\ &= \underbrace{\mathbf{Q}^{(k)\top} \mathbf{Q}^{(k-1)\top} \dots \mathbf{Q}^{(0)\top}}_{\mathbf{V}^{(k)\top}} \mathbf{A} \underbrace{\mathbf{Q}^{(0)} \dots \mathbf{Q}^{(k-1)} \mathbf{Q}^{(k)}}_{\mathbf{V}^{(k)}} \\ &= \mathbf{V}^{(k)\top} \mathbf{A} \mathbf{V}^{(k)}. \end{aligned} \quad (15.11)$$

The above lemma tells that the simple QR algorithms is equivalent to the power iteration algorithms such that the diagonals of  $\mathbf{A}^{(k)}$  converge to the eigenvalues of  $\mathbf{A}$  (under mild conditions).  $\mathbf{A}^{(k)} = \mathbf{V}^{(k)\top} \mathbf{A} \mathbf{V}^{(k)}$  is not only a similarity transformation, but also an *orthogonal similarity transformation* since  $\mathbf{V}^{(k)}$  is orthogonal. This is particular important for the stability of the iterative method as the condition of  $\mathbf{A}^{(k)}$  is not worse than that of the original matrix  $\mathbf{A}$ . Then, the equivalence indicates that the  $i$ -th column of  $\mathbf{V}^{(k)}$  will converge linearly to the  $i$ -th eigenvector of  $\mathbf{A}$ , i.e.,  $\pm \mathbf{q}_i$ . A delve into the process, it shows by Algorithm 46 and 47 that

#### Simple QR Algorithm Property 3

$$\begin{aligned} \mathbf{V}^{(k)} &= \mathbf{V}^{(0)} \mathbf{Q}^{(1)} \mathbf{Q}^{(2)} \dots \mathbf{Q}^{(k)} \\ \mathbf{A}^k &= \mathbf{V}^{(k)} \mathbf{R}^{(k)} \mathbf{R}^{(k-1)} \dots \mathbf{R}^{(0)} \end{aligned} \left. \right\} \xrightarrow{\text{leads to}} \quad (15.12)$$

$$(SQR\ 3) \quad \mathbf{A}^k = \underbrace{\mathbf{Q}^{(0)} \mathbf{Q}^{(1)} \mathbf{Q}^{(2)} \dots \mathbf{Q}^{(k)}}_{=\mathbf{V}^{(k)}, \text{orthogonal}} \underbrace{\mathbf{R}^{(k)} \mathbf{R}^{(k-1)} \dots \mathbf{R}^{(0)}}_{=\mathbf{U}^{(k)}, \text{upper triangular}},$$

<sup>2</sup>. Here, when the nonsingular matrix is orthogonal, it is also known as *orthogonal similarity transformation*.

where the left hand side of the above equation is delayed to be proved in Section 15.10 and where  $\mathbf{Q}^{(0)} = \mathbf{V}^{(0)} = \mathbf{I}$  for simplicity. That is,  $\mathbf{A}^k$  can be expressed as a QR decomposition by  $\mathbf{A}^k = \mathbf{V}^{(k)}\mathbf{U}^{(k)}$ .<sup>3</sup>

The equivalence of the simultaneous power method and the simple QR algorithm (Algorithm 46 and 47) tells us a lot about convergence. Same as that in Section 15.6.3, it follows that

1. If  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_p| > |\lambda_{p+1}| \geq |\lambda_{p+2}| \geq \dots \geq |\lambda_n|$ , then each column  $i$  of  $\mathbf{V}^{(k)}$  (i.e.,  $\mathbf{v}_i^{(k)}$ ) will converge linearly to  $\pm \mathbf{q}_i$  where the rate of removing from the component in the direction of  $\mathbf{q}_j$  is recorded as  $|\frac{\lambda_i}{\lambda_j}|$  ( $0 < i \leq p$  and  $p < j \leq n$ );
2. If some of the eigenvalues have equal magnitude, then the subspace spanned by the corresponding columns of  $\mathbf{V}^{(k)}$  will approximate the subspace spanned by the corresponding eigenvectors associated with those eigenvalues;
3. If  $p = n$ , and  $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ , then we will find all the eigenvectors of  $\mathbf{A}$ . Specifically, suppose

$$\mathbf{A}^{(k)} = \begin{bmatrix} \lambda_1^{(k)} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{12} & \lambda_2^{(k)} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{1,n-1} & a_{2,n-1} & a_{3,n-1} & \ddots & a_{n-1,n} \\ a_{1n} & a_{2n} & a_{3n} & \dots & \lambda_n^{(k)} \end{bmatrix},$$

which is symmetric, and the convergence rate is

$$|a_{i,i-1}| = O\left(\left|\frac{\lambda_i}{\lambda_{i-1}}\right|^k\right).$$

Under different conditions, say,  $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_p| > |\lambda_{p+1}| > \dots > |\lambda_n|$ , then

1. The span of the first  $p$  columns of  $\mathbf{V}^{(k)}$  will converge to the subspace spanned by the first  $p$  orthonormal eigenvectors  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p\}$ ;
2. The span of the last  $n - p$  columns of  $\mathbf{V}^{(k)}$  will converge to the subspace spanned by the last  $n - p$  orthonormal eigenvectors  $\text{span}\{\mathbf{q}_{p+1}, \mathbf{q}_{p+2}, \dots, \mathbf{q}_n\}$ ;
3. Since  $\mathbf{A}$  is assumed to be real and symmetric, it follows that

$$\text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_p\} \perp \text{span}\{\mathbf{q}_{p+1}, \mathbf{q}_{p+2}, \dots, \mathbf{q}_n\}.$$

Therefore, the first  $p$  columns and last  $n - p$  columns of  $\mathbf{V}^{(k)}$  lie in a mutually orthogonal complement space. And the rate of convergence with which the two subspace become orthogonal to each other is linear with a constant  $|\lambda_{p+1}/\lambda_p|$ ;

4. Furthermore, when the above  $p = n$ , it follows that the last column of  $\mathbf{V}^{(k)}$  converges linearly to  $\pm \mathbf{q}_n$  with a constant  $|\lambda_n/\lambda_{n-1}|$ .

By the finding in the shifted inverse power method (Algorithm 40), a shift by the estimate of the smallest eigenvalue (in magnitude) in each iteration can accelerate the convergence of the algorithm, and this reveals a “practical” QR algorithm.

---

<sup>3</sup> Note the difference between the notation  $\mathbf{A}^k$  and  $\mathbf{A}^{(k)}$  where  $\mathbf{A}^k$  here is the  $k$ -th power of  $\mathbf{A}$ .

**Remark 15.2: Asymmetric Matrix  $\mathbf{A}$** 

In the above discussions, we assume  $\mathbf{A}$  is real and symmetric. If  $\mathbf{A}$  is asymmetric with real eigenvalues that are distinct in module, it can be shown that the  $\mathbf{A}^{(k)}$  in the QR algorithm converges to an upper triangular matrix where the eigenvalues lie on the diagonal (see the second form of Schur decomposition, Corollary 12.1, p. 238). For more general matrices  $\mathbf{A}$ , the sequence converges to an upper *quasi-triangular* matrix. See (Quarteroni et al., 2010; Golub and Van Loan, 2013) for more details.

**LU Algorithm** We shall only briefly discuss the LU algorithm here. Instead of factor the matrix in each iteration by a QR decomposition, the LU decomposition can be applied as well. To see this, we consider the procedure in Algorithm 48, and suppose further  $\mathbf{A}$  is nonsingular (such that its LU decomposition has nonsingular factors). We then have

$$\begin{aligned} (\text{SLU 1}) \quad \mathbf{A}^{(k)} &= \mathbf{U}^{(k)} \mathbf{L}^{(k)} = ((\mathbf{L}^{(k)})^{-1} \mathbf{L}^{(k)}) \mathbf{U}^{(k)} \mathbf{L}^{(k)} = (\mathbf{L}^{(k)})^{-1} \mathbf{A}^{(k-1)} \mathbf{L}^{(k)}, \\ (\text{SLU 2}) \quad \mathbf{A}^{(k)} &= (\mathbf{L}^{(k)})^{-1} (\mathbf{L}^{(k-1)})^{-1} \dots (\mathbf{L}^{(0)})^{-1} \mathbf{A} \mathbf{L}^{(0)} \dots \mathbf{L}^{(k-1)} \mathbf{L}^{(k)}. \end{aligned} \quad (15.13)$$

From (SLU 1), the update from  $\mathbf{A}^{(k-1)}$  to  $\mathbf{A}^{(k)}$  is a similarity transformation (not orthogonal similarity transformation now). Therefore, the accuracy depends on the condition of each  $\mathbf{L}^{(k)}$  that may be out of control. This results in the rare use of the LU algorithm. See (Rutishauser, 1958; Francis, 1961) fore more details (which is named as LR algorithm in the original papers).

**Algorithm 48** Simple LU Algorithm (Compare to Algorithm 47)

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric;

- 1:  $\mathbf{A}^{(0)} = \mathbf{A};$
- 2:  $\mathbf{V}^{(0)} = \mathbf{I}_n;$  ▷ initial eigenvector guess
- 3:  $\mathbf{U}^{(0)} = \mathbf{I}_n;$  ▷ compensate the sequence
- 4:  $\mathbf{L}^{(0)} = \mathbf{I}_n;$  ▷ compensate the sequence
- 5: **for**  $k = 1, 2, \dots$  **do**
- 6:    $\mathbf{L}^{(k)}, \mathbf{U}^{(k)} = \mathbf{LU}(\mathbf{A}^{(k-1)});$
- 7:    $\mathbf{A}^{(k)} = \mathbf{U}^{(k)} \mathbf{L}^{(k)};$
- 8:    $\mathbf{V}^{(k)} = \mathbf{V}^{(k-1)} \mathbf{L}^{(k)};$
- 9: **end for**

**15.6.5 A Practical QR Algorithm: with Shifts**

From the above discussion, we could compute the Rayleigh quotient of the last column of  $\mathbf{V}^{(k-1)}$ . Or recall that the matrix  $\mathbf{A}^{(k-1)}$  estimates the eigenvalues in the diagonal such that  $a_{nn}^{(k-1)}$  (the last diagonal of  $\mathbf{A}^{(k-1)}$ ) can be applied as a shift. The procedure is shown in Algorithm 49.

**Algorithm 49** Practical QR Algorithm (The Final Algorithm! Compare to Algorithm 47)

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric;

- 1:  $\mathbf{A}^{(0)} = \mathbf{A};$
- 2:  $\mathbf{V}^{(0)} = \mathbf{I}_n;$  ▷ initial eigenvector guess
- 3:  $\mathbf{R}^{(0)} = \mathbf{I}_n;$  ▷ compensate the sequence
- 4:  $\mathbf{Q}^{(0)} = \mathbf{I}_n;$  ▷ compensate the sequence
- 5: **for**  $k = 1, 2, \dots$  **do**
- 6:     Pick a shift  $\mu^{(k)}$ ; ▷ e.g.,  $\mu^{(k)} = a_{nn}^{(k-1)}$
- 7:      $\mathbf{Q}^{(k)}, \mathbf{R}^{(k)} = QR(\mathbf{A}^{(k-1)} - \mu^{(k)}\mathbf{I});$  ▷ QR decomposition
- 8:      $\mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} + \mu^{(k)}\mathbf{I};$  ▷ diagonals converge to eigenvalues
- 9:      $\mathbf{V}^{(k)} = \mathbf{V}^{(k-1)}\mathbf{Q}^{(k)};$  ▷ columns converge to eigenvectors
- 10: **end for**

Similar observation can be applied:

## Practical QR Algorithm Property 1

$$\left. \begin{array}{l} \mathbf{R}^{(k)} = \mathbf{Q}^{(k)\top} (\mathbf{A}^{(k-1)} - \mu^{(k)}\mathbf{I}) \\ \mathbf{A}^{(k)} = \mathbf{R}^{(k)}\mathbf{Q}^{(k)} + \mu^{(k)}\mathbf{I} \end{array} \right\} \xrightarrow{\text{leads to}} \underbrace{\mathbf{A}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{A}^{(k-1)} \mathbf{Q}^{(k)}}_{(\text{PQR 1})=(\text{SQR 1})}, \quad (15.14)$$

which is the same *similar transformation* as the simple QR algorithm shown in Equation (15.10). Similar to Equation (15.11), the same observation can be applied:

## Practical QR Algorithm Property 2

$$\mathbf{A}^{(k)} = \mathbf{V}^{(k)\top} \mathbf{A} \mathbf{V}^{(k)} \xrightarrow{\text{leads to}} \underbrace{\mathbf{A} = \mathbf{V}^{(k)} \mathbf{A}^{(k)} \mathbf{V}^{(k)\top}}_{(\text{PQR 2})=(\text{SQR 2})}, \quad (15.15)$$

where  $\mathbf{V}^{(k)} = \mathbf{Q}^{(0)}\mathbf{Q}^{(1)}\mathbf{Q}^{(2)} \dots \mathbf{Q}^{(k)}$  (same as (SQR 2)). Again,  $\mathbf{A}^{(k)} = \mathbf{V}^{(k)\top} \mathbf{A} \mathbf{V}^{(k)}$  is an *orthogonal similarity transformation* since  $\mathbf{V}^{(k)}$  is orthogonal. The condition of  $\mathbf{A}^{(k)}$  is still not worse than that of the original matrix  $\mathbf{A}$ . Further, the third property of the practical QR algorithm is slightly different to that of the simple QR algorithm:

## Practical QR Algorithm Property 3

(PQR 3)  $\neq$  (SQR 3) :

$$(\mathbf{A} - \mu^{(k)}\mathbf{I})(\mathbf{A} - \mu^{(k-1)}\mathbf{I}) \dots (\mathbf{A} - \mu^{(1)}\mathbf{I}) = \underbrace{\mathbf{Q}^{(0)}\mathbf{Q}^{(1)}\mathbf{Q}^{(2)} \dots \mathbf{Q}^{(k)}}_{=\mathbf{V}^{(k)}, \text{orthogonal}} \underbrace{\mathbf{R}^{(k)}\mathbf{R}^{(k-1)} \dots \mathbf{R}^{(0)}}_{=\mathbf{U}^{(k)}, \text{upper triangular}}. \quad (15.16)$$

Again, for clarity, the proof of Equation (15.16) is delayed in Section 15.10. Similar to the analysis in the simple QR algorithm (Section 15.6.4), assume first  $\mu^{(k)} = \mu$  is fixed in

Algorithm 49, and the eigenvalues are ordered such that  $|\lambda_1 - \mu| > |\lambda_2 - \mu| > \dots > |\lambda_n - \mu|$ , then the  $(i, i-1)$ -th entry in  $\mathbf{A}^{(k)}$  converges linearly to zero with a constant

$$\left( \left| \frac{\lambda_i - \mu}{\lambda_{i-1} - \mu} \right|^k \right).$$

This implies if  $\mu^{(k)} = a_{nn}^{(k-1)}$ ,  $|\lambda_n - \mu^{(k)}|$  tends to be much smaller than  $|\lambda_i - \mu^{(k)}|$  for  $i \in \{1, 2, \dots, n-1\}$ :

$$|\lambda_n - \mu^{(k)}| \ll |\lambda_i - \mu^{(k)}|, \quad i \in \{1, 2, \dots, n-1\}.$$

This will make the last column of  $\mathbf{A}^{(k)}$  converge to the eigenvalue of  $\mathbf{A}$  rapidly (i.e.,  $a_{nn}^{(k-1)}$  converges to the eigenvalue rapidly).

## 15.7. Apply the Practical QR Algorithm to Tridiagonal Matrices

We observe that the practical QR algorithm will result in the eigenvectors and eigenvalues via a sequence of *orthogonal similarity transformations* by the property (PQR 1) in Equation (15.14):  $\mathbf{A}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{T}^{(k-1)} \mathbf{Q}^{(k)}$ . When  $\mathbf{A}$  is symmetric, a tridiagonal decomposition  $\mathbf{A} = \mathbf{Q}^{(0)\top} \mathbf{T}^{(0)} \mathbf{Q}^{(0)}$  exists (Theorem 9.2, p. 209) which is an orthogonal similarity transformation as well. Such decomposition can be employed as a phase 1 of the QR algorithm since the tridiagonal matrix  $\mathbf{T}^{(0)}$  is close to a diagonal form and easier to converge to the diagonal eigenvalue matrix. The procedure is shown in Algorithm 50.

### 15.7.1 Explicit Shifted QR Algorithm

---

#### Algorithm 50 Practical QR Algorithm (Two Phases, Compare to Algorithm 49)

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric;

- |                                                                                           |                                                |
|-------------------------------------------------------------------------------------------|------------------------------------------------|
| 1: $\mathbf{A} = \mathbf{Q}^{(0)\top} \mathbf{T}^{(0)} \mathbf{Q}^{(0)}$ ;                | ▷ tridiagonal decomposition of $\mathbf{A}$    |
| 2: $\mathbf{V}^{(0)} = \mathbf{Q}^{(0)}$ ;                                                | ▷ previously $\mathbf{V}^{(0)} = \mathbf{I}_n$ |
| 3: $\mathbf{R}^{(0)} = \mathbf{I}_n$ ;                                                    |                                                |
| 4: <b>for</b> $k = 1, 2, \dots$ <b>do</b>                                                 |                                                |
| 5:     Pick a shift $\mu^{(k)}$ ;                                                         | ▷ e.g., $\mu^{(k)} = \mathbf{t}_{nn}^{(k-1)}$  |
| 6: $\mathbf{Q}^{(k)}, \mathbf{R}^{(k)} = QR(\mathbf{T}^{(k-1)} - \mu^{(k)} \mathbf{I})$ ; |                                                |
| 7: $\mathbf{T}^{(k)} = \mathbf{R}^{(k)} \mathbf{Q}^{(k)} + \mu^{(k)} \mathbf{I}$ ;        |                                                |
| 8: $\mathbf{V}^{(k)} = \mathbf{V}^{(k-1)} \mathbf{Q}^{(k)}$ ;                             |                                                |
| 9: <b>end for</b>                                                                         |                                                |
- 

The properties of the practical QR algorithm still hold that

$$(TPQR 1) \quad \mathbf{T}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{T}^{(k-1)} \mathbf{Q}^{(k)};$$

$$(TPQR 2') \quad \mathbf{T}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{Q}^{(k-1)\top} \dots \mathbf{Q}^{(1)\top} \mathbf{T}^{(0)} \mathbf{Q}^{(1)} \dots \mathbf{Q}^{(0)} \mathbf{Q}^{(k)} \mathbf{Q}^{(k)};$$

$$= \underbrace{\mathbf{Q}^{(k)\top} \mathbf{Q}^{(k-1)\top} \dots \mathbf{Q}^{(1)\top}}_{\mathbf{V}^{(k)\top}} \mathbf{A} \underbrace{\mathbf{Q}^{(1)} \dots \mathbf{Q}^{(0)} \mathbf{Q}^{(k)} \mathbf{Q}^{(k)}}_{\mathbf{V}^{(k)}}.$$

(15.17)

Except now that the  $\mathbf{T}^{(k)}$ 's are tridiagonal matrices (see discussion in the next section). Suppose  $\mathbf{T}^{(k-1)}$  has the following form where we want to decide the shift  $\mu^{(k)}$ :

$$\mathbf{T}^{(k-1)} = \begin{bmatrix} a_1 & b_1 & & \dots & 0 \\ b_1 & a_2 & \ddots & & \vdots \\ \ddots & \ddots & \ddots & & \\ & \ddots & a_{n-1} & b_{n-1} \\ \vdots & & b_{n-1} & a_n \\ 0 & \dots & & & \end{bmatrix}. \quad (15.18)$$

As we have discussed, one reasonable choice for the shift is  $\mu^{(k)} = a_n$ . And (Wilkinson, 1968) shows that a more effective choice is to shift by the eigenvalue of

$$\mathbf{T}_{n-1:n, n-1:n}^{(k-1)} = \begin{bmatrix} a_{n-1} & b_{n-1} \\ b_{n-1} & a_n \end{bmatrix}.$$

The above matrix has two possible eigenvalues where the one closer to  $a_n$  is chosen and it is given by

$$\mu^{(k)} = \frac{a_{n-1} + a_n}{2} + \text{sign}(d) \frac{1}{2} \sqrt{(a_{n-1} - a_n)^2 + 4b_{n-1}^2},$$

where  $d = a_n - a_{n-1}$ . This is known as the *Wilkinson shift*. (Wilkinson, 1968) shows the algorithm converges *cubically* to the eigenvectors of  $\mathbf{A}$  with either shift strategy and the Wilkinson shift is preferred for some heuristic reasons.

### 15.7.2 Implicit Shifted QR Algorithm

It happens that the shift  $\mu^{(k)}$  in Algorithm 50 is much larger than some of the diagonal value  $a_i$ 's in  $\mathbf{T}^{(k-1)}$ . Thus it is more reasonable to favor an implicit update on the  $\mathbf{T}^{(k)}$  from  $\mathbf{T}^{(k-1)}$  in which case there is no such sequence of  $\mathbf{V}^{(k)}$  converging to the eigenvectors of  $\mathbf{A}$  and it needs an extra “explicit” computation on the eigenvectors. To see this, we conclude some observations that are necessary for the implicit shift QR algorithm (make sure to recap the properties of the tridiagonal decomposition in Section 9.3, p. 212 before going forward).

- *Preservation of Form in Simple QR Algorithm.* The tridiagonal decomposition of a tridiagonal decomposition may not be a tridiagonal decomposition, i.e.,  $\mathbf{T}_+ = \mathbf{Q}\mathbf{T}\mathbf{Q}^\top$  may not be tridiagonal even if  $\mathbf{T}$  is tridiagonal. However, in our case, if  $\mathbf{T} = \mathbf{Q}\mathbf{R}$  is the QR decomposition of a symmetric and tridiagonal matrix  $\mathbf{T} \in \mathbb{R}^{n \times n}$ , then  $\mathbf{Q}$  has lower bandwidth 1 and  $\mathbf{R}$  has upper bandwidth 2 (Definition 1.1, p. 44). Then the reverse QR update  $\mathbf{T}_+ = \mathbf{R}\mathbf{Q} = \mathbf{Q}^\top \mathbf{Q}(\mathbf{R}\mathbf{Q}) = \mathbf{Q}^\top \mathbf{T}\mathbf{Q}$  is also symmetric and tridiagonal.
- *Positive Lower Subdiagonals of the Tridiagonal Matrix.* The QR decomposition is not unique (Section 3.17, p. 122). However, when restricting the diagonals of  $\mathbf{R}$  to be positive, the QR decomposition is unique (Corollary 3.1, p. 122). Then, if  $\mathbf{T}$  has *positive lower subdiagonals* (which implies  $\mathbf{T}$  is *unreduced*<sup>4</sup>, i.e., nonzero lower

---

<sup>4</sup>. See Definition 9.1, p. 209.

subdiagonals, and further the lower subdiagonals are positive), then  $\mathbf{T}_+ = \mathbf{Q}^\top \mathbf{T} \mathbf{Q}$  also has positive lower subdiagonals.<sup>5</sup>

- *Preservation of Form in Practical QR Algorithm.* If  $\mu \in \mathbb{R}$ , and  $\mathbf{T} - \mu \mathbf{I} = \mathbf{QR}$  is the QR decomposition of shifted  $\mathbf{T}$ , then  $\mathbf{T}_+ = \mathbf{RQ} + \mu \mathbf{I}$  is also symmetric and tridiagonal.
- *Implicit Q Theorem.* We observe that if we restrict the elements in the lower sub-diagonal of the tridiagonal matrix  $\mathbf{T}$  to be positive (if possible), i.e., unreduced with positive lower subdiagonals, then the tridiagonal decomposition  $\mathbf{T}_+ = \mathbf{QTQ}^\top$  is uniquely determined by  $\mathbf{T}$  and the first column of  $\mathbf{Q}$  (Theorem 9.1, p. 213);
- *Tridiagonal Update.* By condition (TPQR 1) in Equation (15.17), when the QR algorithm applied to a tridiagonal matrix, the update of the “eigenvalue matrix” forms a tridiagonal decomposition:  $\mathbf{T}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{T}^{(k-1)} \mathbf{Q}^{(k)} \rightarrow \mathbf{T}^{(k-1)} = \mathbf{Q}^{(k)} \mathbf{T}^{(k)} \mathbf{Q}^{(k)\top}$ , i.e., the tridiagonal decomposition of  $\mathbf{T}^{(k-1)}$  is given by  $\mathbf{T}^{(k-1)} = \mathbf{Q}^{(k)} \mathbf{T}^{(k)} \mathbf{Q}^{(k)\top}$ . We notice that if we assume  $\mathbf{T}^{(k-1)}$  is unreduced with positive lower subdiagonals, then  $\mathbf{T}^{(k)}$  is also unreduced with positive lower subdiagonals. By the implicit Q theorem above, the tridiagonal decomposition is uniquely determined by  $\mathbf{T}^{(k-1)}$  itself and the first column of  $\mathbf{Q}^{(k)}$ .
- If  $\mathbf{T}^{(k-1)}$  has positive lower subdiagonals,  $\mathbf{T}^{(k)}$  will also have positive lower subdiagonals which in turn result in the positive lower subdiagonals in  $\mathbf{T}^{(k+1)}$ . However, at some point, they will converge to zero.
- *Connection to the “simple” QR algorithm.* We notice that when the “simple” QR algorithm without shifts (Algorithm 47, p. 309) applied to the tridiagonal matrix also has this tridiagonal update, however, the difference relies on the first column of  $\mathbf{Q}^{(k)}$ .

We then introduce the implicit update on the practical QR algorithm.

**Step 1: Introducing the bulge** To see the implicit shift algorithm, a  $5 \times 5$  example is given step by step. Suppose at the  $(k-1)$ -th iteration, the elements in  $\mathbf{T}^{(k-1)}$  is given by Equation (15.18). Find the 2 by 2 Givens rotation  $\tilde{\mathbf{G}}_1^\top$  where  $c = \cos(\theta)$ ,  $s = \sin(\theta)$  are computed such that

$$\underbrace{\begin{bmatrix} c & s \\ -s & c \end{bmatrix}}_{\tilde{\mathbf{G}}_1^\top} \begin{bmatrix} a_1 - \mu^{(k)} \\ b_1 \end{bmatrix} = \begin{bmatrix} \otimes \\ 0 \end{bmatrix}.$$

And an  $n$  by  $n$  Givens rotation constructed by

$$\mathbf{G}_{12}^\top = \begin{bmatrix} \tilde{\mathbf{G}}_1^\top & \\ & \mathbf{I}_{n-2} \end{bmatrix},$$

where  $n = 5$  in our example and the subscript “12” of  $\mathbf{G}_{12}^\top$  denotes the position where the rotation happens (Definition 3.1, p. 116). Till now, this is equivalent to what we did in the first step of the QR decomposition of  $(\mathbf{T}^{(k-1)} - \mu^{(k)} \mathbf{I})$  via the Givens rotation (Section 3.15, p. 115). For the  $5 \times 5$  example, we realize that  $\mathbf{G}_{12}^\top$  working on  $(\mathbf{T}^{(k-1)} - \mu^{(k)} \mathbf{I})$  will introduce a zero in entry (2,1) and destroy the zero in entry (1,3). And the process is shown as follows

---

<sup>5</sup> This is an important claim that is usually ignored in many text.

where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{12}^\top \times} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \textcolor{blue}{\boxtimes} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boxtimes & \boxtimes & \mathbf{0} & \mathbf{0} \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \\ (\mathbf{T}^{(k-1)} - \mu^{(k)} \mathbf{I}) \qquad \qquad \qquad \mathbf{G}_{12}^\top (\mathbf{T}^{(k-1)} - \mu^{(k)} \mathbf{I}) \end{array}$$

And  $\mathbf{G}_{12}^\top$  working on  $\mathbf{T}^{(k-1)}$  will destroy the zero in entry (1,3):

$$\begin{array}{ccc} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{12}^\top \times} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \textcolor{blue}{\boxtimes} & \mathbf{0} & \mathbf{0} \\ \boxtimes & \boxtimes & \boxtimes & \mathbf{0} & \mathbf{0} \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\times \mathbf{G}_{12}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \textcolor{blue}{\boxtimes} & 0 & 0 \\ \mathbf{0} & \boxtimes & \boxtimes & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right], \\ \mathbf{T}^{(k-1)} \qquad \qquad \qquad \mathbf{G}_{12}^\top \mathbf{T}^{(k-1)} \qquad \qquad \qquad \mathbf{G}_{12}^\top \mathbf{T}^{(k-1)} \mathbf{G}_{12} \end{array}$$

where we find the Givens  $\mathbf{G}_{12}^\top$  multiplying on the left of a matrix will modify the first two rows of it, and  $\mathbf{G}_{12}$  multiplying on the right will modify the first two columns of it. The blue  $\boxtimes$ 's indicate the zero entries in tridiagonal matrix destroyed by the orthogonal similarity transformation and is known as “*introducing the bulge*”.

**Step 2: Chasing the bulge** Now the problem becomes “*chasing the bulge*” that will introduce the “bulge” back to zero. And in the meantime, as the final tridiagonal decomposition is decided by the first column. The second Givens rotation can be constructed by calculating  $c = \cos(\theta)$ ,  $s = \sin(\theta)$  such that

$$\underbrace{\begin{bmatrix} c & s \\ -s & c \end{bmatrix}}_{\tilde{\mathbf{G}}_2^\top} \underbrace{(\mathbf{G}_{12}^\top \mathbf{T}^{(k-1)} \mathbf{G}_{12})_{1:2,1}}_{\text{the vector in the } \boxed{\text{box}} \text{ of above matrix}} = \begin{bmatrix} \boxtimes \\ 0 \end{bmatrix}.$$

And a Givens rotation constructed by

$$\mathbf{G}_{23}^\top = \begin{bmatrix} 1 & & \\ & \tilde{\mathbf{G}}_2^\top & \\ & & \mathbf{I}_{n-3} \end{bmatrix}.$$

Following the above example, we have

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \textcolor{blue}{\boxtimes} & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{23}^\top \times} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \textcolor{blue}{\boxtimes} & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \textcolor{blue}{\boxtimes} & \mathbf{0} \\ \textcolor{brown}{0} & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\times \mathbf{G}_{23}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \textcolor{brown}{0} & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \textcolor{blue}{\boxtimes} & \mathbf{0} \\ \textcolor{brown}{0} & \mathbf{0} & \boxtimes & \boxtimes & 0 \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right], \\ \mathbf{G}_{12}^\top \mathbf{T}^{(k-1)} \mathbf{G}_{12} \qquad \qquad \qquad \mathbf{G}_{23}^\top (\mathbf{G}_{12}^\top \mathbf{T}^{(k-1)} \mathbf{G}_{12}) \qquad \qquad \qquad \mathbf{G}_{23}^\top (\mathbf{G}_{12}^\top \mathbf{T}^{(k-1)} \mathbf{G}_{12}) \mathbf{G}_{23} \end{array}$$

where we find the Givens  $\mathbf{G}_{23}^\top$  multiplying on the left of a matrix will modify the rows 2, 3 of it, and  $\mathbf{G}_{23}$  multiplying on the right will modify the first columns 2, 3 of it. Now we introduce back zero for the entries (3,1) and (1,3) from step 1, however, introducing “bulge” for entries (4,2) and (2,4) again.

Same process can go on, the example as a whole can be shown as follows where the blue  $\blacksquare$  indicates the bulge introduced, the brown  $\blacksquare$  indicates the bulge was chased out, and boldface indicates the value has just been changed:

### A Complete Example of Implicit QR Algorithm

$$\begin{array}{c}
 \left[ \begin{array}{ccccc} \blacksquare & \blacksquare & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & 0 & 0 \\ 0 & \blacksquare & \blacksquare & \blacksquare & 0 \\ 0 & 0 & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{array} \right] \xrightarrow{\mathbf{G}_{12}} \left[ \begin{array}{ccccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 \\ \mathbf{\blacksquare} & \blacksquare & \blacksquare & \blacksquare & 0 \\ \mathbf{\blacksquare} & \blacksquare & \blacksquare & \blacksquare & 0 \\ 0 & \mathbf{\blacksquare} & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & \mathbf{\blacksquare} & \blacksquare & \blacksquare \end{array} \right] \xrightarrow{\mathbf{G}_{23}} \left[ \begin{array}{ccccc} \blacksquare & \blacksquare & \mathbf{\blacksquare} & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \mathbf{\blacksquare} & \blacksquare & \blacksquare & \blacksquare & 0 \\ 0 & \mathbf{\blacksquare} & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{array} \right] \\
 \mathbf{T}^{(k-1)} \qquad \qquad \qquad \mathbf{G}_{12}^\top(\cdot)\mathbf{G}_{12} \qquad \qquad \qquad \mathbf{G}_{23}^\top(\cdot)\mathbf{G}_{23} \tag{15.19}
 \end{array}$$
  

$$\xrightarrow{\mathbf{G}_{34}} \left[ \begin{array}{ccccc} \blacksquare & \blacksquare & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & \mathbf{\blacksquare} & 0 \\ 0 & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ 0 & \mathbf{\blacksquare} & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & \mathbf{\blacksquare} & \blacksquare & \blacksquare \end{array} \right] \xrightarrow{\mathbf{G}_{45}} \left[ \begin{array}{ccccc} \blacksquare & \blacksquare & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & 0 \\ 0 & \blacksquare & \blacksquare & \blacksquare & \mathbf{\blacksquare} \\ 0 & 0 & \blacksquare & \blacksquare & \blacksquare \\ 0 & 0 & \mathbf{\blacksquare} & \blacksquare & \blacksquare \end{array} \right] \\
 \mathbf{G}_{34}^\top(\cdot)\mathbf{G}_{34} \qquad \qquad \qquad \mathbf{T}^{(k)} = \mathbf{G}_{45}^\top(\cdot)\mathbf{G}_{45}$$

For general matrix with shape  $n \times n$ , one can compute rotations  $\mathbf{G}_{12}, \mathbf{G}_{23}, \dots, \mathbf{G}_{n-1,n}$  with the property that if  $\mathbf{Z} = \mathbf{G}_{12}\mathbf{G}_{23} \dots \mathbf{G}_{n-1,n}$ , then  $\mathbf{T}^{(k-1)} = \mathbf{ZT}^{(k)}\mathbf{Z}^\top$  is the tridiagonal decomposition where the first column of  $\mathbf{Z}$  is given by  $\mathbf{Ze}_1 = \mathbf{G}_{12}\mathbf{e}_1 = \mathbf{Q}^{(k)}$ . We realize that the first column of  $\mathbf{Z}$  and  $\mathbf{Q}^{(k)}$  are identical and therefore  $\mathbf{Z} = \mathbf{Q}^{(k)}$  by implicit Q theorem under the following conditions:

- $\mathbf{T}^{(k-1)}$  is unreduced with positive lower subdiagonals;
- QR decomposition method used in the QR algorithm is uniquely in that the diagonals of the upper triangular matrix have positive diagonals;

**The complete algorithm** For simplicity, we denote the construction of  $\tilde{\mathbf{G}}_i$  such that  $\tilde{\mathbf{G}}_i^\top \mathbf{x} = \tilde{\mathbf{G}}_i^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \blacksquare \\ 0 \end{bmatrix}$  by  $\tilde{\mathbf{G}}_i^\top = \text{givens}(x_1, x_2)$ .

In all iterations,  $\tilde{\mathbf{G}}_i^\top$  will be of size  $2 \times 2$ . And the construction of  $n \times n$  Givens rotation  $\mathbf{G}_{i,i+1}^\top$  by

$$\mathbf{G}_{i,i+1}^\top = G(\tilde{\mathbf{G}}_i^\top) = \begin{bmatrix} \mathbf{I}_{i-1} & & \\ & \tilde{\mathbf{G}}_i^\top & \\ & & \mathbf{I}_{n-i-1} \end{bmatrix}.$$

For further simplicity, we will denote  $\mathbf{G}_{i,i+1}^\top$  by  $\mathbf{G}_i^\top$  which multiplies on the left of another matrix will modify the  $i$ -th and  $i+1$ -th rows implicitly. The full procedure is then formulated in Algorithm 51 where  $t_{ij}^{(k-1)}$  is the  $(i,j)$ -th entry of  $\mathbf{T}^{(k-1)}$ .

---

**Algorithm 51** Practical QR Algorithm with Implicit Shift
 

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric;

```

1: $\mathbf{A} = \mathbf{Q}^{(0)} \mathbf{T}^{(0)} \mathbf{Q}^{(0)\top};$ \triangleright tridiagonal decomposition of \mathbf{A}
2: for $k = 1, 2, \dots$ do
3: Pick a shift $\mu^{(k)}$; \triangleright e.g., $\mu^{(k)} = t_{nn}^{(k-1)}$
4: $x_1 = t_{11} - \mu^{(k)}, x_2 = t_{21};$ $\triangleright t_{ij} = t_{ij}^{(k-1)}$
5: $\mathbf{T}^{(k)} = \mathbf{T}^{(k-1)};$ \triangleright initialize $\mathbf{T}^{(k)}$
6: for $i = 1 : n - 1$ do
7: $\tilde{\mathbf{G}}_i^\top = \text{givens}(x_1, x_2);$
8: $\mathbf{G}_i^\top = G(\tilde{\mathbf{G}}_i^\top);$
9: $\mathbf{T}^{(k)} = \mathbf{G}_i^\top \mathbf{T}^{(k)} \mathbf{G}_i;$
10: if $i < n - 1$ then
11: $x_1 = t_{i+1,i}, x_2 = t_{i+2,i};$ $\triangleright t_{ij} = t_{ij}^{(k-1)}$
12: end if
13: end for
14: $\mathbf{Q}^{(k)\top} = \mathbf{G}_{n-1}^\top \dots \mathbf{G}_1^\top;$ \triangleright this results in $\mathbf{T}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{T}^{(k-1)} \mathbf{Q}^{(k)}$
15: end for

```

---

Suppose for iteration  $p$ ,  $\mathbf{T}^{(p)}$  converges to a diagonal matrix (within the machine error). Then write out the updates in each iteration:

$$\left. \begin{array}{l} \mathbf{T}^{(p)} = \mathbf{Q}^{(p)\top} \mathbf{T}^{(p-1)} \mathbf{Q}^{(p)} \\ \mathbf{T}^{(p-1)} = \mathbf{Q}^{(p-1)\top} \mathbf{T}^{(p-2)} \mathbf{Q}^{(p-1)} \\ \vdots = \vdots \\ \mathbf{T}^{(1)} = \mathbf{Q}^{(1)\top} \mathbf{T}^{(0)} \mathbf{Q}^{(1)} \\ \mathbf{T}^{(0)} = \mathbf{Q}^{(0)\top} \mathbf{A} \mathbf{Q}^{(0)} \end{array} \right\} \xrightarrow{\text{leads to}} \mathbf{A} = \underbrace{\mathbf{Q}^{(0)} \dots \mathbf{Q}^{(p)}}_{\mathbf{Q}} \mathbf{T}^{(p)} \underbrace{(\mathbf{Q}^{(0)} \dots \mathbf{Q}^{(p)})^\top}_{\mathbf{Q}^\top}$$

is the approximated spectral decomposition of real symmetric  $\mathbf{A}$  where  $\mathbf{Q}$  is orthogonal containing the eigenvectors of  $\mathbf{A}$  and  $\mathbf{T}^{(p)}$  is diagonal containing eigenvalues of  $\mathbf{A}$  (Theorem 13.1, p. 241).

**Decouple** However, we can assure that the  $\mathbf{T}^{(k-1)}$  has nonnegative lower subdiagonals (the specific QR decomposition favored as mentioned above), whereas it happens that it is *reduced*, i.e., some lower subdiagonals are 0. In this case, the eigenproblem splits into a pair of smaller problems. For example, when  $\mathbf{T}_{k+1,k}^{(k-1)} = 0$ , then, a “practical” QR algorithm can be applied to the submatrices:

$$\mathbf{T}_{1:k,1:k}^{(k-1)} \quad \text{and} \quad \mathbf{T}_{k+1:n,k+1:n}^{(k-1)}.$$

And the eigenvalues can be obtained by

$$\Lambda(\mathbf{T}^{(k-1)}) = \Lambda(\mathbf{T}_{1:k,1:k}^{(k-1)}) \cup \Lambda(\mathbf{T}_{k+1:n,k+1:n}^{(k-1)}),$$

where  $\Lambda(\cdot)$  is the spectrum of a matrix (Definition 0.3, p. 16).

## 15.8. Jacobi's Method

Jacobi's method is one of the oldest method to compute the eigenvalues of a matrix which was introduced in 1846 by ([Jacobi and Verfahren, 1846](#)). The idea is to diagonalize a small submatrix at each time such that the full matrix can be diagonalized eventually. The mathematical measure for the quantity of the reduction is defined by *off-diagonal norm*

$$\text{off}(A) = \sqrt{\sum_{i=1}^n \sum_{j=1, j \neq i}^n a_{ij}^2},$$

i.e., the Frobenius norm of the off-diagonal entries. The idea behind the method is to iteratively reduce the off-diagonal quantity and relies on the Jacobi's rotation

$$J_{kl} = \begin{bmatrix} 1 & & & & & \\ \ddots & & & & & \\ & 1 & & & & \\ & & c & & & s \\ & & & 1 & & \\ & & & & \ddots & \\ & & -s & & & 1 \\ & & & & c & \\ & & & & & 1 \\ & & & & & \\ & k & & l & & \ddots \end{bmatrix},$$

which is the same as the Givens rotation (Definition 3.1, p. 116). The difference relies on the usage of them where in the Jacobi's rotation, the  $\theta$  for  $s = \cos \theta$  and  $c = \sin \theta$  is chosen to make the submatrix of  $\mathbf{J}^\top \mathbf{A} \mathbf{J}$  to be diagonal.

### 15.8.1 The 2 by 2 Case

To see how this Jacobi's rotation works, suppose we are looking at the 2 by 2 submatrix of a symmetric matrix

$$A(k, l) := \begin{bmatrix} a_{kk} & a_{kl} \\ a_{lk} & a_{ll} \end{bmatrix} = \begin{bmatrix} a_{kk} & a_{kl} \\ a_{kl} & a_{ll} \end{bmatrix}.$$

Then  $\theta$  can be computed such that

$$\begin{aligned} \mathbf{J}^\top \mathbf{A}(k, l) \mathbf{J} &= \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} a_{kk} & a_{kl} \\ a_{kl} & a_{ll} \end{bmatrix} \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \\ &= \begin{bmatrix} (c^2 - cs)a_{kk} + s^2a_{ll} - cs \cdot a_{kl} & (c^2 - s^2)a_{kl} + cs(a_{kk} - a_{ll}) \\ (c^2 - s^2)a_{kl} + cs(a_{kk} - a_{ll}) & s^2a_{kk} + c^2a_{ll} + 2cs \cdot a_{kl} \end{bmatrix} = \begin{bmatrix} \neq 0 & 0 \\ 0 & \neq 0 \end{bmatrix}. \end{aligned}$$

If  $a_{kl} = 0$ , then we can just set  $c = 1, s = 0$ , the submatrix  $\mathbf{A}(k, l)$  remains unchanged. Otherwise, it is trivial that both  $c \neq 0$  and  $s \neq 0$ . Divide the off-diagonal element  $\{(c^2 - s^2)a_{kl} + cs(a_{kk} - a_{ll})\}$  by  $c^2$  and  $a_{kl}$ , it follows that

$$\left(1 - \frac{s^2}{c^2} + \frac{s}{c} \frac{a_{kk} - a_{ll}}{a_{kl}}\right) = 0$$

Let

$$\tan \theta = t = \frac{\sin \theta}{\cos \theta} = \frac{s}{c} \quad \text{and} \quad \tau = \frac{a_{ll} - a_{kk}}{2a_{kl}},$$

it suffices to solve the equation

$$t^2 + 2\tau t - 1 = 0.$$

This gives

$$t = -\tau \pm \sqrt{\tau^2 + 1}.$$

We notice that

$$\|\mathbf{A} - \mathbf{J}_{kl}^\top \mathbf{A} \mathbf{J}_{kl}\|_F^2 = 4(1 - c) \sum_{i \neq k, l} (a_{ik}^2 + a_{il}^2) + 2a_{kl}^2/c^2,$$

and

$$c = \frac{1}{\sqrt{1 + t^2}}.$$

Therefore, the larger the  $c$ , the smaller the  $\|\mathbf{A} - \mathbf{J}_{kl}^\top \mathbf{A} \mathbf{J}_{kl}\|_F^2$ . This implies we should choose a  $t$  with smaller magnitude:

$$t_{min} = \begin{cases} -\tau + \sqrt{\tau^2 + 1}, & \text{if } \tau \geq 0; \\ -\tau - \sqrt{\tau^2 + 1}, & \text{if } \tau < 0. \end{cases}$$

From the discussion above, we can define the *ComputeJacobiRotation* function that computes the Jacobi's rotation from the submatrix of  $\mathbf{A}$ .

---

**Algorithm 52** Compute Jacobi Rotation Given the Submatrix

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric,  $(k, l)$  such that  $1 \leq k < l \leq n$ ;

```

1: function COMPUTEJACOBIRotation(\mathbf{A} , k, l)
2: if $A_{kl} \neq 0$ then
3: $\tau = (a_{ll} - a_{kk})/(2a_{kl})$
4: if $\tau \geq 0$ then
5: $t = -\tau + \sqrt{\tau^2 + 1};$
6: else
7: $t = -\tau - \sqrt{\tau^2 + 1};$
8: end if
9: $c = 1/\sqrt{1 + t^2}, s = tc;$
10: else
11: $c = 1, s = 0;$
12: end if
13: Output $c, s;$
14: end function
```

---

### 15.8.2 The Complete Jacobi's Method

The name of the “complete” comes from the terminology of the complete pivoting that searches the largest element in the matrix to pivot (Section 1.13.2, p. 51). At each iteration, we need to decide the submatrix  $\mathbf{A}(k, l)$  to diagonalize. Then in the complete Jacobi's method, we choose  $(k, l)$  such that the  $(k, l)$ -th entry,  $a_{kl}^2$ , is maximal hoping that the reduction of the off-diagonal quantity to be maximal. The complete search for the largest magnitude defines the Algorithm 53.

---

#### Algorithm 53 Complete Jacobi's Method

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric, a positive tolerance  $tol$  such that  $\delta = tol \cdot \|\mathbf{A}\|_F$ ;

- 1:  $\mathbf{Q} = \mathbf{I}_n$ ;
  - 2:  $i = 0$ ;
  - 3:  $\mathbf{\Lambda} = \mathbf{A}$ ;
  - 4: **while**  $\text{off}(\mathbf{\Lambda}) > \delta$  **do**
  - 5:     Choose  $(k, l)$  so that  $a_{kl}^2 = \arg \max \mathbf{\Lambda}_{ij}^2, \forall 1 \leq i, j \leq n$ ;
  - 6:     Compute  $c, s$  from  $\text{computeJacobiRotation}(\mathbf{\Lambda}, k, l)$ ;
  - 7:     Decide the  $n \times n$  Jacobi's rotation  $\mathbf{J}_{kl}$  by  $c, s$ ;
  - 8:      $\mathbf{\Lambda} = \mathbf{J}_{kl}^\top \mathbf{\Lambda} \mathbf{J}_{kl}$ ;
  - 9:      $\mathbf{Q} = \mathbf{Q} \mathbf{J}_{kl}$ ;
  - 10:     Compute  $\text{off}(\mathbf{\Lambda}^{(i)})^2$ ; ▷ which we will show converges linearly to 0.
  - 11:      $i = i + 1$ ;
  - 12: **end while**
  - 13: Output the sequence  $\text{off}(\mathbf{\Lambda}^{(i)})^2$ , approximated diagonal  $\mathbf{\Lambda}$  and orthogonal  $\mathbf{Q}$ ;
- 

The algorithm computes the spectral decomposition by outputting  $\mathbf{A} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^\top$ . It is not hard to see that, at each iteration, it requires  $O(n^2)$  flops to search for the largest magnitude entry  $(k, l)$  and  $O(n)$  flops to update the iteration. We shall notice that although a symmetric pair of zeros is introduced into the matrix at each iteration, the previous zeros are destroyed. However, what matters is the off-diagonal quantity reduces steadily.

Suppose the index  $(k, l)$  is chosen and the Jacobi's rotation  $\mathbf{J}_{kl}$  is constructed such that

$$\mathbf{\Lambda}_+ = \mathbf{J}_{kl}^\top \mathbf{\Lambda} \mathbf{J}_{kl}.$$

Since the Jacobi's rotation is orthogonal, it follows that

$$\|\mathbf{\Lambda}_+\|_F^2 = \|\mathbf{\Lambda}\|_F^2$$

and

$$\text{off}(\mathbf{\Lambda}_+)^2 = \text{off}(\mathbf{\Lambda})^2 - 2a_{kl}^2.$$

As  $a_{kl}$  has largest magnitude in  $\mathbf{\Lambda}$ , we also have

$$\text{off}(\mathbf{\Lambda})^2 \leq n(n-1)a_{kl}^2.$$

Therefore,

$$\text{off}(\mathbf{\Lambda}_+)^2 = \text{off}(\mathbf{\Lambda})^2 - 2a_{kl}^2 \leq \text{off}(\mathbf{\Lambda})^2 - \frac{2}{n(n-1)}\text{off}(\mathbf{\Lambda})^2 = \left(1 - \frac{2}{n(n-1)}\right)\text{off}(\mathbf{\Lambda})^2.$$

By induction, this implies, the Jacobi's method reduces the off-diagonal quantity after  $k$  iterations by

$$\text{off}(\boldsymbol{\Lambda}^{(i)})^2 \leq \left(1 - \frac{2}{n(n-1)}\right)^2 \text{off}(\boldsymbol{\Lambda}^{(0)}).$$

and

$$\lim_{k \rightarrow \infty} \frac{|\text{off}(\boldsymbol{\Lambda}^{(i)})^2 - 0|}{|\text{off}(\boldsymbol{\Lambda}^{(i-1)})^2 - 0|} = 1 - \frac{2}{n(n-1)} \in (0, 1),$$

such that the sequence  $\text{off}(\boldsymbol{\Lambda}^{(i)})^2$  converges linearly to 0 (Definition 15.2, p. 292). However, in practice, (Henrici, 1958; Schönhage, 1964; Van Kempen, 1966) show that a quadratic convergence (Definition 15.4, p. 292) can be obtained such that

$$\lim_{k \rightarrow \infty} \frac{|\text{off}(\boldsymbol{\Lambda}^{(i+\frac{n(n-1)}{2})})^2|}{|\text{off}(\boldsymbol{\Lambda}^{(i)})^2|^2} = c,$$

where  $c$  is a constant. We shall not give details.

### 15.8.3 The Cyclic-by-Row Jacobi's Method

We mentioned in the last section that most cost of the complete Jacobi's method is from the search for the element with largest magnitude which takes  $O(n^2)$  flops. To tame the explosion, it is reasonable to update in a row-by-row fashion that iteratively updates the  $n(n-1)$  upper triangular entries. This is known as the *cyclic-by-row* algorithm and the procedure is shown in Algorithm 54.

---

#### Algorithm 54 Cyclic-by-Row Jacobi's Method

---

**Require:**  $\boldsymbol{A} \in \mathbb{R}^{n \times n}$  is real and symmetric, a positive tolerance  $tol$  such that  $\delta = tol \cdot \|\boldsymbol{A}\|_F$ ;

- 1:  $\boldsymbol{Q} = \boldsymbol{I}_n$ ;
- 2:  $i = 0$ ;
- 3:  $\boldsymbol{\Lambda} = \boldsymbol{A}$ ;
- 4: **while**  $\text{off}(\boldsymbol{\Lambda}) > \delta$  **do**
- 5:     **for**  $k = 1 : n - 1$  **do**
- 6:         **for**  $l = k + 1 : n$  **do**
- 7:             Compute  $c, s$  from  $\text{computeJacobiRotation}(\boldsymbol{\Lambda}, k, l)$ ;
- 8:             Decide the  $n \times n$  Jacobi's rotation  $\boldsymbol{J}_{kl}$  by  $c, s$ ;
- 9:              $\boldsymbol{\Lambda} = \boldsymbol{J}_{kl}^\top \boldsymbol{\Lambda} \boldsymbol{J}_{kl}$ ;
- 10:           $\boldsymbol{Q} = \boldsymbol{Q} \boldsymbol{J}_{kl}$ ;
- 11:           $\text{off}(\boldsymbol{\Lambda}^{(i)})^2$ ;
- 12:           $i = i + 1$ ;
- 13:     **end for**
- 14:   **end for**
- 15: **end while**
- 16: Output the sequence  $\text{off}(\boldsymbol{\Lambda}^{(i)})^2$ , approximated diagonal  $\boldsymbol{\Lambda}$  and orthogonal  $\boldsymbol{Q}$ ;

---

Similarly, (Wilkinson, 1962; Van Kempen, 1966) show that the cyclic-by-row algorithm converges *quadratically*. And in that it does not need to compare the element in the matrix, this reduces the complexity of calculation.

### 15.8.4 Other Issues

In practice, when computing on a  $p$ -processor computer, it is reasonable to do the Jacobi's algorithm in a block way such that a parallelizable method can be favored. We refer the issue to (Bischof, 1986; Shroff and Schreiber, 1989; Golub and Van Loan, 2013) and also its counterpart in SVD computation (Van Loan, 1985).

## 15.9. Computing the SVD

### 15.9.1 Implicit Shifted QR Algorithm

We previously have shown that any symmetric matrix can be reduced to tridiagonal form via a sequence of Householder reflectors that are applied on the left and the right to the matrix, whilst it is a special case of the Hessenberg decomposition (Section 9.2, p. 209). This can reduce the cost of the QR algorithm applied to computing the spectral decomposition of a matrix. This two-phase computation is not unique but has its counterpart. Since the work of Golub, Kahan, and others on the bidiagonalization in the 1960s (Theorem 10.2, p. 215), an analogous two-phase approach has been standard for the SVD. For computing the SVD, the matrix is firstly reduced to bidiagonal form<sup>6</sup> and then the bidiagonal matrix is diagonalized:

$$\begin{array}{c} \left[ \begin{array}{cccccc} \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare & \blacksquare \end{array} \right] \xrightarrow{\text{phase 1}} \text{bidiagonalize} \left[ \begin{array}{ccccc} \blacksquare & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & \blacksquare & 0 & 0 \\ 0 & 0 & \blacksquare & \blacksquare & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \\ 0 & 0 & 0 & 0 & \blacksquare \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{\text{phase 2}} \text{diagonalize} \left[ \begin{array}{ccccc} \blacksquare & 0 & 0 & 0 & 0 \\ 0 & \blacksquare & 0 & 0 & 0 \\ 0 & 0 & \blacksquare & 0 & 0 \\ 0 & 0 & 0 & \blacksquare & 0 \\ 0 & 0 & 0 & 0 & \blacksquare \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right]. \end{array}$$

The bidiagonalization is different from the tridiagonalization in that it does not require the two orthogonal on the left and on the right to be the transpose of the same matrix, and even the shape of the two orthogonal matrices can be different. This is exactly what we want for the SVD. For the second phase of the SVD computation, we will directly go to the solution via the implicit shift QR algorithm, the development to this final procedure is similar from what we have developed for the spectral decomposition.

**Phase 2 of SVD** Following the fact that  $\mathbf{T} = \mathbf{B}^\top \mathbf{B}$  is tridiagonal if  $\mathbf{B}$  is bidiagonal (Lemma 10.1, p. 224). Following the tridiagonal update in QR algorithm (Section 15.7.2), suppose in the  $k$ -th iteration, we have the bidiagonal matrix  $\mathbf{B}^{(k-1)}$  and its tridiagonal companion  $\mathbf{T}^{(k-1)} = \mathbf{B}^{(k-1)\top} \mathbf{B}^{(k-1)}$ , for  $k = 1, 2, \dots$ . The tridiagonal update is followed by a set of Givens rotations on the left and right iteratively as shown in Equation (15.19) by a  $5 \times 5$  example where the blue  $\blacksquare$  indicates the bulge introduced and the boxed vector

---

6. The bidiagonal matrix we discuss about in this section implicitly means an upper bidiagonal matrix. We will suppress the terminology for simplicity.

indicates how the Givens matrix constructed:

$$\begin{aligned}
 \mathbf{T}^{(k-1)} &= \begin{bmatrix} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{G_{12}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ \textcolor{blue}{\boxed{\boxtimes}} & \boxtimes & \boxtimes & 0 & 0 \\ \textcolor{blue}{\boxed{\boxtimes}} & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{G_{23}} \begin{bmatrix} \boxtimes & \boxtimes & \textcolor{brown}{0} & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \textcolor{blue}{\boxed{\boxtimes}} & 0 \\ \textcolor{brown}{0} & \boxed{\boxtimes} & \boxtimes & \boxtimes & 0 \\ 0 & \textcolor{blue}{\boxed{\boxtimes}} & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{brown}{0} & \boxtimes & \boxtimes \end{bmatrix} \\
 &\quad \mathbf{T}^{(k-1)} \qquad \qquad \qquad \mathbf{G}_{12}^\top(\cdot)\mathbf{G}_{12} \qquad \qquad \qquad \mathbf{G}_{23}^\top(\cdot)\mathbf{G}_{23} \\
 &\xrightarrow{G_{34}} \begin{bmatrix} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \textcolor{brown}{0} & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \textcolor{blue}{\boxed{\boxtimes}} \\ 0 & \textcolor{brown}{0} & \boxed{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxed{\boxtimes}} & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{G_{45}} \begin{bmatrix} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \textcolor{brown}{0} \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \textcolor{brown}{0} & \boxtimes \end{bmatrix} = \mathbf{T}^{(k)}. \\
 &\qquad \qquad \qquad \mathbf{G}_{34}^\top(\cdot)\mathbf{G}_{34} \qquad \qquad \qquad \mathbf{G}_{45}^\top(\cdot)\mathbf{G}_{45}
 \end{aligned}$$

That is,

$$\begin{aligned}
 \mathbf{T}^{(k)} &= (\mathbf{G}_{45}^\top \mathbf{G}_{34}^\top \mathbf{G}_{23}^\top \mathbf{G}_{12}^\top) \mathbf{T}^{(k-1)} (\mathbf{G}_{12} \mathbf{G}_{23} \mathbf{G}_{34} \mathbf{G}_{45}) \\
 &= (\mathbf{G}_{45}^\top \mathbf{G}_{34}^\top \mathbf{G}_{23}^\top \mathbf{G}_{12}^\top) (\mathbf{B}^{(k-1)\top} \mathbf{B}^{(k-1)}) (\mathbf{G}_{12} \mathbf{G}_{23} \mathbf{G}_{34} \mathbf{G}_{45}).
 \end{aligned}$$

If we can find  $\mathbf{T}^{(k)}$  can also be decomposed into a product of bidiagonal matrices, then the “tridiagonal update” can be replaced by a “bidiagonal update”. Then two problems must be tamed:

- Decompose  $\mathbf{T}^{(k)}$  into the bidiagonal form if  $\mathbf{T}^{(k-1)}$  has the bidiagonal form:  $\mathbf{T}^{(k-1)} = \mathbf{B}^{(k-1)\top} \mathbf{B}^{(k-1)}$ ;
- Find the Givens rotations from the bidiagonal matrix  $\mathbf{B}^{(k-1)}$  rather than the tridiagonal one  $\mathbf{T}^{(k-1)}$ .

It is not hard to see that what we want to prove is either

$$\text{Choice 1: } \mathbf{B}^{(k)\top} = (\mathbf{G}_{45}^\top \mathbf{G}_{34}^\top \mathbf{G}_{23}^\top \mathbf{G}_{12}^\top) \mathbf{B}^{(k-1)\top}$$

or

$$\text{Choice 2: } \mathbf{B}^{(k)\top} = (\mathbf{G}_{45}^\top \mathbf{G}_{34}^\top \mathbf{G}_{23}^\top \mathbf{G}_{12}^\top) \mathbf{B}^{(k-1)\top} \mathbf{V}_{12} \mathbf{V}_{23} \mathbf{V}_{34} \mathbf{V}_{45}.$$

I.e., one of the above choices is also lower bidiagonal if  $\mathbf{B}^{(k-1)\top}$  is lower bidiagonal. The  $\mathbf{V}_{i,i+1}$ 's are orthogonal such that they will cancel out in  $\mathbf{B}^{(k)\top} \mathbf{B}^{(k)}$ . We will see second form of  $\mathbf{B}^{(k)\top}$  will be used (to chase the bulge). Let

$$\mathbf{T}^{(k-1)} = \begin{bmatrix} a_1 & b_1 & & \dots & 0 \\ b_1 & a_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & b_4 \\ \vdots & & & & \\ 0 & \dots & & b_4 & a_5 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & & \dots & 0 \\ d_1 & c_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \ddots \\ \vdots & & & & \\ 0 & \dots & & d_4 & c_5 \end{bmatrix} \begin{bmatrix} c_1 & d_1 & & \dots & 0 \\ 0 & c_2 & \ddots & & \vdots \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & d_4 \\ \vdots & & & & \\ 0 & \dots & & 0 & c_5 \end{bmatrix} \\
 \qquad \qquad \qquad \mathbf{B}^{(k-1)\top} \qquad \qquad \qquad \mathbf{B}^{(k-1)}$$

**Introducing and chasing the bulge in the bidiagonal update** Following the Givens rotations constructed in Section 15.7.2, it follows that a Givens rotation constructed by

$$\mathbf{G}_{12}^\top = \begin{bmatrix} \tilde{\mathbf{G}}_1^\top & \\ & \mathbf{I}_{n-2} \end{bmatrix}, \quad \text{where} \quad \underbrace{\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} c_1^2 - \mu^{(k)} \\ c_1 d_1 \end{bmatrix}}_{\tilde{\mathbf{G}}_1^\top} = \begin{bmatrix} \boxtimes \\ 0 \end{bmatrix}.$$

where  $\mu^{(k)}$  is the shift in the  $k$ -th iteration,  $n = 5$  and  $a_1 = c_1^2, b_1 = c_1 d_1$  in our example, this will introduce a bulge in  $\mathbf{B}^{(k-1)\top}$ :

$$\begin{array}{c} \begin{bmatrix} \boxtimes & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & \boxtimes & 0 \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_{12}^\top \times} \begin{bmatrix} \boxtimes & \textcolor{blue}{\boxtimes} & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & \boxtimes & 0 \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \\ \mathbf{B}^{(k-1)\top} \qquad \qquad \qquad \mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top} \end{array}.$$

Then, we observe that  $\mathbf{G}_{23}$  rotating on the left will not help chase out the bulge <sup>7</sup>, a right Givens rotations must be constructed such that

$$\mathbf{V}_{12}^\top = \begin{bmatrix} \tilde{\mathbf{V}}_1^\top & \\ & \mathbf{I}_{n-2} \end{bmatrix}, \quad \text{where} \quad \underbrace{\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} (\mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top})_{11} \\ (\mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top})_{12} \end{bmatrix}}_{\tilde{\mathbf{V}}_1^\top} = \begin{bmatrix} \boxtimes \\ 0 \end{bmatrix}.$$

This results in

$$\begin{array}{c} \begin{bmatrix} \boxtimes & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & \boxtimes & 0 \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_{12}^\top \times} \begin{bmatrix} \boxtimes & \textcolor{blue}{\boxtimes} & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & \boxtimes & 0 \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \xrightarrow{\times \mathbf{V}_{12}} \begin{bmatrix} \boxtimes & \textcolor{brown}{0} & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 \\ \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & \boxtimes & 0 \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \\ \mathbf{B}^{(k-1)\top} \qquad \qquad \qquad \mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top} \qquad \qquad \qquad \mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top} \mathbf{V}_{12} \end{array}.$$

As long as the  $\mathbf{G}_{12}^\top$  is constructed in the same way as it in the “tridiagonal update”, then  $\mathbf{G}_{12}^\top \mathbf{T}^{k-1} \mathbf{G}_{12}$  is equal to  $\underbrace{\mathbf{G}_{12}^\top \mathbf{B}^{k-1\top} \mathbf{V}_{12} \mathbf{V}_{12}^\top}_{\mathbf{I}} \mathbf{B}^{(k-1)\top} \mathbf{G}_{12}$ .

**Step 2** The step 2 is different to what we have did in the “tridiagonal update”. In the “tridiagonal update”, we construct a Givens matrix  $\mathbf{G}_{23}$  by chasing the bulge in  $\mathbf{T}^{(k-1)}$ . Apply the implicit Q theorem again (Theorem 9.1, p. 213), we can decide the Givens rotation by  $\mathbf{B}^{(k-1)}$  directly (*need not to construct  $\mathbf{T}^{(k-1)} = \mathbf{B}^{(k-1)\top} \mathbf{B}^{(k-1)}$  explicitly*), and now denoted by  $\mathbf{U}_{23}$ :

$$\mathbf{U}_{23}^\top = \begin{bmatrix} 1 & \tilde{\mathbf{U}}_2^\top & \\ & \mathbf{I}_{n-3} & \end{bmatrix}, \quad \text{where} \quad \underbrace{\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} (\mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top} \mathbf{V}_{12})_{21} \\ (\mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top} \mathbf{V}_{12})_{31} \end{bmatrix}}_{\tilde{\mathbf{U}}_2^\top} = \begin{bmatrix} \boxtimes \\ 0 \end{bmatrix}.$$

<sup>7</sup>.  $\mathbf{G}_{23}$  working on the left of  $\mathbf{G}_{12} \mathbf{B}^{(k-1)}$  will modify the rows 2 and 3 of it which will not introduce zero back in the first row.

The process can go on, and the full example for chasing the bulge can be shown as follows where the blue  $\blacksquare$  indicates the bulge introduced, the brown  $\blacksquare$  indicates the bulge was chased out, and **boldface** indicates the value has just been changed:

### A Complete Example of Implicit QR Algorithm for SVD

$$\begin{array}{c}
 \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \blacksquare & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & \blacksquare & 0 & 0 \\ 0 & 0 & \blacksquare & \blacksquare & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \xrightarrow{\mathbf{G}_{12}^\top \times} \begin{matrix} \blacksquare & \blacksquare & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \mathbf{0} & 0 & 0 \\ 0 & \blacksquare & \blacksquare & 0 & 0 \\ 0 & 0 & \blacksquare & \blacksquare & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \xrightarrow{\times \mathbf{V}_{12}} \begin{matrix} \blacksquare & \mathbf{0} & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \mathbf{0} & 0 & 0 \\ \mathbf{0} & \blacksquare & \blacksquare & 0 & 0 \\ \mathbf{0} & 0 & \blacksquare & \blacksquare & 0 \\ \mathbf{0} & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \\
 \mathbf{B}^{(k-1)\top} \quad \mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top} \quad \mathbf{G}_{12}^\top \mathbf{B}^{(k-1)\top} \mathbf{V}_{12}
 \end{array}$$
  

$$\begin{array}{c}
 \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \blacksquare & \blacksquare & \blacksquare & \mathbf{0} & 0 \\ \mathbf{0} & \blacksquare & \blacksquare & \mathbf{0} & 0 \\ 0 & 0 & \blacksquare & \blacksquare & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \xrightarrow{\mathbf{U}_{23}^\top \times} \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \mathbf{0} & \blacksquare & \mathbf{0} & 0 & 0 \\ 0 & \blacksquare & \blacksquare & \mathbf{0} & 0 \\ 0 & 0 & \blacksquare & \blacksquare & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \xrightarrow{\times \mathbf{V}_{23}} \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \mathbf{0} & \blacksquare & \mathbf{0} & 0 & 0 \\ 0 & \blacksquare & \blacksquare & \mathbf{0} & 0 \\ 0 & \mathbf{0} & \blacksquare & \blacksquare & 0 \\ 0 & 0 & \mathbf{0} & \blacksquare & \blacksquare \end{matrix} \\
 \mathbf{U}_{23}^\top(\cdot) \quad \mathbf{U}_{23}^\top(\cdot) \mathbf{V}_{23}
 \end{array}$$
  

$$\begin{array}{c}
 \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \blacksquare & \blacksquare & 0 & 0 & 0 \\ \mathbf{0} & \blacksquare & \blacksquare & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & \blacksquare & \blacksquare & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \xrightarrow{\mathbf{U}_{34}^\top \times} \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \mathbf{0} & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & \mathbf{0} & 0 & 0 \\ 0 & 0 & \blacksquare & \mathbf{0} & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \xrightarrow{\times \mathbf{V}_{34}} \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \mathbf{0} & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & \mathbf{0} & 0 & 0 \\ 0 & 0 & \blacksquare & \mathbf{0} & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \\
 \mathbf{U}_{34}^\top(\cdot) \quad \mathbf{U}_{34}^\top(\cdot) \mathbf{V}_{34}
 \end{array}$$
  

$$\begin{array}{c}
 \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \blacksquare & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & \blacksquare & 0 & 0 \\ 0 & 0 & \blacksquare & \blacksquare & \mathbf{0} \\ 0 & 0 & 0 & \mathbf{0} & \blacksquare \end{matrix} \xrightarrow{\mathbf{U}_{45}^\top \times} \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \mathbf{0} & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & \mathbf{0} & 0 & 0 \\ 0 & 0 & \blacksquare & \mathbf{0} & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix} \xrightarrow{\times \mathbf{V}_{45}} \begin{matrix} \blacksquare & 0 & 0 & 0 & 0 \\ \mathbf{0} & \blacksquare & 0 & 0 & 0 \\ 0 & \blacksquare & \mathbf{0} & 0 & 0 \\ 0 & 0 & \blacksquare & \mathbf{0} & 0 \\ 0 & 0 & 0 & \blacksquare & \blacksquare \end{matrix}, \\
 \mathbf{U}_{45}^\top(\cdot) \quad \mathbf{U}_{45}^\top(\cdot) \mathbf{V}_{45}
 \end{array}$$

where  $\mathbf{G}_{12}$  is used in the first step to indicate the same construction as “tridiagonal update”, and  $\mathbf{U}_{i,i+1}$ ’s are used to indicate the different construction via the implicit Q theorem.

For clarity, we will denote  $\mathbf{G}_{12}$  by  $\mathbf{U}_{12}$  and  $\tilde{\mathbf{G}}_1$  by  $\tilde{\mathbf{U}}_1$  in the following algorithm.

**The complete algorithm** Again, for simplicity, we denote the construction of  $\tilde{\mathbf{U}}_i$  such that  $\tilde{\mathbf{U}}_i^\top \mathbf{x} = \tilde{\mathbf{U}}_i^\top \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \blacksquare \\ 0 \end{bmatrix}$  by

$$\tilde{\mathbf{U}}_i^\top = \text{givens}(x_1, x_2).$$

In all iterations,  $\tilde{\mathbf{U}}_i^\top$  will be of size  $2 \times 2$ . And the construction of the  $n \times n$  Givens rotation  $\mathbf{G}_{i,i+1}^\top$  by

$$\mathbf{U}_{i,i+1}^\top = G(\tilde{\mathbf{U}}_i^\top) = \begin{bmatrix} \mathbf{I}_{i-1} & & \\ & \tilde{\mathbf{U}}_i^\top & \\ & & \mathbf{I}_{n-i-1} \end{bmatrix}.$$

For further simplicity, we will denote  $\mathbf{U}_{i,i+1}^\top$  by  $\mathbf{U}_i^\top$  with multiplying on the left of another matrix will modify the  $i$ -th and  $i+1$ -th rows implicitly. The procedure is then shown in Algorithm 55 where the shift  $\mu^{(k)} = t_{nn}^{(k-1)}$  (i.e., the last diagonal of  $\mathbf{T}^{(k-1)}$ ) can be obtained via the last diagonal of the multiplication of  $\mathbf{B}^{(k-1)\top} \mathbf{B}^{(k-1)}$ , i.e.,  $\mu^{(k)} = b_{n,n-1}^2 + b_{nn}^2$ .

---

**Algorithm 55** Golub-Kahan SVD: Practical QR Algorithm with Implicit Shift (Compare to Algorithm 51)

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is real and symmetric;

```

1: $\mathbf{A}^\top = \mathbf{V}^{(0)\top} \mathbf{B}^{(0)\top} \mathbf{Q}^{(0)\top}$; \triangleright bidiagonal decomposition of \mathbf{A}^\top
2: for $k = 1, 2, \dots$ do
3: Pick a shift $\mu^{(k)}$; \triangleright e.g., $\mu^{(k)} = t_{nn}^{(k-1)} = b_{n,n-1}^2 + b_{nn}^2$
4: $x_1 = t_{11} - \mu^{(k)}, x_2 = t_{21};$ $\triangleright t_{ij} = t_{ij}^{(k-1)}$
5: $\mathbf{B}^{(k)} = \mathbf{B}^{(k-1)};$ \triangleright initialize $\mathbf{B}^{(k)}$
6: for $i = 1 : n-1$ do
7: $\mathbf{U}_i^\top = G(\tilde{\mathbf{U}}_i^\top)$, where $\tilde{\mathbf{U}}_i^\top = \text{givens}(x_1, x_2);$
8: $\mathbf{B}^{(k)\top} = \mathbf{U}_i^\top \mathbf{B}^{(k)\top};$ \triangleright left update
9: $x_1 = b_{ii}, x_2 = b_{i,i+1};$ $\triangleright b_{ij} = b_{ij}^{(k)\top}$
10: $\mathbf{V}_i^\top = G(\tilde{\mathbf{V}}_i^\top)$ where $\tilde{\mathbf{V}}_i^\top = \text{givens}(x_1, x_2)$ leads to $\mathbf{V}_i = (\mathbf{V}_i^\top)^\top;$
11: $\mathbf{B}^{(k)\top} = \mathbf{U}_i^\top \mathbf{B}^{(k)\top} \mathbf{V}_i;$ \triangleright right update
12: if $i < n-1$ then
13: $x_1 = b_{i+1,i}, x_2 = b_{i+2,i};$ $\triangleright b_{ij} = b_{ij}^{(k)\top}$
14: end if
15: end for
16: $\mathbf{U}^{(k)\top} = \mathbf{U}_{n-1}^\top \dots \mathbf{U}_1^\top;$
17: $\mathbf{V}^{(k)\top} = \mathbf{V}_1 \dots \mathbf{U}_{n-1};$ \triangleright this results in $\mathbf{B}^{(k)\top} = \mathbf{U}^{(k)\top} \mathbf{B}^{(k-1)\top} \mathbf{V}^{(k)}$
18: end for

```

---

Again, suppose for iteration  $p$ ,  $\mathbf{B}^{(p)}$  converges to a diagonal matrix (within the machine error). Then write out the updates in each iteration:

$$\left. \begin{array}{l} \mathbf{B}^{(p)\top} = \mathbf{U}^{(p)\top} \mathbf{B}^{(p-1)\top} \mathbf{V}^{(p)} \\ \mathbf{B}^{(p-1)\top} = \mathbf{U}^{(p-1)\top} \mathbf{B}^{(p-2)\top} \mathbf{V}^{(p-1)} \\ \vdots = \vdots \\ \mathbf{B}^{(1)\top} = \mathbf{U}^{(1)\top} \mathbf{B}^{(0)\top} \mathbf{V}^{(1)} \\ \mathbf{B}^{(0)\top} = \mathbf{U}^{(0)\top} \mathbf{A} \mathbf{V}^{(0)} \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{ll} \mathbf{T}^{(p)} = \mathbf{B}^{(p)\top} \mathbf{B}^{(p)} & = \mathbf{U}^{(p)\top} \mathbf{T}^{(p-1)} \mathbf{U}^{(p)} \\ \mathbf{T}^{(p-1)} = \mathbf{B}^{(p-1)\top} \mathbf{B}^{(p-1)} & = \mathbf{U}^{(p-1)\top} \mathbf{T}^{(p-2)} \mathbf{U}^{(p-1)} \\ \vdots = \vdots \\ \mathbf{T}^{(1)} = \mathbf{B}^{(1)\top} \mathbf{B}^{(1)} & = \mathbf{U}^{(1)\top} \mathbf{T}^{(0)} \mathbf{U}^{(1)} \\ \mathbf{T}^{(0)} = \mathbf{B}^{(0)\top} \mathbf{B}^{(0)} & = \mathbf{U}^{(0)\top} \mathbf{A} \mathbf{A}^\top \mathbf{U}^{(0)} \end{array} \right.$$

This yields

$$\mathbf{B}^{(p)\top} \approx \mathbf{B}^{(p)} = \underbrace{\mathbf{U}^{(p)\top} \dots \mathbf{U}^{(0)\top}}_{\mathbf{U}^\top} \mathbf{A} \underbrace{\mathbf{V}^{(0)} \mathbf{V}^{(p)}}_{\mathbf{V}} \quad \xrightarrow{\text{leads to}} \quad \begin{cases} \mathbf{A} = \mathbf{U} \mathbf{B}^{(p)} \mathbf{V}^\top \\ \mathbf{A} \mathbf{A}^\top = \mathbf{U} \mathbf{B}^{(p)} \mathbf{B}^{(p)\top} \mathbf{U}^\top \end{cases}$$

which approximates the SVD of  $\mathbf{A}$ .

### 15.9.2 Jacobi's SVD Method

We have discussed the Jacobi's method for computing the spectral decomposition of a matrix where a pair of orthogonal matrices are applied on the left and right iteratively to reduce the off-diagonal quantity. The orthogonal matrices applied on the left and right are equal (but transposed:  $\mathbf{Q}$  and  $\mathbf{Q}^\top$ ). It is then straightforward to apply the Jacobi's method by two different sequences of orthogonal matrices hoping the off-diagonal quantity can also be reduced. The problem remains

$$\mathbf{J}_1^\top \mathbf{A}(k, l) \mathbf{J}_2 = \begin{bmatrix} c_1 & -s_1 \\ s_1 & c_1 \end{bmatrix} \begin{bmatrix} a_{kk} & a_{kl} \\ a_{kl} & a_{ll} \end{bmatrix} \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix} = \begin{bmatrix} \neq 0 & 0 \\ 0 & \neq 0 \end{bmatrix}.$$

Such problem can be decomposed into two parts:

- Find a Jacobi's rotation such that  $\tilde{\mathbf{J}}_1 \mathbf{A}(k, l)$  is symmetric;
- Apply the Jacobi's method on the symmetric  $\tilde{\mathbf{J}}_1 \mathbf{A}(k, l)$ :  $\mathbf{J}_2^\top (\tilde{\mathbf{J}}_1 \mathbf{A}(k, l)) \mathbf{J}_2$  is diagonal;

Therefore, let  $\mathbf{J}_1^\top = \mathbf{J}_2^\top \tilde{\mathbf{J}}_1$ , we diagonalize the submatrix inside  $\mathbf{A}$ . We shall not discuss the details.

### 15.10. Proof of Results

**Proof** [of Lemma 15.1] It is trivial the iteration 0 of the three sequences are equal by same initialization. We will show that if  $\{\hat{\mathbf{A}}^{(k-1)} = \mathbf{A}^{(k-1)}; \hat{\mathbf{R}}^{(k-1)} = \mathbf{R}^{(k-1)}; \hat{\mathbf{V}}^{(k-1)} = \mathbf{V}^{(k-1)}\}$ , if this is also true for the  $k$ -th iteration that  $\{\hat{\mathbf{A}}^{(k)} = \mathbf{A}^{(k)}; \hat{\mathbf{R}}^{(k)} = \mathbf{R}^{(k)}; \hat{\mathbf{V}}^{(k)} = \mathbf{V}^{(k)}\}$ . Then we complete the proof by induction.

By Algorithm 46,

$$\begin{cases} \hat{\mathbf{V}}^{(k)}, \hat{\mathbf{R}}^{(k)} = QR(\mathbf{A} \hat{\mathbf{V}}^{(k-1)}) \\ \hat{\mathbf{A}}^{(k-1)} = \hat{\mathbf{V}}^{(k-1)\top} \underbrace{\mathbf{A} \hat{\mathbf{V}}^{(k-1)}}_{\hat{\mathbf{V}}^{(k)} \hat{\mathbf{R}}^{(k)}} \end{cases} \quad \xrightarrow{\text{leads to}} \quad \hat{\mathbf{V}}^{(k)} \hat{\mathbf{R}}^{(k)} = \mathbf{A} \hat{\mathbf{V}}^{(k-1)}. \quad \xrightarrow{\text{leads to}} \quad \hat{\mathbf{A}}^{(k-1)} = \underbrace{\hat{\mathbf{V}}^{(k-1)\top} \hat{\mathbf{V}}^{(k)}}_{\text{orthogonal}} \hat{\mathbf{R}}^{(k)}.$$

Since  $\hat{\mathbf{V}}^{(k-1)\top} \hat{\mathbf{V}}^{(k)}$  is a multiplication of two orthogonal matrices, it is an orthogonal matrix as well. Therefore,  $\hat{\mathbf{A}}^{(k-1)} = \hat{\mathbf{V}}^{(k-1)\top} \hat{\mathbf{V}}^{(k)} \hat{\mathbf{R}}^{(k)}$  is the QR decomposition of  $\hat{\mathbf{A}}^{(k-1)}$ .

By Algorithm 47,

$$\mathbf{Q}^{(k)}, \mathbf{R}^{(k)} = QR(\mathbf{A}^{(k-1)}) \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}^{(k-1)} = \mathbf{Q}^{(k)} \mathbf{R}^{(k)}.$$

Therefore, a second QR decomposition of  $\mathbf{A}^{(k-1)}$  is given by  $\mathbf{A}^{(k-1)} = \mathbf{Q}^{(k)} \mathbf{R}^{(k)}$ . Although, the QR decomposition is not unique, if we take the diagonal of the upper triangular matrix

to be nonnegative, the QR decomposition is unique (Corollary 3.1, p. 122). Provided  $\widehat{\mathbf{A}}^{(k-1)} = \mathbf{A}^{(k-1)}$ , this shows

$$\widehat{\mathbf{R}}^{(k)} = \mathbf{R}^{(k)}$$

and

$$\begin{aligned} \mathbf{Q}^{(k)} &= \widehat{\mathbf{V}}^{(k-1)\top} \widehat{\mathbf{V}}^{(k)} & \xrightarrow{\text{leads to}} & \widehat{\mathbf{V}}^{(k-1)} \mathbf{Q}^{(k)} = \widehat{\mathbf{V}}^{(k)} \\ && & \underbrace{\mathbf{V}^{(k-1)} \mathbf{Q}^{(k)}}_{\mathbf{V}^{(k)}} = \widehat{\mathbf{V}}^{(k)}. \end{aligned}$$

Therefore, it follows that

$$\widehat{\mathbf{V}}^{(k)} = \mathbf{V}^{(k)}.$$

To see  $\widehat{\mathbf{A}}^{(k)} = \mathbf{A}^{(k)}$ , it follows that

$$\begin{aligned} \widehat{\mathbf{A}}^{(k)} &= \widehat{\mathbf{V}}^{(k)\top} \mathbf{A} \widehat{\mathbf{V}}^{(k)} & & \text{(By Algorithm 46)} \\ &= \mathbf{V}^{(k)\top} \mathbf{A} \widehat{\mathbf{V}}^{(k)} & & (\widehat{\mathbf{V}}^{(k)} = \mathbf{V}^{(k)}) \\ &= (\mathbf{V}^{(k-1)} \mathbf{Q}^{(k)})^\top \mathbf{A} (\mathbf{V}^{(k-1)} \mathbf{Q}^{(k)}) & & \text{(By Algorithm 47)} \\ &= \mathbf{Q}^{(k)\top} \widehat{\mathbf{A}}^{(k-1)} \mathbf{Q}^{(k)} = \mathbf{Q}^{(k)\top} \mathbf{A}^{(k-1)} \mathbf{Q}^{(k)} & & \text{(By Algorithm 46)} \\ &= \mathbf{A}^{(k)}. & & \text{(By Equation (15.10))} \end{aligned}$$

This completes the proof. ■

**Proof** [of Equation (15.12)] The first equality of the LHS of Equation (15.12) comes from the step 7 of Algorithm 47. The second equation is from the fact of Algorithm 46 that

$$\left. \begin{aligned} \mathbf{A} &= \widehat{\mathbf{V}}^{(k)} \widehat{\mathbf{R}}^{(k)} \widehat{\mathbf{V}}^{(k-1)} = \mathbf{V}^{(k)} \mathbf{R}^{(k)} \mathbf{V}^{(k-1)} \\ &= \widehat{\mathbf{V}}^{(k-1)} \widehat{\mathbf{R}}^{(k-1)} \widehat{\mathbf{V}}^{(k-2)} = \mathbf{V}^{(k-1)} \mathbf{R}^{(k-1)} \mathbf{V}^{(k-2)} \\ &\quad \dots \end{aligned} \right\} \xrightarrow{\text{leads to}} \mathbf{A}^k = \mathbf{V}^{(k)} \mathbf{R}^{(k)} \mathbf{R}^{(k-1)} \dots \mathbf{R}^{(0)}.$$

This completes the proof. ■

**Proof** [of Equation (15.16)] For  $k = 1$ , it is trivial to see  $(\mathbf{A} - \mu^{(1)} \mathbf{I}) = \mathbf{Q}^{(1)} \mathbf{R}^{(1)} = \mathbf{Q}^{(0)} \mathbf{Q}^{(1)} \mathbf{R}^{(1)} \mathbf{R}^{(0)}$ . Suppose it is true for  $k - 1$  that

$$(\mathbf{A} - \mu^{(k-1)} \mathbf{I})(\mathbf{A} - \mu^{(k-2)} \mathbf{I}) \dots (\mathbf{A} - \mu^{(1)} \mathbf{I}) = \underbrace{\mathbf{Q}^{(0)} \mathbf{Q}^{(1)} \mathbf{Q}^{(2)} \dots \mathbf{Q}^{(k-1)}}_{=\mathbf{V}^{(k-1)}, \text{orthogonal}} \underbrace{\mathbf{R}^{(k-1)} \mathbf{R}^{(k-2)} \dots \mathbf{R}^{(0)}}_{=\mathbf{U}^{(k-1)}, \text{upper triangular}}.$$

If we can prove it is also true for  $k$ , then we complete the proof by induction. To see this, we have

$$\begin{aligned}
& (\mathbf{A} - \mu^{(k)} \mathbf{I})(\mathbf{A} - \mu^{(k-1)} \mathbf{I}) \dots (\mathbf{A} - \mu^{(1)} \mathbf{I}) \\
&= (\mathbf{A} - \mu^{(k)} \mathbf{I}) \mathbf{V}^{(k-1)} \left( \mathbf{R}^{(k-1)} \mathbf{R}^{(k-2)} \dots \mathbf{R}^{(0)} \right) \\
&= \left( \underbrace{\mathbf{V}^{(k)} \mathbf{A}^{(k)} \mathbf{V}^{(k)\top}}_{\text{Equation (15.15)}} - \mu^{(k)} \mathbf{I} \right) \mathbf{V}^{(k-1)} \left( \mathbf{R}^{(k-1)} \mathbf{R}^{(k-2)} \dots \mathbf{R}^{(0)} \right) \\
&= \left( \mathbf{V}^{(k)} \mathbf{A}^{(k)} \mathbf{V}^{(k)\top} - \mu^{(k)} \underbrace{\mathbf{V}^{(k)} \mathbf{V}^{(k)\top}}_{\mathbf{I}} \right) \mathbf{V}^{(k-1)} \left( \mathbf{R}^{(k-1)} \mathbf{R}^{(k-2)} \dots \mathbf{R}^{(0)} \right) \\
&= \mathbf{V}^{(k)} \left( \mathbf{A}^{(k)} - \mu^{(k)} \mathbf{I} \right) \mathbf{V}^{(k)\top} \mathbf{V}^{(k-1)} \left( \mathbf{R}^{(k-1)} \mathbf{R}^{(k-2)} \dots \mathbf{R}^{(0)} \right) \\
&= \mathbf{V}^{(k)} \left( \underbrace{\mathbf{R}^{(k)} \mathbf{Q}^{(k)}}_{\text{step 7}} \right) \left( \underbrace{\mathbf{V}^{(k-1)} \mathbf{Q}^{(k)\top}}_{\text{step 8}} \right) \mathbf{V}^{(k-1)} \left( \mathbf{R}^{(k-1)} \mathbf{R}^{(k-2)} \dots \mathbf{R}^{(0)} \right) \\
&= \mathbf{V}^{(k)} \left( \mathbf{R}^{(k)} \mathbf{R}^{(k-1)} \mathbf{R}^{(k-2)} \dots \mathbf{R}^{(0)} \right).
\end{aligned}$$

This completes the proof. ■

## **Part VI**

# **Special Topics**



## Chapter 16

# Coordinate Transformation in Matrix Decomposition

### Contents

---

|      |                                                |     |
|------|------------------------------------------------|-----|
| 16.1 | An Overview of Matrix Multiplication . . . . . | 335 |
| 16.2 | Eigenvalue Decomposition . . . . .             | 336 |
| 16.3 | Spectral Decomposition . . . . .               | 336 |
| 16.4 | SVD . . . . .                                  | 337 |
| 16.5 | Polar Decomposition . . . . .                  | 338 |

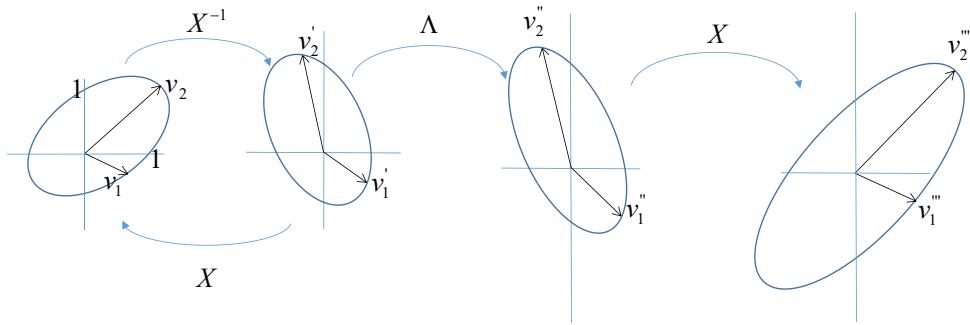
---

Suppose a vector  $\mathbf{v} \in \mathbb{R}^3$  and it has elements  $\mathbf{v} = [3; 7; 2]$ . But what do these values 3, 7, and 2 mean? In the Cartesian coordinate system, it means it has a component of 3 on the  $x$ -axis, a component of 7 on the  $y$ -axis, and a component of 2 on the  $z$ -axis.

### 16.1. An Overview of Matrix Multiplication

**Coordinate defined by a nonsingular matrix** Suppose further a  $3 \times 3$  nonsingular matrix  $\mathbf{B}$  which means  $\mathbf{B}$  is invertible and columns of  $\mathbf{B}$  are linearly independent. Thus the 3 columns of  $\mathbf{B}$  form a basis for the space  $\mathbb{R}^3$ . One step forward, we can take the 3 columns of  $\mathbf{B}$  as a basis for a new coordinate system, which we call the  $\mathbf{B}$  coordinate system. Going back to the Cartesian coordinate system, we also have three vectors as a basis,  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$ . If we put the three vectors into columns of a matrix, the matrix will be an identity matrix. So  $\mathbf{I}\mathbf{v} = \mathbf{v}$  means transfer  $\mathbf{v}$  from the Cartesian coordinate system into the  $\mathbf{B}$  coordinate system, the same coordinate. Similarly,  $\mathbf{B}\mathbf{v} = \mathbf{u}$  is to transfer  $\mathbf{v}$  from the Cartesian coordinate system into the  $\mathbf{B}$  system. Specifically, for  $\mathbf{v} = [3; 7; 2]$  and  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$ , we have  $\mathbf{u} = \mathbf{B}\mathbf{v} = 3\mathbf{b}_1 + 7\mathbf{b}_2 + 2\mathbf{b}_3$ , i.e.,  $\mathbf{u}$  contains 3 of the first basis  $\mathbf{b}_1$  of  $\mathbf{B}$ , 7 of the second basis  $\mathbf{b}_2$  of  $\mathbf{B}$ , and 2 of the third basis  $\mathbf{b}_3$  of  $\mathbf{B}$ . If again, we want to transfer the vector  $\mathbf{u}$  from  $\mathbf{B}$  coordinate system back to the Cartesian coordinate system, we just need to multiply by  $\mathbf{B}^{-1}\mathbf{u} = \mathbf{v}$ .

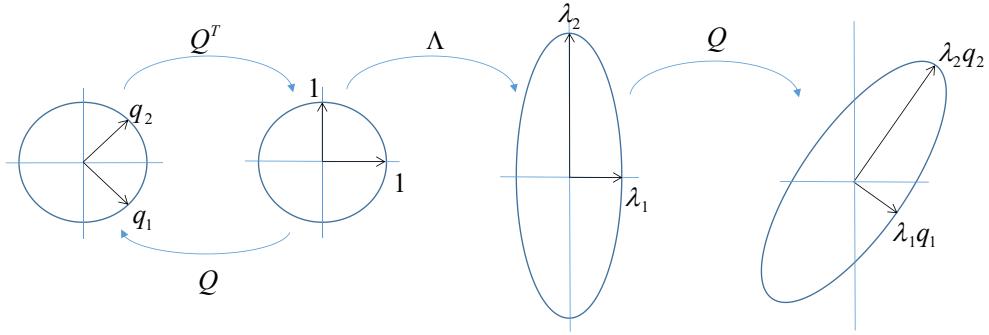
**Coordinate defined by an orthogonal matrix** A  $3 \times 3$  orthogonal matrix  $\mathbf{Q}$  defines a “better” coordinate system since the three columns (i.e., basis) are orthonormal to each other.  $\mathbf{Q}\mathbf{v}$  is to transfer  $\mathbf{v}$  from the Cartesian to the coordinate system defined by the orthogonal matrix. Since the basis vectors from the orthogonal matrix are orthonormal, just like the three vectors  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$  in the Cartesian coordinate system, the transformation defined by the orthogonal matrix just rotates or reflects the Cartesian system.  $\mathbf{Q}^\top$  can help transfer back to the Cartesian coordinate system.



**Figure 16.1:** Eigenvalue Decomposition:  $\mathbf{X}^{-1}$  transforms to a different coordinate system.  $\Lambda$  stretches and  $\mathbf{X}$  transforms back.  $\mathbf{X}^{-1}$  and  $\mathbf{X}$  are nonsingular, which will change the basis of the system, and the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  will **not** be preserved, that is, the angle between  $\mathbf{v}_1$  and  $\mathbf{v}_2$  is **different** from the angle between  $\mathbf{v}'_1$  and  $\mathbf{v}'_2$ . The length of  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are also **not** preserved, that is,  $\|\mathbf{v}_1\| \neq \|\mathbf{v}'_1\|$  and  $\|\mathbf{v}_2\| \neq \|\mathbf{v}'_2\|$ .

## 16.2. Eigenvalue Decomposition

A square matrix  $\mathbf{A}$  with linearly independent eigenvectors can be factored as  $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$  where  $\mathbf{X}$  and  $\mathbf{X}^{-1}$  are nonsingular so that they define a system transformation intrinsically.  $\mathbf{A}\mathbf{u} = \mathbf{X}\Lambda\mathbf{X}^{-1}\mathbf{u}$  firstly transfers  $\mathbf{u}$  into the system defined by  $\mathbf{X}^{-1}$ . Let's call this system the **eigen coordinate system**.  $\Lambda$  is to stretch each component of the vector in the eigen system by the length of the eigenvalue. And then  $\mathbf{X}$  helps to transfer the resulting vector back to the Cartesian coordinate system. A demonstration of how the eigenvalue decomposition transforms between coordinate systems is shown in Figure 16.1 where  $\mathbf{v}_1, \mathbf{v}_2$  are two linearly independent eigenvectors of  $\mathbf{A}$  such that they form a basis for  $\mathbb{R}^2$ .



**Figure 16.2:** Spectral Decomposition  $\mathbf{Q}\Lambda\mathbf{Q}^\top$ :  $\mathbf{Q}^\top$  rotates or reflects,  $\Lambda$  stretches cycle to ellipse, and  $\mathbf{Q}$  rotates or reflects back. Orthogonal matrices  $\mathbf{Q}^\top$  and  $\mathbf{Q}$  only change the basis of the system. However, they preserve the angle between the vectors  $\mathbf{q}_1$  and  $\mathbf{q}_2$ , and the lengths of them.

## 16.3. Spectral Decomposition

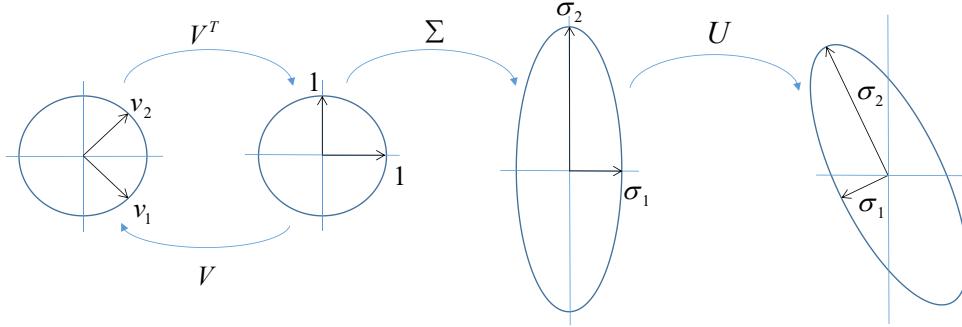
A symmetric matrix  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$  where  $\mathbf{Q}$  and  $\mathbf{Q}^\top$  are orthogonal so that they define a system transformation intrinsically.  $\mathbf{A}\mathbf{u} = \mathbf{Q}\Lambda\mathbf{Q}^\top\mathbf{u}$  firstly rotates or reflects  $\mathbf{u}$  into the system defined by  $\mathbf{Q}^\top$ . Let's call this system the **spectral coordinate system**.  $\Lambda$  is to stretch each component of the vector in the spectral system by the length of eigenvalue. And then  $\mathbf{Q}$  helps to rotate or reflect the resulting vector back to the original coordinate system. A demonstration of how the spectral decomposition transforms between coordinate systems is shown in Figure 16.2 where  $\mathbf{q}_1, \mathbf{q}_2$  are two linearly independent eigenvectors of  $\mathbf{A}$  such that they form a basis for  $\mathbb{R}^2$ . The coordinate transformation in the spectral decomposition is similar to that of the eigenvalue decomposition. Except that in the spectral decomposition, the orthogonal vectors transferred by  $\mathbf{Q}^\top$  are still orthogonal. This is also a property of orthogonal matrices. That is, orthogonal matrices can be viewed as matrices which change the basis of other matrices. Hence they preserve the angle (inner product) between the vectors

$$\mathbf{u}^\top \mathbf{v} = (\mathbf{Q}\mathbf{u})^\top (\mathbf{Q}\mathbf{v}).$$

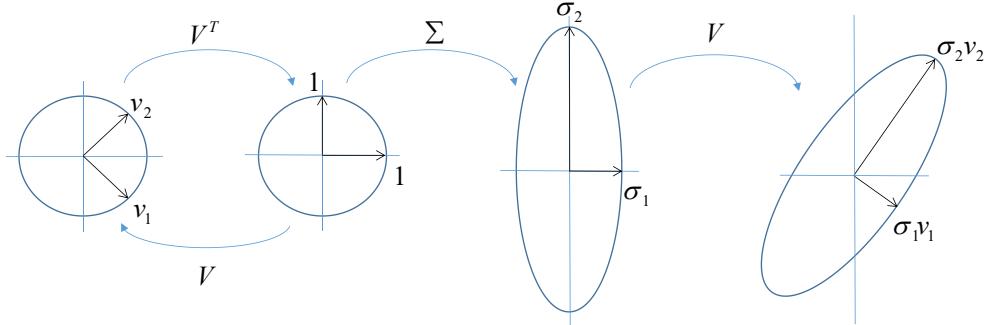
The above invariance of the inner products of angles between the vectors are preserved, which also relies on the invariance of their lengths:

$$\|Qu\| = \|u\|.$$

#### 16.4. SVD



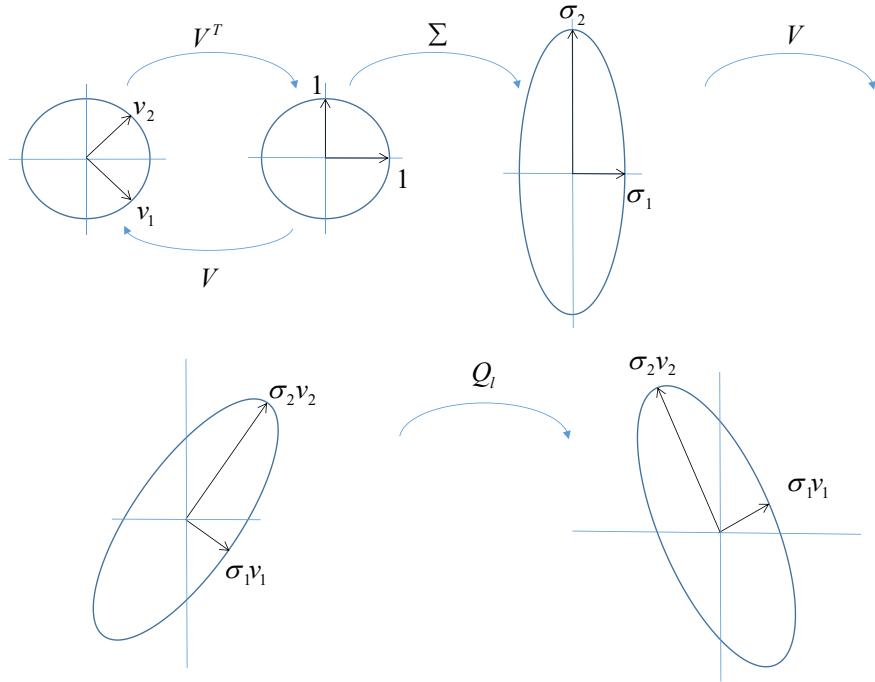
**Figure 16.3:** SVD:  $V^T$  and  $U$  rotate or reflect,  $\Sigma$  stretches the circle to an ellipse. Orthogonal matrices  $V^T$  and  $U$  only change the basis of the system. However, they preserve the angle between the vectors  $v_1$  and  $v_2$ , and the lengths of them.



**Figure 16.4:**  $V\Sigma V^T$  from SVD or Polar decomposition:  $V^T$  rotates or reflects,  $\Sigma$  stretches cycle to ellipse, and  $V$  rotates or reflects back. Orthogonal matrices  $V^T$  and  $V$  only change the basis of the system. However, they preserve the angle between the vectors  $v_1$  and  $v_2$ , and the lengths of them.

Any  $m \times n$  matrix can be factored as  $A = U\Sigma V^T$ .  $Au = U\Sigma V^T u$  then firstly rotates or reflects  $u$  into the system defined by  $V^T$ , which we call the  **$V$  coordinate system**.  $\Sigma$  stretches the first  $r$  components of the resulted vector in the  $V$  system by the lengths of the singular value. If  $n \geq m$ , then  $\Sigma$  only keeps additional  $m - r$  components which are stretched to zero while removing the final  $n - m$  components. If  $m > n$ , the  $\Sigma$  stretches  $n - r$  components to zero, and also adds additional  $m - n$  zero components. Finally,  $U$  rotates or reflects the resulting vector into the  **$U$  coordinate system** defined by  $U$ .

A demonstration of how the SVD transforms in a  $2 \times 2$  example is shown in Figure 16.3. Further, Figure 16.4 demonstrates the transformation of  $\mathbf{V}\Sigma\mathbf{V}^\top$  in a  $2 \times 2$  example. Similar to the spectral decomposition, orthogonal matrices  $\mathbf{V}^\top$  and  $\mathbf{U}$  only change the basis of the system. However, they preserve the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ .




---

**Figure 16.5:** Polar decomposition:  $\mathbf{V}^\top$  rotates or reflects,  $\Sigma$  stretches cycle to ellipse, and  $\mathbf{V}$  rotates or reflects back. Orthogonal matrices  $\mathbf{V}^\top$ ,  $\mathbf{V}$ ,  $\mathbf{Q}_l$  only change the basis of the system. However, they preserve the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , and the lengths of them.

## 16.5. Polar Decomposition

Any  $n \times n$  square matrix  $\mathbf{A}$  can be factored as left polar decomposition  $\mathbf{A} = (\mathbf{U}\mathbf{V}^\top)(\mathbf{V}\Sigma\mathbf{V}^\top) = \mathbf{Q}_l\mathbf{S}$ . Similarly,  $\mathbf{Av} = \mathbf{Q}_l(\mathbf{V}\Sigma\mathbf{V}^\top)\mathbf{u}$  is to transfer  $\mathbf{u}$  into the system defined by  $\mathbf{V}^\top$  and stretch each component by the lengths of the singular values. Then the resulted vector is transferred back into the Cartesian coordinate system by  $\mathbf{V}$ . Finally,  $\mathbf{Q}_l$  will rotate or reflect the resulting vector from the Cartesian coordinate system into the  $Q$  system defined by  $\mathbf{Q}_l$ . The meaning of right polar decomposition shares a similar description. Similar to the spectral decomposition, orthogonal matrices  $\mathbf{V}^\top$  and  $\mathbf{V}$  only change the basis of the system. However, they preserve the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ .

# Chapter 17

## Alternating Least Squares

### Contents

---

|         |                                                         |     |
|---------|---------------------------------------------------------|-----|
| 17.1    | Preliminary: Least Squares Approximations . . . . .     | 340 |
| 17.2    | Netflix Recommender and Matrix Factorization . . . . .  | 342 |
| 17.3    | Regularization: Extension to General Matrices . . . . . | 347 |
| 17.4    | Missing Entries . . . . .                               | 348 |
| 17.5    | Vector Inner Product . . . . .                          | 350 |
| 17.6    | Gradient Descent . . . . .                              | 351 |
| 17.7    | Regularization: A Geometrical Interpretation . . . . .  | 354 |
| 17.8    | Stochastic Gradient Descent . . . . .                   | 356 |
| 17.9    | Bias Term . . . . .                                     | 358 |
| 17.10   | Applications . . . . .                                  | 359 |
| 17.10.1 | Low-Rank Approximation . . . . .                        | 359 |
| 17.10.2 | Movie Recommender . . . . .                             | 361 |

---

### 17.1. Preliminary: Least Squares Approximations

The linear model is the main technique in regression problems and the primary tool for it is the least squares approximation which minimizes a sum of squared errors. This is a natural choice when we're interested in finding the regression function which minimizes the corresponding expected squared error. Over the recent decades, linear models have been used in a wide range of applications, e.g., decision making (Dawes and Corrigan, 1974), time series (Christensen, 1991; Lu, 2017), and in many fields of study, production science, social science and soil science (Fox, 1997; Lane, 2002; Schaeffer, 2004; Mrode, 2014).

Let's consider the overdetermined system  $\mathbf{b} = \mathbf{Ax}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the input data matrix,  $\mathbf{b} \in \mathbb{R}^m$  is the observation matrix (target matrix), and the sample number  $m$  is larger than the dimension number  $n$ .  $\mathbf{x}$  is a vector of weights of the linear model. Normally  $\mathbf{A}$  will have full column rank since the data from real world has large chance to be unrelated. In practice, a bias term is added to the first column of  $\mathbf{A}$  such that the least square is to find the solution of

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = [\mathbf{1}, \mathbf{A}] \begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix} = \mathbf{b}. \quad (17.1)$$

It often happens that  $\mathbf{b} = \mathbf{Ax}$  has no solution. The usual reason is: too many equations, i.e., the matrix has more rows than columns. Define the column space of  $\mathbf{A}$  by  $\{\mathbf{A}\gamma : \forall \gamma \in \mathbb{R}^n\}$  and denoted by  $\mathcal{C}(\mathbf{A})$ . Thus the meaning of  $\mathbf{b} = \mathbf{Ax}$  has no solution is that  $\mathbf{b}$  is outside the column space of  $\mathbf{A}$ . In another word, the error  $\mathbf{e} = \mathbf{b} - \mathbf{Ax}$  cannot get down to zero. When the error  $\mathbf{e}$  is as small as possible in the sense of mean squared error,  $\mathbf{x}_{LS}$  is a least squares solution, i.e.,  $\|\mathbf{b} - \mathbf{Ax}_{LS}\|^2$  is minimum.

**Least squares by calculus** When  $\|\mathbf{b} - \mathbf{Ax}\|^2$  is differentiable, and parameter space of  $\mathbf{x}$  is an open set (the least achievable value is obtained inside the parameter space), the least squares estimator must be the root of  $\|\mathbf{b} - \mathbf{Ax}\|^2$ . We thus come into the following lemma.

#### Lemma 17.1: (Least Squares by Calculus)

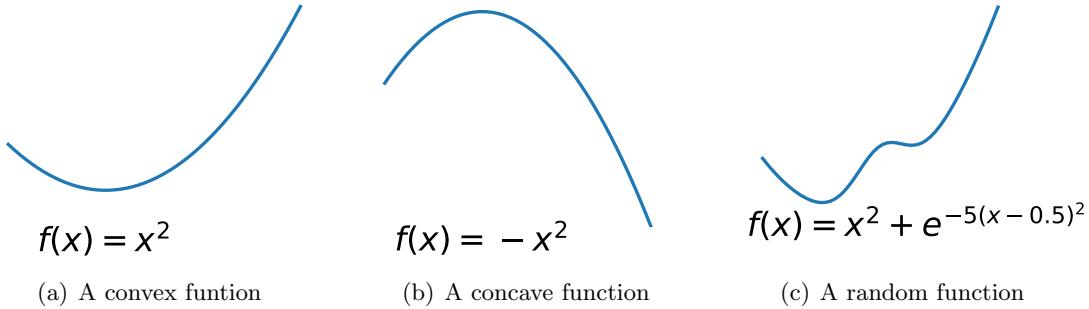
Assume  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is fixed and has full rank (i.e., the columns of  $\mathbf{A}$  are linearly independent) with  $m > n$ . Consider the overdetermined system  $\mathbf{b} = \mathbf{Ax}$ , the least squares solution by calculus via setting the derivative in every direction of  $\|\mathbf{b} - \mathbf{Ax}\|^2$  to be zero is  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ . The value  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  is known as the ordinary least squares (OLS) estimator or simply least squares (LS) estimator of  $\mathbf{x}$ .

To prove the lemma above, we must show  $\mathbf{A}^\top \mathbf{A}$  is invertible. Since we assume  $\mathbf{A}$  has full rank and  $m > n$ .  $\mathbf{A}^\top \mathbf{A} \in \mathbb{R}^{n \times n}$  is invertible if it has rank  $n$  which is the same as the rank of  $\mathbf{A}$ . This has been proved in Lemma 14.2 (p. 267). Apply the observation to  $\mathbf{A}^\top$ , we can also prove that  $\mathbf{A}\mathbf{A}^\top$  and  $\mathbf{A}$  have same rank. This result brings about the ordinary least squares estimator as follows.

**Proof** [of Lemma 17.1] Recall from calculus that a minimum of a function  $f(\mathbf{x})$  occurs at a value  $\mathbf{x}_{LS}$  such that the derivative  $\frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) = \mathbf{0}$ . The differential of  $\|\mathbf{b} - \mathbf{Ax}\|^2$  is  $2\mathbf{A}^\top \mathbf{Ax} - 2\mathbf{A}^\top \mathbf{b}$ .  $\mathbf{A}^\top \mathbf{A}$  is invertible since we assume  $\mathbf{A}$  is fixed and has full rank with  $m > n$  (Lemma 14.2, p. 267). So the OLS solution of  $\mathbf{x}$  is  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  which completes the proof. ■

**Definition 17.2: Normal Equation**

We can write the zero derivative of  $\|\mathbf{b} - \mathbf{Ax}\|^2$  as  $\mathbf{A}^\top \mathbf{Ax}_{LS} = \mathbf{A}^\top \mathbf{b}$ . The equation is also known as the *normal equation*. In the assumption,  $\mathbf{A}$  has full rank with  $m > n$ . So  $\mathbf{A}^\top \mathbf{A}$  is invertible which implies  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ .

**Figure 17.1:** Three functions.

However, we do not actually know the least squares estimator obtained in Lemma 17.1 is the least or largest achievable estimator (or neither). An example is shown in Figure 17.1. All we can get is that there only exists one root for the function  $\|\mathbf{b} - \mathbf{Ax}\|^2$ . The following remark can address this concern.

**Remark 17.3: Verification of Least Squares Solution**

Why does the zero derivative imply least mean squared error? The usual reason is from the convex analysis as we shall see shortly. But here we verify directly that the OLS solution finds the least squares. For any  $\beta \neq \hat{\beta}$ , we have

$$\begin{aligned} \|\mathbf{b} - \mathbf{Ax}\|^2 &= \|\mathbf{b} - \mathbf{Ax}_{LS} + \mathbf{Ax}_{LS} - \mathbf{Ax}\|^2 \\ &= \|\mathbf{b} - \mathbf{Ax}_{LS} + \mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|^2 \\ &= \|\mathbf{b} - \mathbf{Ax}_{LS}\|^2 + \|\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|^2 + 2(\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x}))^\top (\mathbf{b} - \mathbf{Ax}_{LS}) \\ &= \|\mathbf{b} - \mathbf{Ax}_{LS}\|^2 + \|\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|^2 + 2(\mathbf{x}_{LS} - \mathbf{x})^\top (\mathbf{A}^\top \mathbf{b} - \mathbf{A}^\top \mathbf{Ax}_{LS}), \end{aligned}$$

where the third term is zero from the normal equation and  $\|\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|^2 \geq 0$ . Therefore,

$$\|\mathbf{b} - \mathbf{Ax}\|^2 \geq \|\mathbf{b} - \mathbf{Ax}_{LS}\|^2.$$

Thus we show that the OLS estimator indeed gives the minimum, not the maximum or a saddle point via the calculus approach.

Further question would be posed: why does this normal equation magically produce solution for  $\mathbf{x}$ ? A simple example would give the answer.  $x^2 = -1$  has no real solution.

But  $x \cdot x^2 = x \cdot (-1)$  has a real solution  $\hat{x} = 0$  in which case  $\hat{x}$  makes  $x^2$  and  $-1$  as close as possible.

**Example 17.1 (Multiply From Left Can Change The Solution Set )** Consider the matrix

$$\mathbf{A} = \begin{bmatrix} -3 & -4 \\ 4 & 6 \\ 1 & 1 \end{bmatrix} \text{ and } \mathbf{b} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}.$$

It can be easily verified that  $\mathbf{Ax} = \mathbf{b}$  has no solution for  $\mathbf{x}$ . However, if we multiply on the left by

$$\mathbf{B} = \begin{bmatrix} 0 & -1 & 6 \\ 0 & 1 & -4 \end{bmatrix}.$$

Then we have  $\mathbf{x}_{LS} = [1/2, -1/2]^\top$  as the solution of  $\mathbf{BAx} = \mathbf{Bb}$ . This specific example shows why the normal equation can give rise to the least square solution. Multiply on the left of a linear system will change the solution set.  $\square$

### Rank Deficiency

Note here, we assume  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has full rank with  $m > n$  to make  $\mathbf{A}^\top \mathbf{A}$  invertible. But when two or more columns of  $\mathbf{A}$  are perfectly correlated, the matrix  $\mathbf{A}$  will be deficient and  $\mathbf{A}^\top \mathbf{A}$  is singular. Choose the  $\mathbf{x}$  that minimizes  $\mathbf{x}_{LS}^\top \mathbf{x}_{LS}$  which meets the normal equation can help to solve the problem. I.e., choose the shortest magnitude least squares solution. But this is not the main interest of the text. We will leave this topic to the readers. In Section 4.3.1, 14.7.1, we have shortly discussed how to use UTV decomposition and SVD to tackle the rank deficient least squares problem.

## 17.2. Netflix Recommender and Matrix Factorization

In the Netflix prize (Bennett et al., 2007), the goal was to predict ratings of users for different movies, given the existing ratings of those users for other movies. We index  $M$  movies with  $m = 1, 2, \dots, M$  and  $N$  users by  $n = 1, 2, \dots, N$ . We denote the rating of the  $n$ -th user for the  $m$ -th movie by  $a_{mn}$ . Define  $\mathbf{A}$  to be an  $M \times N$  rating matrix with columns  $\mathbf{a}_n \in \mathbb{R}^M$  containing ratings of the  $n$ -th user. Note that many ratings  $a_{mn}$  are missing and our goal is to predict those missing ratings accurately.

We formally consider algorithms for solving the following problem: The matrix  $\mathbf{A}$  is approximately factorized into an  $M \times K$  matrix  $\mathbf{W}$  and a  $K \times N$  matrix  $\mathbf{Z}$ . Usually  $K$  is chosen to be smaller than  $M$  or  $N$ , so that  $\mathbf{W}$  and  $\mathbf{Z}$  are smaller than the original matrix  $\mathbf{A}$ . This results in a compressed version of the original data matrix. An appropriate decision on the value of  $K$  is critical in practice, but the choice of  $K$  is very often problem dependent. The factorization is significant in the sense, suppose  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]$  and  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$  are the column partitions of  $\mathbf{A}, \mathbf{Z}$  respectively, then  $\mathbf{a}_n = \mathbf{W}\mathbf{z}_n$ , i.e., each column  $\mathbf{a}_n$  is approximated by a linear combination of the columns of  $\mathbf{W}$  weighted by the components in  $\mathbf{z}_n$ . Therefore, columns of  $\mathbf{W}$  can be thought of containing column basis of  $\mathbf{A}$ . This is similar to the factorization in the data interpretation part (Part III, p. 152). What's different is that we are not restricting  $\mathbf{W}$  to be exact columns from  $\mathbf{A}$ .

To find the approximation  $\mathbf{A} \approx \mathbf{WZ}$ , we need to define a loss function such that the distance between  $\mathbf{A}$  and  $\mathbf{WZ}$  can be measured. The loss function is selected to be the Frobenius norm between two matrices which vanishes to zero if  $\mathbf{A} = \mathbf{WZ}$  where the advantage will be seen shortly.

To simplify the problem, let us assume that there are no missing ratings firstly. Project data vectors  $\mathbf{a}_n$  to a smaller dimension  $\mathbf{z}_n \in \mathbb{R}^K$  with  $K < M$ , such that the *reconstruction error* measured by Frobenius norm is minimized (assume  $K$  is known):

$$\min_{\mathbf{W}, \mathbf{Z}} \sum_{n=1}^N \sum_{m=1}^M (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2, \quad (17.2)$$

where  $\mathbf{W} = [\mathbf{w}_1^\top; \mathbf{w}_2^\top; \dots; \mathbf{w}_M^\top] \in \mathbb{R}^{M \times K}$  and  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N] \in \mathbb{R}^{K \times N}$  containing  $\mathbf{w}_m$ 's and  $\mathbf{z}_n$ 's as rows and columns respectively. The loss form in Equation (17.2) is known as the *per-example loss*. It can be equivalently written as

$$L(\mathbf{W}, \mathbf{Z}) = \sum_{n=1}^N \sum_{m=1}^M (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2 = \|\mathbf{WZ} - \mathbf{A}\|^2.$$

Moreover, the loss  $L(\mathbf{W}, \mathbf{Z}) = \sum_{n=1}^N \sum_{m=1}^M (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)$  is convex with respect to  $\mathbf{Z}$  given  $\mathbf{W}$  and vice versa. Therefore, we can first minimize with respect to  $\mathbf{Z}$  given  $\mathbf{W}$  and then minimize with respect to  $\mathbf{W}$  given  $\mathbf{Z}$ :

$$\begin{cases} \mathbf{Z} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}); & (\text{ALS1}) \\ \mathbf{W} \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W}, \mathbf{Z}). & (\text{ALS2}) \end{cases}$$

This is known as the *coordinate descent algorithm* in which case we employ the least squares, it is also called the *alternating least squares (ALS)* (Comon et al., 2009; Takács and Tikk, 2012; Giampouras et al., 2018). The convergence is guaranteed if the loss function  $L(\mathbf{W}, \mathbf{Z})$  decreases at each iteration and we shall discuss more on this in the sequel.

### Remark 17.1: Convexity and Global Minimum

Although the loss function defined by Frobenius norm  $\|\mathbf{WZ} - \mathbf{A}\|^2$  is convex in  $\mathbf{W}$  given  $\mathbf{Z}$  or vice versa, it is not convex in both variables together. Therefore we are not able to find the global minimum. However, the convergence is assured to find local minima.

**Given  $\mathbf{W}$ , Optimizing  $\mathbf{Z}$**  Now, let's see what is in the problem of  $\mathbf{Z} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z})$ . When there exists a unique minimum of the loss function  $L(\mathbf{W}, \mathbf{Z})$  with respect to  $\mathbf{Z}$ , we speak of the *least squares* minimizer of  $\arg \min_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z})$ . Given  $\mathbf{W}$ ,  $L(\mathbf{W}, \mathbf{Z})$  can be

written as  $L(\mathbf{Z}|\mathbf{W})$  to emphasize on the variable of  $\mathbf{Z}$ :

$$L(\mathbf{Z}|\mathbf{W}) = \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|^2 = \|\mathbf{W}[\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N] - [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]\|^2 = \left\| \begin{bmatrix} \mathbf{W}\mathbf{z}_1 - \mathbf{a}_1 \\ \mathbf{W}\mathbf{z}_2 - \mathbf{a}_2 \\ \vdots \\ \mathbf{W}\mathbf{z}_N - \mathbf{a}_N \end{bmatrix} \right\|^2. \quad 1$$

Now, if we define

$$\widetilde{\mathbf{W}} = \begin{bmatrix} \mathbf{W} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{W} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{W} \end{bmatrix} \in \mathbb{R}^{MN \times KN}, \quad \widetilde{\mathbf{z}} = \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_N \end{bmatrix} \in \mathbb{R}^{KN}, \quad \widetilde{\mathbf{a}} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_N \end{bmatrix} \in \mathbb{R}^{MN},$$

then the (ALS1) problem can be reduced to the normal least squares for minimizing  $\|\widetilde{\mathbf{W}}\widetilde{\mathbf{z}} - \widetilde{\mathbf{a}}\|^2$  with respect to  $\widetilde{\mathbf{z}}$ . And the solution is given by

$$\widetilde{\mathbf{z}} = (\widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}})^{-1} \widetilde{\mathbf{W}}^\top \widetilde{\mathbf{a}}.$$

The construction may seem reasonable at first glance. But since  $\text{rank}(\widetilde{\mathbf{W}}) = \min\{M, K\}$ ,  $(\widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}})$  is not invertible. A direct way to solve (ALS1) is to find the differential of  $L(\mathbf{Z}|\mathbf{W})$  with respect to  $\mathbf{Z}$ :

$$\begin{aligned} \frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} &= \frac{\partial \text{tr}((\mathbf{W}\mathbf{Z} - \mathbf{A})(\mathbf{W}\mathbf{Z} - \mathbf{A})^\top)}{\partial \mathbf{Z}} \\ &= \frac{\partial \text{tr}((\mathbf{W}\mathbf{Z} - \mathbf{A})(\mathbf{W}\mathbf{Z} - \mathbf{A})^\top)}{\partial(\mathbf{W}\mathbf{Z} - \mathbf{A})} \frac{\partial(\mathbf{W}\mathbf{Z} - \mathbf{A})}{\partial \mathbf{Z}} \\ &\stackrel{*}{=} 2\mathbf{W}^\top(\mathbf{W}\mathbf{Z} - \mathbf{A}) \in \mathbb{R}^{K \times N}, \end{aligned} \quad (17.3)$$

where the first equality is from the definition of Frobenius (Definition 27.5, p. 480) such that  $\|\mathbf{A}\| = \sqrt{\sum_{i=1,j=1}^{m,n} (\mathbf{A}_{ij})^2} = \sqrt{\text{tr}(\mathbf{A}\mathbf{A}^\top)}$ , and equality  $(*)$  comes from the fact that  $\frac{\partial \text{tr}(\mathbf{A}\mathbf{A}^\top)}{\partial \mathbf{A}} = 2\mathbf{A}$ . When the loss function is a differentiable function of  $\mathbf{Z}$ , we may determine the least squares solution by differential calculus, and a minimum of the function  $L(\mathbf{Z}|\mathbf{W})$  must be a root of the equation:

$$\frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} = \mathbf{0}.$$

By finding the root of the above equation, we have the “candidate” update on  $\mathbf{Z}$  that find the minimizer of  $L(\mathbf{Z}|\mathbf{W})$

$$\mathbf{Z} = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{A} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{Z}|\mathbf{W}). \quad (17.4)$$

---

1. The matrix norm used here is the Frobenius norm (Definition 27.5, p. 480) such that  $\|\mathbf{A}\| = \sqrt{\sum_{i=1,j=1}^{m,n} (\mathbf{A}_{ij})^2}$  if  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . And the vector norm used here is the  $l_2$  norm (Section L.1, p. 476) such that  $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$  if  $\mathbf{x} \in \mathbb{R}^n$ .

Before we declare a root of the above equation is actually a minimizer rather than a maximizer (that's why we call the update a "candidate" update above), we need to verify the function is convex such that if the function is twice differentiable, this can be equivalently done by verifying

$$\frac{\partial^2 L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}^2} > 0,$$

i.e., the Hessian matrix is positive definite (recall the definition of positive definiteness, Definition 2.1, p. 56). To see this, we write out the twice differential

$$\frac{\partial^2 L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}^2} = 2\mathbf{W}^\top \mathbf{W} \in \mathbb{R}^{K \times K},$$

which has full rank if  $\mathbf{W} \in \mathbb{R}^{M \times K}$  has full rank (Lemma 14.2, p. 267) and  $K < M$ . We here claim that if  $\mathbf{W}$  has full rank, then  $\frac{\partial^2 L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}^2}$  is positive definite. This can be done by checking that when  $\mathbf{W}$  has full rank,  $\mathbf{W}\mathbf{x} = \mathbf{0}$  only when  $\mathbf{x} = \mathbf{0}$  since the null space of  $\mathbf{W}$  is of dimension 0. Therefore,

$$\mathbf{x}^\top (2\mathbf{W}^\top \mathbf{W})\mathbf{x} > 0, \quad \text{for any nonzero vector } \mathbf{x} \in \mathbb{R}^K.$$

Now, the thing is that we need to check if  $\mathbf{W}$  has full rank so that the Hessian of  $L(\mathbf{Z}|\mathbf{W})$  is positive definiteness, otherwise, we cannot claim the update of  $\mathbf{Z}$  in Equation (17.4) decreases the loss so that the matrix decomposition is going into the right way to better approximate the original matrix  $\mathbf{A}$  by  $\mathbf{W}\mathbf{Z}$ . We will shortly come back to the positive definiteness of the Hessian matrix in the sequel which relies on the following lemma

### Lemma 17.2: (Rank of $\mathbf{Z}$ after Updating)

Suppose  $\mathbf{A} \in \mathbb{R}^{M \times N}$  has full rank with  $M \leq N$  and  $\mathbf{W} \in \mathbb{R}^{M \times K}$  has full rank with  $K < M$ , then the update of  $\mathbf{Z} = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{A} \in \mathbb{R}^{K \times N}$  in Equation (17.4) has full rank.

**Proof** [of Lemma 17.2] Since  $\mathbf{W}^\top \mathbf{W} \in \mathbb{R}^{K \times K}$  has full rank if  $\mathbf{W}$  has full rank (Lemma 14.2, p. 267) such that  $(\mathbf{W}^\top \mathbf{W})^{-1}$  has full rank.

Suppose  $\mathbf{W}^\top \mathbf{x} = \mathbf{0}$ , this implies  $(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{x} = \mathbf{0}$ . Thus

$$\mathcal{N}(\mathbf{W}^\top) \subseteq \mathcal{N}\left((\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top\right).$$

Moreover, suppose  $(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{x} = \mathbf{0}$ , and since  $(\mathbf{W}^\top \mathbf{W})^{-1}$  is invertible. This implies  $\mathbf{W}^\top \mathbf{x} = (\mathbf{W}^\top \mathbf{W})\mathbf{0} = \mathbf{0}$ , and

$$\mathcal{N}\left((\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top\right) \subseteq \mathcal{N}(\mathbf{W}^\top).$$

As a result, by "sandwiching", it follows that

$$\mathcal{N}(\mathbf{W}^\top) = \mathcal{N}\left((\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top\right). \tag{17.5}$$

Therefore,  $(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top$  has full rank  $K$ . Let  $\mathbf{T} = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \in \mathbb{R}^{K \times M}$ , and suppose  $\mathbf{T}^\top \mathbf{x} = \mathbf{0}$ . This implies  $\mathbf{A}^\top \mathbf{T}^\top \mathbf{x} = \mathbf{0}$ , and

$$\mathcal{N}(\mathbf{T}^\top) \subseteq \mathcal{N}(\mathbf{A}^\top \mathbf{T}^\top).$$

Similarly, suppose  $\mathbf{A}^\top (\mathbf{T}^\top \mathbf{x}) = \mathbf{0}$ . Since  $\mathbf{A}$  has full rank with the dimension of the null space being 0:  $\dim(\mathcal{N}(\mathbf{A}^\top)) = 0$ ,  $(\mathbf{T}^\top \mathbf{x})$  must be zero. The claim follows from that since  $\mathbf{A}$  has full rank  $M$  with the row space of  $\mathbf{A}^\top$  being equal to the column space of  $\mathbf{A}$  where  $\dim(\mathcal{C}(\mathbf{A})) = M$  and the  $\dim(\mathcal{N}(\mathbf{A}^\top)) = M - \dim(\mathcal{C}(\mathbf{A})) = 0$ . Therefore,  $\mathbf{x}$  is in the null space of  $\mathbf{T}^\top$  if  $\mathbf{x}$  is in the null space of  $\mathbf{A}^\top \mathbf{T}^\top$ :

$$\mathcal{N}(\mathbf{A}^\top \mathbf{T}^\top) \subseteq \mathcal{N}(\mathbf{T}^\top).$$

By “sandwiching” again,

$$\mathcal{N}(\mathbf{T}^\top) = \mathcal{N}(\mathbf{A}^\top \mathbf{T}^\top). \quad (17.6)$$

Since  $\mathbf{T}^\top$  has full rank  $K < M < N$ ,  $\dim(\mathcal{N}(\mathbf{T}^\top)) = \dim(\mathcal{N}(\mathbf{A}^\top \mathbf{T}^\top)) = 0$ . Therefore,  $\mathbf{Z}^\top = \mathbf{A}^\top \mathbf{T}^\top$  has full rank  $K$ . We complete the proof. ■

**Given  $\mathbf{Z}$ , Optimizing  $\mathbf{W}$**  Given  $\mathbf{Z}$ ,  $L(\mathbf{W}, \mathbf{Z})$  can be written as  $L(\mathbf{W}|\mathbf{Z})$  to emphasize on the variable of  $\mathbf{W}$ :

$$L(\mathbf{W}|\mathbf{Z}) = \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|^2.$$

A direct way to solve (ALS2) is to find the differential of  $L(\mathbf{W}|\mathbf{Z})$  with respect to  $\mathbf{W}$ :

$$\begin{aligned} \frac{\partial L(\mathbf{W}|\mathbf{Z})}{\partial \mathbf{W}} &= \frac{\partial \text{tr}((\mathbf{W}\mathbf{Z} - \mathbf{A})(\mathbf{W}\mathbf{Z} - \mathbf{A})^\top)}{\partial \mathbf{W}} \\ &= \frac{\partial \text{tr}((\mathbf{W}\mathbf{Z} - \mathbf{A})(\mathbf{W}\mathbf{Z} - \mathbf{A})^\top)}{\partial(\mathbf{W}\mathbf{Z} - \mathbf{A})} \frac{\partial(\mathbf{W}\mathbf{Z} - \mathbf{A})}{\partial \mathbf{W}} \\ &= 2(\mathbf{W}\mathbf{Z} - \mathbf{A})\mathbf{Z}^\top \in \mathbb{R}^{M \times K}. \end{aligned}$$

The “candidate” update on  $\mathbf{W}$  is similarly to find the root of the differential  $\frac{\partial L(\mathbf{W}|\mathbf{Z})}{\partial \mathbf{W}}$ .

$$\mathbf{W}^\top = (\mathbf{Z}\mathbf{Z}^\top)^{-1} \mathbf{Z}\mathbf{A}^\top \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W}|\mathbf{Z}). \quad (17.7)$$

Again, we emphasize that the update is only a “candidate” update. We need to further check whether the Hessian is positive definite or not. The Hessian matrix is given by

$$\frac{\partial^2 L(\mathbf{W}|\mathbf{Z})}{\partial \mathbf{W}^2} = 2\mathbf{Z}\mathbf{Z}^\top \in \mathbb{R}^{K \times K}.$$

Therefore, by analogous analysis, if  $\mathbf{Z}$  has full rank with  $K < N$ , the Hessian matrix is positive definite.

**Lemma 17.3: (Rank of  $\mathbf{W}$  after Updating)**

Suppose  $\mathbf{A} \in \mathbb{R}^{M \times N}$  has full rank with  $M \leq N$  and  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  has full rank with  $K < N$ , then the update of  $\mathbf{W}^\top = (\mathbf{Z}\mathbf{Z}^\top)^{-1}\mathbf{Z}\mathbf{A}^\top$  in Equation (17.7) has full rank.

**Proof** [of Lemma 17.3] The proof is slightly different to that of Lemma 17.2. Since  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  and  $\mathbf{A}^\top \in \mathbb{R}^{N \times M}$  have full rank, i.e.,  $\det(\mathbf{Z}) > 0$  and  $\det(\mathbf{A}^\top) > 0$ . The determinant of their product  $\det(\mathbf{Z}\mathbf{A}^\top) = \det(\mathbf{Z})\det(\mathbf{A}^\top) > 0$  such that  $\mathbf{Z}\mathbf{A}^\top$  has full rank (rank  $K$ ). Similarly argument can find  $\mathbf{W}^\top$  also has full rank. ■

Combine the observations in Lemma 17.2 and Lemma 17.3, as long as we initialize  $\mathbf{Z}, \mathbf{W}$  to have full rank, the updates in Equation (17.4) and Equation (17.7) are reasonable. The requirement on the  $\leq N$  is reasonable in that there are always more users than the number of movies. We conclude the process in Algorithm 56.

**Algorithm 56** Alternating Least Squares

**Require:**  $\mathbf{A} \in \mathbb{R}^{M \times N}$  with  $M \leq N$ ;

- 1: initialize  $\mathbf{W} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  with full rank and  $K < M \leq N$ ;
- 2: choose a stop criterion on the approximation error  $\delta$ ;
- 3: choose maximal number of iterations  $C$ ;
- 4:  $iter = 0$ ;
- 5: **while**  $\|\mathbf{A} - \mathbf{W}\mathbf{Z}\| > \delta$  and  $iter < C$  **do**
- 6:      $iter = iter + 1$ ;
- 7:      $\mathbf{Z} = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{A} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{Z} | \mathbf{W})$ ;
- 8:      $\mathbf{W}^\top = (\mathbf{Z}\mathbf{Z}^\top)^{-1} \mathbf{Z}\mathbf{A}^\top \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W} | \mathbf{Z})$ ;
- 9: **end while**
- 10: Output  $\mathbf{W}, \mathbf{Z}$ ;

**17.3. Regularization: Extension to General Matrices**

We can add a regularization to minimize the following loss:

$$L(\mathbf{W}, \mathbf{Z}) = \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|^2 + \lambda_w \|\mathbf{W}\|^2 + \lambda_z \|\mathbf{Z}\|^2, \quad \lambda_w > 0, \lambda_z > 0, \quad (17.8)$$

where the differential with respect to  $\mathbf{Z}, \mathbf{W}$  are given respectively by

$$\begin{cases} \frac{\partial L(\mathbf{W}, \mathbf{Z})}{\partial \mathbf{Z}} = 2\mathbf{W}^\top(\mathbf{W}\mathbf{Z} - \mathbf{A}) + 2\lambda_z \mathbf{Z} \in \mathbb{R}^{K \times N}; \\ \frac{\partial L(\mathbf{W}, \mathbf{Z})}{\partial \mathbf{W}} = 2(\mathbf{W}\mathbf{Z} - \mathbf{A})\mathbf{Z}^\top + 2\lambda_w \mathbf{W} \in \mathbb{R}^{M \times K}. \end{cases} \quad (17.9)$$

The Hessian matrices are given respectively by

$$\begin{cases} \frac{\partial^2 L(\mathbf{W}, \mathbf{Z})}{\partial \mathbf{Z}^2} = 2\mathbf{W}^\top \mathbf{W} + 2\lambda_z \mathbf{I} \in \mathbb{R}^{K \times K}; \\ \frac{\partial^2 L(\mathbf{W}, \mathbf{Z})}{\partial \mathbf{W}^2} = 2\mathbf{Z}\mathbf{Z}^\top + 2\lambda_w \mathbf{I} \in \mathbb{R}^{M \times M}, \end{cases}$$

which are positive definite due to the perturbation by the regularization. To see this,

$$\begin{cases} \mathbf{x}^\top (2\mathbf{W}^\top \mathbf{W} + 2\lambda_z \mathbf{I}) \mathbf{x} = \underbrace{2\mathbf{x}^\top \mathbf{W}^\top \mathbf{W} \mathbf{x}}_{\geq 0} + 2\lambda_z \|\mathbf{x}\|^2 > 0, & \text{for nonzero } \mathbf{x}; \\ \mathbf{x}^\top (2\mathbf{Z} \mathbf{Z}^\top + 2\lambda_w \mathbf{I}) \mathbf{x} = \underbrace{2\mathbf{x}^\top \mathbf{Z} \mathbf{Z}^\top \mathbf{x}}_{\geq 0} + 2\lambda_w \|\mathbf{x}\|^2 > 0, & \text{for nonzero } \mathbf{x}. \end{cases}$$

The regularization makes the Hessian matrices positive definite even if  $\mathbf{W}, \mathbf{Z}$  are rank deficient. And now the matrix decomposition can be extended to any matrix even when  $M > N$ . In rare cases,  $K$  can be chosen as  $K > \max\{M, N\}$  such that a high-rank approximation of  $\mathbf{A}$  is obtained. However, in most scenarios, we want to find the low-rank approximation of  $\mathbf{A}$  such that  $K < \min\{M, N\}$ . For example, the ALS can be utilized to find the low-rank neural networks to reduce the memory of the neural networks whilst increase the performance (Section 20.1, p. 383).

Therefore, the minimizers are given by finding the roots of the differential:

$$\begin{cases} \mathbf{Z} = (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{A}; \\ \mathbf{W}^\top = (\mathbf{Z} \mathbf{Z}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z} \mathbf{A}^\top. \end{cases} \quad (17.10)$$

The regularization parameters  $\lambda_z, \lambda_w \in \mathbb{R}$  are used to balance the trade-off between the accuracy of the approximation and the smoothness of the computed solution. The choice on the selection of the parameters is typically problem dependent and can be obtained by *cross-validation*. Again, we conclude the process in Algorithm 57.

---

**Algorithm 57** Alternating Least Squares with Regularization

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 1: initialize  $\mathbf{W} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  randomly without condition on the rank and the relationship between  $M, N, K$ ;
  - 2: choose a stop criterion on the approximation error  $\delta$ ;
  - 3: choose regularization parameters  $\lambda_w, \lambda_z$ ;
  - 4: choose maximal number of iterations  $C$ ;
  - 5:  $iter = 0$ ;
  - 6: **while**  $\|\mathbf{A} - \mathbf{WZ}\| > \delta$  and  $iter < C$  **do**
  - 7:      $iter = iter + 1$ ;
  - 8:      $\mathbf{Z} = (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{A} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{Z} | \mathbf{W})$ ;
  - 9:      $\mathbf{W}^\top = (\mathbf{Z} \mathbf{Z}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z} \mathbf{A}^\top \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W} | \mathbf{Z})$ ;
  - 10: **end while**
  - 11: Output  $\mathbf{W}, \mathbf{Z}$ ;
- 

## 17.4. Missing Entries

Since the matrix decomposition via the ALS is extensively used in the Netflix recommender data, where many entries are missing since many users have not watched some movies or they will not rate the movies for some reasons. We can make an additional mask matrix  $\mathbf{M} \in \mathbb{R}^{M \times N}$  where  $\mathbf{M}_{mn} \in \{0, 1\}$  means if the user  $n$  has rated the movie  $m$  or not.

Therefore, the loss function can be defined as

$$L(\mathbf{W}, \mathbf{Z}) = \|\mathbf{M} \circledast \mathbf{A} - \mathbf{M} \circledast (\mathbf{W}\mathbf{Z})\|^2,$$

where  $\circledast$  is the *Hadamard product* between matrices. For example, the Hadamard product for a  $3 \times 3$  matrix  $\mathbf{A}$  with a  $3 \times 3$  matrix  $\mathbf{B}$  is

$$\mathbf{A} \circledast \mathbf{B} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \circledast \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & a_{13}b_{13} \\ a_{21}b_{21} & a_{22}b_{22} & a_{23}b_{23} \\ a_{31}b_{31} & a_{32}b_{32} & a_{33}b_{33} \end{bmatrix}.$$

To find the solution of the problem, let's decompose the updates in Equation (17.10) into:

$$\begin{cases} \mathbf{z}_n = (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{a}_n, & \text{for } n \in \{1, 2, \dots, N\}; \\ \mathbf{w}_m = (\mathbf{Z} \mathbf{Z}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z} \mathbf{b}_m, & \text{for } m \in \{1, 2, \dots, M\}, \end{cases} \quad (17.11)$$

where  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ ,  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]$  are the column partitions of  $\mathbf{Z}$ ,  $\mathbf{A}$  respectively. And  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]$ ,  $\mathbf{A}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M]$  are the column partitions of  $\mathbf{W}^\top$ ,  $\mathbf{A}^\top$  respectively. The factorization of the updates indicates the update can be done via a column by column fashion.

**Given  $\mathbf{W}$**  Let  $\mathbf{o}_n \in \mathbb{R}^M$  denote the movies rated by user  $n$  where  $o_{nm} = 1$  if user  $n$  has rated movie  $m$ , and  $o_{nm} = 0$  otherwise. Then the  $n$ -th column of  $\mathbf{A}$  without missing entries can be denoted as the matlab style notation  $\mathbf{a}_n[\mathbf{o}_n]$ . And we want to approximate the existing  $n$ -th column by  $\mathbf{a}_n[\mathbf{o}_n] \approx \mathbf{W}[\mathbf{o}_n, :] \mathbf{z}_n$  which is actually a rank-one least squares problem:

$$\mathbf{z}_n = \left( \mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{W}[\mathbf{o}_n, :] + \lambda_z \mathbf{I} \right)^{-1} \mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{a}_n[\mathbf{o}_n], \quad \text{for } n \in \{1, 2, \dots, N\}. \quad (17.12)$$

Moreover, the loss function with respect to  $\mathbf{z}_n$ :

$$L(\mathbf{z}_n | \mathbf{W}) = \sum_{m \in \mathbf{o}_n} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2$$

and if we are concerned about the loss for all users:

$$L(\mathbf{Z} | \mathbf{W}) = \sum_{n=1}^N \sum_{m \in \mathbf{o}_n} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2$$

**Given  $\mathbf{Z}$**  Similarly, if  $\mathbf{p}_m \in \mathbb{R}^N$  denotes the users that have rated the movie  $m$  with  $p_{dn} = 1$  if the movie  $m$  has been rated by user  $n$ . Then the  $m$ -th row of  $\mathbf{A}$  without missing entries can be denoted as the matlab style notation  $\mathbf{b}_m[\mathbf{p}_m]$ . And we want to approximate the existing  $m$ -th row by  $\mathbf{b}_m[\mathbf{p}_m] \approx \mathbf{Z}[:, \mathbf{p}_m]^\top \mathbf{w}_m$ , <sup>2</sup> which again is a rank-one least squares problem:

$$\mathbf{w}_m = (\mathbf{Z}[:, \mathbf{p}_m] \mathbf{Z}[:, \mathbf{p}_m]^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z}[:, \mathbf{p}_m] \mathbf{b}_m[\mathbf{p}_m], \quad \text{for } m \in \{1, 2, \dots, M\}. \quad (17.13)$$

---

<sup>2</sup>. Note that  $\mathbf{Z}[:, \mathbf{p}_m]^\top$  is the transpose of  $\mathbf{Z}[:, \mathbf{p}_m]$ , which is equal to  $\mathbf{Z}^\top[\mathbf{p}_m, :]$ , i.e., transposing first and then selecting.

Moreover, the loss function with respect to  $\mathbf{w}_n$ :

$$L(\mathbf{w}_n | \mathbf{Z}) = \sum_{n \in \mathbf{p}_m} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2$$

and if we are concerned about the loss for all users:

$$L(\mathbf{W} | \mathbf{Z}) = \sum_{d=1}^M \sum_{n \in \mathbf{p}_m} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2$$

The procedure is again formulated in Algorithm 58.

---

**Algorithm 58** Alternating Least Squares with Missing Entries and Regularization

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 1: initialize  $\mathbf{W} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  randomly without condition on the rank and the relationship between  $M, N, K$ ;
  - 2: choose a stop criterion on the approximation error  $\delta$ ;
  - 3: choose regularization parameters  $\lambda_w, \lambda_z$ ;
  - 4: compute the mask matrix  $\mathbf{M}$  from  $\mathbf{A}$ ;
  - 5: choose maximal number of iterations  $C$ ;
  - 6:  $iter = 0$ ;
  - 7: **while**  $\|\mathbf{M} \circledast \mathbf{A} - \mathbf{M} \circledast (\mathbf{W} \mathbf{Z})\|^2 > \delta$  and  $iter < C$  **do**
  - 8:    $iter = iter + 1$ ;
  - 9:   **for**  $n = 1, 2, \dots, N$  **do**
  - 10:      $\mathbf{z}_n = (\mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{W}[\mathbf{o}_n, :] + \lambda_z \mathbf{I})^{-1} \mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{a}_n[\mathbf{o}_n];$                $\triangleright n\text{-th column of } \mathbf{Z}$
  - 11:   **end for**
  - 12:   **for**  $m = 1, 2, \dots, M$  **do**
  - 13:      $\mathbf{w}_m = (\mathbf{Z}[:, \mathbf{p}_m] \mathbf{Z}[:, \mathbf{p}_m]^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z}[:, \mathbf{p}_m] \mathbf{b}_m[\mathbf{p}_m];$                $\triangleright m\text{-th column of } \mathbf{W}^\top$
  - 14:   **end for**
  - 15: **end while**
  - 16: Output  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M], \mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ ;
- 

### 17.5. Vector Inner Product

We have seen the ALS is to find matrices  $\mathbf{W}, \mathbf{Z}$  such that  $\mathbf{WZ}$  can approximate  $\mathbf{A} \approx \mathbf{WZ}$  in terms of minimum least squared loss:

$$\min_{\mathbf{W}, \mathbf{Z}} \sum_{n=1}^N \sum_{d=1}^M \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2,$$

that is, each entry  $a_{mn}$  in  $\mathbf{A}$  can be approximated by the inner product between the two vectors  $\mathbf{w}_m^\top \mathbf{z}_n$ . The geometric definition of vector inner product is given by

$$\mathbf{w}_m^\top \mathbf{z}_n = \|\mathbf{w}\| \cdot \|\mathbf{z}\| \cos \theta,$$

where  $\theta$  is the angle between  $\mathbf{w}$  and  $\mathbf{z}$ . So if the vector norms of  $\mathbf{w}, \mathbf{z}$  are determined, the smaller the angle, the larger the inner product.

Come back to the Netflix data, where the rating are ranging from 0 to 5 and the larger the better. If  $\mathbf{w}_m$  and  $\mathbf{z}_n$  fall “close” enough, then  $\mathbf{w}^\top \mathbf{z}$  will have a larger value. This reveals the meaning behind the ALS where  $\mathbf{w}_m$  represents the features of movie  $m$ , whilst  $\mathbf{z}_n$  contains the features of user  $n$ . And each element in  $\mathbf{w}_m$  and  $\mathbf{z}_n$  represents a same feature. For example, it could be that the second feature  $\mathbf{w}_{m2}$ <sup>3</sup> represents if the movie is an action movie or not, and  $\mathbf{z}_{n2}$  denotes if the user  $n$  likes action movies or not. If it happens the case, then  $\mathbf{w}_m^\top \mathbf{z}_n$  will be large and approximates  $a_{mn}$  well.

Note that, in the decomposition  $\mathbf{A} \approx \mathbf{WZ}$ , we know the rows of  $\mathbf{W}$  contain the hidden features of the movies and the columns of  $\mathbf{Z}$  contain the hidden features of the users. However, we cannot identify what are the meanings of the rows of  $\mathbf{W}$  or the columns of  $\mathbf{Z}$ . We know they could be something like categories or genres of the movies, that provide some underlying connections between the users and the movies, but we cannot be sure what exactly they are. This is where the terminology “hidden” comes from.

## 17.6. Gradient Descent

In Equation (17.11), we obtain the column-by-column update directly from the full matrix way in Equation (17.10) (with regularization considered). Now let’s see what’s behind the idea. Following from Equation (17.8), the loss under the regularization:

$$L(\mathbf{W}, \mathbf{Z}) = \|\mathbf{WZ} - \mathbf{A}\|^2 + \lambda_w \|\mathbf{W}\|^2 + \lambda_z \|\mathbf{Z}\|^2, \quad \lambda_w > 0, \lambda_z > 0, \quad (17.14)$$

Since we are now considering the minimization of above loss with respect to  $\mathbf{z}_n$ , we can decompose the loss into

$$\begin{aligned} L(\mathbf{z}_n) &= \|\mathbf{WZ} - \mathbf{A}\|^2 + \lambda_w \|\mathbf{W}\|^2 + \lambda_z \|\mathbf{Z}\|^2 \\ &= \|\mathbf{Wz}_n - \mathbf{a}_n\|^2 + \lambda_z \|\mathbf{z}_n\|^2 + \underbrace{\sum_{i \neq n} \|\mathbf{Wz}_i - \mathbf{a}_i\|^2 + \lambda_z \sum_{i \neq n} \|\mathbf{z}_i\|^2 + \lambda_w \|\mathbf{W}\|^2}_{C_{z_n}}, \end{aligned} \quad (17.15)$$

where  $C_{z_n}$  is a constant with respect to  $\mathbf{z}_n$ , and  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ ,  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]$  are the column partitions of  $\mathbf{Z}$ ,  $\mathbf{A}$  respectively. Taking the differential

$$\frac{\partial L(\mathbf{z}_n)}{\partial \mathbf{z}_n} = 2\mathbf{W}^\top \mathbf{W} \mathbf{z}_n - 2\mathbf{W}^\top \mathbf{a}_n + 2\lambda_z \mathbf{z}_n,$$

under which the root is exactly the first update of the column fashion in Equation (17.11):

$$\mathbf{z}_n = (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{a}_n, \quad \text{for } n \in \{1, 2, \dots, N\}.$$

Similarly, we can decompose the loss with respect to  $\mathbf{w}_m$ ,

$$\begin{aligned} L(\mathbf{w}_m) &= \|\mathbf{WZ} - \mathbf{A}\|^2 + \lambda_w \|\mathbf{W}\|^2 + \lambda_z \|\mathbf{Z}\|^2 \\ &= \|\mathbf{Z}^\top \mathbf{W} - \mathbf{A}^\top\|^2 + \lambda_w \|\mathbf{W}^\top\|^2 + \lambda_z \|\mathbf{Z}\|^2 \\ &= \|\mathbf{Z}^\top \mathbf{w}_m - \mathbf{b}_n\|^2 + \lambda_w \|\mathbf{w}_m\|^2 + \underbrace{\sum_{i \neq m} \|\mathbf{Z}^\top \mathbf{w}_i - \mathbf{b}_i\|^2 + \lambda_w \sum_{i \neq m} \|\mathbf{w}_i\|^2 + \lambda_z \|\mathbf{Z}\|^2}_{C_{w_m}}, \end{aligned} \quad (17.16)$$

<sup>3</sup>.  $\mathbf{w}_{m2}$  is the second element of vector  $\mathbf{w}_{m2}$

where  $C_{w_m}$  is a constant with respect to  $\mathbf{w}_m$ , and  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]$ ,  $\mathbf{A}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M]$  are the column partitions of  $\mathbf{W}^\top$ ,  $\mathbf{A}^\top$  respectively. Analogously, taking the differential with respect to  $\mathbf{w}_m$ , it follows that

$$\frac{\partial L(\mathbf{w}_m)}{\partial \mathbf{w}_m} = 2\mathbf{Z}\mathbf{Z}^\top \mathbf{w}_m - 2\mathbf{Z}\mathbf{b}_m + 2\lambda_w \mathbf{w}_m,$$

under which the root is exactly the second update of the column fashion in Equation (17.11):

$$\mathbf{w}_m = (\mathbf{Z}\mathbf{Z}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z}\mathbf{b}_m, \quad \text{for } m \in \{1, 2, \dots, M\}.$$

Now suppose we write out the iteration as the superscript and we want to find the updates  $\{\mathbf{z}_n^{(k+1)}, \mathbf{w}_m^{(k+1)}\}$  base on  $\{\mathbf{Z}^{(k)}, \mathbf{W}^{(k)}\}$ :

$$\begin{cases} \mathbf{z}_n^{(k+1)} \leftarrow \arg \min_{\mathbf{z}_n^{(k)}} L(\mathbf{z}_n^{(k)}); \\ \mathbf{w}_m^{(k+1)} \leftarrow \arg \min_{\mathbf{w}_m^{(k)}} L(\mathbf{w}_m^{(k)}). \end{cases}$$

For simplicity, we will be looking at  $\mathbf{z}_n^{(k+1)} \leftarrow \arg \min_{\mathbf{z}_n^{(k)}} L(\mathbf{z}_n^{(k)} | -)$ , and the derivation for the update on  $\mathbf{w}_m^{(k+1)}$  will be the same. Suppose we want to approximate  $\mathbf{z}_n^{(k+1)}$  by a linear update on  $\mathbf{z}_n^{(k)}$ :

$$\mathbf{z}_n^{(k+1)} = \mathbf{z}_n^{(k)} + \eta \mathbf{v}.$$

The problem now turns to the solution of  $\mathbf{v}$  such that

$$\mathbf{v} = \arg \min_{\mathbf{v}} L(\mathbf{z}_n^{(k)} + \eta \mathbf{v}).$$

By Taylor's formula (Appendix J, p. 472),  $L(\mathbf{z}_n^{(k)} + \eta \mathbf{v})$  can be approximated by

$$L(\mathbf{z}_n^{(k)} + \eta \mathbf{v}) \approx L(\mathbf{z}_n^{(k)}) + \eta \mathbf{v}^\top \nabla L(\mathbf{z}_n^{(k)}),$$

when  $\eta$  is small enough. Then an search under the condition  $\|\mathbf{v}\| = 1$  given positive  $\eta$  is as follows:

$$\mathbf{v} = \arg \min_{\|\mathbf{v}\|=1} L(\mathbf{z}_n^{(k)} + \eta \mathbf{v}) \approx \arg \min_{\|\mathbf{v}\|=1} \left\{ L(\mathbf{z}_n^{(k)}) + \eta \mathbf{v}^\top \nabla L(\mathbf{z}_n^{(k)}) \right\}.$$

This is known as the *greedy search*. The optimal  $\mathbf{v}$  can be obtained by

$$\mathbf{v} = -\frac{\nabla L(\mathbf{z}_n^{(k)})}{\|\nabla L(\mathbf{z}_n^{(k)})\|},$$

i.e.,  $\mathbf{v}$  is in the opposite direction of  $\nabla L(\mathbf{z}_n^{(k)})$ . Therefore, the update of  $\mathbf{z}_n^{(k+1)}$  is reasonable to be taken as

$$\mathbf{z}_n^{(k+1)} = \mathbf{z}_n^{(k)} + \eta \mathbf{v} = \mathbf{z}_n^{(k)} - \eta \frac{\nabla L(\mathbf{z}_n^{(k)})}{\|\nabla L(\mathbf{z}_n^{(k)})\|},$$

**Algorithm 59** Alternating Least Squares with Full Entries and Gradient Descent**Require:**  $A \in \mathbb{R}^{M \times N}$ ;

- 1: initialize  $\mathbf{W} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  randomly without condition on the rank and the relationship between  $M, N, K$ ;
- 2: choose a stop criterion on the approximation error  $\delta$ ;
- 3: choose regularization parameters  $\lambda_w, \lambda_z$ , and step size  $\eta_w, \eta_z$ ;
- 4: choose maximal number of iterations  $C$ ;
- 5:  $iter = 0$ ;
- 6: **while**  $\|A - (\mathbf{W}\mathbf{Z})\|^2 > \delta$  and  $iter < C$  **do**
- 7:    $iter = iter + 1$ ;
- 8:   **for**  $n = 1, 2, \dots, N$  **do**
- 9:      $\mathbf{z}_n^{(k+1)} = \mathbf{z}_n^{(k)} - \eta_z \frac{\nabla L(\mathbf{z}_n^{(k)})}{\|\nabla L(\mathbf{z}_n^{(k)})\|}$ ; ▷  $n$ -th column of  $\mathbf{Z}$
- 10:   **end for**
- 11:   **for**  $m = 1, 2, \dots, M$  **do**
- 12:      $\mathbf{w}_m^{(k+1)} = \mathbf{w}_m^{(k)} - \eta_w \frac{\nabla L(\mathbf{w}_m^{(k)})}{\|\nabla L(\mathbf{w}_m^{(k)})\|}$ ; ▷  $m$ -th column of  $\mathbf{W}^\top$
- 13:   **end for**
- 14: **end while**
- 15: Output  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M], \mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ ;

which usually called the *gradient descent*. Similarly, the gradient descent of  $\mathbf{w}_m^{(k+1)}$  is given by

$$\mathbf{w}_m^{(k+1)} = \mathbf{w}_m^{(k)} + \eta \mathbf{v} = \mathbf{w}_m^{(k)} - \eta \frac{\nabla L(\mathbf{w}_m^{(k)})}{\|\nabla L(\mathbf{w}_m^{(k)})\|}.$$

The updated procedure on Algorithm 57 by a gradient descent way is then formulated in Algorithm 59.

**Geometrical Interpretation of Gradient Descent****Lemma 17.1: (Direction of Gradients)**

An important fact is that gradients are orthogonal to level curves (a.k.a., level surface).

**Proof** [of Lemma 17.1] This is equivalently to prove that the gradients is orthogonal to the tangent of the level curve. For simplicity, let's first look at the 2-dimensional case. Suppose the level curve has the form  $f(x, y) = c$ . This implicitly gives a relation between  $x$  and  $y$  such that  $y = y(x)$  where  $y$  can be thought of as a function of  $x$ . Therefore, the level curve can be written as

$$f(x, y(x)) = c.$$

The chain rule indicates

$$\frac{\partial f}{\partial x} \underbrace{\frac{dx}{dx}}_{=1} + \frac{\partial f}{\partial y} \frac{dy}{dx} = 0.$$

Therefore, the gradients is perpendicular to the tangent:

$$\left\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\rangle \cdot \left\langle \frac{dx}{dx}, \frac{dy}{dx} \right\rangle = 0.$$

Let us now treat the problem in full generality, suppose the level curve of a vector  $\mathbf{x} \in \mathbb{R}^n$ :  $f(\mathbf{x}) = f(x_1, x_2, \dots, x_n) = c$ . Each variable  $x_i$  can be regarded as a function of a variable  $t$  on the level curve  $f(\mathbf{x}) = c$ :  $f(x_1(t), x_2(t), \dots, x_n(t)) = c$ . Differentiate the equation with respect to  $t$  by chain rule:

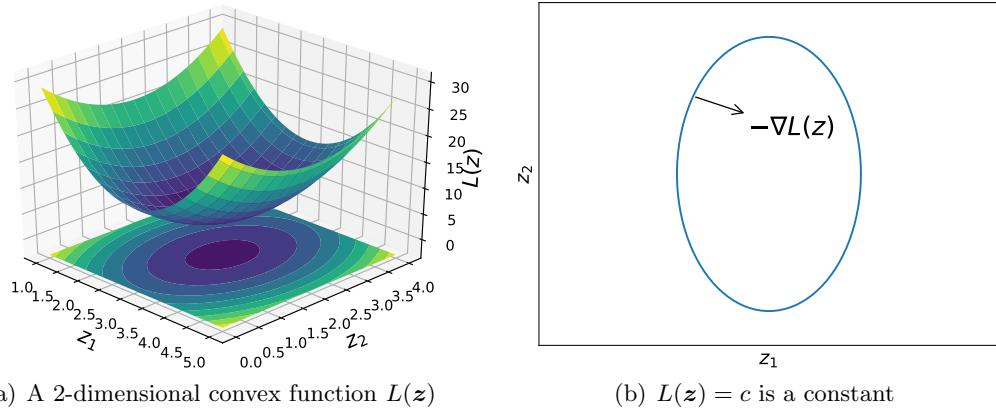
$$\frac{\partial f}{\partial x_1} \frac{dx_1}{dt} + \frac{\partial f}{\partial x_2} \frac{dx_2}{dt} + \dots + \frac{\partial f}{\partial x_n} \frac{dx_n}{dt} = 0.$$

Therefore, the gradients is perpendicular to the tangent in  $n$ -dimensional case:

$$\left\langle \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right\rangle \cdot \left\langle \frac{dx_1}{dt}, \frac{dx_2}{dt}, \dots, \frac{dx_n}{dt} \right\rangle = 0.$$

This completes the proof. ■

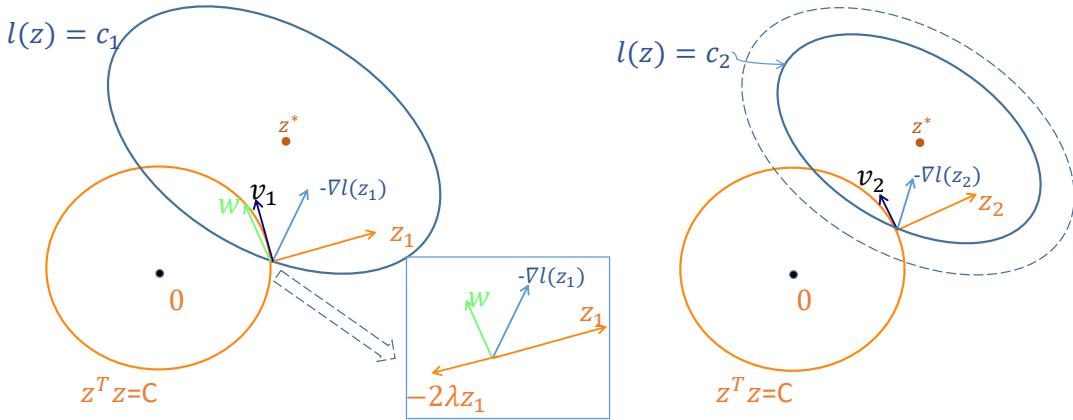
The lemma above reveals the geometrical interpretation of gradient descent. For finding a solution to minimize a convex function  $L(\mathbf{z})$ , gradient descent goes to the negative gradient direction that can decrease the loss. Figure 17.2 depicts a 2-dimensional case, where  $-\nabla L(\mathbf{z})$  pushes the loss to decrease for the convex function  $L(\mathbf{z})$ .



**Figure 17.2:** Figure 17.2(a) shows a function “density” and a contour plot (blue=low, yellow=high) where the upper graph is the “density”, and the lower one is the projection of it (i.e., contour). Figure 17.2(b):  $-\nabla L(\mathbf{z})$  pushes the loss to decrease for the convex function  $L(\mathbf{z})$ .

## 17.7. Regularization: A Geometrical Interpretation

We have seen in Section 17.3 that the regularization can extend the ALS to general matrices. The gradient descent can reveal the geometric meaning of the regularization. To avoid confusion, we denote the loss function without regularization by  $l(\mathbf{z})$  and the loss with



**Figure 17.3:** Constrained gradient descent with  $z^T z \leq C$ . The green vector  $w$  is the projection of  $v_1$  into  $z^T z \leq C$  where  $v_1$  is the component of  $-\nabla l(z)$  perpendicular to  $z_1$ . The right picture is the next step after the update in the left picture.  $z^*$  denotes the optimal solution of  $\{\min l(z)\}$ .

regularization by  $L(\mathbf{z}) = l(\mathbf{z}) + \lambda_z \|\mathbf{z}\|^2$  where  $l(\mathbf{z}) : \mathbb{R}^n \rightarrow \mathbb{R}$ . When minimizing  $l(\mathbf{z})$ , descent method will search in  $\mathbb{R}^n$  for a solution. However, in machine learning, searching in the whole space can cause overfitting. A partial solution is to search in a subset of the vector space, e.g., searching in  $\mathbf{z}^\top \mathbf{z} < C$  for some constant  $C$ . That is

$$\arg \min_{\mathbf{z}} \quad l(\mathbf{z}), \quad s.t., \quad \mathbf{z}^\top \mathbf{z} \leq C.$$

As shown above, a trivial gradient descent method will go further in the direction of  $-\nabla l(\mathbf{z})$ , i.e., update  $\mathbf{z}$  by  $\mathbf{z} \leftarrow \mathbf{z} - \eta \nabla l(\mathbf{z})$  for small step size  $\eta$ . When the level curve is  $l(\mathbf{z}) = c_1$  and the current position of  $\mathbf{z} = \mathbf{z}_1$  where  $\mathbf{z}_1$  is the intersection of  $\mathbf{z}^\top \mathbf{z} = C$  and  $l(\mathbf{z}) = c_1$ , the descent direction  $-\nabla l(\mathbf{z}_1)$  will be perpendicular to the level curve of  $l(\mathbf{z}_1) = c_1$  as shown in the left picture of Figure 17.3. However, if we further restrict the optimal value can only be in  $\mathbf{z}^\top \mathbf{z} \leq C$ , the trivial descent direction  $-\nabla l(\mathbf{z}_1)$  will lead  $\mathbf{z}_2 = \mathbf{z}_1 - \eta \nabla l(\mathbf{z}_1)$  outside of  $\mathbf{z}^\top \mathbf{z} \leq C$ . A solution is to decompose the step  $-\nabla l(\mathbf{z}_1)$  into

$$-\nabla l(\mathbf{z}_1) = a\mathbf{z}_1 + \mathbf{v}_1,$$

where  $a\mathbf{z}_1$  is the component perpendicular to the curve of  $\mathbf{z}^\top \mathbf{z} = C$ , and  $\mathbf{v}_1$  is the component parallel to the curve of  $\mathbf{z}^\top \mathbf{z} = C$ . Keep only the step  $\mathbf{v}_1$ , then the update

$$\mathbf{z}_2 = \text{project}(\mathbf{z}_1 + \eta \mathbf{v}_1) = \text{project} \left( \mathbf{z}_1 + \eta \underbrace{(-\nabla l(\mathbf{z}_1) - a\mathbf{z}_1)}_{\mathbf{v}_1} \right)^4$$

will lead to a smaller loss from  $l(\mathbf{z}_1)$  to  $l(\mathbf{z}_2)$  and still match  $\mathbf{z}^\top \mathbf{z} \leq C$ . This is known as the *projection gradient descent*. It is not hard to see that the update  $\mathbf{z}_2 = \text{project}(\mathbf{z}_1 + \eta \mathbf{v}_1)$  is

4. where the  $\text{project}(\mathbf{x})$  will project the vector  $\mathbf{x}$  to the closest point inside  $\mathbf{z}^\top \mathbf{z} \leq C$ . Notice here the direct update  $\mathbf{z}_2 = \mathbf{z}_1 + \eta \mathbf{v}_1$  can still make  $\mathbf{z}_2$  outside the curve of  $\mathbf{z}^\top \mathbf{z} \leq C$ .

equivalent to finding a vector  $\mathbf{w}$  (shown by the green vector in the left picture of Figure 17.3) such that  $\mathbf{z}_2 = \mathbf{z}_1 + \mathbf{w}$  is inside the curve of  $\mathbf{z}^\top \mathbf{z} \leq C$ . Mathematically, the  $\mathbf{w}$  can be obtained by  $-\nabla l(\mathbf{z}_1) - 2\lambda \mathbf{z}_1$  for some  $\lambda$  as shown in the middle picture of Figure 17.3. This is exactly the negative gradient of  $L(\mathbf{z}) = l(\mathbf{z}) + \lambda \|\mathbf{z}\|^2$  such that

$$\nabla L(\mathbf{z}) = \nabla l(\mathbf{z}) + 2\lambda \mathbf{z},$$

and

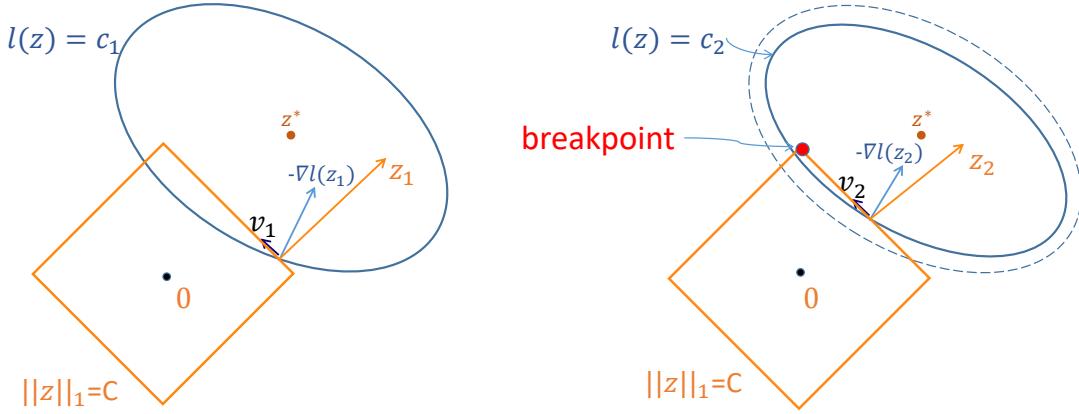
$$\mathbf{w} = -\nabla L(\mathbf{z}) \quad \xrightarrow{\text{leads to}} \quad \mathbf{z}_2 = \mathbf{z}_1 + \mathbf{w} = \mathbf{z}_1 - \nabla L(\mathbf{z}).$$

And in practice, a small step size  $\eta$  can avoid going outside the curve of  $\mathbf{z}^\top \mathbf{z} \leq C$ :

$$\mathbf{z}_2 = \mathbf{z}_1 - \eta \nabla L(\mathbf{z}),$$

which is exactly what we have discussed in Section 17.3, the regularization term.

**Sparsity** In rare cases, we want to find sparse solution  $\mathbf{z}$  such that  $l(\mathbf{z})$  is minimized. Constrained in  $\|\mathbf{z}\|_1 \leq C$  exists to this purpose where  $\|\cdot\|_1$  is the  $l_1$  norm of a vector or a matrix. The illustration of the  $l_1$  in 2-dimensional and 3-dimensional space is shown in Figure 27.8 and 27.9. Similar to the previous case, the  $l_1$  constrained optimization pushes the gradient descent towards the border of the level of  $\|\mathbf{z}\|_1 = C$ . The situation in the 2-dimensional case is shown in Figure 17.4. In a high-dimensional case, many elements in  $\mathbf{z}$  will be pushed into the breakpoint of  $\|\mathbf{z}\|_1 = C$  as shown in the right picture of Figure 17.4.



**Figure 17.4:** Constrained gradient descent with  $\|\mathbf{z}\|_1 \leq C$ , where the red dot denotes the breakpoint in  $l_1$  norm. The right picture is the next step after the update in the left picture.  $\mathbf{z}^*$  denotes the optimal solution of  $\{\min l(\mathbf{z})\}$ .

## 17.8. Stochastic Gradient Descent

Now suppose we come back to the per-example loss:

$$L(\mathbf{W}, \mathbf{Z}) = \sum_{n=1}^N \sum_{m=1}^M \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2 + \lambda_w \|\mathbf{w}_m\|^2 + \lambda_z \|\mathbf{z}_n\|^2.$$

And when we iteratively decrease the per-example loss term  $l(\mathbf{w}_m, \mathbf{z}_n) = (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2$  for all  $m \in [0, M], n \in [1, N]$ , the full loss  $L(\mathbf{W}, \mathbf{Z})$  can also be decreased. This is known as the *stochastic coordinate descent*. The differentials with respect to  $\mathbf{w}_m, \mathbf{z}_n$ , and their roots are given by

$$\begin{cases} \nabla l(\mathbf{z}_n) = \frac{\partial l(\mathbf{w}_m, \mathbf{z}_n)}{\partial \mathbf{z}_n} = 2\mathbf{w}_m \mathbf{w}_m^\top \mathbf{z}_n + 2\lambda_w \mathbf{w}_m - 2a_{mn} \mathbf{w}_m \\ \quad \text{leads to } \mathbf{z}_n = a_{mn} (\mathbf{w}_m \mathbf{w}_m^\top + \lambda_z \mathbf{I})^{-1} \mathbf{w}_m; \\ \nabla l(\mathbf{w}_m) = \frac{\partial l(\mathbf{w}_m, \mathbf{z}_n)}{\partial \mathbf{w}_m} = 2\mathbf{z}_n \mathbf{z}_n^\top \mathbf{w}_m + 2\lambda_z \mathbf{z}_n - 2a_{mn} \mathbf{z}_n \\ \quad \text{leads to } \mathbf{w}_m = a_{mn} (\mathbf{z}_n \mathbf{z}_n^\top + \lambda_w \mathbf{I})^{-1} \mathbf{w}_n. \end{cases}$$

or analogously, the update can be done by gradient descent, and since we update by per-example loss, it is also known as the *stochastic gradient descent*

$$\begin{cases} \mathbf{z}_n = \mathbf{z}_n - \eta_z \frac{\nabla l(\mathbf{z}_n)}{\|\nabla l(\mathbf{z}_n)\|}; \\ \mathbf{w}_m = \mathbf{w}_m - \eta_w \frac{\nabla l(\mathbf{w}_m)}{\|\nabla l(\mathbf{w}_m)\|}. \end{cases}$$

The stochastic gradient descent update for ALS is formulated in Algorithm 60. And in practice, the  $m, n$  in the algorithm can be randomly produced, that's where the name *stochastic* comes from.

---

**Algorithm 60** Alternating Least Squares with Full Entries and Stochastic Gradient Descent

---

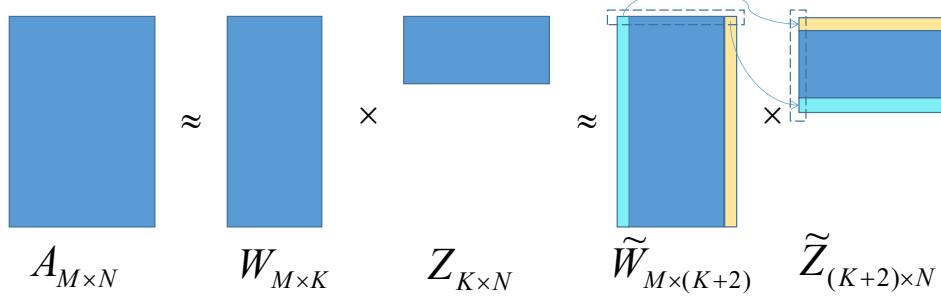
**Require:**  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

```

1: initialize $\mathbf{W} \in \mathbb{R}^{M \times K}$, $\mathbf{Z} \in \mathbb{R}^{K \times N}$ randomly without condition on the rank and the
 relationship between M, N, K ;
2: choose a stop criterion on the approximation error δ ;
3: choose regularization parameters λ_w, λ_z , and step size η_w, η_z ;
4: choose maximal number of iterations C ;
5: $iter = 0$;
6: while $\|\mathbf{A} - (\mathbf{W}\mathbf{Z})\|^2 > \delta$ and $iter < C$ do
7: $iter = iter + 1$;
8: for $n = 1, 2, \dots, N$ do
9: for $m = 1, 2, \dots, M$ do \triangleright in practice, m, n can be randomly produced
10: $\mathbf{z}_n = \mathbf{z}_n - \eta_z \frac{\nabla l(\mathbf{z}_n)}{\|\nabla l(\mathbf{z}_n)\|};$ \triangleright n -th column of \mathbf{Z}
11: $\mathbf{w}_m = \mathbf{w}_m - \eta_w \frac{\nabla l(\mathbf{w}_m)}{\|\nabla l(\mathbf{w}_m)\|};$ \triangleright m -th column of \mathbf{W}^\top
12: end for
13: end for
14: end while
15: Output $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M], \mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N];$

```

---



**Figure 17.5:** Bias terms in alternating least squares where the **yellow** entries denote ones (which are fixed) and **cyan** entries denote the added features to fit the bias terms. The dotted boxes give an example on how the bias terms work.

### 17.9. Bias Term

In ordinary least squares, a bias term is added to the raw matrix as shown in Equation (17.1). A similar idea can be applied to the ALS problem. We can add a fixed column with all 1's to the last column of  $\mathbf{W}$ , thus an extra row should be added to last row of  $\mathbf{Z}$  to fit the features introduced by the bias term in  $\mathbf{W}$ . Analogously, a fixed row with all 1's can be added to the first row of  $\mathbf{Z}$ , and an extra column in the first column of  $\mathbf{W}$  to fit the features. The situation is shown in Figure 17.5.

Following from the loss with respect to the columns of  $\mathbf{Z}$  in Equation (17.15), suppose  $\tilde{\mathbf{z}}_n = \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix}$  is the  $n$ -th column of  $\tilde{\mathbf{Z}}$ , we have

$$\begin{aligned}
L(\mathbf{z}_n) &= \|\tilde{\mathbf{W}}\tilde{\mathbf{Z}} - \mathbf{A}\|^2 + \lambda_w \|\tilde{\mathbf{W}}\|^2 + \lambda_z \|\tilde{\mathbf{Z}}\|^2 \\
&= \left\| \tilde{\mathbf{W}} \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix} - \mathbf{a}_n \right\|^2 + \underbrace{\lambda_z \|\tilde{\mathbf{z}}_n\|^2}_{=\lambda_z \|\mathbf{z}_n\|^2 + \lambda_z} + \sum_{i \neq n} \|\tilde{\mathbf{W}}\tilde{\mathbf{z}}_i - \mathbf{a}_i\|^2 + \lambda_z \sum_{i \neq n} \|\tilde{\mathbf{z}}_i\|^2 + \lambda_w \|\tilde{\mathbf{W}}\|^2 \\
&= \left\| [\bar{\mathbf{w}}_0 \quad \tilde{\mathbf{W}}] \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix} - \mathbf{a}_n \right\|^2 + \lambda_z \|\mathbf{z}_n\|^2 + C_{z_n} = \left\| \bar{\mathbf{W}}\mathbf{z}_n - \underbrace{(\mathbf{a}_n - \bar{\mathbf{w}}_0)}_{\bar{\mathbf{a}}_n} \right\|^2 + \lambda_z \|\mathbf{z}_n\|^2 + C_{z_n},
\end{aligned} \tag{17.17}$$

where  $\bar{\mathbf{w}}_0$  is the first column of  $\tilde{\mathbf{W}}$  and  $C_{z_n}$  is a constant with respect to  $\mathbf{z}_n$ . Let  $\bar{\mathbf{a}}_n = \mathbf{a}_n - \bar{\mathbf{w}}_0$ , the update of  $\mathbf{z}_n$  is just like the one in Equation (17.15) where the differential is given by:

$$\frac{\partial L(\mathbf{z}_n)}{\partial \mathbf{z}_n} = 2\bar{\mathbf{W}}^\top \bar{\mathbf{W}} \mathbf{z}_n - 2\bar{\mathbf{W}}^\top \bar{\mathbf{a}}_n + 2\lambda_z \mathbf{z}_n.$$

Therefore the update on  $\mathbf{z}_n$  is given by the root of the above differential:

$$\text{update on } \tilde{\mathbf{z}}_n = \begin{cases} \mathbf{z}_n = (\bar{\mathbf{W}}^\top \bar{\mathbf{W}} + \lambda_z \mathbf{I})^{-1} \bar{\mathbf{W}}^\top \bar{\mathbf{a}}_n, & \text{for } n \in \{1, 2, \dots, N\}; \\ \tilde{\mathbf{z}}_n = \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix}. \end{cases}$$

Similarly, follow from the loss with respect to each row of  $\mathbf{W}$  in Equation (17.16), suppose  $\tilde{\mathbf{w}}_m = \begin{bmatrix} \mathbf{w}_m \\ 1 \end{bmatrix}$  is the  $m$ -th row of  $\widetilde{\mathbf{W}}$  (or  $m$ -th column of  $\widetilde{\mathbf{W}}^\top$ ), we have

$$\begin{aligned} L(\mathbf{w}_m) &= \|\widetilde{\mathbf{Z}}^\top \widetilde{\mathbf{W}} - \mathbf{A}^\top\|^2 + \lambda_w \|\widetilde{\mathbf{W}}^\top\|^2 + \lambda_z \|\widetilde{\mathbf{Z}}\|^2 \\ &= \|\widetilde{\mathbf{Z}}^\top \tilde{\mathbf{w}}_m - \mathbf{b}_m\|^2 + \underbrace{\lambda_w \|\tilde{\mathbf{w}}_m\|^2}_{=\lambda_w \|\mathbf{w}_m\|^2 + \lambda_w} + \sum_{i \neq m} \|\widetilde{\mathbf{Z}}^\top \tilde{\mathbf{w}}_i - \mathbf{b}_i\|^2 + \lambda_w \sum_{i \neq m} \|\tilde{\mathbf{w}}_i\|^2 + \lambda_z \|\widetilde{\mathbf{Z}}\|^2 \\ &= \left\| \begin{bmatrix} \widetilde{\mathbf{Z}}^\top & \bar{\mathbf{z}}_0 \end{bmatrix} \begin{bmatrix} \mathbf{w}_m \\ 1 \end{bmatrix} - \mathbf{b}_m \right\|^2 + \lambda_w \|\mathbf{w}_m\|^2 + C_{w_m} \\ &= \left\| \bar{\mathbf{Z}}^\top \mathbf{w}_m - (\mathbf{b}_m - \bar{\mathbf{z}}_0) \right\|^2 + \lambda_w \|\mathbf{w}_m\|^2 + C_{w_m}, \end{aligned} \tag{17.18}$$

where  $\bar{\mathbf{z}}_0$  is the last column of  $\widetilde{\mathbf{Z}}^\top$  and  $\bar{\mathbf{Z}}^\top$  is the left columns of it:  $\widetilde{\mathbf{Z}}^\top = \begin{bmatrix} \mathbf{w}_m \\ 1 \end{bmatrix}$ ,  $C_{w_m}$  is a constant with respect to  $\mathbf{w}_m$ , and  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]$ ,  $\mathbf{A}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M]$  are the column partitions of  $\mathbf{W}^\top$ ,  $\mathbf{A}^\top$  respectively. Let  $\bar{\mathbf{b}}_m = \mathbf{b}_m - \bar{\mathbf{z}}_0$ , the update of  $\mathbf{w}_m$  is again just like the one in Equation (17.16) where the differential is given by:

$$\frac{\partial L(\mathbf{w}_m d)}{\partial \mathbf{w}_m} = 2\bar{\mathbf{Z}} \cdot \bar{\mathbf{Z}}^\top \mathbf{w}_m - 2\bar{\mathbf{Z}} \cdot \bar{\mathbf{b}}_m + 2\lambda_w \mathbf{w}_m.$$

Therefore the update on  $\mathbf{w}_m$  is given by the root of the above differential

$$\text{update on } \tilde{\mathbf{w}}_m = \begin{cases} \mathbf{w}_m = (\bar{\mathbf{Z}} \cdot \bar{\mathbf{Z}}^\top + \lambda_w \mathbf{I})^{-1} \bar{\mathbf{Z}} \cdot \bar{\mathbf{b}}_m, & \text{for } m \in \{1, 2, \dots, M\}; \\ \tilde{\mathbf{w}}_m = \begin{bmatrix} \mathbf{w}_m \\ 1 \end{bmatrix}. \end{cases}$$

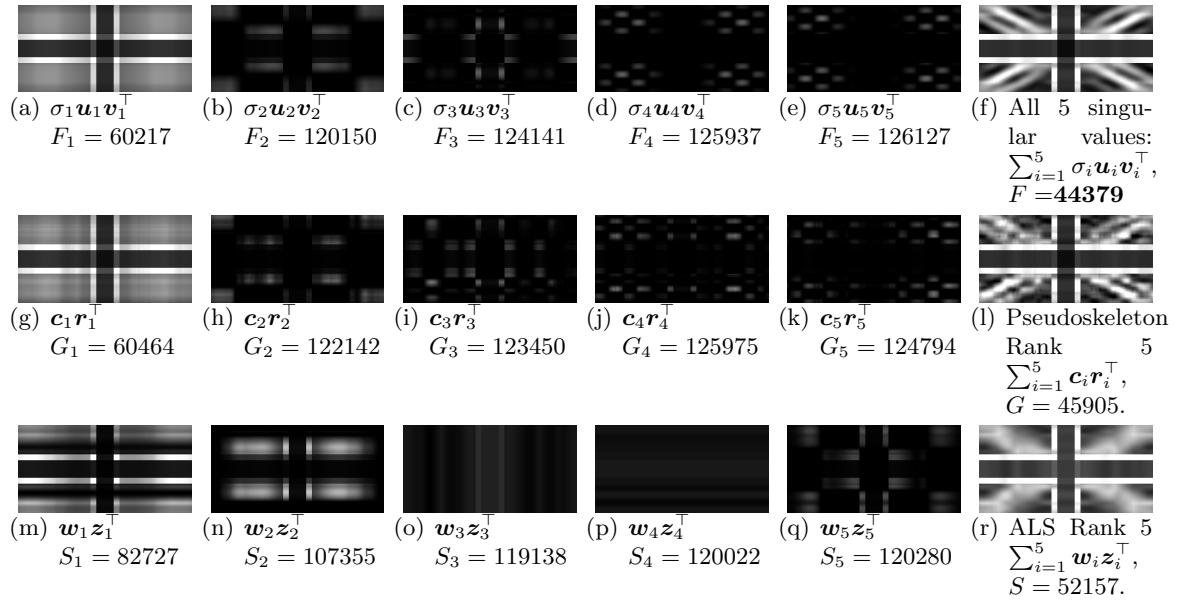
Similar updates by gradient descent under the bias terms or treatment on missing entries can be deduced and we shall not repeat the details (see Section 17.6 and 17.4 for a reference).

## 17.10. Applications

### 17.10.1 Low-Rank Approximation

We have discussed and compared the effects of SVD and pseudoskeleton on low-rank approximation in Section 14.7.4 (p. 284). The image to be compressed is shown in Figure 14.4 with size of  $600 \times 1200$  and rank 402. Figure 14.5 shows the image reconstructed by the first singular value already approximates the original image very well. Figure 14.8 shows the difference of each compression with rank 90, 60, 30, 10. We find SVD does well with rank 90, 60, 30. Pseudoskeleton compresses well in the black horizontal and vertical lines in the image. But it performs poor in the details of the flag.

Similar results can be observed for the low-rank approximation via the ALS decomposition. The ALS approximation is given by  $\mathbf{A} \approx \mathbf{W}\mathbf{Z}$  where  $\mathbf{W} \in \mathbb{R}^{m \times \gamma}$ ,  $\mathbf{Z} \in \mathbb{R}^{\gamma \times n}$  if



**Figure 17.6:** Image compression for gray flag image into a rank-5 matrix via the SVD, and decompose into 5 parts where  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_5$ , i.e.,  $F_1 \leq F_2 \leq \dots \leq F_5$  with  $F_i = \|\sigma_i \mathbf{u}_i \mathbf{v}_i^\top - \mathbf{A}\|_F$  for  $i \in \{1, 2, \dots, 5\}$ . And reconstruct images by single singular value and its corresponding left and right singular vectors,  $\mathbf{c}_i \mathbf{r}_i^\top$ ,  $\mathbf{w}_i \mathbf{z}_i^\top$  respectively.

$\mathbf{A} \in \mathbb{R}^{m \times n}$  such that  $\mathbf{W}$  and  $\mathbf{Z}$  are rank- $\gamma$  matrices. Suppose  $\gamma = 5$ , and

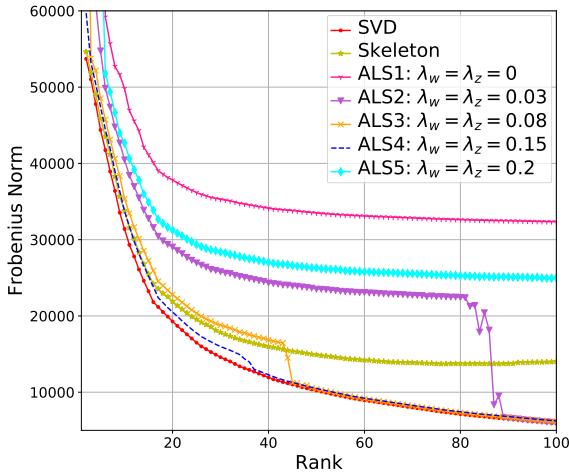
$$\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_5], \quad \text{and} \quad \mathbf{Z} = \begin{bmatrix} \mathbf{z}_1^\top \\ \mathbf{z}_2^\top \\ \vdots \\ \mathbf{z}_5^\top \end{bmatrix},$$

are the column and row partitions of  $\mathbf{W}, \mathbf{Z}$  respectively<sup>5</sup>. Then  $\mathbf{A}$  can be approximated by  $\sum_{i=1}^5 \mathbf{w}_i \mathbf{z}_i^\top$ . The partitions are ordered such that

$$\underbrace{\|\mathbf{w}_1 \mathbf{z}_1^\top - \mathbf{A}\|_F}_{S_1} \leq \underbrace{\|\mathbf{w}_2 \mathbf{z}_2^\top - \mathbf{A}\|_F}_{S_2} \leq \dots \leq \underbrace{\|\mathbf{w}_5 \mathbf{z}_5^\top - \mathbf{A}\|_F}_{S_5}.$$

We observe that  $\mathbf{w}_1 \mathbf{z}_1^\top$  works slightly **different** to that of  $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top$  where the reconstruction error measured by Frobenius norm are not close as well (82,727 in the pseudoskeleton case compared to that of 60,217 in the SVD case). As we mentioned previously,  $\mathbf{c}_1 \mathbf{r}_1^\top$  works similarly to that of  $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top$  since the pseudoskeleton relies on the SVD. However, in ALS, the reconstruction is all from least squares optimization. The key difference between ALS and SVD is in that, in SVD, the importance of each vector in the basis is relative to the value of the singular value associated with that vector. This usually means that the first vector of the basis dominates and is the most used vector to reconstruct data, then the

<sup>5</sup>. For simplicity, note that this definition is different to what we have defined in Section 17.2 where we define  $\mathbf{w}_i$  as the rows of  $\mathbf{W}$ .



**Figure 17.7:** Comparison of reconstruction errors measured by Frobenius norm among the SVD, pseudoskeleton, and ALS where the approximated rank ranges from 3 to 100. ALS with well-selected parameters works similar to SVD.

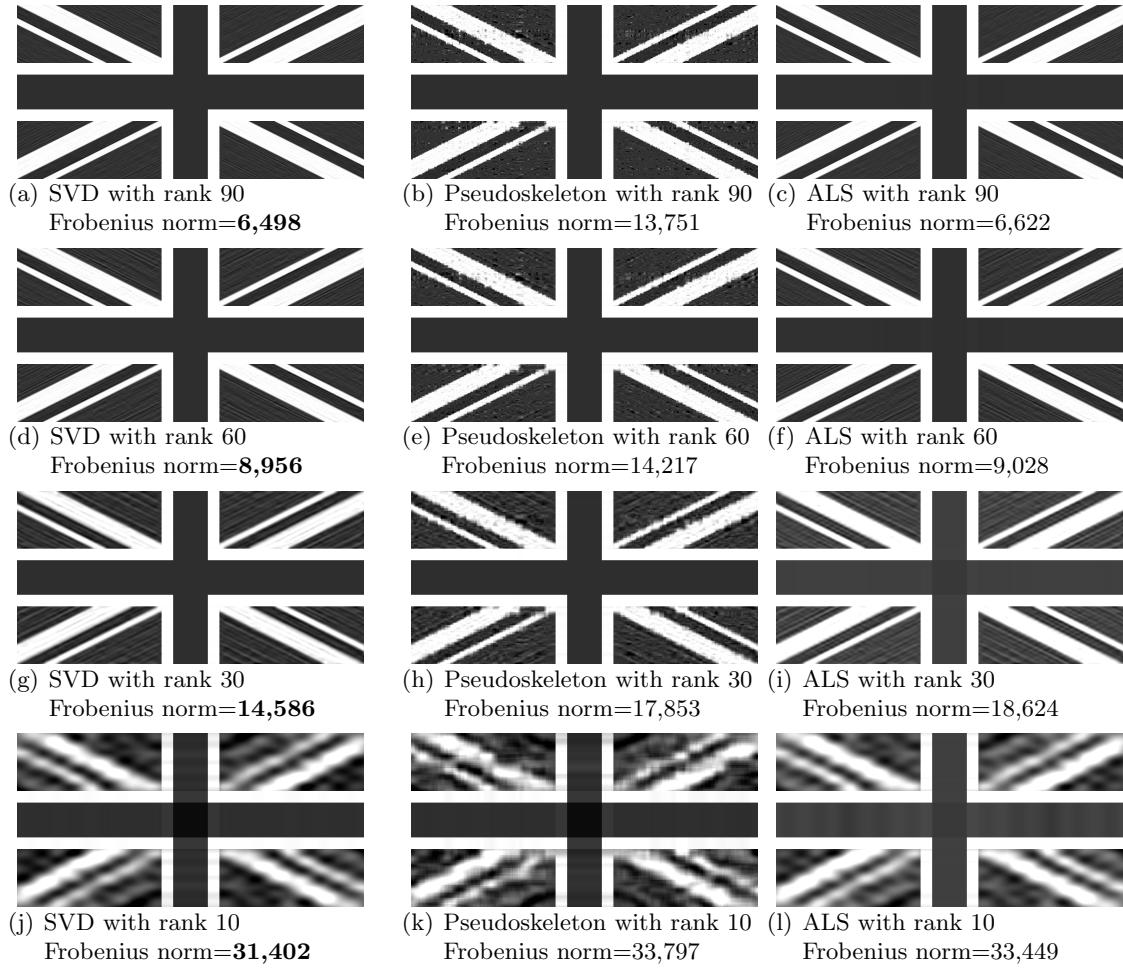
second vector and so on, so the basis in SVD has an implicit hierarchy and that doesn't happen in ALS where we find the second component  $w_2 z_2^\top$  via ALS in Figure 17.6(n) plays an important role in the reconstruction of the original figure, whereas the second component  $\sigma_2 u_2 v_2^\top$  via SVD in Figure 17.6(b) plays a small role in the reconstruction.

We finally compare low-rank approximation among the SVD, pseudoskeleton, and ALS with different ranks. Figure 17.8 shows the difference of each compression with rank 90, 60, 30, 10. We observe that the SVD does well with rank 90, 60, 30. The pseudoskeleton-approximation compresses well in the black horizontal and vertical lines in the image. But it performs poor in the details of the flag. ALS works similarly to the SVD in terms of visual expression and reconstruction errors measured by Frobenius norm. Figure 17.7 shows the comparison of the reconstruction errors among the SVD, the pseudoskeleton, and the ALS approximations measured by Frobenius norm ranging from rank 3 to 100 where we find in all cases, the truncated SVD does best in terms of Frobenius norm. Again, similar results can be observed when applied to spectral norm. The ALS works better than the pseudoskeleton decomposition when  $\lambda_w = \lambda_z = 0.15$ . An interesting cutoff happens when  $\lambda_w = \lambda_z = \{0.03, 0.08, 0.15\}$ . That is, when the rank increases, the ALS will be very close to the SVD in the sense of low-rank approximation.

### 17.10.2 Movie Recommender

The ALS is extensively developed for the movie recommender system. To see this, we obtain the “movielens100k” dataset from MovieLens (Harper and Konstan, 2015)<sup>6</sup>. It consists of 100,000 ratings from 943 users on 1,682 movies. The rating values go from 0 to 5. The data was collected through the MovieLens website during the seven-month period from September 19th, 1997 through April 22nd, 1998. This data has been cleaned up - users who had less than 20 ratings or did not have complete demographic information were removed from this data set such that simple demographic info for the users (age, gender, occupation, zip) can be obtained. However, we will only work on the trivial rating matrix.

<sup>6</sup>. <http://grouplens.org>



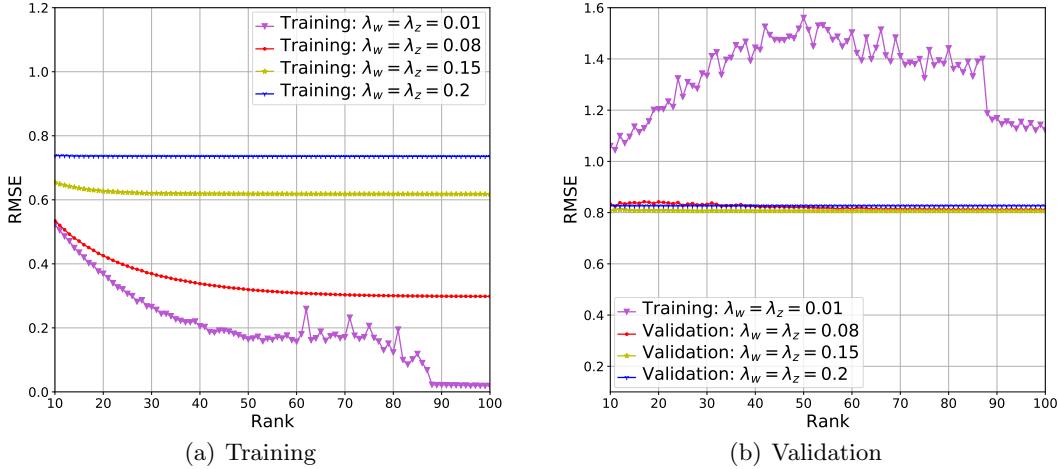
**Figure 17.8:** Image compression for gray flag image with different rank.

The dataset is split into training and validation data, around 95,015 and 4,985 ratings respectively. The error is measured by root mean squared error (RMSE). The RMSE is frequently used as a measure of the differences between values. For a set of values  $\{x_1, x_2, \dots, x_n\}$  and its predictions  $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ , the RMSE can be described as

$$\text{RMSE}(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}.$$

The minimal RMSE for validation is obtained when  $K = 185$  and  $\lambda_w = \lambda_z = 0.15$ , and it is equal to 0.806 as shown in Figure 17.9. Therefore, when the rating ranges from 0 to 5, the ALS at least can predict whether the user likes to watch the movie (ranges 4 to 5) or not much (ranges 0 to 2).

A recommender system can work simply by pushing the movie  $m$  when  $a_{mn} \geq 4$  if user  $n$  have not rated the movie  $m$ . Or in rare cases, it happens that the user  $n$  have rated all the movies he likes (say rates  $\geq 4$ ). Then, a partial solution is to find out similar movies to



**Figure 17.9:** Comparison of training and validation error for “movielens100k” data set with different reduction dimension and regularization parameters.

high-rating movies to push. Suppose user  $n$  likes movie  $m$  very much and he has rated with 5:  $a_{mn} = 5$ . Under the ALS approximation  $\mathbf{A} = \mathbf{W}\mathbf{Z}$  where each row of  $\mathbf{W}$  represents the hidden features of each movie (see Section 17.5 on the vector product). The solution is given by finding the most similar movies to movie  $m$  that user  $n$  has not rated (or watched). In mathematical language,

$$\arg \max_{\mathbf{w}_i} \text{similarity}(\mathbf{w}_i, \mathbf{w}_m), \quad \text{for all } i \neq o_n,$$

where  $\mathbf{w}_i$ 's are the rows of  $\mathbf{W}$  representing the hidden feature of movie  $i$  and  $\mathbf{o}_n$  is a mask vector indicating the movies that user  $n$  has rated.

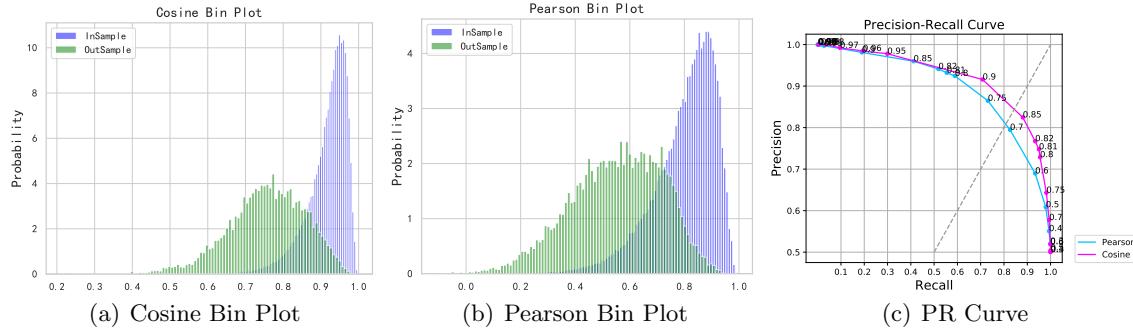
The method above relies on the similarity function between two vectors. The *cosine similarity* is the most commonly used measure. It is defined to equal the cosine of the angle between the two vectors:

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|},$$

where the value ranges from -1 to 1 with -1 is perfectly dissimilar and 1 is perfectly similar. From the above definition, it follows that the cosine similarity depends only on the angle between the two non-zero vectors, but not on their magnitudes since it can be regarded as the inner product between the normalized vectors. A second measure for similarity is known as the *Pearson similarity*:

$$\text{Pearson}(\mathbf{x}, \mathbf{y}) = \frac{\text{Cov}(\mathbf{x}, \mathbf{y})}{\sigma_x \cdot \sigma_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}},$$

whose range varies between -1 and 1, where -1 is perfectly dissimilar and 1 is perfectly similar. The Pearson similarity is usually used to measure the linear correlation between two sets of data. It is the ratio between the covariance of two variables and the product of their standard deviations.



**Figure 17.10:** Distribution of the insample and outsample under cosine and Pearson similarity and the Precision-Recall curve from them.

Following from the example above on the movielens100k dataset, we choose  $\lambda_w = \lambda_z = 0.15$  for the regularization and the rank 62 to minimize the RMSE. We want to look at the similarity between different movie hidden vectors. Define further the “insample” as the similarity between the movies having rates 5 for each user, and “outsample” as the similarity between the movies having rate 5 and 1 for each user. Figure 17.10(b) and 17.10(a) depict the bin plot of the distribution of insample and outsample under cosine and Pearson similarity. Figure 17.10(c) shows the precision-recall (PR) curve of them where we find the cosine similarity works better such that it can find out more than 73% of the potential high-rating movies with a 90% precision. However, Pearson similarity can only separate out about 64% of the high-rating movies to have a 90% precision. In practice, other measure can also be explored, such as negative Euclidean distance in which case the Euclidean distance can measure the “dissimilarity” between two vectors, and a negative one thus represents the similarity of them.

## Chapter 18

# Nonnegative Matrix Factorization (NMF)

### Contents

---

|      |                                            |     |
|------|--------------------------------------------|-----|
| 18.1 | Nonnegative Matrix Factorization . . . . . | 366 |
| 18.2 | NMF via Multiplicative Update . . . . .    | 366 |
| 18.3 | Regularization . . . . .                   | 367 |
| 18.4 | Initialization . . . . .                   | 369 |
| 18.5 | Movie Recommender Context . . . . .        | 369 |

---

### 18.1. Nonnegative Matrix Factorization

Following from the matrix factorization via the ALS, we now consider algorithms for solving the nonnegative matrix factorization (NMF) problem:

- Given a nonnegative matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ , find nonnegative matrix factors  $\mathbf{W} \in \mathbb{R}^{M \times K}$  and  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  such that:

$$\mathbf{A} \approx \mathbf{W}\mathbf{Z}.$$

To measure the approximation, the loss to evaluate is still from the Frobenius norm of the difference between the two matrices:

$$L(\mathbf{W}, \mathbf{Z}) = \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|^2.$$

### 18.2. NMF via Multiplicative Update

Following from Section 17.2, given  $\mathbf{W} \in \mathbb{R}^{M \times K}$ , we want to update  $\mathbf{Z} \in \mathbb{R}^{K \times N}$ , the gradient with respect to  $\mathbf{Z}$  is given by Equation (17.3):

$$\frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} = 2\mathbf{W}^\top(\mathbf{W}\mathbf{Z} - \mathbf{A}) \in \mathbb{R}^{K \times N}.$$

Applying the gradient descent idea in Section 17.6, the trivial update on  $\mathbf{Z}$  can be done by

$$(\text{GD on } \mathbf{Z}) \quad \mathbf{Z} \leftarrow \mathbf{Z} - \eta \left( \frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} \right) = \mathbf{Z} - \eta \left( 2\mathbf{W}^\top \mathbf{W}\mathbf{Z} - 2\mathbf{W}^\top \mathbf{A} \right),$$

where  $\eta$  is a small positive step size. Now if we suppose a different step size for each entry of  $\mathbf{Z}$  and incorporate the constant 2 into the step size, the update can be obtained by

$$\begin{aligned} (\text{GD}' \text{ on } \mathbf{Z}) \quad \mathbf{Z}_{kn} &\leftarrow \mathbf{Z}_{kn} - \frac{\eta_{kn}}{2} \left( \frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} \right)_{kn} \\ &= \mathbf{Z}_{kn} - \eta_{kn}(\mathbf{W}^\top \mathbf{W}\mathbf{Z} - \mathbf{W}^\top \mathbf{A})_{kn}, \quad k \in [1, K], n \in [1, N], \end{aligned}$$

where  $\mathbf{Z}_{kn}$  is the  $(k, n)$ -th entry of  $\mathbf{Z}$ . Now if we rescale the step size:

$$\eta_{kn} = \frac{\mathbf{Z}_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn}},$$

then we obtain the update rule:

$$(\text{Multiplicative update on } \mathbf{Z}) \quad \mathbf{Z}_{kn} \leftarrow \mathbf{Z}_{kn} \frac{(\mathbf{W}^\top \mathbf{A})_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn}}, \quad k \in [1, K], n \in [1, N],$$

which is known as the *multiplicative update* and is first developed in (Lee and Seung, 2001) and further discussed in (Pauca et al., 2006). Analogously, the multiplicative update on  $\mathbf{W}$  can be obtained by

$$(\text{Multiplicative update on } \mathbf{W}) \quad \mathbf{W}_{mk} \leftarrow \mathbf{W}_{mk} \frac{(\mathbf{A}\mathbf{Z}^\top)_{mk}}{(\mathbf{W}\mathbf{Z}\mathbf{Z}^\top)_{mk}}, \quad m \in [1, M], k \in [1, K]. \tag{18.1}$$

**Theorem 18.1: (Convergence of Multiplicative Update)**

The loss  $L(\mathbf{W}, \mathbf{Z}) = \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|^2$  is non-increasing under the multiplicative update rules:

$$\begin{cases} \mathbf{Z}_{kn} \leftarrow \mathbf{Z}_{kn} \frac{(\mathbf{W}^\top \mathbf{A})_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn}}, & k \in [1, K], n \in [1, N]; \\ \mathbf{W}_{mk} \leftarrow \mathbf{W}_{mk} \frac{(\mathbf{A}\mathbf{Z}^\top)_{mk}}{(\mathbf{W}\mathbf{Z}\mathbf{Z}^\top)_{mk}}, & m \in [1, M], k \in [1, K]. \end{cases}$$

We refer the proof of the above theorem to (Lee and Seung, 2001). Clearly the approximations  $\mathbf{W}$  and  $\mathbf{Z}$  remain nonnegative during the updates. It is generally best to update  $\mathbf{W}$  and  $\mathbf{Z}$  “simultaneously”, instead of updating each matrix fully before the other. In this case, after updating a row of  $\mathbf{Z}$ , we update the corresponding column of  $\mathbf{W}$ . In the implementation, a small positive quantity, say the square root of the machine precision, should be added to the denominators in the approximations of  $\mathbf{W}$  and  $\mathbf{Z}$  at each iteration step. And a trivial  $\epsilon = 10^{-9}$  can do the job. The full procedure is shown in Algorithm 61.

**Algorithm 61** NMF via Multiplicative Updates

**Require:**  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

```

1: initialize $\mathbf{W} \in \mathbb{R}^{M \times K}$, $\mathbf{Z} \in \mathbb{R}^{K \times N}$ randomly with nonnegative entries.
2: choose a stop criterion on the approximation error δ ;
3: choose maximal number of iterations C ;
4: $iter = 0$;
5: while $\|\mathbf{A} - (\mathbf{W}\mathbf{Z})\|^2 > \delta$ and $iter < C$ do
6: $iter = iter + 1$;
7: for $k = 1$ to K do
8: for $n = 1$ to N do \triangleright update k -th row of \mathbf{Z}
9: $\mathbf{Z}_{kn} \leftarrow \mathbf{Z}_{kn} \frac{(\mathbf{W}^\top \mathbf{A})_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn} + \epsilon};$
10: end for
11: for $m = 1$ to M do \triangleright update k -th column of \mathbf{W}
12: $\mathbf{W}_{mk} \leftarrow \mathbf{W}_{mk} \frac{(\mathbf{A}\mathbf{Z}^\top)_{mk}}{(\mathbf{W}\mathbf{Z}\mathbf{Z}^\top)_{mk} + \epsilon};$
13: end for
14: end for
15: end while
16: Output \mathbf{W}, \mathbf{Z} ;
```

### 18.3. Regularization

Similar to the ALS with regularization in Section 17.3, recall the regularization helps employ the ALS into general matrices. We can also add a regularization in the context of NMF:

$$L(\mathbf{W}, \mathbf{Z}) = \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|^2 + \lambda_w \|\mathbf{W}\|^2 + \lambda_z \|\mathbf{Z}\|^2, \quad \lambda_w > 0, \lambda_z > 0,$$

---

**Algorithm 62** NMF via Regularized Multiplicative Updates

---

**Require:**  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 1: initialize  $\mathbf{W} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  randomly with nonnegative entries.
- 2: choose a stop criterion on the approximation error  $\delta$ ;
- 3: choose maximal number of iterations  $C$ ;
- 4: choose regularization parameter  $\lambda_z, \lambda_w$ ;
- 5:  $iter = 0$ ;
- 6: **while**  $\|\mathbf{A} - (\mathbf{W}\mathbf{Z})\|^2 > \delta$  and  $iter < C$  **do**
- 7:    $iter = iter + 1$ ;
- 8:   **for**  $k = 1$  to  $K$  **do**
- 9:     **for**  $n = 1$  to  $N$  **do** ▷ update  $k$ -th row of  $\mathbf{Z}$
- 10:        $\mathbf{Z}_{kn} \leftarrow \mathbf{Z}_{kn} \frac{(\mathbf{W}^\top \mathbf{A})_{kn} - \lambda_z \mathbf{Z}_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn} + \epsilon}$ ;
- 11:     **end for**
- 12:     **for**  $m = 1$  to  $M$  **do** ▷ update  $k$ -th column of  $\mathbf{W}$
- 13:        $\mathbf{W}_{mk} \leftarrow \mathbf{W}_{mk} \frac{(\mathbf{A}\mathbf{Z}^\top)_{mk} - \lambda_w \mathbf{W}_{mk}}{(\mathbf{W}\mathbf{Z}\mathbf{Z}^\top)_{mk} + \epsilon}$ ;
- 14:     **end for**
- 15:   **end for**
- 16: **end while**
- 17: Output  $\mathbf{W}, \mathbf{Z}$ ;

---

where the induced matrix norm is still the Frobenius norm. The gradient with respect to  $\mathbf{Z}$  given  $\mathbf{W}$  is the same as that in Equation (17.9):

$$\frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} = 2\mathbf{W}^\top(\mathbf{W}\mathbf{Z} - \mathbf{A}) + 2\lambda_z \mathbf{Z} \in \mathbb{R}^{K \times N}.$$

The trivial gradient descent update can be obtained by

$$(\text{GD on } \mathbf{Z}) \quad \mathbf{Z} \leftarrow \mathbf{Z} - \eta \left( \frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} \right) = \mathbf{Z} - \eta \left( 2\mathbf{W}^\top \mathbf{W}\mathbf{Z} - 2\mathbf{W}^\top \mathbf{A} + 2\lambda_z \mathbf{Z} \right),$$

Analogously, if we suppose a different step size for each entry of  $\mathbf{Z}$  and incorporate the constant 2 into the step size, the update can be obtained by

$$(\text{GD}' \text{ on } \mathbf{Z}) \quad \begin{aligned} \mathbf{Z}_{kn} &\leftarrow \mathbf{Z}_{kn} - \frac{\eta_{kn}}{2} \left( \frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} \right)_{kn} \\ &= \mathbf{Z}_{kn} - \eta_{kn}(\mathbf{W}^\top \mathbf{W}\mathbf{Z} - \mathbf{W}^\top \mathbf{A} + \lambda_z \mathbf{Z})_{kn}, \quad k \in [1, K], n \in [1, N], \end{aligned}$$

Now if we rescale the step size:

$$\eta_{kn} = \frac{\mathbf{Z}_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn}},$$

then we obtain the update rule:

$$(\text{Multiplicative update on } \mathbf{Z}) \quad \mathbf{Z}_{kn} \leftarrow \mathbf{Z}_{kn} \frac{(\mathbf{W}^\top \mathbf{A})_{kn} - \lambda_z \mathbf{Z}_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn}}, \quad k \in [1, K], n \in [1, N].$$

Similarly, the multiplicative update on  $\mathbf{W}$  can be obtained by

$$(\text{Multiplicative update on } \mathbf{W}) \quad \mathbf{W}_{mk} \leftarrow \mathbf{W}_{mk} \frac{(\mathbf{A}\mathbf{Z}^\top)_{mk} - \lambda_w \mathbf{W}_{mk}}{(\mathbf{W}\mathbf{Z}\mathbf{Z}^\top)_{mk}}, \quad m \in [1, M], k \in [1, K].$$

The procedure is then formulated in Algorithm 62.

#### 18.4. Initialization

In the above discussion, we initialize  $\mathbf{W}$  and  $\mathbf{Z}$  randomly. Whereas, there are also alternative strategies designed to obtain better initial estimates in the hope of converging more rapidly to a good solution (Boutsidis and Gallopoulos, 2008; Gillis, 2014). We sketch the methods as follows for a reference:

- *Clustering techniques.* Use some clustering methods on the columns of  $\mathbf{A}$ , and make the cluster means of the top  $K$  clusters as the columns of  $\mathbf{W}$ , and initialize  $\mathbf{Z}$  as a proper scaling of the cluster indicator matrix (that is,  $Z_{kn} \neq 0$  indicates  $a_n$  belongs to the  $k$ -th cluster);
- *Subset selection.* Pick  $K$  columns of  $\mathbf{A}$  and set those as the initial columns for  $\mathbf{W}$ , and analogously,  $K$  rows of  $\mathbf{A}$  are selected to form the rows of  $\mathbf{Z}$ ;
- *SVD-based.* Suppose the SVD of  $\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  where each factor  $\sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  is a rank-one matrix with possible negative values in  $\mathbf{u}_i, \mathbf{v}_i$ , and nonnegative  $\sigma_i$ . Denote  $[x]_+ = \max(x, 0)$ , we notice

$$\mathbf{u}_i \mathbf{v}_i^\top = [\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top + [-\mathbf{u}_i]_+ [-\mathbf{v}_i]_+^\top - [-\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top - [\mathbf{u}_i]_+ [-\mathbf{v}_i]_+^\top.$$

Either  $[\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top$  or  $[-\mathbf{u}_i]_+ [-\mathbf{v}_i]_+^\top$  can be selected as a column and a row in  $\mathbf{W}, \mathbf{Z}$ .

#### 18.5. Movie Recommender Context

Both the NMF and the ALS approximate the matrix and reconstruct the entries in the matrix with a set of basis vectors. The basis in the NMF is composed of vectors with nonnegative elements while the basis in the ALS can have positive or negative values. The difference then is that the NMF reconstructs each vector as a positive summation of the basis vectors with a “relative” small component in the direction of each basis vector. Whereas, In the ALS the data is modeled as a linear combination of the basis you can add or subtract vectors as needed and the components in the direction of each basis vector can be large positive values or negative values. Therefore, depending on the application one or the other factorization can be utilized to describe the data with different meanings.

In the context of a movie recommender system then the rows of  $\mathbf{W}$  represent the feature of the movies and columns of  $\mathbf{Z}$  represent the features of a user. In the NMF you can say that a movie is 0.5 comedy, 0.002 action, and 0.09 romantic. However, In the ALS you can get combinations such as 4 comedy, -0.05 comedy, and -3 drama, i.e., a positive or negative component on that feature.

The ALS and NMF are similar in the sense that the importance of each basis vector is not in a hierarchical manner. Whereas, The key difference between the ALS and the SVD is in that, in the SVD, the importance of each vector in the basis is relative to the value of the

singular value associated with that vector. For the SVD of  $\mathbf{A} = \sum_{i=1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ , this usually means that the reconstruction  $\sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top$  via the first set of basis vectors dominates and is the most used set to reconstruct data, then the second set and so on, so the basis in the SVD has an implicit hierarchy and that doesn't happen in the ALS or the NMF. Recall the low-rank approximation on the flag image in Section 17.10.1 where we find the second component  $\mathbf{w}_2 \mathbf{z}_2^\top$  via the ALS in Figure 17.6(n) plays an important role in the reconstruction of the original figure, whereas the second component  $\sigma_2 \mathbf{u}_2 \mathbf{v}_2^\top$  via the SVD in Figure 17.6(b) plays a small role in the reconstruction.

## Chapter 19

# Biconjugate Decomposition

### Contents

---

|             |                                                       |            |
|-------------|-------------------------------------------------------|------------|
| <b>19.1</b> | <b>Existence of the Biconjugate Decomposition</b>     | <b>372</b> |
| <b>19.2</b> | <b>Properties of the Biconjugate Decomposition</b>    | <b>376</b> |
| <b>19.3</b> | <b>Connection to Well-Known Decomposition Methods</b> | <b>377</b> |
| 19.3.1      | LDU Decomposition                                     | 377        |
| 19.3.2      | Cholesky Decomposition                                | 379        |
| 19.3.3      | QR Decomposition                                      | 379        |
| 19.3.4      | SVD                                                   | 380        |

---

### 19.1. Existence of the Biconjugate Decomposition

The biconjugate decomposition was proposed in (Chu et al., 1995) and discussed in (Yang, 2000). The existence of the biconjugate decomposition relies on the rank-one reduction theorem shown below. And a variety of matrix decomposition methods can be unified via this biconjugate decomposition.

#### Theorem 19.1: (Rank-One Reduction)

Any  $m \times n$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r$ , a pair of vectors  $\mathbf{x} \in \mathbb{R}^n$  and  $\mathbf{y} \in \mathbb{R}^m$  such that  $w = \mathbf{y}^\top \mathbf{A} \mathbf{x} \neq 0$ , then the matrix  $\mathbf{B} = \mathbf{A} - w^{-1} \mathbf{x} \mathbf{y}^\top \mathbf{A}$  has rank  $r - 1$  which has exactly one less than the rank of  $\mathbf{A}$ .

**Proof** [of Theorem 19.1] If we can show that the dimension of  $\mathcal{N}(\mathbf{B})$  is one larger than that of  $\mathbf{A}$ . Then this implicitly shows  $\mathbf{B}$  has rank exactly one less than the rank of  $\mathbf{A}$ .

For any vector  $\mathbf{n} \in \mathcal{N}(\mathbf{A})$ , i.e.,  $\mathbf{A}\mathbf{n} = \mathbf{0}$ , we then have  $\mathbf{B}\mathbf{n} = \mathbf{A}\mathbf{n} - w^{-1} \mathbf{x} \mathbf{y}^\top \mathbf{A}\mathbf{n} = \mathbf{0}$  which means  $\mathcal{N}(\mathbf{A}) \subseteq \mathcal{N}(\mathbf{B})$ .

Now for any vector  $\mathbf{m} \in \mathcal{N}(\mathbf{B})$ , then  $\mathbf{B}\mathbf{m} = \mathbf{A}\mathbf{m} - w^{-1} \mathbf{x} \mathbf{y}^\top \mathbf{A}\mathbf{m} = \mathbf{0}$ .

Let  $k = w^{-1} \mathbf{y}^\top \mathbf{A}\mathbf{m}$ , which is a scalar, thus  $\mathbf{A}(\mathbf{m} - k\mathbf{x}) = \mathbf{0}$ , i.e., for any vector  $\mathbf{n} \in \mathcal{N}(\mathbf{A})$ , we could find a vector  $\mathbf{m} \in \mathcal{N}(\mathbf{B})$  such that  $\mathbf{n} = (\mathbf{m} - k\mathbf{x}) \in \mathcal{N}(\mathbf{A})$ . Note that  $\mathbf{A}\mathbf{x} \neq \mathbf{0}$  from the definition of  $w$ . Thus, the null space of  $\mathbf{B}$  is therefore obtained from the null space of  $\mathbf{A}$  by adding  $\mathbf{x}$  to its basis which will increase the order of the space by 1. Thus the dimension of  $\mathcal{N}(\mathbf{A})$  is smaller than the dimension of  $\mathcal{N}(\mathbf{B})$  by 1 which completes the proof. ■

Suppose matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has rank  $r$ , we can define a rank reducing process to generate a sequence of Wedderburn matrices  $\{\mathbf{A}_k\}$ :

$$\mathbf{A}_1 = \mathbf{A}, \quad \text{and} \quad \mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k,$$

where  $\mathbf{x}_k \in \mathbb{R}^n$  and  $\mathbf{y}_k \in \mathbb{R}^m$  are any vectors satisfying  $w_k = \mathbf{y}_k^\top \mathbf{A} \mathbf{x}_k \neq 0$ . The sequence will terminate in  $r$  steps since the rank of  $\mathbf{A}_k$  decreases by exactly one at each step. Write out the sequence:

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{A}, \\ \mathbf{A}_1 - \mathbf{A}_2 &= w_1^{-1} \mathbf{A}_1 \mathbf{x}_1 \mathbf{y}_1^\top \mathbf{A}_1, \\ \mathbf{A}_2 - \mathbf{A}_3 &= w_2^{-1} \mathbf{A}_2 \mathbf{x}_2 \mathbf{y}_2^\top \mathbf{A}_2, \\ \mathbf{A}_3 - \mathbf{A}_4 &= w_3^{-1} \mathbf{A}_3 \mathbf{x}_3 \mathbf{y}_3^\top \mathbf{A}_3, \\ &\vdots = \vdots \\ \mathbf{A}_{r-1} - \mathbf{A}_r &= w_{r-1}^{-1} \mathbf{A}_{r-1} \mathbf{x}_{r-1} \mathbf{y}_{r-1}^\top \mathbf{A}_{r-1}, \\ \mathbf{A}_r - \mathbf{0} &= w_r^{-1} \mathbf{A}_r \mathbf{x}_r \mathbf{y}_r^\top \mathbf{A}_r. \end{aligned}$$

By adding the sequence we will get

$$(\mathbf{A}_1 - \mathbf{A}_2) + (\mathbf{A}_2 - \mathbf{A}_3) + \dots + (\mathbf{A}_{r-1} - \mathbf{A}_r) + (\mathbf{A}_r - \mathbf{0}) = \mathbf{A} = \sum_{i=1}^r w_i^{-1} \mathbf{A}_i \mathbf{x}_i \mathbf{y}_i^\top \mathbf{A}_i.$$

**Theorem 19.2: (Biconjugate Decomposition: Form 1)**

This equality from rank-reducing process implies the following matrix decomposition

$$\mathbf{A} = \Phi \Omega^{-1} \Psi^\top,$$

where  $\Omega = \text{diag}(w_1, w_2, \dots, w_r)$ ,  $\Phi = [\phi_1, \phi_2, \dots, \phi_r] \in \mathbb{R}^{m \times r}$  and  $\Psi = [\psi_1, \psi_2, \dots, \psi_r]$  with

$$\phi_k = \mathbf{A}_k \mathbf{x}_k, \quad \text{and} \quad \psi_k = \mathbf{A}_k^\top \mathbf{y}_k.$$

Obviously, different choices of  $\mathbf{x}_k$ 's and  $\mathbf{y}_k$ 's will result in different factorizations. So this factorization is rather general and we will show its connection to some well-known decomposition methods.

**Remark 19.3**

For the vectors  $\mathbf{x}_k, \mathbf{y}_k$  in the Wedderburn sequence, we have the following property

$$\begin{aligned} \mathbf{x}_k &\in \mathcal{N}(\mathbf{A}_{k+1}) \perp \mathcal{C}(\mathbf{A}_{k+1}^\top), \\ \mathbf{y}_k &\in \mathcal{N}(\mathbf{A}_{k+1}^\top) \perp \mathcal{C}(\mathbf{A}_{k+1}). \end{aligned}$$

**Lemma 19.4: (General Term Formula of Wedderburn Sequence: V1)**

For each matrix with  $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$ , then  $\mathbf{A}_{k+1}$  can be written as

$$\mathbf{A}_{k+1} = \mathbf{A} - \sum_{i=1}^k w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A},$$

where

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{w_i} \mathbf{u}_i, \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\mathbf{y}_k^\top \mathbf{A} \mathbf{u}_i}{w_i} \mathbf{v}_i.$$

The proof of this lemma is provided in Appendix G. We notice that  $w_i = \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k$  in the general term formula is related to  $\mathbf{A}_k$ . So it's not the true general term formula. We will write  $w_i$  to be related to  $\mathbf{A}$  rather than  $\mathbf{A}_k$  later. From the general term formula of Wedderburn sequence, we have

$$\begin{aligned} \mathbf{A}_{k+1} &= \mathbf{A} - \sum_{i=1}^k w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} \\ \mathbf{A}_k &= \mathbf{A} - \sum_{i=1}^{k-1} w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}. \end{aligned}$$

Thus,  $\mathbf{A}_{k+1} - \mathbf{A}_k = -w_k^{-1} \mathbf{A} \mathbf{u}_k \mathbf{v}_k^\top \mathbf{A}$ . Since we define the sequence by  $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$ . We then find  $w_k^{-1} \mathbf{A} \mathbf{u}_k \mathbf{v}_k^\top \mathbf{A} = w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$ . It is trivial to see

$$\begin{aligned}\mathbf{A} \mathbf{u}_k &= \mathbf{A}_k \mathbf{x}_k, \\ \mathbf{v}_k^\top \mathbf{A} &= \mathbf{y}_k^\top \mathbf{A}_k.\end{aligned}\tag{19.1}$$

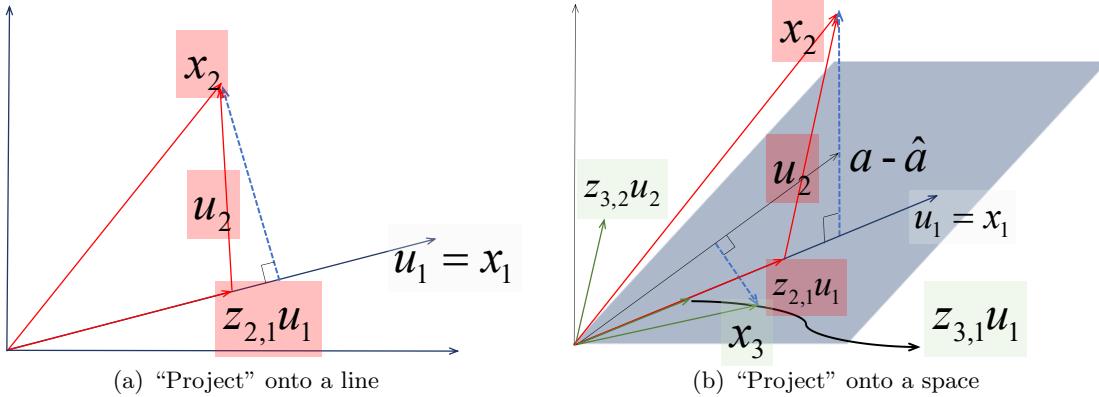
Let  $z_{k,i} = \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{w_i}$  which is a scalar. From the definition of  $\mathbf{u}_k$  and  $\mathbf{v}_k$  in the above lemma, then

- $\mathbf{u}_1 = \mathbf{x}_1$ ;
- $\mathbf{u}_2 = \mathbf{x}_2 - z_{2,1} \mathbf{u}_1$ ;
- $\mathbf{u}_3 = \mathbf{x}_3 - z_{3,1} \mathbf{u}_1 - z_{3,2} \mathbf{u}_2$ ;
- ...

This process is just similar to the Gram-Schmidt process. But now, we do not project  $\mathbf{x}_2$  onto  $\mathbf{x}_1$  with the smallest distance. The vector of  $\mathbf{x}_2$  along  $\mathbf{x}_1$  is now defined by  $z_{2,1}$ . This process is shown in Figure 19.1. In Figure 19.1(a),  $\mathbf{u}_2$  is not perpendicular to  $\mathbf{u}_1$ . But  $\mathbf{u}_2$  does not lie on the same line of  $\mathbf{u}_1$  so that  $\mathbf{u}_1, \mathbf{u}_2$  still could span a  $\mathbb{R}^2$  subspace. Similarly, in Figure 19.1(b),  $\mathbf{u}_3 = \mathbf{x}_3 - z_{3,1} \mathbf{u}_1 - z_{3,2} \mathbf{u}_2$  does not lie in the space spanned by  $\mathbf{u}_1, \mathbf{u}_2$  so that  $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$  could still span a  $\mathbb{R}^3$  subspace.

A moment of reflexion would reveal that the span of  $\mathbf{x}_2, \mathbf{x}_1$  is the same as the span of  $\mathbf{u}_2, \mathbf{u}_1$ . Similarly for  $\mathbf{v}_i$ 's. We have the following property:

$$\begin{cases} \text{span}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_j\} = \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_j\}; \\ \text{span}\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_j\} = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j\}. \end{cases}\tag{19.2}$$



**Figure 19.1:** “Project” a vector onto a line and onto a space.

Further, from the rank-reducing property in the Wedderburn sequence, we have

$$\begin{cases} \mathcal{C}(\mathbf{A}_1) \supset \mathcal{C}(\mathbf{A}_2) \supset \mathcal{C}(\mathbf{A}_3) \supset \dots; \\ \mathcal{N}(\mathbf{A}_1^\top) \subset \mathcal{N}(\mathbf{A}_2^\top) \subset \mathcal{N}(\mathbf{A}_3^\top) \subset \dots. \end{cases}$$

Since  $\mathbf{y}_k \in \mathcal{N}(\mathbf{A}_{k+1}^\top)$ , it then follows that  $\mathbf{y}_j \in \mathcal{N}(\mathbf{A}_{k+1}^\top)$  for all  $j < k + 1$ , i.e.,  $\mathbf{A}_{k+1}^\top \mathbf{y}_j = \mathbf{0}$  for all  $j < k + 1$ . Which also holds true for  $\mathbf{x}_{k+1}^\top \mathbf{A}_{k+1}^\top \mathbf{y}_j = 0$  for all  $j < k + 1$ . From Equation (19.1), we also have  $\mathbf{u}_{k+1}^\top \mathbf{A}^\top \mathbf{y}_j = 0$  for all  $j < k + 1$ . Following from Equation (19.2),

we obtain

$$\mathbf{v}_j^\top \mathbf{A} \mathbf{u}_{k+1} = 0 \text{ for all } j < k + 1.$$

Similarly, we can prove

$$\mathbf{v}_{k+1}^\top \mathbf{A} \mathbf{u}_j = 0 \text{ for all } j < k + 1.$$

Moreover, we defined  $w_k = \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k$ . By Equation (19.1), we can write the  $w_k$  as:

$$\begin{aligned} w_k &= \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k \\ &= \mathbf{v}_k^\top \mathbf{A} \mathbf{x}_k \\ &= \mathbf{v}_k^\top \mathbf{A} (\mathbf{u}_k + \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{w_i} \mathbf{u}_i) \quad (\text{by the definition of } \mathbf{u}_k \text{ in Lemma 19.4}) \\ &= \mathbf{v}_k^\top \mathbf{A} \mathbf{u}_k, \quad (\text{by } \mathbf{v}_k^\top \mathbf{A} \mathbf{u}_j = 0 \text{ for all } j < k) \end{aligned}$$

which can be used to substitute the  $w_k$  in Lemma 19.4 and we then have the full version of the general term formula of the Wedderburn sequence such that the formula does not depend on  $\mathbf{A}_k$ 's (in the form of  $w_k$ 's) with

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{\mathbf{v}_i^\top \mathbf{A} \mathbf{u}_i} \mathbf{u}_i, \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\mathbf{y}_k^\top \mathbf{A} \mathbf{u}_i}{\mathbf{v}_i^\top \mathbf{A} \mathbf{u}_i} \mathbf{v}_i. \quad (19.3)$$

**Gram-Schmidt Process from Wedderburn Sequence:** If  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r] \in \mathbb{R}^{n \times r}$ ,  $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r] \in \mathbb{R}^{m \times r}$  effects a rank-reducing process for  $\mathbf{A}$ . Let  $\mathbf{A}$  be the identity matrix and  $(\mathbf{X}, \mathbf{Y})$  are identical and contain the vectors for which an orthogonal basis is desired, then  $(\mathbf{U} = \mathbf{V})$  give the resultant orthogonal basis.

This form of  $\mathbf{u}_k$  and  $\mathbf{v}_k$  in Equation (19.3) is very close to the projection to the perpendicular space of the Gram-Schmidt process in Equation (3.1). We then define  $\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{y}^\top \mathbf{A} \mathbf{x}$  to explicitly mimic the form of projection in Equation (3.1). We formulate the results so far in the following lemma which can help us have a clear vision about what we have been working on and we will use these results extensively in the sequel:

### Lemma 19.5: (Properties of Wedderburn Sequence)

For each matrix with  $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$ , then  $\mathbf{A}_{k+1}$  can be written as

$$\mathbf{A}_{k+1} = \mathbf{A} - \sum_{i=1}^k w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A},$$

where

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{x}_k, \mathbf{v}_i \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{u}_i, \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{u}_i, \mathbf{y}_k \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{v}_i. \quad (19.4)$$

Further, we have the following properties:

$$\begin{aligned} \mathbf{A}\mathbf{u}_k &= \mathbf{A}_k\mathbf{x}_k, \\ \mathbf{v}_k^\top \mathbf{A} &= \mathbf{y}_k^\top \mathbf{A}_k. \end{aligned} \quad (19.5)$$

$$\langle \mathbf{u}_k, \mathbf{v}_j \rangle = \langle \mathbf{u}_j, \mathbf{v}_k \rangle = 0 \text{ for all } j < k. \quad (19.6)$$

$$w_k = \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k = \langle \mathbf{u}_k, \mathbf{v}_k \rangle \quad (19.7)$$

By substituting Equation (19.5) into Form 1 of biconjugate decomposition, and using Equation (19.7) which implies  $w_k = \mathbf{v}_k^\top \mathbf{A}\mathbf{u}_k$ , we have the Form 2 and Form 3 of this decomposition:

### Theorem 19.6: (Biconjugate Decomposition: Form 2 and Form 3)

The equality from rank-reducing process implies the following matrix decomposition

$$\mathbf{A} = \mathbf{A}\mathbf{U}_r \boldsymbol{\Omega}_r^{-1} \mathbf{V}_r^\top \mathbf{A},$$

where  $\boldsymbol{\Omega}_r = \text{diag}(w_1, w_2, \dots, w_r)$ ,  $\mathbf{U}_r = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \in \mathbb{R}^{m \times r}$  and  $\mathbf{V}_r = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r] \in \mathbb{R}^{n \times r}$  with

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{x}_k, \mathbf{v}_i \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{u}_i, \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{u}_i, \mathbf{y}_k \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{v}_i. \quad (19.8)$$

And also the following decomposition

$$\mathbf{V}_\gamma^\top \mathbf{A}\mathbf{U}_\gamma = \boldsymbol{\Omega}_\gamma, \quad (19.9)$$

where  $\boldsymbol{\Omega}_\gamma = \text{diag}(w_1, w_2, \dots, w_\gamma)$ ,  $\mathbf{U}_\gamma = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_\gamma] \in \mathbb{R}^{m \times \gamma}$  and  $\mathbf{V}_\gamma = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_\gamma] \in \mathbb{R}^{n \times \gamma}$ . Note the difference between the subscripts  $r$  and  $\gamma$  we used here with  $\gamma \leq r$ .

We notice that, in these two forms of biconjugate decomposition, they are independent of the Wedderburn matrices  $\{\mathbf{A}_k\}$ .

**A word on the notation:** we will use the subscript to indicate the dimension of the matrix avoiding confusion in the sequel, e.g., the  $r, \gamma$  in the above theorem.

## 19.2. Properties of the Biconjugate Decomposition

### Corollary 19.1: (Connection of $\mathbf{U}_\gamma$ and $\mathbf{X}_\gamma$ )

If  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma) \in \mathbb{R}^{n \times \gamma} \times \mathbb{R}^{m \times \gamma}$  effects a rank-reducing process for  $\mathbf{A}$ , then there are unique unit upper triangular matrices  $\mathbf{R}_\gamma^{(x)} \in \mathbb{R}^{\gamma \times \gamma}$  and  $\mathbf{R}_\gamma^{(y)} \in \mathbb{R}^{\gamma \times \gamma}$  such that

$$\mathbf{X}_\gamma = \mathbf{U}_\gamma \mathbf{R}_\gamma^{(x)}, \quad \text{and} \quad \mathbf{Y}_\gamma = \mathbf{V}_\gamma \mathbf{R}_\gamma^{(y)},$$

where  $\mathbf{U}_\gamma$  and  $\mathbf{V}_\gamma$  are matrices with columns resulting from the Wedderburn sequence as in Equation (19.9).

**Proof** [of Corollary 19.1] The proof is trivial from the definition of  $\mathbf{u}_k$  and  $\mathbf{v}_k$  in Equation (19.4) or Equation (19.8) by setting the  $j$ -th column of  $\mathbf{R}_\gamma^{(x)}$  and  $\mathbf{R}_\gamma^{(y)}$  as

$$\left[ \frac{\langle \mathbf{x}_j, \mathbf{v}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{v}_1 \rangle}, \frac{\langle \mathbf{x}_j, \mathbf{v}_2 \rangle}{\langle \mathbf{u}_2, \mathbf{v}_2 \rangle}, \dots, \frac{\langle \mathbf{x}_j, \mathbf{v}_{j-1} \rangle}{\langle \mathbf{u}_{j-1}, \mathbf{v}_{j-1} \rangle}, 1, 0, 0, \dots, 0 \right]^\top,$$

and

$$\left[ \frac{\langle \mathbf{u}_1, \mathbf{y}_j \rangle}{\langle \mathbf{u}_1, \mathbf{v}_1 \rangle}, \frac{\langle \mathbf{u}_2, \mathbf{y}_j \rangle}{\langle \mathbf{u}_2, \mathbf{v}_2 \rangle}, \dots, \frac{\langle \mathbf{u}_{j-1}, \mathbf{y}_j \rangle}{\langle \mathbf{u}_{j-1}, \mathbf{v}_{j-1} \rangle}, 1, 0, 0, \dots, 0 \right]^\top.$$

This completes the proof. ■

The  $(\mathbf{U}_\gamma, \mathbf{V}_\gamma) \in \mathbb{R}^{m \times \gamma} \times \mathbb{R}^{n \times \gamma}$  in Theorem 19.6 is called a **biconjugate pair** with respect to  $\mathbf{A}$  if  $\Omega_\gamma$  is nonsingular and diagonal. And let  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma) \in \mathbb{R}^{n \times \gamma} \times \mathbb{R}^{m \times \gamma}$  effect a rank-reducing process for  $\mathbf{A}$ , then  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$  is said to be **biconjugatable** and **biconjugated into a biconjugate pair** of matrices  $(\mathbf{U}_\gamma, \mathbf{V}_\gamma)$ , if there exist unit upper triangular matrices  $\mathbf{R}_\gamma^{(x)}, \mathbf{R}_\gamma^{(y)}$  such that  $\mathbf{X}_\gamma = \mathbf{U}_\gamma \mathbf{R}_\gamma^{(x)}$  and  $\mathbf{Y}_\gamma = \mathbf{V}_\gamma \mathbf{R}_\gamma^{(y)}$ .

### 19.3. Connection to Well-Known Decomposition Methods

#### 19.3.1 LDU Decomposition

**Theorem 19.1: (LDU, Chu et al. (1995) Theorem 2.4)**

If  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma) \in \mathbb{R}^{n \times \gamma} \times \mathbb{R}^{m \times \gamma}$  and  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $\gamma$  in  $\{1, 2, \dots, r\}$ . Then  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$  can be biconjugated if and only if  $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma$  has an LDU decomposition.

**Proof** [of Theorem 19.1] Suppose  $\mathbf{X}_\gamma$  and  $\mathbf{Y}_\gamma$  are biconjugatable, then, there exists a unit upper triangular matrices  $\mathbf{R}_\gamma^{(x)}$  and  $\mathbf{R}_\gamma^{(y)}$  such that  $\mathbf{X}_\gamma = \mathbf{U}_\gamma \mathbf{R}_\gamma^{(x)}$ ,  $\mathbf{Y}_\gamma = \mathbf{V}_\gamma \mathbf{R}_\gamma^{(y)}$  and  $\mathbf{V}_\gamma^\top \mathbf{A} \mathbf{U}_\gamma = \Omega_\gamma$  is a nonsingular diagonal matrix. Then, it follows that

$$\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma = \mathbf{R}_\gamma^{(y)\top} \mathbf{V}_\gamma^\top \mathbf{A} \mathbf{U}_\gamma \mathbf{R}_\gamma^{(x)} = \mathbf{R}_\gamma^{(y)\top} \Omega_\gamma \mathbf{R}_\gamma^{(x)}$$

is the unique unit triangular LDU decomposition of  $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma$ . This form above can be seen as the **fourth form of biconjugate decomposition**, thus we put the proof into a graybox.

Conversely, suppose  $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma = \mathbf{R}_2^\top \mathbf{D} \mathbf{R}_1$  is an LDU decomposition with both  $\mathbf{R}_1$  and  $\mathbf{R}_2$  being unit upper triangular matrices. Then since  $\mathbf{R}_1^{-1}$  and  $\mathbf{R}_2^{-1}$  are also unit upper triangular matrices, and  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$  biconjugates into  $(\mathbf{X}_\gamma \mathbf{R}_1^{-1}, \mathbf{Y}_\gamma \mathbf{R}_2^{-1})$ . ■

**Corollary 19.2: (Determinant)**

Suppose  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma) \in \mathbb{R}^{n \times \gamma} \times \mathbb{R}^{m \times \gamma}$  are biconjugatable. Then

$$\det(\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma) = \prod_{i=1}^{\gamma} w_i.$$

**Proof** [of Corollary 19.2] By Theorem 19.1, since  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$  are biconjugatable, then there are unit upper triangular matrices  $\mathbf{R}_\gamma^{(x)}$  and  $\mathbf{R}_\gamma^{(y)}$  such that  $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma = \mathbf{R}_\gamma^{(y)\top} \boldsymbol{\Omega}_\gamma \mathbf{R}_\gamma^{(x)}$ . The determinant is just product of the trace. ■

**Lemma 19.3: (Biconjugatable in Principal Minors)**

Let  $r = \text{rank}(\mathbf{A}) \geq \gamma$  with  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . In the Wedderburn sequence, take  $\mathbf{x}_i$  as the  $i$ -th basis in  $\mathbb{R}^n$  for  $i \in \{1, 2, \dots, \gamma\}$  (i.e.,  $\mathbf{x}_i = \mathbf{e}_i \in \mathbb{R}^n$ ) and  $\mathbf{y}_i$  as the  $i$ -th basis in  $\mathbb{R}^m$  for  $i \in \{1, 2, \dots, \gamma\}$  (i.e.,  $\mathbf{y}_i = \mathbf{e}_i \in \mathbb{R}^m$ ). That is  $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma$  is the leading principal submatrix of  $\mathbf{A}$ , i.e.,  $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma = \mathbf{A}_{1:\gamma, 1:\gamma}$ . Then,  $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$  is biconjugatable if and only if the  $\gamma$ -th leading principal minor of  $\mathbf{A}$  is nonzero. In this case, the  $\gamma$ -th leading principal minor of  $\mathbf{A}$  is given by  $\prod_{i=1}^{\gamma} w_i$ .

**Proof** [of Lemma 19.3] The proof is trivial that the  $\gamma$ -th leading principal minor of  $\mathbf{A}$  is nonzero will imply that  $w_i \neq 0$  for all  $i \leq \gamma$ . Thus the Wedderburn sequence can be successfully obtained. The converse holds since Corollary 19.2 implies  $\det(\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma)$  is nonzero. ■

We thus finally come to the LDU decomposition for square matrices.

**Theorem 19.4: (LDU: Biconjugate Decomposition for Square Matrices)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $(\mathbf{I}_n, \mathbf{I}_n)$  is biconjugatable if and only if all the leading principal minors of  $\mathbf{A}$  are nonzero. In this case,  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{V}_n^{-\top} \boldsymbol{\Omega}_n \mathbf{U}_n^{-1} = \mathbf{L} \mathbf{D} \mathbf{U},$$

where  $\boldsymbol{\Omega}_n = \mathbf{D}$  is a diagonal matrix with nonzero values on the diagonal,  $\mathbf{V}_n^{-\top} = \mathbf{L}$  is a unit lower triangular matrix and  $\mathbf{U}_n^{-1} = \mathbf{U}$  is a unit upper triangular matrix.

**Proof** [of Theorem 19.4] From Lemma 19.3, it is trivial that  $(\mathbf{I}_n, \mathbf{I}_n)$  is biconjugatable. From Corollary 19.1, we have  $\mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{I}_n$  and  $\mathbf{I}_n = \mathbf{V}_n \mathbf{R}_n^{(y)}$ , thus  $\mathbf{R}_n^{(x)} = \mathbf{U}_n^{-1}$  and  $\mathbf{R}_n^{(y)} = \mathbf{V}_n^{-1}$  are well defined and we complete the proof. ■

### 19.3.2 Cholesky Decomposition

For symmetric and positive definite, the leading principal minors are positive for sure. The proof is provided in Section 2.3.

#### Theorem 19.5: (Cholesky: Biconjugate Decomposition for PD Matrices)

For any symmetric and positive definite matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , the Cholesky decomposition of  $\mathbf{A}$  can be obtained from the Wedderburn sequence applied to  $(\mathbf{I}_n, \mathbf{I}_n)$  as  $(\mathbf{X}_n, \mathbf{Y}_n)$ . In this case,  $\mathbf{A}$  can be factored as

$$\mathbf{A} = \mathbf{U}_n^{-\top} \boldsymbol{\Omega}_n \mathbf{U}_n^{-1} = (\mathbf{U}_n^{-\top} \boldsymbol{\Omega}_n^{1/2})(\boldsymbol{\Omega}_n^{1/2} \mathbf{U}_n^{-1}) = \mathbf{R}^\top \mathbf{R},$$

where  $\boldsymbol{\Omega}_n$  is a diagonal matrix with positive values on the diagonal, and  $\mathbf{U}^{-1}$  is a unit upper triangular matrix.

**Proof** [of Theorem 19.5] Since the leading principal minors of positive definite matrices are positive,  $w_i > 0$  for all  $i \in \{1, 2, \dots, n\}$ . It can be easily verified via the LDU from biconjugation decomposition and the symmetric property of  $\mathbf{A}$  that  $\mathbf{A} = \mathbf{U}_n^{-\top} \boldsymbol{\Omega}_n \mathbf{U}_n^{-1}$ . And since  $w_i$ 's are positive, thus  $\boldsymbol{\Omega}_n$  is positive definite and can be factored as  $\boldsymbol{\Omega}_n = \boldsymbol{\Omega}_n^{1/2} \boldsymbol{\Omega}_n^{1/2}$  which implies  $\boldsymbol{\Omega}_n^{1/2} \mathbf{U}_n^{-1}$  is the Cholesky factor. ■

### 19.3.3 QR Decomposition

Without loss of generality, we shall assume that  $\mathbf{A} \in \mathbb{R}^{n \times n}$  has full column rank so that the columns of  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  with  $\mathbf{Q}, \mathbf{R} \in \mathbb{R}^{n \times n}$

#### Theorem 19.6: (QR: Biconjugate Decomposition for Nonsingular Matrices)

For any nonsingular matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , the QR decomposition of  $\mathbf{A}$  can be obtained from the Wedderburn sequence applied to  $(\mathbf{I}_n, \mathbf{A})$  as  $(\mathbf{X}_n, \mathbf{Y}_n)$ . In this case,  $\mathbf{A}$  can be factored as

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

where  $\mathbf{Q} = \mathbf{V}_n \boldsymbol{\Omega}_n^{-1/2}$  is an orthogonal matrix and  $\mathbf{R} = \boldsymbol{\Omega}_n^{1/2} \mathbf{R}_n^{(x)}$  is an upper triangular matrix with Form 4 in Theorem 19.1 and let  $\gamma = n$

$$\mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n = \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \boldsymbol{\Omega}_n \mathbf{R}_n^{(x)}.$$

where we set  $\gamma = n$  since  $\gamma$  is any value that  $\gamma \leq r$  and the rank  $r = n$ .

**Proof** [of Theorem 19.6] Since  $(\mathbf{X}_n, \mathbf{Y}_n) = (\mathbf{I}_n, \mathbf{A})$ . Then By Theorem 19.1, we have the decomposition

$$\mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n = \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \boldsymbol{\Omega}_n \mathbf{R}_n^{(x)}.$$

Substitute  $(\mathbf{I}_n, \mathbf{A})$  into above decomposition, we have

$$\begin{aligned}\mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n &= \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \boldsymbol{\Omega}_n \mathbf{R}_n^{(x)} \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}_n^{(y)\top} \boldsymbol{\Omega}_n \mathbf{R}_n^{(x)} \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}_1^\top \boldsymbol{\Omega}_n \mathbf{R}_1 \quad (\mathbf{A}^\top \mathbf{A} \text{ is symmetric and let } \mathbf{R}_1 = \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)}) \\ \mathbf{A}^\top \mathbf{A} &= (\mathbf{R}_1^\top \boldsymbol{\Omega}_n^{1/2\top})(\boldsymbol{\Omega}_n^{1/2} \mathbf{R}_1) \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}^\top \mathbf{R}. \quad (\text{Let } \mathbf{R} = \boldsymbol{\Omega}_n^{1/2} \mathbf{R}_1)\end{aligned}\tag{19.10}$$

To see why  $\boldsymbol{\Omega}_n$  can be factored as  $\boldsymbol{\Omega}_n = \boldsymbol{\Omega}_n^{1/2\top} \boldsymbol{\Omega}_n^{1/2}$ . Suppose  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ . We obtain  $w_i = \mathbf{a}_i^\top \mathbf{a}_i > 0$  since  $\mathbf{A}$  is nonsingular. Thus  $\boldsymbol{\Omega}_n = \text{diag}(w_1, w_2, \dots, w_n)$  is positive definite and can be factored as

$$\boldsymbol{\Omega}_n = \boldsymbol{\Omega}_n^{1/2} \boldsymbol{\Omega}_n^{1/2} = \boldsymbol{\Omega}_n^{1/2\top} \boldsymbol{\Omega}_n^{1/2}.\tag{19.11}$$

By  $\mathbf{X}_\gamma = \mathbf{U}_\gamma \mathbf{R}_\gamma^{(x)}$  in Theorem 19.1 for all  $\gamma \in \{1, 2, \dots, n\}$ , we have

$$\begin{aligned}\mathbf{X}_n &= \mathbf{U}_n \mathbf{R}_1 \\ \mathbf{I}_n &= \mathbf{U}_n \mathbf{R}_1, \quad (\text{Since } \mathbf{X}_n = \mathbf{I}_n) \\ \mathbf{U}_n &= \mathbf{R}_1^{-1}\end{aligned}$$

By  $\mathbf{Y}_\gamma = \mathbf{V}_\gamma \mathbf{R}_\gamma^{(y)}$  in Theorem 19.1 for all  $\gamma \in \{1, 2, \dots, n\}$ , we have

$$\begin{aligned}\mathbf{Y}_n &= \mathbf{V}_n \mathbf{R}_1 \\ \mathbf{A} &= \mathbf{V}_n \mathbf{R}_1, \quad (\mathbf{A} = \mathbf{Y}_n) \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}_1^\top \mathbf{V}_n^\top \mathbf{V}_n \mathbf{R}_1 \\ \mathbf{R}_1^\top \boldsymbol{\Omega}_n \mathbf{R}_1 &= \mathbf{R}_1^\top \mathbf{V}_n^\top \mathbf{V}_n \mathbf{R}_1, \quad (\text{From Equation (19.10)}) \\ (\mathbf{R}_1^\top \boldsymbol{\Omega}_n^{1/2\top})(\boldsymbol{\Omega}_n^{1/2} \mathbf{R}_1) &= (\mathbf{R}_1^\top \boldsymbol{\Omega}_n^{1/2\top} \boldsymbol{\Omega}_n^{-1/2\top}) \mathbf{V}_n^\top \mathbf{V}_n (\boldsymbol{\Omega}_n^{-1/2} \boldsymbol{\Omega}_n^{1/2} \mathbf{R}_1), \quad (\text{From Equation (19.11)}) \\ \mathbf{R}^\top \mathbf{R} &= \mathbf{R}^\top (\boldsymbol{\Omega}_n^{-1/2\top} \mathbf{V}_n^\top) (\mathbf{V}_n \boldsymbol{\Omega}_n^{-1/2}) \mathbf{R}\end{aligned}\tag{19.12}$$

Thus,  $\mathbf{Q} = \mathbf{V}_n \boldsymbol{\Omega}_n^{-1/2}$  is an orthogonal matrix. ■

### 19.3.4 SVD

To differentiate the notation, let  $\mathbf{A} = \mathbf{U}^{\text{svd}} \boldsymbol{\Sigma}^{\text{svd}} \mathbf{V}^{\text{svd}\top}$  be the SVD of  $\mathbf{A}$  where  $\mathbf{U}^{\text{svd}} = [\mathbf{u}_1^{\text{svd}}, \mathbf{u}_2^{\text{svd}}, \dots, \mathbf{u}_n^{\text{svd}}]$ ,  $\mathbf{V}^{\text{svd}} = [\mathbf{v}_1^{\text{svd}}, \mathbf{v}_2^{\text{svd}}, \dots, \mathbf{v}_n^{\text{svd}}]$  and  $\boldsymbol{\Sigma}^{\text{svd}} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ . Without loss of generality, we assume  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $\text{rank}(\mathbf{A}) = n$ . Readers can prove the equivalence for  $\mathbf{A} \in \mathbb{R}^{m \times n}$ .

If  $\mathbf{X}_n = \mathbf{V}^{\text{svd}}$ ,  $\mathbf{Y}_n = \mathbf{U}^{\text{svd}}$  effects a rank-reducing process for  $\mathbf{A}$ . From the definition of  $\mathbf{u}_k$  and  $\mathbf{v}_k$  in Equation (19.4) or Equation (19.8), we have

$$\mathbf{u}_k = \mathbf{v}_k^{\text{svd}} \quad \text{and} \quad \mathbf{v}_k = \mathbf{u}_k^{\text{svd}} \quad \text{and} \quad w_k = \mathbf{y}_k^\top \mathbf{A} \mathbf{x}_k = \sigma_k.$$

That is  $\mathbf{V}_n = \mathbf{U}^{\text{svd}}$ ,  $\mathbf{U}_n = \mathbf{V}^{\text{svd}}$ , and  $\boldsymbol{\Omega}_n = \boldsymbol{\Sigma}^{\text{svd}}$ , where we set  $\gamma = n$  since  $\gamma$  is any value that  $\gamma \leq r$  and the rank  $r = n$ .

By  $\mathbf{X}_n = \mathbf{U}_n \mathbf{R}_n^{(x)}$  in Theorem 19.1, we have

$$\mathbf{X}_n = \mathbf{U}_n \mathbf{R}_n^{(x)} \quad \xrightarrow{\text{leads to}} \quad \mathbf{V}^{\text{svd}} = \mathbf{V}^{\text{svd}} \mathbf{R}_n^{(x)} \quad \xrightarrow{\text{leads to}} \quad \mathbf{I}_n = \mathbf{R}_n^{(x)}$$

By  $\mathbf{Y}_n = \mathbf{V}_n \mathbf{R}_n^{(y)}$  in Theorem 19.1, we have

$$\mathbf{Y}_n = \mathbf{V}_n \mathbf{R}_n^{(y)} \quad \xrightarrow{\text{leads to}} \quad \mathbf{U}^{\text{svd}} = \mathbf{U}^{\text{svd}} \mathbf{R}_n^{(y)} \quad \xrightarrow{\text{leads to}} \quad \mathbf{I}_n = \mathbf{R}_n^{(y)}$$

Again, from Theorem 19.1 and let  $\gamma = n$ , we have

$$\mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n = \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \boldsymbol{\Omega}_n \mathbf{R}_n^{(x)}.$$

That is

$$\mathbf{U}^{\text{svd}\top} \mathbf{A} \mathbf{V}^{\text{svd}} = \boldsymbol{\Sigma}^{\text{svd}},$$

which is exactly the form of SVD and we prove the equivalence of SVD and biconjugate decomposition when the Wedderburn sequence is applied to  $(\mathbf{V}^{\text{svd}}, \mathbf{U}^{\text{svd}})$  as  $(\mathbf{X}_n, \mathbf{Y}_n)$ .

# Chapter 20

# Modern Applications

## Contents

---

|      |                                                            |     |
|------|------------------------------------------------------------|-----|
| 20.1 | Low-Rank Neural Networks . . . . .                         | 383 |
| 20.2 | One More Step: Adding a Nonlinear Function Layer . . . . . | 385 |

---

## 20.1. Low-Rank Neural Networks

We start with the basic LeNet5 neural network to illustrate this low-rank neural networks idea. But we modify the fully connected layers as *LenetModified* as shown below in the bluebox. Also, we notice that a layer with input feature 120, output feature 100 is just a matrix of size  $120 \times 100$ . If we put a "regularize" on the matrix, such as the rank of the matrix is 50. All in all, a matrix with  $120 \times 100$  or a matrix via matrix multiplication of two matrices  $120 \times 50, 50 \times 100$  are similar in some sense. The matrix multiplication result also has a size of  $120 \times 100$ . Thus, we will have a low-rank version of the fully connected layer. We call this *LeNetDecom* structured. The fully connected layers in *LenetModified* and *LenetDecom* are shown as follows:

```
LenetModified{
 Convolutional Layers Omitted;
 (Fully Connected Layers):
 (0): Linear(in_features=120, out_features=100)
 (1): Tanh()
 (2): Linear(in_features=100, out_features=80)
 (3): Tanh()
 (4): Linear(in_features=80, out_features=60)
 (5): Tanh()
 (6): Linear(in_features=60, out_features=40)
 (7): Tanh()
 (8): Linear(in_features=40, out_features=10)
 (9): LogSoftmax()
}
```

```
LenetDecom{
 Convolutional Layers Omitted;
 (Fully Connected Layers):
 (0): Linear(in_features=120, out_features=50)
 (1): Linear(in_features=50, out_features=100)
 (2): Tanh()
 (3): Linear(in_features=100, out_features=40)
 (4): Linear(in_features=40, out_features=80)
 (5): Tanh()
 (6): Linear(in_features=80, out_features=30)
 (7): Linear(in_features=30, out_features=60)
 (8): Tanh()
 (9): Linear(in_features=60, out_features=20)
 (10): Linear(in_features=20, out_features=40)
 (11): Tanh()
```

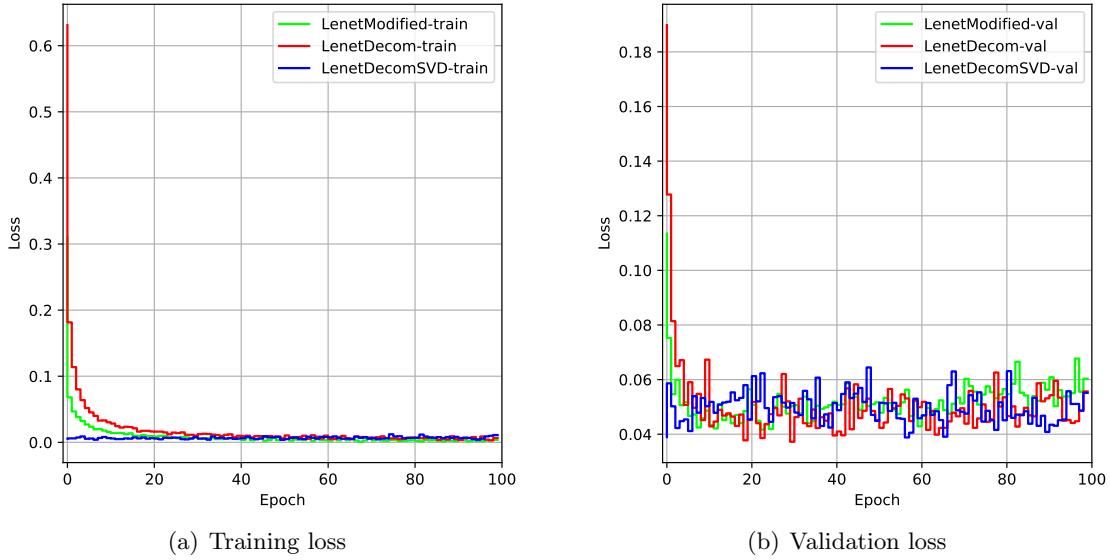
```

(12): Linear(in_features=40, out_features=10)
(13): LogSoftmax()
}

```

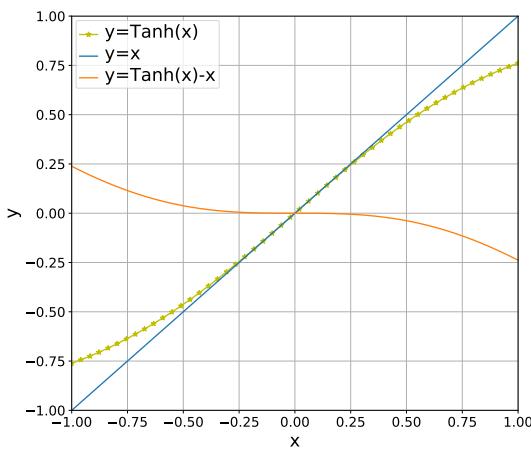
We realize that reducing a fully connected layer with size  $m \times n$  to a low-rank layer  $m \times r, r \times n$  can reduce the space to save the model from  $mn$  values to  $r(m+n)$  values. In this specific example above, we reduce  $120 \times 100 = 12000$  to  $50(120 + 100) = 11000$  values. This reduction also involves the matrix multiplication operations.

After training 100 epochs, we find the minimal training loss in *LenetModified* is smaller than that in *LenetDecom* as shown in Figure 20.1(a). But the validation loss of *LenetModified* (larger than 0.04) is larger than that in *LenetDecom* (smaller than 0.04) as shown in Figure 20.1(b). We see this "regularization" property in low-rank neural networks from this simple example.



**Figure 20.1:** Comparison of full neural networks and low-rank neural networks.

Further, if we have already trained *LenetModified*, and we want to use *LenetDecom* as it saves space and matrix operations. To avoid training from scratch, we could load the weights from *LenetModified* to *LenetDecom*. Then, one method could be used is applying SVD to decompose the weight matrix and keep only the first  $r$  singular values, i.e., set the singular values  $\sigma_{r+1}, \sigma_{r+2}, \dots = 0$ . That is, a weight matrix  $\mathbf{W} = \mathbf{U}_r \Sigma_r \mathbf{V}_r = \mathbf{W}_1 \mathbf{W}_2$ , where we can set  $\mathbf{W}_1 = \mathbf{U}_r \Sigma_r$ ,  $\mathbf{W}_2 = \mathbf{V}_r$ . Then we load the convolutional layers of *LenetDecom* from the convolutional layers of *LenetModified*, and load the fully connected layers of *LenetDecom* via the decomposition of *LenetModified*. The fine-tuning of the network will make the result better. We denote this method as *LenetDecomSVD*. The training and validation losses of *LenetDecomSVD* are shown in Figure 20.1. We find the training loss of



**Figure 20.2:** Demonstration of  $y = \text{Tanh}(x)$  vs  $y = x$ . We notice that in the input field between  $(-0.25, 0.25)$ ,  $\text{Tanh}$  is almost close to a linear function.

*LenetDecomSVD* approaches 0 even in epoch 1 and the validation loss of *LenetDecomSVD* is also smaller than 0.04 which is better than the original *LenetModified*.

This is just one simple example of how the low-rank neural network works. We do not claim that the low-rank neural networks work better in all the scenarios. And also we can apply this matrix decomposition into convolutional layers via the equivalence of convolutional layer and fully connected layers in (Ma and Lu, 2017). Also, a more detailed exploration of going deep in neural networks is explained in (Chen et al., 2015; Wei et al., 2016; Lu et al., 2018)

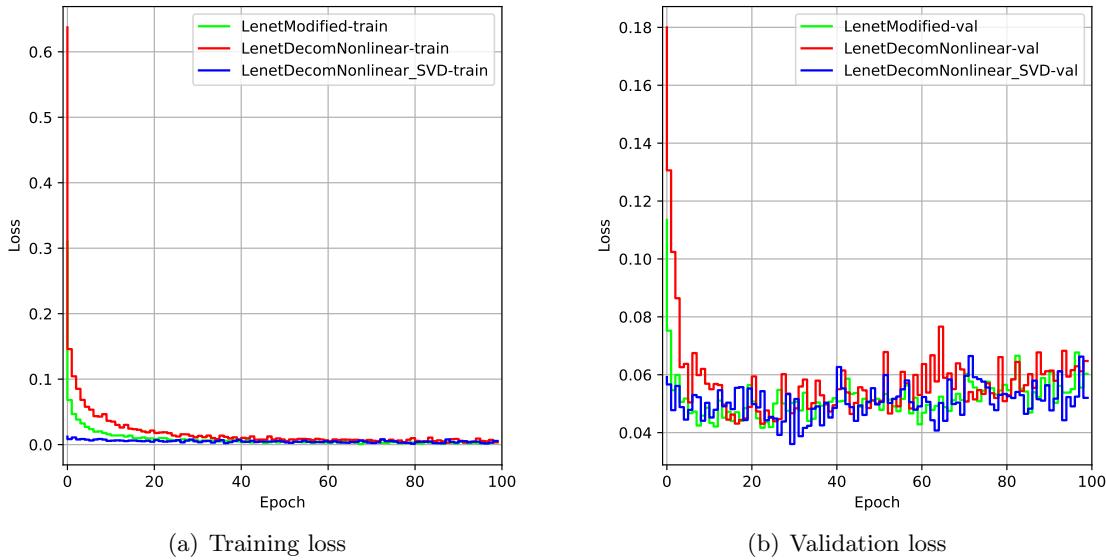
## 20.2. One More Step: Adding a Nonlinear Function Layer

In the above section, we approximate a fully connected layer by a multiplication of two low-rank matrices. The further goal can be putting a nonlinear function in between the two low-rank matrices, i.e., the final layer would be  $f(\mathbf{W}_1)\mathbf{W}_2$ , where  $f()$  is a nonlinear function. We denote this method for the above example as *LenetDecomNonlinear*.

Moreover, if we already trained *LenetModified*, we can also modify the structure of previously trained models to a new structure to avoid training from scratch every time. We then again factor the fully networks by matrix decomposition.

Specifically, when we use *Tanh* function in neural networks as the nonlinear function, where  $\text{Tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ . We notice that in the input field between  $(-0.25, 0.25)$ , *Tanh* is almost close to a linear function as shown in Figure 20.2 where the orange line is the difference between  $y = \text{Tanh}(x)$  and  $y = x$  and the difference is almost 0 when  $x \in (-0.25, 0.25)$ .

After the matrix multiplication of  $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2$ , if we can make all the values  $\mathbf{A}_{in}\mathbf{W}_1$  in the range of  $(-0.25, 0.25)$  where  $\mathbf{A}_1$  is the output from the previous layer, then we can add the *Tanh* without any suffer as it is in the linear space of *Tanh*. Note that matrix decomposition has this equivalence:  $\mathbf{W} = \mathbf{W}_1 * \mathbf{W}_2 = (\sigma\mathbf{W}_1) * (\frac{1}{\sigma}\mathbf{W}_2)$  where  $\sigma$  is any nonzero scalar. Then we can set the value of  $\sigma$  to make the maximal absolute value of  $\mathbf{A}_{in}(\sigma\mathbf{W}_1)$  to be 0.25. Following the specific example in the previous section, we denote this method as



**Figure 20.3:** Comparison of full neural networks and *Tanh* in low-rank neural networks.

*LenetDecomNonlinear\_SVD* where we put an extra *Tanh* function in between the factored matrices as follows:

```
LenetDecomNonlinear {
 Convolutional Layers Omitted;
 (Fully Connected Layers):
 (0): Linear(in_features=120, out_features=50)
 (1): Tanh() [Differece]
 (2): Linear(in_features=50, out_features=100)
 (3): Tanh()
 (4): Linear(in_features=100, out_features=40)
 (5): Tanh() [Differece]
 (6): Linear(in_features=40, out_features=80)
 (7): Tanh()
 (8): Linear(in_features=80, out_features=30)
 (9): Tanh() [Differece]
 (10): Linear(in_features=30, out_features=60)
 (11): Tanh()
 (12): Linear(in_features=60, out_features=20)
 (13): Tanh() [Differece]
 (14): Linear(in_features=20, out_features=40)
 (15): Tanh()
 (16): Linear(in_features=40, out_features=10)
 (17): LogSoftmax()
}
```

After training 100 epochs, we find the minimal training loss in *LenetModified*, *LenetDecomNonlinear*, and *LenetDecomNonlinear\_SVD* are similar as shown in Figure 20.3(a). And the validation loss in *LenetModified* and *LenetDecomNonlinear* are also similar which are larger than 0.04. However, the minimal validation loss in *LenetDecomNonlinear\_SVD* is smaller than 0.04 which gives us a promising result for this matrix decomposition method used in neural networks as shown in Figure 20.3(b). Again, we refer the readers to (Chen et al., 2015; Wei et al., 2016; Lu et al., 2018) for more details about these neural architecture search methods.

## **Part VII**

# **Tensor Decomposition**



## Chapter 21

# Notations and Background

### Contents

---

|      |                                                               |     |
|------|---------------------------------------------------------------|-----|
| 21.1 | Matrices to Tensors . . . . .                                 | 391 |
| 21.2 | Tensor Indexing . . . . .                                     | 391 |
| 21.3 | Inner Product and Frobenius Norm . . . . .                    | 393 |
| 21.4 | Outer Product and Rank-One Tensor . . . . .                   | 393 |
| 21.5 | Diagonal and Identity Tensors . . . . .                       | 393 |
| 21.6 | Matricization: Matrix Representation of a Higher-Order Tensor | 394 |
| 21.7 | Tensor Multiplication . . . . .                               | 396 |
| 21.8 | Special Matrix Products . . . . .                             | 397 |

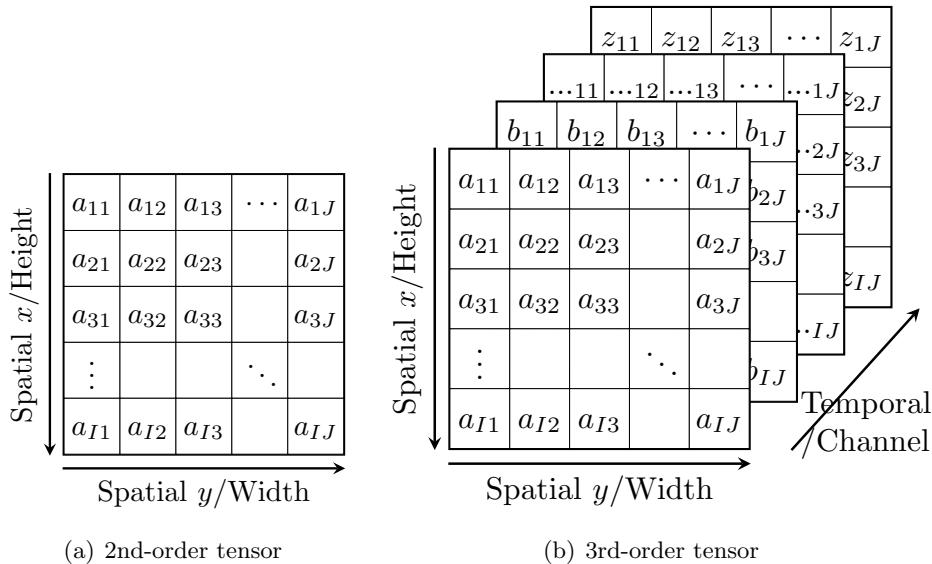
---

## 21.1. Matrices to Tensors

In this section, we briefly overview the background for tensor analysis and we follow the notations largely from the review of (Kolda, 2006; Kolda and Bader, 2009; Cichocki et al., 2016) and (Golub and Van Loan, 2013). A tensor is a multidimensional array where when the dimension is  $N$  we call the tensor an  *$N$ th-order tensor*. For example, a first-order tensor is a vector and a second-order tensor is a matrix. In this sense, tensors are multidimensional extensions of matrices, which are used to represent ubiquitous multidimensional data, such as RGB images, hyperspectral images, and videos.

Previously, we use **boldface** lower case letters possibly with subscripts to denote vectors (e.g.,  $\mu$ ,  $x$ ,  $x_n$ ,  $z$ ) and **boldface** upper case letters possibly with subscripts to denote matrices (e.g.,  $A$ ,  $L_j$ ). To avoid confusion, we will use a **boldface** Euler script letters to denote a tensor with order larger than 2, e.g.,  $\mathcal{X}, \mathcal{Y}, \mathcal{A}, \mathcal{B}, \mathcal{G} \in \mathbb{R}^{I \times J \times K}$  for third-order tensors.

In a second-order tensor (i.e., a matrix), the first axis is usually taken as the **coordinate of spatial  $x$**  or **height**, and the second axis is then taken as the **coordinate of spatial  $y$**  or **width**, e.g., Figure 21.1(a) is an example of a second-order tensor  $X \in \mathbb{R}^{I \times J}$ . A third-order tensor goes further by adding a third dimension which is usually known as a “**temporal**” coordinate (or a “**channel**” coordinate in an RGB picture). The situation of a third-order tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  is shown in Figure 21.1(b) in which case  $K = 3$  in an RGB picture scenario.



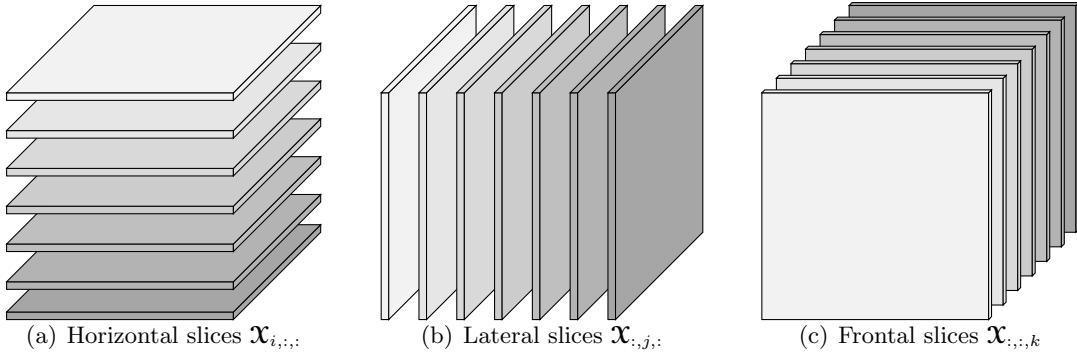
**Figure 21.1:** A 2nd-order tensor into a 3rd-order tensor.

## 21.2. Tensor Indexing

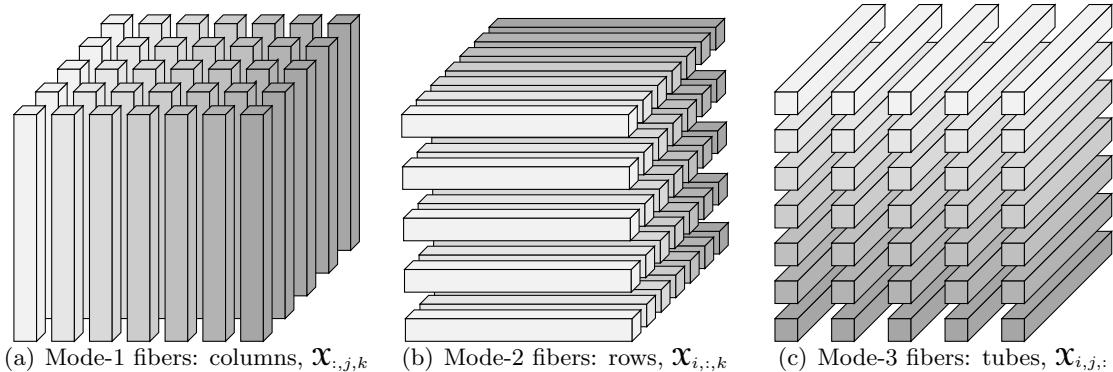
*Slices* are two-dimensional sections of a tensor, defined by fixing all but two indices. Figure 21.2 shows the slices of a third-order tensor  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$  where the *horizontal ones* are

the slices by varying the first index and fixing the last two:  $\mathbf{X}_{i,:,:}$  for all  $i \in \{1, 2, \dots, I\}$ <sup>1</sup>; the *lateral ones* are the slices by varying the second index:  $\mathbf{X}_{:,j,:}$  for all  $j \in \{1, 2, \dots, J\}$ ; and the *frontal ones* are the slices by varying the third index:  $\mathbf{X}_{:,:,k}$  for all  $k \in \{1, 2, \dots, K\}$ .

Similarly, *Fibers* are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. Figure 21.3 shows the fibers of a third-order tensor  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$  where the *column fibers* are the ones by varying only the first index:  $\mathbf{X}_{:,j,k}$ ; the *row fibers* are the ones by varying only the second index:  $\mathbf{X}_{i,:,k}$ ; and the *tube fibers* are the ones by varying only the third index:  $\mathbf{X}_{i,j,:}$ .



**Figure 21.2:** Slices of a third-order tensor  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ . The **darker**, the larger the index.



**Figure 21.3:** Fibers of a third-order tensor  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ .

More generally, the  $(i, j, k)$ -th element of a third-order tensor  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$  is indexed by  $x_{ijk}$ . And the general index for an Nth-order tensor can be implied from the context.

<sup>1</sup> where the indices typically range from 1 to their capital version, here,  $i \in \{1, 2, \dots, I\}$ .

### 21.3. Inner Product and Frobenius Norm

The inner product between two Nth-order tensors with same size  $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is the sum of the product of their entries element-wise:

$$\langle \mathbf{X}, \mathbf{Y} \rangle = \sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \dots \sum_{i_N}^{I_N} (x_{i_1, i_2, \dots, i_N}) \cdot (y_{i_1, i_2, \dots, i_N}).$$

Similarly, the Frobenius norm of an Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is given by  $\sqrt{\langle \mathbf{X}, \mathbf{X} \rangle}$ :

$$\|\mathbf{X}\|_F = \sqrt{\sum_{i_1}^{I_1} \sum_{i_2}^{I_2} \dots \sum_{i_N}^{I_N} (x_{i_1, i_2, \dots, i_N})^2}.$$

### 21.4. Outer Product and Rank-One Tensor

If given two vectors  $\mathbf{a} \in \mathbb{R}^I, \mathbf{b} \in \mathbb{R}^J$ , the outer product is  $\mathbf{a}\mathbf{b}^\top \in \mathbb{R}^{I \times J}$  in a trivial vector language for two vectors. Analogously, in tensor language, we use the symbol “ $\circ$ ” to denote the outer product, e.g.,  $\mathbf{a}\mathbf{b}^\top = \mathbf{a} \circ \mathbf{b}$  for the outer product of two vectors. For higher dimensions, the outer product of  $N$  vectors  $\mathbf{a}^{(1)} \in \mathbb{R}^{I_1}, \mathbf{a}^{(2)} \in \mathbb{R}^{I_2}, \dots, \mathbf{a}^{(N)} \in \mathbb{R}^{I_N}$  is given by

$$\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)},$$

where the  $(i_1, i_2, \dots, i_N)$ -th element can be obtained by

$$(\mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)})_{i_1, i_2, \dots, i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \dots a_{i_N}^{(N)}.$$

An Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is of rank-one if it can be written as the outer product of  $N$  vectors, i.e.,

$$\mathbf{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)},$$

The  $(i_1, i_2, \dots, i_N)$ -th element of  $\mathbf{X}$  is thus given by

$$\mathbf{X}_{i_1, i_2, \dots, i_N} = x_{i_1, i_2, \dots, i_N} = a_{i_1}^{(1)} a_{i_2}^{(2)} \dots a_{i_N}^{(N)}.$$

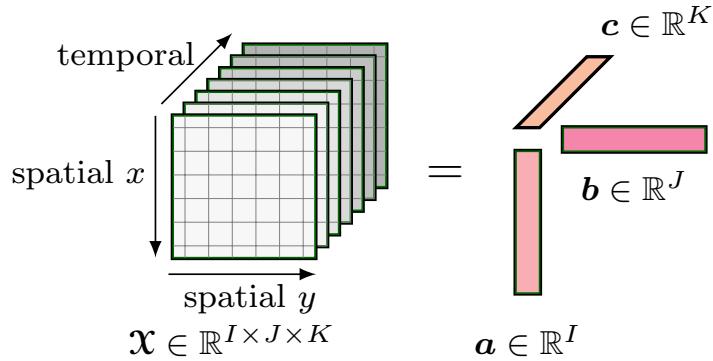
And the situation for a third-order rank-one tensor is shown in Figure 21.4.

### 21.5. Diagonal and Identity Tensors

Diagonal matrices and identity matrices have their counterparts in tensor language:

#### Definition 21.1: Diagonal and Identity Tensors

A tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is called a *diagonal tensor* if  $x_{i_1 i_2 \dots i_N} \neq 0$  if and only if  $i_1 = i_2 = \dots = i_N$ . And when  $x_{i_1 i_2 \dots i_N} = 1$  if and only if  $i_1 = i_2 = \dots = i_N$ , the tensor is known as the *Nth-order identity tensor*.

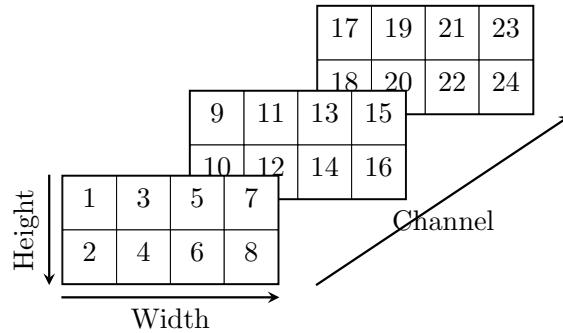


**Figure 21.4:** A third-order tensor with rank-one,  $\mathbf{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$  where the  $(i, j, k)$ -th element of  $\mathbf{X}$  is given by  $x_{ijk} = a_i b_j c_k$ .

## 21.6. Matricization: Matrix Representation of a Higher-Order Tensor

There are several kinds of matrix representations of an  $N$ th-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ . But for the matrix representation along the  $n$ -th mode, the size will always be  $I_n \times (I_1 \dots I_{n-1} I_{n+1} \dots I_N)$ , which is also called the **mode- $n$  matricization of the tensor**  $\mathbf{X}$  and is denoted by  $\mathbf{X}_{(n)}$ .

To see this, we consider a specific example  $\mathbf{Y} \in \mathbb{R}^{2 \times 4 \times 3}$  containing  $\{1, 2, \dots, 24\}$  as entries where each number is stored in ascending order from the first index (height), then the second index (width), and finally the third index (channel) as shown in Figure 21.5:



**Figure 21.5:** An example of a 3rd-order tensor:  $\mathbf{Y} \in \mathbb{R}^{2 \times 4 \times 3}$  where  $y_{ijk} = 1 + (i - 1) + 2(j - 1) + 8(k - 1)$ .

Mathematically, each entry of  $\mathbf{Y}$  can be described by

$$y_{ijk} = 1 + (i - 1) + 2(j - 1) + 8(k - 1).$$

Now, suppose we want the matricized form along the 1st-mode,  $\mathbf{Y}_{(1)} \in \mathbb{R}^{2 \times 12}$ , the number will be **fetched** first from the 1st index of  $\mathbf{Y}$  (height), then the 2nd index (width), and finally the 3rd index (channel). And the number is **stored** into  $\mathbf{Y}_{(1)}$  from the first index,

and then the second:

$$\mathbf{Y}_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 & 17 & 19 & 21 & 23 \\ 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 & 20 & 22 & 24 \end{bmatrix}.$$

If we want the model-2 matricization of tensor  $\mathbf{Y}$ , i.e.,  $\mathbf{Y}_{(2)} \in \mathbb{R}^{4 \times 6}$ , the number will be **fetched** first from the 2nd index (width), then the 1st index (height), and finally the 3rd index (channel). **The storing of the number will always be the same:** the number is **stored** into  $\mathbf{Y}_{(2)}$  from the first index, and then the second:

$$\mathbf{Y}_{(2)} = \begin{bmatrix} 1 & 2 & 9 & 10 & 17 & 18 \\ 3 & 4 & 11 & 12 & 19 & 20 \\ 5 & 6 & 13 & 14 & 21 & 22 \\ 7 & 8 & 15 & 16 & 23 & 24 \end{bmatrix}.$$

If we want the model-3 matricization of tensor  $\mathbf{Y}$ , i.e.,  $\mathbf{Y}_{(3)} \in \mathbb{R}^{3 \times 8}$ , the number will be **fetched** first from the 3rd index (channel), then 1st index (height), and finally the 2nd index (width). **The storing of the number will still be the same:**

$$\mathbf{Y}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{bmatrix}.$$

More generally, given the Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , in the model-n matricization, the tensor element  $(i_1, i_2, \dots, i_N)$  is mapped into matrix element  $(i_n, j)$ :

$$j = 1 + \sum_{k=1, k \neq n}^N (i_k - 1) J_k, \quad \text{where} \quad J_k = \prod_{m=1, m \neq n}^{k-1} I_m.$$

Specially, the *vectorization* of a tensor is defined in a similar way where we fetch the entries in an ordered manner that fetches first from the first index, then the second,  $\dots$ , and finally the last index:

$$vec(\mathbf{X}) = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 24 \end{bmatrix}.$$

However, the matricization and vectorization of a tensor are not unique, as long as we keep the entries consistent, they are the same in the analysis. See also ([De Lathauwer et al., 2000](#); [Kiers, 2000](#); [Kolda and Bader, 2009](#)).

**Norm of the difference of two tensors** The matricization or vectorization can help derive the properties of the tensors. For example, given  $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , it follows that

$$\begin{aligned} \|\mathbf{X} - \mathbf{Y}\|^2 &= \|vec(\mathbf{X}) - vec(\mathbf{Y})\|^2 = \|vec(\mathbf{X})\|^2 - 2vec(\mathbf{X})^\top vec(\mathbf{Y}) + \|vec(\mathbf{Y})\|^2 \\ &= \|\mathbf{X}\|^2 - 2\langle \mathbf{X}, \mathbf{Y} \rangle + \|\mathbf{Y}\|^2. \end{aligned} \tag{21.1}$$

## 21.7. Tensor Multiplication

We consider the *mode-n tensor multiplication*: multiply the tensor in the  $n$ -th mode. Given an  $N$ th-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  and a matrix  $\mathbf{A} \in \mathbb{R}^{M \times I_n}$ , the mode- $n$  tensor multiplication of  $\mathbf{X}$  and  $\mathbf{A}$  will transfer the  $n$ -th dimension of  $\mathbf{X}$  from  $I_n$  to  $M$ . The mode- $n$  tensor multiplication is denoted by  $\mathbf{X} \times_n \mathbf{A} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times \textcolor{blue}{M} \times I_{n+1} \times \dots \times I_N}$ , and is given by

$$(\mathbf{X} \times_n \mathbf{A})_{i_1 \times \dots \times i_{n-1} \times \textcolor{blue}{m} \times i_{n+1} \times \dots \times i_N} = \sum_{i_n=1}^{I_n} (x_{i_1 i_2 \dots i_N})(a_{\textcolor{blue}{m} i_n}). \quad (21.2)$$

**Matrix multiplication as tensor multiplication** In matrix language, suppose two matrices  $\mathbf{A} \in \mathbb{R}^{I \times K}, \mathbf{B} \in \mathbb{R}^{K \times J}$ , then the matrix multiplication can be equivalently denoted by

$$\mathbf{AB} = \mathbf{B} \times_1 \mathbf{A}.$$

Suppose again,  $\mathbf{A} \in \mathbb{R}^{I \times K}, \mathbf{B} \in \mathbb{R}^{J \times K}$ , then the matrix multiplication can be equivalently denoted by

$$\mathbf{AB}^\top = \mathbf{A} \times_2 \mathbf{B}.$$

Going further by supposing  $\mathbf{A} \in \mathbb{R}^{M \times N}$  whose reduced SVD (Theorem 14.1, p. 265) of  $\mathbf{A}$  is given by

$$\underset{M \times N}{\mathbf{A}} = \underset{M \times R}{\mathbf{U}} \underset{R \times R}{\Sigma} \underset{R \times N}{\mathbf{V}^\top}.$$

Then, the reduced SVD of  $\mathbf{A}$  can be equivalently denoted as

$$\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^\top = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V}. \quad (21.3)$$

The full SVD of  $\mathbf{A}$  can be represented in a similar way.

The matricization of a tensor can show what happens in the tensor multiplication.

### Lemma 21.1: (Tensor Multiplication in Matricization)

Given the  $N$ th-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  and the matrix  $\mathbf{A} \in \mathbb{R}^{M \times I_n}$ , it follows that

$$\mathbf{Y} = \mathbf{X} \times_n \mathbf{A} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times \textcolor{blue}{M} \times I_{n+1} \times \dots \times I_N}$$

$$\xrightarrow{\text{leads to}} \mathbf{Y}_{(n)} = \mathbf{AX}_{(n)} \in \mathbb{R}^{M \times I_{-n}},$$

where  $I_{-n} = I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N$ .

From the above lemma, conversely, suppose columns of  $\mathbf{A}$  are mutually orthonormal with  $I_n \leq M$  (semi-orthogonal, see definition in Section 3.5, p. 87). Then it follows that

$$\mathbf{A}^\top \mathbf{Y}_{(n)} = \underbrace{\mathbf{A}^\top \mathbf{A}}_I \mathbf{X}_{(n)} = \mathbf{X}_{(n)}.$$

This reveals an important property of tensor multiplication:  $\mathbf{X} = \mathbf{Y} \times_n \mathbf{A}^\top$ . That is, if  $\mathbf{A}$  is semi-orthogonal, we have

$$\mathbf{Y} = \mathbf{X} \times_n \mathbf{A} \quad \xrightarrow{\text{leads to}} \quad \mathbf{X} = \mathbf{Y} \times_n \mathbf{A}^\top. \quad (21.4)$$

**Lemma 21.2: (Tensor Multiplication)**

Suppose given the Nth-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , we have

1. *Distributive law.* Given further the matrices  $\mathbf{A} \in \mathbb{R}^{J_m \times I_m}$  and  $\mathbf{B} \in \mathbb{R}^{J_n \times I_n}$  where  $m \neq n$ , it follows that

$$\mathcal{X} \times_m \mathbf{A} \times_n \mathbf{B} = (\mathcal{X} \times_m \mathbf{A}) \times_n \mathbf{B} = (\mathcal{X} \times_n \mathbf{B}) \times_m \mathbf{A}.$$

2. Given the matrices  $\mathbf{A} \in \mathbb{R}^{P \times I_m}$  and  $\mathbf{B} \in \mathbb{R}^{Q \times P}$ , it follows that

$$\mathcal{X} \times_n \mathbf{A} \times_n \mathbf{B} = \mathcal{X} \times_n (\mathbf{B}\mathbf{A}) \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times \textcolor{blue}{Q} \times I_{n+1} \times \dots \times I_N}.$$

3. Given  $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times \textcolor{blue}{M} \times I_{n+1} \times \dots \times I_N}$ ,  $\mathcal{Y} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times \textcolor{blue}{K} \times I_{n+1} \times \dots \times I_N}$ , and  $\mathbf{A} \in \mathbb{R}^{M \times K}$ , it follows that

$$\langle \mathcal{X}, \mathcal{Y} \times_n \mathbf{A} \rangle = \langle \mathcal{X} \times_n \mathbf{A}^\top, \mathcal{Y} \rangle.$$

4. Given semi-orthogonal matrix  $\mathbf{A} \in \mathbb{R}^{P \times I_n}$ , it follows that

$$\mathcal{Y} = \mathcal{X} \times_n \mathbf{A} \quad \xrightarrow{\text{leads to}} \quad \mathcal{X} = \mathcal{Y} \times_n \mathbf{A}^\top.$$

and

$$\|\mathcal{Y}\| = \|\mathcal{X}\|,$$

i.e., the length preservation under semi-orthogonal.

## 21.8. Special Matrix Products

Several matrix products will be proved important in the illustration of the algorithms in the sequel.

The *Kronecker product* of vectors  $\mathbf{a} \in \mathbb{R}^I$  and  $\mathbf{b} \in \mathbb{R}^K$  is denoted by  $\mathbf{a} \otimes \mathbf{b}$ :

$$\mathbf{a} \otimes \mathbf{b} = \begin{bmatrix} a_1 \mathbf{b} \\ a_2 \mathbf{b} \\ \vdots \\ a_I \mathbf{b} \end{bmatrix},$$

which is a column vector of size  $(IK)$ .

**Definition 21.1: Matrix Kronecker**

Similarly, the *Kronecker product* of matrices  $\mathbf{A} \in \mathbb{R}^{I \times J}$  and  $\mathbf{B} \in \mathbb{R}^{K \times L}$  is denoted by  $\mathbf{A} \otimes \mathbf{B}$ :

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \dots & a_{IJ}\mathbf{B} \end{bmatrix} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \ \dots \ \mathbf{a}_1 \otimes \mathbf{a}_L \mid \mathbf{a}_2 \otimes \mathbf{b}_1 \ \dots \ \mathbf{a}_2 \otimes \mathbf{a}_L \mid \dots \mid \mathbf{a}_J \otimes \mathbf{b}_1 \ \dots \ \mathbf{a}_J \otimes \mathbf{a}_L],$$

which is a matrix of size  $(IK) \times (JL)$ .

Specifically, we notice that, when four vectors  $\{\mathbf{a} \in \mathbb{R}^I$  and  $\mathbf{b} \in \mathbb{R}^K\}$  and  $\{\mathbf{c} \in \mathbb{R}^I$  and  $\mathbf{d} \in \mathbb{R}^K\}$ , then

$$(\mathbf{a} \otimes \mathbf{b})^\top (\mathbf{c} \otimes \mathbf{d}) = [a_1\mathbf{b}^\top \ a_2\mathbf{b}^\top \ \dots \ a_I\mathbf{b}^\top] \begin{bmatrix} c_1\mathbf{d} \\ c_2\mathbf{d} \\ \vdots \\ c_I\mathbf{d} \end{bmatrix} = \sum_{i=1}^I a_i c_i \mathbf{b}^\top \mathbf{d} = (\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{d}). \quad (21.5)$$

Specially, when  $\mathbf{c} = \mathbf{a}, \mathbf{d} = \mathbf{b}$ , it follows that

$$(\mathbf{a} \otimes \mathbf{b})^\top (\mathbf{a} \otimes \mathbf{b}) = \|\mathbf{a}\|^2 \cdot \|\mathbf{b}\|^2.$$

Similarly, for  $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{I \times J}$  and  $\mathbf{B}, \mathbf{D} \in \mathbb{R}^{K \times L}$ , it follows that

$$(\mathbf{A} \otimes \mathbf{B})^\top (\mathbf{C} \otimes \mathbf{D}) = (\mathbf{A}^\top \mathbf{C}) \otimes (\mathbf{B}^\top \mathbf{D}). \quad (21.6)$$

Note also, for  $\mathbf{A} \in \mathbb{R}^{I \times J}$ ,  $\mathbf{B} \in \mathbb{R}^{K \times L}$ ,  $\mathbf{C} \in \mathbb{R}^{J \times I}$ , and  $\mathbf{D} \in \mathbb{R}^{L \times K}$ , the above equation reduces to

$$(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD}). \quad (21.7)$$

### Pseudo Inverse in Kronecker Product

Moreover, following from (Van Loan, 2000), the pseudo inverse of  $(\mathbf{A} \otimes \mathbf{B})$  is given by

$$(\mathbf{A} \otimes \mathbf{B})^+ = \mathbf{A}^+ \otimes \mathbf{B}^+, \quad (21.8)$$

where  $\mathbf{A}^+$  is the pseudo inverse of matrix  $\mathbf{A}$ . Recall that the pseudo inverse of a full column rank matrix  $\mathbf{A}$  is simply  $\mathbf{A}^+ = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$ . When  $\mathbf{A}, \mathbf{B}$  are both semi-orthogonal (see definition in Section 3.5, p. 87), the pseudo inverse is  $\mathbf{A}^+ = \mathbf{A}^\top, \mathbf{B}^+ = \mathbf{B}^\top$ , and it follows that

$$(\mathbf{A} \otimes \mathbf{B})^+ = \mathbf{A}^\top \otimes \mathbf{B}^\top. \quad (21.9)$$

Analogously, the above pseudo inverse can be applied to the Kronecker product of a sequence of matrices.

### Orthogonality in Kronecker Product

Suppose further  $\mathbf{a} \in \mathbb{R}^I$ , and  $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^K$  with  $\mathbf{b}_1^\top \mathbf{b}_2 = 0$ , then

$$(\mathbf{a} \otimes \mathbf{b}_1) \perp (\mathbf{a} \otimes \mathbf{b}_2). \quad (21.10)$$

Or suppose  $\mathbf{a}_1, \mathbf{a}_2 \in \mathbb{R}^I$  with  $\mathbf{a}_1^\top \mathbf{a}_2 = 0$ , and  $\mathbf{b} \in \mathbb{R}^K$ , then

$$(\mathbf{a}_1 \otimes \mathbf{b}) \perp (\mathbf{a}_2 \otimes \mathbf{b}). \quad (21.11)$$

The above two findings imply  $\mathbf{A} \otimes \mathbf{B}$  contains mutually orthogonal (orthonormal) columns if both  $\mathbf{A} \in \mathbb{R}^{I \times J}$  and  $\mathbf{B} \in \mathbb{R}^{K \times L}$  contain mutually orthogonal (orthonormal) columns.

### Definition 21.2: Khatri-Rao Product

The *Khatri-Rao product* of matrices  $\mathbf{A} \in \mathbb{R}^{I \times K}$  and  $\mathbf{B} \in \mathbb{R}^{J \times K}$  is denoted by  $\mathbf{A} \odot \mathbf{B}$ :

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \dots \quad \mathbf{a}_K \otimes \mathbf{b}_K],$$

which is a matrix of size  $(IJ) \times K$ . And it is known as the “matching columnwise” Kronecker product.

From the above definition on the Khatri-Rao product, for two vectors  $\mathbf{a}$  and  $\mathbf{b}$ , it follows that

$$\mathbf{a} \odot \mathbf{b} = \mathbf{a} \otimes \mathbf{b}.$$

Note that the “distributive law” follows that

$$\mathbf{A} \odot \mathbf{B} \odot \mathbf{C} = (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C} = \mathbf{A} \odot (\mathbf{B} \odot \mathbf{C}).$$

### Definition 21.3: Hadamard Product

The *Hadamard product* of matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{I \times J}$  is denoted by  $\mathbf{A} \circledast \mathbf{B}$ :

$$\mathbf{A} \circledast \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \dots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \dots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \dots & a_{IJ}b_{IJ} \end{bmatrix},$$

which is a matrix of size  $I \times J$ .

Specifically, we further notice that, for two matrices  $\mathbf{A} \in \mathbb{R}^{I \times K}$  and  $\mathbf{B} \in \mathbb{R}^{J \times K}$ , it follows that

$$\mathbf{Z} = (\mathbf{A} \odot \mathbf{B})^\top (\mathbf{A} \odot \mathbf{B}) = \begin{bmatrix} (\mathbf{a}_1 \otimes \mathbf{b}_1)^\top \\ (\mathbf{a}_2 \otimes \mathbf{b}_2)^\top \\ \vdots \\ (\mathbf{a}_K \otimes \mathbf{b}_K)^\top \end{bmatrix} [\mathbf{a}_1 \otimes \mathbf{b}_1 \quad \mathbf{a}_2 \otimes \mathbf{b}_2 \quad \dots \quad \mathbf{a}_K \otimes \mathbf{b}_K], \quad (21.12)$$

where  $\mathbf{Z} \in \mathbb{R}^{K \times K}$  and each entry  $(i, j)$  element  $z_{ij}$  is given by

$$z_{ij} = (\mathbf{a}_i \otimes \mathbf{b}_i)^\top (\mathbf{a}_j \otimes \mathbf{b}_j) = (\mathbf{a}_i^\top \mathbf{a}_j)(\mathbf{b}_i^\top \mathbf{b}_j).$$

where the last equality comes from Equation (21.5). Therefore,  $\mathbf{Z}$  can be equivalently written as

$$\mathbf{Z} = (\mathbf{A} \odot \mathbf{B})^\top (\mathbf{A} \odot \mathbf{B}) = (\mathbf{A}^\top \mathbf{A}) \circledast (\mathbf{B}^\top \mathbf{B}). \quad (21.13)$$

To conclude, it follows that

|                                                                        |                                                                                                                                                                                                                                 |
|------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $(\mathbf{a} \otimes \mathbf{b})^\top (\mathbf{c} \otimes \mathbf{d})$ | $= (\mathbf{a}^\top \mathbf{c})(\mathbf{b}^\top \mathbf{d});$                                                                                                                                                                   |
| $(\mathbf{a} \otimes \mathbf{b})^\top (\mathbf{a} \otimes \mathbf{b})$ | $= \ \mathbf{a}\ ^2 \cdot \ \mathbf{b}\ ^2;$                                                                                                                                                                                    |
| $(\mathbf{A} \otimes \mathbf{B})^\top (\mathbf{C} \otimes \mathbf{D})$ | $= (\mathbf{A}^\top \mathbf{C}) \otimes (\mathbf{B}^\top \mathbf{D}), \quad \left( \begin{array}{l} \text{where } \mathbf{A}, \mathbf{C} \text{ same shape,} \\ \mathbf{B}, \mathbf{D} \text{ same shape} \end{array} \right);$ |
| $(\mathbf{A} \otimes \mathbf{B})^+$                                    | $= \mathbf{A}^+ \otimes \mathbf{B}^+;$                                                                                                                                                                                          |
| $\mathbf{A} \odot \mathbf{B} \odot \mathbf{C}$                         | $= (\mathbf{A} \odot \mathbf{B}) \odot \mathbf{C}$                                                                                                                                                                              |
|                                                                        | $= \mathbf{A} \odot (\mathbf{B} \odot \mathbf{C});$                                                                                                                                                                             |
| $(\mathbf{A} \odot \mathbf{B})^\top (\mathbf{A} \odot \mathbf{B})$     | $= (\mathbf{A}^\top \mathbf{A}) \circledast (\mathbf{B}^\top \mathbf{B}).$                                                                                                                                                      |

(21.14)

# Chapter 22

# CP Decomposition

## Contents

---

|                                               |     |
|-----------------------------------------------|-----|
| 22.1 CP Decomposition . . . . .               | 402 |
| 22.2 Computing the CP Decomposition . . . . . | 405 |

---

## 22.1. CP Decomposition

### Theorem 22.1: (CP Decomposition)

The CP decomposition factorizes a tensor into a sum of component rank-one tensors. For a general Nth-order tensor,  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , it admits the CP decomposition

$$\mathcal{X} \approx [\![\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}]\!] = \sum_{r=1}^R \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)},$$

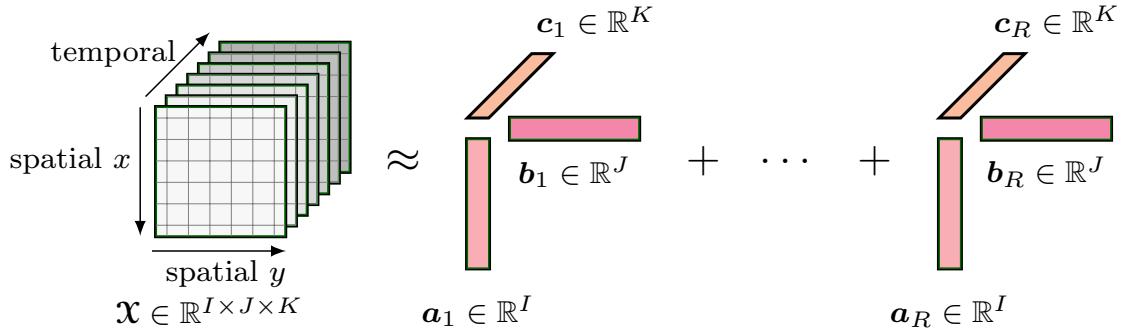
where  $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \dots, \mathbf{a}_R^{(n)}] \in \mathbb{R}^{I_n \times R}$  for all  $n \in \{1, 2, \dots, N\}$  is the column partition of the matrix  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$ .

Or it is often useful to assume that the columns of  $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}$  are normalized to length one with the weights absorbed into the vector  $\boldsymbol{\lambda} \in \mathbb{R}^R$  so that

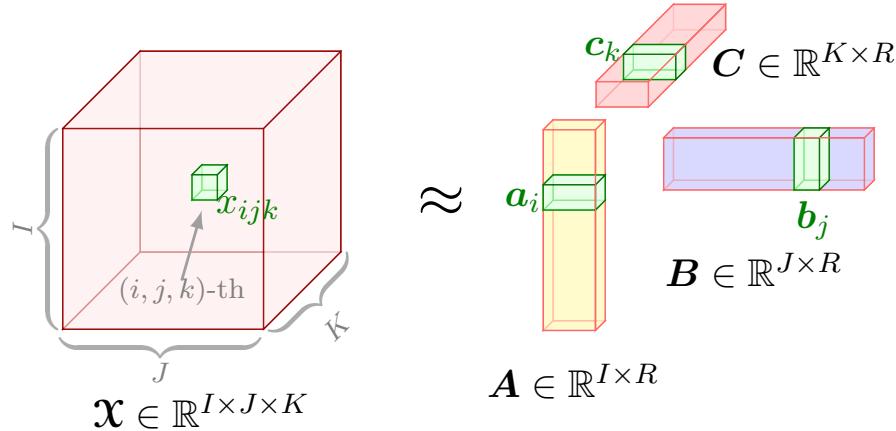
$$\mathcal{X} \approx [\![\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}]\!] = \sum_{r=1}^R \boldsymbol{\lambda}_r \cdot \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)},$$

where we follow the notation  $[\![\dots]\!]$  from (Kruskal, 1977; Kolda and Bader, 2009).

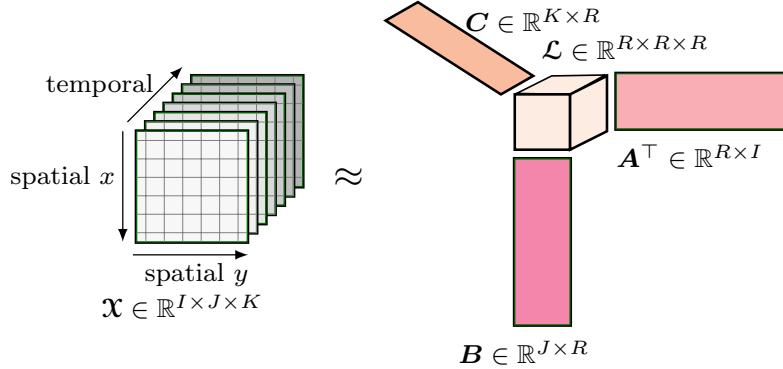
One can find the idea behind the CP decomposition is to express the tensor as a sum of rank-one tensors, i.e., a sum of the outer product of vectors (Section 21.4, p. 393). The CP decomposition is also known as the *Canonical Polyadic Decomposition*, *CANDECOMP-PARAFAC*, or simply *PARAFAC decomposition*.



**Figure 22.1:** The CP decomposition of a third-order tensor:  $\mathcal{X} \approx [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$  where  $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$ ,  $\mathbf{A} \in \mathbb{R}^{I \times R}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times R}$  and  $\mathbf{C} \in \mathbb{R}^{K \times R}$ . Compare to the third-order rank-one tensor in Figure 21.4.



**Figure 22.2:** Index of the CP decomposition of a third-order tensor:  $\mathbf{X} \approx [\mathbf{A}, \mathbf{B}, \mathbf{C}]$ .



**Figure 22.3:** The CP decomposition of a third-order tensor:  $\mathbf{X} \approx [\mathcal{L}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$  where  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ ,  $\mathcal{L} \in \mathbb{R}^{R \times R \times R}$ , and where columns of  $\mathbf{A} \in \mathbb{R}^{I \times R}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times R}$  and  $\mathbf{C} \in \mathbb{R}^{K \times R}$  are of unit length. Compare to the first form of the CP decomposition in Figure 22.1.

**Rank of a tensor** From the CP decomposition, the rank of a tensor  $\mathbf{X}$  can be defined as the smallest  $R$  for which the CP decomposition holds exactly.

**Matricization** When  $\mathbf{X}$  is a 2nd-order tensor (i.e., a matrix), the CP decomposition is just the rank decomposition (Theorem 5.3, p. 165). For simplicity, we consider the CP decomposition for the third-order tensor  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$  where the situation is illustrated in Figure 22.1:

$$\mathbf{X} \approx [\mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \quad (22.1)$$

where  $\mathbf{A} \in \mathbb{R}^{I \times R}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times R}$  and  $\mathbf{C} \in \mathbb{R}^{K \times R}$ . Element-wise, the above Equation (22.1) can be written as

$$x_{ijk} = \sum_{r=1}^R a_{ir} b_{jr} c_{kr},$$

as shown in Figure 22.2. Therefore, the matricized form of the third-order CP decomposition can be written as

$$\begin{cases} \mathbf{X}_{(1)} \approx \mathbf{A}(\mathbf{C} \odot \mathbf{B})^\top \in \mathbb{R}^{I \times (JK)}; \\ \mathbf{X}_{(2)} \approx \mathbf{B}(\mathbf{C} \odot \mathbf{A})^\top \in \mathbb{R}^{J \times (IK)}; \\ \mathbf{X}_{(3)} \approx \mathbf{C}(\mathbf{B} \odot \mathbf{A})^\top \in \mathbb{R}^{K \times (IJ)}, \end{cases} \quad (22.2)$$

where  $\mathbf{X}_{(n)} \in \mathbb{R}^{I \times (JK)}$  is the mode- $n$  matricization of the tensor  $\mathbf{X}$  for  $n \in \{1, 2, 3\}$ , “ $\odot$ ” is the Khatri-Rao product of matrices (Definition 21.2, p. 399) such that

$$\begin{cases} \mathbf{C} \odot \mathbf{B} = [\mathbf{c}_1 \otimes \mathbf{b}_1, \mathbf{c}_2 \otimes \mathbf{b}_2, \dots, \mathbf{c}_R \otimes \mathbf{b}_R] \in \mathbb{R}^{JK \times R}; \\ \mathbf{C} \odot \mathbf{A} = [\mathbf{c}_1 \otimes \mathbf{a}_1, \mathbf{c}_2 \otimes \mathbf{a}_2, \dots, \mathbf{c}_R \otimes \mathbf{a}_R] \in \mathbb{R}^{IK \times R}; \\ \mathbf{B} \odot \mathbf{A} = [\mathbf{b}_1 \otimes \mathbf{a}_1, \mathbf{b}_2 \otimes \mathbf{a}_2, \dots, \mathbf{b}_R \otimes \mathbf{a}_R] \in \mathbb{R}^{IJ \times R}. \end{cases}$$

In full generality, come back to the Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , the mode- $n$  matricized form of  $\mathbf{X} \approx [\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}]$  is given by

$$\underbrace{\mathbf{X}_{(n)}}_{I_n \times (I_{-n})} \approx \underbrace{\mathbf{A}^{(n)} \left( \underbrace{\mathbf{A}^{(N)} \odot \mathbf{A}^{(N-1)} \odot \dots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \dots \odot \mathbf{A}^{(2)} \odot \mathbf{A}^{(1)}}_{R \times (I_{-n})} \right)^\top}_{I_n \times R}$$

where  $I_{-n} = I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N$ .

**Vectorization** Similarly, the vectorization of  $\mathbf{X} \approx [\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}]$  is given by

$$\underbrace{\text{vec}(\mathbf{X})}_{(I_1 I_2 \dots I_N) \times 1} \approx \underbrace{\left( \mathbf{A}^{(N)} \odot \mathbf{A}^{(N-1)} \odot \dots \odot \mathbf{A}^{(2)} \odot \mathbf{A}^{(1)} \right)}_{(I_1 I_2 \dots I_N) \times R} \cdot \underbrace{\mathbf{1}}_{R \times 1},$$

where  $\mathbf{1} = [1, 1, \dots, 1]^\top \in \mathbb{R}^{R \times 1}$ .

**Equivalent forms on the CP decomposition** We mentioned in the Theorem 22.1 when the columns of  $\mathbf{A}^{(n)}$ 's are of unit length with the weights absorbed into the vector  $\boldsymbol{\lambda} \in \mathbb{R}^R$ , then the CP decomposition of the Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  can be written as

$$\mathbf{X} \approx [\boldsymbol{\lambda}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r=1}^R \boldsymbol{\lambda}_r \cdot \mathbf{a}_r^{(1)} \circ \mathbf{a}_r^{(2)} \circ \dots \circ \mathbf{a}_r^{(N)},$$

Suppose further the Nth-order tensor  $\mathcal{L} \in \mathbb{R}^{R \times R \times \dots \times R}$  is a diagonal tensor (Definition 21.1, p. 393) with

$$\begin{cases} l_{i,i,\dots,i} = \lambda_i, & \text{if } i \in \{1, 2, \dots, N\}; \\ l_{i_1,i_2,\dots,i_N} = 0, & \text{if not } \{i_1 = i_2 = \dots = i_N\}, \end{cases} \quad (22.3)$$

then the CP decomposition of the Nth-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  can be written as

$$\mathcal{X} \approx [\mathcal{L}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r_1=1}^R \sum_{r_2=1}^R \dots \sum_{r_N=1}^R l_{r_1 r_2 \dots r_N} \mathbf{a}_{r_1}^{(1)} \circ \mathbf{a}_{r_2}^{(2)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)}. \quad (22.4)$$

By the mode- $n$  tensor multiplication in Equation (21.2), the CP decomposition in Equation (22.4) can also be written as:

$$\mathcal{X} \approx [\mathcal{L}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \mathcal{L} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)}. \quad (22.5)$$

The CP decomposition in the form of Equation (22.5) for a third-order tensor is then shown in Figure 22.3.

## 22.2. Computing the CP Decomposition

---

### Algorithm 63 CP Decomposition via ALS

---

**Require:** Nth-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ;

- 1: Pick a rank  $R$ ;
  - 2: Initialize  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$  for all  $n \in \{1, 2, \dots, N\}$  randomly;
  - 3: Choose maximal number of iterations  $C$ ;
  - 4:  $iter = 0$ ;
  - 5: **while**  $iter < C$  **do**
  - 6:      $iter = iter + 1$ ;
  - 7:     **for**  $n = 1, 2, \dots, N$  **do**
  - 8:          $\mathbf{V} = \mathbf{A}^{(N)\top} \mathbf{A}^{(N)} \circledast \dots \circledast \mathbf{A}^{(n+1)\top} \mathbf{A}^{(n+1)} \circledast \mathbf{A}^{(n-1)\top} \mathbf{A}^{(n-1)} \circledast \dots \circledast \mathbf{A}^{(1)\top} \mathbf{A}^{(1)} \in \mathbb{R}^{R \times R}$ ;
  - 9:          $\mathbf{W} = (\mathbf{A}^{(N)} \odot \mathbf{A}^{(N-1)} \odot \dots \odot \mathbf{A}^{(n+1)} \odot \mathbf{A}^{(n-1)} \odot \dots \odot \mathbf{A}^{(2)} \odot \mathbf{A}^{(1)}) \in \mathbb{R}^{R \times I_{-n}}$ ;
  - 10:          $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_{-n}}$ ; ▷ Calculate the mode- $n$  matricization of tensor  $\mathcal{X}$
  - 11:          $\mathbf{A}^{(n)} = \mathbf{X}_{(n)} \mathbf{W} \mathbf{V}^+ \in \mathbb{R}^{I_n \times R}$ ;
  - 12:     **end for**
  - 13: **end while**
  - 14: Output  $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}$ ;
- 

There is no finite algorithm for determining the rank of a tensor, and the complexity for it is an NP-hard problem (Håstad, 1989). Therefore, the rank  $R$  in the CP decomposition can not be set up front. However, the rank  $R$  can be viewed as a hyperparameter and the CP decomposition thus is approximated by a low-rank tensor decomposition.

In the third-order case,  $\mathcal{X} \approx [\mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ , suppose we fix the tensor rank  $R$ , we want to find a rank- $R$  tensor  $\widehat{\mathcal{X}}$  such that

$$\min_{\widehat{\mathcal{X}}} \|\mathcal{X} - \widehat{\mathcal{X}}\|, \quad \text{with } \widehat{\mathcal{X}} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

Having fixed all but one matrix, the problem reduces to a least squares problem from the matricized form. Therefore, the ALS method can be employed. The ALS approach fixes

$\mathbf{B}$  and  $\mathbf{C}$  to update for  $\mathbf{A}$ ; then fixes  $\mathbf{A}$  and  $\mathbf{C}$  to update for  $\mathbf{B}$ ; then fixes  $\mathbf{A}$  and  $\mathbf{B}$  to update for  $\mathbf{C}$ , and continues to repeat the entire procedure until some convergence criterion is satisfied or it has enough iterations.

**Given  $\mathbf{B}$  and  $\mathbf{C}$ , update for  $\mathbf{A}$**  For example, suppose that  $\mathbf{B}$  and  $\mathbf{C}$  are fixed. Then we want to solve

$$\min_{\widehat{\mathbf{A}}} \|\mathbf{X}_{(1)} - \widehat{\mathbf{A}}(\mathbf{C} \odot \mathbf{B})^\top\|.$$

For simplicity, we only consider the ALS without regularization and bias terms. Recall the update of the ALS in the netflix recommender, the update of  $\widehat{\mathbf{A}}$  is just the same update as that of  $\mathbf{W}$  in Equation (17.7): for fixed matrices  $\mathbf{Z}, \mathbf{D}$ , we want to find  $\mathbf{W}$  that  $\min \|\mathbf{D} - \mathbf{WZ}\|^2$ , if  $\mathbf{ZZ}^\top$  is invertible, the update is given by

$$\mathbf{W} = \mathbf{DZ}^\top (\mathbf{ZZ}^\top)^{-1} \leftarrow \arg \min_{\mathbf{W}} \|\mathbf{D} - \mathbf{WZ}\|^2.$$

Therefore, come back to the update in the third-order CP decomposition, it follows that

$$\begin{aligned} \widehat{\mathbf{A}} &\leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) \left( (\mathbf{C} \odot \mathbf{B})^\top (\mathbf{C} \odot \mathbf{B}) \right)^{-1} \\ &= \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) \left( (\mathbf{C}^\top \mathbf{C}) \circledast (\mathbf{B}^\top \mathbf{B}) \right)^{-1}, \end{aligned}$$

where the last equality comes from Equation (21.13). When  $((\mathbf{C}^\top \mathbf{C}) \circledast (\mathbf{B}^\top \mathbf{B}))$  is not invertible, a pseudo inverse can be applied instead (Appendix E, p. 445):

$$\widehat{\mathbf{A}} \leftarrow \mathbf{X}_{(1)}(\mathbf{C} \odot \mathbf{B}) \left( (\mathbf{C}^\top \mathbf{C}) \circledast (\mathbf{B}^\top \mathbf{B}) \right)^+,$$

where  $\mathbf{A}^+$  is the pseudo-inverse of the matrix  $\mathbf{A}$ . The full general procedure for computing the CP decomposition of an Nth-order tensor is then formulated in Algorithm 63. An analogous update for the *nonnegative CP decomposition* follows immediately from the “similarity” between the ALS and NMF. One can simply change the ALS update by the multiplicative update (see Equation (18.1), p. 366).

## Chapter 23

# Tucker Decomposition

### Contents

---

|             |                                           |            |
|-------------|-------------------------------------------|------------|
| <b>23.1</b> | <b>Tucker Decomposition</b>               | <b>408</b> |
| <b>23.2</b> | <b>Computing the Tucker Decomposition</b> | <b>410</b> |

---

### 23.1. Tucker Decomposition

#### Theorem 23.1: (Tucker Decomposition)

The Tucker decomposition factorizes a tensor into a sum of component rank-one tensors. For a general Nth-order tensor,  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , it admits the Tucker decomposition

$$\mathbf{X} \approx [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} \mathbf{a}_{r_1}^{(1)} \circ \mathbf{a}_{r_2}^{(2)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)},$$

where

- $R_1 < I_1, R_2 < I_2, \dots, R_N < I_N$ ;
- $\mathbf{G}$  of size  $R_1 \times R_2 \times \dots \times R_N$  is called the *core tensor* so that  $\mathbf{G}$  can be thought of as the compressed version of  $\mathbf{X}$ ;
- $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \dots, \mathbf{a}_{R_n}^{(n)}] \in \mathbb{R}^{I_n \times R_n}$  for all  $n \in \{1, 2, \dots, N\}$  is the column partition of the matrix  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ ;
- The  $\mathbf{A}^{(n)}$ 's usually have mutually orthonormal columns and can be thought of as the principal component of each mode. In this sense, the  $\mathbf{A}^{(n)}$ 's are *semi-orthogonal matrices* (see the definition in Section 3.5, p. 87);
- We can complete the semi-orthogonal matrices into *full orthogonal matrices* by adding *silent columns* into  $\mathbf{A}^{(n)}$ 's so that  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times I_n}$  is an orthogonal matrix, in which case,  $\mathbf{G}$  will be expanded to a tensor of size  $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  where  $g_{r_1 r_2 \dots r_N} = 0$  when either one of  $r_n > R_n$  for  $n \in \{1, 2, \dots, N\}$ . This is known as the *full Tucker decomposition*, and the previous one is also called the *reduced* one to avoid confusion; And we shall only consider the reduced case in most of our discussions.

**Equivalent forms on the Tucker decomposition** By the mode- $n$  tensor multiplication in Equation (21.2), the Tucker decomposition can also be written as:

$$\mathbf{X} \approx [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)}. \quad (23.1)$$

By the result in Lemma 21.2 (4), since  $\mathbf{A}^{(n)}$ 's are semi-orthogonal, it also follows that

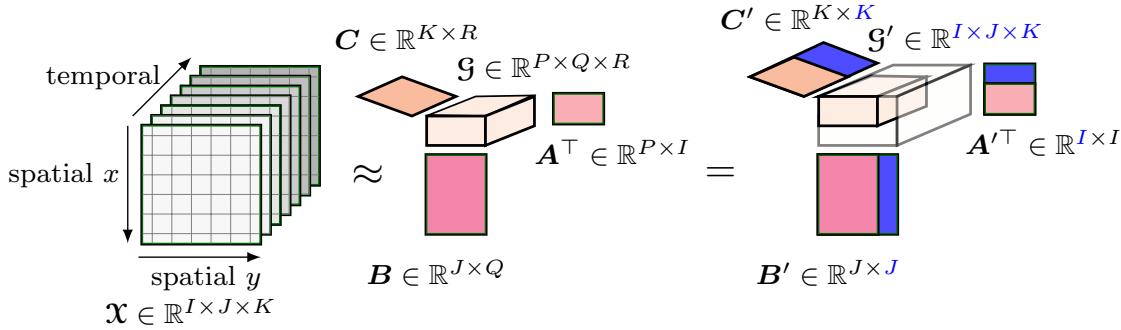
$$\mathbf{G} = [\mathbf{X}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(N)\top}] = \mathbf{X} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \dots \times_N \mathbf{A}^{(N)\top}. \quad (23.2)$$

The operator defined in Equation (23.1) is sometimes referred to as the *Tucker operator* (Kolda, 2006). Element-wise, the  $(i_1, i_2, \dots, i_N)$ -th element of  $\mathbf{X}$  can be obtained by

$$x_{i_1, i_2, \dots, i_N} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} a_{i_1 r_1}^{(1)} a_{i_2 r_2}^{(2)} \dots a_{i_N r_N}^{(N)}.$$

**Matricization** For simplicity, we consider the Tucker decomposition for the third-order tensor  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$  where the situation is illustrated in Figure 23.1:

$$\mathbf{X} \approx [\mathbf{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R g_{pqr} \cdot \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r, \quad (23.3)$$



**Figure 23.1:** The Tucker decomposition of a third-order tensor:  $\mathbf{X} \approx [\mathbf{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R g_{pqr} \cdot \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r$ . Middle: **reduced Tucker decomposition**. Right: **full Tucker decomposition** where the blue entries are silent *orthogonal* columns and the white entries of  $\mathbf{G}'$  are zero.

where  $\mathbf{A} \in \mathbb{R}^{I \times P}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times Q}$ ,  $\mathbf{C} \in \mathbb{R}^{K \times R}$ , and  $\mathbf{G} \in \mathbb{R}^{P \times Q \times R}$ . Analogously, following the matricization of the third-order CP decomposition in Equation (22.2), we have

$$\begin{cases} \mathbf{X}_{(1)} \approx \mathbf{A}\mathbf{G}_{(1)}(\mathbf{C} \otimes \mathbf{B})^\top \in \mathbb{R}^{I \times (JK)}; \\ \mathbf{X}_{(2)} \approx \mathbf{B}\mathbf{G}_{(2)}(\mathbf{C} \otimes \mathbf{A})^\top \in \mathbb{R}^{J \times (IK)}; \\ \mathbf{X}_{(3)} \approx \mathbf{C}\mathbf{G}_{(3)}(\mathbf{B} \otimes \mathbf{A})^\top \in \mathbb{R}^{K \times (IJ)}, \end{cases} \quad (23.4)$$

where now “ $\otimes$ ” is the Kronecker product (Definition 21.1, p. 397).

In full generality, come back to the Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , the mode- $n$  matricized form is given by

$$\underbrace{\mathbf{X}_{(n)}}_{I_n \times (I_{-n})} \approx \underbrace{\mathbf{A}_{(n)}}_{I_n \times R_n} \underbrace{\mathbf{G}_{(n)}}_{R_n \times (R_{-n})} \underbrace{\left( \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \dots \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \right)^\top}_{(R_{-n}) \times (I_{-n})}$$

where  $I_{-n} = I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N$  and  $R_{-n} = R_1 R_2 \dots R_{n-1} R_{n+1} \dots R_N$ .

**Vectorization** Going further from the matricization for the Nth-order tensor  $\mathbf{X}$ , the vectorization is given by

$$\underbrace{\text{vec}(\mathbf{X})}_{(I_1 \dots I_N) \times 1} \approx \underbrace{\left( \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(1)} \right)^\top}_{(I_1 \dots I_N) \times (R_1 \dots R_N)} \underbrace{\text{vec}(\mathbf{G})}_{(R_1 \dots R_N) \times 1}. \quad (23.5)$$

**Counterpart in  $\mathbf{G}$**  For the problem in Equation (23.2), it also has the matricization and vectorization in a similar way:

$$\mathbf{G}_{(n)} \approx \mathbf{A}^{(n)\top} \mathbf{G}_{(n)} \left( \mathbf{A}^{(N)\top} \otimes \mathbf{A}^{(N-1)\top} \otimes \dots \otimes \mathbf{A}^{(n+1)\top} \otimes \mathbf{A}^{(n-1)\top} \otimes \dots \otimes \mathbf{A}^{(2)\top} \otimes \mathbf{A}^{(1)\top} \right)^\top$$

$$\underbrace{\text{vec}(\mathbf{G})}_{(R_1 \dots R_N) \times 1} \approx \underbrace{\left( \mathbf{A}^{(N)\top} \otimes \mathbf{A}^{(N-1)\top} \otimes \dots \otimes \mathbf{A}^{(1)\top} \right)^\top}_{(R_1 \dots R_N) \times (I_1 \dots I_N)} \underbrace{\text{vec}(\mathbf{X})}_{(I_1 \dots I_N) \times 1}. \quad (23.6)$$

**Connection to the CP decomposition** We notice that when  $\mathbf{G}$  is the identity tensor (Definition 21.1, p. 393),  $R_1 = R_2 = \dots = R_N$ , and we do not further restrict the columns of  $\mathbf{A}^{(n)}$ 's are mutually orthonormal, then the Tucker decomposition reduces to a CP decomposition. Note that the  $\mathbf{G}$  has to be an identity tensor in this case and it cannot be simply treated as a tensor with all 1's elements where the Tucker decomposition is given by:

$$\mathbf{X} \approx [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r_1=1}^R \sum_{r_2=1}^R \dots \sum_{r_N=1}^R 1 \cdot \mathbf{a}_{r_1}^{(1)} \circ \mathbf{a}_{r_2}^{(2)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)},$$

which is slightly different to the CP decomposition that has only one summation.

### 23.2. Computing the Tucker Decomposition

---

#### Algorithm 64 Tucker Decomposition via ALS

---

**Require:** Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ;

- 1: Pick a rank  $R_1, R_2, \dots, R_N$ ;
  - 2: Initialize  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$  for all  $n \in \{1, 2, \dots, N\}$  randomly;
  - 3: Choose maximal number of iterations  $C$ ;
  - 4:  $iter = 0$ ;
  - 5: **while**  $iter < C$  **do**
  - 6:      $iter = iter + 1$ ;
  - 7:     **for**  $n = 1, 2, \dots, N$  **do**
  - 8:         Set  $\mathbf{Y} = [\mathbf{X}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(n-1)\top}, \mathbf{I}, \mathbf{A}^{(n+1)\top}, \dots, \mathbf{A}^{(N)\top}]$ ;
  - 9:         Find the matricization along mode- $n$ :  $\mathbf{Y}_{(n)}$ ;
  - 10:         Set the rows of  $\mathbf{A}^{(n)}$  by first  $R_n$  leading left singular vectors of  $\mathbf{Y}_{(n)}$ ;
  - 11:     **end for**
  - 12: **end while**
  - 13:  $\mathbf{G} = [\mathbf{X}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(N)\top}]$ ; ▷ by vectorize and un-vectorize, Eq. (23.7)
  - 14: Output  $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}, \mathbf{G}$ ;
- 

To compute the Tucker decomposition of  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , we now consider algorithms for solving the problem:

$$\{\mathbf{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}\} = \arg \min_{\mathbf{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}} \left\| \mathbf{X} - [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \right\|^2,$$

where  $\mathbf{G}$  is a tensor of size  $R_1 \times R_2 \times \dots \times R_N$ ,  $\mathbf{A}^{(n)}$ 's are semi-orthogonal of size  $I_n \times R_n$  for  $n \in \{1, 2, \dots, N\}$ . Similar to the CP decomposition, an *alternating descent* algorithm can be employed to find the solution approximately.

**Given**  $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}$ , **Update**  $\mathbf{G}$ :

The update on  $\mathbf{G}$  follows immediately since we mentioned in Equation (23.2) that

$$\mathbf{G} = [\mathbf{X}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(N)\top}] = \mathbf{X} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \dots \times_N \mathbf{A}^{(N)\top}.$$

To simplify matters, from the vectorized form in Equation (23.5), we have

$$\begin{aligned} & \left\| \mathbf{x} - [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \right\|^2 \\ &= \left\| \text{vec}(\mathbf{x}) - \left( \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(1)} \right) \text{vec}(\mathbf{G}) \right\|^2, \end{aligned}$$

which is just the least squares problem and the solution is given by

$$\begin{aligned} \text{vec}(\mathbf{G}) &\leftarrow \left( \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(1)} \right)^+ \text{vec}(\mathbf{x}) \\ &= \left( \mathbf{A}^{(N)\top} \otimes \mathbf{A}^{(N-1)\top} \otimes \dots \otimes \mathbf{A}^{(1)\top} \right) \text{vec}(\mathbf{x}), \end{aligned} \quad (23.7)$$

where the last equality comes from Equation (21.9) since  $\mathbf{A}^{(n)}$ 's are semi-orthogonal. As long as we update for the vectorized version, an un-vectorization operation can be applied to find the updated  $\mathbf{G}$ .

**Given  $\mathbf{A}^{(1)}, \dots, \mathbf{A}^{(n-1)}, \mathbf{A}^{(n+1)}, \dots, \mathbf{A}^{(N)}$  and  $\mathbf{G}$ , Update  $\mathbf{A}^{(n)}$ :**

When all but  $\mathbf{A}^{(n)}$  are fixed, by Equation (21.1), the Frobenius norm of the difference is given by

$$\begin{aligned} & \left\| \mathbf{x} - [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \right\|^2 \\ &= \|\mathbf{x}\|^2 - 2\langle \mathbf{x}, [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \rangle + \left\| [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \right\|^2, \end{aligned}$$

where by Lemma 21.2 (3) and Equation (23.1) (23.2), it follows that

$$\begin{aligned} & \langle \mathbf{x}, [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \rangle \\ &= \langle [\mathbf{x}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(N)\top}], \mathbf{G} \rangle \quad (\text{Lemma 21.2 (3), Equation (23.1)}) \\ &= \langle \mathbf{G}, \mathbf{G} \rangle = \|\mathbf{G}\|^2, \quad (\text{Equation (23.2)}) \end{aligned}$$

and where by Equation (23.1) and Lemma 21.2 (4), the length preservation under semi-orthogonal, we obtain

$$\left\| [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \right\|^2 = \|\mathbf{G}\|^2.$$

Combine all the findings,

$$\left\| \mathbf{x} - [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] \right\|^2 = \|\mathbf{x}\|^2 - \|\mathbf{G}\|^2.$$

Hence, minimizing the left of the above equation is equivalent to maximizing  $\|\mathbf{G}\|^2$ . Therefore, to update  $\mathbf{A}^{(n)}$ , the problem becomes

$$\begin{aligned} \max_{\mathbf{A}^{(n)}} \mathbf{G} &= \max_{\mathbf{A}^{(n)}} \left\| \mathbf{x} - [\mathbf{G}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(n-1)\top}, \mathbf{A}^{(n)}, \mathbf{A}^{(n+1)}, \dots, \mathbf{A}^{(N)\top}] \right\|^2 \\ \text{subject to } \mathbf{A}^{(n)} &\in \mathbb{R}^{I_n \times R_n} \text{ is semi-orthogonal.} \end{aligned}$$

By defining  $\mathbf{Y} = [\mathbf{X}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(n-1)\top}, \mathbf{I}, \mathbf{A}^{(n+1)\top}, \dots, \mathbf{A}^{(N)\top}]$ . From Lemma 21.1, this again, is equivalent to find the solution of

$$\boxed{\begin{aligned} \max_{\mathbf{A}^{(n)}} & \|\mathbf{Y} \times_n \mathbf{A}^{(n)}\| = \max_{\mathbf{A}^{(n)}} \|\mathbf{A}^{(n)} \mathbf{Y}_{(n)}\|_F \\ \text{subject to } & \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ is semi-orthogonal,} \end{aligned}}$$

where  $\mathbf{Y}_{(n)} \in \mathbb{R}^{R_n \times R_1 \dots R_{n-1} R_{n+1} \dots R_N}$  is the matricization along the mode- $n$  of  $\mathbf{Y}$ . The solution is by setting the rows of  $\mathbf{A}^{(n)}$  as the first  $R_n$  leading left singular vectors of  $\mathbf{Y}_{(n)}$ , (Kolda, 2006; Kolda and Bader, 2009). We notice the above update on  $\mathbf{A}^{(n)}$  is not dependent on  $\mathbf{G}$  finally, and thus we can update the  $\mathbf{G}$  once after the convergence of  $\mathbf{A}^{(n)}$ 's. The procedure is shown in Algorithm 64.

**Initialization by High-Order SVD (HOSVD)** In Algorithm 64, we initialize  $\mathbf{A}^{(n)}$ 's randomly. However, the High-Order Singular Value Decomposition can be utilized as the initial point. The problem is given by

$$\boxed{\begin{aligned} \max_{\mathbf{A}^{(n)}} & [\mathbf{X}; \mathbf{I}, \mathbf{I}, \dots, \mathbf{I}, \mathbf{A}^{(n)\top}, \mathbf{I}, \dots, \mathbf{I}] \\ \text{subject to } & \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ is semi-orthogonal.} \end{aligned}}$$

Similarly, the problem is equal to finding the solution of

$$\boxed{\begin{aligned} \max_{\mathbf{A}^{(n)}} & \|\mathbf{X} \times_n \mathbf{A}^{(n)\top}\| = \|\mathbf{A}^{(n)\top} \mathbf{X}_{(n)}\|_F \\ \text{subject to } & \mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n} \text{ is semi-orthogonal,} \end{aligned}} \quad (23.8)$$

which is solved by setting columns of  $\mathbf{A}^{(n)}$  as the first  $R_n$  leading left singular vectors of  $\mathbf{X}_{(n)}$ . And we shall shortly see an equivalent derivation by the matricization of the HOSVD in the next section, see Equation (24.6).

## Chapter 24

# High-Order SVD (HOSVD)

### Contents

---

|             |                                |            |
|-------------|--------------------------------|------------|
| <b>24.1</b> | <b>High-Order SVD (HOSVD)</b>  | <b>414</b> |
| <b>24.2</b> | <b>Computing the HOSVD</b>     | <b>416</b> |
| <b>24.3</b> | <b>Properties of the HOSVD</b> | <b>417</b> |
| 24.3.1      | Frobenius Norm                 | 417        |
| 24.3.2      | Low-Rank Approximation         | 417        |

---

### 24.1. High-Order SVD (HOSVD)

We mentioned that the HOSVD can be utilized as the initialization for the calculation of the Tucker decomposition. We now consider the properties of the HOSVD.

#### Theorem 24.1: (High-Order SVD (HOSVD))

The HOSVD factorizes a tensor into a sum of component rank-one tensors. For a general Nth-order tensor,  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , it admits the HOSVD

$$\mathbf{X} \approx [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} \mathbf{a}_{r_1}^{(1)} \circ \mathbf{a}_{r_2}^{(2)} \circ \dots \circ \mathbf{a}_{r_N}^{(N)},$$

where

- $R_1 < I_1, R_2 < I_2, \dots, R_N < I_N$ ;
- $\mathbf{G}$  of size  $R_1 \times R_2 \times \dots \times R_N$  is called the *core tensor* so that  $\mathbf{G}$  can be thought of as the compressed version of  $\mathbf{X}$ ;
- $\mathbf{A}^{(n)} = [\mathbf{a}_1^{(n)}, \mathbf{a}_2^{(n)}, \dots, \mathbf{a}_{R_n}^{(n)}] \in \mathbb{R}^{I_n \times R_n}$  for all  $n \in \{1, 2, \dots, N\}$  is the column partition of the matrix  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times B_n}$ ;
- The  $\mathbf{A}^{(n)}$ 's have mutually orthonormal columns and can be thought of as the principal component of each mode. In this sense, the  $\mathbf{A}^{(n)}$ 's are *semi-orthogonal matrices* (see the definition in Section 3.5, p. 87);
- We can complete the semi-orthogonal matrices into *full orthogonal matrices* by adding *silent columns* into  $\mathbf{A}^{(n)}$ 's so that  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times I_n}$  is an orthogonal matrix, in which case,  $\mathbf{G}$  will be expanded to a tensor of size  $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  where  $g_{r_1 r_2 \dots r_N} = 0$  when either one of  $r_n > R_n$  for  $n \in \{1, 2, \dots, N\}$ . This is known as the *full HOSVD*, and the previous one is also called the *reduced* one to avoid confusion; And we shall only consider the reduced case in most of our discussions.

**Till now, the HOSVD is the same as the Tucker decomposition.** The difference is as follows:

- *All orthogonality.* The slices in each mode are mutually orthogonal. Suppose  $\mathbf{G}_{r_n=\alpha}$  is the slice of  $\mathbf{G}$  where the  $n$ -th index is set to  $\alpha$ , i.e., an  $(N-1)$ -th-order subtensor:  $\mathbf{G}_{r_n=\alpha} = \mathbf{G}_{:, \dots, :, r_n=\alpha, :, \dots, :}$ , then it follows that

$$\langle \mathbf{G}_{r_n=\alpha}, \mathbf{G}_{r_n=\beta} \rangle = 0, \quad \alpha \neq \beta \in \{1, 2, \dots, R_n\}, \quad (24.1)$$

for all possible value of  $n \in \{1, 2, \dots, N\}$ .

- *Ordering.* The Frobenius norms of slices in each mode are decreasing with the increase in the running index:

$$\|\mathbf{G}_{r_n=1}\|_F \geq \|\mathbf{G}_{r_n=2}\|_F \geq \dots \geq \|\mathbf{G}_{r_n=I_n}\|_F \geq 0, \quad (24.2)$$

for all possible value of  $n \in \{1, 2, \dots, N\}$ .

The Frobenius norm  $\|\mathbf{G}_{r_n=i}\|_F$  is usually denoted as  $\|\mathbf{G}_{r_n=i}\|_F = \sigma_i^{(n)}$ , and known as the *mode-n singular values of  $\mathbf{X}$* . And the vector  $\mathbf{a}_i^{(n)}$  is an *i-th mode-n singular vector*. Comparison of the matrix and tensor SVD reveals a clear analogy between the two cases. In matrix language, suppose the reduced SVD of  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \in \mathbb{R}^{M \times N}$ , the singular values in matrix  $\Sigma \in \mathbb{R}^{R \times R}$  is the norm of each row or each column of  $\Sigma$  (since  $\Sigma$  is diagonal). Left and right singular vectors in the columns of  $\mathbf{U}, \mathbf{V}$  respectively now are generalized as the mode- $n$  singular vectors.

**Equivalent forms on the HOSVD** The equivalent forms on the HOSVD is just the same as the Tucker decomposition. For the Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , by the mode- $n$  tensor multiplication in Equation (21.2), the HOSVD can also be written as:

$$\mathbf{X} \approx [\mathbf{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}] = \mathbf{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \dots \times_N \mathbf{A}^{(N)}. \quad (24.3)$$

Note the analogy of SVD for a matrix in Equation (21.3):

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top = \Sigma \times_1 \mathbf{U} \times_2 \mathbf{V} \in \mathbb{R}^{M \times N}. \quad (24.4)$$

By the result in Lemma 21.2 (4), since  $\mathbf{A}^{(n)}$ 's are semi-orthogonal, it also follows that

$$\mathbf{G} = [\mathbf{X}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(N)\top}] = \mathbf{X} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \dots \times_N \mathbf{A}^{(N)\top}. \quad (24.5)$$

Element-wise, the  $(i_1, i_2, \dots, i_N)$ -th element of  $\mathbf{X}$  can be obtained by

$$\mathbf{X}_{i_1, i_2, \dots, i_N} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N} a_{i_1 r_1}^{(1)} a_{i_2 r_2}^{(2)} \dots a_{i_N r_N}^{(N)}.$$

**Matricization** In full generality, for the Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , the mode- $n$  matricized form is given by

$$\underbrace{\mathbf{X}_{(n)}}_{I_n \times (I_{-n})} \approx \underbrace{\mathbf{A}^{(n)}}_{I_n \times R_n} \underbrace{\mathbf{G}_{(n)}}_{R_n \times (R_{-n})} \underbrace{\left( \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \dots \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \right)^\top}_{(R_{-n}) \times (I_{-n})}$$

where  $I_{-n} = I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N$  and  $R_{-n} = R_1 R_2 \dots R_{n-1} R_{n+1} \dots R_N$ . Moreover, since the conditions in Equation (24.1) and (24.2), this implies  $\mathbf{G}_{(n)}$  has mutually orthogonal rows (not orthonormal), having vector Frobenius norms equal to  $\sigma_1^{(n)}, \sigma_2^{(n)}, \dots, \sigma_{R_n}^{(n)}$ .

**Matrix SVD in HOSVD** Let further the diagonal matrix

$$\Sigma^{(n)} = \text{diag} \left( \sigma_1^{(n)}, \sigma_2^{(n)}, \dots, \sigma_{R_n}^{(n)} \right) \in \mathbb{R}^{R_n \times R_n},$$

where  $\sigma_i^{(n)} = \|\mathbf{G}_{r_n=i}\|_F$ . This implies, for the row normalized version  $\tilde{\mathbf{G}}_{(n)}$  of  $\mathbf{G}_{(n)}$ , we have

$$\underbrace{\mathbf{G}_{(n)}}_{R_n \times R_{-n}} = \underbrace{\Sigma^{(n)}}_{R_n \times R_n} \underbrace{\tilde{\mathbf{G}}_{(n)}}_{R_n \times R_{-n}}.$$

Let further,

$$\underbrace{\mathbf{V}^{(n)\top}}_{R_n \times I_{-n}} = \underbrace{\tilde{\mathbf{G}}_{(n)}}_{R_n \times R_{-n}} \underbrace{\left( \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes \dots \otimes \mathbf{A}^{(2)} \otimes \mathbf{A}^{(1)} \right)^\top}_{(R_{-n}) \times (I_{-n})},$$

where the columns of  $\mathbf{V}^{(n)}$  are mutually orthonormal by Equation (21.10) and (21.11). This reveals the (matrix) reduced SVD of  $\mathbf{X}_{(n)}$ :

$$\boxed{\underbrace{\mathbf{X}_{(n)}}_{I_n \times (I_{-n})} \approx \underbrace{\mathbf{A}^{(n)}}_{I_n \times R_n} \underbrace{\boldsymbol{\Sigma}^{(n)}}_{R_n \times R_n} \underbrace{\mathbf{V}^{(n)\top}}_{R_n \times I_{-n}}.} \quad (24.6)$$

And if  $\{\mathbf{G}, \mathbf{A}^{(N)}, \dots, \mathbf{A}^{(n+1)}, \mathbf{A}^{(n-1)}, \dots, \mathbf{A}^{(1)}\}$  are fixed, the update on  $\mathbf{A}^{(n)}$  can be obtained by setting the columns of it via the first  $R_n$  left singular vectors of  $\mathbf{X}_{(n)}$ . This matches the update in the subproblem of Equation (23.8). The uniqueness of the mode- $n$  singular values thus comes from the uniqueness of the (matrix) reduced SVD.

**Vectorization** Going further from the matricization for the Nth-order tensor  $\mathbf{X}$ , the vectorization is given by

$$\boxed{\underbrace{\text{vec}(\mathbf{X})}_{(I_1 \dots I_N) \times 1} \approx \underbrace{\left( \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \dots \otimes \mathbf{A}^{(1)} \right)}_{(I_1 \dots I_N) \times (R_1 \dots R_N)} \underbrace{\text{vec}(\mathbf{G})}_{(R_1 \dots R_N) \times 1}.} \quad (24.7)$$

**Third-order case** For simplicity and a better understanding, we consider the HOSVD for the third-order tensor  $\mathbf{X} \in \mathbb{R}^{I \times J \times K}$ :

$$\mathbf{X} \approx [\mathbf{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}] = \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R g_{pqr} \cdot \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r, \quad (24.8)$$

where  $\mathbf{A} \in \mathbb{R}^{I \times P}$ ,  $\mathbf{B} \in \mathbb{R}^{J \times Q}$ ,  $\mathbf{C} \in \mathbb{R}^{K \times R}$ , and  $\mathbf{G} \in \mathbb{R}^{P \times Q \times R}$ . Then, it follows that

- All orthogonality.

$$\langle \mathbf{G}_{:, \alpha, :}, \mathbf{G}_{:, \beta, :} \rangle = 0, \quad \alpha \neq \beta \in \{1, 2, \dots, J\}.$$

- Ordering. The Frobenius norms of slices in each mode are decreasing with the increase in the running index:

$$\|\mathbf{G}_{:, 1, :}\|_F \geq \|\mathbf{G}_{:, 2, :}\|_F \geq \dots \geq \|\mathbf{G}_{:, J, :}\|_F \geq 0.$$

The illustration of this third-order HOSVD (reduced and full versions) is similar to that of the Tucker decomposition and is shown in Figure 23.1.

## 24.2. Computing the HOSVD

The calculation of the HOSVD is just the (matrix) reduced SVD from the matricized form of the HOSVD in Equation (24.6). The procedure is shown in Algorithm 65 and we shall not repeat the details. See also (De Lathauwer et al., 2000) for a more detailed discussion.

---

**Algorithm 65** HOSVD via ALS

---

**Require:** Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ;

- 1: Pick a rank  $R_1, R_2, \dots, R_N$ ;
- 2: initialize  $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$  for all  $n \in \{1, 2, \dots, N\}$  randomly;
- 3: choose maximal number of iterations  $C$ ;
- 4:  $iter = 0$ ;
- 5: **while**  $iter < C$  **do**
- 6:      $iter = iter + 1$ ;
- 7:     **for**  $n = 1, 2, \dots, N$  **do**
- 8:         Find the matricization along mode- $n$ :  $\mathbf{Y}_{(n)}$ ;
- 9:         Set the rows of  $\mathbf{A}^{(n)}$  by first  $R_n$  leading left singular vectors of  $\mathbf{X}_{(n)}$ ;
- 10:       **end for**
- 11:     **end while**
- 12:      $\mathcal{G} = [\mathbf{X}; \mathbf{A}^{(1)\top}, \mathbf{A}^{(2)\top}, \dots, \mathbf{A}^{(N)\top}]$ ;
- 13:     Output  $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}, \mathcal{G}$ ;

---

## 24.3. Properties of the HOSVD

### 24.3.1 Frobenius Norm

**Lemma 24.1: (Frobenius Norm of a Tensor)**

Suppose the HOSVD of the Nth-order tensor  $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is given by  $\mathbf{X} \approx [\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}]$ , then it follows that

$$\begin{aligned} \|\mathbf{X}\|_F^2 &= \|\mathcal{G}\|_F^2 \\ &= \sum_{i=1}^{R_1} (\sigma_i^{(1)})^2 = \sum_{i=1}^{R_2} (\sigma_i^{(2)})^2 = \dots = \sum_{i=1}^{R_N} (\sigma_i^{(N)})^2. \end{aligned}$$

The squared Frobenius norm of a matrix is defined to be the sum of squares of the singular values of the matrix (Definition 14.2, p. 281). The above lemma tells us that the singular values in each slice of a tensor have a similar property. The proof of the above lemma follows immediately from the (matrix) reduced SVD of the matricized form of the HOSVD in Equation (24.6).

### 24.3.2 Low-Rank Approximation

We discussed the best rank- $k$  approximation of a matrix is from the truncated SVD (Theorem 14.4, p. 284) where

$$\|\mathbf{A} - \mathbf{A}_k\|_F^2 = \sigma_{k+1}^2 + \sigma_{k+2}^2 + \dots + \sigma_{\min\{M,N\}}^2$$

if  $\mathbf{A} \in \mathbb{R}^{M \times N}$ . Similar result follows in the lower-rank tensor approximation.

**Lemma 24.2: (Low-Rank Approximation)**

Suppose the HOSVD of the  $N$ th-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  is given by  $\mathcal{X} \approx [\mathcal{G}; \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(N)}]$ . Define a tensor  $\widehat{\mathcal{X}}$  by discarding the smallest mode- $n$  singular values  $\sigma_{K_n+1}^{(n)}, \sigma_{K_n+2}^{(n)}, \dots, \sigma_{R_n}^{(n)}$  for given values of  $K_n \leq R_n$  (for all  $n \in \{1, 2, \dots, N\}$ ). Then, it follows that

$$\|\mathcal{X} - \widehat{\mathcal{X}}\|_F^2 \leq \sum_{r_1=K_1+1}^{R_1} (\sigma_{r_1}^{(1)})^2 + \sum_{r_2=K_2+1}^{R_2} (\sigma_{r_2}^{(2)})^2 + \dots + \sum_{r_N=K_N+1}^{R_N} (\sigma_{r_N}^{(N)})^2.$$

**Proof** [of Lemma 24.2] From Lemma 24.1, we have the following

$$\begin{aligned} \|\mathcal{X} - \widehat{\mathcal{X}}\|_F^2 &= \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N}^2 - \sum_{r_1=1}^{K_1} \sum_{r_2=1}^{K_2} \dots \sum_{r_N=1}^{K_N} g_{r_1 r_2 \dots r_N}^2 \\ &= \sum_{r_1=K_1+1}^{R_1} \sum_{r_2=K_2+1}^{R_2} \dots \sum_{r_N=K_N+1}^{R_N} g_{r_1 r_2 \dots r_N}^2 \\ &\leq \sum_{\substack{r_1=K_1+1 \\ r_2=K_2+1}}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N}^2 \\ &\quad + \sum_{\substack{r_1=1 \\ r_2=K_2+1}}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N}^2 \\ &\quad + \dots + \sum_{\substack{r_1=1 \\ r_N=K_N+1}}^{R_1} \sum_{r_2=1}^{R_2} \dots \sum_{r_N=1}^{R_N} g_{r_1 r_2 \dots r_N}^2 \\ &= \sum_{r_1=K_1+1}^{R_1} (\sigma_{r_1}^{(1)})^2 + \sum_{r_2=K_2+1}^{R_2} (\sigma_{r_2}^{(2)})^2 + \dots + \sum_{r_N=K_N+1}^{R_N} (\sigma_{r_N}^{(N)})^2. \end{aligned}$$

This completes the proof. ■

## Chapter 25

# Tensor-Train (TT) Decomposition

### Contents

---

|      |                                 |     |
|------|---------------------------------|-----|
| 25.1 | Tensor-Train (TT) Decomposition | 420 |
| 25.2 | Computing the TT Decomposition  | 421 |

---

### 25.1. Tensor-Train (TT) Decomposition

**Theorem 25.1: (Tensor-Train Decomposition (Oseledets, 2011))**

For a general Nth-order tensor,  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , it admits the tensor-train decomposition

$$\mathcal{X} \approx \mathcal{G}^{(1)} \boxtimes \mathcal{G}^{(2)} \boxtimes \dots \boxtimes \mathcal{G}^{(N)},$$

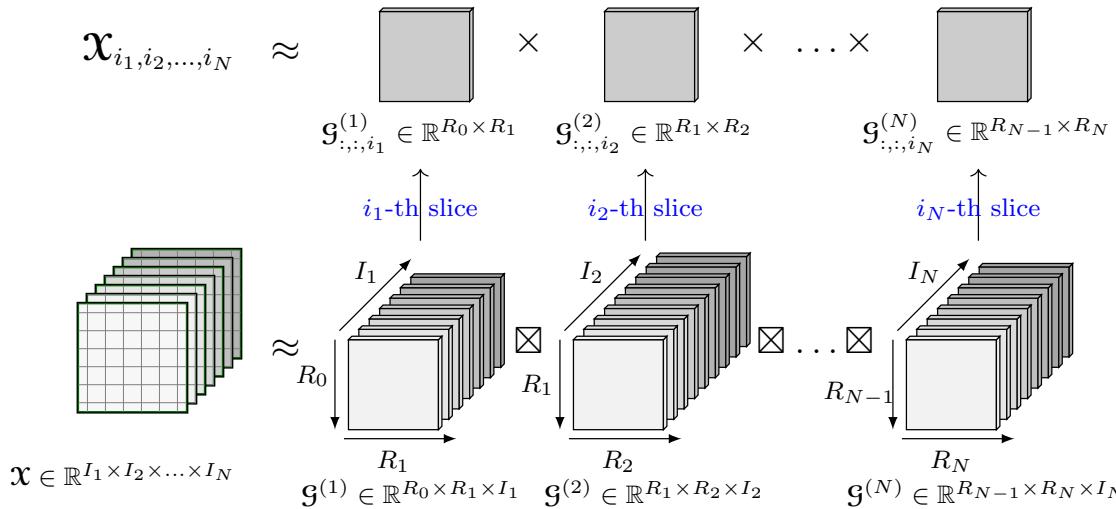
where the symbol “ $\boxtimes$ ” means the elements of  $\mathcal{X}$  can be obtained by a multiplication of  $N$  matrices

$$x_{i_1, i_2, \dots, i_N} = \mathcal{G}_{:, :, i_1}^{(1)} \mathcal{G}_{:, :, i_2}^{(2)} \dots \mathcal{G}_{:, :, i_N}^{(N)}.$$

Note here

- Each  $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times R_n \times I_n}$  for all  $n \in \{1, 2, \dots, N\}$  is a third-order tensor, and is referred to as a *TT core*;
- Each  $\mathcal{G}_{:, :, i_n}^{(n)} \in \mathbb{R}^{R_{n-1} \times R_n}$  for all  $i_n \in \{1, 2, \dots, I_n\}$  is a frontal slice of  $\mathcal{G}^{(n)}$  (see Figure 21.2);
- $R_0, R_N$  are imposed to be 1 for boundary conditions;
- $R_1, R_2, \dots, R_{N-1}$  are known as the *tensor ranks* of corresponding dimensions;

The illustration of how to extract each element of the decomposition is shown in Figure 25.1.



**Figure 25.1:** Tensor-train decomposition of an Nth-order tensor:  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N} \approx \mathcal{G}^{(1)} \boxtimes \mathcal{G}^{(2)} \boxtimes \dots \boxtimes \mathcal{G}^{(N)}$  where each  $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times R_n \times I_n}$  for  $n \in \{1, 2, \dots, N\}$ . Each  $(i_1, i_2, \dots, i_N)$ -th element is obtained by the matrix multiplication of the corresponding frontal slices of  $\mathcal{G}^{(n)}$ 's. Note here  $R_0 = R_N = 1$ .

The illustration of the TT decomposition for an Nth order tensor is shown in Figure 25.1. In other words, the TT format approximates every entry of the tensor  $\mathcal{X}$  with a multiplica-

tion of  $N$  matrices, in particular with a sequence of  $R_n \times R_{n+1}$  matrices, each indexed by the parameter  $i_{n+1}$ . The figure looks like a train with links between them, hence the name “train”. Suppose that the TT-ranks are all equal,  $R_1 = R_2 = \dots = R_{N-1} = R$ , and that  $I_1 = I_2 = \dots = I_N = I$ , then the TT decomposition requires the storage of  $O(NIR^2)$  floats. Thus the memory complexity of the TT decomposition scales linearly with dimension.

**The “ $\boxtimes$ ” notation** The symbol “ $\boxtimes$ ” is defined to be a special tensor product. Suppose  $\mathcal{A} \in \mathbb{R}^{R_1 \times R_2 \times I_1 \times I_2 \times \dots \times I_M}$ ,  $\mathcal{B} \in \mathbb{R}^{R_2 \times R_3 \times J_1 \times J_2 \times \dots \times J_N}$ , then it follows that

$$\mathcal{A} \boxtimes \mathcal{B} \in \mathbb{R}^{R_1 \times R_3 \times I_1 \times I_2 \times \dots \times I_M \times J_1 \times J_2 \times \dots \times J_N},$$

where each element is given by

$$\begin{aligned} & (\mathcal{A} \boxtimes \mathcal{B})_{r_1, r_3, i_1, i_2, \dots, i_M, j_1, j_2, \dots, j_N} \\ &= \sum_{r_2=1}^{R_2} (a_{r_1, \textcolor{blue}{r}_2, i_1, i_2, \dots, i_M})(b_{\textcolor{blue}{r}_2, r_3, j_1, j_2, \dots, j_N}). \end{aligned}$$

## 25.2. Computing the TT Decomposition

---

### Algorithm 66 TT-SVD

---

**Require:** Nth-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ;

- 1: Set the initial matrix  $\mathbf{C} = \overset{(1)}{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 I_3 \dots I_N}$ ;
  - 2: Set  $R_0 = R_N = 1$ ;
  - 3: **for**  $n = 1, 2, \dots, N - 1$  **do**
  - 4:      $\mathbf{C} = \text{reshape}\left(\mathbf{C}, (I_n R_{n-1}), (I_{n+1} I_{n+2} \dots I_N)\right)$ ;
  - 5:     Compute the  $\delta$ -truncated SVD of  $\mathbf{C} = \mathbf{U} \boldsymbol{\Sigma} \mathbf{V}^\top$ , numerical rank  $R_n = \text{rank}(\mathbf{C})$ ;
  - 6:     Set the rows of  $\mathbf{G}_{(2)}^{(n)} \in \mathbb{R}^{R_n \times I_n R_{n-1}}$  by first  $R_n$  leading left singular vectors of  $\mathbf{C}$ ;
  - 7:     Un-matricization:  $\mathcal{G}^{(n)} = \text{reshape}(\mathbf{G}_{(2)}^{(n)}) \in \mathbb{R}^{R_{n-1} \times R_n \times I_n}$ ;
  - 8:     Get the new matrix  $\mathbf{C} = \boldsymbol{\Sigma} \mathbf{V}^\top \in \mathbb{R}^{R_n \times I_{n+1} I_{n+2} \dots I_N}$ ;
  - 9: **end for**
  - 10: Get the last core tensor:  $\mathcal{G}^{(N)} = \text{reshape}(\mathbf{C}) \in \mathbb{R}^{R_{N-1} \times R_N \times I_N}$ ;
  - 11: Output  $\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \dots, \mathcal{G}^{(N)}$ ;
- 

Define the *tensor unfolding* for an Nth-order tensor  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  in the following way

$$\overset{(n)}{\mathbf{X}} \in \mathbb{R}^{(I_1 I_2 \dots I_n) \times (I_{n+1} I_{n+2} \dots I_N)},$$

where the  $\left(\{i_1 \dots i_n\}, \{i_{n+1} \dots i_N\}\right)$ -th element of  $\overset{(n)}{\mathbf{X}}$  is obtained by  $\overset{(n)}{\mathbf{X}}_{i_1 \dots i_n, i_{n+1} \dots i_N} = \mathcal{X}_{i_1, i_2, \dots, i_N}$ . The tensor unfolding can be denoted by a reshape operator:

$$\overset{(n)}{\mathbf{X}} = \text{reshape}\left(\mathcal{X}, (I_1 I_2 \dots I_n), (I_{n+1} I_{n+2} \dots I_N)\right).$$

This unfolding reveals a recursive algorithm to calculate the TT decomposition of  $\mathbf{X}$ .

For this tensor unfolding along mode-1:

$$\overset{(1)}{\mathbf{X}} \in \mathbb{R}^{I_1 \times I_2 I_3 \dots I_N}.$$

Recall the matricization in mode-2 of  $\mathbf{G}^{(1)}$  is given by (Section 21.6, p. 394):

$$\mathbf{G}_{(2)}^{(1)} \in \mathbb{R}^{R_1 \times R_0 I_1} = \mathbb{R}^{R_1 \times I_1}.$$

Then the  $\mathbf{G}_{(2)}^{(1)}$  can be thought of as the data distilled version of  $\overset{(1)}{\mathbf{X}}$ , i.e., the row space of  $\mathbf{G}_{(2)}^{(1)}$  span the same column space as that of  $\overset{(1)}{\mathbf{X}}$ . And therefore, the  $R_1$  can be determined by the numerical rank (Definition 14.6, p. 269) of the SVD of  $\overset{(1)}{\mathbf{X}}$ , i.e., by discarding the singular values of  $\overset{(1)}{\mathbf{X}}$  smaller than  $\delta$ . Similar to the Tucker and HOSVD, the row of  $\mathbf{G}_{(2)}^{(1)}$  can be obtained by the first  $R_1$  left singular vectors of  $\overset{(1)}{\mathbf{X}}$ . For the (truncated) SVD of  $\overset{(1)}{\mathbf{X}}$ :

$$\overset{(1)}{\mathbf{X}} = \mathbf{U}_1 \Sigma_1 \mathbf{V}_1^\top$$

$\xrightarrow{\text{leads to}} \boxed{\mathbf{G}_{(2)}^{(1)} = \mathbf{U}_1^\top \in \mathbb{R}^{R_1 \times I_1}}.$

Now, what's left is  $\Sigma_1 \mathbf{V}_1^\top \in \mathbb{R}^{R_1 \times I_2 I_3 \dots I_N}$ . By similar “matrix unfolding”, suppose we reshape  $\Sigma_1 \mathbf{V}_1^\top$  into a  $I_2 R_1 \times I_3 I_4 \dots I_N$  matrix  $\mathbf{C}$ :

$$\mathbf{C} \in \mathbb{R}^{I_2 R_1 \times I_3 I_4 \dots I_N} = \text{reshape}\left(\Sigma_1 \mathbf{V}_1^\top, (I_2 R_1), (I_3 I_4 \dots I_N)\right).$$

The second core tensor  $\mathbf{G}^{(2)}$  can be obtained in the same way via the (truncated) SVD of  $\mathbf{C}$ :

$$\mathbf{C} = \mathbf{U}_2 \Sigma_2 \mathbf{V}_2^\top$$

$\xrightarrow{\text{leads to}} \boxed{\mathbf{G}_{(2)}^{(2)} = \mathbf{U}_2^\top \in \mathbb{R}^{R_2 \times I_2 R_1}}.$

The same process can go on, the eventually all the  $N$  core tensors  $\{\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \dots, \mathbf{G}^{(N)}\}$  will be obtained via the set of (truncated) SVDs. This is known as the TT-SVD algorithm in (Oseledets, 2011). The full procedure is shown in Algorithm 66.

Moreover, if the rank  $R_n \leq \text{rank}(\mathbf{C})$  in each iteration, a low-rank TT best approximation to  $\mathbf{X}$  in Frobenius norm  $\mathbf{X}_{best}$  always exists. And if further the truncation tolerance for the SVD of each unfolding is set to  $\delta = \epsilon / \sqrt{N - 1} \|\mathbf{X}\|_F$ , the TT-SVD is able to construct the quasi-optimal approximation  $\mathbf{X}_{SVD}$  such that

$$\|\mathbf{X} - \mathbf{X}_{SVD}\|_F \leq \sqrt{N - 1} \|\mathbf{X} - \mathbf{X}_{best}\|_F.$$

**Complexity and curse of dimensionality** Suppose again that the TT-ranks are all equal,  $R_1 = R_2 = \dots = R_{N-1} = R$ , and that  $I_1 = I_2 = \dots = I_N = I$ . And the complexity of the SVD calculation is  $O(MN^2)$  if the matrix is of size  $M \times N$  (Section 14.4, p. 273). The complexity of each step  $n$  in Algorithm 66 can be shown to be

$$f(n) = (RI)(I^{(N-n)})^2.$$

A simple summation shows that the complexity of the TT-SVD algorithm is

$$\text{cost} = f(1) + f(2) + \dots + f(N-1) = RI(I^{2(N-1)} + I^{2(N-2)} + I^2) = RI^{N^2-N+1}.$$

So the complexity grows exponentially with dimension  $I$  and thus the curse of dimensionality is not evolved.

**Further calculation methods** We notice that the calculation of the TT decomposition relies on the rank-revealing decomposition, SVD. Other methods, such as the rank-revealing QR (Section 3.12, p. 107), column-pivoted QR (Theorem 3.1, p. 101), CUR (Theorem 6.1, p. 168), UTV (Theorem 4.1, p. 141), Column ID (Theorem 7.1, p. 178) can be applied in each iteration to do the matrix decomposition that finds the spanning columns. We shall not go to the details. See also (Oseledets, 2011; Bigoni et al., 2016).

# Chapter 26

## Acknowledgments

We thank Gilbert Strang for raising the question formulated in Corollary 6.1, checking the writing of the survey, for a stream of ideas and references about the three factorizations from the steps of elimination, and for the generous sharing of the manuscript of (Strang and Drucker, 2021).

# Chapter 27

# Appendix

## Contents

---

|          |                                                                                  |            |
|----------|----------------------------------------------------------------------------------|------------|
| <b>A</b> | <b>Dimension of Column Space and Row Space . . . . .</b>                         | <b>426</b> |
| <b>B</b> | <b>The Fundamental Theorem of Linear Algebra . . . . .</b>                       | <b>427</b> |
| B.1      | Find the Basis of the Four Subspaces via the CR Decomposition                    | 428        |
| <b>C</b> | <b>The Fundamental Theorem of Linear Algebra: A Least Squares View . . . . .</b> | <b>430</b> |
| <b>D</b> | <b>Projection and Orthogonal Projection . . . . .</b>                            | <b>432</b> |
| D.1      | Properties of Symmetric and Idempotent Matrices . . . . .                        | 432        |
| D.2      | Orthogonal Projection and Geometric Interpretation for LS . . .                  | 434        |
| D.3      | Properties of Orthogonal Projection Matrices . . . . .                           | 439        |
| D.4      | Distance Between Subspaces . . . . .                                             | 441        |
| D.5      | Projection for LS with Noise Disturbance . . . . .                               | 443        |
| <b>E</b> | <b>Pseudo-Inverse . . . . .</b>                                                  | <b>445</b> |
| E.1      | One-sided Inverse . . . . .                                                      | 445        |
| E.2      | Generalized Inverse (g-inverse) . . . . .                                        | 448        |
| E.3      | Reflexive Generalized Inverse (rg-inverse) . . . . .                             | 452        |
| E.4      | Pseudo-Inverse . . . . .                                                         | 455        |
| E.5      | Pseudo-Inverse in SVD . . . . .                                                  | 460        |
| E.6      | Pseudo-Inverse in CR Decomposition and Skeleton Decomposition                    | 462        |
| <b>F</b> | <b>Schur Complement . . . . .</b>                                                | <b>464</b> |
| <b>G</b> | <b>General Term Formula of Wedderburn Sequence . . . . .</b>                     | <b>466</b> |
| <b>H</b> | <b>Decoding Orthogonal Matrix Multiplication . . . . .</b>                       | <b>468</b> |
| <b>I</b> | <b>Cochran's Theorem . . . . .</b>                                               | <b>469</b> |
| <b>J</b> | <b>Taylor's Expansion . . . . .</b>                                              | <b>472</b> |
| <b>K</b> | <b>Famous Inequalities . . . . .</b>                                             | <b>473</b> |
| <b>L</b> | <b>Matrix Norm . . . . .</b>                                                     | <b>476</b> |
| L.1      | Vector Norm . . . . .                                                            | 476        |
| L.2      | Matrix Norm . . . . .                                                            | 480        |

---

## Appendix A. Dimension of Column Space and Row Space

We proved in Theorem 4.2 that the row rank and the column rank of any matrix  $\mathbf{A}$  are equal (i.e., the dimension of the column space of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is equal to the dimension of its row space) via CR decomposition. In this appendix, we prove Theorem 4.2 again by an elementary approach from which it also reveals the fundamental theorem of linear algebra.

**Proof [of Theorem 4.2, p. 148, A Third Way]** We first notice that the null space of  $\mathbf{A}$  is orthogonal complementary to the row space of  $\mathbf{A}$ :  $\mathcal{N}(\mathbf{A}) \perp \mathcal{C}(\mathbf{A}^\top)$  (where the row space of  $\mathbf{A}$  is exactly the column space of  $\mathbf{A}^\top$ ), that is, vectors in the null space of  $\mathbf{A}$  are orthogonal to vectors in the row space of  $\mathbf{A}$ . To see this, suppose  $\mathbf{A}$  has rows  $\mathbf{a}_1^\top, \mathbf{a}_2^\top, \dots, \mathbf{a}_m^\top$  and  $\mathbf{A} = [\mathbf{a}_1^\top; \mathbf{a}_2^\top; \dots; \mathbf{a}_m^\top]$ . For any vector  $\mathbf{x} \in \mathcal{N}(\mathbf{A})$ , we have  $\mathbf{Ax} = \mathbf{0}$ , that is,  $[\mathbf{a}_1^\top \mathbf{x}; \mathbf{a}_2^\top \mathbf{x}; \dots; \mathbf{a}_m^\top \mathbf{x}] = \mathbf{0}$ . And since the row space of  $\mathbf{A}$  is spanned by  $\mathbf{a}_1^\top, \mathbf{a}_2^\top, \dots, \mathbf{a}_m^\top$ . Then  $\mathbf{x}$  is perpendicular to any vectors from  $\mathcal{C}(\mathbf{A}^\top)$  which means  $\mathcal{N}(\mathbf{A}) \perp \mathcal{C}(\mathbf{A}^\top)$ .

Now suppose, the dimension of row space of  $\mathbf{A}$  is  $r$ . Let  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r$  be a set of vectors in  $\mathbb{R}^n$  and form a basis for the row space. Then the  $r$  vectors  $\mathbf{Ar}_1, \mathbf{Ar}_2, \dots, \mathbf{Ar}_r$  are in the column space of  $\mathbf{A}$ , which are linearly independent. To see this, suppose we have a linear combination of the  $r$  vectors:  $x_1\mathbf{Ar}_1 + x_2\mathbf{Ar}_2 + \dots + x_r\mathbf{Ar}_r = \mathbf{0}$ , that is,  $\mathbf{A}(x_1\mathbf{r}_1 + x_2\mathbf{r}_2 + \dots + x_r\mathbf{r}_r) = \mathbf{0}$  and the vector  $\mathbf{v} = x_1\mathbf{r}_1 + x_2\mathbf{r}_2 + \dots + x_r\mathbf{r}_r$  is in null space of  $\mathbf{A}$ . But since  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r$  is a basis for the row space of  $\mathbf{A}$ ,  $\mathbf{v}$  is thus also in the row space of  $\mathbf{A}$ . We have shown that vectors from null space of  $\mathbf{A}$  is perpendicular to vectors from row space of  $\mathbf{A}$ , thus  $\mathbf{v}^\top \mathbf{v} = 0$  and  $x_1 = x_2 = \dots = x_r = 0$ . Then  $\mathbf{Ar}_1, \mathbf{Ar}_2, \dots, \mathbf{Ar}_r$  are in the column space of  $\mathbf{A}$  and they are linearly independent which means the dimension of the column space of  $\mathbf{A}$  is larger than  $r$ . This result shows that **row rank of  $\mathbf{A} \leq$  column rank of  $\mathbf{A}$** .

If we apply this process again for  $\mathbf{A}^\top$ . We will have **column rank of  $\mathbf{A} \leq$  row rank of  $\mathbf{A}$** . This completes the proof. ■

Further information can be drawn from this proof is that if  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r$  is a set of vectors in  $\mathbb{R}^n$  that forms a basis for the row space, then  $\mathbf{Ar}_1, \mathbf{Ar}_2, \dots, \mathbf{Ar}_r$  is a basis for the column space of  $\mathbf{A}$ . We formulate this finding into the following lemma.

### Lemma 27.1: (Column Basis from Row Basis)

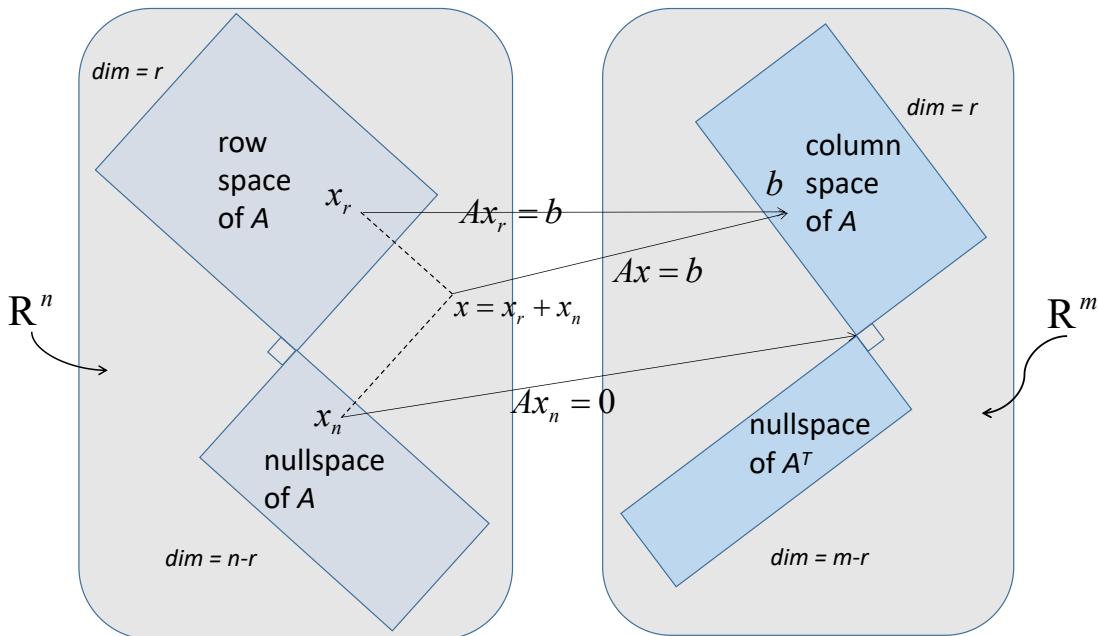
For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , suppose that  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r\}$  is a set of vectors in  $\mathbb{R}^n$  which forms a basis for the row space, then  $\{\mathbf{Ar}_1, \mathbf{Ar}_2, \dots, \mathbf{Ar}_r\}$  is a basis for the column space of  $\mathbf{A}$ .

## Appendix B. The Fundamental Theorem of Linear Algebra

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , it can be easily verified that any vector in the row space of  $\mathbf{A}$  is perpendicular to any vector in the null space of  $\mathbf{A}$ . Suppose  $\mathbf{x}_n \in \mathcal{N}(\mathbf{A})$ , then  $\mathbf{A}\mathbf{x}_n = \mathbf{0}$  such that  $\mathbf{x}_n$  is perpendicular to every row of  $\mathbf{A}$  which agrees with our claim.

Similarly, we can also show that any vector in the column space of  $\mathbf{A}$  is perpendicular to any vector in the null space of  $\mathbf{A}^\top$ . Further, the column space of  $\mathbf{A}$  together with the null space of  $\mathbf{A}^\top$  span the whole  $\mathbb{R}^m$  which is known as the fundamental theorem of linear algebra.

The fundamental theorem contains two parts, the dimension of the subspaces and the orthogonality of the subspaces. The orthogonality can be easily verified as shown above. Moreover, when the row space has dimension  $r$ , the null space has dimension  $n - r$ . This cannot be easily stated and we prove in the following theorem.



**Figure 27.1:** Two pairs of orthogonal subspaces in  $\mathbb{R}^n$  and  $\mathbb{R}^m$ .  $\dim(\mathcal{C}(\mathbf{A}^\top)) + \dim(\mathcal{N}(\mathbf{A})) = n$  and  $\dim(\mathcal{N}(\mathbf{A}^\top)) + \dim(\mathcal{C}(\mathbf{A})) = m$ . The null space component goes to zero as  $\mathbf{A}\mathbf{x}_n = \mathbf{0} \in \mathbb{R}^m$ . The row space component goes to column space as  $\mathbf{A}\mathbf{x}_r = \mathbf{A}(\mathbf{x}_r + \mathbf{x}_n) = \mathbf{b} \in \mathcal{C}(\mathbf{A})$ .

**Theorem 27.1: (The Fundamental Theorem of Linear Algebra)**

Orthogonal Complement and Rank-Nullity Theorem: for any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , we have

- $\mathcal{N}(\mathbf{A})$  is orthogonal complement to the row space  $\mathcal{C}(\mathbf{A}^\top)$  in  $\mathbb{R}^n$ :  $\dim(\mathcal{N}(\mathbf{A})) + \dim(\mathcal{C}(\mathbf{A}^\top)) = n$ ;
- $\mathcal{N}(\mathbf{A}^\top)$  is orthogonal complement to the column space  $\mathcal{C}(\mathbf{A})$  in  $\mathbb{R}^m$ :  $\dim(\mathcal{N}(\mathbf{A}^\top)) + \dim(\mathcal{C}(\mathbf{A})) = m$ ;
- For rank- $r$  matrix  $\mathbf{A}$ ,  $\dim(\mathcal{C}(\mathbf{A}^\top)) = \dim(\mathcal{C}(\mathbf{A})) = r$ , that is,  $\dim(\mathcal{N}(\mathbf{A})) = n - r$  and  $\dim(\mathcal{N}(\mathbf{A}^\top)) = m - r$ .

**Proof** [of Theorem 27.1] Following from the proof of Lemma 4.2 in Appendix A. Let  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r$  be a set of vectors in  $\mathbb{R}^n$  that form a basis for the row space, then  $\mathbf{A}\mathbf{r}_1, \mathbf{A}\mathbf{r}_2, \dots, \mathbf{A}\mathbf{r}_r$  is a basis for the column space of  $\mathbf{A}$ . Let  $\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k \in \mathbb{R}^n$  form a basis for the null space of  $\mathbf{A}$ . Following again from the proof of Lemma 4.2,  $\mathcal{N}(\mathbf{A}) \perp \mathcal{C}(\mathbf{A}^\top)$ , thus,  $\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r$  are perpendicular to  $\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k$ . Then,  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r, \mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k\}$  is linearly independent in  $\mathbb{R}^n$ .

For any vector  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{Ax}$  is in the column space of  $\mathbf{A}$ . Then it can be expressed by a combination of  $\mathbf{A}\mathbf{r}_1, \mathbf{A}\mathbf{r}_2, \dots, \mathbf{A}\mathbf{r}_r$ :  $\mathbf{Ax} = \sum_{i=1}^r a_i \mathbf{A}\mathbf{r}_i$  which states that  $\mathbf{A}(\mathbf{x} - \sum_{i=1}^r a_i \mathbf{r}_i) = \mathbf{0}$  and  $\mathbf{x} - \sum_{i=1}^r a_i \mathbf{r}_i$  is thus in  $\mathcal{N}(\mathbf{A})$ . Since  $\{\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k\}$  is a basis for the null space of  $\mathbf{A}$ ,  $\mathbf{x} - \sum_{i=1}^r a_i \mathbf{r}_i$  can be expressed by a combination of  $\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k$ :  $\mathbf{x} - \sum_{i=1}^r a_i \mathbf{r}_i = \sum_{j=1}^k b_j \mathbf{n}_j$ , i.e.,  $\mathbf{x} = \sum_{i=1}^r a_i \mathbf{r}_i + \sum_{j=1}^k b_j \mathbf{n}_j$ . That is, any vector  $\mathbf{x} \in \mathbb{R}^n$  can be expressed by  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r, \mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_k\}$  and the set forms a basis for  $\mathbb{R}^n$ . Thus the dimension sum to  $n$ :  $r + k = n$ , i.e.,  $\dim(\mathcal{N}(\mathbf{A})) + \dim(\mathcal{C}(\mathbf{A}^\top)) = n$ . Similarly, we can prove  $\dim(\mathcal{N}(\mathbf{A}^\top)) + \dim(\mathcal{C}(\mathbf{A})) = m$ . ■

Figure 27.1 demonstrates two pairs of such orthogonal subspaces and shows how  $\mathbf{A}$  takes  $\mathbf{x}$  into the column space. The dimensions of the row space of  $\mathbf{A}$  and the null space of  $\mathbf{A}$  add to  $n$ . And the dimensions of the column space of  $\mathbf{A}$  and the null space of  $\mathbf{A}^\top$  add to  $m$ . The null space component goes to zero as  $\mathbf{Ax}_n = \mathbf{0} \in \mathbb{R}^m$  which is the intersection of column space of  $\mathbf{A}$  and null space of  $\mathbf{A}^\top$ . The row space component goes to column space as  $\mathbf{Ax}_r = \mathbf{A}(\mathbf{x}_r + \mathbf{x}_n) = \mathbf{b} \in \mathbb{R}^m$ .

### B.1 Find the Basis of the Four Subspaces via the CR Decomposition

For CR decomposition of matrix  $\mathbf{A} = \mathbf{CR} \in \mathbb{R}^{m \times n}$  (Section 5, p. 155), we have  $\mathbf{R} = [\mathbf{I}_r, \mathbf{F}] \mathbf{P}$ , where  $\mathbf{P}$  is an  $n \times n$  permutation to put the columns of the  $r \times r$  identity matrix  $\mathbf{I}_r$  into the correct positions as shown in Section 5.3. We can thus use the  $r$  linearly independent columns of  $\mathbf{C}$  as the basis for the column space of  $\mathbf{A}$ , i.e., basis for  $\mathcal{C}(\mathbf{A})$ .

Further, let  $\mathbf{N} = \mathbf{P}^\top \begin{bmatrix} -\mathbf{F} \\ \mathbf{I}_{n-r} \end{bmatrix}$ , where  $\mathbf{I}_{n-r}$  is an  $(n-r) \times (n-r)$  identity matrix. Then

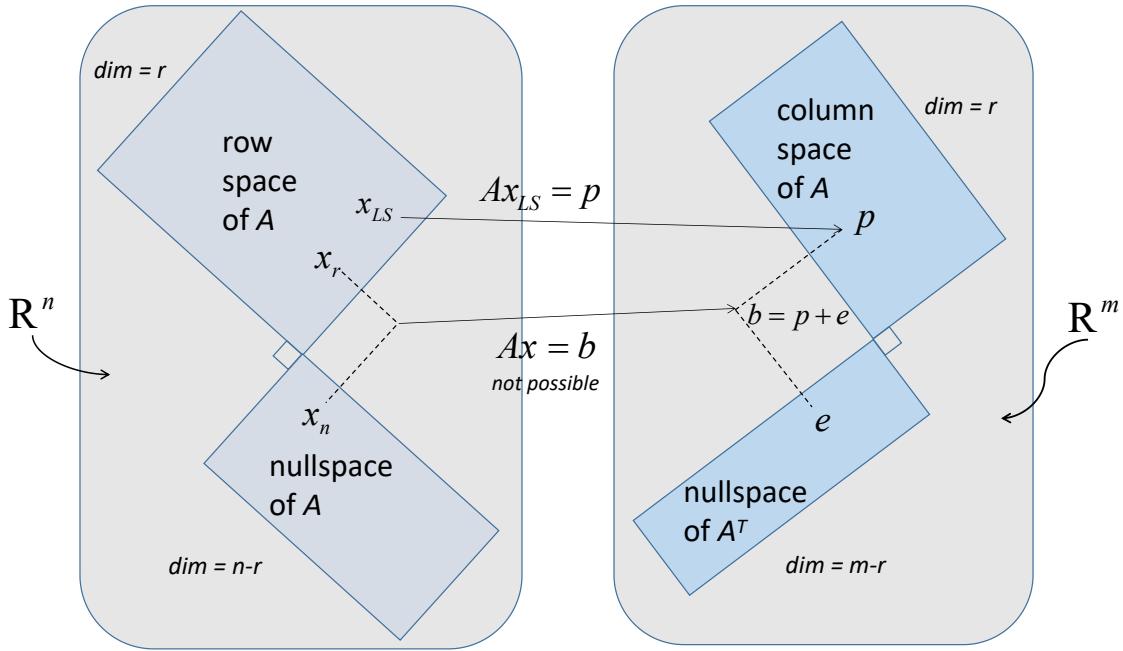
$$\begin{aligned}
\mathbf{A}\mathbf{N} &= [\mathbf{C} \quad \mathbf{C}\mathbf{F}] \mathbf{P}\mathbf{P}^\top \begin{bmatrix} -\mathbf{F} \\ \mathbf{I}_{n-r} \end{bmatrix} \\
&= [\mathbf{C} \quad \mathbf{C}\mathbf{F}] \begin{bmatrix} -\mathbf{F} \\ \mathbf{I}_{n-r} \end{bmatrix} \\
&= -\mathbf{C}\mathbf{F} + \mathbf{C}\mathbf{F} \\
&= \mathbf{0}
\end{aligned}$$

Moreover, the  $n \times (n - r)$  matrix  $\begin{bmatrix} -\mathbf{F} \\ \mathbf{I}_{n-r} \end{bmatrix}$  has  $n - r$  independent columns since  $\mathbf{I}_{n-r}$  is an identity matrix. And multiply from left by a permutation matrix  $\mathbf{P}^\top$  will not change the independent property. Thus,  $\mathbf{N}$  has  $n - r$  linearly independent columns that are in  $\mathcal{N}(\mathbf{A})$ . Furthermore, we have shown that the dimension of  $\mathcal{N}(\mathbf{A})$  is  $n - r$ , thus the columns of  $\mathbf{N}$  form a basis of  $\mathcal{N}(\mathbf{A})$ .

Similarly, from the CR decomposition of  $\mathbf{A}^\top$ , we can find the basis of the row space of  $\mathbf{A}$  and null space of  $\mathbf{A}^\top$ .

In Section 14.3, we further find the orthonormal basis for the four subspaces via SVD. And a more detailed review of this fundamental theorem of linear algebra is provided in (Lu, 2021c) where the authors provide 7 figures from different perspectives to describe the theorem.

## Appendix C. The Fundamental Theorem of Linear Algebra: A Least Squares View



**Figure 27.2:** Least squares: a row space to column space view. Transfer from the row space of  $A$  to the column space of  $A$ . The least squares solution  $x_{LS}$  minimizes the distance of  $\|Ax - b\|^2$ .

The least squares problem is described in Section 3.20.1, Section 4.3.1 and Section 14.7.1 (p. 125, p. 146, and p. 278) via different matrix decompositions. As a recap, let's consider the overdetermined system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the data matrix,  $\mathbf{b} \in \mathbb{R}^m$  with  $m \geq n$  is the observation matrix. Normally  $\mathbf{A}$  will have full column rank since the data from real work has a large chance to be unrelated. And the least squares (LS) solution is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  for minimizing  $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ , where  $\mathbf{A}^\top \mathbf{A}$  is invertible since  $\mathbf{A}$  has full column rank and  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$ .

The solution is to make the error  $\mathbf{b} - \mathbf{A}\mathbf{x}$  as small as possible concerning the mean square error.  $\mathbf{A}\mathbf{x}$  is a combination of the columns of  $\mathbf{A}$ , as a result,  $\mathbf{A}\mathbf{x}$  can never leave the column space of  $\mathbf{A}$ , we should choose the closest point to  $\mathbf{b}$  in the column space (Strang, 1993). This point is the projection  $\mathbf{p}$  of  $\mathbf{b}$ . Then the error vector  $\mathbf{e} = \mathbf{b} - \mathbf{p}$  has minimal length. In another word, the best combination  $\mathbf{p} = \mathbf{A}\mathbf{x}_{LS}$  is the projection of  $\mathbf{b}$  onto the column space. The error  $\mathbf{e}$  is perpendicular to the column space. Therefore  $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{x}_{LS}$  is in the null space of  $\mathbf{A}^\top$ :

$$\mathbf{A}^\top(\mathbf{b} - \mathbf{A}\mathbf{x}_{LS}) = \mathbf{0} \quad \text{or} \quad \mathbf{A}^\top \mathbf{b} = \mathbf{A}^\top \mathbf{A}\mathbf{x}_{LS},$$

which is also known as the **normal equation** of least squares. The relationship between  $\mathbf{e}$  and  $\mathbf{p}$  is shown in Figure 27.2 where  $\mathbf{b}$  is split into  $\mathbf{p} + \mathbf{e}$ . Furthermore, it can be shown that

$\mathbf{x}_{LS}$  is in the row space of  $\mathbf{A}$ , i.e., it cannot be split into a combination of two components that are in row space of  $\mathbf{A}$  and null space of  $\mathbf{A}$  respectively (see  $\mathbf{x}_{LS}$  via the pseudo-inverse of  $\mathbf{A}$  in Section 14.7.1,  $\mathbf{x}_{LS}$  is a linear combination of the orthonormal basis of the row space).

## Appendix D. Projection and Orthogonal Projection

As discussed above, the OLS estimator is to minimize  $\|\mathbf{b} - \mathbf{Ax}\|^2$  which searches for an  $\mathbf{x}_{LS}$  such that  $\mathbf{Ax}_{LS}$  is in  $\mathcal{C}(\mathbf{A})$  to minimize the distance between  $\mathbf{Ax}_{LS}$  and  $\mathbf{b}$ . The nearest point is the projection  $\mathbf{p}$ . The predicted value  $\mathbf{p} = \mathbf{Ax}_{LS}$  is the projection of  $\mathbf{b}$  onto  $\mathcal{C}(\mathbf{A})$  by a projection matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$ :

$$\mathbf{p} = \mathbf{Ax}_{LS} = \mathbf{Hb},$$

where the matrix  $\mathbf{H}$  is also called the hat matrix as it puts a hat on the outputs ( $\mathbf{p}$  is denoted as  $\hat{\mathbf{b}}$  in some texts).

But what is a projection matrix? We could not just say  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  is a projection without any explanation. Before the discussion on the projection matrix, we first provide some properties on symmetric and idempotent matrices that will be extensively used in the next sections.

### D.1 Properties of Symmetric and Idempotent Matrices

Symmetric idempotent matrices have specific eigenvalues which will be often used.

#### Lemma 27.1: (Eigenvalue of Symmetric Idempotent Matrices)

The only possible eigenvalues of any symmetric idempotent matrix are 0 and 1.

In Lemma 27.2, we prove the eigenvalues of idempotent matrices are 1 and 0 as well which relaxes the conditions required here (both idempotent and symmetric). However, the method used in the proof is quite useful so we keep both of the claims. To prove the lemma above, we need to use the result of the spectral theorem.

**Proof** [of Lemma 27.1] Suppose matrix  $\mathbf{A}$  is symmetric idempotent. By spectral theorem (Theorem 13.1, p. 241), we can decompose  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , where  $\mathbf{Q}$  is an orthogonal matrix and  $\Lambda$  is a diagonal matrix. Therefore,

$$\begin{aligned} (\mathbf{Q}\Lambda\mathbf{Q}^\top)^2 &= \mathbf{Q}\Lambda\mathbf{Q} \\ \mathbf{Q}\Lambda\mathbf{Q}^\top\mathbf{Q}\Lambda\mathbf{Q}^\top &= \mathbf{Q}\Lambda\mathbf{Q}^\top \\ \mathbf{Q}\Lambda^2\mathbf{Q}^\top &= \mathbf{Q}\Lambda\mathbf{Q}^\top \\ \Lambda^2 &= \Lambda \\ \lambda_i^2 &= \lambda_i. \end{aligned}$$

Thus the eigenvalues of  $\mathbf{A}$  satisfies that  $\lambda_i \in \{0, 1\}$ ,  $\forall i$ . We complete the proof. ■

In the above lemma, we use the spectral theorem to prove the only eigenvalues of any symmetric idempotent matrices are 1 and 0. This trick from spectral theorem is often used in mathematical proofs (see distribution theory in (Lu, 2021d)). By trivial trick, we can relax the condition from symmetric idempotent to idempotent.

**Lemma 27.2: (Eigenvalue of Idempotent Matrices)**

The only possible eigenvalues of any idempotent matrix are 0 and 1.

**Proof** [of Lemma 27.2] Let  $\mathbf{x}$  be an eigenvector of the idempotent matrix  $\mathbf{A}$  corresponding to eigenvalue  $\lambda$ . That is

$$\mathbf{A}\mathbf{x} = \lambda\mathbf{x}.$$

Also, we have

$$\begin{aligned}\mathbf{A}^2\mathbf{x} &= (\mathbf{A}^2)\mathbf{x} = \mathbf{A}\mathbf{x} = \lambda\mathbf{x} \\ &= \mathbf{A}(\lambda\mathbf{x}) = \mathbf{A}(\lambda\mathbf{x}) = \lambda\mathbf{A}\mathbf{x} = \lambda^2\mathbf{x},\end{aligned}$$

which implies  $\lambda^2 = \lambda$  and  $\lambda$  is 0 or 1. ■

We also prove the rank of a symmetric idempotent matrix is equal to its trace which will be extremely useful in the next sections.

**Lemma 27.3: (Rank and Trace of Symmetric Idempotent Matrices)**

For any  $n \times n$  symmetric idempotent matrix  $\mathbf{A}$ , the rank of  $\mathbf{A}$  equals the trace of  $\mathbf{A}$ .

**Proof** [of Lemma 27.3] From spectral theorem 13.1, we have spectral decomposition for  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . Since  $\mathbf{A}$  and  $\Lambda$  are similar matrices, their rank and trace are the same (Lemma 8.2, p. 198). That is,

$$\begin{aligned}rank(\mathbf{A}) &= rank(\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)), \\ trace(\mathbf{A}) &= trace(\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)),\end{aligned}$$

By Lemma 27.1, the only eigenvalues of  $\mathbf{A}$  are 0 and 1. Then,  $rank(\mathbf{A}) = trace(\mathbf{A})$ . ■

In the above lemma, we prove the rank and trace of any symmetric idempotent matrix are the same. However, it is again rather a loose condition. We here also prove that the only condition on idempotency has the same result. Again, although this lemma is a more general version, we provide both of them since the method used in the proof is quite useful in the sequel.

**Lemma 27.4: (Rank and Trace of an Idempotent Matrix)**

For any  $n \times n$  idempotent matrix  $\mathbf{A}$ , the rank of  $\mathbf{A}$  equals the trace of  $\mathbf{A}$ .

**Proof** [of Lemma 27.4] Any  $n \times n$  rank- $r$  matrix  $\mathbf{A}$  has CR decomposition  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , where  $\mathbf{C} \in \mathbb{R}^{n \times r}$  and  $\mathbf{R} \in \mathbb{R}^{r \times n}$  with  $\mathbf{C}, \mathbf{R}$  having full rank  $r$ . Then,

$$\begin{aligned} \mathbf{A}^2 &= \mathbf{A} \\ \mathbf{C}\mathbf{R}\mathbf{C}\mathbf{R} &= \mathbf{C}\mathbf{R} \\ \mathbf{R}\mathbf{C}\mathbf{R} &= \mathbf{R} \\ \mathbf{R}\mathbf{C} &= \mathbf{I}_r, \end{aligned}$$

where  $\mathbf{I}_r$  is an  $r \times r$  identity matrix. Thus

$$\text{trace}(\mathbf{A}) = \text{trace}(\mathbf{C}\mathbf{R}) = \text{trace}(\mathbf{R}\mathbf{C}) = \text{trace}(\mathbf{I}_r) = r,$$

which equals the rank of  $\mathbf{A}$ . ■

## D.2 Orthogonal Projection and Geometric Interpretation for LS

Formally, we define the projection matrix as follows:

### Definition 27.5: Projection Matrix

A matrix  $\mathbf{H}$  is called a projection matrix onto subspace  $\mathcal{V} \in \mathbb{R}^m$  if and only if  $\mathbf{H}$  satisfies the following properties

- (P1)  $\mathbf{H}\mathbf{b} \in \mathcal{V}$  for all  $\mathbf{b} \in \mathbb{R}^m$ : any vector can be projected onto subspace  $\mathcal{V}$ ;
- (P2)  $\mathbf{H}\mathbf{b} = \mathbf{b}$  for all  $\mathbf{b} \in \mathcal{V}$ : projecting a vector already in that subspace has no effect;
- (P3)  $\mathbf{H}^2 = \mathbf{H}$ , i.e., projecting twice is equal to projecting once because we are already in that subspace, i.e.,  $\mathbf{H}$  is idempotent.

Since we project a vector in  $\mathbb{R}^m$  onto the subspace of  $\mathbb{R}^m$ , so any projection matrix is a square matrix. Otherwise, we will project onto the subspace of  $\mathbb{R}^k$  rather than  $\mathbb{R}^m$ . We realize that  $\mathbf{H}\mathbf{b}$  is always in the column space of  $\mathbf{H}$ , and we would wonder about the relationship between  $\mathcal{V}$  and  $\mathcal{C}(\mathbf{H})$ . And actually, the column space of  $\mathbf{H}$  is equal to the subspace  $\mathcal{V}$  we want to project onto. Suppose  $\mathcal{V} = \mathcal{C}(\mathbf{H})$  and suppose further that  $\mathbf{b}$  is already in the subspace  $\mathcal{V} = \mathcal{C}(\mathbf{H})$ , i.e., there is a vector  $\boldsymbol{\alpha}$  such that  $\mathbf{b} = \mathbf{H}\boldsymbol{\alpha}$ . Given the only condition (P3) above, we have,

$$\mathbf{H}\mathbf{b} = \mathbf{H}\mathbf{H}\boldsymbol{\alpha} = \mathbf{H}\boldsymbol{\alpha} = \mathbf{b}.$$

That is, the condition (P3) implies conditions (P1), (P2).

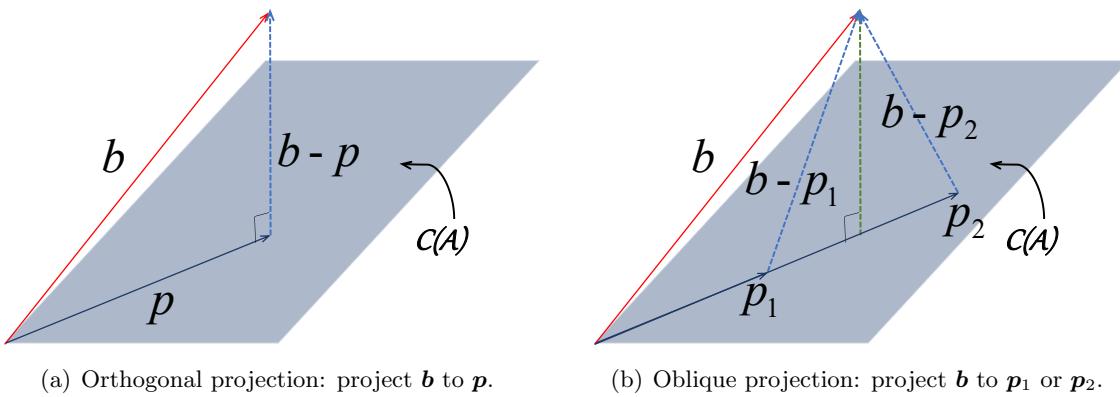
Intuitively, we also want the projection  $\mathbf{p} = \mathbf{H}\mathbf{b}$  of any vector  $\mathbf{b}$  to be perpendicular to  $\mathbf{b} - \mathbf{p}$  such that the distance between  $\mathbf{p}$  and  $\mathbf{b}$  is minimal and agrees with our least squared error requirement. This is called the **orthogonal projection**.

**Definition 27.6: Orthogonal Projection Matrix**

A matrix  $\mathbf{H}$  is called an orthogonal projection matrix onto subspace  $\mathcal{V} \in \mathbb{R}^m$  if and only if  $\mathbf{H}$  is a projection matrix, and the projection  $\mathbf{p}$  of any vector  $\mathbf{b} \in \mathbb{R}^m$  is perpendicular to  $\mathbf{b} - \mathbf{p}$ , i.e.,  $\mathbf{H}$  projects onto  $\mathcal{V}$  and along  $\mathcal{V}^\perp$ .

**Remark 27.7: Orthogonal Projections vs Orthogonal Matrices**

Note here the *orthogonal projection matrix* does not mean the projection matrix is an orthogonal matrix (discussed in Section 3.5, p. 87), but the projection  $\mathbf{p}$  is perpendicular to  $\mathbf{b} - \mathbf{p}$ . This orthogonal projection matrix is so special that we will take a projection implicitly as an orthogonal projection if not particularly clarified in the sequel. Sometimes, to avoid confusion, one may use the term *oblique projection matrix* in the nonorthogonal case where the difference is shown in Figure 27.3.

**Figure 27.3:** Projection onto the hyperplane of  $C(\mathcal{A})$ .**Lemma 27.8: (Symmetric Orthogonal Projection Matrix)**

Any projection matrix  $\mathbf{H}$  is an orthogonal projection matrix if and only if  $\mathbf{H}$  is symmetric.

**Proof** [of Lemma 27.8] We will prove by forward implication and backward implication separately as follows.

**Forward implication** Suppose  $\mathbf{H}$  is an orthogonal projection matrix  $\mathbf{H}$  which projects vectors onto subspace  $\mathcal{V}$ . Then vectors  $\mathbf{v}$  and  $\mathbf{w}$  can be decomposed into a vector lies in  $\mathcal{V}$  ( $\mathbf{v}_p$  and  $\mathbf{w}_p$ ) and a vector lies in  $\mathcal{V}^\perp$  ( $\mathbf{v}_n$  and  $\mathbf{w}_n$ ) such that

$$\begin{aligned}\mathbf{v} &= \mathbf{v}_p + \mathbf{v}_n \\ \mathbf{w} &= \mathbf{w}_p + \mathbf{w}_n.\end{aligned}$$

Since projection matrix  $\mathbf{H}$  will project vectors onto  $\mathcal{V}$ , then  $\mathbf{H}\mathbf{v} = \mathbf{v}_p$  and  $\mathbf{H}\mathbf{w} = \mathbf{w}_p$ . We then have

$$\begin{aligned} (\mathbf{H}\mathbf{v})^\top \mathbf{w} &= \mathbf{v}_p^\top \mathbf{w} & \mathbf{v}^\top (\mathbf{H}\mathbf{w}) &= \mathbf{v}^\top \mathbf{w}_p \\ &= \mathbf{v}_p^\top (\mathbf{w}_p + \mathbf{w}_n) & &= (\mathbf{v}_p + \mathbf{v}_n)^\top \mathbf{w}_p \\ &= \mathbf{v}_p^\top \mathbf{w}_p + \mathbf{v}_p^\top \mathbf{w}_n & &= \mathbf{v}_p^\top \mathbf{w}_p + \mathbf{v}_n^\top \mathbf{w}_p \\ &= \mathbf{v}_p^\top \mathbf{w}_p & &= \mathbf{v}_p^\top \mathbf{w}_p, \end{aligned}$$

where the last equation is from the fact that  $\mathbf{v}_p$  is perpendicular to  $\mathbf{w}_n$ , and  $\mathbf{v}_n$  is perpendicular to  $\mathbf{w}_p$ . Thus we have

$$(\mathbf{H}\mathbf{v})^\top \mathbf{w} = \mathbf{v}^\top (\mathbf{H}\mathbf{w}) \quad \text{leads to} \quad \mathbf{v}^\top \mathbf{H}^\top \mathbf{w} = \mathbf{v}^\top \mathbf{H}\mathbf{w},$$

which implies  $\mathbf{H}^\top = \mathbf{H}$ .

**Backward implication** For the reverse, if a projection matrix  $\mathbf{H}$  (not necessarily an orthogonal projection) is symmetric, then any vector  $\mathbf{v}$  can be decomposed into  $\mathbf{v} = \mathbf{H}\mathbf{v} + (\mathbf{I} - \mathbf{H})\mathbf{v}$ . If we can prove  $\mathbf{H}\mathbf{v}$  is perpendicular to  $(\mathbf{I} - \mathbf{H})\mathbf{v}$ , then we complete the proof. To see this, we have

$$\begin{aligned} (\mathbf{H}\mathbf{v})^\top (\mathbf{I} - \mathbf{H})\mathbf{v} &= \mathbf{v}^\top \mathbf{H}^\top (\mathbf{I} - \mathbf{H})\mathbf{v} \\ &= \mathbf{v}^\top (\mathbf{H}^\top - \mathbf{H}^\top \mathbf{H})\mathbf{v} \\ &= \mathbf{v}^\top (\mathbf{H} - \mathbf{H}\mathbf{H})\mathbf{v} \\ &= \mathbf{v}^\top (\mathbf{H} - \mathbf{H})\mathbf{v} = \mathbf{0}, \end{aligned}$$

which completes the proof. ■

We claimed that the orthogonal projection has minimal length, i.e., the distance between  $\mathbf{b}$  and  $\mathbf{p}$  is minimal. Here we prove this property rigorously.

### Lemma 27.9: (Minimal Distance in Orthogonal Projection)

Let  $\mathcal{V}$  be a subspace of  $\mathbb{R}^m$  and  $\mathbf{H}$  be an orthogonal projection onto  $\mathcal{V}$ . Then

$$\|\mathbf{b} - \mathbf{H}\mathbf{b}\|^2 \leq \|\mathbf{b} - \mathbf{v}\|^2, \quad \forall \mathbf{v} \in \mathcal{V}.$$

**Proof** [of Lemma 27.9] Let  $\mathbf{H} = \mathbf{Q}\Lambda\mathbf{Q}^\top \in \mathbb{R}^{m \times m}$  be the spectral decomposition of orthogonal projection  $\mathbf{H}$ ,  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m]$  be the column partition of  $\mathbf{Q}$  and  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ . Let  $\dim(\mathcal{V}) = r$ . Then from Lemma 27.1, the only possible eigenvalues of orthogonal projection matrix are 1 and 0. Without loss of generality, let  $\lambda_1 = \lambda_2 = \dots = \lambda_r = 1$  and  $\lambda_{r+1} = \lambda_{r+2} = \dots = \lambda_m = 0$ . Then

- $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_m\}$  is an orthonormal basis of  $\mathbb{R}^m$ ;

- $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_r\}$  is an orthonormal basis of  $\mathcal{V}$ . So for any vector  $\mathbf{v} \in \mathcal{V}$ , we have  $\mathbf{v}^\top \mathbf{q}_i = 0$  for  $i \in \{r+1, r+2, \dots, m\}$ . Then we have,

$$\begin{aligned}
\|\mathbf{b} - \mathbf{Hb}\|^2 &= \|\mathbf{Q}^\top \mathbf{b} - \mathbf{Q}^\top \mathbf{Hb}\|^2 && \text{(invariant under orthogonal)} \\
&= \sum_{i=1}^m (\mathbf{b}^\top \mathbf{q}_i - (\mathbf{Hb})^\top \mathbf{q}_i)^2 \\
&= \sum_{i=1}^m (\mathbf{b}^\top \mathbf{q}_i - \mathbf{b}^\top \mathbf{Hq}_i)^2 && (\mathbf{H} \text{ is symmetric}) \\
&= \sum_{i=1}^m (\mathbf{b}^\top \mathbf{q}_i - \lambda_i \mathbf{b}^\top \mathbf{q}_i)^2 && (\mathbf{HQ} = \mathbf{Q}\Lambda) \\
&= 0 + \sum_{i=r+1}^m (\mathbf{b}^\top \mathbf{q}_i)^2 && \text{(eigenvalues are 1 or 0)} \\
&\leq \sum_{i=1}^r (\mathbf{b}^\top \mathbf{q}_i - \mathbf{v}^\top \mathbf{q}_i)^2 + \sum_{i=r+1}^m (\mathbf{b}^\top \mathbf{q}_i)^2 \\
&= \|\mathbf{Q}^\top \mathbf{b} - \mathbf{Q}^\top \mathbf{v}\|^2 && (\mathbf{v}^\top \mathbf{q}_i = 0 \text{ for } i > r) \\
&= \|\mathbf{b} - \mathbf{v}\|^2,
\end{aligned}$$

which completes the proof. ■

### Lemma 27.10: (Angle between the Original and Projected Vectors)

Let  $\mathbf{H}$  be an orthogonal projection onto  $\mathcal{V}$ . Then

1.  $\mathbf{b}^\top (\mathbf{Hb}) \geq 0$ , i.e., angle between  $\mathbf{b}$  and  $\mathbf{Hb}$  is smaller than  $90^\circ$ ;
2.  $\|\mathbf{Hb}\|^2 \leq \|\mathbf{b}\|^2$ .

**Proof** [of Lemma 27.10] By definition of orthogonal projection, we have  $\mathbf{b}^\top (\mathbf{Hb}) = \mathbf{b}^\top \mathbf{H}(\mathbf{Hb}) = \mathbf{b}^\top \mathbf{H}^\top (\mathbf{Hb}) = \|\mathbf{Hb}\|^2 \geq 0$ . And we could decompose  $\mathbf{b}$  by

$$\begin{aligned}
\|\mathbf{b}\|^2 &= \|(\mathbf{I} - \mathbf{H} + \mathbf{H})\mathbf{b}\|^2 = \|(\mathbf{I} - \mathbf{H})\mathbf{b}\|^2 + \|\mathbf{Hb}\|^2 + 2\mathbf{b}^\top (\mathbf{I} - \mathbf{H})^\top \mathbf{Hb} \\
&= \|(\mathbf{I} - \mathbf{H})\mathbf{b}\|^2 + \|\mathbf{Hb}\|^2 \geq \|\mathbf{Hb}\|^2.
\end{aligned}$$

This completes the proof. ■

To conclude, on the origin of the projection and orthogonal projection matrix, we define the projection matrix to be idempotent, and it is symmetric when restricted to the orthogonal projection. From this orthogonal projection, we prove the distance between the original vector and the projected vector is minimal.

**Proposition 27.11: (Projection Matrix from a Set of Vectors)**

If  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$  are linearly independent and are such that  $\mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]) = \mathcal{V}$ , then the orthogonal projection onto  $\mathcal{V}$  can be represented as

$$\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top,$$

where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the matrix with columns  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n$ .

**Proof** [of Proposition 27.11] It can be easily verified  $\mathbf{H}$  is symmetric and idempotent. By SVD of  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , we have  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top = \mathbf{U}\Sigma(\Sigma^\top \Sigma)^{-1} \Sigma^\top \mathbf{U}^\top$ . Let  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m]$  be the column partition of  $\mathbf{U}$ . From Lemma 14.1,  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A})$ . And  $\Sigma(\Sigma^\top \Sigma)^{-1} \Sigma^\top$  in  $\mathbf{H}$  is an  $m \times m$  matrix with the upper-left part being a  $n \times n$  identity matrix and the other parts are zero. Apply this observation of  $\mathbf{H}$  into spectral theorem,  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_n\}$  is also an orthonormal basis of  $\mathcal{C}(\mathbf{H})$ . Thus  $\mathcal{C}(\mathbf{H}) = \mathcal{C}(\mathbf{A})$ , and the orthogonal projection  $\mathbf{H}$  is projecting onto  $\mathcal{C}(\mathbf{A})$  which completes the proof. ■

**Projection In Matrix Decomposition** : Following the above orthogonal projection  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$ . Let  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  and  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$  be the QR and SVD decomposition of  $\mathbf{A}$  respectively. Then the orthogonal projection can also be denoted as  $\mathbf{H} = \mathbf{Q}\mathbf{Q}^\top = \mathbf{U}\mathbf{U}^\top$ .

The proposition above brings us back to the result we have shown at the beginning of this section. For LS estimator to minimize  $\|\mathbf{b} - \mathbf{Ax}\|^2$  which searches for an  $\mathbf{x}_{LS}$  so that  $\mathbf{p} = \mathbf{Ax}_{LS}$  is in  $\mathcal{C}(\mathbf{A})$  to minimize the distance between  $\mathbf{Ax}_{LS}$  and  $\mathbf{b}$ . An orthogonal projection matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  can project  $\mathbf{b}$  onto the column space of  $\mathbf{A}$  and the projected vector is  $\mathbf{p} = \mathbf{Hb}$  with the squared distance between  $\mathbf{p}$  and  $\mathbf{b}$  being minimal (by Lemma 27.9).

To repeat, the hat matrix  $\mathbf{H}$  has a geometric interpretation.  $\mathbf{H}$  drops a perpendicular to the hyperplane. Here,  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  drops  $\mathbf{b}$  onto the column space of  $\mathbf{A}$ :  $\mathbf{p} = \mathbf{Hb}$ . Idempotency also has a geometric interpretation. Additional  $\mathbf{H}$ 's also drop a perpendicular to the hyperplane. But it has no additional effect because we are already on that hyperplane. Therefore  $\mathbf{H}^2\mathbf{b} = \mathbf{Hb}$ . This situation is shown in Figure 27.3(a). The sum of squared error is then equal to the squared Euclidean distance between  $\mathbf{b}$  and  $\mathbf{p}$ . Thus the least squares solution for  $\mathbf{x}$  corresponds to the orthogonal projection of  $\mathbf{b}$  onto the column space of  $\mathbf{A}$ .

**Lemma 27.12: (Column Space of Projection Matrix)**

We notice that the hat matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  is to project any vector in  $\mathbb{R}^m$  onto the column space of  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . That is,  $\mathbf{Hb} \in \mathcal{C}(\mathbf{A})$ . Notice again  $\mathbf{Hb}$  is nothing but a combination of the columns of  $\mathbf{H}$ , thus  $\mathcal{C}(\mathbf{H}) = \mathcal{C}(\mathbf{A})$ .

In general, for any projection matrix  $\mathbf{H}$  to project vector onto subspace  $\mathcal{V}$ , then  $\mathcal{C}(\mathbf{H}) = \mathcal{V}$ .

The proof is trivial from the proof of Proposition 27.11, here we do it in another way.

**Proof** [of Lemma 27.12] Since  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top = \mathbf{AC}$ , the columns of  $\mathbf{H}$  are combinations of the columns of  $\mathbf{A}$ . Thus  $\mathcal{C}(\mathbf{H}) \subseteq \mathcal{C}(\mathbf{A})$ . By Lemma 27.4, we have

$$\text{rank}(\mathbf{H}) = \text{trace}(\mathbf{H}) = \text{trace}(\mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top) = \text{trace}((\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{A}) = \text{trace}(\mathbf{I}_n) = n.$$

where the third equality is from the fact that the trace of a product is invariant under cyclical permutations of the factors:  $\text{trace}(\mathbf{XYZ}) = \text{trace}(\mathbf{YZX}) = \text{trace}(\mathbf{ZXY})$ . Thus, the rank of  $\mathbf{H}$  equals the rank of  $\mathbf{A}$  such that  $\mathcal{C}(\mathbf{H}) = \mathcal{C}(\mathbf{A})$ . ■

### D.3 Properties of Orthogonal Projection Matrices

In fact,  $(\mathbf{I} - \mathbf{H})$  is also symmetric idempotent, and actually, when  $\mathbf{H}$  projects onto a subspace  $\mathcal{V}$ ,  $\mathbf{I} - \mathbf{H}$  projects onto the perpendicular subspace  $\mathcal{V}^\perp$ .

#### Proposition 27.13: (Project onto $\mathcal{V}^\perp$ )

Let  $\mathcal{V}$  be a subspace and  $\mathbf{H}$  be an orthogonal projection onto  $\mathcal{V}$ . Then  $\mathbf{I} - \mathbf{H}$  is the orthogonal projection matrix onto  $\mathcal{V}^\perp$ .

The claim can be extended further that suppose  $\mathcal{V}_1 \subseteq \mathcal{V}$  and  $\mathcal{V}_2 \subseteq \mathcal{V}^\perp$ . Then  $\mathbf{H}_1$  is the orthogonal projection that projects onto  $\mathcal{V}_1$  and  $\mathbf{H}_2$  is the orthogonal projection that projects onto  $\mathcal{V}_2$  if and only if  $\mathbf{H}_1\mathbf{H}_2 = \mathbf{0}$ .

**Proof** [of Proposition 27.13] First,  $(\mathbf{I} - \mathbf{H})$  is symmetric,  $(\mathbf{I} - \mathbf{H})^\top = \mathbf{I} - \mathbf{H}^\top = \mathbf{I} - \mathbf{H}$  since  $\mathbf{H}$  is symmatrix. And

$$(\mathbf{I} - \mathbf{H})^2 = \mathbf{I}^2 - \mathbf{IH} - \mathbf{HI} + \mathbf{H}^2 = \mathbf{I} - \mathbf{H}.$$

Thus  $\mathbf{I} - \mathbf{H}$  is an orthogonal projection matrix. By spectral theorem again, let  $\mathbf{H} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . Then  $\mathbf{I} - \mathbf{H} = \mathbf{QQ}^\top - \mathbf{Q}\Lambda\mathbf{Q}^\top = \mathbf{Q}(\mathbf{I} - \Lambda)\mathbf{Q}^\top$ . Hence the column space of  $\mathbf{I} - \mathbf{H}$  is spanned by the eigenvectors of  $\mathbf{H}$  corresponding to the zero eigenvalues of  $\mathbf{H}$  (by Proposition 27.1), which coincides with  $\mathcal{V}^\perp$ .

For the second part, since  $\mathcal{C}(\mathbf{H}_1) = \mathcal{V}_1$  and  $\mathcal{C}(\mathbf{H}_2) = \mathcal{V}_2$ , every column of  $\mathbf{H}_1$  is perpendicular to columns of  $\mathbf{H}_2$ . Thus  $\mathbf{H}_1\mathbf{H}_2 = \mathbf{0}$ . For the reverse, if  $\mathbf{H}_1\mathbf{H}_2 = \mathbf{0}$ , then  $\mathbf{H}_1(\mathbf{H}_2\mathbf{b}) = 0$  for all  $\mathbf{b}$ . Thus  $\mathcal{V}_1 \perp \mathcal{V}_2$ . Moreover, it can be easily verified when  $\mathcal{V}_1 = \mathcal{V}$  and  $\mathcal{V}_2 = \mathcal{V}^\perp$ , then  $\mathbf{H}(\mathbf{I} - \mathbf{H}) = \mathbf{0}$ . ■

A projection matrix that can project any vector onto a subspace is not unique. However, if restricted to the orthogonal projection, then the projection is unique.

#### Proposition 27.14: (Uniqueness of Orthogonal Projection)

If  $\mathbf{P}$  and  $\mathbf{H}$  are orthogonal projection matrices onto the same subspace  $\mathcal{V}$ , then  $\mathbf{P} = \mathbf{H}$ .

**Proof** [of Proposition 27.14] For any vector  $\mathbf{v}$  in  $\mathbb{R}^m$ , it can be split into a vector  $\mathbf{v}_p$  in  $\mathcal{V}$  and a vector  $\mathbf{v}_n$  in  $\mathcal{V}^\perp$  such that  $\mathbf{v} = \mathbf{v}_p + \mathbf{v}_n$  and  $\mathbf{v}_p^\top \mathbf{v}_n = 0$ . Then

$$\mathbf{P}\mathbf{v} = \mathbf{v}_p = \mathbf{H}\mathbf{v},$$

such that  $(\mathbf{P} - \mathbf{H})\mathbf{v} = \mathbf{0}$ . Since any vector  $\mathbf{v} \in \mathbb{R}^m$  is in the null space of  $\mathbf{P} - \mathbf{H}$ , then  $\mathbf{P} - \mathbf{H}$  is of rank 0 and  $\mathbf{P} = \mathbf{H}$ . ■

### Proposition 27.15: (Nested Projection)

Let  $\mathcal{V}_1 \subseteq \mathcal{V}_2 \subseteq \mathbb{R}^m$  be two nested linear subspaces. If  $\mathbf{H}_1$  is the orthogonal projection onto  $\mathcal{V}_1$ , and  $\mathbf{H}_2$  is the orthogonal projection onto  $\mathcal{V}_2$ , then

1.  $\mathbf{H}_2\mathbf{H}_1 = \mathbf{H}_1 = \mathbf{H}_1\mathbf{H}_2$ ;
2.  $\mathbf{H}_2 - \mathbf{H}_1$  is also an orthogonal projection.

**Proof** [of Proposition 27.15] For all  $\mathbf{b} \in \mathbb{R}^m$ , we have  $\mathbf{H}_1\mathbf{b} \in \mathcal{V}_1$ . This implies  $\mathbf{H}_1\mathbf{b} \in \mathcal{V}_1 \subseteq \mathcal{V}_2$ . Thus

$$\mathbf{H}_2(\mathbf{H}_1\mathbf{b}) = \mathbf{H}_1\mathbf{b}. \quad (\text{from Definition 27.5, p. 434})$$

Then  $(\mathbf{H}_2\mathbf{H}_1 - \mathbf{H}_1)\mathbf{b} = \mathbf{0}$  for all  $\mathbf{b} \in \mathbb{R}^m$ . That is, the dimension of the null space  $\mathcal{N}(\mathbf{H}_2\mathbf{H}_1 - \mathbf{H}_1) = n$  and the rank of  $\mathbf{H}_2\mathbf{H}_1 - \mathbf{H}_1$  is 0 which results in  $\mathbf{H}_2\mathbf{H}_1 = \mathbf{H}_1$ .

For  $\mathbf{H}_1\mathbf{H}_2$ , both  $\mathbf{H}_1$  and  $\mathbf{H}_2$  are symmetric such that  $\mathbf{H}_1\mathbf{H}_2 = \mathbf{H}_1^\top \mathbf{H}_2^\top = (\mathbf{H}_2\mathbf{H}_1)^\top = \mathbf{H}_1^\top = \mathbf{H}_1$  which completes the proof of part 1.

To see the second part, we notice that  $(\mathbf{H}_2 - \mathbf{H}_1)^\top = \mathbf{H}_2 - \mathbf{H}_1$  and

$$\begin{aligned} (\mathbf{H}_2 - \mathbf{H}_1)^2 &= \mathbf{H}_2^2 - \mathbf{H}_2\mathbf{H}_1 - \mathbf{H}_1\mathbf{H}_2 + \mathbf{H}_1^2 \\ &= \mathbf{H}_2 - \mathbf{H}_1 - \mathbf{H}_1 + \mathbf{H}_1 \\ &= \mathbf{H}_2 - \mathbf{H}_1, \end{aligned}$$

which states that  $\mathbf{H}_2 - \mathbf{H}_1$  is both symmetric and idempotent. This completes the proof. ■

From the lemma above, we can also claim that orthogonal projection matrices are positive semi-definite (PSD).

### Proposition 27.16: (Symmetric Projection)

Any orthogonal projection matrix  $\mathbf{H}$  is positive semi-definite.

**Proof** [of Proposition 27.16] Since  $\mathbf{H}$  is symmetric and idempotent. For any vector  $\mathbf{x}$ , we have

$$\mathbf{x}^\top \mathbf{H}\mathbf{x} = \mathbf{x}^\top \mathbf{H}\mathbf{H}\mathbf{x} = \mathbf{x}^\top \mathbf{H}^\top \mathbf{H}\mathbf{x} = \|\mathbf{H}\mathbf{x}\| \geq 0.$$

Thus,  $\mathbf{H}$  is PSD. ■

As a recap, we summarize important facts about the orthogonal projection matrix that is often used in the following remark.

**Remark 27.17: Important Facts About Hat Matrix (Part 1)**

1. As we assumed  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is fixed and has full rank with  $n < m$ . It is known that the rank of  $\mathbf{A}$  is equal to the rank of its Gram matrix, defined as  $\mathbf{A}^\top \mathbf{A}$ , such that

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^\top \mathbf{A});$$

2. The rank of an orthogonal projection matrix is the dimension of the subspace onto which it projects. Hence, the rank of  $\mathbf{H}$  is  $n$  when  $\mathbf{A}$  has full rank and  $m < n$ :

$$\text{rank}(\mathbf{H}) = \text{rank}(\mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top) = \text{rank}((\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{I}_n) = n.$$

3. The column space of  $\mathbf{H}$  is identical to the column space of  $\mathbf{A}$ ;

#### D.4 Distance Between Subspaces

Suppose  $\mathcal{V}_1$  and  $\mathcal{V}_2$  are subspaces of  $\mathbb{R}^n$  and  $\dim(\mathcal{V}_1) = \dim(\mathcal{V}_2)$ . We define the *distance* between the two spaces by

$$\text{dist}(\mathcal{V}_1, \mathcal{V}_2) = \|\mathbf{H}_1 - \mathbf{H}_2\|_2,$$

where  $\mathbf{H}_1$  is the orthogonal projection onto  $\mathcal{V}_1$ , and  $\mathbf{H}_2$  is the orthogonal projection onto  $\mathcal{V}_2$ .

**Lemma 27.18: (Subspace Distance)**

Suppose  $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times n}$  are orthogonal matrices, and the column partitions

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \\ n \times r & n \times (n-r) \end{bmatrix}, \quad \mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 & \mathbf{Y}_2 \\ n \times r & n \times (n-r) \end{bmatrix},$$

where  $\mathbf{X}_r, \mathbf{Y}_r$  are the first  $r$  columns of  $\mathbf{X}$  and  $\mathbf{Y}$ . Suppose further that  $\mathcal{V}_1 = \mathcal{C}(\mathbf{X}_1)$  and  $\mathcal{V}_2 = \mathcal{C}(\mathbf{Y}_1)$ , then

$$\text{dist}(\mathcal{V}_1, \mathcal{V}_2) = \|\mathbf{X}_1^\top \mathbf{Y}_2\|_2 = \|\mathbf{Y}_1^\top \mathbf{X}_2\|_2.$$

**Proof** [of Lemma 27.18] It can be easily verified  $\mathbf{X}_1\mathbf{X}_1^\top$  and  $\mathbf{Y}_1\mathbf{Y}_1^\top$  are the orthogonal projections onto  $\mathcal{V}_1, \mathcal{V}_2$ . Write out the equation

$$\begin{aligned}\text{dist}(\mathcal{V}_1, \mathcal{V}_2) &= \|\mathbf{X}_1\mathbf{X}_1^\top - \mathbf{Y}_1\mathbf{Y}_1^\top\|_2 \\ &= \|\mathbf{X}^\top(\mathbf{X}_1\mathbf{X}_1^\top - \mathbf{Y}_1\mathbf{Y}_1^\top)\mathbf{Y}\|_2 \quad (\text{invariant under orthogonal}) \\ &= \left\| \begin{bmatrix} \mathbf{X}_1^\top \\ \mathbf{X}_2^\top \end{bmatrix} (\mathbf{X}_1\mathbf{X}_1^\top - \mathbf{Y}_1\mathbf{Y}_1^\top)[\mathbf{Y}_1, \mathbf{Y}_2] \right\|_2 \\ &= \left\| \begin{bmatrix} \mathbf{I}_r \\ \mathbf{0} \end{bmatrix} \mathbf{X}_1^\top[\mathbf{Y}_1, \mathbf{Y}_2] - \begin{bmatrix} \mathbf{X}_1^\top \\ \mathbf{X}_2^\top \end{bmatrix} \mathbf{Y}_1[\mathbf{I}_r, \mathbf{0}] \right\|_2, \quad (\mathbf{X}_2^\top\mathbf{X}_1 = \mathbf{0}, \mathbf{Y}_1^\top\mathbf{Y}_2 = \mathbf{0}) \\ &= \left\| \begin{bmatrix} \mathbf{0} & \mathbf{X}_1^\top\mathbf{Y}_2 \\ -\mathbf{X}_2^\top\mathbf{Y}_1 & \mathbf{0} \end{bmatrix} \right\|_2.\end{aligned}$$

We note that  $\mathbf{X}_1^\top\mathbf{Y}_2$  and  $\mathbf{X}_2^\top\mathbf{Y}_1$  are submatrices of the following orthogonal matrix

$$\mathbf{Q} = \mathbf{X}^\top\mathbf{Y} = \begin{bmatrix} \mathbf{Q}_{12} & \mathbf{Q}_{12} \\ \mathbf{Q}_{21} & \mathbf{Q}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_1^\top\mathbf{Y}_1 & \mathbf{X}_1^\top\mathbf{Y}_2 \\ \mathbf{X}_2^\top\mathbf{Y}_1 & \mathbf{X}_2^\top\mathbf{Y}_2 \end{bmatrix}.$$

For any unit vector  $\mathbf{x} \in \mathbb{R}^r$  ( $\|\mathbf{x}\| = 1$ ), it follows that

$$\mathbf{Q} \begin{bmatrix} \mathbf{x} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{Q}_{11}\mathbf{x} \\ \mathbf{Q}_{21}\mathbf{x} \end{bmatrix}.$$

By invariant under orthogonal matrix <sup>1</sup>, we have  $\|\mathbf{x}\|^2 = \|\mathbf{Q}_{11}\mathbf{x}\|^2 + \|\mathbf{Q}_{21}\mathbf{x}\|^2 = 1$ . Thus, by Definition 14.3 of spectral norm, we have

$$\|\mathbf{Q}_{21}\|_2^2 = \max_{\mathbf{x} \in \mathbb{R}^r: \|\mathbf{x}\|_2=1} \|\mathbf{Q}_{21}\mathbf{x}\|_2^2 = 1 - \min_{\mathbf{x} \in \mathbb{R}^r: \|\mathbf{x}\|_2=1} \|\mathbf{Q}_{11}\mathbf{x}\|_2^2 = 1 - \sigma_{\min}(\mathbf{Q}_{11})^2,$$

where  $\sigma_{\min}(\mathbf{Q}_{11})^2$  is the minimal singular value of  $\mathbf{Q}_{11}$ .

Analogously, by applying the process above on  $\mathbf{Q}^\top$ , we will obtain

$$\|\mathbf{Q}_{12}\|_2^2 = 1 - \sigma_{\min}(\mathbf{Q}_{11}^\top)^2.$$

That is

$$\|\mathbf{Q}_{12}\|_2^2 = 1 - \sigma_{\min}(\mathbf{Q}_{11})^2.$$

This implies

$$\|\mathbf{Q}_{21}\|_2 = \|\mathbf{Q}_{12}\|_2.$$

Write out the distance again by

$$\text{dist}(\mathcal{V}_1, \mathcal{V}_2) = \left\| \begin{bmatrix} \mathbf{0} & \mathbf{Q}_{12} \\ -\mathbf{Q}_{21} & \mathbf{0} \end{bmatrix} \right\|_2 = \|\mathbf{Z}\|_2.$$

---

<sup>1</sup>.  $\|\mathbf{x}\| = \|\mathbf{Q}\mathbf{x}\|$  for orthogonal matrix  $\mathbf{Q}$ .

Then, let  $\mathbf{w} \in \mathbb{R}^n$  with  $\|\mathbf{w}\| = 1$ , we have

$$\begin{aligned}\|\mathbf{Z}\mathbf{w}\| &= \left\| \begin{bmatrix} \mathbf{0} & \mathbf{Q}_{12} \\ -\mathbf{Q}_{21} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \right\| \\ &= \left\| \begin{bmatrix} \mathbf{Q}_{12}\mathbf{v} \\ -\mathbf{Q}_{21}\mathbf{u} \end{bmatrix} \right\| \\ &= \sqrt{\|\mathbf{Q}_{12}\mathbf{v}\|^2 + \|\mathbf{Q}_{21}\mathbf{u}\|^2} \\ &\stackrel{*}{\leq} \|\mathbf{Q}_{12}\|_2 \sqrt{\|\mathbf{v}\|^2 + \|\mathbf{u}\|^2} = \|\mathbf{Q}_{21}\|_2 \sqrt{\|\mathbf{v}\|^2 + \|\mathbf{u}\|^2} \\ &= \|\mathbf{Q}_{12}\| \|\mathbf{w}\| = \|\mathbf{Q}_{21}\| \|\mathbf{w}\| \\ &= \|\mathbf{Q}_{12}\| = \|\mathbf{Q}_{21}\|,\end{aligned}$$

where the Inequality (\*) is from the Matrix-vector product that  $\|\mathbf{Ax}\| \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|$  for all matrix  $\mathbf{A}$  and vector  $\mathbf{x}$ , and the upper bound of  $\|\mathbf{Z}\mathbf{w}\|$  is the spectral norm  $\text{dist}(\mathcal{V}_1, \mathcal{V}_2) = \|\mathbf{Z}\|_2$ , and this completes the proof. ■

From the lemma above, we realize that

$$0 \leq \text{dist}(\mathcal{V}_1, \mathcal{V}_2) \leq 1.$$

And it is easy to show that

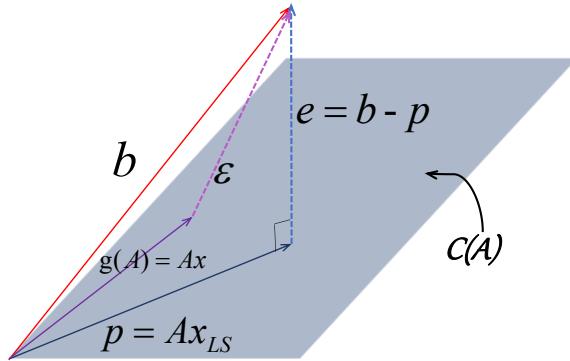
$$\begin{cases} \text{dist}(\mathcal{V}_1, \mathcal{V}_2) = 0 & \rightarrow \quad \mathcal{V}_1 = \mathcal{V}_2; \\ \text{dist}(\mathcal{V}_1, \mathcal{V}_2) = 1 & \rightarrow \quad \mathcal{V}_1 \cap \mathcal{V}_2^\perp \neq \{0\}. \end{cases}$$

## D.5 Projection for LS with Noise Disturbance

Assume further that  $\mathbf{b}$  comes from some ideal function  $g(\mathbf{A}) \in \mathcal{C}(\mathbf{A})$  and  $\mathbf{b} = g(\mathbf{A}) + \boldsymbol{\epsilon}$ , where  $\boldsymbol{\epsilon}$  is the noise. In this case, we assume that the observed values  $\mathbf{b}$  differ from the true function  $g(\mathbf{A}) = \mathbf{Ax}$  by additive noise. This situation is shown in Figure 27.4. We thus get the full picture of the problem.

1. Vector of Outputs (Responses):  $\mathbf{y} \in \mathbb{R}^m$  is an  $m \times 1$  vector of observations of the output variable and  $n$  is the sample size;
2. Design Matrix:  $\mathbf{A}$  is an  $m \times n$  matrix of inputs and  $n$  is the dimension of the inputs for each observation;
3. Vector of Parameters:  $\mathbf{x} \in \mathbb{R}^n$  is a  $n \times 1$  vector of regression coefficients;
4. Vector of Noises:  $\boldsymbol{\epsilon} \in \mathbb{R}^m$  is an  $m \times 1$  vector of noises;
5. Vector of Errors:  $\mathbf{e} \in \mathbb{R}^m$  is an  $m \times 1$  vector of errors. For predicted outputs  $\mathbf{p}$ ,  $\mathbf{e} = \mathbf{b} - \mathbf{p}$ . Thus  $\mathbf{e}$  is different from  $\boldsymbol{\epsilon}$ . The later is unobservable while the former is a byproduct of our linear model. In some texts,  $\mathbf{e}$  is denoted as  $\hat{\boldsymbol{\epsilon}}$  to make connection to  $\boldsymbol{\epsilon}$ .

By introducing the noise vector, we provide more facts about the hat matrix:



**Figure 27.4:** Projection onto the hyperplane of  $C(A)$  and disturbed by noise.

**Remark 27.19: Important Facts About Hat Matrix (Part 2)**

4. Error vector  $e = b - p = \underline{(\mathbf{I} - \mathbf{H})b} = (\mathbf{I} - \mathbf{H})(\mathbf{Ax} + \epsilon) = (\mathbf{I} - \mathbf{H})\mathbf{Ax} + (\mathbf{I} - \mathbf{H})\epsilon = \underline{(\mathbf{I} - \mathbf{H})\epsilon}$ : projecting  $b$  onto the perpendicular space is equivalent to projecting  $\epsilon$  onto the perpendicular space. This property can be easily checked from the geometric meaning of  $b$  and  $\epsilon$  as shown in Figure 27.4;
5.  $p$  and  $e$  are orthogonal,  $H\epsilon$  and  $e$  are orthogonal;
6. Pythagoras:  $\|b\|^2 = \|p\|^2 + \|e\|^2$  and  $\|\epsilon\|^2 = \|H\epsilon\|^2 + \|e\|^2$ ;
7. Pythagoras in general: for any orthogonal projection matrix  $P$ , we have  $\|\mathbf{x}\|^2 = \|Px\|^2 + \|(I - P)x\|^2$ .

The Pythagoras in general can be easily verified that

$$\begin{aligned}
 \|Px\|^2 + \|(I - P)x\|^2 &= \mathbf{x}^\top P^\top Px + \mathbf{x}^\top (I - P)^\top (I - P)x \\
 &= \mathbf{x}^\top Px + \mathbf{x}^\top (I - P)x \\
 &= \mathbf{x}^\top [Px + (I - P)x] \\
 &= \|\mathbf{x}\|^2.
 \end{aligned}$$

A more detailed analysis of this noise disturbed linear model can be found in (Lu, 2021d).

## Appendix E. Pseudo-Inverse

If matrix  $\mathbf{A}$  is nonsingular, then the linear system  $\mathbf{b} = \mathbf{Ax}$  can be easily solved by the inverse of  $\mathbf{A}$  such that  $\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$ . However, the inverse of an  $m \times n$  matrix  $\mathbf{A}$  does not exist if  $\mathbf{A}$  is not square or  $\mathbf{A}$  is singular. But we can still find its pseudo-inverse, an  $n \times m$  matrix denoted by  $\mathbf{A}^+$ . Before the discussion of pseudo-inverse, we firstly introduce one-sided inverse, generalized inverse, and reflexive generalized inverse that are the prerequisites of pseudo-inverse and some properties of them. However, readers can skip the three sections and still get the whole picture of pseudo-inverse.

### E.1 One-sided Inverse

#### Definition 27.1: One-Sided Inverse

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , if there is a matrix  $\mathbf{A}_L^{-1} \in \mathbb{R}^{n \times m}$  such that

$$\mathbf{A}_L^{-1}\mathbf{A} = \mathbf{I}_n,$$

then  $\mathbf{A}_L^{-1}$  is a **left inverse** of  $\mathbf{A}$ , and  $\mathbf{A}$  is called **left-invertible**. Similarly, if there is a matrix  $\mathbf{A}_R^{-1} \in \mathbb{R}^{n \times m}$  such that

$$\mathbf{A}\mathbf{A}_R^{-1} = \mathbf{I}_m,$$

then  $\mathbf{A}_R^{-1}$  is a **right inverse** of  $\mathbf{A}$ , and  $\mathbf{A}$  is called **right-invertible**.

**A word on the notation:** Note here the superscript  $-1$  in  $\mathbf{A}_L^{-1}$  and  $\mathbf{A}_R^{-1}$  does not mean the inverse of  $\mathbf{A}_L$  or  $\mathbf{A}_R$  but the one-sided inverse of  $\mathbf{A}$ .

#### Lemma 27.2: (One-Sided Invertible)

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , we have

- $\mathbf{A}$  is left-invertible if and only if  $\mathbf{A}$  has full column rank (which implies  $m \geq n$ );
- $\mathbf{A}$  is right-invertible if and only if  $\mathbf{A}$  has full row rank (which implies  $m \leq n$ ).

**Proof** [of Lemma 27.2] We will show by forward implication and backward implication separately.

**Backward implication** Suppose  $\mathbf{A}$  has full column rank, then  $\mathbf{A}^\top\mathbf{A} \in \mathbb{R}^{n \times n}$  has full rank (Lemma 14.2, p. 267). Therefore,  $(\mathbf{A}^\top\mathbf{A})^{-1}(\mathbf{A}^\top\mathbf{A}) = \mathbf{I}_n$ . That is  $(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$  is a left inverse of  $\mathbf{A}$ .

**Forward implication** For the reverse, now suppose  $\mathbf{A}$  is left-invertible and there exists an  $\mathbf{A}_L^{-1}$  such that  $\mathbf{A}_L^{-1}\mathbf{A} = \mathbf{I}_n$ . Since all rows of  $\mathbf{A}_L^{-1}\mathbf{A}$  are the combinations of the rows of  $\mathbf{A}$ , that is, the row space of  $\mathbf{A}_L^{-1}\mathbf{A}$  is a subset of the row space of  $\mathbf{A}$ . We then have  $\text{rank}(\mathbf{A}) \geq \text{rank}(\mathbf{A}_L^{-1}\mathbf{A}) = \text{rank}(\mathbf{I}_n) = n$  which implies  $\text{rank}(\mathbf{A}) = n$  and  $\mathbf{A}$  has full column rank.

Similarly, we can show  $\mathbf{A}$  is right-invertible if and only if  $\mathbf{A}$  has full row rank and  $\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}$  is a right inverse of  $\mathbf{A}$ .  $\blacksquare$

We have shown in the above proof that  $(\mathbf{A}^\top\mathbf{A})^{-1}\mathbf{A}^\top$  is a specific left inverse of  $\mathbf{A}$  if  $\mathbf{A}$  has full column rank, and  $\mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top)^{-1}$  is a specific right inverse of  $\mathbf{A}$  if  $\mathbf{A}$  has full row rank. However, the inverse of a  $k \times k$  nonsingular matrix requires  $2k^3$  floating points operations (flops) which is a complex procedure (Section 1.11, p. 46). In our case, the inverse of  $\mathbf{A}^\top\mathbf{A}$  requires  $2n^3$  flops and the inverse of  $\mathbf{A}\mathbf{A}^\top$  requires  $2m^3$  flops. A simpler way to find a one-sided inverse is through elementary operations.

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has full column rank and we apply **row elementary operations**  $\mathbf{E} \in \mathbb{R}^{m \times m}$  on  $[\mathbf{A}, \mathbf{I}_m]$  such that

$$\mathbf{E} [\mathbf{A} \quad \mathbf{I}_m] = \begin{bmatrix} \mathbf{I}_n & \mathbf{G} \\ \mathbf{0} & \mathbf{Z} \end{bmatrix}, \quad (27.1)$$

where  $\mathbf{G} \in \mathbb{R}^{n \times m}$ ,  $\mathbf{I}_m$  is an  $m \times m$  identity matrix,  $\mathbf{I}_n$  is an  $n \times n$  identity matrix, and  $\mathbf{Z}$  is an  $(m - n) \times m$  matrix. Then, it can be easily verified that  $\mathbf{G}\mathbf{A} = \mathbf{I}_n$  and  $\mathbf{G}$  is a left inverse of  $\mathbf{A}$ .

Similarly, suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has full row rank and we apply **column elementary operations**  $\mathbf{E} \in \mathbb{R}^{n \times n}$  on  $[\mathbf{A}^\top, \mathbf{I}_n]^\top$  such that

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{I}_n \end{bmatrix} \mathbf{E} = \begin{bmatrix} \mathbf{I}_m & \mathbf{0} \\ \mathbf{G} & \mathbf{Z} \end{bmatrix}, \quad (27.2)$$

where  $\mathbf{Z}$  is an  $n \times (n - m)$  matrix. Then,  $\mathbf{G} \in \mathbb{R}^{n \times m}$  is a right inverse of  $\mathbf{A}$ .

More generally, the following two propositions show us how to find more left inverses or right inverses of a matrix.

### Proposition 27.3: (Finding Left Inverses)

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is left-invertible ( $m \geq n$ ), then

$$\mathbf{A}_L^{-1} = [(\mathbf{A}_1^{-1} - \mathbf{Y}\mathbf{A}_2\mathbf{A}_1^{-1}) \quad \mathbf{Y}] \mathbf{E},$$

is a left inverse of  $\mathbf{A}$ , where  $\mathbf{Y} \in \mathbb{R}^{n \times (m-n)}$  can be any matrix, and  $\mathbf{E}\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$  is the row elementary transformation of  $\mathbf{A}$  such that  $\mathbf{A}_1 \in \mathbb{R}^{n \times n}$  is invertible (since  $\mathbf{A}$  has full column rank  $n$ ) and  $\mathbf{E} \in \mathbb{R}^{m \times m}$ .

One can verify that  $\mathbf{G}$  in Equation (27.1) is a specific left inverse of  $\mathbf{A}$  by setting  $\mathbf{Y} = \mathbf{0}$ . Since  $\mathbf{E} = \begin{bmatrix} \mathbf{G} \\ * \end{bmatrix}$ ,  $\mathbf{A}_1 = \mathbf{I}_p$ , and  $\mathbf{A}_2 = \mathbf{0}$ , we have

$$\mathbf{A}_L^{-1} = [(\mathbf{A}_1^{-1} - \mathbf{Y}\mathbf{A}_2\mathbf{A}_1^{-1}) \quad \mathbf{Y}] \mathbf{E} = \mathbf{G} + \mathbf{Y}\mathbf{Z} = \mathbf{G},$$

where the last equation is from the assumption that  $\mathbf{Y} = \mathbf{0}$ .

**Proposition 27.4: (Finding Right Inverses)**

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is right-invertible ( $m \leq n$ ), then

$$\mathbf{A}_R^{-1} = \mathbf{E} \begin{bmatrix} (\mathbf{A}_1^{-1} - \mathbf{A}_1^{-1} \mathbf{A}_2 \mathbf{Y}) \\ \mathbf{Y} \end{bmatrix},$$

is a right inverse of  $\mathbf{A}$ , where  $\mathbf{Y} \in \mathbb{R}^{(n-m) \times m}$  can be any matrix, and  $\mathbf{AE} = [\mathbf{A}_1 \ \mathbf{A}_2]$  is the column elementary transformation of  $\mathbf{A}$  such that  $\mathbf{A}_1 \in \mathbb{R}^{m \times m}$  is invertible (since  $\mathbf{A}$  has full row rank  $m$ ) and  $\mathbf{E} \in \mathbb{R}^{n \times n}$ .

Similarly, one can verify that  $\mathbf{G}$  in Equation (27.2) is a specific right inverse of  $\mathbf{A}$  by setting  $\mathbf{Y} = \mathbf{0}$ . Since  $\mathbf{E} = [\mathbf{G}, \mathbf{Z}]$ ,  $\mathbf{A}_1 = \mathbf{I}_m$ , and  $\mathbf{A}_2 = \mathbf{0}$ , we have

$$\mathbf{A}_R^{-1} = \mathbf{E} \begin{bmatrix} (\mathbf{A}_1^{-1} - \mathbf{A}_1^{-1} \mathbf{A}_2 \mathbf{Y}) \\ \mathbf{Y} \end{bmatrix} = \mathbf{G} + \mathbf{ZY} = \mathbf{G},$$

where again the last equality is from the assumption that  $\mathbf{Y} = \mathbf{0}$ .

Under specific conditions, the linear system  $\mathbf{Ax} = \mathbf{b}$  has a unique solution.

**Proposition 27.5: (Unique Linear System Solution)**

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is left-invertible ( $m \geq n$ ), and  $\mathbf{A}_L^{-1}$  is a left inverse of  $\mathbf{A}$ . Then the linear system  $\mathbf{Ax} = \mathbf{b}$  has a **unique** solution if and only if

$$(\mathbf{I}_m - \mathbf{AA}_L^{-1})\mathbf{b} = \mathbf{0}.$$

And the unique solution is given by

$$\hat{\mathbf{x}} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}.$$

**Proof** [of Proposition 27.5] We will prove by forward implication and backward implication separately as follows.

**Forward implication** Suppose  $x_0$  is the solution of  $\mathbf{Ax} = \mathbf{b}$ , then

$$\begin{aligned} \mathbf{AA}_L^{-1}(\mathbf{Ax}_0) &= \mathbf{AA}_L^{-1}\mathbf{b} \\ \mathbf{A}(\mathbf{A}_L^{-1}\mathbf{A})\mathbf{x}_0 &= \mathbf{Ax}_0 = \mathbf{b}. \end{aligned}$$

That is,  $\mathbf{AA}_L^{-1}\mathbf{b} = \mathbf{b}$  and  $(\mathbf{I}_m - \mathbf{AA}_L^{-1})\mathbf{b} = \mathbf{0}$ .

**Backward implication** For the reverse, suppose  $(\mathbf{I}_m - \mathbf{AA}_L^{-1})\mathbf{b} = \mathbf{0}$ , and let  $\mathbf{x}_0 = \mathbf{A}_L^{-1}\mathbf{b}$ . Then substitute  $\mathbf{x}_0 = \mathbf{A}_L^{-1}\mathbf{b}$  into  $(\mathbf{I}_m - \mathbf{AA}_L^{-1})\mathbf{b} = \mathbf{0}$ , we have

$$\mathbf{Ax}_0 = \mathbf{b},$$

which implies  $\mathbf{x}_0 = \mathbf{A}_L^{-1}\mathbf{b}$  is a solution of  $\mathbf{Ax} = \mathbf{b}$  if  $(\mathbf{I}_m - \mathbf{AA}_L^{-1})\mathbf{b} = \mathbf{0}$ .

To prove the uniqueness, suppose  $\mathbf{x}_0$  and  $\mathbf{x}_1$  are two solutions of  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . We have  $\mathbf{A}\mathbf{x}_0 = \mathbf{A}\mathbf{x}_1 = \mathbf{b}$  such that  $\mathbf{A}(\mathbf{x}_0 - \mathbf{x}_1) = \mathbf{0}$ . Since  $\mathbf{A}$  is left-invertible so that  $\mathbf{A}$  has full column rank  $n$ , the dimension of the row space of  $\mathbf{A}$  is  $n$  as well such that the null space of  $\mathbf{A}$  is of dimension 0 (i.e.,  $\dim(\mathcal{C}(\mathbf{A}^\top)) + \dim(\mathcal{N}(\mathbf{A})) = n$  by the fundamental theorem of linear algebra, see Theorem 27.1, p. 428). Then  $\mathbf{x}_0 = \mathbf{x}_1$  which completes the proof. ■

In the fundamental theorem of linear algebra Figure 27.1, the row space of  $\mathbf{A}$  is the whole space of  $\mathbb{R}^n$  if  $\mathbf{A}$  is left-invertible (i.e.,  $\mathbf{A}$  has full column rank  $n$ ). If the condition  $(\mathbf{I}_m - \mathbf{A}\mathbf{A}_L^{-1})\mathbf{b} = \mathbf{0}$  is satisfied, then  $\mathbf{A}\mathbf{A}_L^{-1}\mathbf{b} = \mathbf{b}$ , it implies that  $\mathbf{b}$  is in the column space of  $\mathbf{A}$  such that  $\mathbf{A}\mathbf{x} = \mathbf{b}$  has at least one solution and the above proposition shows that this solution is unique.

### Proposition 27.6: (Always Have Solution)

Suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is right-invertible (which implies  $m \leq n$ ), and  $\mathbf{A}_R^{-1}$  is a right inverse of  $\mathbf{A}$ . Then for any  $\mathbf{b} \in \mathbb{R}^m$ , the linear system  $\mathbf{A}\mathbf{x} = \mathbf{b}$  has solutions, and the solution is given by

$$\hat{\mathbf{x}} = \mathbf{A}_R^{-1}\mathbf{b},$$

where the right inverse  $\mathbf{A}_R^{-1}$  is not necessarily unique.

**Proof** [of Proposition 27.6] It can be easily verified that

$$(\mathbf{A}\mathbf{A}_R^{-1})\mathbf{b} = \mathbf{I}_n\mathbf{b} = \mathbf{b},$$

so that  $\mathbf{A}_R^{-1}\mathbf{b}$  is a solution of  $\mathbf{A}\mathbf{x} = \mathbf{b}$ . ■

We notice that if  $\mathbf{A}$  is right-invertible, then  $\mathbf{A}$  has full row rank  $m$ . In the fundamental theorem of linear algebra Figure 27.1, the column space of  $\mathbf{A}$  is the whole space of  $\mathbb{R}^m$  if  $\mathbf{A}$  is right-invertible. Then any vector  $\mathbf{b} \in \mathbb{R}^m$  is in the column space of  $\mathbf{A}$  such that  $\mathbf{A}\mathbf{x} = \mathbf{b}$  has at least one solution.

## E.2 Generalized Inverse (g-inverse)

We mentioned previously that if matrix  $\mathbf{A}$  is nonsingular, then the linear system  $\mathbf{b} = \mathbf{A}\mathbf{x}$  can be easily solved by the inverse of  $\mathbf{A}$  such that  $\hat{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$ . However, the inverse of an  $m \times n$  matrix  $\mathbf{A}$  does not exist if  $\mathbf{A}$  is not square or  $\mathbf{A}$  is singular. But still, when  $\mathbf{b}$  is in the column space of  $\mathbf{A}$ , we can find the solution of the linear system as well. The relationship between the solution  $\hat{\mathbf{x}}$  and  $\mathbf{b}$  is given by the generalized inverse of  $\mathbf{A}$ :  $\hat{\mathbf{x}} = \mathbf{A}^{-}\mathbf{b}$ .

### Definition 27.7: Generalized Inverse

Any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has rank  $r$  with  $r \leq \min\{n, m\}$ , i.e., could be full ranked or non-full ranked. Then a generalized inverse  $\mathbf{A}^{-} \in \mathbb{R}^{n \times m}$  of  $\mathbf{A}$  is a matrix that satisfies

$$(C1) \quad \mathbf{A}\mathbf{A}^{-}\mathbf{A} = \mathbf{A},$$

or equivalently,

$$(C1.1) \quad \mathbf{A}\mathbf{A}^{-}\mathbf{b} = \mathbf{b}$$

for any  $\mathbf{b} \in \mathcal{C}(\mathbf{A})$ .

To see the equivalence between (C1) and (C1.1), that is, we want to show  $\mathbf{A}$  satisfies (C1) if and only if it satisfies (C1.1).

**Forward implication** For any  $\mathbf{b} \in \mathcal{C}(\mathbf{A})$ , we can find an  $\mathbf{x} \in \mathbb{R}^n$  such that  $\mathbf{Ax} = \mathbf{b}$ . If  $\mathbf{A}$  and  $\mathbf{A}^{-}$  satisfy (C1), then

$$\mathbf{A}\mathbf{A}^{-}\mathbf{Ax} = \mathbf{Ax} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}\mathbf{A}^{-}\mathbf{b} = \mathbf{b},$$

which implies  $\mathbf{A}$  and  $\mathbf{A}^{-}$  also satisfy (C1.1).

**Backward implication** For the reverse, suppose  $\mathbf{A}$  and  $\mathbf{A}^{-}$  satisfy (C1.1), then

$$\mathbf{A}\mathbf{A}^{-}\mathbf{b} = \mathbf{b} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}\mathbf{A}^{-}\mathbf{Ax} = \mathbf{Ax},$$

which implies  $\mathbf{A}$  and  $\mathbf{A}^{-}$  also satisfy (C1).

Multiply on the left of (C1) by  $\mathbf{A}^{-}$  and follow the definition of projection matrix in Definition 27.5, we obtain  $\mathbf{A}^{-}\mathbf{AA}^{-}\mathbf{A} = \mathbf{A}^{-}\mathbf{A}$  such that  $\mathbf{A}^{-}\mathbf{A}$  is idempotent, which implies  $\mathbf{A}^{-}\mathbf{A}$  is a projection matrix (not necessarily to be an orthogonal projection).

### Lemma 27.8: (Projection Matrix from Generalized Inverse)

For any matrix  $\mathbf{A}$ , and its generalized inverse  $\mathbf{A}^{-}$ ,  $\mathbf{A}^{-}\mathbf{A}$  is a projection matrix but not necessarily an orthogonal projection. Same claim can be applied to  $\mathbf{AA}^{-}$  as well.

### Lemma 27.9: (Rank of Generalized Inverse)

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and its generalized inverse  $\mathbf{A}^{-} \in \mathbb{R}^{n \times m}$ , then

$$\text{rank}(\mathbf{A}^{-}) \geq \text{rank}(\mathbf{A}).$$

Specifically, we also have  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{AA}^{-}) = \text{rank}(\mathbf{A}^{-}\mathbf{A})$ .

**Proof** [of Lemma 27.9] From (C1), we notice that  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{AA}^{-}\mathbf{A})$ . And

$$\text{rank}(\mathbf{AA}^{-}\mathbf{A}) \leq \text{rank}(\mathbf{AA}^{-}) \leq \text{rank}(\mathbf{A}^{-}),$$

where the first inequality comes from the fact that the columns of  $\mathbf{AA}^{-}\mathbf{A}$  are combinations of columns of  $\mathbf{AA}^{-}$ , and the second inequality comes from the fact that the rows of  $\mathbf{AA}^{-}$  are combinations of rows of  $\mathbf{A}^{-}$ . This implies  $\text{rank}(\mathbf{A}^{-}) \geq \text{rank}(\mathbf{A})$ .

For the second part, we have

$$\text{rank}(\mathbf{A}) \geq \text{rank}(\mathbf{AA}^{-}) \geq \text{rank}(\mathbf{AA}^{-}\mathbf{A}),$$

where the first inequality is from the fact that the columns of  $\mathbf{A}\mathbf{A}^-$  are combinations of the columns of  $\mathbf{A}$ , and the second inequality is from the fact that the columns of  $\mathbf{A}\mathbf{A}^-\mathbf{A}$  are combinations of the columns of  $\mathbf{A}\mathbf{A}^-$ . From (C1) again,  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}\mathbf{A}^-\mathbf{A})$  which implies by “sandwiching” that

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}\mathbf{A}^-) = \text{rank}(\mathbf{A}\mathbf{A}^-\mathbf{A}).$$

Similarly, we also have

$$\text{rank}(\mathbf{A}) \geq \text{rank}(\mathbf{A}^-\mathbf{A}) \geq \text{rank}(\mathbf{A}\mathbf{A}^-\mathbf{A}),$$

where the first inequality is from the fact that the rows of  $\mathbf{A}^-\mathbf{A}$  are combinations of the rows of  $\mathbf{A}$ , and the second inequality is from the fact that the rows of  $\mathbf{A}\mathbf{A}^-\mathbf{A}$  are combinations of the rows of  $\mathbf{A}^-\mathbf{A}$ . By “sandwiching” again, we have

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^-\mathbf{A}) = \text{rank}(\mathbf{A}\mathbf{A}^-\mathbf{A}),$$

which completes the proof. ■

In Lemma 27.2, we have shown that the left inverse exists if and only if  $\mathbf{A}$  has full column rank, and the right inverse exists if and only if  $\mathbf{A}$  has full row rank. However, this is not required in generalized inverses. When this full rank condition is satisfied, we have the following property for generalized inverses.

#### **Lemma 27.10: (Full Rank Generalized Inverse)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and its generalized inverse  $\mathbf{A}^- \in \mathbb{R}^{n \times m}$ , then we have

- 1).  $\mathbf{A}$  has full column rank if and only if  $\mathbf{A}^-\mathbf{A} = \mathbf{I}_n$ ;
- 2).  $\mathbf{A}$  has full row rank if and only if  $\mathbf{A}\mathbf{A}^- = \mathbf{I}_m$ .

**Proof** [of Lemma 27.10] For 1). We will prove by forward implication and backward implication separately as follows.

**Forward implication** Suppose  $\mathbf{A}$  has full column rank, and we have shown in Lemma 27.9 that  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}\mathbf{A}^-) = \text{rank}(\mathbf{A}^-\mathbf{A})$ . Then  $\text{rank}(\mathbf{A}^-\mathbf{A}) = \text{rank}(\mathbf{A}) = n$  and  $\mathbf{A}^-\mathbf{A} \in \mathbb{R}^{n \times n}$  is nonsingular. We obtain

$$\mathbf{I}_n = (\mathbf{A}^-\mathbf{A})(\mathbf{A}^-\mathbf{A})^{-1} = \mathbf{A}^-(\mathbf{A}\mathbf{A}^-\mathbf{A})(\mathbf{A}^-\mathbf{A})^{-1} = \mathbf{A}^-\mathbf{A}.$$

**Backward implication** For the reverse, suppose  $\mathbf{A}^-\mathbf{A} = \mathbf{I}_n$  which implies  $\text{rank}(\mathbf{A}^-\mathbf{A}) = n$ . From  $\text{rank}(\mathbf{A}^-\mathbf{A}) = \text{rank}(\mathbf{A})$ , we have  $\text{rank}(\mathbf{A}) = n$  such that  $\mathbf{A}$  has full column rank.

Similarly, we can show  $\mathbf{A}$  has full row rank if and only if  $\mathbf{A}\mathbf{A}^- = \mathbf{I}_m$ . ■

**Lemma 27.11: (Constructing Generalized Inverse)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and its generalized inverse  $\mathbf{A}^- \in \mathbb{R}^{n \times m}$ , there exists an  $n \times m$  matrix  $\mathbf{L}$  such that

$$\bar{\mathbf{A}} = \mathbf{A}^- + \mathbf{L} - \mathbf{A}^- \mathbf{A} \mathbf{L} \mathbf{A} \mathbf{A}^- \quad (27.3)$$

is also a generalized inverse of  $\mathbf{A}$ . In addition, for any generalized inverse  $\bar{\mathbf{A}}$ , there exists a matrix  $\mathbf{L}$  so that Equation (27.3) is satisfied.

**Proof** [of Lemma 27.11] Write out the equation

$$\begin{aligned} \mathbf{A} \bar{\mathbf{A}} \mathbf{A} &= \mathbf{A}(\mathbf{A}^- + \mathbf{L} - \mathbf{A}^- \mathbf{A} \mathbf{L} \mathbf{A} \mathbf{A}^-) \mathbf{A} = \mathbf{A} \mathbf{A}^- \mathbf{A} + \mathbf{A} \mathbf{L} \mathbf{A} - (\mathbf{A} \mathbf{A}^- \mathbf{A}) \mathbf{L} (\mathbf{A} \mathbf{A}^- \mathbf{A}) \\ &= \mathbf{A} \mathbf{A}^- \mathbf{A} + \mathbf{A} \mathbf{L} \mathbf{A} - \mathbf{A} \mathbf{L} \mathbf{A} = \mathbf{A}, \end{aligned}$$

so that  $\bar{\mathbf{A}}$  is a generalized inverse of  $\mathbf{A}$ .

Suppose now that  $\mathbf{M}$  is any generalized inverse of  $\mathbf{A}$ , and define  $\mathbf{L} = \mathbf{M} - \mathbf{A}^-$ . Recall that  $\mathbf{A} \mathbf{M} \mathbf{A} = \mathbf{A}$ , we have

$$\begin{aligned} \mathbf{A}^- + \mathbf{L} - \mathbf{A}^- \mathbf{A} \mathbf{L} \mathbf{A} \mathbf{A}^- &= \mathbf{A}^- + (\mathbf{M} - \mathbf{A}^-) - \mathbf{A}^- \mathbf{A} (\mathbf{M} - \mathbf{A}^-) \mathbf{A} \mathbf{A}^- \\ &= \mathbf{M} - \mathbf{A}^- (\mathbf{A} \mathbf{M} \mathbf{A}) \mathbf{A}^- + \mathbf{A}^- (\mathbf{A} \mathbf{A}^- \mathbf{A}) \mathbf{A}^- \\ &= \mathbf{M} - \mathbf{A}^- \mathbf{A} \mathbf{A}^- + \mathbf{A}^- \mathbf{A} \mathbf{A}^- \\ &= \mathbf{M}, \end{aligned}$$

which implies  $\mathbf{L}$  can be constructed for any generalized inverse  $\mathbf{M}$ . ■

**Lemma 27.12: (Generalized Inverse Properties)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and its generalized inverse  $\mathbf{A}^- \in \mathbb{R}^{n \times m}$ , then

- 1).  $(\mathbf{A}^\top)^- = (\mathbf{A}^-)^\top$ , i.e.,  $(\mathbf{A}^-)^\top$  is the generalized inverse of  $\mathbf{A}^\top$ ;
- 2). For any  $a \neq 0$ ,  $\frac{1}{a}\mathbf{A}^-$  is the generalized inverse of  $a\mathbf{A}$ ;
- 3). Suppose  $\mathbf{L} \in \mathbb{R}^{m \times m}$  and  $\mathbf{M} \in \mathbb{R}^{n \times n}$  are both invertible, then  $\mathbf{M}^{-1}\mathbf{A}^-\mathbf{L}^{-1}$  is a generalized inverse of  $\mathbf{L}\mathbf{A}\mathbf{M}$ ;
- 4).  $\mathcal{C}(\mathbf{A}\mathbf{A}^-) = \mathcal{C}(\mathbf{A})$  and  $\mathcal{N}(\mathbf{A}^-\mathbf{A}) = \mathcal{N}(\mathbf{A})$ .

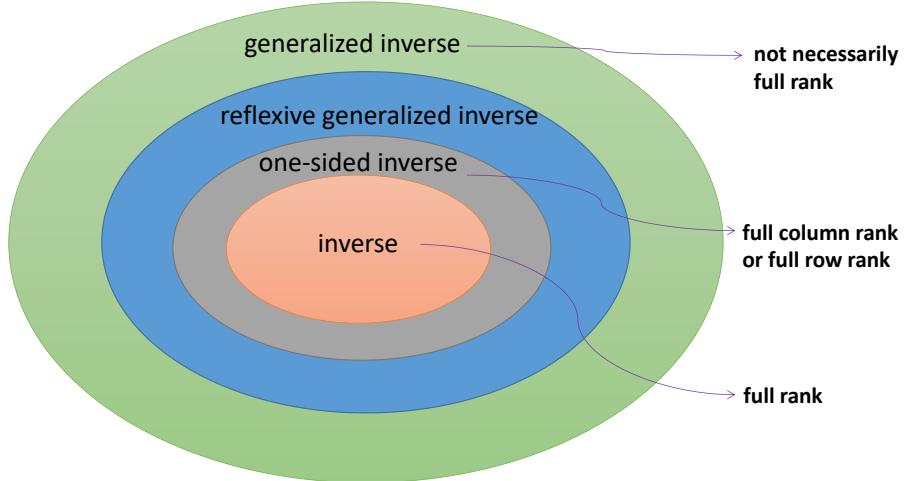
**Proof** [of Lemma 27.12] For 1), from (C1),  $\mathbf{A}\mathbf{A}^-\mathbf{A} = \mathbf{A}$ , we have  $\mathbf{A}^\top(\mathbf{A}^-)^\top\mathbf{A}^\top = \mathbf{A}^\top$  such that  $(\mathbf{A}^-)^\top$  is the generalized inverse of  $\mathbf{A}^\top$ .

For 2), it can be easily verified that  $(a\mathbf{A})(\frac{1}{a}\mathbf{A}^-)(a\mathbf{A}) = (a\mathbf{A})$  such that  $\frac{1}{a}\mathbf{A}^-$  is the generalized inverse of  $a\mathbf{A}$  for any  $a \neq 0$ .

For 3), we realize that  $(\mathbf{L}\mathbf{A}\mathbf{M})(\mathbf{M}^{-1}\mathbf{A}^-\mathbf{L}^{-1})(\mathbf{L}\mathbf{A}\mathbf{M}) = \mathbf{L}\mathbf{A}\mathbf{A}^-\mathbf{A}\mathbf{M} = \mathbf{L}\mathbf{A}\mathbf{M}$  which implies  $\mathbf{M}^{-1}\mathbf{A}^-\mathbf{L}^{-1}$  is a generalized inverse of  $\mathbf{L}\mathbf{A}\mathbf{M}$ .

For 4), since the columns of  $\mathbf{A}\mathbf{A}^-$  are the combinations of the columns of  $\mathbf{A}$ , then  $\mathcal{C}(\mathbf{A}\mathbf{A}^-) \subseteq \mathcal{C}(\mathbf{A})$ . And we proved that  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}\mathbf{A}^-)$  in Lemma 27.9, then  $\mathcal{C}(\mathbf{A}\mathbf{A}^-) = \mathcal{C}(\mathbf{A})$ . Similarly, we could prove  $\mathcal{N}(\mathbf{A}^-\mathbf{A}) = \mathcal{N}(\mathbf{A})$ . ■

### E.3 Reflexive Generalized Inverse (rg-inverse)



**Figure 27.5:** Relationship of different inverses.

#### Definition 27.13: Reflexive Generalized Inverse

Any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has rank  $r$  with  $r \leq \min\{n, m\}$ , i.e., could be full ranked or non-full ranked. Then a reflexive generalized inverse  $\mathbf{A}_r^- \in \mathbb{R}^{n \times m}$  of  $\mathbf{A}$  is a matrix that satisfies

$$(C1) \quad \mathbf{A}\mathbf{A}_r^-\mathbf{A} = \mathbf{A},$$

and

$$(C2) \quad \mathbf{A}_r^-\mathbf{A}\mathbf{A}_r^- = \mathbf{A}_r^-.$$

That is  $\mathbf{A}_r^-$  is a g-inverse of  $\mathbf{A}$ , and  $\mathbf{A}$  is a g-inverse of  $\mathbf{A}_r^-$ .

Suppose  $\mathbf{A}$  has rank  $r$ , then it can be factored as  $\mathbf{A} = \mathbf{E}_1 \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{E}_2$ , where  $\mathbf{E}_1 \in \mathbb{R}^{m \times m}$ ,  $\mathbf{E}_2 \in \mathbb{R}^{n \times n}$  are elementary transformations on  $\mathbf{A}$ . Then, we can construct such a reflexive generalized inverse of  $\mathbf{A}$  as

$$\mathbf{A}_r^- = \mathbf{E}_2^{-1} \begin{bmatrix} \mathbf{I}_r & \mathbf{L} \\ \mathbf{M} & \mathbf{ML} \end{bmatrix} \mathbf{E}_1^{-1},$$

where  $\mathbf{L} \in \mathbb{R}^{r \times (m-r)}$ ,  $\mathbf{M} \in \mathbb{R}^{(n-r) \times r}$  can be any matrix so that the reflexive generalized inverse is **not unique**. This construction of the reflexive generalized inverse also shows that reflexive generalized inverse exists for any matrix. This implies reflexive generalized inverse is a more general inverse of  $\mathbf{A}$  compared to the one-sided inverse which may not exist.

**Lemma 27.14: (Reflexive Generalized Inverse from g-inverse)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and  $\mathbf{L}, \mathbf{M}$  are two g-inverses of  $\mathbf{A}$ , then

$$\mathbf{Z} = \mathbf{LAM}$$

is a reflexive generalized inverse of  $\mathbf{A}$ .

It can be easily verified that  $\mathbf{AZA} = \mathbf{A}$  and  $\mathbf{ZAZ} = \mathbf{Z}$ .

**Lemma 27.15: (Reflexive Generalized Inverse from g-inverse)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , then the following two matrices  $\mathbf{L}$  and  $\mathbf{M}$  are two reflexive generalized inverses of  $\mathbf{A}$ :

$$\begin{aligned}\mathbf{L} &= (\mathbf{A}^\top \mathbf{A})^- \mathbf{A}^\top, \\ \mathbf{M} &= \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^-, \end{aligned}$$

where  $(\mathbf{A}^\top \mathbf{A})^-$  is a g-inverse of  $(\mathbf{A}^\top \mathbf{A})$ , and  $(\mathbf{A} \mathbf{A}^\top)^-$  is a g-inverse of  $(\mathbf{A} \mathbf{A}^\top)$ .

**Proof** [of Lemma 27.15] To prove the lemma, let's first check the following result.

$$\mathcal{C}(\mathbf{A}^\top \mathbf{A}) = \mathcal{C}(\mathbf{A}^\top) \quad \text{and} \quad \mathcal{N}(\mathbf{A}^\top \mathbf{A}) = \mathcal{N}(\mathbf{A})$$

Since the columns of  $\mathbf{A}^\top \mathbf{A}$  are combinations of the columns of  $\mathbf{A}^\top$ , we have  $\mathcal{C}(\mathbf{A}^\top \mathbf{A}) \subseteq \mathcal{C}(\mathbf{A}^\top)$ . In Lemma 14.2, we proved that,  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$ . This implies  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A}^\top)$  and  $\mathcal{C}(\mathbf{A}^\top \mathbf{A}) = \mathcal{C}(\mathbf{A}^\top)$ . Furthermore, the orthogonal complement of  $\mathcal{C}(\mathbf{A}^\top)$  is  $\mathcal{N}(\mathbf{A})$ , and the orthogonal complement of  $\mathcal{C}(\mathbf{A}^\top \mathbf{A})$  is  $\mathcal{N}(\mathbf{A}^\top \mathbf{A})$ . Therefore, by fundamental theorem of linear algebra in Appendix B, we have

$$\mathcal{C}(\mathbf{A}^\top \mathbf{A}) = \mathcal{C}(\mathbf{A}^\top), \quad \mathcal{N}(\mathbf{A}^\top \mathbf{A}) = \mathcal{N}(\mathbf{A}).$$

Then there exists a set of vectors  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n \in \mathbb{R}^n$  such that column- $i$  of  $\mathbf{A}^\top$  can be expressed as  $\mathbf{A}^\top \mathbf{Az}_i$ . That is, for  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n]$ , we have

$$\mathbf{A}^\top = \mathbf{A}^\top \mathbf{AZ}.$$

Then,

$$\mathbf{ALA} = (\mathbf{A}^\top \mathbf{AZ})^\top (\mathbf{A}^\top \mathbf{A})^- \mathbf{A}^\top \mathbf{A} = \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} (\mathbf{A}^\top \mathbf{A})^- \mathbf{A}^\top \mathbf{A}.$$

By condition (C1.1) of g-inverse, we have  $\mathbf{A}^\top \mathbf{A} (\mathbf{A}^\top \mathbf{A})^- \mathbf{b} = \mathbf{b}$  for any  $\mathbf{b}$ . This implies  $\mathbf{A}^\top \mathbf{A} (\mathbf{A}^\top \mathbf{A})^- \mathbf{A}^\top \mathbf{A} = \mathbf{A}^\top \mathbf{A}$  and

$$\mathbf{ALA} = (\mathbf{A}^\top \mathbf{AZ})^\top (\mathbf{A}^\top \mathbf{A})^- \mathbf{A}^\top \mathbf{A} = \mathbf{Z}^\top \mathbf{A}^\top \mathbf{A} = \mathbf{A}. \quad (27.4)$$

Write out  $\mathbf{LAL}$ , we have

$$\mathbf{LAL} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{A} (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top.$$

Same argument can be applied to  $\mathbf{A}^\top \mathbf{A} (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top = \mathbf{A}^\top$ . Then,

$$\mathbf{LAL} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top = \mathbf{L}. \quad (27.5)$$

Combine Equation (27.4) and Equation (27.5), we conclude that  $\mathbf{L}$  is a reflexive generalized inverse of  $\mathbf{A}$ . Similarly, same process can be applied to show  $\mathbf{M}$  is a reflexive generalized inverse of  $\mathbf{A}$  as well. ■

From the definition, we realize that reflexive generalized inverse is a special generalized inverse. Under specific conditions, the two inverses are equivalent.

#### Lemma 27.16: (Reflexive Generalized Inverse in G-Inverse)

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and  $\mathbf{A}^- \in \mathbb{R}^{n \times m}$  is a generalized inverse of  $\mathbf{A}$ , then  $\mathbf{A}^-$  is a reflexive generalized inverse of  $\mathbf{A}$  if and only if  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^-)$ .

**Proof** [of Lemma 27.16] We will prove by forward implication and backward implication separately as follows.

**Forward implication** Suppose  $\mathbf{A}^-$  is a generalized inverse of  $\mathbf{A}$ , then  $\mathbf{AA}^- \mathbf{A} = \mathbf{A}$ . Suppose further,  $\mathbf{A}^-$  is also a reflexive generalized inverse, then  $\mathbf{A}^- \mathbf{AA}^- = \mathbf{A}^-$ . We want to show  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^-)$ . Firstly, we have

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{AA}^- \mathbf{A}) \leq \text{rank}(\mathbf{A}^-) = \text{rank}(\mathbf{A}^- \mathbf{AA}^-) \leq \text{rank}(\mathbf{A})$$

where the two inequalities are from Lemma 27.9. This implies  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^-)$  by “sandwiching”.

**Backward implication** For the reverse, suppose  $\mathbf{A}^-$  is a generalized inverse of  $\mathbf{A}$ , then  $\mathbf{AA}^- \mathbf{A} = \mathbf{A}$ . And suppose further  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{A}^-)$ , we want to show  $\mathbf{A}^-$  is also a reflexive generalized inverse. First, we have

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{AA}^- \mathbf{A}) \leq \text{rank}(\mathbf{A}^- \mathbf{A}) \leq \text{rank}(\mathbf{A}^-) = \text{rank}(\mathbf{A}),$$

where the first inequality is from the fact that the rows of  $\mathbf{AA}^- \mathbf{A}$  are combinations of the rows of  $\mathbf{A}^- \mathbf{A}$ , and the second inequality is from the fact that the columns of  $\mathbf{A}^- \mathbf{A}$  are combinations of the columns of  $\mathbf{A}^-$ . Again by “sandwiching”, we have

$$\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{AA}^- \mathbf{A}) = \text{rank}(\mathbf{A}^- \mathbf{A}) = \text{rank}(\mathbf{A}^-) = \text{rank}(\mathbf{A}),$$

This equality  $\text{rank}(\mathbf{A}^- \mathbf{A}) = \text{rank}(\mathbf{A}^-)$  and the subspace inequality  $\mathcal{C}(\mathbf{A}^- \mathbf{A}) \subseteq \mathcal{C}(\mathbf{A}^-)$  imply  $\mathcal{C}(\mathbf{A}^- \mathbf{A}) = \mathcal{C}(\mathbf{A}^-)$ . Then there exists a set of vectors  $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_m \in \mathbb{R}^n$  such that column- $i$  of  $\mathbf{A}^-$  can be expressed as  $\mathbf{A}^- \mathbf{A} \boldsymbol{\alpha}_i$ . That is, for  $\mathbf{L} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \dots, \boldsymbol{\alpha}_m]$ , we have

$$\mathbf{A}^- = \mathbf{A}^- \mathbf{A} \mathbf{L}.$$

We realize again that  $\mathbf{A} = \mathbf{AA}^{\perp}\mathbf{A}$ , then

$$\mathbf{A} = \mathbf{AA}^{\perp}\mathbf{A} = \mathbf{A}(\mathbf{A}^{\perp}\mathbf{AL})\mathbf{A} = \mathbf{ALA},$$

where the last equality is form condition (C1) and thus  $\mathbf{L}$  is a generalized inverse of  $\mathbf{A}$ . From Lemma 27.14,  $\mathbf{A}^{\perp} = \mathbf{A}^{\perp}\mathbf{AL}$  is a reflexive generalized inverse of  $\mathbf{A}$  which completes the proof.  $\blacksquare$

### Proposition 27.17: (Rank of Reflexive Generalized Inverse)

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and its generalized inverse  $\mathbf{A}_r^{\perp} \in \mathbb{R}^{n \times m}$ . Combine the result in Lemma 27.16 and the result from the rank of g-inverses in Lemma 27.9, we have

$$\text{rank}(\mathbf{A}_r^{\perp}) = \text{rank}(\mathbf{A}) = \text{rank}(\mathbf{AA}_r^{\perp}) = \text{rank}(\mathbf{A}_r^{\perp}\mathbf{A}).$$

### Lemma 27.18: (Reflexive Generalized Inverse Properties)

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and its reflexive generalized inverse  $\mathbf{A}_r^{\perp} \in \mathbb{R}^{n \times m}$ , then

1.  $\mathcal{C}(\mathbf{AA}_r^{\perp}) = \mathcal{C}(\mathbf{A})$  and  $\mathcal{N}(\mathbf{A}_r^{\perp}\mathbf{A}) = \mathcal{N}(\mathbf{A})$ .
2.  $\mathcal{C}(\mathbf{A}_r^{\perp}\mathbf{A}) = \mathcal{C}(\mathbf{A}_r^{\perp})$  and  $\mathcal{N}(\mathbf{AA}_r^{\perp}) = \mathcal{N}(\mathbf{A}_r^{\perp})$ .

**Proof** [of Lemma 27.18] Suppose  $\mathbf{A}^{\perp}$  is a g-inverse of  $\mathbf{A}$ , we proved in Lemma 27.12 that  $\mathcal{C}(\mathbf{AA}^{\perp}) = \mathcal{C}(\mathbf{A})$  and  $\mathcal{N}(\mathbf{A}^{\perp}\mathbf{A}) = \mathcal{N}(\mathbf{A})$ . Since  $\mathbf{A}_r^{\perp}$  is a g-inverse of  $\mathbf{A}$ , and  $\mathbf{A}$  is a g-inverse of  $\mathbf{A}_r^{\perp}$ , we complete the proof.  $\blacksquare$

## E.4 Pseudo-Inverse

As we mentioned previously, for a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , we can find its pseudo-inverse, an  $n \times m$  matrix denoted by  $\mathbf{A}^+$ . In words, when  $\mathbf{A}$  multiplies a vector  $\mathbf{x}$  in its row space, this produces  $\mathbf{Ax}$  in the column space (see Figure 27.1, p. 427). Those two spaces have equal dimension  $r$ , i.e., the rank.  $\mathbf{A}$  is always invertible when restricted to these spaces and  $\mathbf{A}^+$  inverts  $\mathbf{A}$ . That is,  $\mathbf{A}^+\mathbf{Ax} = \mathbf{x}$  when  $\mathbf{x}$  is in the row space of  $\mathbf{A}$ . And  $\mathbf{AA}^+\mathbf{b} = \mathbf{b}$  when  $\mathbf{b}$  is in the column space of  $\mathbf{A}$  (see Figure 27.6, p. 457).

The null space of  $\mathbf{A}^+$  is the null space of  $\mathbf{A}^{\top}$ . It contains the vectors  $\mathbf{b}$  in  $\mathbb{R}^m$  with  $\mathbf{A}^{\top}\mathbf{b} = \mathbf{0}$ . Those vectors  $\mathbf{b}$  are perpendicular to every  $\mathbf{Ax}$  in the column space. We delay the proof of this property in Lemma 27.21.

More formally, the pseudo-inverse, or also known as Moore-Penrose pseudo-inverse,  $\mathbf{A}^+$ , is defined by the unique  $n \times m$  matrix satisfying the following four criteria:

|                                                                                                                                                                                                                                      |                                                                                                                  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| $(C1) \quad \mathbf{AA}^+ \mathbf{A} = \mathbf{A}$<br>$(C2) \quad \mathbf{A}^+ \mathbf{AA}^+ = \mathbf{A}^+$<br>$(C3) \quad (\mathbf{AA}^+)^T = \mathbf{AA}^+$<br>$(C4) \quad (\mathbf{A}^+ \mathbf{A})^T = \mathbf{A}^+ \mathbf{A}$ | $(\mathbf{A}^+ \text{ is a g-inverse of } \mathbf{A})$<br>$(\mathbf{A} \text{ is a g-inverse of } \mathbf{A}^+)$ |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|

(27.6)

In Lemma 27.8, we claimed that  $\mathbf{AA}^+$  and  $\mathbf{A}^+ \mathbf{A}$  are idempotent if  $\mathbf{A}^+$  is a g-inverse of  $\mathbf{A}$ , and thus they are both projection matrices. For  $\mathbf{A}^+$  to be pseudo-inverse, by (C3), (C4) conditions, they are symmetric such that they are **orthogonal** projections as well.

From the pseudo-inverse of the matrix from CR decomposition, we can also claim that any matrix has a pseudo-inverse.

### Lemma 27.19: (Existence of Pseudo-Inverse)

Every matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has a pseudo-inverse.

**Proof** [of Lemma 27.19] For the CR decomposition of  $\mathbf{A} = \mathbf{CR}$ . Let

$$\mathbf{A}^+ = \mathbf{R}^+ \mathbf{C}^+ = \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top,$$

where  $\mathbf{R}^+ = \mathbf{R}^\top (\mathbf{RR}^\top)^{-1}$  and  $\mathbf{C}^+ = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top$ . <sup>2</sup>  $\mathbf{RR}^\top$  and  $\mathbf{C}^\top \mathbf{C}$  are invertible since  $\mathbf{C} \in \mathbb{R}^{m \times r}$  and  $\mathbf{R} \in \mathbb{R}^{r \times n}$  have full rank  $r$  from the property of CR decomposition.

Then, we can check that

$$\begin{aligned}
 (C1) \quad \mathbf{AA}^+ \mathbf{A} &= \mathbf{CR} \left( \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \right) \mathbf{CR} = \mathbf{CR} = \mathbf{A}, \\
 (C2) \quad \mathbf{A}^+ \mathbf{AA}^+ &= \left( \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \right) \mathbf{CR} \left( \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \right) \\
 &= \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top = \mathbf{A}^+, \\
 (C3) \quad (\mathbf{AA}^+)^T &= \mathbf{C} (\mathbf{C}^\top \mathbf{C})^{-1} (\mathbf{RR}^\top)^{-1} \mathbf{RR}^\top \mathbf{C}^\top = \mathbf{C} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \\
 &= \mathbf{C} \mathbf{R} \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top = \mathbf{AA}^+, \\
 (C4) \quad (\mathbf{A}^+ \mathbf{A})^T &= \mathbf{R}^\top \mathbf{C}^\top \mathbf{C} (\mathbf{C}^\top \mathbf{C})^{-1} (\mathbf{RR}^\top)^{-1} \mathbf{R} = \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} \mathbf{R} \\
 &= \mathbf{R}^\top (\mathbf{RR}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{CR} = \mathbf{A}^+ \mathbf{A}.
 \end{aligned}$$

This implies  $\mathbf{A}^+$  is the pseudo-inverse of  $\mathbf{A}$  and therefore proves the existence of the pseudo-inverse. ■

---

<sup>2</sup>. It can be easily checked that  $\mathbf{R}^+$  is the pseudo-inverse of  $\mathbf{R}$  and  $\mathbf{C}^+$  is the pseudo-inverse of  $\mathbf{C}$ .

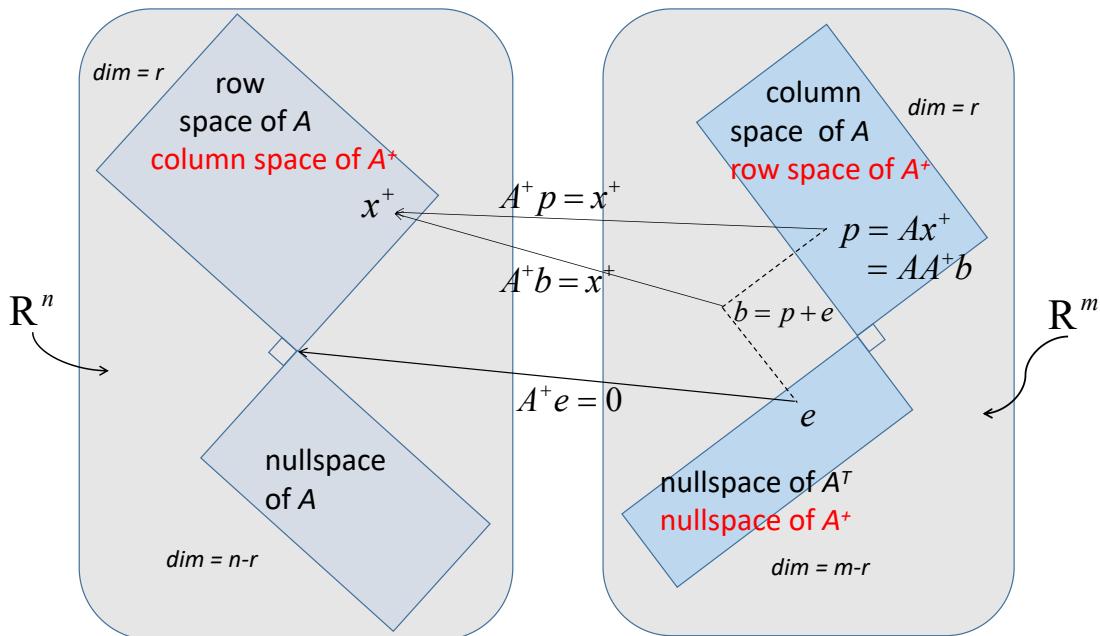
**Lemma 27.20: (Uniqueness of Pseudo-Inverse)**

Every matrix  $A$  has a unique pseudo-inverse.

**Proof** [of Lemma 27.20] Suppose  $A_1^+$  and  $A_2^+$  are two pseudo-inverses of  $A$ . Then

$$\begin{aligned}
 A_1^+ &= A_1^+ A A_1^+ = A_1^+ (A A_2^+ A) A_1^+ = A_1^+ (A A_2^+) (A A_1^+) \quad (\text{by (C2), (C1)}) \\
 &= A_1^+ (A A_2^+)^T (A A_1^+)^T = A_1^+ A_2^{+\top} A^T A_1^{+\top} A^T \quad (\text{by (C3)}) \\
 &= A_1^+ A_2^{+\top} (A A_1^+ A)^T = A_1^+ A_2^{+\top} A^T \quad (\text{by (C1)}) \\
 &= A_1^+ (A A_2^+)^T = A_1^+ A A_2^+ = A_1^+ (A A_2^+ A) A_2^+ \quad (\text{by (C3), (C1)}) \\
 &= (A_1^+ A) (A_2^+ A) A_2^+ = (A_1^+ A)^T (A_2^+ A)^T A_2^+ \quad (\text{by (C4)}) \\
 &= (A A_1^+ A)^T A_2^{+\top} A_2^+ = A^T A_2^{+\top} A_2^+ \quad (\text{by (C1)}) \\
 &= (A_2^+ A)^T A_2^+ = A_2^+ A A_2^+ = A_2^+, \quad (\text{by (C4), (C2)})
 \end{aligned}$$

which implies the uniqueness of pseudo-inverse. ■



**Figure 27.6:** Column space and row space of pseudo-inverse  $A^+$ .  $A$  transfers from row space to column space.  $A^+$  transfers from column space to row space. The split of  $b$  into  $p+e$  and the transformation to  $x^+$  are discussed in Section E.5. This is a more detailed picture of the pseudo-inverse compared to Figure 14.3.

We are now ready to show the four subspaces in pseudo-inverse.

**Lemma 27.21: (Four Subspaces in Pseudo-Inverse)**

For the pseudo-inverse  $\mathbf{A}^+$  of  $\mathbf{A}$ , we have the following properties:

- The column space of  $\mathbf{A}^+$  is the same as the row space of  $\mathbf{A}$ ;
- The row space of  $\mathbf{A}^+$  is the same as the column space of  $\mathbf{A}$ ;
- The null space of  $\mathbf{A}^+$  is the same as the null space of  $\mathbf{A}^\top$ ;
- The null space of  $\mathbf{A}^{+\top}$  is the same as the null space of  $\mathbf{A}$ .

The relationship of the four subspaces is shown in Figure 27.6.

**Proof** [of Lemma 27.21] Since  $\mathbf{A}^+$  is a special rg-inverse, by Lemma 27.18, we have

$$\begin{aligned}\mathcal{C}(\mathbf{AA}^+) &= \mathcal{C}(\mathbf{A}) & \text{and} & \quad \mathcal{N}(\mathbf{A}^+\mathbf{A}) = \mathcal{N}(\mathbf{A}) \\ \mathcal{C}(\mathbf{A}^+\mathbf{A}) &= \mathcal{C}(\mathbf{A}^+) & \text{and} & \quad \mathcal{N}(\mathbf{AA}^+) = \mathcal{N}(\mathbf{A}^+).\end{aligned}$$

By (C3) and (C4) of the definition of pseudo-inverse, we also have

$$(\mathbf{A}^+\mathbf{A})^\top = \mathbf{A}^+\mathbf{A} \quad \text{and} \quad (\mathbf{AA}^+)^\top = \mathbf{AA}^+.$$

By the fundamental theorem of linear algebra, we realize that  $\mathcal{C}(\mathbf{AA}^+)$  is the orthogonal complement to  $\mathcal{N}((\mathbf{AA}^+)^\top)$ , and  $\mathcal{C}(\mathbf{A}^+\mathbf{A})$  is the orthogonal complement to  $\mathcal{N}((\mathbf{A}^+\mathbf{A})^\top)$ :

$$\begin{array}{lll}\mathcal{C}(\mathbf{AA}^+) \perp \mathcal{N}((\mathbf{AA}^+)^\top) & \xrightarrow{\text{leads to}} & \mathcal{C}(\mathbf{AA}^+) \perp \mathcal{N}(\mathbf{AA}^+) \\ \mathcal{C}(\mathbf{A}^+\mathbf{A}) \perp \mathcal{N}((\mathbf{A}^+\mathbf{A})^\top) & \xrightarrow{\text{leads to}} & \mathcal{C}(\mathbf{A}^+\mathbf{A}) \perp \mathcal{N}(\mathbf{A}^+\mathbf{A}).\end{array}$$

This implies

$$\mathcal{C}(\mathbf{A}) \perp \mathcal{N}(\mathbf{A}^+) \quad \text{and} \quad \mathcal{C}(\mathbf{A}^+) \perp \mathcal{N}(\mathbf{A}).$$

That is,  $\mathcal{N}(\mathbf{A}^+) = \mathcal{N}(\mathbf{A}^\top)$  and  $\mathcal{C}(\mathbf{A}^+) = \mathcal{C}(\mathbf{A}^\top)$ . By the fundamental theorem of linear algebra, this also implies  $\mathcal{C}(\mathbf{A}^{+\top}) = \mathcal{C}(\mathbf{A})$  and  $\mathcal{N}(\mathbf{A}^{+\top}) = \mathcal{N}(\mathbf{A})$ . And we complete the proof.  $\blacksquare$

To conclude, we compare the properties for different inverses of  $\mathbf{A}$  in Table 27.1.

|           | g-inverse                                                                                                                                  | rg-inverse                                                                                                                                                                                                                                                             | pseudo-inverse                                                                                                                                                                                                                                                                                                                                                                            |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| subspaces | $\mathcal{C}(\mathbf{AA}_r^-) = \mathcal{C}(\mathbf{A})$<br>$\mathcal{N}(\mathbf{A}_r^-\mathbf{A}) = \mathcal{N}(\mathbf{A})$              | $\mathcal{C}(\mathbf{AA}_r^-) = \mathcal{C}(\mathbf{A})$<br>$\mathcal{N}(\mathbf{A}_r^-\mathbf{A}) = \mathcal{N}(\mathbf{A})$<br>$\mathcal{C}(\mathbf{A}_r^-\mathbf{A}) = \mathcal{C}(\mathbf{A}_r^-)$<br>$\mathcal{N}(\mathbf{AA}_r^-) = \mathcal{N}(\mathbf{A}_r^-)$ | $\mathcal{C}(\mathbf{AA}^+) = \mathcal{C}(\mathbf{A}) = \mathcal{C}(\mathbf{A}^\top)$<br>$\mathcal{N}(\mathbf{A}^+\mathbf{A}) = \mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^{+\top})$<br>$\mathcal{C}(\mathbf{A}^+\mathbf{A}) = \mathcal{C}(\mathbf{A}^+) = \mathcal{C}(\mathbf{A}^\top)$<br>$\mathcal{N}(\mathbf{AA}^+) = \mathcal{N}(\mathbf{A}^+) = \mathcal{N}(\mathbf{A}^\top)$ |
| rank      | $\text{rank}(\mathbf{AA}^-)$<br>$= \text{rank}(\mathbf{A}^-\mathbf{A})$<br>$= \text{rank}(\mathbf{A})$<br>$\leq \text{rank}(\mathbf{A}^-)$ | $\text{rank}(\mathbf{A}_r^-)$<br>$= \text{rank}(\mathbf{A})$<br>$= \text{rank}(\mathbf{AA}_r^-)$<br>$= \text{rank}(\mathbf{A}_r^-\mathbf{A})$                                                                                                                          | $\text{rank}(\mathbf{A}^+)$<br>$= \text{rank}(\mathbf{A})$<br>$= \text{rank}(\mathbf{AA}^+)$<br>$= \text{rank}(\mathbf{A}^+\mathbf{A})$                                                                                                                                                                                                                                                   |

**Table 27.1:** Comparison of different inverses

**Lemma 27.22: (Projection onto Column Space and Row Space)**

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and its pseudo-inverse  $\mathbf{A}^+ \in \mathbb{R}^{n \times m}$ ,  $\mathbf{H} = \mathbf{A}\mathbf{A}^+$  is the orthogonal projection onto column space of  $\mathbf{A}$ . Similarly,  $\mathbf{P} = \mathbf{A}^+\mathbf{A}$  is the orthogonal projection onto row space of  $\mathbf{A}$ .

**Proof** [of Lemma 27.22] As  $\mathbf{H}^\top = (\mathbf{A}\mathbf{A}^+)^\top = \mathbf{A}\mathbf{A}^+ = \mathbf{H}$  from the definition of the pseudo-inverse, and  $\mathbf{H}$  is idempotent such that  $\mathbf{H}$  is an orthogonal projection. From Table 27.1, we conclude that  $\mathcal{C}(\mathbf{H}) = \mathcal{C}(\mathbf{A}\mathbf{A}^+) = \mathcal{C}(\mathbf{A})$ . This implies  $\mathbf{H}$  is the orthogonal projection onto the column space of  $\mathbf{A}$ . Similarly, we can prove  $\mathbf{P} = \mathbf{A}^+\mathbf{A}$  is the orthogonal projection onto row space of  $\mathbf{A}$ . ■

**Pseudo-Inverse in Different Cases**

Specifically, we define in either of the following ways:

- Case  $m > n = r$ , i.e., matrix  $\mathbf{A}$  has linearly independent columns:  
 $\mathbf{A}^\top \mathbf{A}$  is an  $n \times n$  invertible matrix, and we define the left-pseudo-inverse:

$$\boxed{\text{left-pseudo-inverse} = \mathbf{A}^+ = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top},$$

which satisfies

$$\mathbf{A}^+ \mathbf{A} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{A} = \mathbf{I}_n.$$

But

$$\mathbf{A} \mathbf{A}^+ = \mathbf{A} (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \neq \mathbf{I}.$$

- Case  $n > m = r$ , i.e., matrix  $\mathbf{A}$  has linearly independent rows:  
 $\mathbf{A} \mathbf{A}^\top$  is an  $m \times m$  invertible matrix, and we define the right-pseudo-inverse:

$$\boxed{\text{right-pseudo-inverse} = \mathbf{A}^+ = \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1}},$$

which satisfies

$$\mathbf{A} \mathbf{A}^+ = \mathbf{A} \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1} = \mathbf{I}_m.$$

But

$$\mathbf{A}^+ \mathbf{A} = \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1} \mathbf{A} \neq \mathbf{I}. \quad (27.7)$$

- Case rank-deficient: we delay the pseudo-inverse for rank-deficient matrices in the next section.

We can also show that  $(\mathbf{A}^+)^+ = \mathbf{A}$ . If  $m > n = r$ , we have

$$\begin{aligned}
 (\mathbf{A}^+)^+ &= [(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top]^+ \\
 &= \mathbf{A}^{+\top} (\mathbf{A}^+ \mathbf{A}^{+\top})^{-1} \\
 &= \left[ (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \right]^\top \left\{ \left[ (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \right] \left[ (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \right]^\top \right\}^{-1} \\
 &= \mathbf{A} (\mathbf{A}^\top \mathbf{A})^{-1} \left\{ (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{A} (\mathbf{A}^\top \mathbf{A})^{-1} \right\}^{-1} \\
 &= \mathbf{A}.
 \end{aligned} \tag{m > n = r}$$

Similarly, we can show  $(\mathbf{A}^+)^+ = \mathbf{A}$  if  $n > m = r$ .

In particular, when  $m = n$ ,  $\mathbf{A}$  is a square invertible matrix, then both left and right-pseudo-inverse are the inverse of  $\mathbf{A}$ :

$$\begin{aligned}
 \text{left-pseudo-inverse} &= \mathbf{A}^+ = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top = \mathbf{A}^{-1} \mathbf{A}^{-\top} \mathbf{A}^\top = \mathbf{A}^{-1}, \\
 \text{right-pseudo-inverse} &= \mathbf{A}^+ = \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1} = \mathbf{A}^\top \mathbf{A}^{-\top} \mathbf{A}^{-1} = \mathbf{A}^{-1}.
 \end{aligned}$$

## E.5 Pseudo-Inverse in SVD

### Pseudo-Inverse in Different Cases via SVD

For full SVD of matrix  $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^\top$ . Consider the following cases:

- Case  $m > n = r$ :

$$\begin{aligned}
 \text{left-pseudo-inverse} &= \mathbf{A}^+ = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \\
 &= (\mathbf{V} \Sigma^\top \mathbf{U}^\top \mathbf{U} \Sigma \mathbf{V}^\top)^{-1} \mathbf{V} \Sigma^\top \mathbf{U}^\top \\
 &= \mathbf{V} (\Sigma^\top \Sigma)^{-1} \mathbf{V}^\top \mathbf{V} \Sigma^\top \mathbf{U}^\top \\
 &= \mathbf{V} [(\Sigma^\top \Sigma)^{-1} \Sigma^\top] \mathbf{U}^\top \\
 &= \mathbf{V} \Sigma^+ \mathbf{U}^\top. \tag{\Sigma^+ = (\Sigma^\top \Sigma)^{-1} \Sigma^\top}
 \end{aligned}$$

- Case  $n > m = r$ :

$$\begin{aligned}
 \text{right-pseudo-inverse} &= \mathbf{A}^+ = \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1} \\
 &= (\mathbf{U} \Sigma \mathbf{V}^\top)^\top [(\mathbf{U} \Sigma \mathbf{V}^\top) (\mathbf{U} \Sigma \mathbf{V}^\top)^\top]^{-1} \\
 &= \mathbf{V} \Sigma^\top \mathbf{U}^\top (\mathbf{U} \Sigma \mathbf{V}^\top \mathbf{V} \Sigma^\top \mathbf{U}^\top)^{-1} \\
 &= \mathbf{V} \Sigma^\top \mathbf{U}^\top \mathbf{U}^{-\top} (\Sigma \Sigma^\top)^{-1} \mathbf{U}^{-1} \\
 &= \mathbf{V} \Sigma^\top (\Sigma \Sigma^\top)^{-1} \mathbf{U}^{-1} \\
 &= \mathbf{V} \Sigma^+ \mathbf{U}^\top. \tag{\Sigma^+ = \Sigma^\top (\Sigma \Sigma^\top)^{-1}}
 \end{aligned}$$

- Case rank-deficient:  $\mathbf{A}^+ = \mathbf{V} \Sigma^+ \mathbf{U}^\top$ , where the upper-left side of  $\Sigma^+ \in \mathbb{R}^{n \times m}$  is a diagonal matrix  $\text{diag}(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r})$ . It can be easily verified that this definition of pseudo-inverse satisfies the four criteria in Equation (27.6).

In either case, we have  $\Sigma^+$  as the pseudo-inverse of  $\Sigma$  with  $1/\sigma_1, 1/\sigma_2, \dots, 1/\sigma_r$  on its diagonal. We thus conclude the pseudo-inverse from SVD in Table 27.2.

|     | $\mathbf{A}$                      | $\mathbf{A}^\top$                 | $\mathbf{A}^+$                      | $\mathbf{A}^{+\top}$                |
|-----|-----------------------------------|-----------------------------------|-------------------------------------|-------------------------------------|
| SVD | $\mathbf{U}\Sigma\mathbf{V}^\top$ | $\mathbf{V}\Sigma\mathbf{U}^\top$ | $\mathbf{V}\Sigma^+\mathbf{U}^\top$ | $\mathbf{U}\Sigma^+\mathbf{V}^\top$ |

**Table 27.2:** Pseudo-inverse in SVD

From the pseudo-inverse via the SVD, we can provide another way to see the orthogonal projection in pseudo-inverse.

### Another Way to See the Orthogonal Projection in Pseudo-Inverse via SVD

We have shown previously that  $\mathbf{H} = \mathbf{AA}^+$  is an orthogonal projection, so we only need to show that it projects onto  $\mathcal{C}(\mathbf{A})$ . For any vector  $\mathbf{b} \in \mathbb{R}^m$ , we have

$$\mathbf{Hb} = \mathbf{AA}^+\mathbf{b} = \mathbf{Ax}^+,$$

which is a linear combination of columns of  $\mathbf{A}$ . Thus  $\mathcal{C}(\mathbf{H}) \subseteq \mathcal{C}(\mathbf{A})$ .

Moreover, since  $\mathbf{H}$  is an orthogonal projection, by Lemma 27.4, we have  $\text{rank}(\mathbf{H}) = \text{trace}(\mathbf{H}) = \text{trace}(\mathbf{AA}^+) = \text{trace}(\mathbf{U}\Sigma\mathbf{V}^\top\mathbf{V}\Sigma^+\mathbf{U}^\top) = \text{trace}(\mathbf{U}\Sigma\Sigma^+\mathbf{U}^\top) = r$ , where  $\mathbf{U}\Sigma\mathbf{V}^\top$  is the SVD of  $\mathbf{A}$ . That is the rank of  $\mathbf{H}$  equals the rank of  $\mathbf{A}$ . This implies  $\mathcal{C}(\mathbf{H}) = \mathcal{C}(\mathbf{A})$  and we complete the proof.

Similarly, we can prove  $\mathbf{P} = \mathbf{A}^+\mathbf{A}$  is the orthogonal projection onto the row space of  $\mathbf{A}$  via the pseudo-inverse from SVD.

We finally provide another proof of the important property of the four subspaces in the pseudo-inverse  $\mathbf{A}^+$  via the SVD as well. Firstly, we need to show the following lemma that  $\mathbf{A}^+\mathbf{A}^{+\top}$  and  $\mathbf{A}^+$  have the same rank.

#### Lemma 27.23: (Rank of $\mathbf{A}^+\mathbf{A}^{+\top}$ )

$\mathbf{A}^+\mathbf{A}^{+\top}$  and  $\mathbf{A}^+$  have same rank.

**Proof** [of Lemma 27.23] Let  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^{+\top})$ , where  $\mathbf{A}^{+\top}$  is the transpose of  $\mathbf{A}^+$ , we have

$$\mathbf{A}^{+\top}\mathbf{x} = \mathbf{0} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}^+\mathbf{A}^{+\top}\mathbf{x} = \mathbf{0},$$

i.e.,  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^{+\top})$  leads to  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^+\mathbf{A}^{+\top})$ , therefore  $\mathcal{N}(\mathbf{A}^{+\top}) \in \mathcal{N}(\mathbf{A}^+\mathbf{A}^{+\top})$ .

Further, let  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^+\mathbf{A}^{+\top})$ , we have

$$\mathbf{A}^+\mathbf{A}^{+\top}\mathbf{x} = \mathbf{0} \xrightarrow{\text{leads to}} \mathbf{x}^\top\mathbf{A}^+\mathbf{A}^{+\top}\mathbf{x} = 0 \xrightarrow{\text{leads to}} \|\mathbf{A}^{+\top}\mathbf{x}\|^2 = 0 \xrightarrow{\text{leads to}} \mathbf{A}^{+\top}\mathbf{x} = \mathbf{0},$$

i.e.,  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^+\mathbf{A}^{+\top})$  leads to  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^{+\top}\mathbf{x})$ , therefore  $\mathcal{N}(\mathbf{A}^+\mathbf{A}^{+\top}) \in \mathcal{N}(\mathbf{A}^{+\top})$ . As a result,  $\mathcal{N}(\mathbf{A}^{+\top}) = \mathcal{N}(\mathbf{A}^+\mathbf{A}^{+\top})$  and  $\dim(\mathcal{N}(\mathbf{A}^{+\top})) = \dim(\mathcal{N}(\mathbf{A}^+\mathbf{A}^{+\top}))$ . By the fundamental theorem of linear algebra in Appendix B,  $\mathbf{A}^+\mathbf{A}^{+\top}$  and  $\mathbf{A}^+$  have the same rank. ■

By the lemma above, we can provide another way to show the four subspaces in pseudo-inverse.

### Another Way to See the Subspaces in Pseudo-Inverse via the SVD

Let the SVD of  $\mathbf{A}$  be  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , and its pseudo-inverse  $\mathbf{A}^+ = \mathbf{V}\Sigma^+\mathbf{U}^\top$ . From Lemma 13.7, for symmetric matrix  $\mathbf{A}^+\mathbf{A}^{+\top} = \mathbf{V}(\Sigma^\top\Sigma)^{-1}\mathbf{V}^\top$  (it can be seen as a spectral decomposition of  $\mathbf{A}^+\mathbf{A}^{+\top}$  and  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  are the eigenvectors of  $\mathbf{A}^+\mathbf{A}^{+\top}$ ),  $\mathcal{C}(\mathbf{A}^+\mathbf{A}^{+\top})$  is spanned by the eigenvectors, thus  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A}^+\mathbf{A}^{+\top})$ .

Since,

1.  $\mathbf{A}^+\mathbf{A}^{+\top}$  is symmetric, then the row space of  $\mathbf{A}^+\mathbf{A}^{+\top}$  equals the column space of  $\mathbf{A}^+\mathbf{A}^{+\top}$ .

2. All columns of  $\mathbf{A}^+\mathbf{A}^{+\top}$  are combination of columns of  $\mathbf{A}^+$ , so the column space of  $\mathbf{A}^+\mathbf{A}^{+\top} \subseteq$  the column space of  $\mathbf{A}^+$ , i.e.,  $\mathcal{C}(\mathbf{A}^+\mathbf{A}^{+\top}) \subseteq \mathcal{C}(\mathbf{A}^+)$ .

3. Since  $\text{rank}(\mathbf{A}^+\mathbf{A}^{+\top}) = \text{rank}(\mathbf{A}^+)$  by Lemma 27.23, we then have

The row space of  $\mathbf{A}^+\mathbf{A}^{+\top} =$  the column space of  $\mathbf{A}^+\mathbf{A}^{+\top} =$  the column space of  $\mathbf{A}^+$ , i.e.,  $\mathcal{C}(\mathbf{A}^+\mathbf{A}^{+\top}) = \mathcal{C}(\mathbf{A}^+)$ . Thus  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of  $\mathcal{C}(\mathbf{A}^+)$ .

We also proved in Lemma 14.1 that  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis of the row space of  $\mathbf{A}$  (i.e., basis of  $\mathcal{C}(\mathbf{A}^\top)$ ). So  $\mathcal{C}(\mathbf{A}^+) = \mathcal{C}(\mathbf{A}^\top)$  as shown in Figure 27.6.

Similarly, if we apply this process to  $\mathbf{A}^{+\top}\mathbf{A}^+$ , we can show the row space of  $\mathbf{A}^+$  is equal to the column space of  $\mathbf{A}$ , and the null space of  $\mathbf{A}^+$  is equal to the null space of  $\mathbf{A}^\top$ .

**1).**  $\mathbf{A}^+\mathbf{p} = \mathbf{x}^+$  where  $\mathbf{p} \in \mathcal{C}(\mathbf{A})$  and  $\mathbf{x}^+ \in \mathcal{C}(\mathbf{A}^\top)$ : Moreover, for  $\mathbf{x}^+$  in row space of  $\mathbf{A}$ , we have  $\mathbf{x}^+ = \sum_{i=1}^r x_i \mathbf{v}_i$  since  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  is an orthonormal basis for the row space of  $\mathbf{A}$  (Lemma 14.1, p. 269). For vector  $\mathbf{p} = \mathbf{Ax}^+$  in the column space of  $\mathbf{A}$ , we have  $\mathbf{p} = \mathbf{U}\Sigma\mathbf{V}^\top \mathbf{x}^+$  and

$$\mathbf{A}^+\mathbf{p} = \mathbf{V}\Sigma^+\mathbf{U}^\top \mathbf{U}\Sigma\mathbf{V}^\top \mathbf{x}^+ = (\mathbf{V}\Sigma^+\Sigma\mathbf{V}^\top)(\mathbf{x}^+) = \left(\sum_{i=1}^r \mathbf{v}_i \mathbf{v}_i^\top\right) \left(\sum_{i=1}^r x_i \mathbf{v}_i\right) = \sum_{i=1}^r x_i \mathbf{v}_i = \mathbf{x}^+.$$

**2).**  $\mathbf{A}^+\mathbf{b} = \mathbf{x}^+$  where  $\mathbf{p} \in \mathbb{R}^m$  and  $\mathbf{x}^+ \in \mathcal{C}(\mathbf{A}^\top)$ : For vector  $\mathbf{e}$  in the null space of  $\mathbf{A}^\top$ , we have  $\mathbf{A}^+\mathbf{e} = \mathbf{0}$  since  $\mathcal{N}(\mathbf{A}^+) = \mathcal{N}(\mathbf{A}^\top)$ . Any vector  $\mathbf{b} \in \mathbb{R}^m$  can be split into  $\mathbf{b} = \mathbf{p} + \mathbf{e}$ , where  $\mathbf{p}$  is a vector in the column space of  $\mathbf{A}$  and  $\mathbf{e}$  is a vector in the null space of  $\mathbf{A}^\top$ . That is

$$\mathbf{A}^+\mathbf{p} = \mathbf{A}^+\mathbf{b} = \mathbf{x}^+,$$

where  $\mathbf{x}^+$  is in the row space of  $\mathbf{A}$ .

In conclude, for any vector  $\mathbf{x}^+$  in row space of  $\mathbf{A}$  and there exists a vector  $\mathbf{p}$  in the column space of  $\mathbf{A}$ , we have

$$\mathbf{p} = \mathbf{Ax}^+ \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}^+\mathbf{p} = \mathbf{x}^+,$$

and the relationship is depicted in Figure 27.6.

### E.6 Pseudo-Inverse in CR Decomposition and Skeleton Decomposition

Suppose the CR decomposition of matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is given by  $\mathbf{A} = \mathbf{CR}$  where  $\mathbf{C} \in \mathbb{R}^{m \times r}$  and  $\mathbf{R} \in \mathbb{R}^{r \times n}$ . And since in the CR decomposition, we have the row number in  $\mathbf{C}$  being

larger or equal than the column number, and the row number in  $\mathbf{R}$  being smaller or equal than the column number:  $r \leq \min\{m, n\}$ . Then we have the pseudo-inverse of  $\mathbf{C}$  (left-pseudo-inverse) and  $\mathbf{R}$  (right-pseudo-inverse):

$$\mathbf{C}^+ = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \quad \text{and} \quad \mathbf{R}^+ = \mathbf{R}^\top (\mathbf{R} \mathbf{R}^\top)^{-1}$$

Consider the following two cases:

- Case  $m > n = r$ :

$$\begin{aligned} \text{left-pseudo-inverse} &= \mathbf{A}^+ = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \\ &= (\mathbf{R}^\top \mathbf{C}^\top \mathbf{C} \mathbf{R})^{-1} \mathbf{R}^\top \mathbf{C}^\top. \end{aligned}$$

We then have  $(\mathbf{R}^\top \mathbf{C}^\top \mathbf{C} \mathbf{R}) \mathbf{R}^+ \mathbf{C}^+ = (\mathbf{R}^\top \mathbf{C}^\top \mathbf{C} \mathbf{R}) \mathbf{R}^\top (\mathbf{R} \mathbf{R}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top = \mathbf{R}^\top \mathbf{C}^\top$ . Thus

$$\mathbf{A}^+ = \mathbf{R}^+ \mathbf{C}^+.$$

Moreover, since  $m > n$ , we have  $\mathbf{A}^+ \mathbf{A} = \mathbf{I}_n$ , and

$$\begin{aligned} \mathbf{A}^+ \mathbf{A} &= \mathbf{R}^+ \mathbf{C}^+ \mathbf{C} \mathbf{R} \\ &= \mathbf{R}^\top (\mathbf{R} \mathbf{R}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{C} \mathbf{R} \\ &= \mathbf{R}^\top (\mathbf{R} \mathbf{R}^\top)^{-1} \mathbf{R} \\ &= \mathbf{R}^+ \mathbf{R} = \mathbf{I}_n, \quad (\text{Since } \mathbf{A}^+ \mathbf{A} = \mathbf{I}_n) \end{aligned} \tag{27.8}$$

which is quite confusing since we claimed in Equation (27.7) that  $\mathbf{R}^+ \mathbf{R} \neq \mathbf{I}_n$ . However, this is not true. We assume  $\mathbf{A}$  has full rank, and  $m > n$  which implies  $\mathbf{R}$  is a square invertible matrix. So  $\mathbf{R}^+ = \mathbf{R}^{-1}$  and the result in Equation (27.8) cannot be applied to any right-pseudo-inverse of other matrices.

- Case  $n > m = r$ :

$$\begin{aligned} \text{right-pseudo-inverse} &= \mathbf{A}^+ = \mathbf{A}^\top (\mathbf{A} \mathbf{A}^\top)^{-1} \\ &= \mathbf{R}^\top \mathbf{C}^\top (\mathbf{C} \mathbf{R} \mathbf{R}^\top \mathbf{C}^\top)^{-1}. \end{aligned}$$

Again, we have  $\mathbf{R}^+ \mathbf{C}^+ (\mathbf{C} \mathbf{R} \mathbf{R}^\top \mathbf{C}^\top) = \mathbf{R}^\top (\mathbf{R} \mathbf{R}^\top)^{-1} (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \mathbf{C} \mathbf{R} \mathbf{R}^\top \mathbf{C}^\top = \mathbf{R}^\top \mathbf{C}^\top$ , and

$$\mathbf{A}^+ = \mathbf{R}^+ \mathbf{C}^+.$$

We can then conclude for any full rank matrix  $\mathbf{A}$  (either  $m > n$  or  $n > m$ ), the pseudo-inverse of  $\mathbf{A}$  is

$$\mathbf{A}^+ = \mathbf{R}^+ \mathbf{C}^+.$$

The pseudo-inverse of  $\mathbf{A}$  for skeleton decomposition is defined in a similar way. For skeleton decomposition of  $\mathbf{A} = \mathbf{C} \mathbf{U}^{-1} \mathbf{R}$ , then the pseudo-inverse of  $\mathbf{A}$  is  $\mathbf{A}^+ = \mathbf{R}^+ \mathbf{U} \mathbf{C}^+$ , where

$$\mathbf{C}^+ = (\mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top \quad \text{and} \quad \mathbf{R}^+ = \mathbf{R}^\top (\mathbf{R} \mathbf{R}^\top)^{-1}$$

## Appendix F. Schur Complement

Let  $\mathbf{M}$  be an  $n \times n$  square matrix with  $2 \times 2$  block matrix format:

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix},$$

where  $\mathbf{A}$  is a  $p \times p$  matrix and  $\mathbf{D}$  is a  $q \times q$  matrix with  $n = p + q$ . And it is trivial to see that  $\mathbf{B}$  is a  $p \times q$  matrix and  $\mathbf{C}$  is a  $q \times p$  matrix. We have

- If  $\mathbf{D}$  is invertible,  $\Delta_{\mathbf{D}} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$  is called the Schur complement of  $\mathbf{D}$  in  $\mathbf{M}$ ;
- If  $\mathbf{A}$  is invertible,  $\Delta_{\mathbf{A}} = \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$  is called the Schur complement of  $\mathbf{A}$  in  $\mathbf{M}$ .

By mimicking Gaussian elimination in the block matrix, we can lower triangularize and upper triangularize  $\mathbf{M}$  as follows:

$$\begin{aligned} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{0} & \Delta_{\mathbf{A}} \end{bmatrix} \\ \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} & \Delta_{\mathbf{A}} \end{bmatrix}. \end{aligned}$$

Then we can multiply  $\mathbf{M}$  from left by the lower triangular matrix and from right by the upper triangular matrix:

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathbf{A}} \end{bmatrix}$$

or

$$\begin{aligned} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathbf{A}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}^{-1} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \mathbf{M} \\ \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathbf{A}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \mathbf{M}, \end{aligned}$$

which is a multiplication of an upper triangular, a diagonal matrix, and a lower triangular matrix. And in the above equation, we use the fact that

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} = \mathbf{I} \quad \text{and} \quad \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = \mathbf{I}.$$

It is then easy to compute the inverse of  $\mathbf{M}$  since the inverse of the upper triangular, the diagonal matrix, and the lower triangular matrix is relatively easy. If we assume  $\Delta_{\mathbf{A}}$  is invertible, then:

$$\begin{aligned} \mathbf{M} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathbf{A}} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \\ \mathbf{M}^{-1} &= \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & -\mathbf{A}^{-1}\mathbf{B} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \Delta_{\mathbf{A}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{bmatrix}. \end{aligned}$$

It is then similar to apply this process to the Schur complement of  $\mathbf{D}$  and we formulate in the following remark.

**Remark 27.1**

If  $\mathbf{D}$  is invertible, then we can use the Schur complement of  $\mathbf{D}$ ,  $\Delta_{\mathbf{D}} = \mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$  to obtain the following decomposition of  $\mathbf{M}$ :

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta_{\mathbf{D}} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix},$$

and if  $\Delta_{\mathbf{D}}$  is invertible, the inverse of  $\mathbf{M}$  is

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{D}^{-1}\mathbf{C} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \Delta_{\mathbf{D}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{B}\mathbf{D}^{-1} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

Moreover, write out the two inverse formulas by  $\Delta_{\mathbf{A}}$  and  $\Delta_{\mathbf{D}}$  and let  $\mathbf{D} = \mathbf{I}$ , we will have

$$(\mathbf{A} + \mathbf{B}\mathbf{C})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}(\mathbf{I} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1},$$

which is known as the **matrix inversion lemma** (Boyd et al., 2004; Gallier et al., 2010).

## Appendix G. General Term Formula of Wedderburn Sequence

We define the Wedderburn sequence of  $\mathbf{A}$  by  $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$  and  $\mathbf{A}_1 = \mathbf{A}$ . The proof of the general term formula of this sequence is then:

**Proof** [of Lemma 19.4] For  $\mathbf{A}_2$ , we have:

$$\begin{aligned}\mathbf{A}_2 &= \mathbf{A}_1 - w_1^{-1} \mathbf{A}_1 \mathbf{x}_1 \mathbf{y}_1^\top \mathbf{A}_1 \\ &= \mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}, \text{ where } \mathbf{u}_1 = \mathbf{x}_1, \mathbf{v}_1 = \mathbf{y}_1.\end{aligned}$$

For  $\mathbf{A}_3$ , we can write out the equation:

$$\begin{aligned}\mathbf{A}_3 &= \mathbf{A}_2 - w_2^{-1} \mathbf{A}_2 \mathbf{x}_2 \mathbf{y}_2^\top \mathbf{A}_2 \\ &= (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) \\ &\quad - w_2^{-1} (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) \mathbf{x}_2 \mathbf{y}_2^\top (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) \quad (\text{substitute } \mathbf{A}_2) \\ &= (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) \\ &\quad - w_2^{-1} \mathbf{A} (\mathbf{x}_2 - w_1^{-1} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A} \mathbf{x}_2) (\mathbf{y}_2^\top - w_1^{-1} \mathbf{y}_2^\top \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top) \mathbf{A} \quad (\text{take out } \mathbf{A}) \\ &= \mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A} - w_2^{-1} \mathbf{A} \mathbf{u}_2 \mathbf{v}_2^\top \mathbf{A} \\ &= \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A},\end{aligned}$$

where  $\mathbf{u}_2 = \mathbf{x}_2 - w_1^{-1} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A} \mathbf{x}_2 = \mathbf{x}_2 - \frac{\mathbf{v}_1^\top \mathbf{A} \mathbf{x}_2}{w_1} \mathbf{u}_1$ ,  $\mathbf{v}_2 = \mathbf{y}_2 - w_1^{-1} \mathbf{y}_2^\top \mathbf{A} \mathbf{u}_1 \mathbf{v}_1 = \mathbf{y}_2 - \frac{\mathbf{y}_2^\top \mathbf{A} \mathbf{u}_1}{w_1} \mathbf{v}_1$ . Similarly, we can find the expression of  $\mathbf{A}_4$  by  $\mathbf{A}$ :

$$\begin{aligned}\mathbf{A}_4 &= \mathbf{A}_3 - w_3^{-1} \mathbf{A}_3 \mathbf{x}_3 \mathbf{y}_3^\top \mathbf{A}_3 \\ &= \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} \\ &\quad - w_3^{-1} (\mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}) \mathbf{x}_3 \mathbf{y}_3^\top (\mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}) \quad (\text{substitute } \mathbf{A}_3) \\ &= \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} \\ &\quad - w_3^{-1} \mathbf{A} (\mathbf{x}_3 - \sum_{i=1}^2 w_i^{-1} \mathbf{x}_3 \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}) (\mathbf{y}_3^\top - \sum_{i=1}^2 w_i^{-1} \mathbf{y}_3^\top \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top) \mathbf{A} \quad (\text{take out } \mathbf{A}) \\ &= \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} - w_3^{-1} \mathbf{A} \mathbf{u}_3 \mathbf{v}_3^\top \mathbf{A} \\ &= \mathbf{A} - \sum_{i=1}^3 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A},\end{aligned}$$

where  $\mathbf{u}_3 = \mathbf{x}_3 - \sum_{i=1}^2 \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_3}{w_i} \mathbf{u}_i$ ,  $\mathbf{v}_3 = \mathbf{y}_3 - \sum_{i=1}^2 \frac{\mathbf{y}_i^\top \mathbf{A} \mathbf{u}_i}{w_i} \mathbf{v}_i$ .

Continue this process, we can define

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{w_i} \mathbf{u}_i, \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\mathbf{y}_k^\top \mathbf{A} \mathbf{u}_i}{w_i} \mathbf{v}_i,$$

and find the general term of Wedderburn sequence. ■

## Appendix H. Decoding Orthogonal Matrix Multiplication

### Lemma 27.1: (Orthogonal Matrix Multiplication: A Decoding View)

Suppose an  $(n+1) \times (n+1)$  orthogonal matrix  $\mathbf{Q}$  and an  $n \times n$  orthogonal matrix  $\mathbf{P}$ , then

$$\mathbf{R} = \mathbf{Q} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{P} \end{bmatrix}$$

is also an orthogonal matrix of size  $(n+1) \times (n+1)$ .

It is trivial that the product of two orthogonal matrices is also orthogonal, this appendix provides a decoding view of what happens inside the operation.

**Proof** [of Lemma 27.1] Suppose  $\mathbf{Q} = [\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$  and  $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n]$ . Then

$$\begin{aligned} \mathbf{R} &= [\mathbf{q}_0, \sum_{i=0}^n \mathbf{p}_{1i} \mathbf{q}_i, \sum_{i=0}^n \mathbf{p}_{2i} \mathbf{q}_i, \dots, \sum_{i=0}^n \mathbf{p}_{ni} \mathbf{q}_i] \\ &= [\mathbf{q}_0, \mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n]. \end{aligned}$$

It is trivial that  $\mathbf{q}_0$  is orthogonal to  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_n$ . For  $\mathbf{z}_1$ , we have

$$\mathbf{z}_1^\top \mathbf{z}_1 = \sum_{i=1}^n \mathbf{p}_{1i}^2 = 1.$$

And for  $\mathbf{z}_1, \mathbf{z}_2$ , we have

$$\mathbf{z}_1^\top \mathbf{z}_2 = \sum_{i=1}^n \mathbf{p}_{1i} \mathbf{p}_{2i} = 0,$$

which completes the proof by applying  $\mathbf{z}_i^\top \mathbf{z}_j$  for all  $i, j \in \{1, 2, \dots, n\}$ . ■

## Appendix I. Cochran's Theorem

In this appendix, we provide a proof for Theorem 27.1, the Cochran's theorem (James, 1952; Tan, 1975; Anderson and Styan, 1980; Gut, 2009).

### Theorem 27.1: (Cochran's Theorem)

Let  $\mathbf{y}$  be random variable in  $\mathbb{R}^n$  and  $\mathbf{y}^\top \mathbf{y}$  can be factored into  $k > 0$  terms:

$$\mathbf{y}^\top \mathbf{y} = \mathbf{y}^\top \mathbf{A}_1 \mathbf{y} + \mathbf{y}^\top \mathbf{A}_2 \mathbf{y} + \dots + \mathbf{y}^\top \mathbf{A}_k \mathbf{y}.$$

And  $\mathbf{A}_i$ 's meet the following requirements

- i).  $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_k$  are positive semi-definite (PSD);
- ii).  $\mathbf{A}_1 + \mathbf{A}_2 + \dots + \mathbf{A}_k = \mathbf{I}_n$  is the  $n \times n$  identity matrix;
- iii). Set  $r_i = \text{rank}(\mathbf{A}_i)$ , and  $r_1 + r_2 + \dots + r_k = n$ .

Then, Set  $q_i = \mathbf{y}^\top \mathbf{A}_i \mathbf{y}$ , we have the following results:

1. If  $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ , then  $q_i \sim \sigma^2 \chi^2_{(r_i)}$ .
2. If  $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I})$  and  $\boldsymbol{\mu}^\top \mathbf{A}_i \boldsymbol{\mu} = 0$ , then  $q_i \sim \sigma^2 \chi^2_{(r_i)}$ ;
3.  $q_i$ 's are independent to each other.

To prove the Cochran's theorem, we need the following lemma.

### Lemma 27.2: (Idempotent Decomposition: Rank-Additivity)

For  $n \times n$  square matrices  $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_k$ , and  $\mathbf{A}_1 + \mathbf{A}_2 + \dots + \mathbf{A}_k = \mathbf{I}_n$ , then the following three conditions are equivalent:

- i).  $\mathbf{A}_i^2 = \mathbf{A}_i$ , for all  $i \in \{1, 2, \dots, k\}$ , i.e.,  $\mathbf{A}_i$ 's are idempotent;
- ii).  $\text{rank}(\mathbf{A}_1) + \text{rank}(\mathbf{A}_2) + \dots + \text{rank}(\mathbf{A}_k) = n$ ;
- iii).  $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$  for all  $i \neq j$  and  $i, j \in \{1, 2, \dots, k\}$ .

**Proof** [of Lemma 27.2] From i) to ii), by Lemma 27.4, the trace and rank of any idempotent matrix are the same. Then,

$$\sum_{i=1}^k \text{rank}(\mathbf{A}_i) = \sum_{i=1}^k \text{trace}(\mathbf{A}_i) = \text{trace}(\mathbf{I}_n) = n.$$

From ii) to iii), we have the following block Gaussian elimination for a  $(k+1) \times (k+1)$  block matrix (that is,  $(k+1)n \times (k+1)n$  matrix) where  $(2, 2), (3, 3), \dots, (k+1, k+1)$  blocks

are  $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_k$  respectively.

$$\mathbf{X} = \begin{bmatrix} \mathbf{0}_n & & \\ & \mathbf{A}_1 & \\ & & \ddots & \\ & & & \mathbf{A}_k \end{bmatrix} \xrightarrow{\mathbf{E}_1 \times} \begin{bmatrix} \mathbf{0}_n & \mathbf{A}_1 & \dots & \mathbf{A}_k \\ & \mathbf{A}_1 & & \\ & & \ddots & \\ & & & \mathbf{A}_k \end{bmatrix} \xrightarrow{\mathbf{E}_2 \times} \begin{bmatrix} \mathbf{I}_n & \mathbf{A}_1 & \dots & \mathbf{A}_k \\ \mathbf{A}_1 & \mathbf{A}_1 & & \\ \vdots & & \ddots & \\ \mathbf{A}_k & & & \mathbf{A}_k \end{bmatrix} \xrightarrow{\mathbf{E}_3 \times}$$

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{A}_1 & \dots & \mathbf{A}_k \\ \mathbf{A}_1 - \mathbf{A}_1^2 & -\mathbf{A}_1 \mathbf{A}_k & & \\ \vdots & \ddots & \vdots & \\ -\mathbf{A}_k \mathbf{A}_1 & \mathbf{A}_k - \mathbf{A}_k^2 & & \end{bmatrix} \xrightarrow{\mathbf{E}_4 \times} \begin{bmatrix} \mathbf{I}_n & \mathbf{A}_1 - \mathbf{A}_1^2 & & -\mathbf{A}_1 \mathbf{A}_k \\ \vdots & \ddots & \vdots & \\ -\mathbf{A}_k \mathbf{A}_1 & \mathbf{A}_k - \mathbf{A}_k^2 & & \end{bmatrix} = \mathbf{Y},$$

where black entries indicate zeros. And

- $\mathbf{E}_1$  is adding row-2, row-3, ..., row-( $k+1$ ) to row-1;
- $\mathbf{E}_2$  is adding column-2, column-3, ..., column-( $k+1$ ) to column-1;
- $\mathbf{E}_3$  is subtracting the row-2 by  $\mathbf{A}_1^*(\text{row-1})$ , subtracting the row-3 by  $\mathbf{A}_2^*(\text{row-1})$ , ...;
- $\mathbf{E}_4$  is subtracting the column-2 by  $\mathbf{A}_1^*(\text{column-1})$ , subtracting column-3 by  $\mathbf{A}_2^*(\text{column-1})$ , ...

We notice that elementary operations/transformations will not change the rank of the matrix. Since  $\mathbf{X}$  is of rank  $\sum_{i=1}^k r_k = n$ , and  $\mathbf{I}_n$  in  $\mathbf{Y}$  is of rank- $n$  as well. We must have

$$\begin{bmatrix} \mathbf{A}_1 - \mathbf{A}_1^2 & -\mathbf{A}_1 \mathbf{A}_k \\ \vdots & \ddots & \vdots \\ -\mathbf{A}_k \mathbf{A}_1 & \mathbf{A}_k - \mathbf{A}_k^2 \end{bmatrix} = \mathbf{0},$$

which implies  $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$  for all  $i \neq j$  and  $i, j \in \{1, 2, \dots, k\}$ .

From iii) to i). we have

$$\begin{aligned} \mathbf{A}_i &= \mathbf{A}_i \mathbf{I}_n \\ &= \mathbf{A}_i (\mathbf{A}_1 + \mathbf{A}_2 + \dots + \mathbf{A}_k) \\ &= \mathbf{A}_i^2. \end{aligned}$$

This completes the proof. ■

**Note on nomenclature:** we say that the  $\mathbf{A}_i$ 's are **orthogonal** if iii) of Lemma 27.2 is satisfied and the **rank is additive** if ii) is satisfied.

Now we are ready to prove the Cochran's Theorem as follows:

**Proof [of Theorem 27.1]** From Lemma 27.2,  $\mathbf{A}_i$ 's are idempotent.

**Case 1,** If  $\mathbf{y} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ :

By Spectral Theorem 13.1 and Lemma 27.2 (the only possible eigenvalues of idempotent matrices are 0 and 1), we can rewrite the  $q_i$  by  $q_i = \mathbf{y}^\top \mathbf{A}_i \mathbf{y} = \mathbf{y}^\top (\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top) \mathbf{y}$ , where  $\mathbf{A}_i = \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top$  is the spectral decomposition of  $\mathbf{A}_i$ . From rotations on the normal distribution do not effect the distribution<sup>3</sup>, we can define

$$\boldsymbol{\eta} = \mathbf{Q}^\top \mathbf{y} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}).$$

---

3. Rotations on the Gaussian distribution do not effect the distribution. That is for any orthogonal matrix  $\mathbf{Q}$  with  $\mathbf{Q}\mathbf{Q}^\top = \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$ , if  $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ , then  $\mathbf{Q}\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$ .

Thus,

$$q_i = \boldsymbol{\eta}^\top \boldsymbol{\Lambda} \boldsymbol{\eta} \sim \sigma^2 \chi_{\text{rank}(\mathbf{A}_i)}^2 \sim \sigma^2 \chi_{(r_i)}^2,$$

**Case 2:** If  $\mathbf{y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I})$ , and  $\boldsymbol{\mu}^\top \mathbf{A}_i \boldsymbol{\mu} = 0$ :

Let  $p_i = (\mathbf{y} - \boldsymbol{\mu})^\top \mathbf{A}_i (\mathbf{y} - \boldsymbol{\mu})$ , similarly, we can rewrite the  $p_i$  by  $p_i = (\mathbf{y} - \boldsymbol{\mu})^\top \mathbf{A}_i (\mathbf{y} - \boldsymbol{\mu}) = (\mathbf{y} - \boldsymbol{\mu})^\top (\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top) (\mathbf{y} - \boldsymbol{\mu})$ , where  $\mathbf{A}_i = \mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top$  is the spectral decomposition of  $\mathbf{A}_i$ . From the fact that rotations on the normal distribution do not effect the distribution, we can define

$$\boldsymbol{\eta} = \mathbf{Q}^\top (\mathbf{y} - \boldsymbol{\mu}) \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}).$$

Thus,

$$p_i = \boldsymbol{\eta}^\top \boldsymbol{\Lambda} \boldsymbol{\eta} \sim \sigma^2 \chi_{\text{rank}(\mathbf{A}_i)}^2 \sim \sigma^2 \chi_{(r_i)}^2.$$

Then, we decompose the  $p_i$  into

$$\begin{aligned} p_i &= \mathbf{y}^\top \mathbf{A}_i \mathbf{y} - 2\mathbf{y}^\top \mathbf{A}_i \boldsymbol{\mu} + \boldsymbol{\mu}^\top \mathbf{A}_i \boldsymbol{\mu} \\ &= q_i - 2\mathbf{y}^\top (\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top) \boldsymbol{\mu} + \boldsymbol{\mu}^\top (\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top) \boldsymbol{\mu}. \end{aligned}$$

Since, we assume  $\boldsymbol{\mu}^\top \mathbf{A}_i \boldsymbol{\mu} = \boldsymbol{\mu}^\top (\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top) \boldsymbol{\mu} = 0$ , and  $\boldsymbol{\Lambda}$  contains only 1 and 0 on the diagonal. We have  $\boldsymbol{\Lambda} = \boldsymbol{\Lambda} \boldsymbol{\Lambda}^\top$ . That is

$$\boldsymbol{\mu}^\top (\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top) \boldsymbol{\mu} = \boldsymbol{\mu}^\top (\mathbf{Q} \boldsymbol{\Lambda} \boldsymbol{\Lambda}^\top \mathbf{Q}^\top) \boldsymbol{\mu} = \|\boldsymbol{\Lambda}^\top \mathbf{Q}^\top \boldsymbol{\mu}\|^2 = 0,$$

which implies  $2\mathbf{y}^\top \mathbf{A}_i \boldsymbol{\mu} = 2\mathbf{y}^\top (\mathbf{Q} \boldsymbol{\Lambda} \mathbf{Q}^\top) \boldsymbol{\mu} = 0$ . Thus  $q_i = q_i \sim \sigma^2 \chi_{(r_i)}^2$ .

**Case 3:** From Lemma 27.2,  $\mathbf{A}_i \mathbf{A}_j = \mathbf{0}$  if  $i \neq j$ . Let  $\mathbf{A}_i = \mathbf{Q}_i \boldsymbol{\Lambda}_i \mathbf{Q}_i^\top$ ,  $\mathbf{A}_j = \mathbf{Q}_j \boldsymbol{\Lambda}_j \mathbf{Q}_j^\top$  be the spectral decomposition of  $\mathbf{A}_i$  and  $\mathbf{A}_j$ . Then, we have

$$\begin{aligned} \mathbf{A}_i \mathbf{A}_j &= \mathbf{Q}_i \boldsymbol{\Lambda}_i \mathbf{Q}_i^\top \mathbf{Q}_j \boldsymbol{\Lambda}_j \mathbf{Q}_j^\top = \mathbf{0} \\ \mathbf{Q}_i^\top \mathbf{A}_i \mathbf{A}_j \mathbf{Q}_j &= \boldsymbol{\Lambda}_i \mathbf{Q}_i^\top \mathbf{Q}_j \boldsymbol{\Lambda}_j = \mathbf{0}. \end{aligned} \tag{27.9}$$

Write out  $q_i$  and  $q_j$ :

$$\begin{aligned} q_i &= \mathbf{y}^\top \mathbf{Q}_i \boldsymbol{\Lambda}_i \mathbf{Q}_i^\top \mathbf{y} = \mathbf{y}^\top \mathbf{Q}_i \boldsymbol{\Lambda}_i \boldsymbol{\Lambda}_i^\top \mathbf{Q}_i^\top \mathbf{y} \\ q_j &= \mathbf{y}^\top \mathbf{Q}_j \boldsymbol{\Lambda}_j \mathbf{Q}_j^\top \mathbf{y} = \mathbf{y}^\top \mathbf{Q}_j \boldsymbol{\Lambda}_j \boldsymbol{\Lambda}_j^\top \mathbf{Q}_j^\top \mathbf{y}. \end{aligned}$$

Let  $\mathbf{a}_i = \boldsymbol{\Lambda}_i^\top \mathbf{Q}_i^\top \mathbf{y}$  and  $\mathbf{a}_j = \boldsymbol{\Lambda}_j^\top \mathbf{Q}_j^\top \mathbf{y}$ , we have

$$\text{Cov}[\mathbf{a}_i, \mathbf{a}_j] = \boldsymbol{\Lambda}_i^\top \mathbf{Q}_i^\top \text{Cov}[\mathbf{y}, \mathbf{y}] \mathbf{Q}_j \boldsymbol{\Lambda}_j = \sigma^2 \boldsymbol{\Lambda}_i^\top \mathbf{Q}_i^\top \mathbf{Q}_j \boldsymbol{\Lambda}_j = \mathbf{0},$$

where the last equality is from Equation (27.9). This implies  $\text{Cov}[q_i, q_j] = 0$  since  $q_i = \mathbf{a}_i^\top \mathbf{a}_i$  and  $q_j = \mathbf{a}_j^\top \mathbf{a}_j$ . ■

Another proof is provided in (Gut, 2009). But the author did not provide inductive cases for  $k > 2$ . Interesting readers can refer to it.

## Appendix J. Taylor's Expansion

### Theorem 27.1: (Taylor's Expansion with Lagrange Remainder)

Let  $f(x) : \mathbb{R} \rightarrow \mathbb{R}$  be  $k$ -times continuously differentiable on the closed interval  $I$  with endpoints  $x$  and  $y$ , for some  $k \geq 0$ . If  $f^{(k+1)}$  exists on the interval  $I$ , then there exists a  $x^* \in (x, y)$  such that

$$\begin{aligned} & f(x) \\ &= f(y) + f'(y)(x - y) + \frac{f''(y)}{2!}(x - y)^2 + \dots + \frac{f^{(k)}(y)}{k!}(x - y)^k + \frac{f^{(k+1)}(x^*)}{(k+1)!}(x - y)^{k+1} \\ &= \sum_{i=0}^k \frac{f^{(i)}(y)}{i!}(x - y)^i + \frac{f^{(k+1)}(x^*)}{(k+1)!}(x - y)^{k+1}. \end{aligned}$$

The Taylor's expansion can be extended to a function of vector  $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  or a function of matrix  $f(\mathbf{X}) : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ .

The Taylor's expansion, or also known as the *Taylor's series*, approximates the function  $f(x)$  around the value of  $y$  by a polynomial in a single indeterminate  $x$ . To see where does this series come from, we recall from the elementary calculus course that the approximated function around  $\theta = 0$  for  $\cos(\theta)$  is given by

$$\cos(\theta) \approx 1 - \frac{\theta^2}{2}.$$

That is, the  $\cos \theta$  is approximated by a polynomial with degree of 2. Suppose we want to approximate  $\cos \theta$  by the more general polynomial with degree of 2 by  $f(\theta) = c_1 + c_2\theta + c_3\theta^2$ . A intuitive idea is to match the gradients around the 0 point. That is,

$$\left\{ \begin{array}{l} \cos(0) = f(0); \\ \cos'(0) = f'(0); \\ \cos''(0) = f''(0); \end{array} \right. \quad \xrightarrow{\text{leads to}} \quad \left\{ \begin{array}{l} 1 = c_1; \\ -\sin(0) = 0 = c_2; \\ -\cos(0) = -1 = 2c_3. \end{array} \right.$$

This makes  $f(\theta) = c_1 + c_2\theta + c_3\theta^2 = 1 - \frac{\theta^2}{2}$  and agrees with our claim that  $\cos(\theta) \approx 1 - \frac{\theta^2}{2}$  around the 0 point. We shall not give the details of the proof.

## Appendix K. Famous Inequalities

In this section, we introduce some famous inequalities that will be often used.

### Lemma 27.1: (Cauchy-Schwarz Inequality)

For any random  $m \times n$  matrices  $\mathbf{X}$  and  $\mathbf{Y}$ , we have

$$\mathbb{E} [\|\mathbf{X}^\top \mathbf{Y}\|] \leq \mathbb{E} [\|\mathbf{X}\|^2]^{1/2} \mathbb{E} [\|\mathbf{Y}\|^2]^{1/2},$$

where the inner product is defined as  $\langle \mathbf{X}, \mathbf{Y} \rangle = \mathbb{E} [\|\mathbf{X}^\top \mathbf{Y}\|]$ .

### Lemma 27.2: (Schwarz Matrix Inequality)

For any random  $m \times n$  matrices  $\mathbf{X}$  and  $\mathbf{Y}$ , we have

$$\|\mathbf{X}^\top \mathbf{Y}\| \leq \|\mathbf{X}\| \cdot \|\mathbf{Y}\|.$$

This is a special form of the Cauchy-Schwarz inequality, where the inner product is defined as  $\langle \mathbf{X}, \mathbf{Y} \rangle = \|\mathbf{X}^\top \mathbf{Y}\|$ .

### Lemma 27.3: (Markov's Inequality)

Let  $X$  be a non-negative random variable. Then, given any  $\epsilon > 0$ , we have

$$\mathbb{P}[X \geq \epsilon] \leq \frac{\mathbb{E}[X]}{\epsilon}.$$

**Proof** [of Lemma 27.3] We notice the trick that  $0 \leq \epsilon \mathbb{1}\{X \geq \epsilon\} \leq X$  since  $X$  is non-negative. This implies  $\mathbb{E}[\epsilon \mathbb{1}\{X \geq \epsilon\}] \leq \mathbb{E}[X]$ . We also have

$$\mathbb{E}[\epsilon \mathbb{1}\{X \geq \epsilon\}] = \epsilon \mathbb{E}[\mathbb{1}\{X \geq \epsilon\}] = \epsilon (1 \cdot \mathbb{P}[X \geq \epsilon] + 0 \cdot \mathbb{P}[X < \epsilon]) = \epsilon \cdot \mathbb{P}[X \geq \epsilon] \leq \mathbb{E}[X].$$

This completes the proof. ■

### Lemma 27.4: (Chebyshev's Inequality)

Let  $X$  be a random variable with finite mean  $\mathbb{E}[X] < \infty$ . Then, given any  $\epsilon > 0$ , we have

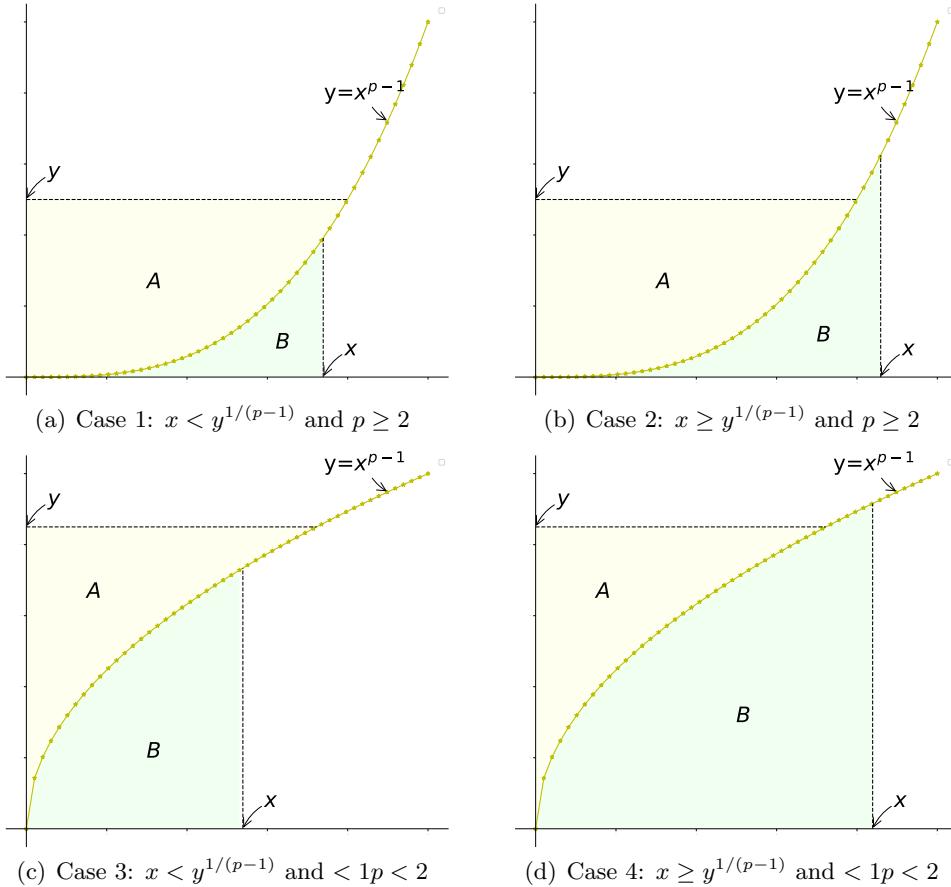
$$\mathbb{P}[|X - \mathbb{E}[X]| \geq \epsilon] \leq \frac{\text{Var}[X]}{\epsilon^2}.$$

The Chebyshev's inequality can be easily verified by defining  $Y = (X - \mathbb{E}[X])^2$  (which is non-negative) and applying Markov's inequality to  $Y$ .

**Lemma 27.5: (Hölder's Inequality, V1)**

For nonnegative numbers  $x, y \geq 0$ , and positive real numbers  $p, q > 1$  with  $\frac{1}{p} + \frac{1}{q} = 1$ , it follows that

$$xy \geq \frac{1}{p}x^p + \frac{1}{q}y^q. \quad (27.10)$$



**Figure 27.7:** Demonstration of Hölder's inequality, V1.

**Proof** [of Lemma 27.5] The area  $xy$  is smaller than the areas of the sum of the two trapezoids with curved edges (the colored ones  $A$  and  $B$ ) as shown in Figure 27.7:

$$\text{area } B = \int_0^x x^{p-1} dx, \quad \text{area } A = \int_0^y y^{1/(p-1)} dy.$$

That is,

$$\begin{aligned}
 xy &\leq \int_0^x x^{p-1} dx + \int_0^y y^{1/(p-1)} dy \\
 &= \frac{1}{p} x^p + \int_0^y y^{q/p} dy \\
 &= \frac{1}{p} x^p + \left(\frac{q}{p} + 1\right)^{-1} y^{q/p+1} \\
 &= \frac{1}{p} x^p + \frac{1}{q} y^q.
 \end{aligned}$$

This completes the proof. ■

### Lemma 27.6: (Hölder's Inequality)

Suppose  $p, q > 1$  such that  $\frac{1}{p} + \frac{1}{q} = 1$ . Then, it follows that for any vector  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , we have

$$\sum_{i=1}^n |x_i| |y_i| \leq \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} \left( \sum_{i=1}^n |y_i|^q \right)^{1/q} = \|\mathbf{x}\|_p \|\mathbf{y}\|_q,$$

where  $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$  is known as the  $l_p$  **norm** or  $p$  **norm** of vector  $\mathbf{x}$ .

**Proof** [of Lemma 27.6] Let  $u = \frac{|x_i|}{\|\mathbf{x}\|_p}$  and  $v = \frac{|y_i|}{\|\mathbf{y}\|_q}$ , From Equation (27.10), it follows that

$$uv = \frac{|x_i| |y_i|}{\|\mathbf{x}\|_p \|\mathbf{y}\|_q} \leq \frac{1}{p} \frac{|x_i|^p}{\|\mathbf{x}\|_p^p} + \frac{1}{q} \frac{|y_i|^q}{\|\mathbf{y}\|_q^q}, \quad \forall i \in \{1, 2, \dots, n\}.$$

Therefore

$$\sum_{i=1}^n \frac{|x_i| |y_i|}{\|\mathbf{x}\|_p \|\mathbf{y}\|_q} \leq \frac{1}{p} \frac{1}{\|\mathbf{x}\|_p^p} \sum_{i=1}^n |x_i|^p + \frac{1}{q} \frac{1}{\|\mathbf{y}\|_q^q} \sum_{i=1}^n |y_i|^q = \frac{1}{p} + \frac{1}{q} = 1.$$

That is

$$\sum_{i=1}^n |x_i| |y_i| \leq \|\mathbf{x}\|_p \|\mathbf{y}\|_q,$$

from which the result follows. ■

## Appendix L. Matrix Norm

A *norm* on a vector or a matrix satisfies the following properties.

### Definition 27.1: Vector Norm and Matrix Norm

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and any vector  $\mathbf{x} \in \mathbb{R}^n$ , we have

- *Nonnegativity.*  $\|\mathbf{A}\| \geq 0$  or  $\|\mathbf{x}\| \geq 0$ , and the equality obtained if and only if  $\mathbf{A} = \mathbf{0}$  or  $\mathbf{x} = \mathbf{0}$ .
- *Positive homogeneity.*  $\|\lambda\mathbf{A}\| = |\lambda|\|\mathbf{A}\|$  for any  $\lambda \in \mathbb{R}$ .
- *Triangle inequality.*  $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ , or  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$  for any matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$  or vectors  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ .

### Definition 27.2: Inner Product

In most cases, the norm can be derived from the vector *inner product*, which satisfies three criteria as follows

- *Commutativity.*  $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$  for any  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ .
- *Linearity.*  $\langle \lambda_1 \mathbf{x}_1 + \lambda_2 \mathbf{x}_2, \mathbf{y} \rangle = \lambda_1 \langle \mathbf{x}_1, \mathbf{y} \rangle + \lambda_2 \langle \mathbf{x}_2, \mathbf{y} \rangle$  for any  $\lambda_1, \lambda_2 \in \mathbb{R}$  and  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ .
- *Positive definiteness.*  $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$  for any  $\mathbf{x} \in \mathbb{R}^n$ , and  $\langle \mathbf{x}, \mathbf{x} \rangle = 0$  if and only if  $\mathbf{x} = \mathbf{0}$ .

## L.1 Vector Norm

The vector norm is from the inner product. In most cases, the inner product in  $\mathbb{R}^n$  is the *dot product* defined by

$$\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i.$$

The  $l_2$  norm is induced from the dot product

$$\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^n x_i^2}.$$

More generally, for a given  $p \geq 1$ , the  $l_p$  norm is given by

$$\|\mathbf{x}\|_p = \sqrt[p]{\sum_{i=1}^n |x_i|^p}.$$

From where, the  $l_1$  norm can be obtained by

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|.$$

And the  $l_\infty$  norm can be obtained by

$$\|\mathbf{x}\|_\infty = \max_{i=1,2,\dots,n} |x_i|.$$

We notice that

$$(\|\mathbf{x}\|_\infty)^p = (\max_{i=1,2,\dots,n} |x_i|)^p \leq \sum_{i=1}^n |x_i|^p \leq n \max_{i=1,2,\dots,n} |x_i|^p = n(\|\mathbf{x}\|_\infty)^p,$$

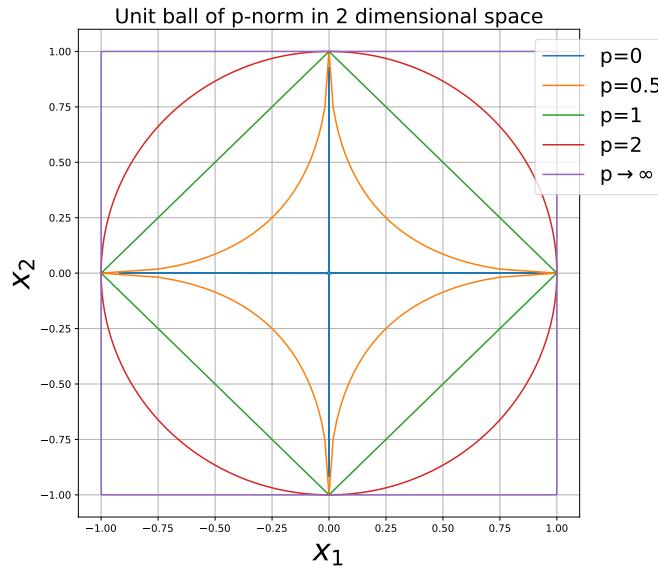
from which it follows that

$$\|\mathbf{x}\|_\infty \leq \|\mathbf{x}\|_p \leq n^{1/p} \|\mathbf{x}\|_\infty.$$

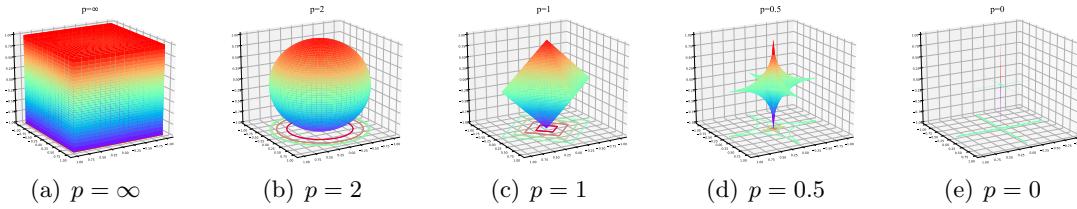
The unit ball of the norms describes the set of all points whose distance from the origin (i.e., the zero vector) is 1. So if our distance is induced by the  $l_p$  norm, then the unit ball is the collection of

$$\mathcal{B} = \{\mathbf{x} : \|\mathbf{x}\|_p = 1\}.$$

The comparison of  $l_p$  norm in 2-dimensional and 3-dimensional space with different values of  $p$  is depicted in Figure 27.8 and Figure 27.9.



**Figure 27.8:** Unit ball of  $l_p$  norm in 2-dimensional space.



**Figure 27.9:** Unit ball of  $l_p$  norm in 3-dimensional space.

The norms of a vector are quite useful in machine learning. In Section 3.20.1, we mentioned the least squares is to minimize the squared distance between observation  $\mathbf{b}$  and expected observation  $\mathbf{Ax}$ :  $\|\mathbf{Ax} - \mathbf{b}\|_2^2$ , i.e., the  $l_2$  norm of  $\mathbf{Ax} - \mathbf{b}$ . On the contrary, minimizing the  $l_1$  norm between the observation and expected observation can result in a robust estimator of  $\mathbf{x}$  (Zoubir et al., 2012).

The dot product is not the only possible inner product that can be defined over  $\mathbb{R}^n$ . For a positive definite  $n \times n$  matrix  $\mathbf{Q}$ , a  $\mathbf{Q}$ -inner product can be defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathbf{A}} = \mathbf{x}^\top \mathbf{Q} \mathbf{y}.$$

One can check the  $\mathbf{Q}$ -inner product defined above satisfies the three criteria for the inner product discussed at the beginning of this section. When  $\mathbf{Q} = \mathbf{I}$ , we recover the dot product. From the  $\mathbf{Q}$ -inner product, the  $\mathbf{Q}$ -norm can be defined as

$$\|\mathbf{x}\|_{\mathbf{Q}} = \sqrt{\mathbf{x}^\top \mathbf{Q} \mathbf{x}}.$$

### Corollary 27.3: (Vector Norm Properties)

From the definition of vector norm, we have the following properties of vector norm.

- For any vector  $\mathbf{x} \in \mathbb{R}^n$ , it follows that  $\|-\mathbf{x}\| = \|\mathbf{x}\|$ ;
- For vector  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , it follows that  $|\|\mathbf{x}\| - \|\mathbf{y}\|| \leq \|\mathbf{x} - \mathbf{y}\|$ .

**Proof** [of Corollary 27.3] From positive homogeneity of vector norm, we have  $\|-\mathbf{x}\| = |-1| \cdot \|\mathbf{x}\| = \|\mathbf{x}\|$ .

From the triangle inequality of vector norm, we have

$$\begin{aligned} \|\mathbf{x}\| &= \|\mathbf{x} - \mathbf{y} + \mathbf{y}\| \leq \|\mathbf{x} - \mathbf{y}\| + \|\mathbf{y}\|, \\ \|\mathbf{y}\| &= \|\mathbf{y} - \mathbf{x} + \mathbf{x}\| \leq \|\mathbf{y} - \mathbf{x}\| + \|\mathbf{x}\|, \end{aligned}$$

which implies

$$\begin{aligned} \|\mathbf{x}\| - \|\mathbf{y}\| &\leq \|\mathbf{x} - \mathbf{y}\|, \\ \|\mathbf{y}\| - \|\mathbf{x}\| &\leq \|\mathbf{y} - \mathbf{x}\|, \end{aligned}$$

from which the result  $|\|\mathbf{x}\| - \|\mathbf{y}\|| \leq \|\mathbf{x} - \mathbf{y}\|$  follows. ■

Following from the definition of  $l_p$ -norm, we can obtain the famous Hölder's inequality in Lemma 27.6. Thus the  $l_p$  norm is sometimes referred to as the *Hölder's norm*. The Hölder's inequality, on the contrary, can prove the validity of the  $l_p$  norm.

### Triangle Inequality of $l_p$ Norm

**Validity of  $l_p$  norm Case  $p > 1$  for  $\|\mathbf{x}\|_p$ :** From the Hölder's inequality ( $p, q$  required to be larger than 1), we could prove the validity of  $l_p$  norm. Let  $\frac{1}{p} + \frac{1}{q} = 1$ , it

follows that

$$\begin{aligned}
\sum_{i=1}^n (|x_i| + |y_i|)^p &= \sum_{i=1}^n |x_i|(|x_i| + |y_i|)^{p-1} + \sum_{i=1}^n |y_i|(|x_i| + |y_i|)^{p-1} \\
&\leq \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} \left( \sum_{i=1}^n (|x_i| + |y_i|)^{(p-1)q} \right)^{1/q} \\
&\quad + \left( \sum_{i=1}^n |y_i|^p \right)^{1/p} \left( \sum_{i=1}^n (|x_i| + |y_i|)^{(p-1)q} \right)^{1/q} \\
&= \left[ \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} + \left( \sum_{i=1}^n |y_i|^p \right)^{1/p} \right] \left( \sum_{i=1}^n (|x_i| + |y_i|)^{(p-1)q} \right)^{1/q}
\end{aligned}$$

Since  $(p-1)q = p$ , the above inequality implies

$$\left( \sum_{i=1}^n (|x_i| + |y_i|)^p \right)^{1/p} \leq \left( \sum_{i=1}^n |x_i|^p \right)^{1/p} + \left( \sum_{i=1}^n |y_i|^p \right)^{1/p} = \|\mathbf{x}\|_p + \|\mathbf{y}\|_p.$$

It is trivial that

$$\|\mathbf{x} + \mathbf{y}\|_p = \left( \sum_{i=1}^n (|x_i + y_i|)^p \right)^{1/p} \leq \left( \sum_{i=1}^n (|x_i| + |y_i|)^p \right)^{1/p}.$$

Therefore

$$\|\mathbf{x} + \mathbf{y}\|_p \leq \|\mathbf{x}\|_p + \|\mathbf{y}\|_p.$$

**Validity of  $l_p$  norm Case  $p = 1$  for  $\|\mathbf{x}\|_1$ :** It is trivial that  $\|\mathbf{x} + \mathbf{y}\|_1 \leq \|\mathbf{x}\|_1 + \|\mathbf{y}\|_1$ .

**Validity of  $l_p$  norm Case  $p < 1$  for  $\|\mathbf{x}\|_p$ :** However, when  $p < 1$ , we will not meet the triangle inequality conditions of the vector norm definition. For example, when  $p = 1/3$ , and  $\mathbf{x} = [0, 1]^\top$  and  $\mathbf{y} = [1, 0]^\top$ . It follows that

$$\begin{aligned}
\|\mathbf{x} + \mathbf{y}\|_{1/3} &= 8, & \|\mathbf{x}\|_{1/3} &= 1, & \|\mathbf{y}\|_{1/3} &= 1, \\
\|\mathbf{x} + \mathbf{y}\|_{1/3} &> \|\mathbf{x}\|_{1/3} + \|\mathbf{y}\|_{1/3}.
\end{aligned}$$

Alternatively, the triangle inequality for  $l_2$  norm can be proved by Cauchy-Schwarz inequality.

### Triangle Inequality of Vector Norm by Cauchy-Schwarz Inequality

By Cauchy-Schwarz inequality (see (Wu and Wu, 2009) for various proofs for it), we have

$$\mathbf{x}^\top \mathbf{y} \leq \|\mathbf{x}\| \cdot \|\mathbf{y}\|.$$

So

$$\begin{aligned}\|\mathbf{x} + \mathbf{y}\|^2 &= \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 + 2\mathbf{x}^\top \mathbf{y} \\ &\leq \|\mathbf{x}\|^2 + \|\mathbf{y}\|^2 + 2\|\mathbf{x}\| \cdot \|\mathbf{y}\| \\ &= (\|\mathbf{x}\| + \|\mathbf{y}\|)^2.\end{aligned}$$

This results in  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ .

We conclude this section by introducing an important property of vector norm that will be useful often.

#### Theorem 27.4: (Equivalence of Vector Norms)

Let  $\|\cdot\|_a$  and  $\|\cdot\|_b$  both two different vector norms:  $\mathbb{R}^n \rightarrow \mathbb{R}$ . Then there exist positive scalars  $\alpha$  and  $\beta$  such that for all  $\mathbf{x} \in \mathbb{R}^n$

$$\alpha\|\mathbf{x}\|_a \leq \|\mathbf{x}\|_b \leq \beta\|\mathbf{x}\|_a.$$

The proof of the above theorem can be found in ([van de Geijn and Myers, 2020](#)) and we shall not discuss it here.

## L.2 Matrix Norm

The norm of a matrix serves the same purpose as the norm of a vector. Two important matrix norms can be defined as follows.

#### Definition 27.5: Frobenius Norm

The Frobenius norm of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is defined as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1,j=1}^{m,n} (\mathbf{A}_{ij})^2} = \sqrt{\text{tr}(\mathbf{A}\mathbf{A}^\top)} = \sqrt{\text{tr}(\mathbf{A}^\top \mathbf{A})} = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_r^2},$$

i.e., the square root of the sum of the squares of the elements of  $\mathbf{A}$ .

Apparently, the Frobenius norm can be also defined by a vector 2-norm such that  $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^n \|\mathbf{a}_i\|^2}$  where  $\mathbf{a}_i$  for all  $i \in \{1, 2, \dots, n\}$  are the columns of  $\mathbf{A}$ .

#### Lemma 27.6: (Orthogonal Equivalence)

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and any orthogonal matrices  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$ , then

$$\|\mathbf{A}\|_F = \|\mathbf{U}^\top \mathbf{A} \mathbf{V}\|_F = \|\mathbf{U} \mathbf{A} \mathbf{V}^\top\|_F = \|\mathbf{U} \mathbf{A}\|_F = \|\mathbf{A} \mathbf{V}\|_F = \|\mathbf{U} \mathbf{A} \mathbf{V}\|_F.$$

**Proof** [of Lemma 27.6] We notice that

$$\begin{aligned}\|\mathbf{U}^\top \mathbf{A} \mathbf{V}\|_F &= \sqrt{\text{tr}((\mathbf{U}^\top \mathbf{A} \mathbf{V})(\mathbf{U}^\top \mathbf{A} \mathbf{V})^\top)} = \sqrt{\text{tr}(\mathbf{U}^\top \mathbf{A} \mathbf{A}^\top \mathbf{U})} \\ &= \sqrt{\text{tr}(\mathbf{A} \mathbf{A}^\top \mathbf{U} \mathbf{U}^\top)} = \sqrt{\text{tr}(\mathbf{A} \mathbf{A}^\top)} = \|\mathbf{A}\|_F,\end{aligned}$$

where the third equality is from the fact the trace is invariant under cyclic permutation. Similarly, we could prove the left of the claims. ■

### Definition 27.7: Spectral Norm

The spectral norm of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is defined as

$$\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} = \max_{\mathbf{u} \in \mathbb{R}^n : \|\mathbf{u}\|_2=1} \|\mathbf{Ax}\|,$$

which is also the maximal singular value of  $\mathbf{A}$ , i.e.,  $\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})$ . And it is also known as the *2-norm*. The second equality is from the fact that if we scale  $\mathbf{x}$  with a nonzero scalar, the norm is defined equivalently such that

$$\frac{\|\lambda \cdot \mathbf{Ay}\|}{\|\lambda \cdot \mathbf{y}\|} = \|\mathbf{Ay}\|.$$

**A word on the notation** We include the subscripts in the matrix norm and exclude the subscripts in the vector to distinguish between them.

### Definition 27.8: Induced Matrix Norm: General Matrix Norm

More generally, many matrix norms can be generated by using the concept of induced norms. Given a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , and two norms  $\|\cdot\|_a$  and  $\|\cdot\|_b$  on  $\mathbb{R}^n$  and  $\mathbb{R}^m$  respectively, the **induced matrix norm**  $\|\mathbf{A}\|_{a,b}$  is defined by

$$\|\mathbf{A}\|_{a,b} = \max_{\mathbf{x}} \{\|\mathbf{Ax}\|_b : \|\mathbf{x}\|_a \leq 1\}.$$

The matrix-vector product inequality can be easily obtained from the above definition

$$\|\mathbf{Ax}\|_b \leq \|\mathbf{A}\|_{a,b} \|\mathbf{x}\|_a.$$

The induced matrix norm can also be referred to as the  $(a, b)$ -norm. When  $a = b$ , we simply call it as an  **$a$ -norm**, and use the notation  $\|\mathbf{A}\|_a$  instead of  $\|\mathbf{A}\|_{a,a}$ .

From the definition of induced norm, we can find the spectral norm is a special induced norm when  $a = b = 2$  such that  $\|\mathbf{A}\|_2 = \|\mathbf{A}\|_{2,2}$ . Similarly, matrix 1-norm can be obtained by

$$\|\mathbf{A}\|_1 = \max_{j=1,2,\dots,n} \sum_{i=1}^m |\mathbf{A}_{ij}|,$$

which is also called the *maximum absolute column sum norm*. And the  $\infty$ -norm can be obtained by

$$\|\mathbf{A}\|_\infty = \max_{i=1,2,\dots,m} \sum_{j=1}^n |\mathbf{A}_{ij}|,$$

which is also called the *maximum absolute row sum norm*.

Similar to the  $l_p$  vector norm, we can also define the  $l_p$  matrix norm as follows.

### Definition 27.9: $l_p$ Matrix Norm

Suppose matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . For a given  $p \geq 1$ , the  $l_p$  matrix norm is given by

$$\|\mathbf{A}\|_p = \sqrt[p]{\sum_{j=1}^n \sum_{i=1}^m |\mathbf{A}_{ij}|^p}.$$

From the definition of the  $l_p$  matrix norm, specifically, we have  $l_1, l_2, l_\infty$  matrix norm

$$\begin{aligned}\|\mathbf{A}\|_{m_1} &= \sum_{j=1}^n \sum_{i=1}^m |\mathbf{A}_{ij}|, \\ \|\mathbf{A}\|_{m_2} &= \left( \sum_{j=1}^n \sum_{i=1}^m |\mathbf{A}_{ij}|^2 \right)^{1/2}, \\ \|\mathbf{A}\|_{m_\infty} &= \max_{i,j} |\mathbf{A}_{ij}|, \forall i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, n\},\end{aligned}$$

where the subscript  $m_p$  is to differentiate it from the vector  $l_p$  norm.

Take the matrix 2-norm as an example, we have the following properties that will often be used.

### Remark 27.10: Properties of the matrix 2-norm

The following properties of the matrix 2-norm follow from the definition.

1. *Homogeneity.*  $\|\lambda \mathbf{A}\|_2 = |\lambda| \cdot \|\mathbf{A}\|_2$ .
2. *Triangle inequality.*  $\|\mathbf{A} + \mathbf{B}\|_2 \leq \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2$ . And also  $\|\mathbf{A}\|_2 - \|\mathbf{B}\|_2 \leq \|\mathbf{A} - \mathbf{B}\|_2$
3. *Matrix-vector product.*  $\|\mathbf{A}\mathbf{x}\| \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|$  for all vectors  $\mathbf{x}$  that have dimension matched with  $\mathbf{A}$ .
4. *Definiteness.*  $\|\mathbf{A}\|_2 \geq 0$  for all  $\mathbf{A}$  and  $\|\mathbf{A}\|_2 = 0$  if and only if  $\mathbf{A} = \mathbf{0}$ .
5. *Matrix product.*  $\|\mathbf{AB}\|_2 \leq \|\mathbf{A}\|_2 \|\mathbf{B}\|_2$  for all matrices  $\mathbf{B}$  that have dimension matched with  $\mathbf{A}$ .
6. *Transpose.*  $\|\mathbf{A}\|_2 = \|\mathbf{A}^\top\|_2$ .
7. *Normalization.* When  $\mathbf{A} \neq \mathbf{0}$ , we have  $\|\frac{1}{\|\mathbf{A}\|_2} \mathbf{A}\|_2 = 1$ .

**Proof** [of Remark 27.10] For 1). We have

$$\|\lambda \mathbf{A}\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\lambda \mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} = \max_{\mathbf{x} \neq \mathbf{0}} \frac{|\lambda| \cdot \|\mathbf{A}\mathbf{x}\|}{\|\mathbf{x}\|} = |\lambda| \cdot \|\mathbf{A}\|_2.$$

For 2). Write out the equation, we have

$$\begin{aligned} \|\mathbf{A} + \mathbf{B}\|_2 &= \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|(\mathbf{A} + \mathbf{B})\mathbf{x}\|}{\|\mathbf{x}\|} \\ &\leq \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Ax}\| + \|\mathbf{Bx}\|}{\|\mathbf{x}\|} \\ &\leq \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} + \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Bx}\|}{\|\mathbf{x}\|} \\ &= \|\mathbf{A}\|_2 + \|\mathbf{B}\|_2, \end{aligned}$$

where the first inequality comes from the triangle inequality of vector norm:  $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ .

For 3). By definition,  $\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}$ , it is trivial that  $\|\mathbf{A}\|_2 \geq \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}$ . This implies  $\|\mathbf{Ax}\| \leq \|\mathbf{A}\|_2 \|\mathbf{x}\|$ .

For 4). It is trivial that the vector norm is positive such that

$$\|\mathbf{A}\|_2 \geq \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|} > 0.$$

For 5). Write out the equation, we have

$$\begin{aligned} \|\mathbf{AB}\|_2 &= \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{ABx}\|}{\|\mathbf{x}\|} \\ &\leq \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{A}\|_2 \cdot \|\mathbf{Bx}\|}{\|\mathbf{x}\|} \quad \text{by matrix-vector product inequality} \\ &= \|\mathbf{A}\|_2 \cdot \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Bx}\|}{\|\mathbf{x}\|} \\ &\leq \|\mathbf{A}\|_2 \cdot \|\mathbf{B}\|_2. \end{aligned}$$

The proof for 6) and 7) are trivial. ■

# Bibliography

- Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 557–563, 2006.
- Theodore Wilbur Anderson and George PH Styan. Cochran’s theorem, rank additivity, and tripotent matrices. Technical report, STANFORD UNIV CA DEPT OF STATISTICS, 1980.
- A Baarland. Perturbation bounds for the LDLH and the LU factorizations. *BIT*, 31: 341–352, 1991.
- Zhaojun Bai and James W Demmel. Computing the generalized singular value decomposition. *SIAM Journal on Scientific Computing*, 14(6):1464–1486, 1993.
- Sudipto Banerjee and Anindya Roy. *Linear algebra and matrix analysis for statistics*, volume 181. Crc Press Boca Raton, FL, USA:, 2014.
- Amir Beck. *First-Order Methods in Optimization*, volume 25. SIAM, 2017.
- James Bennett, Stan Lanning, et al. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35. New York, NY, USA., 2007.
- Daniele Bigoni, Allan P Engsig-Karup, and Youssef M Marzouk. Spectral tensor-train decomposition. *SIAM Journal on Scientific Computing*, 38(4):A2405–A2439, 2016.
- Christian Bischof. *The two-sided block Jacobi method on a hypercube architecture*, volume 86. Mathematical Sciences Institute, Cornell University, 1986.
- Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.
- Adam W Bojanczyk, RP Brent, Paul Van Dooren, and FR De Hoog. A note on downdating the Cholesky factorization. *SIAM Journal on Scientific and Statistical Computing*, 8(3): 210–221, 1987.
- Christos Boutsidis and Efstratios Gallopoulos. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern recognition*, 41(4):1350–1362, 2008.
- Christos Boutsidis, Michael W Mahoney, and Petros Drineas. An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 968–977. SIAM, 2009.

Stephen Boyd and Lieven Vandenberghe. *Introduction to applied linear algebra: vectors, matrices, and least squares*. Cambridge university press, 2018.

Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

M Isabel Bueno and Froilán M Dopico. Stability and sensitivity of tridiagonal LU factorization without pivoting. *BIT Numerical Mathematics*, 44(4):651–673, 2004.

Tony F Chan. An improved algorithm for computing the singular value decomposition. *ACM Transactions on Mathematical Software (TOMS)*, 8(1):72–83, 1982.

Tony F Chan. Rank revealing QR factorizations. *Linear algebra and its applications*, 88: 67–82, 1987.

Xiao-Wen Chang. *Perturbation Analysis of Some Matrix Factorizations*. McGill University Montreal, 1997.

Xiao-Wen Chang and Christopher C Paige. On the sensitivity of the LU factorization. *BIT Numerical Mathematics*, 38(3):486–501, 1998.

Xiao-Wen Chang, Christopher C Paige, and GW Stewart. Perturbation analyses for the QR factorization. *SIAM Journal on Matrix Analysis and Applications*, 18(3):775–791, 1997.

Tianqi Chen, Ian Goodfellow, and Jonathon Shlens. Net2net: Accelerating learning via knowledge transfer. *arXiv preprint arXiv:1511.05641*, 2015.

Yanqing Chen, Timothy A Davis, William W Hager, and Sivasankaran Rajamanickam. Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/down-date. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):1–14, 2008.

Hongwei Cheng, Zydrunas Gimbutas, Per-Gunnar Martinsson, and Vladimir Rokhlin. On the compression of low rank matrices. *SIAM Journal on Scientific Computing*, 26(4): 1389–1404, 2005.

Ronald Christensen. *Linear models for multivariate, time series, and spatial data*, volume 1. Springer, 1991.

Moody T Chu, Robert E Funderlic, and Gene H Golub. A rank-one reduction formula and its applications to matrix factorizations. *SIAM review*, 37(4):512–530, 1995.

Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P Mandic. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9 (4-5):249–429, 2016.

Barry A Cipra. The best of the 20th century: Editors name top 10 algorithms. *SIAM news*, 33(4):1–2, 2000.

- Pierre Comon, Xavier Luciani, and André LF De Almeida. Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 23(7-8):393–405, 2009.
- Timothy A Davis. User guide for CHOLMOD: a sparse Cholesky factorization and modification package. *Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA*, 2008.
- Timothy A Davis and William W Hager. Modifying a sparse Cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, 20(3):606–627, 1999.
- Robyn M Dawes and Bernard Corrigan. Linear models in decision making. *Psychological bulletin*, 81(2):95, 1974.
- Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- Jack Dongarra and Francis Sullivan. Guest editors’ introduction: The top 10 algorithms. *IEEE Computer Architecture Letters*, 2(01):22–23, 2000.
- Froilán M Dopico, Charles R Johnson, and Juan M Molera. Multiple LU factorizations of a singular matrix. *Linear algebra and its applications*, 419(1):24–36, 2006.
- Petros Drineas, Malik Magdon-Ismail, Michael W Mahoney, and David P Woodruff. Fast approximation of matrix coherence and statistical leverage. *The Journal of Machine Learning Research*, 13(1):3475–3506, 2012.
- Ricardo D Fierro and Per Christian Hansen. Low-rank revealing UTV decompositions. *Numerical Algorithms*, 15(1):37–55, 1997.
- Leslie V Foster. Solving rank-deficient and ill-posed problems using UTV and QR factorizations. *SIAM journal on matrix analysis and applications*, 25(2):582–600, 2003.
- John Fox. *Applied regression analysis, linear models, and related methods*. Sage Publications, Inc, 1997.
- John GF Francis. The QR transformation a unitary analogue to the LR transformation—part 1. *The Computer Journal*, 4(3):265–271, 1961.
- John GF Francis. The QR transformation—part 2. *The Computer Journal*, 4(4):332–345, 1962.
- Jean Gallier and Jocelyn Quaintance. *Fundamentals of Linear Algebra and Optimization*. Department of Computer and Information Science, University of Pennsylvania, 2017.
- Jean Gallier et al. The schur complement and symmetric positive semidefinite (and definite) matrices. *Penn Engineering*, pages 1–12, 2010.
- Carl Friedrich Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*, volume 1. FA Perthes, 1809.

Carl Friedrich Gauss. *Disquisitio de elementis elliptics Palladis*. Dieterich, 1810.

James E Gentle. *Numerical linear algebra for applications in statistics*. Springer Science & Business Media, 1998.

James E Gentle. Matrix algebra. *Springer texts in statistics, Springer, New York, NY, doi, 10:978–0, 2007.*

Paris V Giampouras, Athanasios A Rontogiannis, and Konstantinos D Koutroumbas. Alternating iteratively reweighted least squares minimization for low-rank matrix factorization. *IEEE Transactions on Signal Processing*, 67(2):490–503, 2018.

George T Gilbert. Positive definite matrices and Sylvester’s criterion. *The American Mathematical Monthly*, 98(1):44–46, 1991.

Philip E Gill, Gene H Golub, Walter Murray, and Michael A Saunders. Methods for modifying matrix factorizations. *Mathematics of computation*, 28(126):505–535, 1974.

Philip E Gill, Walter Murray, and Margaret H Wright. *Numerical linear algebra and optimization*. SIAM, 2021.

Nicolas Gillis. The why and how of nonnegative matrix factorization. *Connections*, 12:2–2, 2014.

Israel Gohberg and Seymour Goldberg. A simple proof of the jordan decomposition theorem for matrices. *The American Mathematical Monthly*, 103(2):157–159, 1996.

Donald Goldfarb. Factorized variable metric methods for unconstrained optimization. *Mathematics of Computation*, 30(136):796–811, 1976.

Gene Golub and William Kahan. Calculating the singular values and pseudo-inverse of a matrix. *Journal of the Society for Industrial and Applied Mathematics, Series B: Numerical Analysis*, 2(2):205–224, 1965.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2013.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Sergei A Goreinov and Eugene E Tyrtyshnikov. The maximal-volume concept in approximation by low-rank matrices. *Contemporary Mathematics*, 280:47–52, 2001.

Sergei A Goreinov, Nikolai Leonidovich Zamarashkin, and Evgenii Evgen’evich Tyrtyshnikov. Pseudo-skeleton approximations by matrices of maximal volume. *Mathematical Notes*, 62(4):515–519, 1997.

Sergei A Goreinov, Ivan V Oseledets, Dmitry V Savostyanov, Eugene E Tyrtyshnikov, and Nikolay L Zamarashkin. How to find a good submatrix. In *Matrix Methods: Theory, Algorithms And Applications: Dedicated to the Memory of Gene Golub*, pages 247–256. World Scientific, 2010.

- Allan Gut. Quadratic forms and Cochran's theorem. In *An Intermediate Course in Probability*, pages 117–145. Springer, 2009.
- AW Hales and IBS Passi. Jordan decomposition. In *Algebra*, pages 75–87. Springer, 1999.
- Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.
- F Maxwell Harper and Joseph A Konstan. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19, 2015.
- Johan Håstad. Tensor rank is np-complete. In *International Colloquium on Automata, Languages, and Programming*, pages 451–460. Springer, 1989.
- Peter Henrici. On the speed of convergence of cyclic and quasicyclic jacobi methods for computing eigenvalues of hermitian matrices. *Journal of the Society for Industrial and Applied Mathematics*, 6(2):144–162, 1958.
- Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002.
- Nicholas J Higham. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):251–254, 2009.
- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- Alston S Householder. *Principles of numerical analysis*. Courier Corporation, 2006.
- Tsung-Min Hwang, Wen-Wei Lin, and Eugene K Yang. Rank revealing LU factorizations. *Linear algebra and its applications*, 175:115–141, 1992.
- C. G. J. Jacobi and Über ein leichtes Verfahren. die in der Theorie der säkular-störungen vorkommenden gleichungen numerisch aufzulösen. *Crelle's Journal* 30, 4(4):51–94, 1846.
- GS James. Notes on a theorem of Cochran. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 48, pages 443–446. Cambridge University Press, 1952.
- Camille Jordan. *Traité des substitutions et des équations algébriques*. Gauthier-Villars, 1870.
- Henk AL Kiers. Towards a standardized notation and terminology in multiway analysis. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 14(3):105–122, 2000.
- N Kishore Kumar and Jan Schneider. Literature survey on low rank approximation of matrices. *Linear and Multilinear Algebra*, 65(11):2212–2244, 2017.
- Martin Koeber and Uwe Schäfer. The unique square root of a positive semidefinite matrix. *International Journal of Mathematical Education in Science and Technology*, 37(8):990–992, 2006.

- Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- Tamara Gibson Kolda. Multilinear operators for higher-order decompositions. Technical report, Citeseer, 2006.
- Joseph B Kruskal. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138, 1977.
- PW Lane. Generalized linear models in soil science. *European Journal of Soil Science*, 53(2):241–251, 2002.
- Charles L Lawson and Richard J Hanson. *Solving least squares problems*. SIAM, 1995.
- Daniel D Lee and Hyunjune Sebastian Seung. Algorithms for non-negative matrix factorization. In *14th Annual Neural Information Processing Systems Conference, NIPS 2000*. Neural information processing systems foundation, 2001.
- Cornelius T Leondes. *Multidimensional Systems: Signal Processing and Modeling Techniques: Advances in Theory and Applications*. Elsevier, 1995.
- Jun Lu. Machine learning modeling for time series problem: Predicting flight ticket prices. *arXiv preprint arXiv:1705.07205*, 2017.
- Jun Lu. A survey on Bayesian inference for Gaussian mixture model. *arXiv preprint arXiv:2108.11753*, 2021a.
- Jun Lu. On the column and row ranks of a matrix. *arXiv preprint arXiv:2112.06638*, 2021b.
- Jun Lu. Revisit the fundamental theorem of linear algebra. *arXiv preprint arXiv:2108.04432*, 2021c.
- Jun Lu. A rigorous introduction for linear models. *arXiv preprint arXiv:2105.04240*, 2021d.
- Jun Lu. Matrix decomposition and applications. *arXiv preprint arXiv:2201.00145*, 2022.
- Jun Lu, Wei Ma, and Boi Faltings. Compnet: Neural networks growing via the compact network morphism. *arXiv preprint arXiv:1804.10316*, 2018.
- Wei Ma and Jun Lu. An equivalence of fully connected layer and convolutional layer. *arXiv preprint arXiv:1712.01252*, 2017.
- Michael W Mahoney and Petros Drineas. CUR matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, 106(3):697–702, 2009.
- Per-Gunnar Martinsson. Randomized methods for matrix computations. *The Mathematics of Data*, 25(4):187–231, 2019.
- PG Martinsson and JA Tropp. Randomized numerical linear algebra: foundations & algorithms (2020). *arXiv preprint arXiv:2002.01387*.

- L Miranian and Ming Gu. Strong rank revealing LU factorizations. *Linear algebra and its applications*, 367:1–16, 2003.
- Raphael A Mrode. *Linear models for the prediction of animal breeding values*. Cabi, 2014.
- Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
- Christopher C Paige and Michael A Saunders. Towards a generalized singular value decomposition. *SIAM Journal on Numerical Analysis*, 18(3):398–405, 1981.
- C-T Pan. On the existence and computation of rank-revealing LU factorizations. *Linear Algebra and its Applications*, 316(1-3):199–222, 2000.
- V Paul Pauca, Jon Piper, and Robert J Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear algebra and its applications*, 416(1):29–47, 2006.
- Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.
- Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.
- Heinz Rutishauser. Solution of eigenvalue problems with the LR-transformation. *National Bureau of Standards, Applied Mathematics Series*, 49:47–81, 1958.
- Lawrence R Schaeffer. Application of random regression models in animal breeding. *Livestock Production Science*, 86(1-3):35–45, 2004.
- Wil HA Schilders. Solution of indefinite linear systems using an LQ decomposition for the linear constraints. *Linear algebra and its applications*, 431(3-4):381–395, 2009.
- A Schönhage. On the quadratic convergence of the jacobi process. *Numer. Math.*, 6(410):12, 1964.
- Matthias Seeger. Low rank updates for the Cholesky decomposition. Technical report, 2004.
- Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.
- Gautam Shroff and Robert Schreiber. On the convergence of the cyclic jacobi method for parallel block orderings. *SIAM journal on matrix analysis and applications*, 10(3):326–346, 1989.
- Gilbert W Stewart. *Matrix Algorithms: Volume 1: Basic Decompositions*. SIAM, 1998.
- GW Stewart. On the perturbation of LU, Cholesky, and QR factorizations. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1141–1145, 1993.
- GW Stewart. On the perturbation of LU and Cholesky factors. *IMA Journal of Numerical Analysis*, 17(1):1–6, 1997.

GW Stewart. The decompositional approach to matrix computation. *Computing in Science & Engineering*, 2(1):50–59, 2000.

Gilbert Strang. The fundamental theorem of linear algebra. *The American Mathematical Monthly*, 100(9):848–855, 1993.

Gilbert Strang. *Introduction to linear algebra*. Wellesley-Cambridge Press Wellesley, 4th edition, 2009.

Gilbert Strang. *Linear algebra and learning from data*. Wellesley-Cambridge Press Cambridge, 2019.

Gilbert Strang. *Linear algebra for everyone*. Wellesley-Cambridge Press Wellesley, 2021.

Gilbert Strang and Daniel Drucker. Three matrix factorizations from the steps of elimination. 2021.

Gilbert Strang and Cleve Moler. LU and CR elimination. 2021.

Ji-Guang Sun. Componentwise perturbation bounds for some matrix decompositions. *BIT Numerical Mathematics*, 32(4):702–714, 1992a.

Ji-guang Sun. Rounding-error and perturbation bounds for the Cholesky and LDLT factorizations. *Linear algebra and its applications*, 173:77–97, 1992b.

Kuduvally Swamy. On Sylvester’s criterion for positive-semidefinite matrices. *IEEE Transactions on Automatic Control*, 18(3):306–306, 1973.

Panagiotis Symeonidis and Andreas Ziopoulos. *Matrix and Tensor Factorization Techniques for Recommender Systems*, volume 1. Springer, 2016.

Gábor Takács and Domonkos Tikk. Alternating least squares for personalized ranking. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 83–90, 2012.

WY Tan. Some matrix results and extensions of Cochran’s theorem. *SIAM Journal on Applied Mathematics*, 28(3):547–554, 1975.

Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

Robert van de Geijn and Margaret Myers. Advanced linear algebra: Foundations to frontiers. *Creative Commons NonCommercial (CC BY-NC)*, 2020.

HPM Van Kempen. On the quadratic convergence of the special cyclic jacobi method. *Numerische Mathematik*, 9(1):19–22, 1966.

Charles Van Loan. The block jacobi method for computing the singular value decomposition. Technical report, Cornell University, 1985.

Charles F Van Loan. The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1-2):85–100, 2000.

- Field G Van Zee, Robert A Van De Geijn, Gregorio Quintana-Ortí, and G Joseph Elizondo. Families of algorithms for reducing a matrix to condensed form. *ACM Transactions on Mathematical Software (TOMS)*, 39(1):1–32, 2012.
- Field G Van Zee, Robert A Van de Geijn, and Gregorio Quintana-Ortí. Restructuring the tridiagonal and bidiagonal QR algorithms for performance. *ACM Transactions on Mathematical Software (TOMS)*, 40(3):1–34, 2014.
- Sergey Voronin and Per-Gunnar Martinsson. Efficient algorithms for CUR and interpolative matrix decompositions. *Advances in Computational Mathematics*, 43(3):495–516, 2017.
- Tao Wei, Changhu Wang, Yong Rui, and Chang Wen Chen. Network morphism. In *International Conference on Machine Learning*, pages 564–572. PMLR, 2016.
- James H Wilkinson. Global convergene of tridiagonal QR algorithm with origin shifts. *Linear Algebra and its Applications*, 1(3):409–420, 1968.
- James Hardy Wilkinson. Note on the quadratic convergence of the cyclic jacobi process. *Numerische Mathematik*, 4(1):296–300, 1962.
- JH Wilkinson. The algebraic eigenvalue problem. In *Handbook for Automatic Computation, Volume II, Linear Algebra*. Springer-Verlag New York, 1971.
- Hui-Hua Wu and Shanhe Wu. Various proofs of the Cauchy-Schwarz inequality. *Octogon mathematical magazine*, 17(1):221–229, 2009.
- Ming Yang. Matrix decomposition. *Northwestern University, Class Notes*, 2000.
- H Zha. Restricted singular value decomposition of matrix triplets, number 89-2 in. *Scientific Report*, 1989.
- Xian-Da Zhang. *Matrix analysis and applications*. Cambridge University Press, 2017.
- X Zhu and W Lin. Randomised pseudo-skeleton approximation and its application in electromagnetics. *Electronics letters*, 47(10):590–592, 2011.
- Abdelhak M Zoubir, Visa Koivunen, Yacine Chakhchoukh, and Michael Muma. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Processing Magazine*, 29(4):61–80, 2012.

# Alphabetical Index

Algebraic multiplicity, 248

ALS, 343, 405, 410

Bidiagonal matrix, 215

Bulge, 316

Characteristic polynomial, 247

Cochran's theorem, 469

Column-pivoted QR (CPQR), 101

Coordinate transformation, 335

Cramer's rule, 179

Eckart-Young-Misky theorem, 284

Elementary transformation, 32

Fibonacci number, 234

Frobenius norm, 281

Fundamental theorem of linear algebra, 427, 430

Gaussian elimination complexity, 160

Generalized inverse, 448

Geometric multiplicity, 248

Givens rotation, 116

Golub-Kahan, 215

Hessenberg matrix, 197

HOSVD, 412

Householder reflector, 108

Idempotent decomposition, 469

Idempotent matrices, 433

Implicit Q theorem, 205, 213, 316

Induction, 65

Interlacing property, 272

Jacobi's rotation, 320

Jordan block, 232

Krylov matrix, 207

Lagrange multiplier, 293

Lauchli matrix, 97

Leading principal minors, 31, 378

Least squares, 125, 146, 278

LHC bidiagonalization, 221

Low-rank approximation, 284

Low-rank neural networks, 383

Matrix bandwidth, 44

Matrix norm, 476

Matrix products, 397

Matrix rank, 148, 426

Nested projection, 440

NMF, 365

Noise disturbance, 443

Nonlinear function layer, 385

Off-diagonal norm, 320

One-sided inverse, 445

Orthogonal, 87

Orthogonal equivalent matrices, 271

Orthogonal projection, 270, 432

Orthonormal, 87

Pivot, 34, 63

Positive definite, 56, 69, 260

Positive semidefinite, 56, 72, 260

Principal component analysis, 281

Principal minors, 30

Projection matrix, 258, 432

Pseudo-inverse, 279, 445

Rank deficient, 146

Rank-one reduction, 372

Rank-one tensor, 393

- Rank-one update, 73, 127
- Rank-revealing, 52, 72, 73, 106, 261
- Rank-revealing LU, 52
- Rank-revealing QR, 107, 187
- Rank-two update, 76
- Rate of convergence, 291
- Rayleigh quotient, 293
- Reduced row echelon form, 158
- Reflexive generalized inverse, 452
- Row echelon form, 159
- Row reduced echelon form, 160
- Row-pivoted, 124
- RPLQ, 124
- Schur complement, 58, 464
- Semidefinite rank-revealing, 72, 261
- Similar matrices, 198, 248
- Similarity transformation, 198
- Skew-symmetric matrix, 255
- Sylvester's criterion, 61
- Tensor fibers, 392
- Tensor indexing, 391
- Tensor slices, 392
- Trace, 165, 198, 433
- Tridiagonal matrix, 209
- Uniqueness, 66, 122
- Vector norm, 476
- Wedderburn sequence, 373