

Matrix Decomposition and Applications

Jun Lu

JUN.LU.LOCKY@GMAIL.COM

Preface

The realm of matrices is as vast as it is indispensable, with applications spanning from the minutiae of quantum systems to the expansive challenges of large-scale data analytics. At the heart of matrix analysis lies the transformative process of matrix decomposition—a method of reducing a complex matrix into simpler, constituent parts that illuminate its structure and utility. Far from being merely an abstract mathematical concept, matrix decomposition has become a cornerstone in fields as diverse as computer science, engineering, physics, and economics.

At its essence, matrix decomposition simplifies the representation and manipulation of matrices by breaking them down into manageable components. This process enables efficient solutions to linear systems, reduces computational complexity, and provides insights into data's inherent structure. Its applications are far-reaching, influencing everything from machine learning and optimization to image processing and recommender systems.

The historical roots of matrix decomposition trace back to the foundational work of Alston S. Householder in the mid-20th century, which set the stage for modern numerical analysis. Over the decades, the field has seen tremendous advancements, including innovations like backpropagation for neural networks, dimensionality reduction techniques in machine learning, and the utilization of low-rank matrices in natural language processing and large language models.

Today, matrix decomposition underpins technologies in statistics, optimization, and artificial intelligence. It is fundamental to the functioning of algorithms in deep neural networks, recommendation systems, and high-dimensional data analysis, among others. These applications not only underscore its practical significance but also highlight the evolving complexity of its theoretical underpinnings.

This book seeks to serve as a comprehensive and accessible introduction to matrix decomposition, offering readers a bridge between theoretical concepts and practical applications. It is designed for readers with a foundational understanding of linear algebra and aims to achieve the following objectives:

- *Explore core principles.* Present the mathematical foundations of matrix decomposition, ranging from basic methods such as LU, Cholesky, and QR decomposition to advanced techniques like SVD, eigenvalue decomposition, and their modern extensions.

- *Highlight practical applications.* Demonstrate the relevance of decomposition methods in diverse fields, including optimization, machine learning, neural network compression, and data interpretation.
- *Facilitate problem-solving.* Equip readers with the tools to understand and solve problems involving matrices, emphasizing how decomposition can simplify complex tasks and provide deeper insights.

Keywords: Existence and computing of matrix decompositions, 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.

Acknowledgement: We extend our deepest gratitude to Gilbert Strang for posing the problem articulated in Corollary 6.2, reviewing the manuscript, and providing invaluable insights and references on the three factorizations derived from elimination steps. We are especially thankful for his generosity in sharing the manuscript of [Strang and Drucker \(2021\)](#), which greatly enriched our understanding of the subject. We also extend our heartfelt appreciation to the anonymous professors who offered their consultation, feedback, and expressed interest in adopting this book as course material for college-level instruction. The author also acknowledges the collaborative contributions of Joerg Osterrieder, Christine P. Chai, and Xuanyu Ye in developing the Bayesian approach for nonnegative matrix factorization and (intervened) interpolative decomposition. Their work has significantly illuminated the structure and content of several sections in this book, providing critical perspectives and innovative methodologies.

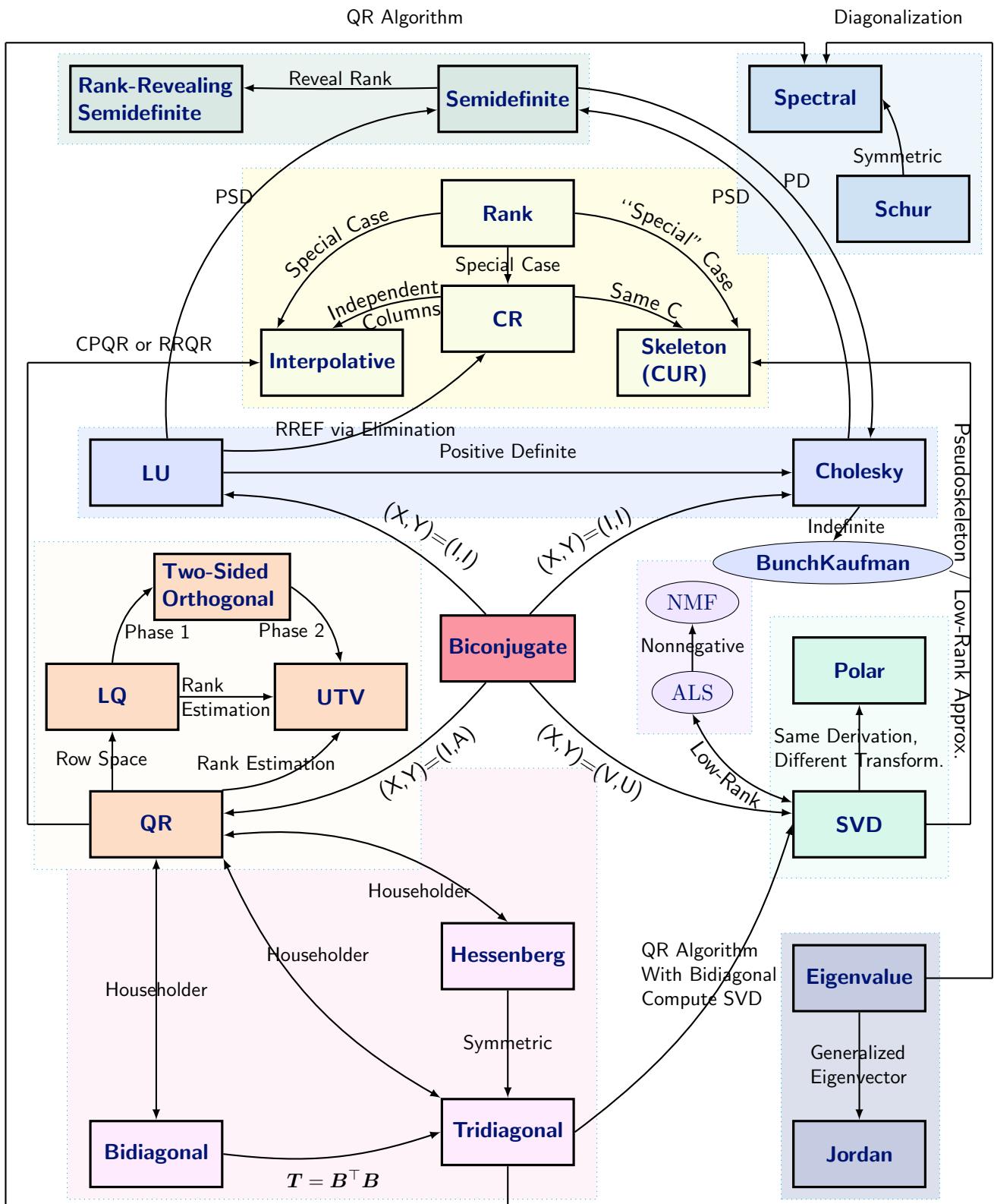


Figure 1: Matrix Decomposition World Map.

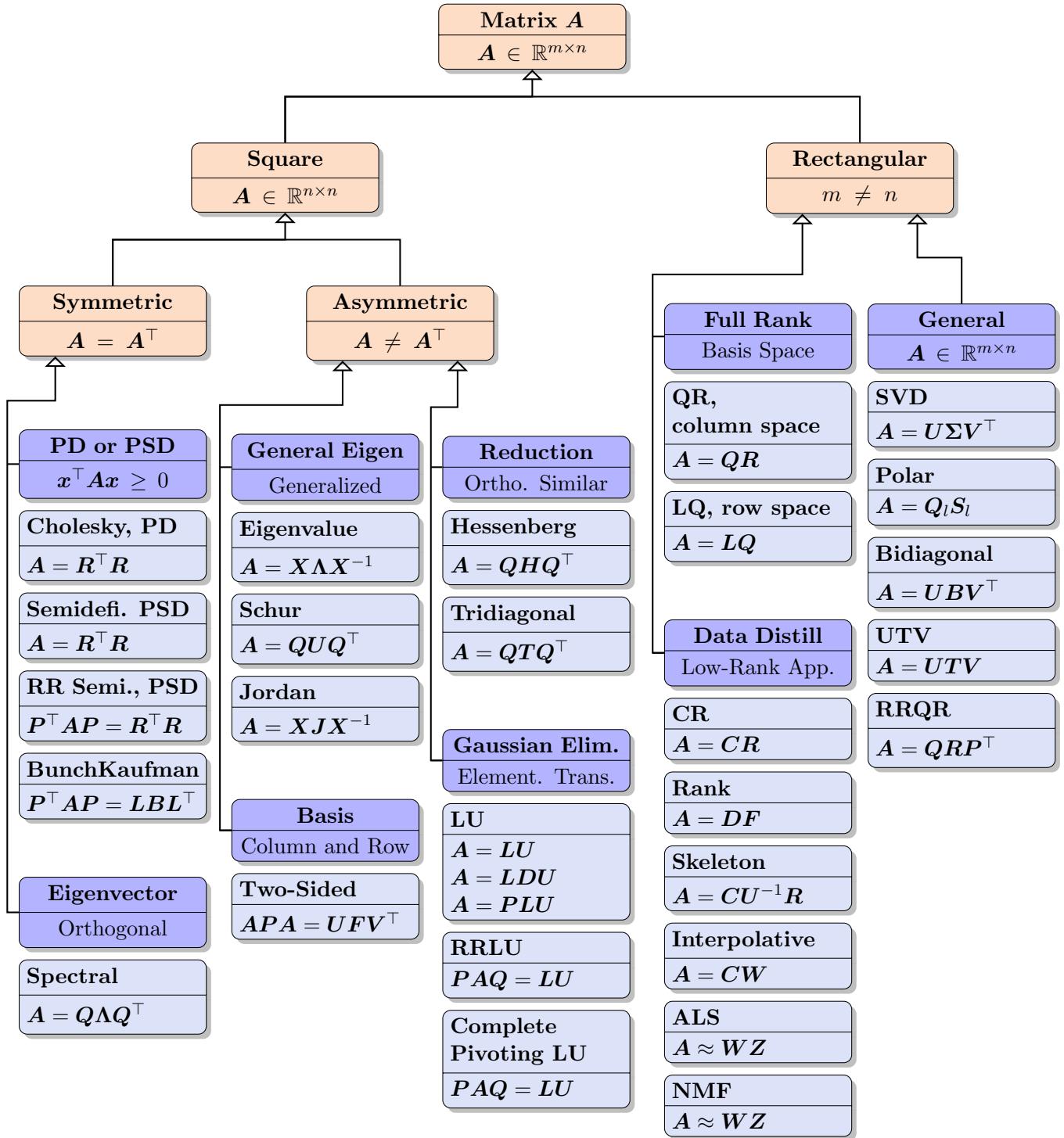


Figure 2: Matrix Decomposition World Map Under Conditions.

Contents

Introduction and Background	1
I Gaussian Elimination	12
1 LU Decomposition	14
1.1 LU Decomposition	14
1.2 Relation to Gaussian Elimination	15
1.3 Existence of LU Decomposition without Permutation	18
1.4 Existence of LU Decomposition with Permutation	19
1.5 Bandwidth Preserving in LU Decomposition without Permutation	20
1.6 Block LU Decomposition	21
1.7 Partial, Complete, and Rook Pivoting	22
1.8 Rank-Revealing LU Decomposition	24
1.9 Application: Linear System via LU Decomposition	24
1.10 Application: Computing the Inverse of Nonsingular Matrices	25
1.11 Application: Computing the Determinant	25
Chapter 1 Problems	26
2 Cholesky Decomposition	28
2.1 Cholesky Decomposition	28
2.2 Existence of Cholesky Decomposition via Recursive Calculation	29
2.3 Sylvester's Criterion: Leading Principal Minors of PD Matrices	33
2.4 Existence of Cholesky Decomposition via LU without Permutation	34
2.5 Existence of Cholesky Decomposition via Induction	36
2.6 Uniqueness of Cholesky Decomposition	37
2.7 Computing Cholesky Decomposition	38
2.8 Final Remarks on Positive Definite Matrices	39
2.9 Decomposition for Semidefinite Matrices	40
2.10 Application: Rank-One and Rank-Two Update/Downdate	42
2.11 Application: Indefinite Rank-Two Update	45
2.12 Application: Modified Newton's Method and Nearest Correlation	46
2.13 Application: Obtain Orthonormal Basis	46
2.14 Application: Low-Rank Approximation	47

Chapter 2 Problems	48
II Triangularization, Orthogonalization, and Gram–Schmidt Process	50
3 QR Decomposition	52
3.1 QR Decomposition	52
3.2 Project a Vector Onto Another Vector and Onto a Plane	53
3.3 Existence of QR Decomposition via Gram–Schmidt Process	54
3.4 Orthogonal vs Orthonormal	55
3.5 Computing Reduced QR Decomposition via CGS and MGS	56
3.6 Computing Full QR Decomposition via Gram–Schmidt Process	60
3.7 Dependent Columns	60
3.8 QR with Column Pivoting: Column-Pivoted QR (CPQR)	61
3.9 QR with Column Pivoting: Revealing Rank-One Deficiency	63
3.10 QR with Column Pivoting: Revealing Rank-r Deficiency*	64
3.11 Existence of QR Decomposition via Householder Reflector	65
3.12 Existence of QR Decomposition via Givens Rotation	67
3.13 Uniqueness of QR Decomposition	71
3.14 LQ, RRLQ, QL, RQ Decomposition	72
3.15 Two-Sided Orthogonal Decomposition	73
3.16 Application: Rank-One Changes	74
3.17 Application: Appending or Deleting a Column	75
3.18 Application: Appending or Deleting a Row	77
3.19 Application: Gauss–Newton and Levenberg–Marquardt Method	78
3.20 Application: Low-Rank Approximation	80
Chapter 3 Problems	81
4 UTV Decomposition: ULV and URV Decomposition	84
4.1 UTV Decomposition	84
4.2 Complete Orthogonal Decomposition	86
4.3 Computing the UTV Decomposition	86
4.4 Rank-Revealing UTV Decomposition and Other Issues	87
4.5 Application: Row Rank equals Column Rank via UTV	90
Chapter 4 Problems	91
III Data Interpretation and Information Distillation	92
5 CR Decomposition	94
5.1 CR Decomposition	94
5.2 Existence of CR Decomposition	94
5.3 Reduced Row Echelon Form (RREF)	95
5.4 Rank Decomposition	99
5.5 Application: Idempotent Matrix and Matrix Rank	100
Chapter 5 Problems	101

6 Skeleton/CUR and Interpolative Decomposition	102
6.1 Skeleton/CUR Decomposition	102
6.2 Existence of the Skeleton Decomposition	103
6.3 Interpolative Decomposition (ID)	105
6.4 Existence of the Column Interpolative Decomposition	107
6.5 Row ID and Two-Sided ID	110
6.6 Application: Low-Rank Approximation via Pseudoskeleton	111
6.7 Application: Feature Selection	112
Chapter 6 Problems	113
IV Reduction to Hessenberg, Tridiagonal, and Bidiagonal Form	116
7 Hessenberg and Tridiagonal Decomposition	118
7.1 Hessenberg Decomposition	119
7.2 (Orthogonal) Similarity Transformation	120
7.3 Existence of Hessenberg Decomposition	121
7.4 Properties of Hessenberg Decomposition	123
7.5 Hessenberg-Triangular Decomposition	125
7.6 Tridiagonal Decomposition: Hessenberg in Symmetric Matrices	126
7.7 Properties of Tridiagonal Decomposition	126
Chapter 7 Problems	127
8 Bidiagonal Decomposition	130
8.1 Bidiagonal Decomposition	130
8.2 Existence of Bidiagonal Decomposition: Three Approaches	131
8.3 Connection to Tridiagonal Decomposition	136
Chapter 8 Problems	137
V Eigenvalue Problem	138
9 Eigenvalue, Jordan, and Schur Decomposition	140
9.1 Eigenvalue Decomposition	140
9.2 Jordan Decomposition	141
9.3 Schur Decomposition	143
9.4 Other Forms of Schur Decomposition	145
9.5 Application: Computing Fibonacci Numbers	146
9.6 Application: Matrix Polynomials	146
9.7 Applications and Properties of Schur Decomposition	148
Chapter 9 Problems	149
10 Spectral Decomposition (Theorem)	152
10.1 Spectral Decomposition (Theorem)	152
10.2 Existence of Spectral Decomposition	153
10.3 Uniqueness of Spectral Decomposition	157
10.4 Other Forms, Connecting Eigenvalue Decomposition*	158
10.5 Skew-Symmetric Matrix and its Properties*	163
10.6 Applications in Optimization, Linear Algebra, Machine Learning	166
Chapter 10 Problems	173

11 Singular Value Decomposition (SVD)	176
11.1 Singular Value Decomposition (SVD)	176
11.2 Existence of the SVD	177
11.3 Properties of the SVD	179
11.4 Polar Decomposition	182
11.5 Coordinate Transformation in Matrix Decomposition	183
11.6 Application: LS via Cholesky, QR, UTV, SVD, Bidiagonalization	187
11.7 Application: PCA via Spectral Decomposition and SVD	193
11.8 Application: Low-Rank Approximation	195
Chapter 11 Problems	197
VI Special Topics	200
12 Alternating Least Squares (ALS)	202
12.1 Preliminary: Least Squares Approximations	203
12.2 Netflix Recommender and Matrix Factorization	207
12.3 Regularization and Identifiability: Extension to General Matrices	213
12.4 Missing Entries and Rank-One Update	215
12.5 Vector Inner Product and Hidden Vectors	217
12.6 Gradient Descent	218
12.7 Regularization: A Geometric Interpretation	221
12.8 Stochastic Gradient Descent	223
12.9 Bias Term	224
12.10 Low-Rank Hadamard Decomposition	226
12.11 Application: Movie Recommender	229
12.12 Application: Outlier Detection	232
12.13 Application: Spectral Clustering and Link Prediction	232
Chapter 12 Problems	232
13 Nonnegative Matrix Factorization (NMF)	238
13.1 Nonnegative Matrix Factorization	238
13.2 NMF via Alternating Projected Gradient Descent (APGD)	240
13.3 NMF via Alternating Nonnegative Least Squares (ANLS)	241
13.4 NMF via Hierarchical Alternating Nonnegative Least Squares	242
13.5 NMF via Alternating Direction Methods of Multipliers (ADMM)	243
13.6 NMF via Multiplicative Update (MU)	245
13.7 NMF with Three Factors	250
13.8 β -Divergence, Alternative Perspectives of MU	250
13.9 Movie Recommender Context	257
13.10 Other Applications	257
Chapter 13 Problems	260
14 Biconjugate Decomposition	262
14.1 Existence of the Biconjugate Decomposition	262
14.2 Properties of the Biconjugate Decomposition	268
14.3 Connection to Well-Known Decomposition Methods	269
14.4 Proof: General Term Formula of Wedderburn Sequence	272
Chapter 14 Problems	273

Introduction and Background

Matrix decomposition is a cornerstone of modern numerical linear algebra, with applications in diverse fields such as statistics (Banerjee and Roy, 2014; Gentle, 1998), optimization (Gill et al., 2021), and machine learning (Goodfellow et al., 2016; Bishop, 2006), particularly in deep learning. As an essential computational framework, it simplifies complex matrix operations by breaking a matrix into more manageable components. This approach is critical not only for theoretical insights but also for practical implementations, enabling efficient algorithms and enhancing interpretability.

The prominence of matrix decomposition techniques is largely due to advances like the backpropagation algorithm for neural network training and the use of low-rank neural networks in efficient deep learning architectures (Lu et al., 2025). The primary goal of this book is to provide a self-contained introduction to the concepts and mathematical tools of linear algebra and matrix analysis, laying a solid foundation for understanding matrix decomposition techniques and their applications in subsequent sections. This book explores the fundamental techniques and applications of matrix decomposition. It begins with foundational methods such as LU and Cholesky decomposition, which are integral to solving linear systems and understanding positive definiteness. It then delves into more advanced topics, including QR decomposition, spectral decomposition, and singular value decomposition (SVD), which have broad applications ranging from eigenvalue problems to low-rank approximations and data compression. This introduction is designed for readers with a foundational knowledge of linear algebra and aims to bridge the gap between theory and application, equipping them with the necessary tools to navigate this critical area of numerical mathematics.

However, we clearly realize our inability to cover all the useful and interesting results concerning matrix decomposition. Given the scope limitations, topics such as the analysis of Euclidean space, Hermitian space, and Hilbert space are not addressed in detail here. For a more comprehensive introduction to these areas, readers are encouraged to consult the literature on linear algebra, including works such as Trefethen and Bau III (1997); Strang (2009); Stewart (2000); Gentle (2007); Higham (2002a); Quarteroni et al. (2010); Golub and Van Loan (2013); Beck (2017); Gallier and Quaintance (2019); Boyd and Vandenberghe (2018); Strang (2019); van de Geijn and Myers (2020); Strang (2021). It is important to note that this book specifically focuses on providing compact proofs for the existence of various matrix decomposition methods. For a more in-depth exploration of topics such as reducing computational complexity, detailed discussions of applications, and insights into tensor decomposition, readers are encouraged to refer to Lu (2021c).

A matrix decomposition involves breaking down a complex matrix into its constituent parts, simplifying its representation. The underlying principle of this approach is that, rather than solving specific problems directly, matrix algorithms focus on simplifying more complex matrix operations. These operations can be performed on the decomposed components, rather than the original matrix itself. At a general level, a matrix decomposition task for a matrix \mathbf{A} can be formulated as follows:

- $\mathbf{A} = \mathbf{Q}\mathbf{U}$: Here, \mathbf{Q} is an orthogonal matrix that contains the same column space as \mathbf{A} , while \mathbf{U} is a relatively simple and sparse matrix used to reconstruct \mathbf{A} .
- $\mathbf{A} = \mathbf{Q}\mathbf{T}\mathbf{Q}^\top$: In this case, \mathbf{Q} is orthogonal such that \mathbf{A} and \mathbf{T} are *similar matrices*¹ that share essential properties such as eigenvalues and sparsity. Additionally, working with \mathbf{T} is computationally simpler than working with \mathbf{A} .

¹. See Definition 7.3 for a rigorous definition.

- $\mathbf{A} = \mathbf{UTV}$: In this formulation, \mathbf{U} and \mathbf{V} are orthogonal matrices such that the columns of \mathbf{U} and the rows of \mathbf{V} form orthonormal bases for the column space and row space of \mathbf{A} , respectively.
- $\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{B}} \underset{r \times n}{\mathbf{C}}$: Here, \mathbf{B} and \mathbf{C} are full-rank matrices capable of reducing the memory storage requirements for \mathbf{A} . In practical applications, a low-rank approximation, $\underset{m \times n}{\mathbf{A}} \approx \underset{m \times k}{\mathbf{D}} \underset{k \times n}{\mathbf{F}}$, where $k < r$ is the *numerical rank* of the matrix, proves beneficial. This approximation allows for more efficient storage of the matrix \mathbf{A} , requiring only $k(m + n)$ floats instead of mn numbers. Additionally, it facilitates the efficient computation of matrix-vector products, $\mathbf{b} = \mathbf{Ax}$, through intermediate steps involving $\mathbf{c} = \mathbf{Fx}$ and $\mathbf{b} = \mathbf{Dc}$. This approximation method is also valuable for data interpretation and other computational tasks.
- Although typically computationally demanding, a matrix decomposition can be leveraged to solve new problems related to the original matrix in various contexts. For instance, once the factorization of \mathbf{A} is obtained, it can be reused to solve a 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, matrix decomposition aids in understanding the internal structure and logic of operations involving matrix multiplication. Each component of the decomposition contributes to a geometrical transformation, as discussed in Section 11.5.

Matrix decomposition algorithms can be classified into several categories. Below are six fundamental types:

1. Factorizations based on Gaussian elimination, such as LU decomposition and its positive definite counterpart, Cholesky decomposition.
2. Factorizations achieved by orthogonalizing either the columns or rows of a matrix, enabling effective data representation in an orthonormal basis.
3. Factorizations involving skeleton matrices, where a subset of columns or rows can sufficiently represent the entire dataset with minimal reconstruction error, while preserving sparsity and nonnegativity.
4. Reduction to Hessenberg, tridiagonal, or bidiagonal forms, allowing the properties of the matrix (such as rank and eigenvalues) to be explored within these reduced forms.
5. Factorizations derived from the computation of matrix eigenvalues.
6. Other specialized methods, which involve optimization techniques and high-level concepts. These may not fit neatly into the categories above but still represent important classes of decompositions.

The visual representations of matrix decomposition in Figures 1 and 2 illustrate the connections between various decomposition methods based on their underlying relationships. These figures also distinguish the methods according to specific criteria or prerequisites. Further details about these visualizations are provided in the accompanying text.

Objectives of This Work

This book aims to provide a comprehensive yet accessible introduction to the principles, methods, and applications of matrix decomposition. Designed for readers with a foundational understanding of linear algebra, it bridges the gap between theoretical rigor and practical applications. The goals include:

1. Presenting core concepts: Introducing the mathematical foundations of matrix decomposition, including LU, Cholesky, QR, and SVD, along with more advanced methods like eigenvalue and Jordan decompositions.

-
2. Highlighting applications: Demonstrating the relevance of these techniques in various domains, such as optimization, machine learning, and signal processing.
 3. Providing rigorous proofs: Ensuring that the presented methods are mathematically rigorous, with proofs and derivations to deepen understanding.

Notation and preliminaries. In the remainder of this section, we introduce and review fundamental concepts from linear algebra. We will also introduce additional important notions as necessary to ensure clarity. Throughout the text, our focus will be on real matrices. Unless otherwise specified, the eigenvalues of the matrices under discussion are assumed to be real as well.

Scalars are represented in non-bold font, potentially with subscripts (e.g., a , α , α_i). Vectors are denoted using **boldface** lowercase letters, possibly with subscripts (e.g., μ , x , x_n , z), while matrices are represented by **boldface** uppercase letters, possibly with subscripts (e.g., A , L_j). The i -th element of a vector z is written as z_i in non-bold font. For a matrix A , the value in the i -th row and j -th column is represented as a_{ij} . Additionally, we also adopt **Matlab-style notation**: the submatrix of A from the i -th to j -th rows and k -th to m -th columns is denoted by $A_{i:j,k:m} = A[i : j, k : m]$. When the indices are not continuous, with ordered subindex sets \mathbb{I} and \mathbb{J} , $A[\mathbb{I}, \mathbb{J}]$ indicates the submatrix of A obtained by extracting the rows and columns indexed by \mathbb{I} and \mathbb{J} , respectively. Similarly, $A[:, \mathbb{J}]$ denotes the submatrix of A obtained by extracting the columns of A indexed by \mathbb{J} .

All vectors are represented in column format rather than row format. A row vector is indicated by the transpose of a column vector, e.g., denoted by a^\top . A specific column vector with values is delineated by the semicolon symbol “;”, for example, $x = [1; 2; 3]$ is a column vector in \mathbb{R}^3 . Similarly, a row vector with specific values is separated by commas, e.g., $y = [1, 2, 3]$ is a row vector with three values. Furthermore, a column vector can be expressed as the transpose of a row vector, for instance, $y = [1, 2, 3]^\top$ is a column vector.

The transpose of a matrix A is denoted by A^\top , and its inverse is denoted by A^{-1} . The $p \times p$ identity matrix is denoted by I_p . A vector or matrix consisting entirely of zeros is denoted by the **boldface** zero, $\mathbf{0}$, with its size inferred from context. Specifically, $\mathbf{0}_p$ signifies a vector of all zeros with p entries, and $\mathbf{0}_{p \times q}$ represents a matrix of all zeros with dimensions $p \times q$.

Definition 0.1 (Eigenvalue and eigenvector). Given any vector space \mathbb{F} and any linear map $A : \mathbb{F} \rightarrow \mathbb{F}$ (or simply a real matrix $A \in \mathbb{R}^{n \times n}$), a scalar $\lambda \in \mathbb{K}$ is called a (*right*) *eigenvalue, or proper value, or characteristic value* of A , if there exists a nonzero vector $u \in \mathbb{F}$ such that

$$Au = \lambda u.$$

And u is called a (*right*) *eigenvector* of A associated with λ .

On the other hand, κ is referred to as a *left eigenvalue* if there exists a nonzero vector $v \in \mathbb{F}$ such that

$$v^\top A = \kappa v^\top.$$

And v is called a (*left*) *eigenvector* of A associated with κ .

When it is clear from the context, we will simply use the term “eigenvalue/eigenvector” instead of “right eigenvalue/eigenvector.”

In simple terms, an eigenvector u of a matrix A represents a direction that remains unchanged when transformed into the coordinate system defined by the columns of A (see Section 11.5 for more details on coordinate transformations). In fact, real-valued matrices

can have complex eigenvalues. However, all the eigenvalues of symmetric matrices are real (see Theorem 10.1).

Definition 0.2 (Spectrum and spectral radius). The set of all eigenvalues of \mathbf{A} is called the *spectrum* of \mathbf{A} and is denoted by $\Lambda(\mathbf{A})$. The largest magnitude of the eigenvalues is known as the *spectral radius* $\rho(\mathbf{A})$:

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

Moreover, the pair (λ, \mathbf{u}) mentioned above is commonly referred to as an *eigenpair*. Intuitively, the above definitions indicate that multiplying the matrix \mathbf{A} by the vector \mathbf{u} yields a new vector that lies in the same direction as \mathbf{u} , but scaled by a factor λ . For any eigenvector \mathbf{u} , it can be scaled by a scalar s such that $s\mathbf{u}$ remains an eigenvector of \mathbf{A} . This is why we refer to \mathbf{u} as an eigenvector of \mathbf{A} associated with the eigenvalue λ . To avoid any ambiguity, we usually assume that the eigenvector is normalized to have unit length, and its first entry is positive, since both \mathbf{u} and $-\mathbf{u}$ are valid eigenvectors.

In linear algebra, it is a fundamental property that every vector space has a basis. Any vector in the space can be expressed as a linear combination of the basis vectors. Using this concept, we define the *span* and *dimension* of a subspace in terms of its basis.

Definition 0.3 (Subspace). A nonempty subset \mathcal{V} of \mathbb{R}^n is called a *subspace* if for all $\mathbf{a}, \mathbf{b} \in \mathcal{V}$ and all $x, y \in \mathbb{R}$, the linear combination $x\mathbf{a} + y\mathbf{b}$ also belongs to \mathcal{V} .

Definition 0.4 (Span). If every vector \mathbf{v} in a subspace \mathcal{V} can be expressed as a linear combination of the vectors $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$, then the set $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ is said to span \mathcal{V} .

In linear algebra, the concept of linear independence is fundamental when studying sets of vectors. Two equivalent definitions are provided below.

Definition 0.5 (Linearly independent). A set of vectors $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ is called *linearly independent* if the equation $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \dots + x_m\mathbf{a}_m = \mathbf{0}$ has only the trivial solution where all scalars $x_i = 0$. 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 the preceding vectors $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{k-1}\}$.

Exercise 0.6. Show that the columns of the $m \times n$ matrix \mathbf{A} are linearly independent if and only if $f(\mathbf{x}) = \mathbf{Ax}$ is a one-to-one function.

Definition 0.7 (Basis and dimension). A set of vectors $\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_m\}$ is called a *basis* of a subspace \mathcal{V} if they are linearly independent and span \mathcal{V} . All bases of a given subspace contain the same number of vectors, and this common number of vectors in any basis is called the *dimension* of the subspace \mathcal{V} .

By convention, the subspace containing only the zero vector, $\{\mathbf{0}\}$, has dimension zero. Furthermore, every nonzero subspace has a basis consisting of mutually *orthogonal* vectors (i.e., the vectors in the basis are mutually perpendicular).

Definition 0.8 (Column space (range)). For an $m \times n$ real matrix \mathbf{A} , the *column space (or range)* of \mathbf{A} is defined as the set of all linear combinations of its columns:

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

Similarly, the *row space* of \mathbf{A} is the set of all linear combinations of its rows, which is equal to the column space of the transpose \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 0.9 (Null space (nullspace, kernel)). For an $m \times n$ real matrix \mathbf{A} , the *null space* (also called the *kernel or nullspace*) of \mathbf{A} is the set of all vectors in \mathbb{R}^n that satisfy:

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

Similarly, the null space of \mathbf{A}^\top (i.e., the *left null space* of \mathbf{A}) 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 . Moreover, every vector in $\mathcal{N}(\mathbf{A}^\top)$ is orthogonal to $\mathcal{C}(\mathbf{A})$, and vice versa; similarly, every vector in $\mathcal{N}(\mathbf{A})$ is also orthogonal to $\mathcal{C}(\mathbf{A}^\top)$, and vice versa.

Definition 0.10 (Rank). 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 maximum number of linearly independent columns of \mathbf{A} , and is also the maximum number of linearly independent rows of \mathbf{A} . The rank of \mathbf{A} is equal to the rank of its transpose, \mathbf{A}^\top . Additionally, \mathbf{A} is said to have full rank if its rank equals $\min\{m, n\}$. 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{uv}^\top is of rank 1. In short, the rank of a matrix is equal to:

- the number of linearly independent columns;
- the number of linearly independent rows;
- and remarkably, these two quantities are always equal (see Theorem 0.15).

Exercise 0.11 (Rank of matrix addition). Let \mathbf{A} and \mathbf{B} be two matrices with ranks a and b , respectively. Show that the rank of $\mathbf{A} + \mathbf{B}$ is at most $a + b$ and at least $|a - b|$.

Exercise 0.12 (Rank of matrix multiplication, a.k.a., Sylvester's inequality). Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times p}$ be two matrices with ranks a and b , respectively. Show that the rank of \mathbf{AB} is at most $\min\{a, b\}$ and at least $a + b - n$.

Definition 0.13 (Orthogonal complement in general). The orthogonal complement \mathcal{V}^\perp of a subspace \mathcal{V} consists of all vectors that are perpendicular to \mathcal{V} . Formally,

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

The two subspaces are *disjoint* (i.e., their intersection is $\{\mathbf{0}\}$) and together span the entire space. The dimensions of \mathcal{V} and \mathcal{V}^\perp add up to the dimension of the full space.

Furthermore, taking the orthogonal complement twice returns the original subspace: $(\mathcal{V}^\perp)^\perp = \mathcal{V}$.

For example, we can explicitly define the orthogonal complement of the column space as follows:

Definition 0.14 (Orthogonal complement of column space). For an $m \times n$ real matrix \mathbf{A} , the orthogonal complement of its column space $\mathcal{C}(\mathbf{A})$, denoted by $\mathcal{C}^\perp(\mathbf{A})$, is the subspace:

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

We now introduce the *four fundamental subspaces* associated with any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ of rank r , as outlined in Theorem 0.17. To establish this fundamental theorem of linear algebra, we first need to verify a key result: the equality of the row rank and column rank of a matrix.

Theorem 0.15: (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. In other words, the row rank and the column rank of a matrix \mathbf{A} are equal.

Proof [of Theorem 0.15] We begin by observing that the null space of \mathbf{A} is orthogonal to the row space of \mathbf{A} : $\mathcal{N}(\mathbf{A}) \perp \mathcal{C}(\mathbf{A}^\top)$ (where the row space of \mathbf{A} corresponds to 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 $\mathbf{A} = [\mathbf{a}_1^\top; \mathbf{a}_2^\top; \dots; \mathbf{a}_m^\top]$ is the row partition. For any vector $\mathbf{x} \in \mathcal{N}(\mathbf{A})$, we have $\mathbf{A}\mathbf{x} = \mathbf{0}$, or equivalently, $[\mathbf{a}_1^\top \mathbf{x}; \mathbf{a}_2^\top \mathbf{x}; \dots; \mathbf{a}_m^\top \mathbf{x}] = \mathbf{0}$. Since the row space of \mathbf{A} is spanned by $\{\mathbf{a}_1^\top, \mathbf{a}_2^\top, \dots, \mathbf{a}_m^\top\}$, it follows that \mathbf{x} is perpendicular to all vectors in $\mathcal{C}(\mathbf{A}^\top)$, which means $\mathcal{N}(\mathbf{A}) \perp \mathcal{C}(\mathbf{A}^\top)$.

Next, suppose the dimension of the 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 $\{\mathbf{A}\mathbf{r}_1, \mathbf{A}\mathbf{r}_2, \dots, \mathbf{A}\mathbf{r}_r\}$ lie in the column space of \mathbf{A} . We claim that these vectors are linearly independent. To verify this, suppose there exists a linear combination of the r vectors: $x_1 \mathbf{A}\mathbf{r}_1 + x_2 \mathbf{A}\mathbf{r}_2 + \dots + x_r \mathbf{A}\mathbf{r}_r = \mathbf{0}$, 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 $\mathbf{v} = x_1 \mathbf{r}_1 + x_2 \mathbf{r}_2 + \dots + x_r \mathbf{r}_r$ belongs to the 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} must also lie in the row space of \mathbf{A} . We have shown that vectors from the null space of \mathbf{A} are perpendicular to vectors from the row space of \mathbf{A} ; thus, it holds that $\mathbf{v}^\top \mathbf{v} = 0$, which implies that $x_1 = x_2 = \dots = x_r = 0$. Hence, $\mathbf{A}\mathbf{r}_1, \mathbf{A}\mathbf{r}_2, \dots, \mathbf{A}\mathbf{r}_r$ lie in the column space of \mathbf{A} , and they are linearly independent. Since these r linearly independent vectors are in the column space of \mathbf{A} , the column space must have dimension at least r . This proves that **row rank of $\mathbf{A} \leq$ column rank of \mathbf{A}** .

Applying the same reasoning to \mathbf{A}^\top , we conclude that **column rank of $\mathbf{A} \leq$ row rank of \mathbf{A}** . Combining these results, we obtain the equality of the row rank and column rank of \mathbf{A} . This completes the proof. ■

Additional insights from this proof reveal that if $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r\}$ forms a basis for the row space of $\mathbf{A} \in \mathbb{R}^{m \times n}$, then $\{\mathbf{A}\mathbf{r}_1, \mathbf{A}\mathbf{r}_2, \dots, \mathbf{A}\mathbf{r}_r\}$ constitutes a basis for the column space of \mathbf{A} . This result is formalized in the following lemma:

Lemma 0.16: (Column basis from row basis) For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, 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 of \mathbf{A} , then $\{\mathbf{A}\mathbf{r}_1, \mathbf{A}\mathbf{r}_2, \dots, \mathbf{A}\mathbf{r}_r\}$ forms a basis for the column space of \mathbf{A} .

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 orthogonal to any vector in the null space of \mathbf{A} . Specifically, if $\mathbf{x}_n \in \mathcal{N}(\mathbf{A})$, then $\mathbf{A}\mathbf{x}_n = \mathbf{0}$, which implies that \mathbf{x}_n is perpendicular to every row of \mathbf{A} , thus supporting this assertion.

Similarly, any vector in the column space of \mathbf{A} is orthogonal to any vector in the null space of \mathbf{A}^\top . Moreover, the column space of \mathbf{A} together with the null space of \mathbf{A}^\top span the entire space \mathbb{R}^m . This observation is a key part of the fundamental theorem of linear algebra.

The fundamental theorem consists of two essential components: the dimensions of the subspaces and the orthogonality relationships between pairs of subspaces. The orthogonality relationships have already been demonstrated above. Additionally, when the row space has dimension r , the null space has dimension $n - r$. These relationships are rigorously established in the following theorem.

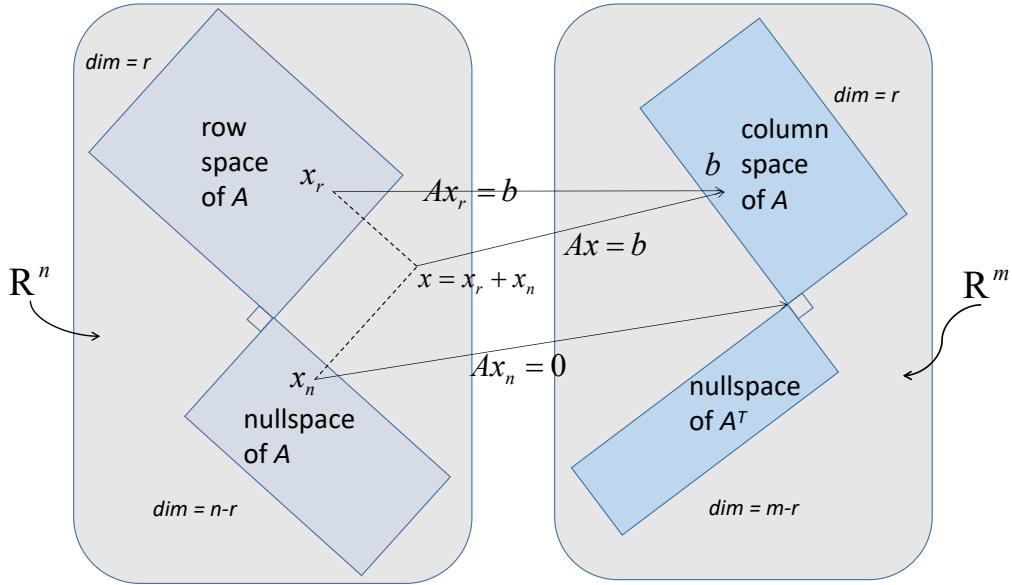


Figure 3: Two pairs of orthogonal subspaces in \mathbb{R}^n and \mathbb{R}^m . $\dim(\mathcal{C}(\mathbf{A}^\top)) + \dim(\mathcal{N}(\mathbf{A})) = n$ and $\dim(\mathcal{N}(\mathbf{A}^\top)) + \dim(\mathcal{C}(\mathbf{A})) = m$. The null space component maps to zero as $\mathbf{A}\mathbf{x}_n = \mathbf{0} \in \mathbb{R}^m$. The row space component maps to the column space as $\mathbf{A}\mathbf{x}_r = \mathbf{A}(\mathbf{x}_r + \mathbf{x}_n) = \mathbf{b} \in \mathcal{C}(\mathbf{A})$.

Theorem 0.17: (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

- The null space $\mathcal{N}(\mathbf{A})$ is orthogonal complement to the row space $\mathcal{C}(\mathbf{A}^\top)$ in \mathbb{R}^n : $\dim(\mathcal{N}(\mathbf{A})) + \dim(\mathcal{C}(\mathbf{A}^\top)) = n$;
- The left null space $\mathcal{N}(\mathbf{A}^\top)$ 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 a 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 0.17] From the proof of Theorem 0.15, let $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r\}$ be a set of vectors in \mathbb{R}^n that 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} . 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 Theorem 0.15, $\mathcal{N}(\mathbf{A}) \perp \mathcal{C}(\mathbf{A}^\top)$, 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} lies in the column space of \mathbf{A} , so it can be written as a linear combination of $\mathbf{Ar}_1, \mathbf{Ar}_2, \dots, \mathbf{Ar}_r$: $\mathbf{Ax} = \sum_{i=1}^r a_i \mathbf{Ar}_i$. This implies 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 represented as a linear 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 represented 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 dimensions satisfy: $r + k = n$, i.e., $\dim(\mathcal{N}(\mathbf{A})) + \dim(\mathcal{C}(\mathbf{A}^\top)) = n$. Similarly, we can prove that $\dim(\mathcal{N}(\mathbf{A}^\top)) + \dim(\mathcal{C}(\mathbf{A})) = m$. ■

Figure 3 illustrates two pairs of such orthogonal subspaces and demonstrates how \mathbf{A} maps \mathbf{x} into the column space. The dimensions of the row space of \mathbf{A} and the null space of \mathbf{A} sum to n . And the dimensions of the column space of \mathbf{A} and the null space of \mathbf{A}^\top add up to m . The null space component is mapped to zero, as $\mathbf{Ax}_n = \mathbf{0} \in \mathbb{R}^m$, which is the intersection of the column space of \mathbf{A} and the null space of \mathbf{A}^\top . Conversely, the row space component is mapped to the column space, as $\mathbf{Ax}_r = \mathbf{A}(\mathbf{x}_r + \mathbf{x}_n) = \mathbf{b} \in \mathbb{R}^m$.

Definition 0.18 (Orthogonal matrix). A real square matrix \mathbf{Q} is called an *orthogonal matrix* if its inverse equals its transpose, that is, $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ and $\mathbf{QQ}^\top = \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$. Equivalently, 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, $\mathbf{q}_i^\top \mathbf{q}_j = \delta(i, j)$, where $\delta(i, j)$ is the Kronecker delta function. If \mathbf{Q} contains only γ of these columns with $\gamma < n$, the relation $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_\gamma$ still holds, where \mathbf{I}_γ is the $\gamma \times \gamma$ identity matrix. However, in this case, the equation $\mathbf{QQ}^\top = \mathbf{I}$ no longer holds; and \mathbf{Q} is known as a *semi-orthogonal matrix*. An orthogonal matrix also preserves the length of any vector \mathbf{x} , i.e., $\|\mathbf{Q}\mathbf{x}\| = \|\mathbf{x}\|$ (see Definition 0.26).

Definition 0.19 (Permutation matrix). A permutation matrix \mathbf{P} is a square binary matrix with exactly one entry of 1 in each row and each column; and all other entries are 0.

Row perspective. A permutation matrix \mathbf{P} can be viewed as having the rows of the identity matrix \mathbf{I} arranged in a specific order. This order determines the sequence of row permutations. To permute the rows of a matrix \mathbf{A} , multiply \mathbf{A} on the left by \mathbf{P} , yielding \mathbf{PA} .

Column perspective. Alternatively, \mathbf{P} can be viewed as having the columns of the identity matrix \mathbf{I} rearranged. This order determines the sequence of column permutations. To permute the columns of \mathbf{A} , multiply \mathbf{A} on the right by \mathbf{P} , yielding \mathbf{AP} .

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

Example 0.20 (Permutation). Let $\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ and $\mathbf{P} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$. The row permutation and the column permutation are given by $\mathbf{PA} = \begin{bmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \end{bmatrix}$ and $\mathbf{AP} = \begin{bmatrix} 3 & 1 & 2 \\ 6 & 4 & 5 \\ 9 & 7 & 8 \end{bmatrix}$, 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 order of the columns of \mathbf{A} appearing in \mathbf{AP} matches the order of the columns of \mathbf{I} in \mathbf{P} . \square

Geometrically, the determinant of an $n \times n$ matrix \mathbf{A} , denoted by $\det(\mathbf{A})$, is the (signed) volume of the n -dimensional parallelepiped defined by its row (or column) vectors. For example, an orthogonal matrix always forms a unit hypercube, and so the absolute value of its determinant is always 1. The above definition is self-consistent because the volume defined by the row vectors and the volume defined by the column vectors of a square matrix can be mathematically shown to be the same. We can recursively define the determinant of a matrix as follows:

Definition 0.21 (Determinant: Laplace expansion by minors). Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be any square matrix, and let $\mathbf{A}_{ij} \in \mathbb{R}^{(n-1) \times (n-1)}$ denote the submatrix of \mathbf{A} obtained by deleting the i -th row and j -th column. The *determinant* of \mathbf{A} can be computed recursively using the following equations:

$$\det(\mathbf{A}) = \sum_{k=1}^n (-1)^{i+k} a_{ik} \det(\mathbf{A}_{ik}) = \sum_{k=1}^n (-1)^{k+j} a_{kj} \det(\mathbf{A}_{kj}), \quad (0.1)$$

where the first equation is the *Laplace expansion by minors along row i* , and the second equation is the *Laplace expansion by minors along column j* . Equivalently, given a cardinality r , and consider an index set $\mathbb{J} \subseteq \{1, 2, \dots, n\}$ with cardinality r ($|\mathbb{J}| = r$) and its complementary set $\mathbb{J}^C = \{1, 2, \dots, n\} \setminus \mathbb{J}$. Then we have:

$$\det(\mathbf{A}) = \sum_{\mathbb{I}} (-1)^\gamma \det(\mathbf{A}[\mathbb{I}, \mathbb{J}]) \det(\mathbf{A}[\mathbb{I}^C, \mathbb{J}^C]) = \sum_{\mathbb{I}} (-1)^\gamma \det(\mathbf{A}[\mathbb{J}, \mathbb{I}]) \det(\mathbf{A}[\mathbb{J}^C, \mathbb{I}^C]),$$

where $\gamma = \sum_{i \in \mathbb{I}} i + \sum_{j \in \mathbb{J}} j$, and the sum is taken over all the index sets $\mathbb{I} \subseteq \{1, 2, \dots, n\}$ with cardinality r . When $r = 1$, this reduces to (0.1).

Remark 0.22 (Determinant: alternating sums and permutations). Let the function $p : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ be a one-to-one function of permutations, i.e., $p(i) = i$ in the identity case. Then, there are $n!$ distinct permutations of the set $\{1, 2, \dots, n\}$. For a given permutation, let $\text{sgn}(p) = 1$ if the minimum number of transpositions to achieve this permutation is even; and $\text{sgn}(p) = -1$ otherwise. Then, the determinant can be equivalently defined as $\det(\mathbf{A}) = \sum_p (\text{sgn}(p) \prod_{i=1}^n a_{ip(i)})$.

A quantity closely related to the determinant is the adjugate of a matrix, which we now define.

Definition 0.23 (Adjugate). Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be any square matrix. Then, the *adjugate* of \mathbf{A} , denoted $\text{adj}(\mathbf{A})$, is an $n \times n$ matrix whose (i, j) -th element is given by

$$\text{adj}(\mathbf{A})_{ij} = (-1)^{i+j} \det(\mathbf{A}[\{j\}^C, \{i\}^C]), \quad (0.2)$$

where $\{i\}^C$ is the complementary set of $\{1, 2, \dots, n\}$: $\{i\}^C = \{1, 2, \dots, n\} \setminus i$. Comparing this with the definition of determinants, we have

$$\text{adj}(\mathbf{A})\mathbf{A} = \mathbf{A}\text{adj}(\mathbf{A}) = \det(\mathbf{A})\mathbf{I}. \quad (0.3)$$

This shows that $\text{adj}(\mathbf{A})$ is nonsingular (resp., upper triangular, diagonal) if \mathbf{A} is nonsingular (resp., upper triangular, diagonal):

$$\text{adj}(\mathbf{A}) = \det(\mathbf{A})\mathbf{A}^{-1}. \quad (0.4)$$

If \mathbf{A} is singular, then every column of \mathbf{A} lies in the null space of $\text{adj}(\mathbf{A})$: the dimension of the null space of $\text{adj}(\mathbf{A})$ is at least the rank of \mathbf{A} .

For example, $\text{adj}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$. Additional properties of the adjugate of a matrix are discussed in Problem 6.9, where we introduce the *interpolative decomposition* of a matrix.

From an introductory linear algebra course, we observe the following equivalences regarding nonsingular matrices.

Remark 0.24 (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 of \mathbf{A} are nonzero.

It is important to keep these equivalences in mind, as misunderstanding them can easily lead to confusion. On the other hand, the following remark outlines the corresponding set of equivalent conditions for singular matrices.

Remark 0.25 (List of equivalence of singularity for a matrix). For a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ with an 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 $(\mathbf{A} - \lambda \mathbf{I})$ are linearly dependent;
- Rows of $(\mathbf{A} - \lambda \mathbf{I})$ are linearly dependent;
- $(\mathbf{A} - \lambda \mathbf{I})$ has rank $r < n$;
- Dimension of column space = dimension of row space = $r < n$;
- $(\mathbf{A} - \lambda \mathbf{I})^\top (\mathbf{A} - \lambda \mathbf{I})$ is symmetric semidefinite;
- $(\mathbf{A} - \lambda \mathbf{I})$ has $r < n$ nonzero (positive) singular values;
- Zero is an eigenvalue of $(\mathbf{A} - \lambda \mathbf{I})$.

Norms provide a measure of the magnitude of a vector or matrix, which is useful in many applications, such as determining the length of a vector in Euclidean space or the size of a matrix in a multidimensional setting. Additionally, norms enable us to define distances between vectors or matrices. The distance between two vectors \mathbf{u} and \mathbf{v} can be computed using the norm of their difference $\|\mathbf{u} - \mathbf{v}\|$. This is critical for tasks involving proximity measures, such as clustering algorithms in machine learning.

For a vector $\mathbf{x} \in \mathbb{R}^n$, we define the vector ℓ_2 norm as follows.

Definition 0.26 (Vector ℓ_2 norm). For a vector $\mathbf{x} \in \mathbb{R}^n$, the ℓ_2 vector norm is defined as $\|\mathbf{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$.

For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, we define the (matrix) Frobenius norm as follows.

Definition 0.27 (Matrix Frobenius norm). The *Frobenius norm* of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is defined as

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i=1, j=1}^{m, n} (a_{ij})^2} = \sqrt{\text{tr}(\mathbf{A}\mathbf{A}^\top)} = \sqrt{\text{tr}(\mathbf{A}^\top \mathbf{A})} = \sqrt{\sigma_1^2 + \sigma_2^2 + \dots + \sigma_r^2},$$

where $\sigma_1, \sigma_2, \dots, \sigma_r$ are the nonzero singular values of \mathbf{A} (see Section 11.1). The squared Frobenius norm of a matrix is often referred to as the *energy* of the matrix in the machine learning community.

The Frobenius norm can be interpreted as the ℓ_2 norm applied to the vectorized form of the matrix. Additionally, the spectral norm of a matrix is defined as follows.

Definition 0.28 (Matrix spectral norm). The *spectral norm* of a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is defined as

$$\|\mathbf{A}\|_2 = \max_{\mathbf{x} \neq 0} \frac{\|\mathbf{A}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \max_{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2=1} \|\mathbf{A}\mathbf{x}\|_2,$$

which corresponds to the largest singular value of \mathbf{A} , i.e., $\|\mathbf{A}\|_2 = \sigma_{\max}(\mathbf{A})$. The definition also implies the inequality: $\|\mathbf{A}\mathbf{x}\|_2 \leq \sigma_{\max}(\mathbf{A}) \|\mathbf{x}\|_2$ for any $\mathbf{x} \in \mathbb{R}^n$.

For simplicity, we will not always explicitly indicate the full subscript for both the vector ℓ_2 norm and the matrix Frobenius norm when it is clear from the context which one we are referring to; that is, we may write $\|\mathbf{A}\| = \|\mathbf{A}\|_F$ and $\|\mathbf{x}\| = \|\mathbf{x}\|_2$.

Part I

Gaussian Elimination

1

LU Decomposition

1.1. LU Decomposition

One of the most well-known and foundational matrix decompositions is the LU decomposition. The details are outlined in the following theorem, and the proof of its existence will be discussed in subsequent sections.¹

Theorem 1.1: (LU decomposition with permutation) Let \mathbf{A} be a nonsingular $n \times n$ square matrix. Then, it can be decomposed 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., a lower triangular matrix with all 1's on the diagonal), and \mathbf{U} is a *nonsingular* upper triangular matrix.

In certain cases, the use of the permutation matrix is unnecessary. This decomposition depends on the (leading) principal minors. We provide a precise definition, which is crucial for the subsequent illustration.

Definition 1.2 (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} .

Definition 1.3 (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 the 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 the k -th order *leading principal minor* of \mathbf{A} .

If the leading principal minors of matrix \mathbf{A} satisfy mild conditions, the LU decomposition does not require a permutation matrix, which we now recall in the following theorem:

¹ Note that, in the subsequent text, decomposition-related results will be presented in blue boxes, while other claims and theorems will be in gray boxes. This convention will be consistently applied throughout the remainder of the book without further notification.

Theorem 1.4: (LU decomposition without permutation) Let \mathbf{A} be an $n \times n$ square matrix with nonzero leading principal minors, i.e., $\det(\mathbf{A}_{1:k,1:k}) \neq 0$, for all $k \in \{1, 2, \dots, n\}$. Then, \mathbf{A} can be decomposed as

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

where \mathbf{L} is a unit lower triangular matrix (i.e., a 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 1.8.

In Theorem 1.4, we assume that the leading principal minors are nonzero, implying that the leading principal submatrices and the matrix \mathbf{A} are nonsingular. In the previous theorem, we also assumed that \mathbf{A} is nonsingular. However, it is important to note that an LU decomposition can still exist even when \mathbf{A} is singular. As will be explained in the next section, if \mathbf{A} is singular, some of the pivots during Gaussian elimination will be zero, resulting in corresponding zero diagonal entries in the matrix \mathbf{U} .

Even when \mathbf{A} is nonsingular, its leading principal submatrices may still be singular. Furthermore, if certain leading principal minors are zero, an LU decomposition may still exist, but it is no longer guaranteed to be unique under these conditions.

Additionally, LU decomposition can be generalized to handle non-square or singular matrices—examples include the *rank-revealing LU decomposition*. Interested readers are encouraged to consult Pan (2000); Miranian and Gu (2003); Dopico et al. (2006) for further details, or refer to Section 1.8 for a brief overview.

1.2. Relation to Gaussian Elimination

Solving the linear system equation $\mathbf{Ax} = \mathbf{b}$ is a fundamental problem in linear algebra. One widely used method for solving such systems is *Gaussian elimination*, which simplifies a linear system by transforming it into an upper triangular form through a sequence of *elementary row operations* (or *elementary row transformations*). This process unfolds over $n - 1$ stages for a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$. As a result, the system becomes much easier to solve using *backward substitution*. The elementary operations involved are formally defined as follows.

Definition 1.5 (Elementary transformations). Given a square matrix \mathbf{A} , the following three transformations are referred to as *elementary row (resp., column) transformations*:

1. Interchanging two rows (resp., columns) of \mathbf{A} .
2. Multiplying all elements of a single row (resp., column) of \mathbf{A} by a nonzero value.
3. Adding a multiple of one row (resp., column) to another row (resp., column).

Specifically, elementary row transformations of \mathbf{A} are represented by unit *lower* triangular matrices that act on the left of \mathbf{A} (e.g., \mathbf{EA}), while elementary column transformations are represented by unit *upper* triangular matrices that act on the right of \mathbf{A} (e.g., \mathbf{AE}).

Gaussian elimination is based on the third type of elementary row transformation listed above. Suppose the upper triangular matrix obtained through Gaussian elimination is given by $\mathbf{U} = \mathbf{E}_{n-1}\mathbf{E}_{n-2}\dots\mathbf{E}_1\mathbf{A}$ (which corresponds to $n - 1$ steps). And at the k -th stage ($k \leq n - 1$), consider the k -th column of $\mathbf{E}_{k-1}\mathbf{E}_{k-2}\dots\mathbf{E}_1\mathbf{A}$, denoted by $\mathbf{x} \in \mathbb{R}^n$. Gaussian elimination aims to introduce zeros below the diagonal of \mathbf{x} using a transformation of the

form

$$\mathbf{E}_k = \mathbf{I} - \mathbf{z}_k \mathbf{e}_k^\top, \quad (1.1)$$

where $\mathbf{e}_k \in \mathbb{R}^n$ is the k -th standard basis vector, and $\mathbf{z}_k \in \mathbb{R}^n$ is defined as

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

We observe that \mathbf{E}_k is a unit lower triangular matrix (with 1's on its diagonal), where only the entries below the diagonal in the k -th column are nonzero:

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

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

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

As an example, we outline the Gaussian elimination steps for a 4×4 matrix. For simplicity, we assume no row permutations. In the following matrix, \boxtimes represents a value that may not be zero, and **boldface** indicates the value has just been changed.

A Trivial Gaussian Elimination For a 4×4 Matrix:

$$\begin{array}{c} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \\ \mathbf{A} \end{array} \xrightarrow{\mathbf{E}_1} \begin{array}{c} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} \\ \mathbf{E}_1 \mathbf{A} \end{array} \xrightarrow{\mathbf{E}_2} \begin{array}{c} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} \\ \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} \end{array} \xrightarrow{\mathbf{E}_3} \begin{array}{c} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} \end{bmatrix} \\ \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} \end{array}, \quad (1.2)$$

where \mathbf{E}_1 , \mathbf{E}_2 , and \mathbf{E}_3 are lower triangular matrices. Specifically, as discussed earlier, 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 $\{k+1, k+2, \dots, n\}$ in order to create zeros below the diagonal in the k -th column of the matrix, without using rows $\{1, 2, \dots, k-1\}$.

To make this more concrete, consider stage 1 of the example above. We multiply on the left by \mathbf{E}_1 , which subtracts suitable multiples of the first row from rows 2, 3, and 4, resulting in zeros in the first entry of each of these rows. Similar operations occur at steps 2 and 3. By defining $\mathbf{L} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \mathbf{E}_3^{-1}$ and letting \mathbf{U} denote the matrix obtained after elimination,² we obtain the decomposition $\mathbf{A} = \mathbf{LU}$. Thus, we have constructed an LU decomposition for the 4×4 matrix \mathbf{A} .

². Unit lower triangular matrices have two important properties: their inverses are also unit lower triangular, and the product of such matrices results in another unit lower triangular matrix.

In the process of Gaussian elimination, we systematically eliminate entries below the diagonal to transform a matrix into an upper triangular form. A key element in guiding this elimination procedure is the first nonzero entry encountered in each row at every step. This special entry not only determines the feasibility of the elimination but also plays a crucial role in numerical stability. We now formally define this important concept.

Definition 1.6 (Pivot). The first nonzero entry in the row after each elimination step is referred to as a *pivot*. For example, the blue crosses in Equation (1.2) indicate the positions of the pivots.

However, the entry a_{11} (the (1,1) element of the matrix \mathbf{A}) may occasionally be zero. In such cases, no such elimination matrix \mathbf{E}_1 can successfully carry out the next elimination step. Therefore, we must swap the first and second rows using a permutation matrix \mathbf{P}_1 . This is known as *pivoting*, or simply *permutation*.

Gaussian Elimination With a Permutation in the Beginning:

$$\begin{array}{c} \left[\begin{array}{cccc} 0 & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{P}_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{E}_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{E}_2} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{E}_3} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & 0 & 0 & \textcolor{blue}{\boxtimes} \end{array} \right] \\ \mathbf{A} \qquad \mathbf{P}_1 \mathbf{A} \qquad \mathbf{E}_1 \mathbf{P}_1 \mathbf{A} \qquad \mathbf{E}_2 \mathbf{E}_1 \mathbf{P}_1 \mathbf{A} \qquad \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1 \mathbf{P}_1 \mathbf{A} \end{array}.$$

By defining $\mathbf{L} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \mathbf{E}_3^{-1}$ and $\mathbf{P} = \mathbf{P}_1^{-1}$, the expression $\mathbf{A} = \mathbf{PLU}$ represents a complete LU decomposition with permutation for the 4×4 matrix \mathbf{A} .

In certain cases, additional permutation matrices such as $\mathbf{P}_2, \mathbf{P}_3, \dots$ may be required between the lower triangular transformations \mathbf{E}_i 's. An example is provided below.

Gaussian Elimination With a Permutation in Between:

$$\begin{array}{c} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{E}_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{P}_1} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{E}_2} \left[\begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{\boxtimes} & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{\boxtimes} & \boxtimes \\ 0 & 0 & 0 & \textcolor{blue}{\boxtimes} \end{array} \right] \\ \mathbf{A} \qquad \mathbf{E}_1 \mathbf{A} \qquad \mathbf{P}_1 \mathbf{E}_1 \mathbf{A} \qquad \mathbf{E}_2 \mathbf{P}_1 \mathbf{E}_1 \mathbf{A} \end{array}.$$

In this scenario, we find that $\mathbf{U} = \mathbf{E}_2 \mathbf{P}_1 \mathbf{E}_1 \mathbf{A}$. In Section 1.4 or Section 1.7.1, we will demonstrate that incorporating interleaved permutations still leads to the form $\mathbf{A} = \mathbf{PLU}$, where \mathbf{P} accounts for all permutations performed.

The provided examples can be easily extended to any $n \times n$ matrix, assuming there are no row permutations involved. For such matrices, we apply $n - 1$ such lower triangular transformations. The k -th transformation, \mathbf{E}_k , introduces zeros below the diagonal in the k -th column of \mathbf{A} by subtracting multiples of the k -th row from rows $\{k + 1, k + 2, \dots, n\}$. Finally, by defining $\mathbf{L} = \mathbf{E}_1^{-1} \mathbf{E}_2^{-1} \dots \mathbf{E}_{n-1}^{-1}$, we obtain the LU decomposition $\mathbf{A} = \mathbf{LU}$ without the need for permutations.

From the examples above involving elementary row operations in the Gaussian elimination process, we can draw the following conclusion about the row spaces after performing (elementary) row transformations.

Proposition 1.7: (Row space after row operations) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be a matrix that undergoes a sequence of elementary row operations represented by $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_k$, and define $\mathbf{E} = \mathbf{E}_k \mathbf{E}_{k-1} \dots \mathbf{E}_1$. Then, the row space of $\mathbf{B} = \mathbf{EA}$ is identical to the row space of \mathbf{A} .

Proof [of Proposition 1.7] Since the rows of \mathbf{B} are linear combinations of the rows of \mathbf{A} , it follows that $\mathcal{C}(\mathbf{B}^\top) \subseteq \mathcal{C}(\mathbf{A}^\top)$. Moreover, since the row transformations are invertible, $\mathbf{E} = \mathbf{E}_k \mathbf{E}_{k-1} \dots \mathbf{E}_1$ is also invertible. Therefore, we can write: $\mathbf{A} = \mathbf{E}^{-1} \mathbf{B}$. This implies that the rows of \mathbf{A} are also linear combinations of the rows of \mathbf{B} : $\mathcal{C}(\mathbf{A}^\top) \subseteq \mathcal{C}(\mathbf{B}^\top)$. Combining the two results, we conclude that $\mathcal{C}(\mathbf{A}^\top) = \mathcal{C}(\mathbf{B}^\top)$. ■

Note, however, that the column spaces of \mathbf{A} and \mathbf{B} may differ. Nonetheless, since the dimension of the row space equals the dimension of the column space (i.e., the rank of the matrix), the dimensions of the column spaces of \mathbf{A} and \mathbf{B} are the same.

1.3. Existence of LU Decomposition without Permutation

Gaussian elimination, or Gaussian transformation, provides insight into the foundation of LU decomposition. We now rigorously prove Theorem 1.4, which establishes the existence of LU decomposition without permutation, using mathematical induction.

Proof [of Theorem 1.4: LU decomposition without permutation] We will prove by induction that every $n \times n$ square matrix \mathbf{A} with nonzero leading principal minors admits the LU decomposition of the form $\mathbf{A} = \mathbf{L}\mathbf{U}$. For the base case ($n = 1$), the result is trivial: set $\mathbf{L} = 1$ and $\mathbf{U} = A$ so that $A = \mathbf{L}\mathbf{U}$.

Now assume that any $k \times k$ matrix \mathbf{A}_k with all leading principal minors being nonzero has an LU decomposition without permutation. Our goal is to establish that any $(k+1) \times (k+1)$ matrix \mathbf{A}_{k+1} can also be expressed in this LU decomposition form without permutation.

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 $\mathbf{A}_k = \mathbf{L}_k \mathbf{U}_k$, where \mathbf{L}_k is a unit lower triangular matrix and \mathbf{U}_k is a nonsingular upper triangular matrix, as per the assumption. Express \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 following 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, we have

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

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

Because all the leading principal minors of \mathbf{A}_{k+1} are nonzero, we have $\det(\mathbf{A}_{k+1}) = \det(\mathbf{A}_k) \cdot \det(d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b}) = \det(\mathbf{A}_k) \cdot (d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b}) \neq 0$, since $d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b}$ is a scalar. As $\det(\mathbf{A}_k) \neq 0$ from the assumption, we conclude that $d - \mathbf{c}^\top \mathbf{A}_k^{-1} \mathbf{b} \neq 0$. By substituting $\mathbf{b} = \mathbf{L}_k \mathbf{y}$ 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 exactly matches 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.⁴ This completes

3. 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 $\det(\mathbf{M}) = \det(\mathbf{A}) \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$.

4. A triangular matrix (upper or lower) is nonsingular if and only if all the entries on its main diagonal are nonzero.

the proof. ■

We further show that the LU decomposition is unique when no permutation matrix is involved.

Corollary 1.8: (Uniqueness of LU decomposition without permutation) Let \mathbf{A} be an $n \times n$ square matrix with nonzero leading principal minors. Then, the LU decomposition of \mathbf{A} is unique.

Proof [of Corollary 1.8] Suppose, for contradiction, that the LU decomposition is not unique. Then, there exist two decompositions, $\mathbf{A} = \mathbf{L}_1 \mathbf{U}_1 = \mathbf{L}_2 \mathbf{U}_2$, which implies $\mathbf{L}_2^{-1} \mathbf{L}_1 = \mathbf{U}_2 \mathbf{U}_1^{-1}$. The left-hand side of the equation is a unit lower triangular matrix, while the right-hand side is an upper triangular matrix. Consequently, both sides must be 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 remains a unit lower triangular matrix, we deduce that $\mathbf{L}_2^{-1} \mathbf{L}_1 = \mathbf{I}$. The equality implies that both sides are identity matrices such that $\mathbf{L}_1 = \mathbf{L}_2$ and $\mathbf{U}_1 = \mathbf{U}_2$, resulting in a contradiction. This completes the proof. ■

In the proof of Theorem 1.4, we showed that the diagonal values of the upper triangular matrix are all nonzero if the leading principal minors of \mathbf{A} are all nonzero. Dividing each row of \mathbf{U} by its corresponding diagonal entry leads to an equivalent decomposition known as the *LDU decomposition*.

Corollary 1.9: (LDU decomposition) Let \mathbf{A} be an $n \times n$ square matrix with nonzero leading principal minors, 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 decomposed as

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

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 result follows directly from the LU decomposition of $\mathbf{A} = \mathbf{L}\mathbf{R}$, where \mathbf{L} is unit lower triangular and \mathbf{R} is upper triangular. Define $\mathbf{D} = \text{diag}(r_{11}, r_{22}, \dots, r_{nn})$, where r_{ii} 's are the diagonal entries of \mathbf{R} . Then, $\mathbf{D}^{-1}\mathbf{R} = \mathbf{U}$ is a unit upper triangular matrix. The uniqueness of the LDU decomposition follows from the uniqueness of the LU decomposition: since \mathbf{L} and \mathbf{R} are uniquely determined, so too are \mathbf{D} and \mathbf{U} .

1.4. Existence of LU Decomposition with Permutation

In Theorem 1.4, we require that \mathbf{A} has nonzero leading principal minors. However, this condition is not strictly necessary. Even if some leading principal minors are zero, a non-singular matrix can still admit an LU decomposition, provided that row permutations are allowed. The proof still proceeds by induction.

To formalize this, we first introduce the concept of the *Schur complement*.

Definition 1.10 (Schur complement). Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a matrix, and suppose its (1,1) entry, a_{11} , is nonzero. Then, the *Schur complement* of a_{11} in \mathbf{A} is $\mathbf{A}_2 = \mathbf{A}_{2:n, 2:n} - \frac{1}{a_{11}} \mathbf{A}_{2:n, 1} \mathbf{A}_{1, 2:n}$.

We now provide a rigorous proof for Theorem 1.1.

Proof [of Theorem 1.1: LU decomposition with permutation] We start with the base case: any 1×1 nonsingular matrix has a full LU decomposition $A = PLU$ by taking $P = 1$, $L = 1$, and $U = A$. Assume that every $(n - 1) \times (n - 1)$ nonsingular matrix admits an LU decomposition with permutation. We now show this holds for any $n \times n$ nonsingular matrix A .

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

At least one element in the first column of A must be nonzero; otherwise, A would be singular. Therefore, we can apply a row permutation to move a nonzero element to the $(1, 1)$ position. Let P_1 be a permutation matrix that moves a nonzero entry to the $(1, 1)$ position. Define $B = P_1 A$ such that $b_{11} \neq 0$. Since both A and P_1 are nonsingular, their product B is also nonsingular.

Schur complement of B is also nonsingular. The Schur complement of b_{11} in B is the $(n - 1) \times (n - 1)$ matrix given by $\hat{B} = B_{2:n,2:n} - \frac{1}{b_{11}} B_{2:n,1} B_{1,2:n}$. To prove that \hat{B} is nonsingular, assume there exists an $(n - 1)$ -vector x such that $\hat{B}x = \mathbf{0}$. Then, x and $y = -\frac{1}{b_{11}} B_{1,2:n} \cdot x$ satisfy

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

Since B is nonsingular, x and y must be zero. Therefore, $\hat{B}x = \mathbf{0}$ holds only if $x = \mathbf{0}$, which implies that the null space of \hat{B} has dimension 0. Hence, \hat{B} is nonsingular.

By the induction assumption, any $(n - 1) \times (n - 1)$ nonsingular matrix can be written in the full LU decomposition form:

$$\hat{B} = P_2 L_2 U_2.$$

We then factor A as

$$\begin{aligned} A &= P_1^\top \begin{bmatrix} b_{11} & B_{1,2:n} \\ B_{2:n,1} & B_{2:n,2:n} \end{bmatrix} = P_1^\top \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix} b_{11} & B_{1,2:n} \\ P_2^\top B_{2:n,1} & P_2^\top B_{2:n,2:n} \end{bmatrix} \\ &= P_1^\top \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix} b_{11} & B_{1,2:n} \\ P_2^\top B_{2:n,1} & L_2 U_2 + P_2^\top \frac{1}{b_{11}} B_{2:n,1} B_{1,2:n} \end{bmatrix} \\ &= P_1^\top \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{1}{b_{11}} P_2^\top B_{2:n,1} & L_2 \end{bmatrix} \begin{bmatrix} b_{11} & B_{1,2:n} \\ \mathbf{0} & U_2 \end{bmatrix}. \end{aligned}$$

Thus, the full LU decomposition of $A = PLU$ is given by:

$$P = P_1^\top \begin{bmatrix} 1 & 0 \\ 0 & P_2 \end{bmatrix}, \quad L = \begin{bmatrix} 1 & 0 \\ \frac{1}{b_{11}} P_2^\top B_{2:n,1} & L_2 \end{bmatrix}, \quad U = \begin{bmatrix} b_{11} & B_{1,2:n} \\ \mathbf{0} & U_2 \end{bmatrix}.$$

This completes the proof. ■

1.5. Bandwidth Preserving in LU Decomposition without Permutation

We will demonstrate that the bandwidth of a matrix remains unchanged after LU decomposition, provided no row permutations are used. The bandwidth of a matrix is defined as follows.

Definition 1.11 (Matrix bandwidth). For any matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, where the entry at position (i, j) is denoted as a_{ij} , the matrix \mathbf{A} has *upper bandwidth* q if $a_{ij} = 0$ whenever $j > i + q$, and *lower bandwidth* p if $a_{ij} = 0$ whenever $i > j + p$.

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

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

We now prove that the bandwidth of a matrix is preserved during LU decomposition when no row permutations are applied.

Lemma 1.12: (Bandwidth preserving) Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ have upper bandwidth q and lower bandwidth p . If \mathbf{A} admits the LU decomposition $\mathbf{A} = \mathbf{L}\mathbf{U}$, then \mathbf{L} has a lower bandwidth of p , and \mathbf{U} has an upper bandwidth of q .

Proof [of Lemma 1.12] The LU decomposition without permutation can be obtained as follows:

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

where $\mathbf{S} = \mathbf{A}_{2:n,2:n} - \frac{1}{a_{11}}\mathbf{A}_{2:n,1}\mathbf{A}_{1,2:n}$ is the Schur complement of a_{11} in \mathbf{A} . This process is referred to as the s -decomposition of \mathbf{A} . The first column of \mathbf{L}_1 and the first row of \mathbf{U}_1 retain the required bandwidth properties (bandwidth p and q , respectively). Furthermore, the Schur complement \mathbf{S} of a_{11} has upper bandwidth $q - 1$ and lower bandwidth $p - 1$, respectively. By applying induction on the s -decomposition to \mathbf{S} , the result follows. ■

1.6. Block LU Decomposition

Another form of LU decomposition involves factoring a matrix into block triangular matrices.

Theorem 1.13: (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 uniquely 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}_{ij} 's and \mathbf{U}_{ij} 's are certain block matrices.

It is important to note that that matrix \mathbf{U} in the above theorem is not necessarily upper triangular. For example:

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

The standard non-block LU decomposition fails for \mathbf{A} because the $(1, 1)$ entry is zero. However, the block LU decomposition still applies.

1.7. Partial, Complete, and Rook Pivoting

In the field of numerical linear algebra, solving systems of linear equations is a fundamental task that often relies on matrix factorization techniques such as LU decomposition. However, the direct application of LU decomposition can lead to numerical instability, especially when dealing with matrices that have small diagonal entries in their upper triangular form \mathbf{U} . To address this issue, *pivoting* strategies are employed to improve the robustness and accuracy of the decomposition process. This section explores the concept of pivoting, focusing specifically on *partial pivoting*, *complete pivoting*, and *rook pivoting*.

1.7.1 Partial Pivoting

In practice, it is often advantageous to apply pivoting even when it is not strictly necessary. When solving a linear system using LU decomposition, as described in Algorithm 3, selecting the pivot as the largest entry in the current column—especially when the diagonal elements of \mathbf{U} are small—helps mitigate inaccuracies in the solutions. This technique, known as *partial pivoting*, is widely adopted to enhance numerical stability. For example, in the partial pivoting process applied to a 4×4 matrix, the following transformation may occur:

$$\begin{matrix} \begin{bmatrix} \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{bmatrix} & \xrightarrow{\mathbf{E}_1} & \begin{bmatrix} \square & \square & \square & \square \\ 0 & \mathbf{2} & \square & \square \\ 0 & 5 & \square & \square \\ 0 & 7 & \square & \square \end{bmatrix} & \xrightarrow{\mathbf{P}_1} & \begin{bmatrix} \square & \square & \square & \square \\ 0 & 7 & \square & \square \\ 0 & 5 & \square & \square \\ 0 & 2 & \square & \square \end{bmatrix} & \xrightarrow{\mathbf{E}_2} & \begin{bmatrix} \square & \square & \square & \square \\ 0 & 7 & \square & \square \\ 0 & 0 & \square & \square \\ 0 & 0 & 0 & \square \end{bmatrix} \\ \mathbf{A} & & \mathbf{E}_1\mathbf{A} & & \mathbf{P}_1\mathbf{E}_1\mathbf{A} & & \mathbf{E}_2\mathbf{P}_1\mathbf{E}_1\mathbf{A} \end{matrix}. \quad (1.3)$$

In this example, after applying the transformation \mathbf{E}_1 , the element 7 is chosen as the pivot. Although this pivot is not strictly necessary, it ensures that no multiplier exceeds an absolute value of 1 during Gaussian elimination.

The general procedure for computing the LU decomposition with partial pivoting for a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is detailed in Algorithm 1. The computational cost of this algorithm is approximately $\sim 2/3(n^3)$ floating point operations (*flops*), along with $(n - 1) + (n - 2) + \dots + 1 \sim \mathcal{O}(n^2)$ comparisons due to the pivoting process (Lu, 2021c). The result of this process is an upper triangular matrix \mathbf{U} , given by

$$\mathbf{U} = \mathbf{E}_{n-1}\mathbf{P}_{n-1} \dots \mathbf{E}_2\mathbf{P}_2\mathbf{E}_1\mathbf{P}_1\mathbf{A}. \quad (1.4)$$

Computing the final \mathbf{L} . We still need to reconstruct the LU decomposition in the standard form:

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

where $\mathbf{P} = \mathbf{P}_1\mathbf{P}_2 \dots \mathbf{P}_{n-1}$ represents the overall permutation matrix, \mathbf{U} is the upper triangular matrix obtained directly from the algorithm, and \mathbf{L} is a unit lower triangular matrix with $|l_{ij}| \leq 1$ for all $1 \leq i, j \leq n$. The submatrix $\mathbf{L}_{k+1:n,k}$ is a permuted version of \mathbf{E}_k 's

Algorithm 1 LU Decomposition with Partial Pivoting

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

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

multipliers. To see this, we notice that the permutation matrices used in the algorithm involve only swaps of two rows. *This implies each \mathbf{P}_k is symmetric and satisfies $\mathbf{P}_k^2 = \mathbf{I}$, for $k \in \{1, 2, \dots, n - 1\}$.* Let

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

Substituting this into Equation (1.4), \mathbf{U} can be expressed as $\mathbf{U} = \mathbf{M}_{n-1} \dots \mathbf{M}_2 \mathbf{M}_1 \mathbf{P}^\top \mathbf{A}$. To analyze \mathbf{M}_k , recall that each \mathbf{P}_{k+1} is a permutation matrix with an identity submatrix in the upper-left $k \times k$ block. Thus,

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

This shows that \mathbf{M}_k is unit lower triangular, with its k -th column being a permuted version of \mathbf{E}_k . Consequently, the final lower triangular matrix is:

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

Thus, we obtain the full LU decomposition $\mathbf{A} = \mathbf{PLU}$.

Algorithm 2 LU Decomposition with Complete Pivoting

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

- 1: Set $\mathbf{U} = \mathbf{A}$;
 - 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 u_{kk} with the largest element in $|\mathbf{U}_{k:n,k:n}|$, say $u_{ab} = \max |\mathbf{U}_{k:n,k:n}|$;
 - 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 in column k of \mathbf{U} ;
 - 6: $\mathbf{U} = \mathbf{E}_k \mathbf{U}$;
 - 7: **end for**
 - 8: Output \mathbf{U} ;
-

1.7.2 Complete Pivoting

In partial pivoting, zeros below the diagonal in the k -th column of \mathbf{U} are introduced by selecting the pivot as the largest absolute value in the subcolumn $\mathbf{U}_{k:n,k}$. In contrast, *complete pivoting* identifies the pivot as the largest absolute entry in the current submatrix $\mathbf{U}_{k:n,k:n}$, which is then moved to the (k, k) position in \mathbf{U} . This requires applying an additional *column permutation* \mathbf{Q}_k at each step. The final upper triangular matrix \mathbf{U} is expressed as

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

The complete pivoting procedure is detailed in Algorithm 2.

The algorithm requires $2/3(n^3)$ flops, along with $(n^2 + (n-1)^2 + \dots + 1^2) \sim \mathcal{O}(n^3)$ comparisons due to the more extensive pivoting process. With $\mathbf{P} = \mathbf{P}_1 \mathbf{P}_2 \dots \mathbf{P}_{n-1}$, $\mathbf{Q} = \mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_{n-1}$,

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

and $\mathbf{L} = \mathbf{M}_1^{-1} \mathbf{M}_2^{-1} \dots \mathbf{M}_{n-1}^{-1}$, the final decomposition is $\mathbf{A} = \mathbf{PLUQ}^\top$, or equivalently, $\mathbf{P}^\top \mathbf{A} \mathbf{Q} = \mathbf{LU}$.

1.7.3 Rook Pivoting

Rook pivoting provides an alternative to partial and complete pivoting strategies. Instead of selecting the largest absolute value in $|\mathbf{U}_{k:n,k:n}|$ at the k -th step, it identifies an element that is *maximal in both its row and column* within that submatrix. This method is non-unique; multiple elements may satisfy the criteria. For instance, consider the following submatrix:

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

In this case, complete pivoting would select the element 7. In contrast, rook pivoting could select any of the entries 5, 4, or 7, since each of these values is the maximum in both its respective row and column.

1.8. Rank-Revealing LU Decomposition

In many applications, applying Gaussian elimination with pivoting to a matrix \mathbf{A} of rank r results in a factorization that reveals the rank structure 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} and \mathbf{Q} are permutation matrices. Such a factorization can be obtained using Gaussian elimination with either rook pivoting or complete pivoting; see Hwang et al. (1992); Higham (2002a) for more details.

1.9. Application: Linear System via LU Decomposition

For a well-determined linear system $\mathbf{Ax} = \mathbf{b}$, where \mathbf{A} is an $n \times n$ nonsingular matrix, directly computing \mathbf{A}^{-1} is computationally inefficient. Instead, the system can be solved

using LU decomposition. If \mathbf{A} admits an LU decomposition $\mathbf{A} = \mathbf{PLU}$, the solution can be obtained using the following algorithm:

Algorithm 3 Solving Linear Equations by LU Decomposition

Require: \mathbf{A} is a nonsingular $n \times n$ matrix; solve $\mathbf{Ax} = \mathbf{b}$;

- | | | |
|----|--|---------------------------------|
| 1: | LU decomposition: factor \mathbf{A} as $\mathbf{A} = \mathbf{PLU}$; | $\triangleright (2/3)n^3$ flops |
| 2: | Apply permutation: compute $\mathbf{w} = \mathbf{P}^\top \mathbf{b}$; | $\triangleright 0$ flops |
| 3: | Solve $\mathbf{Lv} = \mathbf{w}$ using forward substitution; | $\triangleright n^2$ flops |
| 4: | Solve $\mathbf{Ux} = \mathbf{v}$ using backward substitution; | $\triangleright n^2$ flops |
-

The LU decomposition requires a computational complexity of $(2/3)n^3$ flops (Lu, 2021c). Both the backward and forward substitution steps require n^2 flops in total, which can be derived from the sum $1 + 3 + \dots + (2n - 1) = n^2$ flops. Therefore, the overall computational cost is approximately $(2/3)n^3 + 2n^2$ flops. For large values of n , the dominant cost comes from the LU decomposition step, which scales as $(2/3)n^3$ flops. Additionally, in the case of a block LU decomposition, where $\mathbf{A} = \mathbf{LU}$, solving the systems $\mathbf{Lv} = \mathbf{w}$ and $\mathbf{Ux} = \mathbf{v}$ involves additional computational effort. This is because \mathbf{U} is generally not upper triangular, unlike in standard LU decomposition.

1.10. Application: Computing the Inverse of Nonsingular Matrices

By Theorem 1.1, any nonsingular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ admits a full LU factorization of the form $\mathbf{A} = \mathbf{PLU}$. The inverse of \mathbf{A} can be obtained by solving the matrix equation: $\mathbf{AX} = \mathbf{I}$, which involves solving n linear systems of the form $\mathbf{Ax}_i = \mathbf{e}_i$ for all $i \in \{1, 2, \dots, n\}$, where \mathbf{x}_i is the i -th column of \mathbf{X} and \mathbf{e}_i represents the i -th column of \mathbf{I} (i.e., the i -th standard basis vector).

Theorem 1.14: (Inverse of nonsingular matrix by linear system) Computing the inverse of a nonsingular matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ using n linear systems requires $\sim (2/3)n^3 + n(2n^2) = (8/3)n^3$ flops, where $(2/3)n^3$ corresponds to the cost of performing the LU decomposition of \mathbf{A} .

This result follows directly from Algorithm 3. However, computational efficiency can be improved by leveraging the triangular structure of \mathbf{U} and \mathbf{L} . Specifically, the inverse of \mathbf{A} can be expressed as $\mathbf{A}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}^{-1} = \mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{P}^\top$. Using this approach, the total computational cost can be reduced from $(8/3)n^3$ to $2n^3$ flops (Lu, 2021c).

1.11. Application: Computing the Determinant

The LU decomposition also simplifies the computation of the determinant of a matrix. If $\mathbf{A} = \mathbf{LU}$, then $\det(\mathbf{A}) = \det(\mathbf{LU}) = \det(\mathbf{L})\det(\mathbf{U}) = u_{11}u_{22}\dots u_{nn}$, where u_{ii} denotes the i -th diagonal element of \mathbf{U} (for $i \in \{1, 2, \dots, n\}$).⁵

Furthermore, for an LU decomposition with permutation, where $\mathbf{A} = \mathbf{PLU}$, the determinant of \mathbf{A} becomes $\det(\mathbf{A}) = \det(\mathbf{PLU}) = \det(\mathbf{P})u_{11}u_{22}\dots u_{nn}$. The determinant of a permutation matrix is either 1 or -1 because after changing rows around (which changes

5. The determinant of a lower triangular matrix (or an upper triangular matrix) is the product of its diagonal entries.

the sign of the determinant ⁶), a permutation matrix becomes the identity matrix \mathbf{I} , whose determinant is one.

Decorative swirls

Chapter 1 Problems

Decorative swirls

- Solve the following system of equations using row reduction:

$$\begin{aligned} 2x_1 + 3x_2 + 4x_3 &= 9, \\ x_1 + 2x_2 + 3x_3 &= 5, \\ 3x_1 + 4x_2 + 5x_3 &= 7. \end{aligned}$$

- Two matrices \mathbf{A} and \mathbf{B} are said to be *row equivalent* (denoted by $\mathbf{A} \xrightarrow{r} \mathbf{B}$) if \mathbf{A} can be transformed into \mathbf{B} using a sequence of elementary row operations.

- Show that $\mathbf{A} \xrightarrow{r} \mathbf{B}$ if and only if $\mathbf{A} = \mathbf{P}\mathbf{B}$ for some nonsingular matrix \mathbf{P} .
- Show that if $\mathbf{A} \xrightarrow{r} \mathbf{C}$ and $\mathbf{B} \xrightarrow{r} \mathbf{C}$, then $\mathbf{A} \xrightarrow{r} \mathbf{B}$.
- Show that if $\mathbf{A} \xrightarrow{r} \mathbf{B}$ and $\mathbf{B} \xrightarrow{r} \mathbf{C}$, then $\mathbf{A} \xrightarrow{r} \mathbf{C}$.
- Show that if $\mathbf{A} \xrightarrow{r} \mathbf{B}$, then $\mathbf{B} \xrightarrow{r} \mathbf{A}$.
- Show that $\mathbf{A} \xrightarrow{r} \mathbf{B}$ if both \mathbf{A} and \mathbf{B} are nonsingular.

- Let $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$ be $n \times n$ matrices that are *strictly upper triangular* (having zeros on the diagonal). Show that the product of $\mathbf{A}_1, \mathbf{A}_2, \dots, \mathbf{A}_n$ is the zero matrix.

- Given two matrices \mathbf{E} and \mathbf{F} obtained from the identity matrix by adding multiples of row i to rows j and k with $i \neq j$ and $i \neq k$, respectively, show that $\mathbf{EF} = \mathbf{FE}$.

- Show that the LU decomposition of the matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ does not exist.

- Suppose \mathbf{L}_1 and \mathbf{L}_2 are nonsingular lower triangular, and \mathbf{U}_1 and \mathbf{U}_2 are nonsingular upper triangular. Prove that $\mathbf{L}_1\mathbf{U}_1 = \mathbf{L}_2\mathbf{U}_2$ if and only if there exists a nonsingular diagonal matrix \mathbf{D} such that $\mathbf{L}_1 = \mathbf{L}_2\mathbf{D}$ and $\mathbf{U}_1 = \mathbf{D}^{-1}\mathbf{U}_2$.

- We know that elementary row operations on a matrix can be represented by left-multiplying the matrix with a corresponding transformation matrix (Definition 1.5). Describe the transformation matrices for:

- Interchanging two rows,
- Multiplying all elements of a row by a scalar,
- Adding a scalar multiple of one row to another row.

Extend this discussion to the three elementary column transformations.

- Consider the three types of elementary row transformation defined in Definition 1.5. Show the following effects on the determinant:

- Type-1 (row interchange): Multiplies the determinant by -1 .
- Type-2 (row scaling): Multiplies the determinant by the scaling factor.

- Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be a permutation matrix. Discuss how the matrix \mathbf{P} can be converted to the identity matrix using at most n elementary row transformations of a single type. Use this fact to express \mathbf{A} as the product of at most n elementary matrix operators.

- Suppose we reorder all the columns of an invertible matrix \mathbf{A} using a random permutation, and we already know \mathbf{A}^{-1} , the inverse of the original matrix. Show how we can compute the inverse of this reordered matrix directly from \mathbf{A}^{-1} without having to perform a full inversion from scratch. Use elementary matrices in the explanation.

⁶. The determinant changes sign when two rows are exchanged (sign reversal).

11. Prove or disprove each of the following statements by providing a counterexample if applicable:
- The sequence in which two elementary row transformations are applied to a matrix does not influence the final outcome.
 - The sequence in which one elementary row transformation and one elementary column transformation are applied to a matrix does not influence the final outcome.
12. **Complexity of vector inner product.** Given two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^n$, the inner product of the two vectors $\mathbf{v}^\top \mathbf{w}$ is calculated as $\mathbf{v}^\top \mathbf{w} = v_1 w_1 + v_2 w_2 + \dots + v_n w_n$. Show that the computational complexity of evaluating the inner product is $2n - 1$ floating-point operations (flops).
13. **Complexity of matrix multiplication.** Given two matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$ and $\mathbf{B} \in \mathbb{R}^{n \times k}$, show that the computational complexity of their product, \mathbf{AB} , is $mk(2n - 1)$ flops.
14. Discuss and provide algorithms used to compute the LU decomposition of a matrix. Use the results from Problems 1.12 and 1.13 to determine the computational complexity of the decomposition.
15. **Matrix inversion lemma.** Let \mathbf{A} be an invertible $n \times n$ matrix and let \mathbf{B}, \mathbf{C} be $n \times k$ nonzero matrices for some small value of k . Show that the matrix $\mathbf{A} + \mathbf{BC}^\top$ is invertible if and only if the $k \times k$ matrix $(\mathbf{I} + \mathbf{C}^\top \mathbf{A}^{-1} \mathbf{B})$ is invertible. Furthermore, show that the inverse is given by the following:

$$(\mathbf{A} + \mathbf{BC}^\top)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{I} + \mathbf{C}^\top \mathbf{A}^{-1} \mathbf{B})^{-1} \mathbf{C}^\top \mathbf{A}^{-1}.$$

This is also known as the *Sherman–Morrison–Woodbury identity*. Hint: Use Schur complements.

16. **Matrix inversion lemma.** Let $\mathbf{P} \in \mathbb{R}^{n \times n}$ be any matrix. Show that

$$(\mathbf{I} + \mathbf{P})^{-1} = \mathbf{I} - (\mathbf{I} + \mathbf{P})^{-1} \mathbf{P} = \mathbf{I} - \mathbf{P} (\mathbf{I} + \mathbf{P})^{-1}. \quad (1.5)$$

Hint: Premultiply and postmultiply the above identities with appropriate matrices.

17. **Push-through identity (Aggarwal, 2020).** Let \mathbf{A} and \mathbf{B} be two $m \times n$ matrices. Show the following result:

$$\mathbf{A}^\top (\mathbf{I}_m + \mathbf{BA}^\top)^{-1} = (\mathbf{I}_n + \mathbf{A}^\top \mathbf{B})^{-1} \mathbf{A}^\top. \quad (1.6)$$

Use the above result to show the following for any $m \times n$ matrix \mathbf{C} and scalar $\lambda > 0$:

$$\mathbf{C}^\top (\lambda \mathbf{I}_m + \mathbf{CC}^\top)^{-1} = (\lambda \mathbf{I}_n + \mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top. \quad (1.7)$$

The push-through identity derives its name from the fact that we push in a matrix on the left and it comes out on the right.

- Show that the inverse of a symmetric matrix is symmetric using LU decomposition.
- Consider the 3×3 row addition transformation $\mathbf{A} = \begin{bmatrix} 1 & c & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ with $c \neq 0$. When multiplied with another matrix on the right, it adds a multiple of one row to another row. Derive the inverse of \mathbf{A} by inverting a sum of matrices and using (1.5).

2

Cholesky Decomposition

2.1. Cholesky Decomposition

The property of positive definiteness or positive semidefiniteness is a significant characteristic of matrices. It not only provides insights into a matrix's fundamental nature but also underpins various mathematical and computational applications. In this chapter, we introduce decomposition methods for two special types of matrices, examining their unique properties and applications. We begin with the widely recognized Cholesky decomposition, a powerful method for revealing the positive definiteness of a matrix by factoring it into the product of a lower (or an upper) triangular matrix and its transpose. This decomposition facilitates numerical computations and is indispensable in optimization, statistical modeling, and other fields where ensuring positive definiteness is essential.

Theorem 2.1: (Cholesky decomposition) Every *positive definite* (PD) matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ can be decomposed 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 entries**. This factorization is known as the *Cholesky decomposition* of \mathbf{A} , and \mathbf{R} is referred to as the *Cholesky factor* or *Cholesky triangle* of \mathbf{A} .

Alternatively, \mathbf{A} can be expressed as $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$, where $\mathbf{L} = \mathbf{R}^\top$ is a lower triangular matrix **with positive diagonal elements**. Importantly, the Cholesky decomposition is **unique** (see Corollary 2.9).

The Cholesky decomposition derives its name from the French military officer and mathematician, *André-Louis Cholesky* (1875–1918), credited with its development during his surveying work. Similar to the LU decomposition, the Cholesky decomposition is primarily used to solve linear systems involving positive definite matrices. The approach for solving such systems parallels that of the LU decomposition, as discussed in Section 1.9, and will not be repeated here.

Applications: an overview. We will discuss additional applications of the Cholesky decomposition in Sections 2.12~2.14. Here, we provide a brief overview of its general use. Given a covariance matrix Σ , by applying the Cholesky decomposition $\Sigma = \mathbf{L}\mathbf{L}^\top$, we can transform independent standard normal random variables \mathbf{z} into multivariate normal random variables \mathbf{x} with covariance matrix Σ through the transformation $\mathbf{x} = \mathbf{L}\mathbf{z}$. Mathematically, this process can be described as follows:

- Let $\mathbf{z} = [z_1, z_2, \dots, z_n]^\top$ be a vector of independent standard normal random variables, i.e., $z_i \sim \mathcal{N}(0, 1)$ for all i .
- Let Σ denote the corresponding $n \times n$ positive definite covariance matrix.
- The Cholesky decomposition of Σ gives us $\Sigma = \mathbf{L}\mathbf{L}^\top$, where \mathbf{L} is a lower triangular matrix with positive diagonal entries.
- Then, the random vector $\mathbf{x} = \mathbf{L}\mathbf{z}$ follows a multivariate normal distribution with mean vector $\boldsymbol{\mu} = \mathbf{0}$ and covariance matrix Σ , since $\text{Cov}[\mathbf{x}] = \text{Cov}[\mathbf{L}\mathbf{z}] = \mathbf{L}\text{Cov}[\mathbf{z}]\mathbf{L}^\top = \Sigma$.

This transformation plays a crucial role in simulation processes across various domains. In finance, *Monte Carlo simulations* are widely used to model portfolios containing multiple assets. Since asset returns are often correlated, accurately capturing these dependencies is essential (Lu and Yi, 2022). Using the Cholesky decomposition, one can generate simulated paths of asset returns that reflect historical correlations embedded in the covariance matrix. Applications include *Value-at-Risk (VaR)* estimation, stress testing, and pricing multi-asset derivatives (Turkay et al., 2003). The method is also valuable in machine learning, particularly in sampling from *Gaussian processes*, which are used in Gaussian process regression and Bayesian optimization (Williams and Rasmussen, 2006; Lu, 2021d). Additionally, in probabilistic generative models such as *variational autoencoders (VAEs)* or diffusion models, sampling from a multivariate normal distribution is a key step in the generation process (Kingma et al., 2019; Lu, 2023). In *quantization* of large language or neural network models, the computational process of the Cholesky decomposition can be applied for efficient computation of quantization (Frantar et al., 2022).

In summary, the Cholesky decomposition offers a computationally efficient method for generating correlated random variables from uncorrelated ones. This makes it an essential tool in stochastic simulation and probabilistic modeling across diverse fields.

On the other hand, this decomposition has wide applications in optimization algorithms. For example, the goal of a *quadratic programming* problem is to minimize a quadratic function while satisfying a set of linear constraints. The standard form of a quadratic programming problem can be expressed as:

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^\top \mathbf{A} \mathbf{x} - \mathbf{b}^\top \mathbf{x},$$

where \mathbf{x} is the vector of decision variables, \mathbf{A} is a symmetric positive definite matrix, and \mathbf{b} is a constant vector. When the matrix \mathbf{A} is symmetric and positive definite, Cholesky decomposition can be used to simplify the solving process. The specific steps follow by replacing \mathbf{A} in the original objective function with its Cholesky decomposition $\mathbf{L}\mathbf{L}^\top$, resulting in the new objective function $\frac{1}{2}\mathbf{x}^\top(\mathbf{L}\mathbf{L}^\top)\mathbf{x} - \mathbf{b}^\top \mathbf{x}$. Let $\mathbf{y} = \mathbf{L}^\top \mathbf{x}$, then the optimization problem becomes:

$$\min_{\mathbf{y}} \frac{1}{2} \mathbf{y}^\top \mathbf{y} - (\mathbf{L}^{-\top} \mathbf{b})^\top \mathbf{y}.$$

This is a simple quadratic function in terms of \mathbf{y} , which is easy to solve. By solving the quadratic function in terms of \mathbf{y} , we obtain the optimal solution \mathbf{y}^* . This, in turn, yields the optimal solution \mathbf{x}^* of the original problem using backward substitution.

2.2. Existence of Cholesky Decomposition via Recursive Calculation

In this section, we demonstrate the existence of the Cholesky decomposition using recursive calculation. In Section 10.6.5, we will provide an alternative proof of its existence using QR decomposition and spectral decomposition. Before proving the existence of the Cholesky decomposition, we introduce the following definitions and lemmas.

Definition 2.2 (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 $\mathbf{x} \in \mathbb{R}^n$, denoted as $\mathbf{A} \succ \mathbf{0}$. 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 $\mathbf{x} \in \mathbb{R}^n$, denoted as $\mathbf{A} \succeq \mathbf{0}$.^a

- a. In discussions regarding positive definite or positive semidefinite matrices, it is essential to note that these matrices are symmetric. Therefore, the concept of a positive definite matrix holds significance only in the context of symmetric matrices.

One requirement for the existence of the Cholesky decomposition is the concept of positive definiteness. Several key properties of positive definite matrices are summarized below:

Positive Definite Matrix Property 1 of 5

A matrix \mathbf{A} is positive definite if and only if all of its eigenvalues are positive. Similarly, \mathbf{A} is positive semidefinite if and only if all of its eigenvalues are nonnegative. A detailed proof of this equivalence is presented in Section 10.6.3, based on the spectral theorem.

While not all components of a positive definite matrix need to be positive, it is guaranteed that the diagonal components of such a matrix are positive:

Positive Definite Matrix Property 2 of 5

Lemma 2.3: (Positive diagonals of positive definite matrices) The diagonal elements of a positive definite matrix \mathbf{A} are all *positive*. Likewise, the diagonal elements of a positive semidefinite matrix \mathbf{B} are all *nonnegative*.

Proof [of Lemma 2.3] By definition, for a positive definite matrix \mathbf{A} , we have $\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0$ for all nonzero vectors \mathbf{x} . In particular, let $\mathbf{x} = \mathbf{e}_i$, where \mathbf{e}_i is the i -th standard basis vector with 1 in the i -th position and 0 elsewhere. Then:

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

where a_{ii} represents the i -th diagonal component. A similar argument applies to PSD matrices, where $a_{ii} \geq 0$. This completes the proof. ■

Like the LU decomposition, the existence of the Cholesky decomposition also relies on properties of the Schur complement.

Positive Definite Matrix Property 3 of 5

Lemma 2.4: (Schur complement of positive definite matrices) For a positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the Schur complement of a_{11} is given by $\mathbf{S}_{n-1} = \mathbf{A}_{2:n, 2:n} - \frac{1}{a_{11}} \mathbf{A}_{2:n, 1} \mathbf{A}_{2:n, 1}^\top$. The Schur complement \mathbf{S}_{n-1} is also positive definite.

A note on notation. The subscript $n-1$ indicates that \mathbf{S}_{n-1} is an $(n-1) \times (n-1)$ matrix obtained from an $n \times n$ positive definite matrix. This notation will be used consistently in the following sections.

Proof [of Lemma 2.4] Let $\mathbf{v} \in \mathbb{R}^{n-1}$ be any nonzero vector. Construct a corresponding vector $\mathbf{x} \in \mathbb{R}^n$ as $\mathbf{x} = \begin{bmatrix} -\frac{1}{a_{11}} \mathbf{A}_{2:n,1}^\top \mathbf{v} \\ \mathbf{v} \end{bmatrix}$, which is nonzero. Now compute:

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

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, \mathbf{S}_{n-1} is positive definite as well. \blacksquare

This argument extends to PSD matrices as well: if \mathbf{A} is PSD, then its Schur complement \mathbf{S}_{n-1} is also PSD.

In the proof of Theorem 1.1, we showed that the Schur complement $\mathbf{S}_{n-1} = \mathbf{A}_{2:n,2:n} - \frac{1}{a_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{2:n,1}^\top$ is nonsingular if \mathbf{A} is nonsingular and $a_{11} \neq 0$. Similarly, the Schur complement of a_{nn} in \mathbf{A} is given by $\mathbf{S}'_{n-1} = \mathbf{A}_{1:n-1,1:n-1} - \frac{1}{a_{nn}} \mathbf{A}_{1:n-1,n} \mathbf{A}_{1:n-1,n}^\top$, which is also positive definite if \mathbf{A} is positive definite. This property is critical in proving that the leading principal minors of a PD matrix are all positive; further details can be found in Section 2.3.

Using these results, we now demonstrate the existence of the Cholesky decomposition via recursion.

Proof [of Theorem 2.1: existence of Cholesky decomposition recursively] For any positive definite matrix \mathbf{A} , note that $a_{11} > 0$ by Lemma 2.3. We can express \mathbf{A} as:

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

where $\mathbf{R}_1 = \begin{bmatrix} \sqrt{a_{11}} & \frac{1}{\sqrt{a_{11}}} \mathbf{A}_{2:n,1}^\top \\ \mathbf{0} & \mathbf{I} \end{bmatrix}$. By Lemma 2.4, \mathbf{S}_{n-1} is positive definite. Thus, we can factor it similarly: $\mathbf{S}_{n-1} = \widehat{\mathbf{R}}_2^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_{n-2} \end{bmatrix} \widehat{\mathbf{R}}_2$, where \mathbf{S}_{n-2} is also positive definite. Substituting this back, we obtain:

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

Repeating this process recursively, we eventually express \mathbf{A} as:

$$\mathbf{A} = \mathbf{R}_1^\top \mathbf{R}_2^\top \dots \mathbf{R}_n^\top \mathbf{R}_n \dots \mathbf{R}_2 \mathbf{R}_1 = \mathbf{R}^\top \mathbf{R},$$

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 above process can also be used to compute the Cholesky decomposition and analyze the computational complexity of the algorithm.

To go in the reverse direction, we can prove that the scatter matrix $\mathbf{R}^\top \mathbf{R}$ is positive definite under mild conditions.

Lemma 2.5: ($\mathbf{R}^\top \mathbf{R}$ is PD) Given any upper triangular matrix \mathbf{R} with positive diagonal elements, the matrix $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ is positive definite.

Proof [of Lemma 2.5] Since \mathbf{R} has positive diagonals, it has full column rank, and its null space is of dimension 0 by the fundamental theorem of linear algebra (Theorem 0.17). Consequently, $\mathbf{R}\mathbf{x} \neq \mathbf{0}$ for any nonzero vector \mathbf{x} . Therefore, $\mathbf{x}^\top \mathbf{A}\mathbf{x} = \|\mathbf{R}\mathbf{x}\|^2 > 0$ for any nonzero vector \mathbf{x} . \blacksquare

This lemma extends to any \mathbf{R} with linearly independent columns.

A word on the two claims. Combining Theorem 2.1 and Lemma 2.5, we conclude that a 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.

Algorithm 4 Cholesky Decomposition via Recursive Algorithm: Pseudo Code

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

- 1: **for** $k = 1$ to n **do** ▷ compute the k -th row of \mathbf{R}
 - 2: $r_{kk} \leftarrow \sqrt{a_{kk}}$; ▷ first element of k -th row
 - 3: $\mathbf{R}_{k,k+1:n} \leftarrow \frac{1}{r_{kk}} \mathbf{A}_{k,k+1:n}$; ▷ the rest elements of k -th row
 - 4: $\mathbf{A}_{k+1:n,k+1:n} \leftarrow \mathbf{A}_{k+1:n,k+1:n} - \mathbf{R}_{k,k+1:n}^\top \mathbf{R}_{k,k+1:n}$;
 - 5: **end for**
 - 6: Output $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$.
-

An alternative perspective of the recursive algorithm. The previous proof of the Cholesky decomposition can also be used to compute the decomposition itself; see Algorithm 4. Since $\mathbf{L} = \mathbf{R}^\top$ is lower triangular. The lower triangular factor \mathbf{L} can be computed as a product of a sequence of lower triangular matrices. To see this, we have

$$\mathbf{A} = \begin{bmatrix} a_{11} & \mathbf{A}_{1,2:n} \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} l_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} l_{11} & \mathbf{L}_{21}^\top \\ \mathbf{0} & \mathbf{L}_{22}^\top \end{bmatrix} = \mathbf{L}\mathbf{L}^\top.$$

Then we still have

$$\begin{bmatrix} a_{11} & \mathbf{A}_{1,2:n} \\ \mathbf{A}_{2:n,1} & \mathbf{A}_{2:n,2:n} \end{bmatrix} = \begin{bmatrix} l_{11}^2 & l_{11}\mathbf{L}_{21}^\top \\ l_{11}\mathbf{L}_{21} & \mathbf{L}_{21}\mathbf{L}_{21}^\top + \mathbf{L}_{22}\mathbf{L}_{22}^\top \end{bmatrix} \implies \begin{cases} l_{11} &= \sqrt{a_{11}}; \\ \mathbf{L}_{21} &= \frac{1}{l_{11}}\mathbf{A}_{2:n,1}; \\ \mathbf{L}_{22}\mathbf{L}_{22}^\top &= \mathbf{A}_{2:n,2:n} - \mathbf{L}_{21}\mathbf{L}_{21}^\top. \end{cases}$$

The second perspective involves constructing $n+1$ set of $n \times n$ matrices: $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(n+1)}$, where $\mathbf{A}^{(1)} = \mathbf{A}$, and we want to obtain $\mathbf{A}^{(n+1)} = \mathbf{I}$ via the relation:

$$\mathbf{A}^{(i)} = \mathbf{L}^{(i)} \mathbf{A}^{(i+1)} \mathbf{L}^{(i)\top}, \quad \forall i \in \{1, 2, \dots, n\}. \quad (2.1)$$

If these $\mathbf{L}^{(i)}$, $\forall i$ are lower triangular, then we obtain the Cholesky decomposition by

$$\mathbf{A} = (\mathbf{L}^{(1)} \mathbf{L}^{(2)} \dots \mathbf{L}^{(n)}) (\mathbf{L}^{(1)} \mathbf{L}^{(2)} \dots \mathbf{L}^{(n)})^\top = \mathbf{L}\mathbf{L}^\top.$$

This is indeed the case. To see this, we can construct

$$\mathbf{A}^{(i)} = \begin{bmatrix} \mathbf{I}_{i-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & a_{ii} & \mathbf{b}_i^\top \\ \mathbf{0} & \mathbf{b}_i & \mathbf{B}^{(i)} \end{bmatrix} \quad \text{and} \quad \mathbf{L}^{(i)} = \begin{bmatrix} \mathbf{I}_{i-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sqrt{a_{ii}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{a_{ii}}}\mathbf{b}_i & \mathbf{I}_{n-i} \end{bmatrix},$$

satisfying $\mathbf{A}^{(i)} = \mathbf{L}^{(i)} \mathbf{A}^{(i+1)} (\mathbf{L}^{(i)})^\top$:

$$\mathbf{A}^{(i+1)} = \begin{bmatrix} \mathbf{I}_{i-1} & 0 & \mathbf{0} \\ 0 & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^{(i)} - \frac{1}{a_{ii}} \mathbf{b}_i \mathbf{b}_i^\top \end{bmatrix} = \begin{bmatrix} \mathbf{I}_i & 0 & \mathbf{0} \\ 0 & a_{i+1,i+1} & \mathbf{b}_{i+1}^\top \\ \mathbf{0} & \mathbf{b}_{i+1} & \mathbf{B}^{(i+1)} \end{bmatrix}.$$

Therefore, \mathbf{A} can be decomposed as a set of lower triangular matrices in (2.1). Using the result in Exercise 2.6 can show that the algorithm for this perspective is equivalent to Algorithm 4.

Exercise 2.6. Verify that $\mathbf{L}_{i:,i}^{(i)} = \mathbf{L}_{i,i}$, $i = 1, 2, \dots, n$.

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

In Lemma 2.4, we proved that for any positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the Schur complement of a_{11} is given by $\mathbf{S}_{n-1} = \mathbf{A}_{2:n,2:n} - \frac{1}{a_{11}} \mathbf{A}_{2:n,1} \mathbf{A}_{2:n,1}^\top$, which is also positive definite. Similarly, the Schur complement of a_{nn} , $\mathbf{S}'_{n-1} = \mathbf{A}_{1:n-1,1:n-1} - \frac{1}{a_{nn}} \mathbf{A}_{1:n-1,n} \mathbf{A}_{1:n-1,n}^\top$, is also positive definite.

We now claim that all leading principal minors (Definition 1.3) of a positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ are positive, a result known as *Sylvester's criterion* (Swamy, 1973; Gilbert, 1991). Recall that these positive leading principal minors imply the existence of the LU decomposition for any positive definite matrix, as established in Theorem 1.4.

To prove Sylvester's criterion, we begin by establishing the following lemma:

Positive Definite Matrix Property 4 of 5

Lemma 2.7: (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 2.7] If \mathbf{A} is positive definite, then for any nonzero vector \mathbf{x} , $\mathbf{x}^\top \mathbf{E}^\top \mathbf{A} \mathbf{E} \mathbf{x} = \mathbf{y}^\top \mathbf{A} \mathbf{y} > 0$, since \mathbf{E} is invertible such that $\mathbf{E}\mathbf{x}$ is nonzero. ¹ Thus, $\mathbf{E}^\top \mathbf{A} \mathbf{E}$ is PD.

Conversely, if $\mathbf{E}^\top \mathbf{A} \mathbf{E}$ is positive definite, for any nonzero \mathbf{x} , $\mathbf{x}^\top \mathbf{E}^\top \mathbf{A} \mathbf{E} \mathbf{x} > 0$. For any nonzero \mathbf{y} , there exists a nonzero \mathbf{x} such that $\mathbf{y} = \mathbf{E}\mathbf{x}$, since \mathbf{E} is invertible. Hence, \mathbf{A} is also PD. ■

We now provide a rigorous proof of Sylvester's criterion.

Positive Definite Matrix Property 5 of 5

Theorem 2.8: (Sylvester's criterion) A real symmetric matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is positive definite if and only if all of its leading principal minors are positive.

Proof [of Theorem 2.8] We prove the forward implication by induction. Base case ($n = 1$): Since all the components on the diagonal of positive definite matrices are positive (Lemma 2.3), for a scalar matrix \mathbf{A} , $\det(\mathbf{A}) > 0$ if \mathbf{A} is positive definite.

Assume all leading principal minors of any $k \times k$ positive definite matrix are positive. For a $(k+1) \times (k+1)$ positive definite matrix \mathbf{M} , expressed in block form as $\mathbf{M} = \begin{bmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{b}^\top & d \end{bmatrix}$, where

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 $\mathbf{x} = \mathbf{0}$.

\mathbf{A} is a $k \times k$ positive definite submatrix. Its Schur complement of d , $\mathbf{S}_k = \mathbf{A} - \frac{1}{d}\mathbf{bb}^\top$, is also positive definite, and its determinant is positive by the inductive hypothesis. Therefore, $\det(\mathbf{M}) = \det(d) \det(\mathbf{A} - \frac{1}{d}\mathbf{bb}^\top) = \textcolor{blue}{2} d \cdot \det(\mathbf{A} - \frac{1}{d}\mathbf{bb}^\top) > 0$, establishing the result for $(k+1) \times (k+1)$ matrices.

Conversely, if all the leading principal minors of $\mathbf{A} \in \mathbb{R}^{n \times n}$ are positive, then all leading principal submatrices are nonsingular. Denote the (i, j) -th entry of \mathbf{A} as a_{ij} . By assumption, $a_{11} > 0$. To simplify \mathbf{A} , subtract appropriate multiples of its first row from the rows below to zero out the entries in the first column beneath the diagonal element a_{11} . This operation can be expressed as:

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

Next, subtract appropriate multiples of the first column of $\mathbf{E}_1 \mathbf{A}$ from the other columns to zero out the entries in the first row to the right of the diagonal element a_{11} . Due to the symmetry of \mathbf{A} , this operation can also be performed by multiplying $\mathbf{E}_1 \mathbf{A}$ on the right by \mathbf{E}_1^\top . The result is:

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

This operation preserves the principal minors of \mathbf{A} . Consequently, the leading principal minors of $\mathbf{E}_1 \mathbf{A} \mathbf{E}_1^\top$ are identical to those of \mathbf{A} .

By repeating this process iteratively, we transform \mathbf{A} into a diagonal matrix of the form $\mathbf{E}_n \dots \mathbf{E}_1 \mathbf{A} \mathbf{E}_1^\top \dots \mathbf{E}_n^\top$, where the diagonal entries match the diagonal entries of \mathbf{A} and are positive. Let $\mathbf{E} = \mathbf{E}_n \dots \mathbf{E}_2 \mathbf{E}_1$, which is an invertible matrix. Clearly, $\mathbf{E} \mathbf{A} \mathbf{E}^\top$ is PD, which implies that \mathbf{A} is also PD, as per Lemma 2.7. \blacksquare

2.4. Existence of Cholesky Decomposition via LU without Permutation

By Theorem 2.8 on Sylvester's criterion and Theorem 1.4 regarding the existence of an LU decomposition without permutation, a unique LU decomposition exists for a positive definite matrix \mathbf{A} of the form $\mathbf{A} = \mathbf{L}\mathbf{U}_0$, where \mathbf{L} is a unit lower triangular matrix and \mathbf{U}_0 is an upper triangular matrix. It is also established that *the signs of the pivots of a symmetric matrix are the same as the signs of the eigenvalues* (Strang, 2009):

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

The decomposition $\mathbf{A} = \mathbf{L}\mathbf{U}_0$ can be expressed as follows:

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

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 $\det(\mathbf{M}) = \det(\mathbf{D}) \det(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})$.

Here, the diagonal entries of \mathbf{U}_0 correspond to the pivots of \mathbf{A} . Moreover, as all eigenvalues of PD matrices are positive (by Lemma 10.30, a consequence of the spectral decomposition), it follows that the diagonal entries of \mathbf{U}_0 are also positive.

Let us now arrange the diagonal entries of \mathbf{U}_0 into a diagonal matrix \mathbf{D} such that $\mathbf{U}_0 = \mathbf{D}\mathbf{U}$. Substituting this into the decomposition yields:

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

This simplifies to $\mathbf{A} = \mathbf{LDU}$, where \mathbf{U} is a *unit* upper triangular matrix. By the uniqueness of the LU decomposition without permutation in Corollary 1.8 and the symmetry of \mathbf{A} , we conclude that $\mathbf{U} = \mathbf{L}^\top$, and hence $\mathbf{A} = \mathbf{LDL}^\top$. Since the diagonal entries of \mathbf{D} are positive, we can define $\mathbf{R} = \mathbf{D}^{1/2}\mathbf{L}^\top$, where $\mathbf{D}^{1/2} = \text{diag}(\sqrt{u_{11}}, \sqrt{u_{22}}, \dots, \sqrt{u_{nn}})$. Thus, we obtain $\mathbf{A} = \mathbf{R}^\top\mathbf{R}$, which represents the Cholesky decomposition of \mathbf{A} . The matrix \mathbf{R} is upper triangular with positive diagonal entries.

2.4.1 Diagonal Values of the Upper Triangular Matrix

Assume that \mathbf{A} is a positive definite matrix. We can express \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 $\mathbf{A}_k \in \mathbb{R}^{k \times k}$. The block LU decomposition of \mathbf{A} is given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_k & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} = \mathbf{L}\mathbf{U}_0 = \begin{bmatrix} \mathbf{L}_k & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{U}_k & \mathbf{U}_{12} \\ \mathbf{0} & \mathbf{U}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{L}_k\mathbf{U}_k & \mathbf{L}_k\mathbf{U}_{12} \\ \mathbf{L}_{21}\mathbf{U}_{k11} & \mathbf{L}_{21}\mathbf{U}_{12} + \mathbf{L}_{22}\mathbf{U}_{22} \end{bmatrix}.$$

The k -th order leading principal minor of \mathbf{A} is defined as $\Delta_k = \det(\mathbf{A}_{1:k, 1:k}) = \det(\mathbf{A}_k)$ (Definition 1.3). From the block LU decomposition, we have:

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

Since \mathbf{L}_k is a unit lower triangular matrix, its determinant is 1. Moreover, by the fact that *the determinant of a lower triangular matrix (or an upper triangular matrix) is equal to the product of the diagonal entries*, we obtain:

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

i.e., the k -th order leading principal minor of \mathbf{A} is equal to the determinant of the $k \times k$ leading submatrix of \mathbf{U}_0 , which is also the product of the first k diagonal entries of \mathbf{D} (from the decomposition $\mathbf{A} = \mathbf{LDL}^\top$). Let $\mathbf{D} = \text{diag}(d_1, d_2, \dots, d_n)$. Then:

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

The entries of \mathbf{D} can also be expressed in terms of the leading principal minors of \mathbf{A} as:

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

where Δ_k denotes the k -th order leading principal minor of \mathbf{A} , for all $k \in \{1, 2, \dots, n\}$. Consequently, the diagonal entries of \mathbf{R} (from the Cholesky decomposition $\mathbf{A} = \mathbf{R}^\top\mathbf{R}$) are:

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

2.4.2 Block Cholesky Decomposition

Building on the previous discussion, let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a PD matrix expressed in block form as $\mathbf{A} = \begin{bmatrix} \mathbf{A}_k & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$, where $\mathbf{A}_k \in \mathbb{R}^{k \times k}$. Its block LU decomposition is given by:

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

The k -th order leading principal submatrix \mathbf{A}_k of \mathbf{A} also admits its own LU decomposition: $\mathbf{A}_k = \mathbf{L}_k\mathbf{U}_k$. This implies that the Cholesky decomposition of an $n \times n$ matrix \mathbf{A} contains $n - 1$ smaller Cholesky decompositions for its leading principal submatrices: $\mathbf{A}_k = \mathbf{R}_k^\top \mathbf{R}_k$, for all $k \in \{1, 2, \dots, n-1\}$. This is particularly true because any leading principal submatrix \mathbf{A}_k of a positive definite matrix \mathbf{A} is also positive definite. To see this, for a PD matrix $\mathbf{A}_{k+1} \in \mathbb{R}^{(k+1) \times (k+1)}$, consider a vector $\mathbf{x}_k \in \mathbb{R}^k$ extended by a zero element, $\mathbf{x}_{k+1} = \begin{bmatrix} \mathbf{x}_k \\ 0 \end{bmatrix}$. Then,

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

Thus, \mathbf{A}_k is positive definite. By recursively applying this argument starting from $\mathbf{A} \in \mathbb{R}^{n \times n}$, we demonstrate that $\mathbf{A}_{n-1}, \mathbf{A}_{n-2}, \dots, \mathbf{A}_1$ are all positive definite. Consequently, each of these matrices admits a Cholesky decomposition.

2.5. Existence of Cholesky Decomposition via Induction

In the previous section, we demonstrated the existence of the Cholesky decomposition using the LU decomposition without permutation. Building on the proof of the LU decomposition presented in Section 1.3, we now demonstrate that the existence of the Cholesky decomposition can also be directly established using mathematical induction.

Proof [of Theorem 2.1: existence of Cholesky decomposition by induction] We will use induction to prove that every $n \times n$ positive definite matrix \mathbf{A} can be decomposed as $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$. The base case for 1×1 matrices is straightforward; setting $R = \sqrt{A}$ renders $A = R^2$.

Assume that any $k \times k$ positive definite matrix \mathbf{A}_k admits a Cholesky decomposition. We now show that any $(k+1) \times (k+1)$ PD matrix \mathbf{A}_{k+1} can also be factored as this Cholesky decomposition. Write \mathbf{A}_{k+1} as a block matrix $\mathbf{A}_{k+1} = \begin{bmatrix} \mathbf{A}_k & \mathbf{b} \\ \mathbf{b}^\top & d \end{bmatrix}$, where \mathbf{A}_k is a $k \times k$ PD matrix. By the inductive hypothesis, \mathbf{A}_k admits a Cholesky decomposition: $\mathbf{A}_k = \mathbf{R}_k^\top \mathbf{R}_k$. Construct the upper triangular matrix $\mathbf{R}_{k+1} = \begin{bmatrix} \mathbf{R}_k & \mathbf{r} \\ \mathbf{0} & s \end{bmatrix}$. Then,

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

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

$$\mathbf{b} = \mathbf{R}_k^\top \mathbf{r} \quad \text{and} \quad d = \mathbf{r}^\top \mathbf{r} + s^2.$$

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

$$\mathbf{r} = \mathbf{R}_k^{-\top} \mathbf{b} \quad \text{and} \quad s = \sqrt{d - \mathbf{r}^\top \mathbf{r}} = \sqrt{d - \mathbf{b}^\top \mathbf{A}_k^{-1} \mathbf{b}},$$

where we assume s is nonnegative. To ensure $s > 0$, note that 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 $\det(\mathbf{A}_k) > 0$, it follows that $(d - \mathbf{b}^\top \mathbf{A}_k^{-1}\mathbf{b}) > 0$. Thus, $s > 0$, and this completes the proof. \blacksquare

2.6. Uniqueness of Cholesky Decomposition

This uniqueness of the Cholesky decomposition follows directly from the uniqueness of the LU decomposition without permutation. Alternatively, a more detailed proof of this uniqueness is provided below.

Corollary 2.9: (Uniqueness of Cholesky decomposition) The Cholesky decomposition $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ of any positive definite matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is unique.

Proof [of Corollary 2.9] Suppose, for contraction, that the Cholesky decomposition is not unique. Then, there exist two distinct decompositions such that $\mathbf{A} = \mathbf{R}_1^\top \mathbf{R}_1 = \mathbf{R}_2^\top \mathbf{R}_2$. This 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-hand side of the previous equation is an upper triangular matrix, while the right-hand side is a lower triangular matrix. For both sides to be equal, they must both be diagonal matrices, 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 $\mathbf{\Lambda} = \mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{R}_2 \mathbf{R}_1^{-1}$ be the diagonal matrix. We notice that each diagonal value of $\mathbf{\Lambda}$ 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}). Suppose

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

We have

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

Given that both \mathbf{R}_1 and \mathbf{R}_2 have positive diagonals, it follows that $r_{11} = s_{11}, r_{22} = s_{22}, \dots, r_{nn} = s_{nn}$. And $\mathbf{\Lambda} = \mathbf{R}_1 \mathbf{R}_2^{-1} = \mathbf{R}_2 \mathbf{R}_1^{-1} = \mathbf{I}$. In other words, $\mathbf{R}_1 = \mathbf{R}_2$, which contradicts the assumption that the decomposition is not unique. \blacksquare

As a consequence of this proof, if we do not require the diagonal entries of \mathbf{R}_1 and \mathbf{R}_2 to be positive, then for each diagonal entry, we could have $r_{ii} = \pm s_{ii}$. In that case, the factorization $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ would not be unique.

³ Similarly, 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 Cholesky Decomposition

We presented a recursive algorithm for computing the Cholesky decomposition in Algorithm 4. It is also common to compute the Cholesky decomposition using element-level equations derived directly from the matrix equation $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$. Observe that the (i, j) -th entry of \mathbf{A} is $a_{ij} = \mathbf{R}_{:,i}^\top \mathbf{R}_{:,j} = \sum_{k=1}^i r_{ki} r_{kj}$ if $i < j$. This further implies, if $i < j$, we have

$$a_{ij} = \mathbf{R}_{:,i}^\top \mathbf{R}_{:,j} = \sum_{k=1}^i r_{ki} r_{kj} = \sum_{k=1}^{i-1} r_{ki} r_{kj} + r_{ii} r_{ij} \implies r_{ij} = (a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}) / r_{ii}, \quad \text{if } i < j.$$

On the other hand, if $i = j$, we have

$$a_{jj} = \sum_{k=1}^j r_{kj}^2 = \sum_{k=1}^{j-1} r_{kj}^2 + r_{jj}^2 \implies r_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} r_{kj}^2}. \quad (2.2)$$

If we equate the elements of \mathbf{R} by taking a column at a time and start with $r_{11} = \sqrt{a_{11}}$, we arrive at the element-level formulation of the Cholesky decomposition, as described in Algorithm 5.

Algorithm 5 Cholesky Decomposition Element-Wise: $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$

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

- ```

1: for $j = 1$ to n do ▷ Compute the j -th column of \mathbf{R}
2: for $i = 1$ to $j - 1$ do
3: $r_{ij} \leftarrow (a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}) / r_{ii}$, since $i < j$;
4: end for
5: $r_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} r_{kj}^2}$;
6: end for
7: Output $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$.

```
- 

On the other hand, Algorithm 5 can be adapted to compute the Cholesky decomposition in the form  $\mathbf{A} = \mathbf{L}\mathbf{D}\mathbf{L}^\top$ , where  $\mathbf{L}$  is unit lower triangular and  $\mathbf{D}$  is diagonal, as outlined in Algorithm 6, where Step 3 and Step 5 are derived from (since  $l_{ii} = 1$ ,  $\forall i \in \{1, 2, \dots, n\}$ ):

$$a_{jj} = \sum_{k=1}^{j-1} d_{kk} l_{jk}^2 + d_{jj}; \quad a_{ij} = d_{jj} l_{ij} + \sum_{k=1}^{j-1} d_{kk} l_{ik} l_{jk}, \quad \text{if } i > j.$$

**Exercise 2.10.** Derive the complexity of Algorithms 5 and 6.

This alternative form of the Cholesky decomposition is particularly useful for estimating the *condition number* of a PD matrix. In essence, the condition number of a function measures how sensitive its output is to small perturbations in the input; a smaller condition number indicates greater numerical stability. For positive definite linear systems, the condition number is defined as the ratio of the largest eigenvalue to the smallest eigenvalue of the PD matrix. The condition number of a positive definite matrix is lower bounded by the diagonal matrix obtained from its Cholesky decomposition:

$$\text{cond}(\mathbf{A}) \geq \text{cond}(\mathbf{D}). \quad (2.3)$$

This inequality can be proven by showing that  $\lambda_{\max} \geq d_{\max}$  and  $\lambda_{\min} \leq d_{\min}$ , where  $\lambda_{\max}$  and  $\lambda_{\min}$  denote the largest and smallest eigenvalues of  $\mathbf{A}$ , and  $d_{\max}$  and  $d_{\min}$  represent the largest and smallest diagonals of  $\mathbf{D}$ . Therefore, this form of the Cholesky decomposition can be used to improve the numerical behavior of *Newton's method*; see § 2.12.

---

**Algorithm 6** Cholesky Decomposition Element-Wise:  $\mathbf{A} = \mathbf{LDL}^{\top}$ 


---

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

```

1: for $j = 1$ to n do \triangleright Compute the j -th column of \mathbf{L}
2: $l_{jj} \leftarrow 1$;
3: $d_{jj} \leftarrow a_{jj} - \sum_{k=1}^{j-1} d_{kk} l_{jk}^2$;
4: for $i = j + 1$ to n do
5: $c_{ij} \leftarrow a_{ij} - \sum_{k=1}^{j-1} d_{kk} l_{ik} l_{jk}$, since $i > j$;
6: $l_{ij} \leftarrow \frac{c_{ij}}{d_{jj}}$;
7: end for
8: end for
9: Output $\mathbf{A} = \mathbf{LDL}^{\top}$, where $\mathbf{D} = \text{diag}(d_{11}, d_{22}, \dots, d_{nn})$.

```

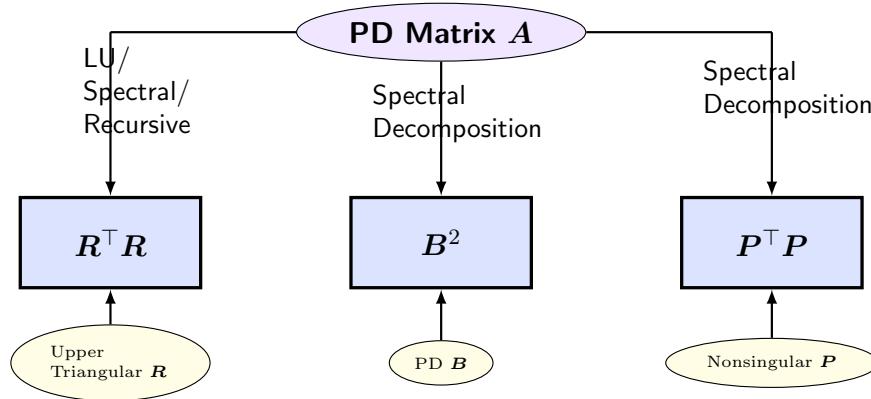
---

## 2.8. Final Remarks on Positive Definite Matrices

In Section 10.6.3, 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. Furthermore, in Section 10.6.6, we will demonstrate that a PD matrix  $\mathbf{A}$  admits a unique factorization  $\mathbf{A} = \mathbf{B}^2$ , where  $\mathbf{B}$  is also PD. Both results are derived from the spectral decomposition of positive definite matrices. To summarize, for a PD matrix  $\mathbf{A}$ , we can obtain the following factorizations:

- $\mathbf{A} = \mathbf{R}^{\top} \mathbf{R}$ , where  $\mathbf{R}$  is an upper triangular matrix with positive diagonals, as established in Theorem 2.1 via the Cholesky decomposition;
- $\mathbf{A} = \mathbf{P}^{\top} \mathbf{P}$ , where  $\mathbf{P}$  is nonsingular, as stated in Theorem 10.31;
- and  $\mathbf{A} = \mathbf{B}^2$ , where  $\mathbf{B}$  is PD, as given in Theorem 10.32.

For a comprehensive overview, these factorizations of a positive definite matrix  $\mathbf{A}$  are summarized in Figure 2.1.



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

## 2.9. Decomposition for Semidefinite Matrices

For positive semidefinite matrices, the Cholesky decomposition can still exist, though slight modifications are required.

**Theorem 2.11: (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. The diagonal elements of  $\mathbf{R}$  may be zero, and it is important to note that the factorization is generally **not unique**.

In such a decomposition, the diagonal elements of  $\mathbf{R}$  may not accurately reflect the rank of  $\mathbf{A}$  (Higham, 2009).

**Example 2.12 ((Higham, 2009)).** Consider the matrix

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

A semidefinite decomposition is given by

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

The matrix  $\mathbf{A}$  has a rank of 2, while  $\mathbf{R}$  has only one nonzero diagonal element.  $\square$

It is worth noting that all PD matrices are full-rank, a property that plays a key role in many earlier proofs. Sylvester's criterion (Theorem 2.8) confirms this by stating that all leading principal minors of PD matrices are positive. Alternatively, one can argue that if a PD matrix  $\mathbf{A}$  were rank-deficient, the null space of  $\mathbf{A}$  would have a positive dimension. This would imply the existence of a vector  $\mathbf{x}$  in the null space such that  $\mathbf{A}\mathbf{x} = \mathbf{0}$ , contradicting the definition of positive definiteness.

However, this property does not hold for PSD matrices, whose null space may have a dimension greater than zero. To address this limitation, we introduce a more general, rank-revealing decomposition for semidefinite matrices.

**Theorem 2.13: (Semidefinite rank-revealing decomposition)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a positive semidefinite matrix of rank  $r$ . Then, it 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  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ .

**Proof** [of Theorem 2.13] The proof is constructive and follows a similar approach to the second perspective used for constructing the Cholesky decomposition, as described in Section 2.2. The algorithm begins with  $\mathbf{A}^{(1)} = \mathbf{A}$  and generates a sequence of matrices defined as

$$\mathbf{A}^{(k)} = [a_{ij}^{(k)}] = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^{(k)} \end{bmatrix}, \quad k = 1, 2, \dots, \quad \text{with } \mathbf{B}^{(k)} \in \mathbb{R}^{(n-k+1) \times (n-k+1)}.$$

At the beginning of step  $k$ , we select the largest diagonal element of  $\mathbf{A}^{(k)}$ ,

$$s_q^{(k)} = \max_{k \leq i \leq n} a_{ii}^{(k)},$$

and interchange rows and columns  $q$  and  $k$  to bring this into pivot position; that is,  $s_q^{(k)}$  appears in the  $(k, k)$  position of  $\mathbf{P}^{(k)\top} \mathbf{A}^{(k)} \mathbf{P}^{(k)}$ , where the permutation matrix  $\mathbf{P}^{(k)}$  has the form

$$\mathbf{P}^{(k)} = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{P}}^{(k)} \end{bmatrix},$$

and  $\tilde{\mathbf{P}}^{(k)} \in \mathbb{R}^{(n-k+1) \times (n-k+1)}$  is a smaller permutation matrix. This pivot must be positive for  $k < r$ , because otherwise  $\mathbf{B}^{(k)} = \mathbf{0}$ , which implies that  $\text{rank}(\mathbf{A}) < r$ . Next, the elements in the permuted  $\mathbf{A}^{(k)}$  are transformed according to the Cholesky Algorithm 4:

$$\begin{aligned} r_{kk} &= \sqrt{a_{kk}^{(k)}}, \quad r_{kj} = a_{kj}^{(k)} / r_{kk}, \quad j = k+1 : n, \\ a_{ij}^{(k+1)} &= a_{ij}^{(k)} - r_{ki} r_{kj}, \quad i, j = k+1 : n. \end{aligned}$$

This process is equivalent to subtracting a symmetric rank-one matrix  $\mathbf{r}_j \mathbf{r}_j^\top$  from  $\mathbf{A}^{(k)}$ , where  $\mathbf{r}_j = \mathbf{e}_j^\top \mathbf{R}$  is the  $j$ -th row of  $\mathbf{R}$ . The algorithm stops when  $k = r+1$ . Then all the remaining diagonal elements are zero, which implies that  $\mathbf{A}^{(r+1)} = \begin{bmatrix} \mathbf{I}_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ .

**Construction algorithm.** Below contains more constructive analysis. Following the second perspective for computing the Cholesky decomposition in Section 2.2, we can construct

$$\mathbf{P}^{(k)\top} \mathbf{A}^{(k)} \mathbf{P}^{(k)} = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & a_{kk}^{(k)} & \mathbf{b}_k^\top \\ \mathbf{0} & \mathbf{b}_k & \mathbf{B}^{(k)} \end{bmatrix} \quad \text{and} \quad \mathbf{L}^{(k)} = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \sqrt{a_{kk}^{(k)}} & \mathbf{0} \\ \mathbf{0} & \frac{1}{\sqrt{a_{kk}^{(k)}}} \mathbf{b}_k & \mathbf{I}_{n-k} \end{bmatrix},$$

satisfying  $\mathbf{P}^{(k)\top} \mathbf{A}^{(k)} \mathbf{P}^{(k)} = \mathbf{L}^{(k)} \mathbf{A}^{(k+1)} (\mathbf{L}^{(k)})^\top$ :

$$\mathbf{A}^{(k+1)} = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^{(k)} - \frac{1}{a_{kk}^{(k)}} \mathbf{b}_k \mathbf{b}_k^\top \end{bmatrix} = \begin{bmatrix} \mathbf{I}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & a_{k+1,k+1}^{(k+1)} & \mathbf{b}_{k+1}^\top \\ \mathbf{0} & \mathbf{b}_{k+1} & \mathbf{B}^{(k+1)} \end{bmatrix}.$$

However, we notice that these permutation matrices  $\mathbf{P}^{(1)}, \mathbf{P}^{(2)}, \dots, \mathbf{P}^{(r)}$  are used to permute two columns; therefore, they are symmetric satisfying  $\mathbf{P}^{(k)} \cdot \mathbf{P}^{(k)\top} = \mathbf{I}$  for all  $k$ . Let  $\mathbf{P} = \mathbf{P}^{(1)} \mathbf{P}^{(2)} \dots \mathbf{P}^{(r)}$ . Since  $(\mathbf{A}^{(r+1)})^2 = \mathbf{A}^{(r+1)}$ ,  $\mathbf{A}^{(1)} = \mathbf{A}$  can be expressed as

$$\mathbf{P}^\top \mathbf{A}^{(1)} \mathbf{P} = \mathbf{L} \mathbf{L}^\top; \tag{2.4}$$

$$\mathbf{L} = \left\{ \mathbf{P}^{(r)} \mathbf{P}^{(r-1)} \dots \mathbf{P}^{(2)} \mathbf{P}^{(1)} \right\} \left\{ \mathbf{P}^{(1)} \mathbf{L}^{(1)} \right\} \left\{ \mathbf{P}^{(2)} \mathbf{L}^{(2)} \right\} \dots \left\{ \mathbf{P}^{(r)} \mathbf{L}^{(r)} \right\} \mathbf{A}^{(r+1)}. \tag{2.5}$$

To complete the proof, it suffice to show that  $\mathbf{L}$  is lower triangular with the rank-revealing property. On the other hand, each lower triangular  $\mathbf{L}^{(k)}$  can be written as

$$\mathbf{L}^{(k)} = \mathbf{I} - \mathbf{l}_k \mathbf{e}_k^\top \quad \text{with} \quad \mathbf{l}_k = [\mathbf{0}_{k-1}, l_k, l_{k+1}, \dots, l_n]^\top,$$

where  $\mathbf{e}_k$  is the  $k$ -th standard unit basis, and  $\mathbf{l}_k$  is a vector containing  $k-1$  zeros. Note that  $1 - l_k \equiv \sqrt{a_{kk}^{(k)}}$  in this notation. For  $k \in \{1, 2, \dots, r-1\}$ , define

$$\begin{aligned} \mathbf{M}_k &= \mathbf{P}^{(r)} \mathbf{P}^{(r-1)} \dots \mathbf{P}^{(k+1)} \mathbf{L}^{(k)} \mathbf{P}^{(k+1)} \dots \mathbf{P}^{(r-1)} \mathbf{P}^{(r)} \\ &= \mathbf{P}^{(r)} \mathbf{P}^{(r-1)} \dots \mathbf{P}^{(k+1)} (\mathbf{I} - \mathbf{l}_k \mathbf{e}_k^\top) \mathbf{P}^{(k+1)} \dots \mathbf{P}^{(r-1)} \mathbf{P}^{(r)} \\ &= \mathbf{I} - (\mathbf{P}^{(r)} \mathbf{P}^{(r-1)} \dots \mathbf{P}^{(k+1)} \mathbf{l}_k) (\mathbf{e}_k^\top \mathbf{P}^{(k+1)} \dots \mathbf{P}^{(r-1)} \mathbf{P}^{(r)}) \\ &= \mathbf{I} - (\mathbf{P}^{(r)} \mathbf{P}^{(r-1)} \dots \mathbf{P}^{(k+1)} \mathbf{l}_k) \mathbf{e}_k^\top, \end{aligned}$$

where the last equality follows since  $\mathbf{e}_k^\top \mathbf{P}^{(k+1)} \dots \mathbf{P}^{(r-1)} \mathbf{P}^{(r)} = \mathbf{e}_k^\top$ . This implies  $\mathbf{M}_k$  is lower triangular with its  $k$ -th column representing a permuted version of  $\mathbf{L}^{(k)}$ . Therefore, it holds that

$$\mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_{r-1} = \left\{ \mathbf{P}^{(r)} \mathbf{P}^{(r-1)} \dots \mathbf{P}^{(2)} \right\} \left\{ \mathbf{L}^{(1)} \mathbf{P}^{(2)} \right\} \left\{ \mathbf{L}^{(2)} \mathbf{P}^{(3)} \right\} \dots \left\{ \mathbf{L}^{(r-1)} \mathbf{P}^{(r)} \right\};$$

$$\mathbf{L} \equiv \mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_{r-1} \mathbf{L}^{(r)} \mathbf{A}^{(r+1)}.$$

From the above analysis,  $\mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_{r-1}$  is lower triangular, and  $\mathbf{L}^{(r)} \mathbf{A}^{(r+1)}$  has the form

$$\mathbf{L}^{(r)} \mathbf{A}^{(r+1)} = \begin{bmatrix} \mathbf{M}_{11} & \mathbf{0} \\ \mathbf{M}_{21} & \mathbf{0} \end{bmatrix}, \quad \text{with lower triangular } \mathbf{M}_{11}.$$

Therefore,  $\mathbf{L} = \mathbf{M}_1 \mathbf{M}_2 \dots \mathbf{M}_{r-1} \mathbf{L}^{(r)} \mathbf{A}^{(r+1)}$  has the desired form

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{0} \end{bmatrix}, \quad \text{with lower triangular } \mathbf{L}_{11}.$$

This completes the proof. ■

A more compact proof of this rank-revealing decomposition for semidefinite matrices will be presented in Section 10.6.4, relying on the spectral decomposition (Theorem 10.1) and the column-pivoted QR decomposition (Theorem 3.2). Whereas, the proof for the trivial semidefinite decomposition Theorem 2.11 can be derived directly from the spectral decomposition and the standard QR decomposition (Theorem 3.1).

When the matrix  $\mathbf{A}$  is symmetric and indefinite, we can employ a *symmetric indefinite decomposition* or *Bunch–Kaufman decomposition* (Bunch and Kaufman, 1977).

**Theorem 2.14: (Bunch–Kaufman decomposition)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a symmetric (indefinite) matrix. Then, it can be factored as

$$\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{L} \mathbf{B} \mathbf{L}^\top,$$

where  $\mathbf{P}$  is a permutation matrix,  $\mathbf{L}$  is a unit lower triangular matrix, and  $\mathbf{B}$  is a block-diagonal matrix with each diagonal block of  $\mathbf{B}$  being either a  $1 \times 1$  or a  $2 \times 2$  matrix.

This type of decomposition is sometimes referred to as an  *$\mathbf{L} \mathbf{B} \mathbf{L}^\top$  decomposition*. It is particularly useful in practical applications, such as solving linear systems and computing eigenvalues of matrices, especially in cases where a direct Cholesky decomposition cannot be applied (e.g., when the matrix is not positive definite) (Dumas and Pernet, 2018).

## 2.10. Application: Rank-One and Rank-Two Update/Downdate

Updating linear systems after low-rank modifications of the system matrix is a common procedure in fields such as machine learning, statistics, and more (Lu, 2021d). For example, when computing the least squares solution using Cholesky decomposition (see Section 11.6), we may want to add or remove one or more data points from the data matrix  $\mathbf{A}$  and the observed data vector  $\mathbf{y}$  (that is, to add or delete a row in both  $\mathbf{A}$  and  $\mathbf{y}$ ) in order to analyze the performance of the updated system. However, it is well known that such updates can become numerically unstable in the presence of round-off errors (Seeger, 2004). If the system matrix is positive definite, a more numerically stable approach involves using a representation based on the Cholesky decomposition. In this section, we will provide a proof of the rank-one update/downdate using Cholesky decomposition.

### 2.10.1 Rank-One Update

A rank-one update  $\mathbf{A}'$  of a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  by a vector  $\mathbf{v}$  is defined as follows:

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

If we have already calculated the Cholesky factor  $\mathbf{R}$  of  $\mathbf{A}$ , then the Cholesky factor  $\mathbf{R}'$  of  $\mathbf{A}'$  can be calculated efficiently. This avoids recomputing the decomposition from scratch, reducing the computational cost from  $\mathcal{O}(n^3)$  to  $\mathcal{O}(n^2)$ . Specifically,  $\mathbf{R}'$  is obtained via a *rank-one Cholesky update*, leveraging the fact that  $\mathbf{A}'$  differs from  $\mathbf{A}$  only by a symmetric rank-one matrix. To derive  $\mathbf{R}'$ , consider a set of orthogonal matrices  $\mathbf{Q}_n \mathbf{Q}_{n-1} \dots \mathbf{Q}_1$  such that:

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

The Cholesky factor  $\mathbf{R}'$  can be determined by analyzing the above transformation. Specifically, the left-hand side of the equation, when multiplied by its transpose, yields:

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

Similarly, the right-hand side, when multiplied by its transpose, results in:

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

which agrees with the left-hand side equation. *Givens rotations* are such orthogonal matrices that can transfer  $\mathbf{R}$  and  $\mathbf{v}$  into  $\mathbf{R}'$ .

**Definition 2.15 ( $n$ -th Order Givens rotation).** An  $n$ -th order *Givens rotation* is a matrix  $\mathbf{G}_{kl}$  of the following form:

$$\mathbf{G}_{kl} = \mathbf{I} + (c - 1)(\delta_k \delta_k^\top + \delta_l \delta_l^\top) + s(\delta_k \delta_l^\top - \delta_l \delta_k^\top), \quad (2.6)$$

where  $\delta_k \in \mathbb{R}^n$  is the  $k$ -th standard unit basis. The subscripts  $k$  and  $l$  indicate that the **rotation occurs in the plane defined by the  $k$ -th and  $l$ -th dimensions**. In other words, we have

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

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 angle  $\theta$ . Using this angle, the Given rotation in (2.6) can be more precisely denoted as

$$\mathbf{G}_{kl} = \mathbf{G}_{kl}(\theta). \quad (2.7)$$

Specifically, one can also define the  $n$ -th order Givens rotation, where  $(k, k)$ ,  $(k, l)$ ,  $(l, k)$ , and  $(l, l)$  entries are  $c$ ,  $-\mathbf{s}$ ,  $\mathbf{s}$ , and  $c$ , respectively. The ideas are the same.

Some fundamental significance of Givens rotations, crucial for proving the existence of the QR decomposition, will be discussed shortly in Section 3.12.

It can be easily verified that the  $n$ -th order Givens rotation is orthogonal, and its determinant is 1. For any vector  $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top \in \mathbb{R}^n$ , the effect of applying the Givens rotation matrix  $\mathbf{G}_{kl}$  to  $\mathbf{x}$  is given by:

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

In other words, a Givens rotation applied to  $\mathbf{x}$  rotates the components  $x_k$  and  $x_l$  of  $\mathbf{x}$  by an angle  $\theta$ , while leaving all other components unchanged.

Now suppose we have an  $(n+1)$ -th order Givens rotation indexed from 0 to  $n$ :

$$\mathbf{G}_k = \mathbf{I} + (c_k - 1)(\boldsymbol{\delta}_0 \boldsymbol{\delta}_0^\top + \boldsymbol{\delta}_k \boldsymbol{\delta}_k^\top) + s_k(\boldsymbol{\delta}_0 \boldsymbol{\delta}_k^\top - \boldsymbol{\delta}_k \boldsymbol{\delta}_0^\top),$$

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

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

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

Let the  $k$ -th element of  $\mathbf{v}$  be  $v_k$ , and the  $k$ -th diagonal of  $\mathbf{R}$  be  $r_{kk}$ . Since  $\sqrt{v_k^2 + r_{kk}^2} \neq 0$ , we can define  $c_k = \frac{r_{kk}}{\sqrt{v_k^2 + r_{kk}^2}}$ ,  $s_k = -\frac{v_k}{\sqrt{v_k^2 + r_{kk}^2}}$ . Then,

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

In other words, the Givens rotation  $\mathbf{G}_k$  will set the  $k$ -th element of  $\mathbf{v}$  to zero and assign a nonzero value to  $r_{kk}$ . This result is essential for performing a rank-one update. A sequence of Givens rotations  $\mathbf{G}_n \mathbf{G}_{n-1} \dots \mathbf{G}_1$  transforms the augmented matrix as follows:

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

Each rotation requires  $6n$  floating-point operations (flops), yielding a total computational cost of  $6n^2$  flops for  $n$  such rotations. This approach significantly reduces the complexity of calculating the Cholesky factor of  $\mathbf{A}'$  from  $\frac{1}{3}n^3$  to  $6n^2$  flops, assuming the Cholesky factor of  $\mathbf{A}$  is already known (Lu, 2021c). The algorithm is particularly useful in reducing the computational complexity of posterior calculations in Bayesian inference for *Gaussian mixture models* (Lu, 2021a). At each stage,  $k$  new samples are added or removed from an existing cluster, which corresponds to performing  $k$  rank-one updates.

## 2.10.2 Rank-One Downdate

Now suppose that the Cholesky factor of  $\mathbf{A}$  has been computed, and  $\mathbf{A}'$  is a *rank-one downdate* of  $\mathbf{A}$ , defined as:

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

The algorithm for performing such a downdate follows a similar procedure:

$$\mathbf{G}_1 \mathbf{G}_2 \dots \mathbf{G}_n \begin{bmatrix} \mathbf{0} \\ \mathbf{R} \end{bmatrix} = \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}' \end{bmatrix}. \quad (2.8)$$

Once again, each transformation,  $\mathbf{G}_k = \mathbf{I} + (c_k - 1)(\boldsymbol{\delta}_0 \boldsymbol{\delta}_0^\top + \boldsymbol{\delta}_k \boldsymbol{\delta}_k^\top) + s_k(\boldsymbol{\delta}_0 \boldsymbol{\delta}_k^\top - \boldsymbol{\delta}_k \boldsymbol{\delta}_0^\top)$ , can be constructed in the following way. Taking out the  $k$ -th column of the following equation

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

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

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

To ensure that  $\mathbf{A}'$  remains positive definite, it is necessary that  $r_{kk}^2 > v_k^2$ . If this condition is not satisfied, then  $c_k$ , as defined above, will not be real-valued, and the update cannot proceed. As a verification step, one can check that multiplying the left-hand side of (2.8) by its transpose yields:

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

Similarly, multiplying the right-hand side by its transpose gives:

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

This confirms that  $\mathbf{R}'^\top \mathbf{R}' = \mathbf{R}^\top \mathbf{R} - \mathbf{v} \mathbf{v}^\top$ .

## 2.11. Application: Indefinite Rank-Two Update

Let  $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$  be the Cholesky decomposition of  $\mathbf{A}$ . Goldfarb (1976); Seeger (2004) introduced a stable method for performing an 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

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

Now suppose that 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. Then, we can express  $\mathbf{A}'$  as

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

Finally, let  $\mathbf{R}' = \mathbf{R}^\top \mathbf{L}$ , which is a lower triangular matrix. This establishes the Cholesky decomposition of  $\mathbf{A}'$ .

---

<sup>4</sup>. This will be introduced in Theorem 3.11.

## 2.12. Application: Modified Newton's Method and Nearest Correlation

When optimizing or minimizing a function  $f(\mathbf{x})$  over  $\mathbf{x}$ , the standard Newton's method<sup>5</sup> updates the estimate at the  $t$ -th iteration as

$$\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \mathbf{d}^{(t)},$$

where  $(\nabla^2 f(\mathbf{x}^{(t)}))\mathbf{d}^{(t)} = -\nabla f(\mathbf{x}^{(t)})$  determines the “candidate” descent direction  $\mathbf{d}^{(t)}$ . The vector  $\mathbf{d}^{(t)}$  is a descent direction only when the Hessian  $(\nabla^2 f(\mathbf{x}^{(t)}))$  is PD, which is not always the case.

The modified Newton's method addresses this issue by approximating the Hessian with  $\mathbf{H}^{(t)} = \nabla^2 f(\mathbf{x}^{(t)}) + \mathbf{E}^{(t)}$ , ensuring that  $\mathbf{H}^{(t)}$  is PD (Gill et al., 2019; Lu, 2025). Given the Cholesky decomposition in the form  $\nabla^2 f(\mathbf{x}^{(t)}) = \mathbf{L}\mathbf{D}\mathbf{L}^\top = \mathbf{R}^\top \mathbf{R}$  (where  $\mathbf{R} = \mathbf{D}^{1/2} \mathbf{L}^\top$ ) and the condition number inequality  $\text{cond}(\nabla^2 f(\mathbf{x}^{(t)})) \geq \text{cond}(\mathbf{D})$  (see Equation (2.3)), the goal of the modified Newton's method can be approximately achieved by adjusting the diagonals of  $\mathbf{D}$ . To be more specific, when computing the Cholesky decomposition using Algorithm 6, the modified Newton's method imposes bounds on the diagonal  $d_{jj}$ , given two parameters  $\alpha$  and  $\beta$ , such that

$$d_{jj} \geq \alpha, \quad l_{ij} \sqrt{d_{jj}} \leq \beta, \quad i = \{j+1, j+2, \dots, n\}.$$

The latter constraint serves to upper-bound each row of  $\mathbf{R}$ , since  $\mathbf{R} = \mathbf{D}^{1/2} \mathbf{L}^\top$ . And this is equivalent to updating each  $d_{jj}$  in Algorithm 6 by

$$d_{jj} \leftarrow \max \left\{ |c_{jj}|, \beta, \max_{i>j} |c_{ij}| \right\}.$$

**Nearest correlation matrix problem.** The modified Cholesky decomposition discussed above can also be applied to the *nearest correlation matrix (NCM)* problem. In statistical modeling, a correlation matrix is often used to represent the correlation coefficients between a set of two or more random variables. The  $(i, j)$ -th entry of such a matrix represents the correlation coefficient between the variables  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . Clearly, such a matrix must be symmetric, have ones along the diagonal, and be positive semidefinite.

In many practical applications, however, a matrix that is intended to represent correlations between variables may fail to be a valid correlation matrix—most commonly because it is not positive semidefinite. There are several reasons this might occur, but it is typically due to missing data being estimated or matrix entries being altered, either intentionally or out of necessity. One specific example where this issue arises is in financial stress testing, which often involves modifying the elements of a matrix that represents the correlations among various stocks (Higham, 2002b; Higham and Strabic, 2016; McSweeney, 2017). In such cases, we often seek to find the nearest correlation matrix to the given one, which can then serve as the “true” matrix for further computations. This problem has long been of interest, especially in the finance industry.

## 2.13. Application: Obtain Orthonormal Basis

The Cholesky decomposition can be used to orthonormalize a basis set in an  $n$ -dimensional vector space. Let  $\mathbf{S} \in \mathbb{R}^{n \times n}$  be a full-rank matrix with Cholesky decomposition  $\mathbf{S}^\top \mathbf{S} =$

---

<sup>5</sup>. See, for example, Lu (2025).

$\mathbf{L}\mathbf{L}^\top$ , where  $\mathbf{L}$  is a lower triangular matrix. Consider the transformation  $\mathbf{Q} = \mathbf{S}(\mathbf{L}^{-1})^\top$ . We verify that  $\mathbf{Q}$  is orthogonal by computing:

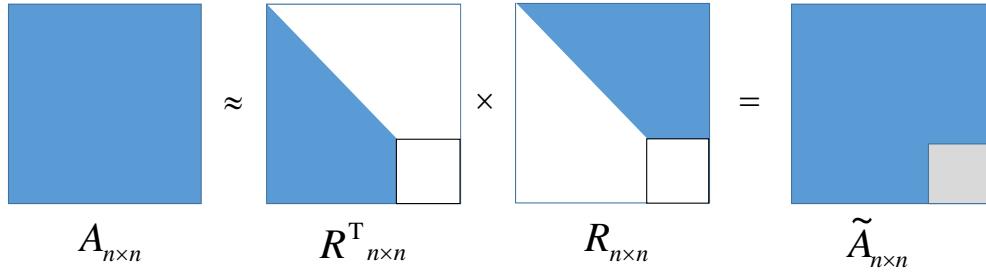
$$\mathbf{Q}^\top \mathbf{Q} = \mathbf{L}^{-1} \mathbf{S}^\top \mathbf{S} (\mathbf{L}^{-1})^\top = \mathbf{L}^{-1} \mathbf{L} \mathbf{L}^\top (\mathbf{L}^{-1})^\top = \mathbf{L}^{-1} \mathbf{L} (\mathbf{L}^{-1} \mathbf{L})^\top = \mathbf{I}. \quad (2.9)$$

To implement this transformation in a program, we can transpose both sides:  $\mathbf{Q}^\top = \mathbf{L}^{-1} \mathbf{S}^\top$ . Denoting the  $i$ -th row vectors of  $\mathbf{Q}$  and  $\mathbf{S}$  as  $\mathbf{q}_i$  and  $\mathbf{s}_i$ , respectively, we obtain:

$$\mathbf{q}_i = \mathbf{L}^{-1} \mathbf{s}_i \implies \mathbf{L} \mathbf{q}_i = \mathbf{s}_i, \quad i = 1, 2, \dots, n.$$

Since  $\mathbf{L}$  is lower triangular, each system can be efficiently solved using forward substitution. For simplicity, dropping the index  $i$ , consider the system  $\mathbf{L}\mathbf{q} = \mathbf{s}$ . The solution via forward substitution is given by the following recursion:

$$q_1 = \frac{s_1}{l_{11}}, \quad q_i = \frac{1}{l_{ii}} \left( s_i - \sum_{j=1}^{i-1} l_{ij} q_j \right), \quad i = 2, 3, \dots, n.$$



**Figure 2.2:** Demonstration of low-rank approximation using pivoted Cholesky decomposition. White entries represent zeros, while gray entries represent the approximated values.

## 2.14. Application: Low-Rank Approximation

We will discuss low-rank approximation in more detail in Section 11.8 and Chapter 12. The Cholesky decomposition of a positive definite matrix can also be used to construct a low-rank approximation of that matrix. Let  $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$  be the Cholesky decomposition of a PD matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . We then observe that the  $(i, j)$ -th entry of  $\mathbf{A}$  is given by

$$a_{ij} = \sum_{k=1}^{\min(i,j)} r_{ki} r_{kj}, \quad \forall i, j. \quad (2.10)$$

A low-rank approximation of  $\mathbf{A}$  can be obtained by truncating summation over  $k$  in (2.10) at some  $k \leq r \ll n$ . This is achieved by applying row and column permutations during each iteration of the Cholesky decomposition so that the largest diagonal element appears at the top of the currently considered submatrix (Golub and Van Loan, 2013). This approach is equivalent to performing a complete pivoting strategy (see Section 1.7.2), since in a PD matrix, the largest absolute entry always lies on the diagonal (see Problem 2.1).

This method is implemented in the pivoted Cholesky algorithm described in Algorithm 7, where the iterations are truncated once the largest remaining diagonal element falls below

a specified threshold  $\delta$ . At the termination of the algorithm,  $r$  represents the rank of the approximation, and the resulting rank- $r$  approximation of  $\mathbf{A}$  is given by

$$a_{ij} \approx \sum_{k=1}^{\min(i,j,r)} r_{ki} r_{kj}, \quad \forall i, j. \quad (2.11)$$

This process is illustrated in Figure 2.2.

---

**Algorithm 7** Low-Rank Approximation via Pivoted Cholesky decomposition

---

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

```

1: for $j = 1$ to n do \triangleright Compute the j -th column of \mathbf{R}
2: $v = \arg \max_{k \in \{j, \dots, n\}} a_{kk}$;
3: if $a_{vv} < \delta$ then
4: $r = j - 1$;
5: break
6: end if
7: $\mathbf{A}_{j,:} \leftrightarrow \mathbf{A}_{v,:}$; \triangleright Swap j -th and v -th rows
8: $\mathbf{A}_{:,j} \leftrightarrow \mathbf{A}_{:,v}$; \triangleright Swap j -th and v -th columns
9: for $i = 1$ to $j - 1$ do
10: $r_{ij} = (a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}) / r_{ii}$, since $i < j$;
11: end for
12: $r_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} r_{kj}^2}$;
13: end for
14: Output $\mathbf{A} = \mathbf{R}^\top \mathbf{R}$ and rank r .

```

---

## ❖ Chapter 2 Problems ❖

1. Show that the largest element in a positive definite matrix lies on the diagonal. And a similar argument applies to positive semidefinite matrices.
2. Suppose that  $\mathbf{A}_1$  and  $\mathbf{A}_2$  are  $n \times n$  positive semidefinite matrices of ranks  $k_1$  and  $k_2$ , respectively, where  $k_2 > k_1$ . Prove that  $\mathbf{A}_1 - \mathbf{A}_2$  cannot be positive semidefinite.
3. **PD.** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Show that the following two statements are equivalent:
  - $\mathbf{x}^* \mathbf{A} \mathbf{x} > 0$  for all nonzero  $\mathbf{x} \in \mathbb{C}^n$ .
  - $\mathbf{x}^\top \mathbf{A} \mathbf{x} > 0$  for all nonzero  $\mathbf{x} \in \mathbb{R}^n$ .
4. **Trace, det of PD/PSD/ND matrices.** Let  $\mathbf{A}$  be positive definite (resp., positive semidefinite), show that  $\text{tr}(\mathbf{A})$ ,  $\det(\mathbf{A})$ , and the principal minors of  $\mathbf{A}$  are all positive (resp., nonnegative). Moreover,  $\text{tr}(\mathbf{A}) = 0$  if and only if  $\mathbf{A} = \mathbf{0}$ . Let  $\mathbf{B} \in \mathbb{R}^{n \times n}$  be negative definite. Show that  $\text{tr}(\mathbf{B})$  is negative;  $\det(\mathbf{B})$  is negative for odd  $n$  and positive for even  $n$ .
5. Show that the following matrix is positive definite and compute its Cholesky decomposition:  $\mathbf{A} = \begin{bmatrix} 5 & -1 & 0 \\ -1 & 4 & 2 \\ 0 & 2 & 8 \end{bmatrix}$ .
6. Given two positive semidefinite matrices  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ , show that  $\mathbf{A} + \mathbf{B}$  is also positive semidefinite.
7. Given two symmetric matrices  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $\mathbf{B} \in \mathbb{R}^{m \times m}$ . Prove that the following two claims are equivalent:

- (a)  $\mathbf{A}$  and  $\mathbf{B}$  are positive semidefinite.
- (b)  $\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix}$  is positive semidefinite.
8. Let  $\mathbf{B} \in \mathbb{R}^{n \times k}$  and  $\mathbf{A} = \mathbf{B}\mathbf{B}^\top$ . Show that  $\mathbf{A}$  is positive semidefinite; and  $\mathbf{A}$  is positive definite if and only if  $\mathbf{B}$  has full row rank.
9. Show that if  $\mathbf{A}$  is positive semidefinite, then  $\mathbf{A}^{-1}$  is positive definite (if exists).
10. Prove that any positive definite matrix  $\mathbf{A}$  is nonsingular. Hint: Consider  $\mathbf{Ax} = \mathbf{0}$  and analyze  $\mathbf{x}^\top \mathbf{Ax} = 0$ .
11. Let  $\mathbf{A}$  be positive semidefinite. Show that  $\mathbf{x}^\top \mathbf{Ax} = 0$  if and only if  $\mathbf{Ax} = \mathbf{0}$ . Furthermore, prove that a positive semidefinite  $\mathbf{A}$  is positive definite if and only if it is nonsingular.
12. **Quadratic form.** Consider the quadratic form  $L(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Ax} - \mathbf{b}^\top \mathbf{x} + c$ , where  $\mathbf{A} \in \mathbb{R}^{d \times d}$ ,  $\mathbf{x} \in \mathbb{R}^d$ , and  $c \in \mathbb{R}$ . Suppose  $\mathbf{A}$  is positive semidefinite. Show that  $L(\mathbf{x})$  is bounded below over  $\mathbb{R}^d$  if and only if  $\mathbf{b}$  is in the column space of  $\mathbf{A}$ .
13. **Quadratic form.** Consider the quadratic form  $L(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Ax} - \mathbf{b}^\top \mathbf{x} + c$ . Show that  $L(\mathbf{x})$  is coercive if and only if  $\mathbf{A}$  is PD (A function  $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  is called coercive if  $\lim_{\|\mathbf{x}\| \rightarrow \infty} f(\mathbf{x}) = \infty$ ).
14. **Quadratic form.** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a general square matrix (not necessarily symmetric). Show that  $\mathbf{x}^\top \mathbf{Ax} = \mathbf{x}^\top [\frac{1}{2}(\mathbf{A} + \mathbf{A}^\top)]\mathbf{x}$ . The latter quadratic form is induced from a symmetric matrix.
15. **Symmetric form.** Define  $P(\mathbf{A}) = \frac{1}{2}(\mathbf{A} + \mathbf{A}^\top)$  for  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Show that
  - Null space.  $\mathcal{N}(\mathbf{A}) \subset \mathcal{N}(P(\mathbf{A}))$  and  $\mathcal{N}(\mathbf{A}^\top) \subset \mathcal{N}(P(\mathbf{A}))$  such that  $\text{rank}(P(\mathbf{A})) \leq \text{rank}(\mathbf{A})$ .
  - When  $\text{rank}(P(\mathbf{A})) = \text{rank}(\mathbf{A})$ , then  $\mathbf{A}$ ,  $\mathbf{A}^\top$ , and  $P(\mathbf{A})$  have the same null space.
- Hint: Consider the quadratic form  $\mathbf{x}^\top \mathbf{Ax}$  and  $\mathbf{x}^\top \mathbf{A}^\top \mathbf{x}$ , and use Problem 2.11.
16. Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ . Show that the matrix  $\mathbf{AB} - \mathbf{BA}$  can never be positive semidefinite unless it is the zero matrix. Hint: Use the fact that the trace of a symmetric matrix is equal to the sum of its eigenvalues.
17. **Givens rotation and rotary embedding.** Let  $\mathbf{G}(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$  be a Givens rotation matrix, and let  $\mathbf{v}(\theta) = \begin{bmatrix} \sin(\theta) \\ \cos(\theta) \end{bmatrix}$ . Show that  $\mathbf{G}(\theta)\mathbf{v}(\theta t) = \mathbf{v}(\theta(t + \delta))$ , where  $t$  can represent time or position.
18. **Givens rotation and rotary embedding.** Using the notation from Problem 2.17, plot the dot product  $\mathbf{v}(\theta t)^\top \mathbf{v}(\theta(t + \delta))$  as a function of  $\delta$ . What do you observe?
19. **Givens rotation and rotary embedding.** Using the notation from Problem 2.17, plot the dot product  $\mathbf{v}(\theta t)^\top \mathbf{W}\mathbf{v}(\theta(t + \delta))$  as a function of  $\delta$ , where  $\mathbf{W}$  is an appropriately chosen random matrix. What behavior do you observe?
20. What is the difference between a Givens rotation with entries  $(k, k), (k, l), (l, k)$ , and  $(l, l)$  set to  $c, -s, s$ , and  $c$ , versus one where those entries are set to  $c, s, -s$ , and  $c$ ?
21. Verify that a Givens rotation is an orthogonal matrix, and its determinant is 1.

## Part II

# Triangularization, Orthogonalization, and Gram–Schmidt Process



# 3

## QR Decomposition

### 3.1. QR Decomposition

In many applications, the column space of a matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$  is of particular interest. The sequence of subspaces spanned by the columns  $\mathbf{a}_1, \mathbf{a}_2, \dots$  of  $\mathbf{A}$  is given by

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

where  $\mathcal{C}(\dots)$  denotes the subspace spanned by the vectors enclosed in the brackets. The principle behind *QR decomposition* is to construct an orthonormal basis set  $\mathbf{q}_1, \mathbf{q}_2, \dots$  that spans the same sequence of subspaces:

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

Orthogonal basis sets have many useful properties, such as simplifying coordinate transformations, projections, and distance computations. The QR decomposition, stated below, summarizes the result. A detailed discussion of its existence follows in subsequent sections.

**Theorem 3.1: (QR decomposition)** Any  $m \times n$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  (whether its columns are linearly independent or not) with  $m \geq n$  can be decomposed as

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

where

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

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

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

If  $\mathbf{A}$  has full rank, i.e.,  $\mathbf{A}$  has linearly independent columns,  $\mathbf{R}$  also has linearly independent columns, and  $\mathbf{R}$  is nonsingular in the *reduced* case. This implies the diagonals of  $\mathbf{R}$  are nonzero. Under the additional condition that the diagonal entries of  $\mathbf{R}$  are positive,

the reduced QR decomposition is **unique**. However, the full QR decomposition is typically not unique because the rightmost  $(m - n)$  columns of  $\mathbf{Q}$  can be arranged in any order.

Note that geometrically, the diagonal element  $r_{ii}$  of the upper triangular matrix  $\mathbf{R}_0$  is the distance (w.r.t. the  $\ell_2$  norm) between  $\mathbf{a}_i$  (the  $i$ -th column of  $\mathbf{A}$ ) and  $\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{i-1}\}$ ,  $i = 2, 3, \dots, n$ ; see Section 3.2.

Once the decomposition  $\mathbf{A} = \mathbf{QR}$  is known (for a square invertible  $\mathbf{A}$ ), inverting  $\mathbf{A}$  is easy:

$$\mathbf{A}^{-1} = \mathbf{R}^{-1} \mathbf{Q}^\top.$$

Since  $\mathbf{R}$  is upper triangular, computing  $\mathbf{R}^{-1}$  by backward substitution is much simpler and more stable than inverting  $\mathbf{A}$  directly (see Problem 3.24). Numerical software exploits this fact (often under the hood) to compute inverses or pseudo-inverses via QR factorizations.

The method for computing the QR decomposition was formally introduced by Erhard Schmidt in 1907 (Schmidt, 1907). However, Schmidt himself observed that similar mathematical expressions had already appeared in the earlier work of Gram in 1883 (Gram, 1883). Despite this historical overlap, contemporary literature generally distinguishes between the two formulations. The procedure based on Schmidt's derivation is commonly referred to as the classical Gram–Schmidt process, whereas the version derived from Gram's original approach is known as the modified Gram–Schmidt process. For a more in-depth comparison and analysis, refer to Section 3.5.

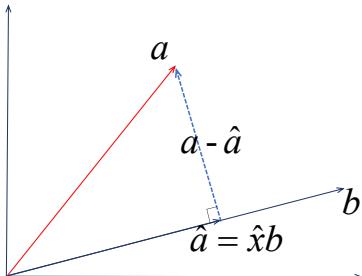
### 3.2. Project a Vector Onto Another Vector and Onto a Plane

An important concept in deriving the QR decomposition of a matrix is the projection of a vector onto another vector or onto a subspace.

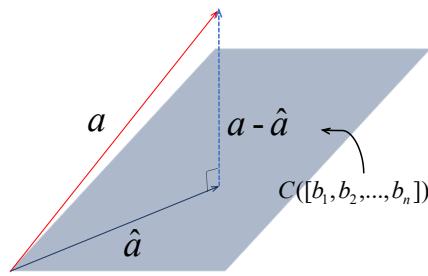
**Project a vector onto another vector.** Projecting a vector  $\mathbf{a}$  onto another vector  $\mathbf{b}$  involves finding the vector that is closest to  $\mathbf{a}$  along the line defined by  $\mathbf{b}$ . The projected vector, denoted as  $\hat{\mathbf{a}}$ , is a scalar multiple of  $\mathbf{b}$ :  $\hat{\mathbf{a}} = \hat{x}\mathbf{b}$ . By construction,  $\mathbf{a} - \hat{\mathbf{a}}$  is perpendicular to  $\mathbf{b}$ , as illustrated in Figure 3.1(a). This orthogonality condition leads to the following result:

Project vector  $\mathbf{a}$  onto vector  $\mathbf{b}$

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



(a) Project onto a line.



(b) Project onto a space.

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

**Project a vector onto a plane.** Similarly, the projection of a vector  $\mathbf{a}$  onto a space spanned by the vectors  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n$  involves finding the vector that is closest to  $\mathbf{a}$  within the column space of  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$ . The projected vector, denoted as  $\hat{\mathbf{a}}$ , is expressed as a linear 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 process defines a least squares problem, which is solved using 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  $\hat{\mathbf{x}} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n]$ . Each individual projection of  $\mathbf{a}$  onto a single vector  $\mathbf{b}_i$  can be computed as:  $\hat{\mathbf{a}}_i = \frac{\mathbf{b}_i \mathbf{b}_i^\top}{\mathbf{b}_i^\top \mathbf{b}_i} \mathbf{a}, \forall i \in \{1, 2, \dots, n\}$ . The total projection is then obtained by summing all of these individual projections:  $\hat{\mathbf{a}} = \sum_{i=1}^n \hat{\mathbf{a}}_i$ . Consequently, the residual vector is orthogonal to the entire column space of  $\mathbf{B}$ :  $\mathbf{a}^\perp = (\mathbf{a} - \hat{\mathbf{a}}) \perp \mathcal{C}(\mathbf{B})$ , as shown in Figure 3.1(b).

### 3.3. Existence of QR Decomposition via Gram–Schmidt Process

Given three linearly independent vectors  $\mathbf{a}_1, \mathbf{a}_2$ , and  $\mathbf{a}_3$  that span a space denoted by  $\mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ —which corresponds to the column space of the matrix  $[\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ —we aim to construct three orthogonal vectors  $\{\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$  such that  $\mathcal{C}([\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3])$ . By normalizing these orthogonal vectors (dividing each by its norm), we obtain three mutually orthonormal vectors:  $\mathbf{q}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|}$ ,  $\mathbf{q}_2 = \frac{\mathbf{b}_2}{\|\mathbf{b}_2\|}$ , and  $\mathbf{q}_3 = \frac{\mathbf{b}_3}{\|\mathbf{b}_3\|}$ .

To achieve this, we begin by setting  $\mathbf{b}_1 = \mathbf{a}_1$ . The second vector,  $\mathbf{b}_2$ , must be orthogonal to  $\mathbf{b}_1$ . It is obtained by subtracting the projection of  $\mathbf{a}_2$  onto  $\mathbf{b}_1$ :

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

The first equation shows that  $\mathbf{b}_2$  is computed by applying the matrix  $\left( \mathbf{I} - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1} \right)$  to  $\mathbf{a}_2$ , which projects  $\mathbf{a}_2$  onto the orthogonal complement of  $\mathcal{C}([\mathbf{b}_1])$ . The second equality expresses  $\mathbf{a}_2$  as a linear combination of its projection onto  $\mathbf{b}_1$  and a component orthogonal to  $\mathbf{b}_1$ :  $\mathbf{b}_2 \perp \mathbf{b}_1$ . This ensures that  $\mathcal{C}([\mathbf{b}_1, \mathbf{b}_2]) = \mathcal{C}([\mathbf{a}_1, \mathbf{a}_2])$ . Figure 3.2(a) illustrates the process, where **the direction of  $\mathbf{b}_1$  is aligned with the  $x$ -axis of a Cartesian coordinate system**.  $\hat{\mathbf{a}}_2$  is the projection of  $\mathbf{a}_2$  onto the line defined by  $\mathbf{b}_1$ . From the figure, it is clear that  $\mathbf{b}_2 = \mathbf{a}_2 - \hat{\mathbf{a}}_2$  is the component of  $\mathbf{a}_2$  orthogonal to  $\mathbf{b}_1$ .

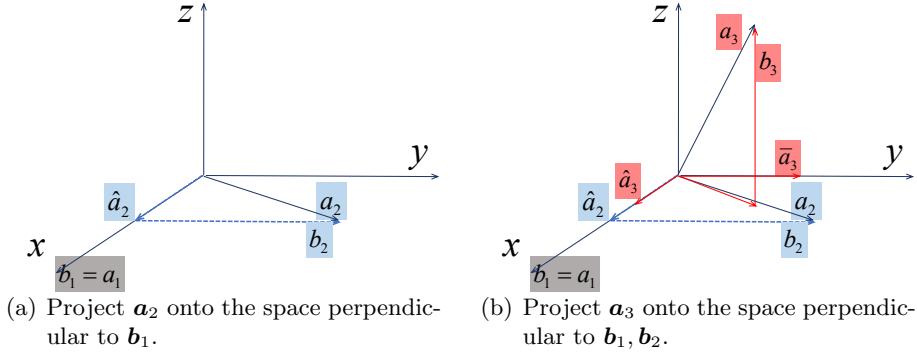
Similarly, the third vector,  $\mathbf{b}_3$ , must be orthogonal to both  $\mathbf{b}_1$  and  $\mathbf{b}_2$ . It is constructed by subtracting the projections of  $\mathbf{a}_3$  onto the subspaces spanned by  $\mathbf{b}_1$  and  $\mathbf{b}_2$ :

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

Once again, the first equation shows that the third vector  $\mathbf{b}_3$  is a multiplication of the matrix  $\left( \mathbf{I} - \frac{\mathbf{b}_1 \mathbf{b}_1^\top}{\mathbf{b}_1^\top \mathbf{b}_1} - \frac{\mathbf{b}_2 \mathbf{b}_2^\top}{\mathbf{b}_2^\top \mathbf{b}_2} \right)$  and the vector  $\mathbf{a}_3$ , i.e., projecting  $\mathbf{a}_3$  onto the orthogonal complement space of  $\mathcal{C}([\mathbf{b}_1, \mathbf{b}_2])$ . The second equality expresses  $\mathbf{a}_3$  as a linear combination of  $\mathbf{b}_1, \mathbf{b}_2$ , and  $\mathbf{b}_3$ . 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 identical to the space spanned by  $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ . Figure 3.2(b) illustrates this step, where **the direction of  $\mathbf{b}_2$  is aligned with the  $y$ -axis of the Cartesian coordinate system**. Here,  $\hat{\mathbf{a}}_3$  is the projection of  $\mathbf{a}_3$  onto  $\mathbf{b}_1$ , while  $\bar{\mathbf{a}}_3$  is the projection of  $\mathbf{a}_3$  onto  $\mathbf{b}_2$ . The figure also shows that the component of  $\mathbf{a}_3$  orthogonal to both  $\mathbf{b}_1$  and  $\mathbf{b}_2$  is  $\mathbf{b}_3 = \mathbf{a}_3 - \hat{\mathbf{a}}_3 - \bar{\mathbf{a}}_3$ .

Finally, each vector is normalized to produce the orthonormal set:  $\mathbf{q}_1 = \frac{\mathbf{b}_1}{\|\mathbf{b}_1\|}$ ,  $\mathbf{q}_2 = \frac{\mathbf{b}_2}{\|\mathbf{b}_2\|}$ , and  $\mathbf{q}_3 = \frac{\mathbf{b}_3}{\|\mathbf{b}_3\|}$ .



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

This process, called the *(classical) Gram–Schmidt process*, generalizes to any set of linearly independent vectors (Gram, 1883; Schmidt, 1907). The method effectively triangularizes the matrix  $\mathbf{A}$ , yielding its QR decomposition.

As mentioned previously, the goal of the QR decomposition is to construct a sequence of orthonormal vectors  $\mathbf{q}_1, \mathbf{q}_2, \dots$  that span the same successive subspaces as the original vectors:

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

This implies that any vector  $\mathbf{a}_k$  lies in the space spanned by  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k])$ .<sup>1</sup> Once the orthonormal vectors are determined, the original matrix  $\mathbf{A}$  can be reconstructed as  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ , where  $\mathbf{Q}$  is orthogonal and  $\mathbf{R}$  is upper triangular.

While the Gram–Schmidt process is a classical approach to obtain the QR decomposition of a matrix, it is not the only one. Other algorithms, such as *Householder reflections* and *Givens rotations*, are also widely used and often preferred due to their superior numerical stability in the presence of rounding errors. Additionally, these alternative methods may process the columns of  $\mathbf{A}$  in a different order; see Sections 3.11 and 3.12.

### 3.4. Orthogonal vs Orthonormal

The vectors  $\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n \in \mathbb{R}^m$  are said to be *mutually orthogonal* if their dot products satisfy  $\mathbf{q}_i^\top \mathbf{q}_j = 0$  whenever  $i \neq j$ . If each of these vectors is normalized to have unit length, they become *mutually orthonormal*. These orthonormal vectors can be arranged as columns in 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 a matrix  $\mathbf{Q}$  with  $m > n$  is sometimes referred to as a *semi-orthogonal* matrix.

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

- When  $m = n$ : the matrix  $\mathbf{Q}$  is square, and the condition  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}$  implies that  $\mathbf{Q}^\top = \mathbf{Q}^{-1}$ , meaning the transpose of  $\mathbf{Q}$  is its inverse. Then we also have  $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ , i.e.,  $\mathbf{Q}^\top$  is the *two-sided inverse* of  $\mathbf{Q}$ . In this case,  $\mathbf{Q}$  is called an *orthogonal matrix*.<sup>2</sup>

Orthogonal matrices can be interpreted as transformations that change the basis of a vector space while preserving both angles (inner products) and lengths. Specifically:

- The length of a vector is also preserved:  $\|\mathbf{Q}\mathbf{u}\| = \|\mathbf{u}\|$ .
- The angle (inner product) between two vectors remains invariant:  $\mathbf{u}^\top \mathbf{v} = (\mathbf{Q}\mathbf{u})^\top (\mathbf{Q}\mathbf{v})$ .

In real-valued cases, multiplying a vector by an orthogonal matrix  $\mathbf{Q}$  results in a *rotation* (if  $\det(\mathbf{Q}) = 1$ ) or a *reflection* (if  $\det(\mathbf{Q}) = -1$ ) in the vector space. Many decomposition algorithms produce two orthogonal matrices, leading to two such transformations (rotations or reflections); see Chapters 4 and 11.

### 3.5. Computing Reduced QR Decomposition via CGS and MGS

We express the reduced QR decomposition in the form  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ , where  $\mathbf{Q} \in \mathbb{R}^{m \times n}$  and  $\mathbf{R} \in \mathbb{R}^{n \times n}$ , as follows:

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

The orthogonal matrix  $\mathbf{Q}$ , with orthonormal columns, can be easily calculated using the Gram–Schmidt process. To understand why the matrix  $\mathbf{R}$  is upper triangular, we explicitly write the corresponding equations:

$$\begin{aligned} \mathbf{a}_1 &= r_{11}\mathbf{q}_1 &= \sum_{i=1}^1 r_{i1}\mathbf{q}_1, \\ &\vdots & \vdots \\ \mathbf{a}_k &= r_{1k}\mathbf{q}_1 + r_{2k}\mathbf{q}_2 + \dots + r_{kk}\mathbf{q}_k &= \sum_{i=1}^k r_{ik}\mathbf{q}_k, \\ &\vdots & \ddots \end{aligned}$$

This formulation aligns with the second equation in Equation (3.1) and confirms the upper triangular structure of  $\mathbf{R}$ . Extending the idea of Equation (3.1) to the  $k$ -th term, we obtain:

$$\mathbf{a}_k = \sum_{i=1}^{k-1} (\mathbf{q}_i^\top \mathbf{a}_k) \mathbf{q}_i + \mathbf{a}_k^\perp = \sum_{i=1}^{k-1} (\mathbf{q}_i^\top \mathbf{a}_k) \mathbf{q}_i + \|\mathbf{a}_k^\perp\| \cdot \mathbf{q}_k, \quad (3.2)$$

---

<sup>2</sup> Although  $\mathbf{Q}$  has orthonormal columns, the term “orthonormal matrix” is **not** used due to historical conventions.

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

$$\begin{cases} r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, \quad \forall i \in \{1, 2, \dots, k-1\}; \\ \mathbf{a}_k^\perp = \mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i; \\ r_{kk} = \|\mathbf{a}_k^\perp\|; \\ \mathbf{q}_k = \mathbf{a}_k^\perp / r_{kk}. \end{cases} \quad (3.3)$$

This again shows that the diagonal element  $r_{ii}$  of the upper triangular matrix is the distance (w.r.t. the  $\ell_2$  norm) between  $\mathbf{a}_i$  (the  $i$ -th column of  $\mathbf{A}$ ) and  $\text{span}\{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{i-1}\}$ ,  $i = 2, 3, \dots, n$ . The procedure is outlined in Algorithm 8.

---

**Algorithm 8** Reduced QR Decomposition via Gram–Schmidt Process

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with linearly independent columns, where  $m \geq n$ ;

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

---

**Orthogonal projection.** From Equation (3.3), particularly Steps 2 to 6 of Algorithm 8, we observe that the first two equalities imply that

$$\left. \begin{array}{l} r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, \quad \forall i \in \{1, 2, \dots, k-1\} \\ \mathbf{a}_k^\perp = \mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i \end{array} \right\} \rightarrow \mathbf{a}_k^\perp = \mathbf{a}_k - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top \mathbf{a}_k = (\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top) \mathbf{a}_k, \quad (3.4)$$

where  $\mathbf{Q}_{k-1} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]$ . This implies  $\mathbf{q}_k$  can be computed as:

$$\mathbf{q}_k = \frac{\mathbf{a}_k^\perp}{\|\mathbf{a}_k^\perp\|} = \frac{(\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top) \mathbf{a}_k}{\|(\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top) \mathbf{a}_k\|}.$$

The matrix  $(\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top)$  in the above expression is known as an *orthogonal projection matrix* (symmetric and idempotent; see Problem 3.1) that projects  $\mathbf{a}_k$  along the column space of  $\mathbf{Q}_{k-1}$ , ensuring the projected vector is orthogonal to the column space of  $\mathbf{Q}_{k-1}$  (Lu, 2021c). As a result, the vector  $\mathbf{a}_k^\perp$  or  $\mathbf{q}_k$  calculated in this manner will be orthogonal to  $\mathcal{C}(\mathbf{Q}_{k-1})$ , i.e., it lies in the null space of  $\mathbf{Q}_{k-1}^\top$ :  $\mathcal{N}(\mathbf{Q}_{k-1}^\top)$ , according to the fundamental theorem of linear algebra (Theorem 0.17).

Let  $\mathbf{P}_1 = (\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top)$ . We assert that  $\mathbf{P}_1 = (\mathbf{I} - \mathbf{Q}_{k-1} \mathbf{Q}_{k-1}^\top)$  is an orthogonal projection matrix, which projects any vector  $\mathbf{v}$  onto the null space of  $\mathbf{Q}_{k-1}^\top$ . Additionally, 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  $\mathbf{P}_2 \mathbf{v}$  projects any vector  $\mathbf{v}$  onto the column space of  $\mathbf{Q}_{k-1}$ .

Why can the matrices  $\mathbf{P}_1$  and  $\mathbf{P}_2$  effectively project vectors onto the corresponding subspaces? It can be shown that the column space of  $\mathbf{Q}_{k-1}$  is equal to the column space of  $\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top$ :  $\mathcal{C}(\mathbf{Q}_{k-1}) = \mathcal{C}(\mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top) = \mathcal{C}(\mathbf{P}_2)$ . Hence,  $\mathbf{P}_2\mathbf{v}$  represents a linear combination of the columns of  $\mathbf{P}_2$ , which lies in the column space of  $\mathbf{P}_2$  or the column space of  $\mathbf{Q}_{k-1}$ .

A *projection matrix*  $\mathbf{P}$  is formally defined as an idempotent matrix satisfying  $\mathbf{P}^2 = \mathbf{P}$ . This property reflects the intuitive idea that projecting a vector twice is the same as projecting it once. What distinguishes  $\mathbf{P}_2 = \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top$  is that the projection  $\hat{\mathbf{v}}$  of any vector  $\mathbf{v}$  is orthogonal to  $\mathbf{v} - \hat{\mathbf{v}}$ :

$$(\hat{\mathbf{v}} = \mathbf{P}_2\mathbf{v}) \perp (\mathbf{v} - \hat{\mathbf{v}}).$$

This property is the defining characteristic of an *orthogonal projection matrix*. In contrast, a projection that is not orthogonal is called an *oblique projection matrix*. When  $\mathbf{P}_2$  is an orthogonal projection matrix, the matrix  $\mathbf{P}_1 = \mathbf{I} - \mathbf{P}_2$  is also an orthogonal projection matrix, projecting any vector onto the space perpendicular to  $\mathcal{C}(\mathbf{Q}_{k-1})$ , i.e.,  $\mathcal{N}(\mathbf{Q}_{k-1}^\top)$  (see Proposition 10.29). Thus, we conclude that there are two complementary orthogonal projections:

$$\begin{cases} \mathbf{P}_1 : & \text{project onto } \mathcal{N}(\mathbf{Q}_{k-1}^\top), \text{ along the column space of } \mathbf{Q}_{k-1}; \\ \mathbf{P}_2 : & \text{project onto } \mathcal{C}(\mathbf{Q}_{k-1}), \text{ onto the column space of } \mathbf{Q}_{k-1}. \end{cases}$$

**Modified Gram–Schmidt process (MGS).** An additional noteworthy result arises when the columns of  $\mathbf{Q}_{k-1}$  are mutually orthonormal. In this case, we observe the following decomposition:

$$\mathbf{P}_1 = \mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top = (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^\top)(\mathbf{I} - \mathbf{q}_2\mathbf{q}_2^\top) \dots (\mathbf{I} - \mathbf{q}_{k-1}\mathbf{q}_{k-1}^\top), \quad (3.5)$$

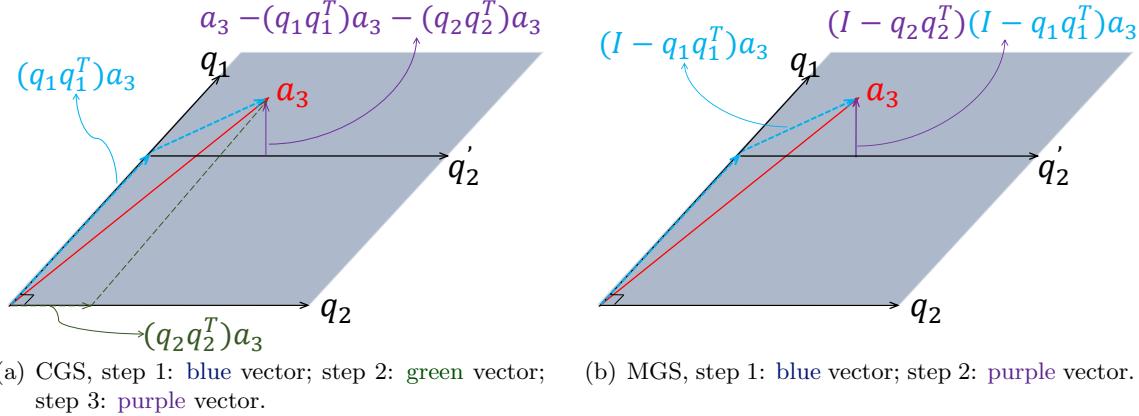
where  $\mathbf{Q}_{k-1} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]$ , and each term  $(\mathbf{I} - \mathbf{q}_i\mathbf{q}_i^\top)$  serves to project a vector onto the subspace orthogonal to  $\mathbf{q}_i$ . This finding is crucial for advancing towards a *modified Gram–Schmidt process (MGS)*, where projections and subtractions are performed iteratively. To avoid confusion, the original Gram–Schmidt method is often referred to as the *classical Gram–Schmidt process (CGS)*.

The primary distinction between CGS and MGS lies in how they perform projections and subtractions. In CGS, the same vector is projected onto all previously computed orthonormal vectors before performing the subtraction. Conversely, in MGS, projection and subtraction are interleaved. To illustrate this difference, consider a three-column matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ , as shown in Figure 3.3, where each step is represented using a different color. Below is a summary of the processes for computing  $\mathbf{q}_k$  from the  $k$ -th column  $\mathbf{a}_k$  of  $\mathbf{A}$ , given the orthonormalized vectors  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$ :

$$\begin{aligned} (\text{CGS}) : & \text{obtain } \mathbf{q}_k \text{ by normalizing } \mathbf{a}_k^\perp = (\mathbf{I} - \mathbf{Q}_{k-1}\mathbf{Q}_{k-1}^\top)\mathbf{a}_k; \\ (\text{MGS}) : & \text{obtain } \mathbf{q}_k \text{ by normalizing } \mathbf{a}_k^\perp = \left\{ (\mathbf{I} - \mathbf{q}_{k-1}\mathbf{q}_{k-1}^\top) \dots \left[ (\mathbf{I} - \mathbf{q}_2\mathbf{q}_2^\top) \left( (\mathbf{I} - \mathbf{q}_1\mathbf{q}_1^\top)\mathbf{a}_k \right) \right] \right\}, \end{aligned}$$

where the nested parentheses in MGS indicate the order of operations—each projection is applied sequentially, and the result is updated immediately.

**What's the difference?** To illustrate, consider the three-column matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3]$ . Suppose we have already computed  $\{\mathbf{q}_1, \mathbf{q}_2\}$ , where  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2\} = \text{span}\{\mathbf{a}_1, \mathbf{a}_2\}$ , our objective is to compute  $\mathbf{q}_3$ .



**Figure 3.3:** Comparison of CGS and MGS in three-dimensional space. Here,  $\mathbf{q}'_2$  is parallel to  $\mathbf{q}_2$ , so projecting onto  $\mathbf{q}_2$  is equivalent to projecting onto  $\mathbf{q}'_2$ .

In the CGS algorithm, the orthogonalization of column  $\mathbf{a}_3$  against  $\{\mathbf{q}_1, \mathbf{q}_2\}$  is accomplished by simultaneously projecting the original column  $\mathbf{a}_3$  of  $\mathbf{A}$  onto  $\mathbf{q}_1$  and  $\mathbf{q}_2$ , followed by subtracting these projections at once (Figure 3.3(a)):

$$\begin{cases} \mathbf{a}_3^\perp = \mathbf{a}_3 - (\mathbf{q}_1^\top \mathbf{a}_3) \mathbf{q}_1 - (\mathbf{q}_2^\top \mathbf{a}_3) \mathbf{q}_2 = \mathbf{a}_3 - (\mathbf{q}_1 \mathbf{q}_1^\top) \mathbf{a}_3 - \boxed{(\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3}; \\ \mathbf{q}_3 = \frac{\mathbf{a}_3^\perp}{\|\mathbf{a}_3^\perp\|}. \end{cases} \quad (3.6)$$

In contrast, the MGS algorithm subtracts the components along  $\{\mathbf{q}_1, \mathbf{q}_2\}$  from  $\mathbf{a}_3$  sequentially as they 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 the CGS, but rather by projecting onto updated intermediate vectors—those obtained after subtracting previously computed components. This distinction is crucial because it leads to smaller error components of  $\mathbf{q}_3$  in  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2\}$  (a point discussed further in the following paragraphs).

More precisely, in the MGS algorithm, the orthogonalization of column  $\mathbf{a}_3$  against  $\mathbf{q}_1$  is accomplished 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  $\mathbf{a}_3^{(1)}$  represents the component of  $\mathbf{a}_3$  that is orthogonal to  $\mathbf{q}_1$ . The subsequent step is then executed by

$$\mathbf{a}_3^{(2)} = (\mathbf{I} - \mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)} = \mathbf{a}_3^{(1)} - (\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)} = \mathbf{a}_3 - (\mathbf{q}_1 \mathbf{q}_1^\top) \mathbf{a}_3 - \boxed{(\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)}}, \quad (3.7)$$

where  $\mathbf{a}_3^{(2)}$  denotes the component of  $\mathbf{a}_3^{(1)}$  that is orthogonal to  $\mathbf{q}_2$ . The distinction from CGS (Equation (3.6)) is highlighted in blue text. Consequently,  $\mathbf{a}_3^{(2)}$  corresponds to the component of  $\mathbf{a}_3$  that is orthogonal to the entire subspace  $\{\mathbf{q}_1, \mathbf{q}_2\}$ , as shown in Figure 3.3(b).

**Main difference and catastrophic cancellation.** The key difference is that the vector  $\mathbf{a}_3$  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 that result in large relative errors in them—a phenomenon known as *catastrophic cancellation*. In contrast, in MGS, the intermediate

vector  $\mathbf{a}_3^{(1)}$  is already orthogonal to  $\mathbf{q}_1$  and has only a small “error” (residual) component in the direction of  $\mathbf{q}_1$ . This significantly reduces the chance of large cancellations occurring in subsequent steps. A comparison of the boxed terms in Equations (3.6) and (3.7) reveals that  $(\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3^{(1)}$  in Equation (3.7) is computed more accurately than  $(\mathbf{q}_2 \mathbf{q}_2^\top) \mathbf{a}_3$  in CGS, as argued above. Because of this reduced error in each projection step, the MGS method generally results in smaller orthogonalization errors at each stage compared to CGS. In fact, this difference can be quantified. It can be shown that the final orthogonal matrix  $\mathbf{Q}$  obtained using CGS satisfies the bound:

$$\|\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top\| \leq \mathcal{O}(\epsilon\kappa^2(\mathbf{A})),$$

where  $\kappa(\mathbf{A})$  is a value larger than 1 determined by  $\mathbf{A}$ . Whereas, in the MGS, the corresponding error satisfies

$$\|\mathbf{I} - \mathbf{Q}\mathbf{Q}^\top\| \leq \mathcal{O}(\epsilon\kappa(\mathbf{A})).$$

That is, the matrix  $\mathbf{Q}$  obtained via MGS is “more orthogonal” than that obtained via CGS, making MGS a more numerically stable algorithm in practice.

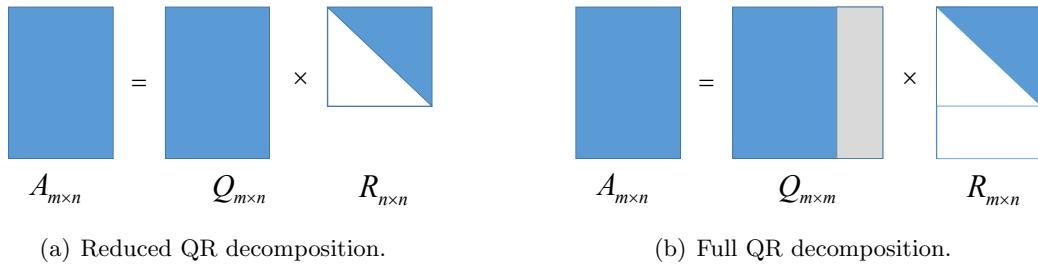
**More to go, preliminaries for Householder and Givens methods.** While MGS generally outperforms CGS in practice, it is not entirely immune to the *catastrophic cancellation* issue. For example, in iteration  $k$  of the MGS algorithm, if  $\mathbf{a}_k$  is nearly in the span of  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$ , then the resulting  $\mathbf{a}_k^\perp$  will have only a small component perpendicular to  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$ . This amplifies the “error” component in  $\text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}\}$ , leading to a less orthogonal  $\mathbf{Q}$ . In such scenarios, a more robust approach involves finding a sequence of orthogonal matrices  $\{\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_l\}$  such that the product  $\mathbf{Q}_l \dots \mathbf{Q}_2 \mathbf{Q}_1 \mathbf{A}$  becomes triangular. In this case, the resulting orthogonal matrix  $\mathbf{Q} = (\mathbf{Q}_l \dots \mathbf{Q}_2 \mathbf{Q}_1)^\top$  will be “more” orthogonal than those produced by either CGS or MGS. These more stable techniques will be explored in Section 3.11 and Section 3.12 using Householder reflectors and Givens rotations.

### 3.6. Computing Full QR Decomposition via Gram–Schmidt Process

A full QR decomposition of an  $m \times n$  matrix with linearly independent columns involves extending the decomposition by appending additional  $m - n$  orthonormal columns to  $\mathbf{Q}$ , transforming it into an  $m \times m$  orthogonal matrix. Simultaneously, rows of zeros are added to  $\mathbf{R}$ , making it an  $m \times n$  upper triangular matrix. The additional columns in  $\mathbf{Q}$  are referred to as *silent columns*, while the additional rows in  $\mathbf{R}$  are called *silent rows*. These do not affect the original decomposition but complete  $\mathbf{Q}$  to be a full orthogonal matrix. Figure 3.4 illustrates the differences between the reduced and full QR decompositions, where silent columns in  $\mathbf{Q}$  are denoted in gray, blank entries are zero, and blue entries indicate elements that are not necessarily zero.

### 3.7. Dependent Columns

Previously, we assumed that the matrix  $\mathbf{A}$  has linearly independent columns. However, this assumption is not always necessary. Suppose that in step  $k$  of the CGS or MGS process, the vector  $\mathbf{a}_k$  lies 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., the vectors  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$  are dependent. When this dependency occurs, the entry  $r_{kk}$  becomes zero (see Equation (3.3) or Algorithm 8), rendering  $\mathbf{q}_k$  undefined due to division by zero. To handle such scenarios, one can arbitrarily



**Figure 3.4:** Comparison between the reduced and full QR decompositions.

choose  $\mathbf{q}_k$  as any normalized vector orthogonal to the column space  $\mathcal{C}([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}])$  and proceed with the Gram–Schmidt process. For a matrix  $\mathbf{A}$  with dependent columns, both reduced and full QR decomposition algorithms are still applicable. The procedure for step  $k$  in the algorithm is redefined as follows:

$$\mathbf{q}_k = \begin{cases} (\mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i) / r_{kk}, & r_{ik} = \mathbf{q}_i^\top \mathbf{a}_k, r_{kk} = \left\| \mathbf{a}_k - \sum_{i=1}^{k-1} r_{ik} \mathbf{q}_i \right\|, \text{ if } r_{kk} \neq 0, \\ \text{pick one vector in } \mathcal{C}^\perp([\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{k-1}]), \text{ and normalize,} & \text{if } r_{kk} = 0. \end{cases}$$

This idea can be further extended: when  $q_k$  does not exist, we simply skip the current step and add the silent columns at the end of the process. Consequently, the QR decomposition of a matrix with dependent columns is generally not unique.

This framework also provides a practical method for determining linear independence. If  $r_{kk} = 0$  at any step in CGS or MGS, the vectors  $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_k$  are reported as linearly dependent. At this point, the algorithm can be terminated for the purpose of detecting linear dependence.

### 3.8. QR with Column Pivoting: Column-Pivoted QR (CPQR)

If the columns of  $\mathbf{A}$  are linearly dependent, a *column-pivoted QR* (CPQR) decomposition can be obtained as follows:

**Theorem 3.2: (Column-pivoted QR decomposition)** Any  $m \times n$  matrix  $A = [a_1, a_2, \dots, a_n]$  with  $m \geq n$  and rank  $r$  can be decomposed as

$$AP = Q \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix},$$

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 known as the *full* CPQR decomposition. Similarly, the *reduced* version is given by

$$AP = Q_r \begin{bmatrix} R_{11} & R_{12} \end{bmatrix},$$

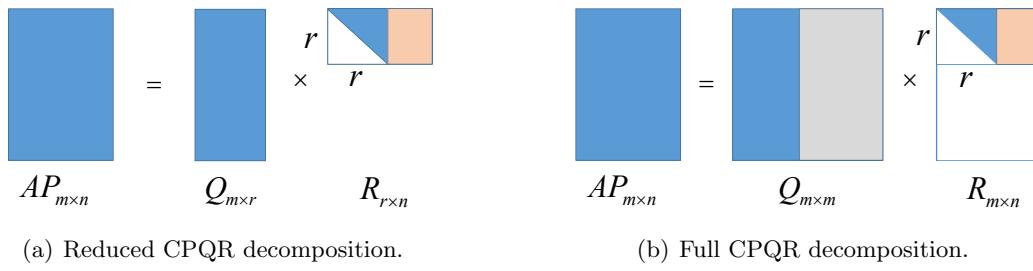
where  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is upper triangular,  $\mathbf{R}_{12} \in \mathbb{R}^{r \times (n-r)}$ ,  $\mathbf{Q}_r \in \mathbb{R}^{m \times r}$  contains orthonormal columns, and  $\mathbf{P}$  is a permutation matrix.

### 3.8.1 A Simple CPQR via CGS

The CPQR decomposition can be computed using the classical Gram–Schmidt process. In the context of QR decomposition for matrices with linearly dependent columns, if  $r_{kk} = 0$ , this indicates that column  $k$  of  $\mathbf{A}$  is linearly dependent on the previous  $k - 1$  columns. In such cases, a column permutation is performed, moving the dependent column to the end, after which the Gram–Schmidt process continues. Here,  $\mathbf{P}$  represents the permutation matrix that reorders the dependent columns into the last  $n - r$  positions. Suppose the first  $r$  columns of  $\mathbf{AP}$  are  $[\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \dots, \hat{\mathbf{a}}_r]$ . The span of these columns is equivalent to the span of  $\mathbf{Q}_r$  (in the reduced version) or the span of  $\mathbf{Q}_{:,1:r}$  (in the full version):

$$\mathcal{C}([\hat{\mathbf{a}}_1, \hat{\mathbf{a}}_2, \dots, \hat{\mathbf{a}}_r]) = \mathcal{C}(\mathbf{Q}_r) = \mathcal{C}(\mathbf{Q}_{:,1:r}).$$

The matrix  $\mathbf{R}_{12}$  recovers the dependent  $n - r$  columns from the column space of  $\mathbf{Q}_r$  or  $\mathbf{Q}_{:,1:r}$ . Figure 3.5 compares the reduced and full CPQR decompositions, where silent columns in  $\mathbf{Q}$  are shown in gray, blank entries represent zeros, and blue/orange entries denote elements that are not necessarily zero.



**Figure 3.5:** Comparison between the reduced and full CPQR decompositions.

### 3.8.2 A Practical CPQR via CGS

We observe that the simple CPQR algorithm pivots the first  $r$  independent columns to become the first  $r$  columns of  $\mathbf{AP}$ . Let  $\mathbf{A}_1$  represent the first  $r$  columns of  $\mathbf{AP}$ , and  $\mathbf{A}_2$  represent the remaining columns. In the context of the full CPQR decomposition, we have:

$$\mathbf{AP} = [\mathbf{A}_1, \mathbf{A}_2] = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \left[ \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} \\ \mathbf{0} \end{bmatrix}, \mathbf{Q} \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{0} \end{bmatrix} \right].$$

It is evident 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 second equality follows from the invariance of the matrix norm under orthogonal transformations. Therefore, the norm of  $\mathbf{R}_{12}$  is directly determined by the norm of  $\mathbf{A}_2$ . For a well-conditioned CPQR, it is desirable for  $\mathbf{R}_{12}$  to have a small norm. A practical CPQR algorithm achieves this by first permuting the columns of  $\mathbf{A}$  so that they are ordered in decreasing 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

$$\|\mathbf{a}_{j_1}\| \geq \|\mathbf{a}_{j_2}\| \geq \dots \geq \|\mathbf{a}_{j_n}\|.$$

The “simple” reduced CPQR decomposition process is then applied to  $\tilde{\mathbf{A}}$ , resulting in  $\tilde{\mathbf{A}}\mathbf{P}_1 = \mathbf{Q}_r[\mathbf{R}_{11}, \mathbf{R}_{12}]$ . The “practical” reduced CPQR of  $\mathbf{A}$  is then recovered as

$$\mathbf{A} \underbrace{\mathbf{P}_0 \mathbf{P}_1}_P = \mathbf{Q}_r[\mathbf{R}_{11}, \mathbf{R}_{12}].$$

The CPQR algorithm can be further enhanced by using the MGS process. This improved approach has the additional advantage of stopping automatically when the factorization encounters a rank-deficient submatrix, thereby revealing the numerical rank of the matrix. This method is known as *partial factorization*; see, for example, Lu (2021c) for more details.

### 3.9. QR with Column Pivoting: Revealing Rank-One Deficiency

Column-pivoted QR (CPQR) is one of several methods used to determine an appropriate column permutation when the matrix  $\mathbf{A}$  is rank-deficient. This process rearranges the first  $r$  linearly independent columns of  $\mathbf{A}$  to occupy the first  $r$  columns of  $\mathbf{AP}$ . If  $\mathbf{A}$  is nearly rank-one deficient, the goal becomes identifying a column permutation of  $\mathbf{A}$  that minimizes the pivotal element  $r_{nn}$  in the resulting QR decomposition. This is commonly known as the *revealing rank-one deficiency* problem for *rank-revealing QR (RRQR)* decomposition.

The RRQR problem is particularly useful in the sense that it allows us to infer the numerical rank of a matrix without explicitly computing its singular value decomposition (SVD), which can be a significant advantage in terms of computational time and resources. In least squares problems, where one seeks the best approximate solution to an over-determined system of equations, the rank of the coefficient matrix plays a crucial role. An RRQR factorization can help identify the effective/numerical rank and thus the number of linearly independent equations, which is essential for solving such problems accurately. On the other hand, in statistical modeling and machine learning, selecting a subset of regressors that best explains the variability in the response variable is a common task. RRQR factorization can assist in identifying the most relevant subset of variables by revealing the rank structure of the matrix formed by these variables, e.g., finding independent and significant alpha signals for quantitative strategies (Lu and Osterrieder, 2022).

**Theorem 3.3: (Revealing rank-one deficiency (Chan, 1987))** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and let  $\mathbf{v} \in \mathbb{R}^n$  be a unit vector (i.e.,  $\|\mathbf{v}\| = 1$ ). There exists a permutation matrix  $\mathbf{P}$  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 element of  $\mathbf{R}$ . In this decomposition,  $\mathbf{Q} \in \mathbb{R}^{m \times n}$  and  $\mathbf{R} \in \mathbb{R}^{n \times n}$ .

**Proof** [of Theorem 3.3] Let  $\mathbf{P} \in \mathbb{R}^{n \times n}$  be a permutation matrix such that  $\mathbf{w} = \mathbf{P}^\top \mathbf{v}$ , where

$$|w_n| = \max |v_i|, \quad \forall i \in \{1, 2, \dots, n\}.$$

That is, we swap the entry with the largest magnitude to the last position, ensuring that the last component of  $\mathbf{w}$  equals 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,

$$\epsilon = \|\mathbf{Av}\| = \|(\mathbf{Q}^\top \mathbf{AP})(\mathbf{P}^\top \mathbf{v})\| = \|\mathbf{R}\mathbf{w}\| = \left\| \begin{bmatrix} \vdots \\ r_{nn} w_n \end{bmatrix} \right\| \geq |r_{nn} w_n| \geq |r_{nn}|/\sqrt{n},$$

where the second equality follows from the invariance of vector norms under orthogonal transformations, and  $\mathbf{P}$  is an orthogonal matrix satisfying  $\mathbf{PP}^\top = \mathbf{I}$ . This concludes the proof.  $\blacksquare$

The following discussion makes use of the singular value decomposition (SVD), which will

be introduced in Section 11.1. You may skip this paragraph on 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 singular values satisfying  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ , i.e.,  $\sigma_n$  is the smallest singular value, and  $\mathbf{u}_i$ 's and  $\mathbf{v}_i$ 's are the corresponding left and right singular vectors, respectively. Then, if we let  $\mathbf{v} = \mathbf{v}_n$  such that  $\mathbf{A}\mathbf{v}_n = \sigma_n \mathbf{u}_n$ ,<sup>3</sup> we have

$$\|\mathbf{A}\mathbf{v}\| = \sigma_n.$$

By constructing a permutation matrix  $\mathbf{P}$  satisfying

$$|\mathbf{P}^\top \mathbf{v}|_n = \max |v_i|, \quad \forall i \in \{1, 2, \dots, n\},$$

we obtain a QR decomposition of  $\mathbf{AP} = \mathbf{QR}$  where the pivotal element  $r_{nn}$  satisfies  $r_{nn} \leq \sqrt{n}\sigma_n$ . If  $\mathbf{A}$  is rank-one deficient ( $\sigma_n \approx 0$ ), then  $r_{nn}$  will also be close to zero, effectively revealing the matrix's near-rank deficiency.

### 3.10. QR with Column Pivoting: Revealing Rank-r Deficiency\*

Building on the previous section, we now focus on computing the reduced QR decomposition of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  that is approximately rank- $r$  deficient<sup>4</sup> with  $r > 1$ . The goal now becomes finding a permutation matrix  $\mathbf{P}$  such that:

$$\mathbf{AP} = \mathbf{QR} = \mathbf{Q} \begin{bmatrix} \mathbf{L} & \mathbf{M} \\ \mathbf{0} & \mathbf{N} \end{bmatrix}, \quad (3.8)$$

where  $\mathbf{N} \in \mathbb{R}^{r \times r}$ , and  $\|\mathbf{N}\|$  is small in some norm. A recursive algorithm can be employed to achieve this. Suppose we have already isolated a small  $k \times k$  block  $\mathbf{N}_k$ . If we can isolate a small  $(k+1) \times (k+1)$  block  $\mathbf{N}_{k+1}$ , the permutation matrix can be determined recursively. To reiterate, assume the existence of a permutation  $\mathbf{P}_k$  such that  $\mathbf{N}_k \in \mathbb{R}^{k \times k}$  has a small norm:

$$\mathbf{AP}_k = \mathbf{Q}_k \mathbf{R}_k = \mathbf{Q}_k \begin{bmatrix} \mathbf{L}_k & \mathbf{M}_k \\ \mathbf{0} & \mathbf{N}_k \end{bmatrix}.$$

Now, we aim 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:

$$\mathbf{AP}_{k+1} = \mathbf{Q}_{k+1} \mathbf{R}_{k+1} = \mathbf{Q}_{k+1} \begin{bmatrix} \mathbf{L}_{k+1} & \mathbf{M}_{k+1} \\ \mathbf{0} & \mathbf{N}_{k+1} \end{bmatrix}.$$

Using the algorithm described earlier, there exists an  $(n-k) \times (n-k)$  permutation matrix  $\tilde{\mathbf{P}}_{k+1}$  such that the matrix  $\mathbf{L}_k \in \mathbb{R}^{(n-k) \times (n-k)}$  has the QR decomposition  $\mathbf{L}_k \tilde{\mathbf{P}}_{k+1} = \tilde{\mathbf{Q}}_{k+1} \tilde{\mathbf{L}}_k$ , where the entry  $(n-k, n-k)$  of  $\tilde{\mathbf{L}}_k$  is small. We then construct the following:

$$\mathbf{P}_{k+1} = \mathbf{P}_k \begin{bmatrix} \tilde{\mathbf{P}}_{k+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{Q}_{k+1} = \mathbf{Q}_k \begin{bmatrix} \tilde{\mathbf{Q}}_{k+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}.$$

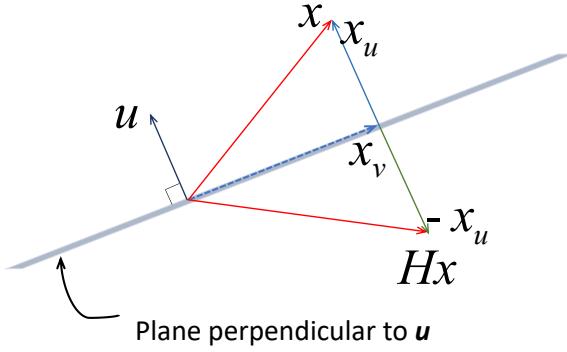
This leads to:

$$\mathbf{AP}_{k+1} = \mathbf{Q}_{k+1} \begin{bmatrix} \tilde{\mathbf{L}}_k & \tilde{\mathbf{Q}}_{k+1}^\top \mathbf{M}_k \\ \mathbf{0} & \mathbf{N}_k \end{bmatrix}.$$

Since the  $(n-k, n-k)$ -th entry of  $\tilde{\mathbf{L}}_k$  is small, proving that the last row of  $\tilde{\mathbf{Q}}_{k+1}^\top \mathbf{M}_k$  is also small in norm will reveal the rank- $(k+1)$  deficiency of  $\mathbf{A}$  (see Chan (1987) for a formal proof).

<sup>3</sup>. We will prove that the right singular vector of  $\mathbf{A}$  is equal to the right singular vector of  $\mathbf{R}$  if  $\mathbf{A}$  admits the QR decomposition  $\mathbf{A} = \mathbf{QR}$  in Lemma 11.11. The claim can also be applied to the singular values. So  $\mathbf{v}_n$  here is also a right singular vector of  $\mathbf{R}$ .

<sup>4</sup>. Note that rank  $r$  here means the matrix has a rank of  $(\min\{m, n\} - r)$ , not  $r$ .



**Figure 3.6:** Illustration of the Householder reflector. The Householder reflector obtained by  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ , where  $\|\mathbf{u}\| = 1$ , reflects a vector  $\mathbf{x}$  across the hyperplane orthogonal to  $\mathbf{u}$ :  $\mathbf{x} = \mathbf{x}_v + \mathbf{x}_u \rightarrow \mathbf{x}_v - \mathbf{x}_u$ .

### 3.11. Existence of QR Decomposition via Householder Reflector

*Householder matrices*, also known as *Householder reflectors*, are fundamental tools in numerical linear algebra. They are widely used in solving linear systems, estimating least squares solutions, and reducing matrices to Hessenberg form. This section illustrates how Householder reflectors can be utilized to prove the existence of the QR decomposition.

We begin by formally defining a Householder reflector and then examine its key properties.

**Definition 3.4 (Householder reflector).** Let  $\mathbf{u} \in \mathbb{R}^n$  be a vector of unit length (i.e.,  $\|\mathbf{u}\| = 1$ ). The matrix  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$  is called a *Householder reflector* or a *Householder transformation*. This matrix is associated with the unit vector  $\mathbf{u}$ , which is referred to as the *Householder vector*. When a vector  $\mathbf{x}$  is multiplied by  $\mathbf{H}$ , it is reflected across the hyperplane orthogonal to  $\text{span}\{\mathbf{u}\}$  (denoted as  $\text{span}\{\mathbf{u}\}^\perp$ ).

If  $\|\mathbf{u}\| \neq 1$ , the Householder reflector is defined as:  $\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}}$ .

Derived from the definition of the Householder reflector, we obtain the following corollary, indicating that certain vectors remain unaltered when subjected to the Householder reflector.

**Corollary 3.5: (Unreflected by Householder)** Given a unit vector  $\mathbf{u}$ , the Householder reflector  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$  leaves any vector  $\mathbf{v}$  that is orthogonal to  $\mathbf{u}$  unchanged. In other words, if  $\mathbf{u}^\top\mathbf{v} = 0$ , then  $\mathbf{H}\mathbf{v} = \mathbf{v}$ .

This result follows directly from substitution:  $(\mathbf{I} - 2\mathbf{u}\mathbf{u}^\top)\mathbf{v} = \mathbf{v} - 2\mathbf{u}\mathbf{u}^\top\mathbf{v} = \mathbf{v}$ .

Let  $\mathbf{u}$  be a unit vector with  $\|\mathbf{u}\| = 1$ , and let  $\mathbf{v}$  be a vector orthogonal to  $\mathbf{u}$ . Then any vector  $\mathbf{x}$  in the plane can be decomposed into two components:  $\mathbf{x} = \mathbf{x}_v + \mathbf{x}_u$ , where the first component  $\mathbf{x}_u$  is parallel to  $\mathbf{u}$  and the second one  $\mathbf{x}_v$  is orthogonal to  $\mathbf{u}$  (i.e., parallel to  $\mathbf{v}$ ). Using the projection formula from Section 3.2, the component parallel to  $\mathbf{x}_u$  is  $\mathbf{x}_u = \frac{\mathbf{u}\mathbf{u}^\top}{\mathbf{u}^\top\mathbf{u}}\mathbf{x} = \mathbf{u}\mathbf{u}^\top\mathbf{x}$ . Applying the Householder reflector  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$  to  $\mathbf{x}$ , we get:  $\mathbf{H}\mathbf{x} = (\mathbf{I} - 2\mathbf{u}\mathbf{u}^\top)(\mathbf{x}_v + \mathbf{x}_u) = \mathbf{x}_v - \mathbf{u}\mathbf{u}^\top\mathbf{x} = \mathbf{x}_v - \mathbf{x}_u$ . This demonstrates that the Householder reflector reflects  $\mathbf{x}$  across the hyperplane  $\text{span}\{\mathbf{u}\}^\perp$ . In other words, the subspace perpendicular to  $\mathbf{u}$  acts as a mirror, reflecting  $\mathbf{x}$ . This transformation is illustrated in Figure 3.6.

The previous explanation explains how to determine the reflection of a vector using a Householder reflector. However, an additional question arises: Given two vectors that are reflections of each other via a Householder transformation, how can we determine the corresponding Householder reflector? This question is especially relevant in the context of

QR decomposition, where the objective is often to transform a column of a matrix into a desired form—typically one with zeros below the diagonal.

**Corollary 3.6: (Finding the Householder reflector)** Suppose a vector  $\mathbf{x}$  is reflected to another vector  $\mathbf{y}$  by a Householder reflector, and assume that  $\|\mathbf{x}\| = \|\mathbf{y}\|$ . Then, the corresponding Householder reflector can be computed as:

$$\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top, \text{ where } \mathbf{u} = \frac{\mathbf{x} - \mathbf{y}}{\|\mathbf{x} - \mathbf{y}\|}.$$

**Proof** [of Corollary 3.6] Starting from the definition of the reflection, we have:

$$\mathbf{H}\mathbf{x} = \mathbf{x} - 2\mathbf{u}\mathbf{u}^\top\mathbf{x} = \mathbf{x} - 2\frac{(\mathbf{x} - \mathbf{y})(\mathbf{x}^\top - \mathbf{y}^\top)}{(\mathbf{x} - \mathbf{y})^\top(\mathbf{x} - \mathbf{y})}\mathbf{x} = \mathbf{x} - (\mathbf{x} - \mathbf{y}) = \mathbf{y}.$$

The condition  $\|\mathbf{x}\| = \|\mathbf{y}\|$  is necessary to ensure that this result holds true.  $\blacksquare$

Householder reflectors are particularly useful for zeroing out specific components of a vector. For example, it is often desirable to eliminate all elements of a vector  $\mathbf{a} \in \mathbb{R}^n$  except for the  $i$ -th element. In such cases, the Householder vector can be defined as:

$$\mathbf{u} = \frac{\mathbf{a} - r\mathbf{e}_i}{\|\mathbf{a} - r\mathbf{e}_i\|}, \quad \text{where } r = \pm \|\mathbf{a}\|,$$

which is a valid Householder vector since  $\|\mathbf{a}\| = \|r\mathbf{e}_i\| = |r|$ . Specifically, if  $r = \|\mathbf{a}\|$ , then the Householder reflector  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$  reflects  $\mathbf{a}$  to  $\|\mathbf{a}\|\mathbf{e}_i$ ; conversely, if  $r = -\|\mathbf{a}\|$ , the reflector maps  $\mathbf{a}$  to  $-\|\mathbf{a}\|\mathbf{e}_i$ .

**Remark 3.7 (Householder properties).** A Householder reflector  $\mathbf{H}$  has the following properties:

- $\mathbf{H}\mathbf{H} = \mathbf{I}$ : reflecting a vector twice yields the original vector.
- Symmetry:  $\mathbf{H} = \mathbf{H}^\top$ .
- Orthogonality:  $\mathbf{H}^\top\mathbf{H} = \mathbf{H}\mathbf{H}^\top = \mathbf{I}$ , meaning  $\mathbf{H}$  is an orthogonal matrix.
- $\mathbf{H}\mathbf{u} = -\mathbf{u}$ , if  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ .

**QR using Householder.** In the Gram–Schmidt process, we observe that the QR decomposition employs a triangular matrix to orthogonalize a given matrix  $\mathbf{A}$ . An alternative and computationally efficient approach involves constructing a sequence of orthogonal matrices that iteratively reduce  $\mathbf{A}$  to upper triangular form—this also results in a QR decomposition. For example, consider an orthogonal matrix  $\mathbf{Q}_1$  that introduces zeros into all elements of the first column of  $\mathbf{A}$  except for the entry (1,1); similarly, another orthogonal matrix  $\mathbf{Q}_2$  zeros out all entries of the second column below (2,2); and so forth. By applying such a sequence of transformations, we can obtain the QR decomposition of  $\mathbf{A}$ . This method uses reflections to align columns of the matrix with the standard basis vector  $\mathbf{e}_1$ , which has all entries equal to zero except for the first one.

To be more specific, let  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$  be the column partition of  $\mathbf{A}$ . Define:

$$r_1 = \|\mathbf{a}_1\|, \quad \mathbf{u}_1 = \frac{\mathbf{a}_1 - r_1\mathbf{e}_1}{\|\mathbf{a}_1 - r_1\mathbf{e}_1\|}, \quad \text{and} \quad \mathbf{H}_1 = \mathbf{I} - 2\mathbf{u}_1\mathbf{u}_1^\top. \quad (3.9)$$

Here,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^m$  denotes the first standard basis vector in  $\mathbb{R}^m$ . Applying the reflector  $\mathbf{H}_1$  to  $\mathbf{A}$  gives:

$$\mathbf{H}_1\mathbf{A} = [\mathbf{H}_1\mathbf{a}_1, \mathbf{H}_1\mathbf{a}_2, \dots, \mathbf{H}_1\mathbf{a}_n] = \begin{bmatrix} r_1 & \mathbf{R}_{1,2:n} \\ \mathbf{0} & \mathbf{B}_2 \end{bmatrix}. \quad (3.10)$$

This operation reflects  $\mathbf{a}_1$  to  $r_1\mathbf{e}_1$ , zeroing out all entries below the diagonal in the first column. Notably, we reflect  $\mathbf{a}_1$  to  $\|\mathbf{a}_1\|\mathbf{e}_1$ , where the two vectors have the same length (i.e., the transformation preserves the norm) rather than directly to  $\mathbf{e}_1$  to ensure numerical stability; and this aligns with the conditions stated in Corollary 3.6.

Next, we apply the same process to the submatrix  $\mathbf{B}_2$  from Equation (3.10), aiming to zero out all elements below the (2,2) entry. This selective application ensures that previously introduced zeros in the first column are preserved. Let  $\mathbf{B}_2 = [\mathbf{b}_2, \mathbf{b}_3, \dots, \mathbf{b}_n]$  be the column partition of  $\mathbf{B}_2$ , and define

$$r_2 = \|\mathbf{b}_2\|, \quad \mathbf{u}_2 = \frac{\mathbf{b}_2 - r_2\mathbf{e}_1}{\|\mathbf{b}_2 - r_2\mathbf{e}_1\|}, \quad \widetilde{\mathbf{H}}_2 = \mathbf{I} - 2\mathbf{u}_2\mathbf{u}_2^\top, \quad \text{and} \quad \mathbf{H}_2 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \widetilde{\mathbf{H}}_2 \end{bmatrix}.$$

In this context,  $\mathbf{e}_1$  now denotes the first unit basis in  $\mathbb{R}^{m-1}$ , and  $\mathbf{H}_2$  is orthogonal because  $\widetilde{\mathbf{H}}_2$  is orthogonal. Applying  $\mathbf{H}_2$  yields:

$$\mathbf{H}_2\mathbf{H}_1\mathbf{A} = [\mathbf{H}_2\mathbf{H}_1\mathbf{a}_1, \mathbf{H}_2\mathbf{H}_1\mathbf{a}_2, \dots, \mathbf{H}_2\mathbf{H}_1\mathbf{a}_n] = \begin{bmatrix} r_1 & r_{12} & \mathbf{R}_{1,3:n} \\ 0 & r_2 & \mathbf{R}_{2,3:n} \\ \mathbf{0} & \mathbf{0} & C_3 \end{bmatrix}.$$

By repeating this process iteratively, we eventually transform  $\mathbf{A}$  into upper triangular form:  $\mathbf{A} = (\mathbf{H}_n\mathbf{H}_{n-1}\dots\mathbf{H}_1)^{-1}\mathbf{R} = \mathbf{Q}\mathbf{R}$ . Since each  $\mathbf{H}_i$  is symmetric and orthogonal, the inverse simplifies to:  $\mathbf{Q} = (\mathbf{H}_n\mathbf{H}_{n-1}\dots\mathbf{H}_1)^{-1} = \mathbf{H}_1\mathbf{H}_2\dots\mathbf{H}_n$ .

For example, consider applying this method to a  $5 \times 4$  matrix. The transformation proceeds as follows, where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed:

$$\begin{array}{ccccc} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_1} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_2} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_3} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{0} & \boxtimes \\ 0 & 0 & \mathbf{0} & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_4} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & \mathbf{0} \end{bmatrix} \\ \mathbf{A} & & \mathbf{H}_1\mathbf{A} & & \mathbf{H}_2\mathbf{H}_1\mathbf{A} & & \mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A} & & \mathbf{H}_4\mathbf{H}_3\mathbf{H}_2\mathbf{H}_1\mathbf{A} \end{array}$$

The Householder algorithm is a powerful technique for transforming a matrix into upper triangular form using a sequence of orthogonal transformations. In contrast to the Gram–Schmidt process (both CGS and MGS), which employs a triangular matrix to orthogonalize a given matrix, the Householder algorithm relies on orthogonal matrices to achieve triangularization. This key distinction can be summarized as follows:

- Gram–Schmidt algorithm (triangular orthogonalization): Uses projections to orthogonalize vectors, resulting in a triangular matrix.
- Householder algorithm (orthogonal triangularization): Applies orthogonal transformations to triangularize the matrix.

Moreover, both the Householder algorithm and the Givens rotation method (to be discussed shortly) produce a *full* QR decomposition by applying a sequence of orthogonal transformations. In contrast, the QR decomposition obtained via CGS or MGS typically results in a *reduced* QR factorization. While it is possible to extend the reduced decomposition to full form by appending silent orthogonal columns or rows, this extension is not inherent to the CGS or MGS methods.

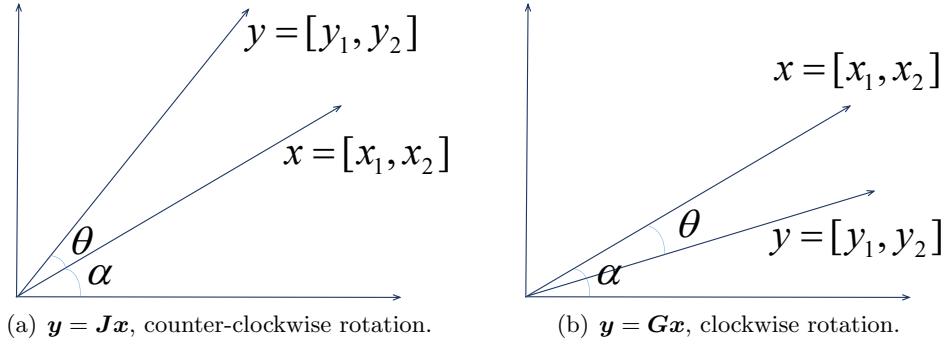
### 3.12. Existence of QR Decomposition via Givens Rotation

In Definition 2.15, we introduced the concept of a Givens rotation, particularly in the context of finding the rank-one update or downdate of the Cholesky decomposition. Let us

now examine the specific effects of Givens rotations through illustrative examples. Consider the following  $2 \times 2$  orthogonal matrices:

$$\mathbf{F} = \begin{bmatrix} -c & s \\ s & c \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}, \quad \text{and} \quad \mathbf{G} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix},$$

where  $s = \sin \theta$  and  $c = \cos \theta$  for some angle  $\theta$ . The first matrix has determinant  $\det(\mathbf{F}) = -1$  and represents a special case of a Householder reflector in two dimensions. It can be expressed as  $\mathbf{F} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top$ , where  $\mathbf{u} = \left[ \sqrt{\frac{1+c}{2}}, \sqrt{\frac{1-c}{2}} \right]^\top$  or  $\mathbf{u} = \left[ -\sqrt{\frac{1+c}{2}}, -\sqrt{\frac{1-c}{2}} \right]^\top$ . This matrix reflects vectors across a specific axis. In contrast, the matrices  $\mathbf{J}$  and  $\mathbf{G}$  have determinants  $\det(\mathbf{J}) = \det(\mathbf{G}) = 1$  and perform rotations rather than reflections. Such matrices are referred to as *Givens rotations*.



**Figure 3.7:** Illustration of two Givens rotations.

Figure 3.7(a) demonstrate the rotation of a vector  $\mathbf{x}$  under the action of the matrix  $\mathbf{J}$ , resulting in  $\mathbf{y} = \mathbf{J}\mathbf{x}$  with  $\begin{cases} y_1 = c \cdot x_1 - s \cdot x_2; \\ y_2 = s \cdot x_1 + c \cdot x_2. \end{cases}$  We aim to verify that the angle between the vectors  $\mathbf{x}$  and  $\mathbf{y}$  is indeed  $\theta$  (indicating a counter-clockwise rotation) under the Givens rotation  $\mathbf{J}$ . Firstly, we note that

$$\begin{cases} \cos(\alpha) = \frac{x_1}{\sqrt{x_1^2 + x_2^2}}; \\ \sin(\alpha) = \frac{x_2}{\sqrt{x_1^2 + x_2^2}}, \end{cases} \quad \text{and} \quad \begin{cases} \cos(\theta) = c; \\ \sin(\theta) = s. \end{cases}$$

This implies that  $\cos(\theta + \alpha) = \cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha)$ . If we can show that  $\cos(\theta + \alpha) = \cos(\theta)\cos(\alpha) - \sin(\theta)\sin(\alpha)$  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, we can verify 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 the vectors  $\mathbf{y} = \mathbf{G}\mathbf{x}$  and  $\mathbf{x}$  is also  $\theta$  in Figure 3.7(b), and the rotation is clockwise.

It can be easily verified that the  $n$ -th order Givens rotation (Definition 2.15) is an orthogonal matrix with determinant 1. For any vector  $\mathbf{x} = [x_1, x_2, \dots, x_n]^\top \in \mathbb{R}^n$ , applying a Givens rotation yields  $\mathbf{y} = \mathbf{G}_{kl}\mathbf{x}$  (the subscripts  $k, l$  indicate the rotations occur **in plane  $k$  and  $l$** ), where

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

In other words, a Givens rotation applied to  $\mathbf{x}$  rotates two components of  $\mathbf{x}$  by an angle  $\theta$ , while leaving all other components unchanged. When  $\sqrt{x_k^2 + x_l^2} \neq 0$ , let  $c = \frac{x_k}{\sqrt{x_k^2 + x_l^2}}$  and  $s = \frac{x_l}{\sqrt{x_k^2 + x_l^2}}$ . Then we have

$$\begin{cases} y_k = \sqrt{x_k^2 + x_l^2}, \\ y_l = 0, \\ y_j = x_j. \end{cases} \quad (j \neq k, l)$$

This result is critical for implementing the QR decomposition using Givens rotations.

**Corollary 3.8: (Basis transformation using forward Givens rotations)** 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 standard basis vector in  $\mathbb{R}^n$ .

**Proof** [of Corollary 3.8] From earlier results, we can construct a sequence of Givens rotations  $\mathbf{G}_{12}, \mathbf{G}_{13}$ , and  $\mathbf{G}_{14}$  such that

$$\begin{aligned} \mathbf{G}_{12} \mathbf{x} &= \left[ \sqrt{x_1^2 + x_2^2}, 0, x_3, \dots, x_n \right]^\top; \\ \mathbf{G}_{13} \mathbf{G}_{12} \mathbf{x} &= \left[ \sqrt{x_1^2 + x_2^2 + x_3^2}, 0, 0, x_4, \dots, x_n \right]^\top; \\ \mathbf{G}_{14} \mathbf{G}_{13} \mathbf{G}_{12} \mathbf{x} &= \left[ \sqrt{x_1^2 + x_2^2 + x_3^2 + x_4^2}, 0, 0, 0, x_5, \dots, x_n \right]^\top. \end{aligned}$$

By continuing this process, we eventually obtain:  $\mathbf{G}_{1n} \dots \mathbf{G}_{13} \mathbf{G}_{12} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$ . This completes the proof.  $\blacksquare$

**Remark 3.9 (Basis transformation using reverse Givens rotations).** In Corollary 3.8, the Givens rotations are applied in forward order, introducing zeros starting from the second entry up to the  $n$ -th entry. However, in some cases, it may be desirable to apply the rotations in reverse order, introducing zeros from the  $n$ -th entry down to the second 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 analogous to the forward case. Specifically, we construct Givens rotations  $\mathbf{G}_{1n}, \mathbf{G}_{1,(n-1)}, \mathbf{G}_{1,(n-2)}$  as follows:

$$\begin{aligned} \mathbf{G}_{1n} \mathbf{x} &= \left[ \sqrt{x_1^2 + x_n^2}, x_2, x_3, \dots, x_{n-1}, 0 \right]^\top; \\ \mathbf{G}_{1,(n-1)} \mathbf{G}_{1n} \mathbf{x} &= \left[ \sqrt{x_1^2 + x_{n-1}^2 + x_n^2}, x_2, x_3, \dots, x_{n-2}, 0, 0 \right]^\top; \\ \mathbf{G}_{1,(n-2)} \mathbf{G}_{1,(n-1)} \mathbf{G}_{1n} \mathbf{x} &= \left[ \sqrt{x_1^2 + x_{n-2}^2 + x_{n-1}^2 + x_n^2}, x_2, x_3, \dots, x_{n-3}, 0, 0, 0 \right]^\top. \end{aligned}$$

Continuing this process, we ultimately obtain:  $\mathbf{G}_{12} \mathbf{G}_{13} \dots \mathbf{G}_{1n} \mathbf{x} = \|\mathbf{x}\| \mathbf{e}_1$ .

Alternatively, there exists another sequence of 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

$$\begin{aligned}\mathbf{G}_{(n-1),n}\mathbf{x} &= \left[ x_1, x_2, \dots, x_{n-2}, \sqrt{x_{n-1}^2 + x_n^2}, 0 \right]^\top; \\ \mathbf{G}_{(n-2),(n-1)}\mathbf{G}_{(n-1),n}\mathbf{x} &= \left[ x_1, x_2, \dots, x_{n-3}, \sqrt{x_{n-2}^2 + x_{n-1}^2 + x_n^2}, 0, 0 \right]^\top; \\ \mathbf{G}_{(n-3),(n-2)}\mathbf{G}_{(n-2),(n-1)}\mathbf{G}_{(n-1),n}\mathbf{x} &= \left[ x_1, x_2, \dots, x_{n-4}, \sqrt{x_{n-3}^2 + x_{n-2}^2 + x_{n-1}^2 + x_n^2}, 0, 0, 0 \right]^\top.\end{aligned}$$

By continuing this process, we ultimately obtain:  $\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n}\mathbf{x} = \|\mathbf{x}\|\mathbf{e}_1$ .

This reverse application of Givens rotations will prove useful in the context of rank-one updates to the QR decomposition (Section 3.16).

**QR using Givens.** From Corollary 3.8, we know that we can introduce zeros by rotating the columns of a matrix to align with the basis vector  $\mathbf{e}_1$ . 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

$$\mathbf{G}_1 = \mathbf{G}_{1m} \dots \mathbf{G}_{13}\mathbf{G}_{12}. \quad (3.11)$$

Then,

$$\mathbf{G}_1\mathbf{A} = [\mathbf{G}_1\mathbf{a}_1, \mathbf{G}_1\mathbf{a}_2, \dots, \mathbf{G}_1\mathbf{a}_n] = \begin{bmatrix} \|\mathbf{a}_1\| & \mathbf{R}_{1,2:n} \\ \mathbf{0} & \mathbf{B}_2 \end{bmatrix}, \quad (3.12)$$

which rotates the first column  $\mathbf{a}_1$  to  $\|\mathbf{a}_1\|\mathbf{e}_1$ , introducing zeros below the diagonal in the first column.

Next, we apply this process to the submatrix  $\mathbf{B}_2$  from Equation (3.12), aiming to eliminate all entries below the (2,2) position. Suppose  $\mathbf{B}_2 = [\mathbf{b}_2, \mathbf{b}_3, \dots, \mathbf{b}_n]$ , 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 inferred from the context. Applying both rotations yields:

$$\mathbf{G}_2\mathbf{G}_1\mathbf{A} = [\mathbf{G}_2\mathbf{G}_1\mathbf{a}_1, \mathbf{G}_2\mathbf{G}_1\mathbf{a}_2, \dots, \mathbf{G}_2\mathbf{G}_1\mathbf{a}_n] = \begin{bmatrix} \|\mathbf{a}_1\| & r_{12} & \mathbf{R}_{1,3:n} \\ 0 & \|\mathbf{b}_2\| & \mathbf{R}_{2,3:n} \\ 0 & 0 & \mathbf{C}_3 \end{bmatrix}.$$

This procedure can be repeated iteratively until the entire matrix  $\mathbf{A}$  is upper triangularized. The final result is:  $\mathbf{A} = (\mathbf{G}_n\mathbf{G}_{n-1}\dots\mathbf{G}_1)^{-1}\mathbf{R} = \mathbf{Q}\mathbf{R}$ . Since each matrix  $\mathbf{G}_i$  is orthogonal for  $i \in \{1, 2, \dots, n\}$ , we have  $\mathbf{Q} = (\mathbf{G}_n\mathbf{G}_{n-1}\dots\mathbf{G}_1)^{-1} = \mathbf{G}_1^\top\mathbf{G}_2^\top\dots\mathbf{G}_n^\top$ , and

$$\begin{aligned}\mathbf{G}_1^\top\mathbf{G}_2^\top\dots\mathbf{G}_n^\top &= (\mathbf{G}_n \dots \mathbf{G}_2\mathbf{G}_1)^\top \\ &= \{(\mathbf{G}_{nm} \dots \mathbf{G}_{n,(n+1)}) \dots (\mathbf{G}_{2m} \dots \mathbf{G}_{23})(\mathbf{G}_{1m} \dots \mathbf{G}_{12})\}^\top.\end{aligned} \quad (3.13)$$

In practice, the Givens rotation algorithm often outperforms the Householder method when the matrix  $\mathbf{A}$  already contains many zeros below the main diagonal. Therefore, Givens rotations are particularly suited for rank-one changes in the QR decomposition, as these changes introduce only a small number of nonzero values (Section 3.16). An example of a  $5 \times 4$  matrix is presented below, 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 can express  $\mathbf{G}_1 = \mathbf{G}_{15}\mathbf{G}_{14}\mathbf{G}_{13}\mathbf{G}_{12}$ . The process is shown below:

$$\begin{array}{c} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ \square & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{G}_{12}} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{G}_{13}} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ 0 & \square & \square & \square \\ \mathbf{0} & \square & \square & \square \\ \square & \square & \square & \square \\ \square & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{G}_{14}} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & \square & \square & \square \\ \mathbf{0} & \square & \square & \square \\ \square & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{G}_{15}} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & \square & \square & \square \\ \mathbf{0} & \square & \square & \square \end{array} \right] \\ \mathbf{A} \qquad \qquad \qquad \mathbf{G}_{12}\mathbf{A} \qquad \qquad \qquad \mathbf{G}_{13}\mathbf{G}_{12}\mathbf{A} \qquad \qquad \qquad \mathbf{G}_{14}\mathbf{G}_{13}\mathbf{G}_{12}\mathbf{A} \qquad \qquad \qquad \mathbf{G}_{15}\mathbf{G}_{14}\mathbf{G}_{13}\mathbf{G}_{12}\mathbf{A} \end{array}.$$

**Givens rotation as a big picture.** When we consider  $\mathbf{G}_1, \mathbf{G}_2, \mathbf{G}_3, \mathbf{G}_4$  as a single matrix, we have:

$$\begin{array}{c} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ \square & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{G}_1} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ \mathbf{0} & \square & \square & \square \end{array} \right] \xrightarrow{\mathbf{G}_2} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & \mathbf{0} & \square & \square \\ 0 & 0 & \square & \square \\ 0 & 0 & \square & \square \end{array} \right] \xrightarrow{\mathbf{G}_3} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & 0 & \square & \square \\ 0 & 0 & \mathbf{0} & \square \\ 0 & 0 & 0 & \square \end{array} \right] \xrightarrow{\mathbf{G}_4} \left[ \begin{array}{cccc} \square & \square & \square & \square \\ 0 & \square & \square & \square \\ 0 & 0 & \square & \square \\ 0 & 0 & 0 & \square \\ 0 & 0 & 0 & \mathbf{0} \end{array} \right] \\ \mathbf{A} \qquad \qquad \qquad \mathbf{G}_1\mathbf{A} \qquad \qquad \qquad \mathbf{G}_2\mathbf{G}_1\mathbf{A} \qquad \qquad \qquad \mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{A} \qquad \qquad \qquad \mathbf{G}_4\mathbf{G}_3\mathbf{G}_2\mathbf{G}_1\mathbf{A} \end{array}.$$

### 3.13. Uniqueness of QR Decomposition

The results of QR decomposition can vary depending on the method used—such as the Gram–Schmidt process, the Householder algorithm, or the Givens algorithm. Even within the Householder algorithm, different strategies exist for selecting the sign of  $r_1$  in Equation (3.9). As a result, the QR decomposition of a matrix is not necessarily unique.

However, the uniqueness of the *reduced* QR decomposition for a full-column-rank matrix  $\mathbf{A}$  is guaranteed when the diagonal elements of  $\mathbf{R}$  are positive. Here, we provide a proof for the uniqueness of the reduced QR decomposition under the assumption that the diagonal elements of  $\mathbf{R}$  are positive. This proof also offers insight into the *implicit Q theorem* used in Hessenberg decomposition (Section 7.4) and tridiagonal decomposition (Section 7.7).

**Corollary 3.10: (Uniqueness of reduced QR decomposition)** Let  $\mathbf{A}$  be an  $m \times n$  matrix with full column rank  $n$ , where  $m \geq n$ . Then, the *reduced* QR decomposition is **unique** if the main diagonal values of  $\mathbf{R}$  are positive.

**Proof** [of Corollary 3.10] Assume that the reduced QR decomposition is not unique. Then, it can be extended to a full QR decomposition, and we can find two such decompositions satisfying  $\mathbf{A} = \mathbf{Q}_1\mathbf{R}_1 = \mathbf{Q}_2\mathbf{R}_2$ . This 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. Expanding this equation gives:

$$\mathbf{R}_1 = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & \dots & r_{2n} \\ 0 & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1m} \\ v_{21} & v_{22} & \dots & v_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \dots & v_{mm} \end{bmatrix} \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1n} \\ s_{21} & s_{22} & \dots & s_{2n} \\ 0 & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} = \mathbf{V}\mathbf{R}_2,$$

This implies

$$r_{11} = v_{11}s_{11}, \quad v_{21} = v_{31} = v_{41} = \dots = v_{m1} = 0.$$

Since  $\mathbf{V}$  is an orthogonal matrix and its columns are mutually orthonormal, with the first column having a norm of 1, it follows that  $v_{11} = \pm 1$ . By assumption,  $r_{ii} > 0$  and  $s_{ii} > 0$

for  $i \in \{1, 2, \dots, n\}$ , meaning that  $r_{11} > 0$  and  $s_{11} > 0$ , so  $v_{11}$  must be positive 1. Since  $\mathbf{V}$  is an orthogonal matrix, we also have

$$v_{12} = v_{13} = v_{14} = \dots = v_{1m} = 0.$$

By applying this reasoning to the submatrices of  $\mathbf{R}_1$ ,  $\mathbf{V}$ , and  $\mathbf{R}_2$ , we find that the upper-left submatrix of  $\mathbf{V}$  is the identity:  $\mathbf{V}[1 : n, 1 : n] = \mathbf{I}_n$ , indicating  $\mathbf{R}_1 = \mathbf{R}_2$ . This implies  $\mathbf{Q}_1[:, 1 : n] = \mathbf{Q}_2[:, 1 : n]$  and leads to a contradiction. Therefore, the reduced QR decomposition is unique.  $\blacksquare$

### 3.14. LQ, RRLQ, QL, RQ Decomposition

We previously established the existence of the QR decomposition using the Gram–Schmidt process, which is applied to the column space of a matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n}$ . However, in many applications (see, for example, [Schilders \(2009\)](#) or Chapter 4), there is also interest 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  $\mathbf{b}_i$  denotes 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 has an analogous counterpart that characterizes the orthogonal row space. If we perform the QR decomposition on  $\mathbf{B}^\top = \mathbf{Q}_0 \mathbf{R}$ , we obtain the LQ decomposition of the matrix  $\mathbf{B} = \mathbf{L} \mathbf{Q}$ , where  $\mathbf{Q} = \mathbf{Q}_0^\top$  and  $\mathbf{L} = \mathbf{R}^\top$ . The LQ decomposition is useful in demonstrating the existence of the UTV decomposition in the following chapter.

**Theorem 3.11: (LQ decomposition)** Any  $m \times n$  matrix  $\mathbf{B}$  (whether its rows are linearly independent or not) with  $n \geq m$  can be decomposed 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, 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, known as the *full LQ decomposition*. If we further restrict the lower triangular matrix to be square, the full LQ decomposition can be written as:

$$\mathbf{B} = [\mathbf{L}_0 \ 0] \mathbf{Q},$$

where  $\mathbf{L}_0$  is an  $m \times m$  square lower triangular matrix.

**Row-pivoted LQ (RPLQ).** Additionally, similar to the column-pivoted QR discussed in Section 3.8, there exists a *row-pivoted LQ (RPLQ)* decomposition for a rank- $r$  matrix  $\mathbf{B} \in \mathbb{R}^{m \times n}$ :

$$\left\{ \begin{array}{ll} \text{Reduced RPLQ:} & \mathbf{PB} = \underbrace{\begin{bmatrix} \mathbf{L}_{11} \\ \mathbf{L}_{21} \end{bmatrix}}_{m \times r} \underbrace{\mathbf{Q}_r}_{r \times n}; \\ \text{Full RPLQ:} & \mathbf{PB} = \underbrace{\begin{bmatrix} \mathbf{L}_{11} & 0 \\ \mathbf{L}_{21} & 0 \end{bmatrix}}_{m \times m} \underbrace{\mathbf{Q}}_{n \times n}, \end{array} \right.$$

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 interchanges independent rows into the upper-most rows.

**Exercise 3.12 (QL and RQ decomposition).** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $m \geq n$ . Show that there exists a permutation matrix  $\mathbf{P}$  such that  $\mathbf{AP} = \mathbf{QL}$ , where  $\mathbf{Q}$  is orthogonal and  $\mathbf{L}$  is lower triangular. Similarly, let  $\mathbf{B} \in \mathbb{R}^{m \times n}$  with  $m \leq n$ . Show that there exists a permutation matrix  $\mathbf{P}$  such that  $\mathbf{PB} = \mathbf{RQ}$ , where  $\mathbf{Q}$  is orthogonal and  $\mathbf{R}$  is upper triangular.

### 3.15. Two-Sided Orthogonal Decomposition

To this end, an immediate consequence of the CPQR and RPLQ decompositions is the *two-sided orthogonal decomposition*, which simultaneously identifies orthonormal bases for both the row space and column space of a matrix.

**Theorem 3.13: (Two-sided orthogonal decomposition)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a square matrix of rank  $r$ . Suppose the full CPQR and RPLQ decompositions of  $\mathbf{A}$  are given by

$$\mathbf{AP}_1 = \mathbf{Q}_1 \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{P}_2 \mathbf{A} = \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{0} \end{bmatrix} \mathbf{Q}_2,$$

respectively. Then, we obtain:

$$\mathbf{APA} = \mathbf{Q}_1 \underbrace{\begin{bmatrix} \mathbf{R}_{11}\mathbf{L}_{11} + \mathbf{R}_{12}\mathbf{L}_{21} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\text{rank } r} \mathbf{Q}_2,$$

where the first  $r$  columns of  $\mathbf{Q}_1$  span the same column space as  $\mathbf{A}$ , the first  $r$  rows of  $\mathbf{Q}_2$  span the same row space as  $\mathbf{A}$ , and  $\mathbf{P}$  is a permutation matrix. This decomposition is known as the *two-sided orthogonal decomposition*.

This decomposition exhibits a similarity with the singular value decomposition (SVD):  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where the first  $r$  columns of  $\mathbf{U}$  span the same column space as  $\mathbf{A}$ , and the first  $r$  columns of  $\mathbf{V}$  span the same row space as  $\mathbf{A}$  (as will be shown in Proposition 11.8). Thus, the two-sided orthogonal decomposition can be viewed as a computationally inexpensive alternative to the SVD for square matrices. The orthonormal bases from the two-sided orthogonal decomposition are collected in the following proposition.

**Proposition 3.14: (Four orthonormal basis)** Given the two-sided orthogonal decomposition of the 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  $\mathbf{V}$ , respectively, the following properties hold:

- $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  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  $\mathcal{N}(\mathbf{A}^\top)$ .

### 3.16. Application: Rank-One Changes

In Section 2.10, we discussed the rank-one update and downdate of the Cholesky decomposition. In the context of least squares problems, the Cholesky decomposition (along with its rank-one update/downdate) is applied to the Gram matrix of the data matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ :  $\mathbf{A}^\top \mathbf{A} = \mathbf{R}^\top \mathbf{R}$ ; see Section 11.6. Each row of the data matrix represents a data point, while each column corresponds to a feature variable. Therefore, the rank-one update/downdate of the Choleaky decomposition is useful for efficiently adding or removing a small number of data points from the least squares problem.

Similarly, a rank-one change of a matrix  $\mathbf{A}$  in the QR decomposition is defined as follows:

$$\begin{aligned}\mathbf{A}' &= \mathbf{A} + \mathbf{u}\mathbf{v}^\top, \\ &\quad \downarrow \quad \downarrow \\ \mathbf{Q}'\mathbf{R}' &= \mathbf{Q}\mathbf{R} + \mathbf{u}\mathbf{v}^\top,\end{aligned}$$

where the downdate form can be recovered by setting  $\mathbf{A}' = \mathbf{A} - (-\mathbf{u})\mathbf{v}^\top$ . This shows that the update and downdate forms in the QR decomposition are essentially equivalent. Since the least squares problem using the QR decomposition is applied directly to the data matrix ( $\mathbf{A} = \mathbf{Q}\mathbf{R}$ ; see Section 11.6), rank-one changes to the QR decomposition cannot be applied for adding or deleting a few data points from the least squares problem. However, it can be used for modifying the features in the system. For example, when  $\mathbf{u} = \mathbf{1}_m$  and  $\mathbf{v} = \mathbf{1}_n$ , the rank-one change effectively adds one to each feature across all data points.

To restate, the rank-one update/downdate problem involves determining the QR decomposition of  $\mathbf{A}'$  efficiently, given the QR decomposition of  $\mathbf{A}$ . Let  $\mathbf{w} = \mathbf{Q}^\top \mathbf{u}$ . Then we have

$$\mathbf{A}' = \mathbf{Q}(\mathbf{R} + \mathbf{w}\mathbf{v}^\top).$$

Using the second form from Remark 3.9, which introduces zeros in reverse order, 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  $\mathbf{G}_{(k-1),k}$  denotes the Givens rotation in the plane corresponding to indices  $k-1$  and  $k$ , and it zeros out the  $k$ -th entry of  $\mathbf{w}$ . Applying these same rotations 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 (backward rotations)* are useful to transform the upper triangular  $\mathbf{R}$  into a “simple” *upper Hessenberg matrix*—a nearly triangular matrix (see Definition 7.1 that will be introduced in the Hessenberg decomposition). In contrast, using forward rotations to transform  $\mathbf{w}$  into  $\pm \|\mathbf{w}\| \mathbf{e}_1$ , as demonstrated in Corollary 3.8, does not yield an upper Hessenberg matrix. Instead, it typically produces a dense matrix. For example, considering  $\mathbf{R} \in \mathbb{R}^{4 \times 4}$ , backward rotations preserve many zeros, simplifying the structure, where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed:

$$\text{Backward (Right Way)} : \begin{array}{c} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{34}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{23}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{12}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{R} \qquad \qquad \qquad \mathbf{G}_{34}\mathbf{R} \qquad \qquad \qquad \mathbf{G}_{23}\mathbf{G}_{34}\mathbf{R} \qquad \qquad \qquad \mathbf{G}_{12}\mathbf{G}_{23}\mathbf{G}_{34}\mathbf{R} \end{array}.$$

While forward rotations eliminate these zeros, resulting in a **dense matrix**:

$$\begin{array}{ll} \text{Forward} & : \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{12}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{23}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{array} \right] \xrightarrow{\mathbf{G}_{34}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right]. \\ (\text{Wrong Way}) & \quad \mathbf{R} \qquad \qquad \mathbf{G}_{12}\mathbf{R} \qquad \qquad \mathbf{G}_{23}\mathbf{G}_{12}\mathbf{R} \qquad \qquad \mathbf{G}_{34}\mathbf{G}_{23}\mathbf{G}_{12}\mathbf{R} \end{array}$$

In general, backward rotations yield:

$$\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n}(\mathbf{R} + \mathbf{wv}^\top) = \mathbf{H}_0 \pm \|\mathbf{w}\| \mathbf{e}_1 \mathbf{v}^\top = \mathbf{H},$$

which is also upper Hessenberg. Subsequently, as in the triangularization process via Givens rotation in Section 3.12, 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. To illustrate this process, following the example of a  $4 \times 4$  matrix, the sequence of rotations progressively simplifies the matrix, preserving and introducing zeros step by step:

$$\underbrace{\mathbf{H}_0 \pm \|\mathbf{w}\| \mathbf{e}_1 \mathbf{v}^\top}_{\mathbf{H}} = \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{J}_{12}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{J}_{23}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{J}_{34}} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{array} \right].$$

The QR decomposition of  $\mathbf{A}'$  can then be expressed as:  $\mathbf{A}' = \mathbf{Q}'\mathbf{R}'$ , where

$$\begin{cases} \mathbf{R}' = (\mathbf{J}_{(n-1),n} \dots \mathbf{J}_{23}\mathbf{J}_{12})(\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n})(\mathbf{R} + \mathbf{wv}^\top); \\ \mathbf{Q}' = \mathbf{Q} \{ (\mathbf{J}_{(n-1),n} \dots \mathbf{J}_{23}\mathbf{J}_{12})(\mathbf{G}_{12}\mathbf{G}_{23}\dots\mathbf{G}_{(n-1),n}) \}^\top. \end{cases} \quad (3.14)$$

### 3.17. Application: Appending or Deleting a Column

In certain applications, such as an  $F$ -test for least squares via QR decomposition (Lu, 2021d), it is often necessary to either delete or append a column (i.e., a feature variable) from the observed matrix. The goal, once again, is to efficiently compute the QR decomposition of the modified matrix.

**Deleting a column.** Suppose the QR decomposition of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is given by  $\mathbf{A} = \mathbf{QR}$ , where the columns of  $\mathbf{A}$  are partitioned as  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ . If the  $k$ -th column of  $\mathbf{A}$  is removed, the resulting matrix is denoted by  $\mathbf{A}' = [\mathbf{a}_1, \dots, \mathbf{a}_{k-1}, \mathbf{a}_{k+1}, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times (n-1)}$ . Our goal is to efficiently compute the QR decomposition of  $\mathbf{A}'$ . The matrix  $\mathbf{R}$  can be expressed using the following block structure:

$$\mathbf{R} = \left[ \begin{array}{ccc} \mathbf{R}_{11} & \mathbf{a} & \mathbf{R}_{12} \\ \mathbf{0} & r_{kk} & \mathbf{b}^\top \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{22} \end{array} \right] \begin{matrix} k-1 \\ 1 \\ m-k \end{matrix}, \quad \text{such that} \quad \mathbf{Q}^\top \mathbf{A}' = \left[ \begin{array}{cc} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{b}^\top \\ \mathbf{0} & \mathbf{R}_{22} \end{array} \right] = \mathbf{H}$$

is upper Hessenberg. An illustrative example is presented below, demonstrating the scenario of a  $6 \times 5$  matrix. Here,  $k = 3$ , and the column corresponding to  $k$  has been removed:

$$\begin{array}{c} \left[ \begin{array}{cccc|c} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \rightarrow \left[ \begin{array}{cccc|c} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \\ 0 & \boxtimes & \boxtimes & \boxtimes & \\ 0 & 0 & \boxtimes & \boxtimes & \\ 0 & 0 & 0 & \boxtimes & \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] . \\ \boldsymbol{R} = \boldsymbol{Q}^\top \boldsymbol{A} \quad \boldsymbol{H} = \boldsymbol{Q}^\top \boldsymbol{A}' \end{array}$$

To transform  $\boldsymbol{H}$  into a triangular matrix, we apply a sequence of Givens rotations  $\boldsymbol{G}_{k,k+1}, \boldsymbol{G}_{k+1,k+2}, \dots, \boldsymbol{G}_{n-1,n}$  to eliminate specific off-diagonal entries  $h_{k+1,k}, h_{k+2,k+1}, \dots, h_{n,n-1}$  of  $\boldsymbol{H}$ . The resulting triangular matrix  $\boldsymbol{R}'$  is then computed as:

$$\boldsymbol{R}' = \boldsymbol{G}_{n-1,n} \dots \boldsymbol{G}_{k+1,k+2} \boldsymbol{G}_{k,k+1} \boldsymbol{Q}^\top \boldsymbol{A}'.$$

The updated orthogonal matrix is given by:

$$\boldsymbol{Q}' = (\boldsymbol{G}_{n-1,n} \dots \boldsymbol{G}_{k+1,k+2} \boldsymbol{G}_{k,k+1} \boldsymbol{Q}^\top)^\top = \boldsymbol{Q} \boldsymbol{G}_{k,k+1}^\top \boldsymbol{G}_{k+1,k+2}^\top \dots \boldsymbol{G}_{n-1,n}^\top, \quad (3.15)$$

such that  $\boldsymbol{A}' = \boldsymbol{Q}' \boldsymbol{R}'$ . The  $6 \times 5$  example is shown below, where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{k=3} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \\ 0 & \boxtimes & \boxtimes & \boxtimes & \\ 0 & 0 & \boxtimes & \boxtimes & \\ 0 & 0 & 0 & \boxtimes & \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{\boldsymbol{G}_{34}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \\ 0 & \boxtimes & \boxtimes & \boxtimes & \\ 0 & 0 & \boxtimes & \boxtimes & \\ 0 & 0 & 0 & \boxtimes & \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{\boldsymbol{G}_{45}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \\ 0 & \boxtimes & \boxtimes & \boxtimes & \\ 0 & 0 & \boxtimes & \boxtimes & \\ 0 & 0 & 0 & \boxtimes & \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] . \\ \boldsymbol{R} = \boldsymbol{Q}^\top \boldsymbol{A} \quad \boldsymbol{H} = \boldsymbol{Q}^\top \boldsymbol{A}' \quad \boldsymbol{G}_{34} \boldsymbol{H} \quad \boldsymbol{G}_{45} \boldsymbol{G}_{34} \boldsymbol{H} \end{array}$$

**Appending a column.** Similarly, consider the case where a vector  $\boldsymbol{w}$  is appended as the  $(k+1)$ -th column of  $\boldsymbol{A}$ , resulting in the updated matrix  $\tilde{\boldsymbol{A}} = [\tilde{\boldsymbol{a}}_1, \tilde{\boldsymbol{a}}_k, \boldsymbol{w}, \tilde{\boldsymbol{a}}_{k+1}, \dots, \tilde{\boldsymbol{a}}_n]$ . The goal becomes to efficiently compute the QR decomposition of  $\tilde{\boldsymbol{A}}$ . Applying the orthogonal transformation  $\boldsymbol{Q}^\top$  to  $\tilde{\boldsymbol{A}}$ , we have

$$\boldsymbol{Q}^\top \tilde{\boldsymbol{A}} = [\boldsymbol{Q}^\top \tilde{\boldsymbol{a}}_1, \dots, \boldsymbol{Q}^\top \tilde{\boldsymbol{a}}_k, \boldsymbol{Q}^\top \boldsymbol{w}, \boldsymbol{Q}^\top \tilde{\boldsymbol{a}}_{k+1}, \dots, \boldsymbol{Q}^\top \tilde{\boldsymbol{a}}_n] = \tilde{\boldsymbol{H}}.$$

Next, a sequence of Givens rotations  $\boldsymbol{J}_{m-1,m}, \boldsymbol{J}_{m-2,m-1}, \dots, \boldsymbol{J}_{k+1,k+2}$  can be applied to zero out the elements  $\tilde{h}_{m,k+1}, \tilde{h}_{m-1,k+1}, \dots, \tilde{h}_{k+2,k+1}$  in  $\tilde{\boldsymbol{H}}$ , transforming it into an upper triangular matrix:

$$\tilde{\boldsymbol{R}} = \boldsymbol{J}_{k+1,k+2} \dots \boldsymbol{J}_{m-2,m-1} \boldsymbol{J}_{m-1,m} \boldsymbol{Q}^\top \tilde{\boldsymbol{A}}.$$

To illustrate, suppose  $\tilde{\boldsymbol{H}}$  is a  $6 \times 5$  matrix, and  $k = 2$ . Then the process is shown as follows:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & 0 & \boxtimes \\ 0 & 0 & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & 0 & 0 \end{array} \right] \xrightarrow{\boldsymbol{J}_{56}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & 0 & \boxtimes \\ 0 & 0 & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & 0 & 0 \end{array} \right] \xrightarrow{\boldsymbol{J}_{45}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & 0 & \boxtimes \\ 0 & 0 & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & 0 & 0 \end{array} \right] \xrightarrow{\boldsymbol{J}_{34}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & 0 & \boxtimes \\ 0 & 0 & \boxtimes & 0 & 0 \\ 0 & 0 & \boxtimes & 0 & 0 \end{array} \right] . \\ \tilde{\boldsymbol{H}} \quad \boldsymbol{J}_{56} \tilde{\boldsymbol{H}} \rightarrow \tilde{h}_{63} = 0 \quad \boldsymbol{J}_{45} \boldsymbol{J}_{56} \tilde{\boldsymbol{H}} \rightarrow \tilde{h}_{53} = 0 \quad \boldsymbol{J}_{34} \boldsymbol{J}_{45} \boldsymbol{J}_{56} \tilde{\boldsymbol{H}} \rightarrow \tilde{h}_{43} = 0 \end{array}$$

Finally, the updated orthogonal matrix is given by:

$$\tilde{\boldsymbol{Q}} = (\boldsymbol{J}_{k+1,k+2} \dots \boldsymbol{J}_{m-2,m-1} \boldsymbol{J}_{m-1,m} \boldsymbol{Q}^\top)^\top = \boldsymbol{Q} \boldsymbol{J}_{m-1,m}^\top \boldsymbol{J}_{m-2,m-1}^\top \dots \boldsymbol{J}_{k+1,k+2}^\top, \quad (3.16)$$

such that  $\tilde{\boldsymbol{A}} = \tilde{\boldsymbol{Q}} \tilde{\boldsymbol{R}}$ .

**Real world application.** This method is particularly valuable for efficient variable selection in least squares problems using QR decomposition. At each step, a column of the data matrix  $\mathbf{A}$  is removed, and an  $F$ -test is performed to assess the significance of the corresponding variable. Variables that are statistically insignificant are removed, leading to a simpler and more interpretable model (Lu, 2021d).

### 3.18. Application: Appending or Deleting a Row

Analogously, in the context of least squares problems using the QR decomposition (see Section 11.6), it may become necessary to append or delete a row (representing a data point) from the observed matrix. This is often done to evaluate how the updated data affects system performance or to accommodate an online data setting, in which data arrives sequentially. The objective, as before, is to efficiently compute the QR decomposition of the updated matrix.

**Appending a row.** Suppose the full QR decomposition of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is given by  $\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \mathbf{Q}\mathbf{R}$ , where  $\mathbf{A}_1 \in \mathbb{R}^{k \times n}$  and  $\mathbf{A}_2 \in \mathbb{R}^{(m-k) \times n}$ . Now, if we append a row, the resulting matrix becomes  $\mathbf{A}' = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{w}^\top \\ \mathbf{A}_2 \end{bmatrix} \in \mathbb{R}^{(m+1) \times n}$ . Our goal is to efficiently compute the full QR decomposition of  $\mathbf{A}'$ . To achieve this, we construct a permutation matrix:

$$\mathbf{P} = \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{m-k} \end{bmatrix} \longrightarrow \mathbf{P} \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{w}^\top \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{w}^\top \\ \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} \implies \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^\top \end{bmatrix} \mathbf{P} \mathbf{A}' = \begin{bmatrix} \mathbf{w}^\top \\ \mathbf{R} \end{bmatrix} = \mathbf{H},$$

such that  $\mathbf{H}$  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 in the elements  $h_{21}, h_{32}, \dots, h_{n+1,n}$  of  $\mathbf{H}$ . The triangular matrix  $\mathbf{R}'$  is given by

$$\mathbf{R}' = \mathbf{G}_{n,n+1} \dots \mathbf{G}_{23} \mathbf{G}_{12} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^\top \end{bmatrix} \mathbf{P} \mathbf{A}'.$$

The updated orthogonal matrix is then computed as

$$\mathbf{Q}' = \left( \mathbf{G}_{n,n+1} \dots \mathbf{G}_{23} \mathbf{G}_{12} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}^\top \end{bmatrix} \mathbf{P} \right)^\top = \mathbf{P}^\top \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \mathbf{G}_{12}^\top \mathbf{G}_{23}^\top \dots \mathbf{G}_{n,n+1}^\top,$$

such that  $\mathbf{A}' = \mathbf{Q}'\mathbf{R}'$  gives the QR decomposition of the updated matrix  $\mathbf{A}'$ .

**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  $\mathbf{A}_1 \in \mathbb{R}^{k \times n}$ ,  $\mathbf{A}_2 \in \mathbb{R}^{(m-k-1) \times n}$ , and the full QR decomposition is given by  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ , with  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  being orthogonal and  $\mathbf{R} \in \mathbb{R}^{m \times n}$  being upper triangular. We aim to compute the full QR decomposition of  $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$  efficiently (assuming  $m - 1 \geq n$ ). Similarly, to achieve this, we construct a permutation matrix  $\mathbf{P}$  as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{m-k-1} \end{bmatrix} \implies \mathbf{P}\mathbf{A} = \begin{bmatrix} \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{I}_k & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}_{m-k-1} \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{w}^\top \\ \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{w}^\top \\ \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix} = \mathbf{P}\mathbf{Q}\mathbf{R} = \mathbf{M}\mathbf{R},$$

where  $\mathbf{M} = \mathbf{P}\mathbf{Q}$  is an orthogonal matrix. Let  $\mathbf{m}^\top$  denote the first row of  $\mathbf{M}$ . A series of Givens rotations,  $\mathbf{G}_{m-1,m}, \mathbf{G}_{m-2,m-1}, \dots, \mathbf{G}_{1,2}$ , can be applied to zero out the elements

$m_m, m_{m-1}, \dots, m_2$  of  $\mathbf{m}$ , resulting in  $\mathbf{G}_{1,2} \dots \mathbf{G}_{m-2,m-1} \mathbf{G}_{m-1,m} \mathbf{m} = \alpha \mathbf{e}_1$ , where  $\alpha = \pm 1$ . Consequently,

$$\mathbf{G}_{1,2} \dots \mathbf{G}_{m-2,m-1} \mathbf{G}_{m-1,m} \mathbf{R} = \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}_1 \end{bmatrix} \begin{array}{c} 1 \\ m-1 \end{array},$$

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 because  $\alpha = \pm 1$  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 its first column denoted as  $\mathbf{g}$ . Writing  $\mathbf{M}$  as the row partition  $\mathbf{M} = [\mathbf{m}^\top; \mathbf{m}_2^\top; \mathbf{m}_3^\top; \dots; \mathbf{m}_m^\top]$ , we have

$$\begin{aligned} \mathbf{m}^\top \mathbf{g} &= \pm 1 & \rightarrow & \mathbf{g} = \pm \mathbf{m}, \\ \mathbf{m}_i^\top \mathbf{m} &= 0, & \forall i \in \{2, 3, \dots, m\}. \end{aligned}$$

Thus, we can write:

$$\begin{aligned} \mathbf{P}\mathbf{A} &= \mathbf{M}\mathbf{R} = (\mathbf{M}\mathbf{G}_{m-1,m}^\top \mathbf{G}_{m-2,m-1}^\top \dots \mathbf{G}_{1,2}^\top)(\mathbf{G}_{1,2} \dots \mathbf{G}_{m-2,m-1} \mathbf{G}_{m-1,m} \mathbf{R}) \\ &= \begin{bmatrix} \alpha & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_1 \end{bmatrix} \begin{bmatrix} \mathbf{v}^\top \\ \mathbf{R}_1 \end{bmatrix} = \begin{bmatrix} \alpha \mathbf{v}^\top \\ \mathbf{Q}_1 \mathbf{R}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{w}^\top \\ \tilde{\mathbf{A}} \end{bmatrix}. \end{aligned}$$

This shows that  $\mathbf{Q}_1 \mathbf{R}_1$  is the full QR decomposition of  $\tilde{\mathbf{A}} = \begin{bmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \end{bmatrix}$ .

### 3.19. Application: Gauss–Newton and Levenberg–Marquardt Method

The QR decomposition is helpful for solving the Gauss–Newton and Levenberg–Marquardt methods for nonlinear least squares problems. In Section 11.6, we will introduce the (linear) least squares problem for linear systems:

$$\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{Ax} - \mathbf{b}\|_2^2. \quad (3.17)$$

When the residual  $\mathbf{r}(\mathbf{x})$  in Equation (3.17) is nonlinear, we obtain the *nonlinear least squares* problem <sup>5</sup>:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \left\{ f(\mathbf{x}) = \frac{1}{2} \|\mathbf{r}(\mathbf{x})\|_2^2 \right\}, \quad \mathbf{r}(\mathbf{x}) \in \mathbb{R}^m, \mathbf{x} \in \mathbb{R}^n, m \geq n.$$

When  $\mathbf{r}(\mathbf{x}) = \mathbf{Ax} - \mathbf{b}$ , this reduces to the linear least squares problem given in (3.17). The gradient and Hessian of  $f(\mathbf{x})$  are

$$\nabla f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top \mathbf{r}(\mathbf{x}) \quad \text{and} \quad \nabla^2 f(\mathbf{x}) = \mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \sum_{i=1}^m r_i(\mathbf{x}) \nabla^2 r_i(\mathbf{x}), \quad (3.18)$$

where  $\mathbf{J}(\mathbf{x}) \in \mathbb{R}^{m \times n}$  is the Jacobian matrix (see Problem 3.33). The standard Newton's method (see Section 2.12) is an iterative optimization algorithm. At the  $t$ -th iteration, the update is given by:

$$\mathbf{x}^{(t+1)} \leftarrow \mathbf{x}^{(t)} + \mathbf{d}^{(t)},$$

---

<sup>5</sup>. More details can refer to, for example, Madsen et al. (2004).

where  $(\nabla^2 f(\mathbf{x}^{(t)}))\mathbf{d}^{(t)} = -\nabla f(\mathbf{x}^{(t)})$  determines the “candidate” descent direction  $\mathbf{d}^{(t)}$ . For brevity, we omit the superscript  $t$  and apply a linear Taylor’s approximation:

$$\nabla f(\mathbf{x} + \mathbf{d}) \approx \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})^\top \mathbf{d}.$$

Therefore, Newton’s method can be interpreted as finding a direction  $\mathbf{d}$  such that  $\nabla f(\mathbf{x} + \mathbf{d}) = \nabla f(\mathbf{x}) + \nabla^2 f(\mathbf{x})^\top \mathbf{d}$  approaches  $\mathbf{0}$  (i.e., a *stationary point*). To see this, taking the quadratic Taylor’s approximation, we have

$$f(\mathbf{x} + \mathbf{d}) = f(\mathbf{x}) + \nabla f(\mathbf{x})^\top \mathbf{d} + \frac{1}{2} \mathbf{d}^\top \nabla^2 f(\mathbf{x}) \mathbf{d} + o(\|\mathbf{d}\|_2^2). \quad (3.19)$$

If  $\mathbf{x}$  is a stationary point, then  $\nabla f(\mathbf{x}) = \mathbf{0}$ . Suppose further that the Hessian of  $f(\mathbf{x})$  is positive definite:  $\nabla^2 f(\mathbf{x}) \succ 0$ ; this implies that the smallest eigenvalue  $\lambda_{\min}$  of  $\nabla^2 f(\mathbf{x})$  satisfies  $\lambda_{\min} > 0$  (see Section 2.2), and  $\mathbf{d}^\top \nabla^2 f(\mathbf{x}) \mathbf{d} \geq \lambda \|\mathbf{d}\|_2^2$  for all  $\lambda_{\min} > \lambda > 0$ . This in turn implies that the third term in (3.19) dominates the fourth term. Therefore,  $\mathbf{x}$  is a *local minimizer* (a minimum point within a neighborhood of  $\mathbf{x}$  with some radius  $r$ ) when  $\mathbf{x}$  is a stationary point and  $\nabla^2 f(\mathbf{x})$  is positive definite (as long as  $\|\mathbf{d}\|$  is small enough).

**Gauss–Newton method.** However, since the Hessian  $\nabla^2 r_i(\mathbf{x})$  can be difficult to compute or intractable, the *Gauss–Newton method* approximates the Hessian  $\nabla^2 f(\mathbf{x})$  using only  $\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x})$ . This leads to the following equation for determining the “candidate” descent direction

$$\mathbf{J}(\mathbf{x}^{(t)})^\top \mathbf{J}(\mathbf{x}^{(t)}) \mathbf{d}^{(t)} = -\mathbf{J}(\mathbf{x}^{(t)})^\top \mathbf{r}(\mathbf{x}^{(t)}).$$

The “candidate” descent direction can also be equivalently obtained by solving the following optimization problem:

$$(\text{Gauss–Newton}): \quad \mathbf{d}^{(t)} = \arg \min_{\mathbf{d}} \left\| \mathbf{J}(\mathbf{x}^{(t)}) \mathbf{d} + \mathbf{r}(\mathbf{x}^{(t)}) \right\|_2^2, \quad (3.20)$$

which is a linear least squares problem and can be solved using QR decomposition (see Theorem 11.16, when  $\mathbf{J}(\mathbf{x}^{(t)})$  has full rank). Let  $\mathbf{J}(\mathbf{x}^{(t)})$  admit the reduced QR decomposition  $\mathbf{J}(\mathbf{x}^{(t)}) = \mathbf{Q}^{(t)} \mathbf{R}^{(t)}$ . Then the “candidate” descent direction can be obtained by

$$\mathbf{d}^{(t)} \leftarrow -(\mathbf{R}^{(t)})^{-1} (\mathbf{Q}^{(t)})^\top \mathbf{r}(\mathbf{x}^{(t)}).$$

This approach avoids the need to explicitly compute the inverse of  $\mathbf{J}(\mathbf{x}^{(t)})^\top \mathbf{J}(\mathbf{x}^{(t)})$ . When  $(\mathbf{d}^{(t)})^\top \nabla f(\mathbf{x}^{(t)}) \leq 0$ , the direction  $\mathbf{d}^{(t)}$  is called a “true” descent direction (as opposed to the “candidate” descent direction we used previously). We can verify that when  $\mathbf{J}(\mathbf{x}^{(t)})$  has full rank  $n$  (since  $m \geq n$ ), we have

$$(\mathbf{d}^{(t)})^\top \nabla f(\mathbf{x}^{(t)}) = (\mathbf{d}^{(t)})^\top \mathbf{J}(\mathbf{x}^{(t)})^\top \mathbf{r}(\mathbf{x}^{(t)}) = -\left\| \mathbf{J}(\mathbf{x}^{(t)}) \mathbf{d}^{(t)} \right\|_2^2 \leq 0.$$

Therefore, the resulting direction  $\mathbf{d}^{(t)}$  is indeed a “true” descent direction.

**Levenberg–Marquardt (LM) method.** Additionally, the *Levenberg–Marquardt method* also addresses the same problem in (3.20), but introduces an additional constraint  $\|\mathbf{d}\|_2 \leq \Delta^{(t)}$  (Levenberg, 1944; Marquardt, 1963; Wright and Holt, 1985):

$$(\text{LM-1}): \quad \mathbf{d}^{(t)} = \arg \min_{\mathbf{d}} \left\| \mathbf{J}(\mathbf{x}^{(t)}) \mathbf{d} + \mathbf{r}(\mathbf{x}^{(t)}) \right\|_2^2, \quad \text{s.t.} \quad \|\mathbf{d}\|_2 \leq \Delta^{(t)}. \quad (3.21)$$

This is equivalently to, using Lagrange multiplier, the following problem

$$\begin{aligned} \text{(LM-2): } \mathbf{d}^{(t)} &= \arg \min_{\mathbf{d}} \left\| \mathbf{J}(\mathbf{x}^{(t)}) \mathbf{d} + \mathbf{r}(\mathbf{x}^{(t)}) \right\|_2^2 + \lambda \|\mathbf{d}\|_2^2 \\ &= \arg \min_{\mathbf{d}} \left\| \begin{bmatrix} \mathbf{J}(\mathbf{x}^{(t)}) \\ \sqrt{\lambda} \mathbf{I} \end{bmatrix} \mathbf{d} + \begin{bmatrix} \mathbf{r}(\mathbf{x}^{(t)}) \\ \mathbf{0} \end{bmatrix} \right\|_2^2, \end{aligned} \quad (3.22)$$

where  $\lambda$  is a Lagrange multiplier associated with the trust-region radius  $\Delta^{(t)}$ . The second form above represents an updated least squares problem. Given the knowledge of the QR decomposition of  $\mathbf{J}(\mathbf{x}^{(t)}) = \mathbf{Q}^{(t)} \mathbf{R}^{(t)}$ , the least squares problem can be solved using the *update of least squares problems* (i.e., appending rows to the existing data matrix; see Section 3.18).

### 3.20. Application: Low-Rank Approximation

We will discuss low-rank approximation or dimensionality reduction in more detail in Section 11.8 and Chapter 12. The QR decomposition of a data matrix (which may be triangular) can also be used to construct a low-rank approximation of that matrix. In this context, the goal is to approximate a large matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with a low-rank matrix  $\tilde{\mathbf{A}}$  of rank  $k \ll \min(m, n)$ . This is particularly useful when  $\mathbf{A}$  is too large to store or process directly, or when the data in  $\mathbf{A}$  approximately lies in a lower-dimensional subspace. In such cases, we may wish to compress the data, reduce noise, or accelerate downstream computations.

To achieve this, we will introduce the *truncated SVD* in Section 11.8. Given the SVD of  $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^\top$ , we keep only the top  $k$  singular values:  $\mathbf{A} \approx \tilde{\mathbf{A}}_k = \mathbf{U}_k \Sigma_k \mathbf{V}_k^\top$ , where  $\mathbf{U}_k \in \mathbb{R}^{m \times k}$ ,  $\Sigma_k \in \mathbb{R}^{k \times k}$ , and  $\mathbf{V}_k \in \mathbb{R}^{n \times k}$ . This gives the best rank- $k$  approximation to  $\mathbf{A}$  (in terms of Frobenius or spectral norm). But it's computationally expensive, requiring  $\mathcal{O}(mn^2)$  operations.

In such cases, we seek a faster method to compute an approximate basis for the column space of  $\mathbf{A}$ . One efficient approach is to use a randomized range finder combined with QR decomposition:

- *Generate a random test matrix.* Let  $\Omega \in \mathbb{R}^{n \times k}$  be a random Gaussian matrix or structured random matrix (e.g., subsampled Hadamard; see, for example, [Mahoney \(2016\)](#)).
- *Form a sample matrix.* Compute  $\mathbf{Y} = \mathbf{A} \Omega \in \mathbb{R}^{m \times k}$ , which projects  $\mathbf{A}$  onto random directions. This means each column of  $\mathbf{Y}$  is a random linear combination of the columns of  $\mathbf{A}$ . If the top  $k$ -dimensional column space dominates, then  $\mathbf{Y}$  will “capture” most of it.
- *Compute a reduced QR decomposition of  $\mathbf{Y}$ .* Let  $\mathbf{Y} = \mathbf{Q} \mathbf{R}$ , where  $\mathbf{Q} \in \mathbb{R}^{m \times k}$  with orthonormal columns (i.e.,  $\mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_k$ ), and  $\mathbf{R} \in \mathbb{R}^{k \times k}$ . Now, the columns of  $\mathbf{Q}$  form an orthonormal basis for an approximate column space of  $\mathbf{A}$ . This step is sometimes called *orthonormalization of the sample space*.
- *Project  $\mathbf{A}$  onto the subspace spanned by  $\mathbf{Q}$ .* Compute  $\tilde{\mathbf{A}} = \mathbf{Q} \mathbf{Q}^\top \mathbf{A}$ , where  $\mathbf{Q} \mathbf{Q}^\top$  is an orthogonal projector onto the  $k$ -dimensional subspace spanned by  $\mathbf{Q}$  (see Section 3.5). That is,  $\tilde{\mathbf{A}} \in \mathbb{R}^{m \times n}$  is a rank- $k$  approximation to  $\mathbf{A}$  ([Drineas et al., 2006](#)).
- Optionally, compute a small matrix  $\mathbf{B} \in \mathbb{R}^{k \times n}$ :

$$\mathbf{B} = \mathbf{Q}^\top \mathbf{A} \implies \tilde{\mathbf{A}} = \mathbf{Q} \mathbf{B}.$$

Now we've reduced the problem to a small matrix  $\mathbf{B}$ , making further computations (e.g. SVD, regression, classification, clustering) more efficient.

The randomized QR algorithm has a computational cost of  $\mathcal{O}(mnk)$ , which is faster than the truncated SVD algorithm, whose cost is  $\mathcal{O}(mn^2)$  for approximating the matrix. This efficiency can be very beneficial in practice. For example, suppose we have a large document-term matrix  $\mathbf{A} \in \mathbb{R}^{100000 \times 10000}$  from some natural language processing (NLP) tasks. We can compute  $\mathbf{Q} \in \mathbb{R}^{100000 \times 200}$  that captures the dominant 200-dimensional structure using randomized QR, and then work with  $\mathbf{Q}^\top \mathbf{A} \in \mathbb{R}^{200 \times 10000}$  instead—greatly reducing both time and memory requirements.

## 🌊 Chapter 3 Problems 🌊

1. **Orthogonal projection.** Prove that an orthogonal projection  $\mathbf{H}$  is an idempotent and symmetric matrix such that  $\mathbf{H}\mathbf{v} \perp (\mathbf{v} - \mathbf{H}\mathbf{v})$  and  $\mathbf{H}\mathbf{v} \in \mathcal{C}(\mathbf{H})$  for any vector  $\mathbf{v} \notin \mathcal{C}(\mathbf{H})$ .
2. **Adjugate of orthogonal.** Let  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  be orthogonal. Show that  $\text{adj}(\mathbf{Q}) = \det(\mathbf{Q})\mathbf{Q}^\top$  such that  $\text{adj}(\mathbf{Q})$  is also orthogonal (Definition 0.23).
3. Prove that if  $\mathbf{A}$  is triangular and orthogonal, then  $\mathbf{A}$  must be diagonal.
4. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be skew-symmetric ( $\mathbf{A}^\top = -\mathbf{A}$ ). Show that the matrix  $(\mathbf{I} - \mathbf{A})^{-1}(\mathbf{I} + \mathbf{A})$  is orthogonal.
5. Let  $\mathbf{u}$  and  $\mathbf{v}$  be two orthogonal unit vectors. Show that  $\mathbf{u} + \mathbf{v}$  is orthogonal to  $\mathbf{u} - \mathbf{v}$ .
6. **Reflector.** Let  $\mathbf{u} \in \mathbb{R}^n$  and  $\mathbf{v} \in \mathbb{R}^n$  be two orthogonal vectors (not necessarily unit), where  $\mathbf{u} \in \mathcal{V}$  and  $\mathbf{v} \in \mathcal{V}^\perp$ . Define  $\mathbf{a} = \mathbf{u} + \mathbf{v}$  and  $\mathbf{b} = \mathbf{u} - \mathbf{v}$ . Show that there exists a unique Householder reflector  $\mathbf{H} \in \mathbb{R}^{n \times n}$  (Definition 3.4) such that  $\mathbf{H}\mathbf{a} = \mathbf{b}$ . Moreover, if  $\mathcal{V} = \{\mathbf{w}\}^\perp$ , show that  $\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{w}\mathbf{w}^\top}{\mathbf{w}^\top \mathbf{w}}$ .
7. Let  $\mathbf{u} = [-\sin(\theta), \cos(\theta)]^\top$  be a unit vector. Show that the Householder reflector determined by  $\mathbf{u}$  is  $\mathbf{H} = \begin{bmatrix} \cos(2\theta) & \sin(2\theta) \\ \sin(2\theta) & -\cos(2\theta) \end{bmatrix}$ .
8. Let  $\mathbf{Q}, \mathbf{U} \in \mathbb{R}^{m \times n}$  be two semi-orthogonal matrices with  $m \geq n$ . Show that  $\mathbf{Q}$  and  $\mathbf{U}$  have the same column space if and only if there exists an orthogonal matrix  $\mathbf{P} \in \mathbb{R}^{n \times n}$  such that  $\mathbf{Q} = \mathbf{U}\mathbf{P}$ .
9. Let  $\mathbf{Q}, \mathbf{U} \in \mathbb{R}^{n \times n}$  be orthogonal. Show that there exists an orthogonal matrix  $\mathbf{P}$  such that  $\mathbf{Q} = \mathbf{P}\mathbf{U}$ .
10. Let  $\mathbf{Q}, \mathbf{U} \in \mathbb{R}^{m \times n}$  be two semi-orthogonal matrices with  $m \geq n$ . Show that there exists an orthogonal matrix  $\mathbf{P} \in \mathbb{R}^{m \times m}$  such that  $\mathbf{Q} = \mathbf{P}\mathbf{U}$ . Compare this result with Problems 3.8 and 3.9. *Hint: Complete the semi-orthogonal matrices into  $m \times m$  orthogonal matrices.*
11. Let  $\mathbf{A}$  admit the QR decomposition  $\mathbf{A} = \mathbf{QR}$ . Show that  $\mathbf{A} = \mathbf{QR}$  is normal ( $\mathbf{A}^\top \mathbf{A} = \mathbf{AA}^\top$ ) if and only if  $\mathbf{RQ}$  is normal.
12. Consider the partition of an orthogonal matrix  $\mathbf{Q} = \begin{bmatrix} \mathbf{A}_{p \times p} & \mathbf{B}_{p \times q} \\ \mathbf{C}_{q \times p} & \mathbf{D}_{q \times q} \end{bmatrix} \in \mathbb{R}^{n \times n}$ . Show that  $\text{rank}(\mathbf{B}) = \text{rank}(\mathbf{C})$  and  $\text{rank}(\mathbf{D}) = n + \text{rank}(\mathbf{A}) - 2p$ .
13. Consider the rank of matrices:
  - Suppose matrices  $\mathbf{A}$  and  $\mathbf{B}$  have full column ranks. Show that  $\mathbf{AB}$  has full column rank.
  - Suppose  $\mathbf{AB}$  has full column ranks. Show that  $\mathbf{B}$  also has full column rank, but  $\mathbf{A}$  may not necessarily have full column rank.
  - Discuss the rank of the upper triangular matrices obtained from the QR decompositions of  $\mathbf{AB}$ ,  $\mathbf{A}$ , and  $\mathbf{B}$  in various cases of the matrices involved.

14. In Theorem 3.1, we stated that  $\mathbf{R}$  is nonsingular in the reduced QR decomposition when  $\mathbf{A}$  has full column rank  $n$ . Suppose  $\mathbf{A}$  does not have full column rank. Examine the relationship between the rank of  $\mathbf{A}$  and the number of nonzero entries in  $\mathbf{R}$ .
15. Use the Gram–Schmidt process, Householder transformations, Givens rotations to find an orthonormal basis for the space spanned by the vectors

$$\mathbf{v}_1 = [1, 3, 7, 5]^\top, \quad \mathbf{v}_2 = [6, 3, 6, 3]^\top, \quad \mathbf{v}_3 = [5, 2, 7, 4]^\top.$$

16. **Distance between a vector and a hyperplane.** Given a nonzero vector  $\mathbf{a} \neq \mathbf{0} \in \mathbb{R}^n$  and a scalar  $\beta$ , define the hyperplane  $H(\mathbf{a}, \beta) = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}^\top \mathbf{x} + \beta = 0\}$ . For any  $\mathbf{y} \in \mathbb{R}^n$ , use the projection along a line (see Section 3.2) to show that the distance between  $\mathbf{y}$  and  $H(\mathbf{a}, \beta)$  is given by  $d(\mathbf{y}, H(\mathbf{a}, \beta)) = \frac{|\mathbf{a}^\top \mathbf{y} + \beta|}{\|\mathbf{a}\|_2}$ . Hint: Choose two random points on the plane and first show that  $\mathbf{a}$  is orthogonal to the plane.
17. Although we have used the fact that every orthogonal (or orthonormal) list of vectors is linearly independent throughout our discussions, provide a rigorous proof of this claim. Hint: Assume the vectors are linearly dependent and derive a contradiction.
18. Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be given with  $m \geq n$ . Provide an algorithm using Householder reflectors to compute an orthogonal matrix  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  such that  $\mathbf{A} = \mathbf{Q}\mathbf{L}$ , where  $\mathbf{L}[1:n, 1:n]$  is lower triangular and  $\mathbf{L}[n+1:m, :] = \mathbf{0}$ .
19. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with rank  $r$ . Show that  $\mathbf{A}$  is range-symmetric (i.e.,  $\mathcal{C}(\mathbf{A}) = \mathcal{C}(\mathbf{A}^\top)$ ) if and only if there exist a nonsingular matrix  $\mathbf{S} \in \mathbb{R}^{n \times n}$  and a nonsingular matrix  $\mathbf{M} \in \mathbb{R}^{r \times r}$  such that  $\mathbf{A} = \mathbf{S} \begin{bmatrix} \mathbf{M} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{S}^\top$ . Hint: Consider the QR decomposition of  $\mathbf{S} = \mathbf{Q}\mathbf{R}$ .
20. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be skew-symmetric (i.e.,  $\mathbf{A}^\top = -\mathbf{A}$ ). Show that  $\mathbf{I} + \mathbf{A}$  is nonsingular,  $\mathbf{B} = (\mathbf{I} - \mathbf{A})(\mathbf{I} + \mathbf{A})^{-1}$  is orthogonal,  $\mathbf{I} + \mathbf{B} = 2(\mathbf{I} + \mathbf{A})^{-1}$ , and  $\det(\mathbf{B}) = 1$ .
21. Prove that the following statements about a square matrix  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  are equivalent:
- $\mathbf{Q}$  is orthogonal.
  - $\mathbf{Q}^\top$  is orthogonal.
  - $\mathbf{Q}$  is nonsingular and  $\mathbf{Q}^\top = \mathbf{Q}^{-1}$ .
  - The rows of  $\mathbf{Q}$  are orthonormal.
  - The columns of  $\mathbf{Q}$  are orthonormal.
  - For all  $\mathbf{x} \in \mathbb{R}^n$ , it follows that  $\|\mathbf{x}\|_2 = \|\mathbf{Q}\mathbf{x}\|_2$ .
22. **Orthogonal preservation.** Let  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  be orthogonal. Show that  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$  are orthogonal if and only if  $\mathbf{Q}\mathbf{x}$  and  $\mathbf{Q}\mathbf{y}$  are orthogonal.
23. **Orthogonal preservation.** Let  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  be orthogonal, and let  $\lambda$  be an eigenvalue of  $\mathbf{Q}$ . Show that  $\lambda = \pm 1$ , and  $\mathbf{x} \in \mathbb{R}^n$  is a (right) eigenvector of  $\mathbf{Q}$  associated with  $\lambda$  if and only if  $\mathbf{x}$  is a left eigenvector of  $\mathbf{Q}$  associated with  $\lambda$ .
24. **Inverses with QR decomposition.** Suppose you perform QR decomposition of an invertible  $n \times n$  matrix as  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ . Show how you can use this decomposition relationship for finding the inverse of  $\mathbf{A}$  by solving  $n$  different triangular systems of linear equations, each of which can be solved by back-substitution. Show how to compute the left or right inverse of a matrix with QR decomposition and back-substitution.
25. Use the results from Problems 1.12 and 1.13 to determine the computational complexity of QR decomposition using the CGS, MGS, Householder, and Givens approaches.
26. **Elementary row interchanging as a rotation and a reflection.** Prove that an  $n \times n$  elementary row interchange matrix can be expressed as the product of a  $90^\circ$  Givens rotation (i.e., of the form  $\mathbf{G}(i, j, \theta) = \mathbf{I} + (\cos(\theta) - 1)(\mathbf{e}_i \mathbf{e}_i^\top + \mathbf{e}_j \mathbf{e}_j^\top) +$

- $\sin(\theta)(\mathbf{e}_i \mathbf{e}_j^\top - \mathbf{e}_j \mathbf{e}_i^\top)$  with  $\theta = 90^\circ$ ) and a Householder reflector. Hint: We need to understand the properties and forms of these matrices.
27. **Givens geometric decomposition.** Show that all  $n \times n$  orthogonal matrices can be written as a product of at most  $\mathcal{O}(n^2)$  Givens rotations and at most a single elementary reflection matrix (obtained by negating one diagonal element of the identity matrix).
  28. **Householder geometric decomposition.** Show that all  $n \times n$  orthogonal matrices can be written as a product of at most  $n$  Householder reflectors.
  29. Demonstrate that a sequence of  $k$  Householder transformations, whose corresponding unit vectors are mutually orthonormal, can be represented as  $\mathbf{I} - 2\mathbf{Q}\mathbf{Q}^\top$ , where  $\mathbf{Q}$  is an  $n \times k$  semi-orthogonal matrix. Identify the  $(n - k)$ -dimensional plane across which this reflection occurs.
  30. Consider the  $4 \times 4$  Givens rotation matrix  $\mathbf{G}_{2,4}(90^\circ)$  (Definition 2.15). This matrix performs a  $90^\circ$  clockwise rotation of a 4-dimensional vector in the plane of the second and fourth dimensions (see Figure 3.7). Show how to obtain this matrix as the product of two Householder reflectors. Hint: Think geometrically.
  31. Consider two orthogonal matrices  $\mathbf{Q}_1 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$  and  $\mathbf{Q}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ . Are these matrices rotation or reflection matrices?
  32. Use Householder reflectors or Givens rotations to compute the LQ, QL, and RQ decompositions discussed in Section 3.14.
  33. Prove Equation (3.18), the gradient and Hessian of nonlinear least squares problems.  
Hint: Derive element-wise:

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \sum_{i=1}^m r_i(\mathbf{x}) \frac{\partial r_i(\mathbf{x})}{\partial x_j}, \quad \frac{\partial^2 f(\mathbf{x})}{\partial x_j \partial x_k} = \sum_{i=1}^m \left( \frac{\partial r_i(\mathbf{x})}{\partial x_j} \frac{\partial r_i(\mathbf{x})}{\partial x_k} + r_i(\mathbf{x}) \frac{\partial^2 r_i(\mathbf{x})}{\partial x_j \partial x_k} \right).$$

# 4

## UTV Decomposition: ULV and URV Decomposition

### 4.1. UTV Decomposition

The UTV decomposition generalizes the QR factorization of a matrix  $\mathbf{A}$  into two orthogonal matrices,  $\mathbf{U}$  and  $\mathbf{V}$ , and a (upper or lower) triangular matrix  $\mathbf{T}$ , such that  $\mathbf{A} = \mathbf{UTV}$ .<sup>1</sup> The triangular matrix  $\mathbf{T}$  supports rank estimation. The decomposition takes different forms depending on the triangular structure of  $\mathbf{T}$ : if  $\mathbf{T}$  is lower triangular, it is called the *ULV decomposition*; if  $\mathbf{T}$  is upper triangular, it is referred to as the URV decomposition. The UTV decomposition framework resembles the singular value decomposition (SVD; see Section 11.1) in structure and serves as a computationally efficient alternative to the SVD. Both methods can be applied to find the least squares solution for rank-deficient matrices (Theorem 11.17).

**Theorem 4.1: (Full ULV decomposition)** Any  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$  can be decomposed 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 orthogonal matrices, and  $\mathbf{L} \in \mathbb{R}^{r \times r}$  is a lower triangular matrix of full rank.

The existence of the ULV decomposition follows from those of the QR and LQ decomposition.

**Proof** [of Theorem 4.1] For any rank- $r$  matrix  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ , a column permutation matrix  $\mathbf{P}$  (Definition 0.19) can be used to reorder the columns of  $\mathbf{A}$ , placing its linearly independent columns in the first  $r$  positions of  $\mathbf{AP}$ . Without loss of generality, let  $\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r$  denote the  $r$  linearly independent columns of  $\mathbf{A}$ . Then,

$$\mathbf{AP} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r, \mathbf{b}_{r+1}, \dots, \mathbf{b}_n].$$

Define  $\mathbf{Z} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r] \in \mathbb{R}^{m \times r}$ . Since each  $\mathbf{b}_i$  lies in the column space of  $\mathbf{Z}$ , there exists a matrix  $\mathbf{E} \in \mathbb{R}^{r \times (n-r)}$  such that

$$[\mathbf{b}_{r+1}, \mathbf{b}_{r+2}, \dots, \mathbf{b}_n] = \mathbf{ZE}.$$

---

<sup>1</sup>. These decompositions belong to a class known as *double-sided orthogonal decomposition*. We will see the UTV decomposition, complete orthogonal decomposition (Theorem 4.3), and singular value decomposition are all instances of this framework.

Consequently,

$$\mathbf{A}\mathbf{P} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_r, \mathbf{b}_{r+1}, \dots, \mathbf{b}_n] = \mathbf{Z} [\mathbf{I}_r \quad \mathbf{E}],$$

where  $\mathbf{I}_r$  is the  $r \times r$  identity matrix. Additionally, the matrix  $\mathbf{Z} \in \mathbb{R}^{m \times r}$  has full column rank, so it admits the full QR decomposition:  $\mathbf{Z} = \mathbf{U} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$ , where  $\mathbf{R} \in \mathbb{R}^{r \times r}$  is an upper triangular matrix of full rank, and  $\mathbf{U}$  is an orthogonal matrix. Substituting this into the previous expression gives:

$$\mathbf{A}\mathbf{P} = \mathbf{Z} [\mathbf{I}_r \quad \mathbf{E}] = \mathbf{U} \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} [\mathbf{I}_r \quad \mathbf{E}] = \mathbf{U} \begin{bmatrix} \mathbf{R} & \mathbf{RE} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \quad (4.1)$$

Since  $\mathbf{R}$  has full rank,  $[\mathbf{R} \quad \mathbf{RE}]$  also has full rank. 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 orthogonal matrix. Substituting this into Equation (4.1), we have

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}_0 \mathbf{P}^{-1}.$$

Finally, let  $\mathbf{V} = \mathbf{V}_0 \mathbf{P}^{-1}$ , which is orthogonal since it is a product of two orthogonal matrices. This completes the proof.  $\blacksquare$

An alternative proof of the ULV decomposition will be discussed in Theorem 4.3 using the rank-revealing QR decomposition and the standard QR decomposition.

Now, suppose the ULV decomposition of a matrix  $\mathbf{A}$  is given 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  $\mathbf{V}_0 = \mathbf{V}_{1:r,:}$ , where  $\mathbf{U}_0$  consists of the first  $r$  columns of  $\mathbf{U}$ , and  $\mathbf{V}_0$  consists of the first  $r$  rows of  $\mathbf{V}$ . Then, we can write  $\mathbf{A} = \mathbf{U}_0 \mathbf{L} \mathbf{V}_0$ . This form is called the *reduced ULV decomposition*. Similarly, the URV decomposition can be derived as follows:

**Theorem 4.2: (URV decomposition)** Any  $m \times n$  matrix  $\mathbf{A}$  with rank  $r$  can be decomposed 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 of full rank. The *reduced URV decomposition* can be obtained as  $\mathbf{A} = \mathbf{U}_{:,1:r} \mathbf{R} \mathbf{V}_{1:r,:}$ .

The proof closely resembles that of the ULV decomposition, and is left as an exercise. Collectively, the ULV and URV decompositions are referred to as the UTV decomposition framework (Hanson and Lawson, 1969; Fierro and Hansen, 1997; Golub and Van Loan, 2013).

**Range and null space.** This decomposition framework, first introduced by Hanson and Lawson (1969), provides explicit orthogonal bases for the range and null space of  $\mathbf{A}$ , as well as a representation for the pseudo-inverse (see Problem 4.10). We will soon observe that the structures of ULV and URV decompositions closely resemble that of the singular value decomposition (SVD). All three decompositions factorize the matrix  $\mathbf{A}$  into two orthogonal matrices. More specifically, both ULV and URV decompositions provide orthonormal bases for the four fundamental subspaces of  $\mathbf{A}$ , as described in the fundamental theorem of linear algebra (Theorem 0.17). For example, in the ULV decomposition, the first  $r$  columns of  $\mathbf{U}$  form an orthonormal basis for the column space  $\mathcal{C}(\mathbf{A})$ , while the last  $(m - r)$  columns

of  $\mathbf{U}$  form an orthonormal basis for the left null space  $\mathcal{N}(\mathbf{A}^\top)$ . Similarly, the first  $r$  rows of  $\mathbf{V}$  form an orthonormal basis for the row space  $\mathcal{C}(\mathbf{A}^\top)$ , while the last  $(n - r)$  rows provide an orthonormal basis for the null space  $\mathcal{N}(\mathbf{A})$  (resembling the two-sided orthogonal decomposition; Theorem 3.13):

$$\begin{aligned}\mathcal{C}(\mathbf{A}) &= \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}, & \mathcal{N}(\mathbf{A}) &= \text{span}\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}, \\ \mathcal{C}(\mathbf{A}^\top) &= \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}, & \mathcal{N}(\mathbf{A}^\top) &= \text{span}\{\mathbf{u}_{r+1}, \mathbf{u}_{r+2}, \dots, \mathbf{u}_m\}.\end{aligned}\quad (4.2)$$

The SVD extends this framework by establishing direct relationships between the corresponding two pairs of orthonormal bases. It characterizes the linear transformations between the column space and row space, as well as between the left null space and (right) null space:  $\mathbf{A}\mathbf{v}_i = \sigma_i \mathbf{u}_i$  for all  $i$ . These connections will be explored in greater detail in the chapter on the SVD.

## 4.2. Complete Orthogonal Decomposition

The UTV decomposition is closely related to the concept of the *complete orthogonal decomposition*, which also involves factoring a matrix into two orthogonal matrices.

**Theorem 4.3: (Complete orthogonal decomposition)** Any  $m \times n$  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 a matrix of full rank  $r$ .

**Proof** [of Theorem 4.3] Using the column-pivoted QR decomposition (Theorem 3.2), the matrix  $\mathbf{A}$  can be decomposed 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. Next, we construct a decomposition that satisfies:

$$\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix} = \mathbf{Q}_2 \begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix}, \quad (4.3)$$

where  $\mathbf{Q}_2$  is an orthogonal matrix, and  $\mathbf{S}$  is a rank- $r$  matrix. This decomposition is valid because 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 entire space  $\mathbb{R}^n$ , where we can assume that 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  $\begin{bmatrix} \mathbf{S} \\ \mathbf{0} \end{bmatrix}$  serves to map  $\mathbf{Q}_2$  back to  $\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix}$ . Finally, substituting this decomposition, we find:  $\mathbf{Q}_1^\top \mathbf{A} \mathbf{P} \mathbf{Q}_2 = \begin{bmatrix} \mathbf{S}^\top & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ . Setting  $\mathbf{U} = \mathbf{Q}_1$ ,  $\mathbf{V} = \mathbf{Q}_2^\top \mathbf{P}^\top$ , and  $\mathbf{T} = \mathbf{S}^\top$ , we complete the proof. ■

Note that the complete orthogonal decomposition is quite general. When Equation (4.3) is interpreted as the reduced QR decomposition of  $\begin{bmatrix} \mathbf{R}_{11}^\top \\ \mathbf{R}_{12}^\top \end{bmatrix}$ , the complete orthogonal decomposition simplifies to the ULV decomposition.

## 4.3. Computing the UTV Decomposition

The CPQR decomposition introduced in Section 3.8 can be applied to find the UTV decomposition of a matrix. The CPQR factorization of a rank-deficient matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is

given by

$$\mathbf{A}\mathbf{P} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  is nonsingular ( $r < n$ ). Here  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  give orthogonal bases for  $\mathcal{C}(\mathbf{A})$  and  $\mathcal{N}(\mathbf{A}^\top)$ , respectively. However, this factorization is less useful for applications that need a basis for  $\mathcal{N}(\mathbf{A})$ . To address this, the off-diagonal block  $\mathbf{R}_{12}$  then can be annihilated by postmultiplying  $\mathbf{R}$  with a sequence of Householder reflectors:

$$[\mathbf{R}_{11}, \mathbf{R}_{12}] \mathbf{H}_r \dots \mathbf{H}_2 \mathbf{H}_1 = [\hat{\mathbf{R}}, \mathbf{0}], \quad \mathbf{H}_j = \mathbf{I} - 2\mathbf{u}_j \mathbf{u}_j^\top, \quad (4.4)$$

$j = r, r-1, \dots, 1$ , where each vector  $\mathbf{u}_j$  has nonzero entries only in positions  $j, r+1, \dots, n$ . This process is equivalent to performing a QL factorization on the transpose of the triangular factor  $\mathbf{R}$ :

$$\begin{bmatrix} \mathbf{R}_{11}^\top & \mathbf{0} \\ \mathbf{R}_{12}^\top & \mathbf{0} \end{bmatrix} = \hat{\mathbf{Q}} \begin{bmatrix} \hat{\mathbf{R}}^\top \\ \mathbf{0} \end{bmatrix}, \quad (4.5)$$

where the Householder reflectors are applied from the left rather than from the right. And this requires  $2r^2(n-r)$  flops (see Problem 4.7). As a result, we obtain a URV decomposition of the form:

$$\mathbf{A}\mathbf{P} = \hat{\mathbf{Q}} \begin{bmatrix} \hat{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^\top, \quad \mathbf{V} = \mathbf{H}_1 \mathbf{H}_2 \dots \mathbf{H}_r. \quad (4.6)$$

For example, the first three steps for a matrix with  $n = 6$  and  $r = 4$  in the reduction are shown below:

$$\begin{array}{cccc} \begin{bmatrix} \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_4} & \begin{bmatrix} \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \mathbf{0} \\ \boxtimes & \boxtimes & \boxtimes & \mathbf{0} \end{bmatrix} & \xrightarrow{\mathbf{H}_3} & \begin{bmatrix} \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \mathbf{0} & 0 \\ \boxtimes & \boxtimes & \mathbf{0} & 0 \end{bmatrix} & \xrightarrow{\mathbf{H}_2} & \begin{bmatrix} \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 \end{bmatrix} & \dots \\ \mathbf{A} & & \mathbf{H}_4 \mathbf{A} & & \mathbf{H}_3 \mathbf{H}_4 \mathbf{A} & & \mathbf{H}_2 \mathbf{H}_3 \mathbf{H}_4 \mathbf{A} & & \end{array}$$

Note that the application of  $\mathbf{H}_3$  does not affect the last column, and the premultiplication of  $\mathbf{H}_2$  does not affect the last two columns, as explained by Corollary 3.5.

**Exercise 4.4 (ULV).** Find a way to compute the ULV decomposition of a matrix.

#### 4.4. Rank-Revealing UTV Decomposition and Other Issues

**Rank-revealing URV.** For matrices  $\mathbf{A} \in \mathbb{R}^{m \times n}$  that are nearly rank-deficient with rank  $r < n$ , Stewart (2002) introduced the ranking-revealing URV decomposition. This decomposition takes the form

$$\mathbf{A}\mathbf{P} = \mathbf{U} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{V}^\top, \quad \mathbf{R}_{11} \in \mathbb{R}^{r \times r}, \quad (4.7)$$

where  $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2] \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2] \in \mathbb{R}^{n \times n}$  are orthogonal matrices, and  $\mathbf{R}_{11} \in \mathbb{R}^{r \times r}$  and  $\mathbf{R}_{22} \in \mathbb{R}^{(m-r) \times (n-r)}$  are upper triangular. If the singular values <sup>2</sup> of  $\mathbf{A}$  are ordered such that

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \gg \sigma_{r+1} \geq \dots \geq \sigma_n,$$

<sup>2</sup>. Once again, see Section 11.1 for more details.

then the decomposition (4.7) is said to be rank-revealing if it satisfies the following conditions:

$$\sigma_r(\mathbf{R}_{11}) \geq \sigma_r/c, \quad (\|\mathbf{R}_{12}\|_F^2 + \|\mathbf{R}_{22}\|_F^2)^{1/2} \leq c\sigma_{r+1},$$

where  $c$  is bounded by a low-degree polynomial in terms of  $r$  and  $n$ . For  $\mathbf{P} = \mathbf{I}$ , it follows from (4.7) that

$$\|\mathbf{A}\mathbf{V}_2\|_F = \left\| \begin{bmatrix} \mathbf{R}_{12} \\ \mathbf{R}_{22} \end{bmatrix} \right\|_F \leq c\sigma_{r+1}.$$

Thus,  $\mathbf{V}_2$  forms an orthogonal basis for the approximate null space of  $\mathbf{A}$ . The URV decomposition is particularly useful in applications such as subspace tracking in signal processing, where there is a need to compute an approximate null space and update this basis as rows are added or removed from  $\mathbf{A}$  (Björck, 2024).

The rank-revealing process begins with a pivoted QR decomposition (Theorem 3.3) and identifies a vector  $\mathbf{v}$  such that  $\|\mathbf{R}\mathbf{v}\|_2$  is small. Such a vector exists. For example,  $\mathbf{v} = \mathbf{v}_n$ , where  $\mathbf{v}_n$  is the right singular vector of  $\mathbf{R}$  corresponding to the smallest singular value  $\sigma_n$  and left singular vector  $\mathbf{u}_n$  such that  $\mathbf{R}\mathbf{v}_n = \sigma_n \mathbf{u}_n$  and  $\|\mathbf{R}\mathbf{v}_n\|_2 = \sigma_n$ . If  $\mathbf{A}$  or  $\mathbf{R}$  is rank-deficient,  $\sigma_n$  is small; see Sections 3.9 and 3.10. Next, a sequence of Givens rotations  $\mathbf{G}_{12}, \mathbf{G}_{23}, \dots, \mathbf{G}_{n-1,n}$  is determined such that

$$\mathbf{G}^\top \mathbf{v} = \mathbf{G}_{n-1,n}^\top \dots \mathbf{G}_{23}^\top \mathbf{G}_{12}^\top \mathbf{v} = \|\mathbf{v}\|_2 \mathbf{e}_n.$$

Then, an orthogonal matrix  $\mathbf{U}$  is computed such that  $\mathbf{U}^\top \mathbf{R}\mathbf{G} = \mathbf{U}^\top \mathbf{R}\mathbf{G}_{12} \dots \mathbf{G}_{n-1,n}$  is upper triangular. When applying  $\mathbf{G}_{i-1,i}$ , a nonzero element—known as a “bulge”—is introduced just below the diagonal of  $\mathbf{R}$ . To restore the triangular form, a left rotation is used to “chase the bulge.” These left rotations amount to the orthogonal matrix  $\mathbf{U}$ .

An example is shown below for a  $4 \times 4$  upper triangular matrix  $\mathbf{R}$ , where  $\boxtimes$  denotes an upper triangular entry of  $\mathbf{R}$ , boldface indicates a value that has just been modified,  $\blacksquare$  denotes a bulge value, and  $\mathbf{0}$  denotes the zero is introduced back during the process of chasing the bulge:

$$\begin{array}{cccc} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{G}_{12}} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \textcolor{brown}{\blacksquare} & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{U}_{12} \times} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & 0 & \boxtimes \end{bmatrix} \xrightarrow{\mathbf{G}_{23}} \begin{bmatrix} 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{brown}{\blacksquare} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{bmatrix} \\ \mathbf{R} & \mathbf{RG}_{12} & \mathbf{U}_{12}\mathbf{RG}_{12} & \mathbf{U}_{12}\mathbf{RG}_{12}\mathbf{G}_{23} \\ \xrightarrow{\mathbf{U}_{23} \times} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{G}_{34}} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{U}_{34} \times} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \mathbf{0} & \boxtimes \end{bmatrix} = \hat{\mathbf{R}}. \\ \mathbf{U}_{23}\mathbf{U}_{12}\mathbf{RG}_{12}\mathbf{G}_{23} & \mathbf{U}_{23}\mathbf{U}_{12}\mathbf{R} \dots \mathbf{G}_{34} & \mathbf{U}_{34} \dots \mathbf{R} \dots \mathbf{G}_{34} & \end{array}$$

This process of transforming  $\mathbf{R}$  to  $\hat{\mathbf{R}}$  requires  $\mathcal{O}(n^2)$  multiplications. We now have

$$\mathbf{U}^\top \mathbf{R}\mathbf{v} = (\underbrace{\mathbf{U}^\top \mathbf{R}\mathbf{G}}_{=\hat{\mathbf{R}}})(\mathbf{G}^\top \mathbf{v}) = \|\mathbf{v}\|_2 \hat{\mathbf{R}}\hat{\mathbf{e}}_n.$$

Since  $\mathbf{U}$  is orthogonal, it follows that if  $\|\mathbf{R}\mathbf{v}\|_2 < |r_{nn}|$ , then  $\|\hat{\mathbf{R}}\hat{\mathbf{e}}_n\|_2 < \gamma/\|\mathbf{v}\|_2$  for some  $\gamma$ . This bounds the norm for the last column of the transformed matrix  $\hat{\mathbf{R}}$ . If  $|r_{n-1,n-1}|$  is small, this process can be continued on the leading principal submatrix of order  $n-1$  of  $\hat{\mathbf{R}}$ .

**Appending a row.** Just as with the rank-one update of the Cholesky decomposition (Section 2.10) and the addition of a row to a QR decomposition (Section 3.18), we are often interested in efficiently updating solutions to least squares problems when new data arrive, particularly in online or streaming data settings; see Section 11.6 for related applications. In such cases, we may want to append a new row to the observed data matrix  $\mathbf{A}$  and compute the (rank-revealing) UTV decomposition along with its corresponding least squares solution (Theorem 11.17) in an efficient manner. For simplicity in notation, we denote the rank-revealing URV decomposition in (4.7) as

$$\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{R} & \mathbf{J} \\ \mathbf{0} & \mathbf{F} \end{bmatrix} \mathbf{V}^\top, \quad \mathbf{R} \in \mathbb{R}^{r \times r}, \quad (4.8)$$

where  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal, and  $\mathbf{R} \in \mathbb{R}^{r \times r}$  and  $\mathbf{F} \in \mathbb{R}^{(m-r) \times (n-r)}$  are upper triangular. Let  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$  be the singular values of  $\mathbf{A}$ , and assume that for some  $r < n$ , we have  $\sigma_r \gg \sigma_{r+1} \leq \delta$ , where  $\delta$  is a given tolerance. Then, the numerical  $\delta$ -rank of  $\mathbf{A}$  equals  $r$  (see Definition 11.7). Furthermore, if

$$\sigma_r(\mathbf{R}) \geq \frac{1}{c} \sigma_r, \quad (\|\mathbf{J}\|_F^2 + \|\mathbf{F}\|_F^2)^{1/2} \leq c\sigma_{r+1}$$

for some constant  $c$ , the decomposition (4.8) reveals the rank and null space of  $\mathbf{A}$ . The URV decomposition can be updated in  $\mathcal{O}(n^2)$  operations when a new row  $\mathbf{a}^\top$  is added to  $\mathbf{A}$ . To see this, we have

$$\begin{bmatrix} \mathbf{U}^\top & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^\top \end{bmatrix} \mathbf{V} = \begin{bmatrix} \mathbf{R} & \mathbf{J} \\ \mathbf{0} & \mathbf{F} \\ \mathbf{x}^\top & \mathbf{y}^\top \end{bmatrix}, \quad (4.9)$$

where  $\mathbf{a}^\top \mathbf{V} = [\mathbf{x}^\top, \mathbf{y}^\top]$  and  $(\|\mathbf{J}\|_F^2 + \|\mathbf{F}\|_F^2)^{1/2} = \nu \leq \delta$ . In the simplest case the inequality

$$\sqrt{\nu^2 + \|\mathbf{y}\|_2^2} \leq \delta \quad (4.10)$$

is satisfied. In this case, it suffices to reduce the matrix in (4.9) to upper triangular form using a sequence of left Givens rotations. Note that the updated matrix  $\mathbf{R}$  cannot become effectively rank-deficient because its singular values cannot decrease.

If (4.10) is not satisfied, we first reduce  $\mathbf{y}^\top$  in (4.9) so that it becomes proportional to  $\mathbf{e}_1^\top$ , while preserving the upper triangular structure of  $\mathbf{F}$ . This can be achieved by a sequence of (interleaved) right and left Givens rotations.

An example is shown below for a matrix  $\mathbf{A}$  with dimensions  $m - r = 3$  and  $n - r = 3$ . Note that here the  $j$ 's represent entire **columns** of  $\mathbf{J}$ ,  $f$  denotes an element of  $\mathbf{F}$ ,  $y$  denotes an element of  $\mathbf{y}$ , and **boldface** indicates the value has just been changed. Additionally,  $\boxtimes$  denotes a nonzero value, known as a bulge, introduced by the right Givens rotations, and  $\mathbf{0}$  denotes a zero value that is reintroduced (i.e., chasing the bulge).

**Step 1: Interleaved left and right Givens rotations.** We first consider the right-most part of  $\begin{bmatrix} \mathbf{U}^\top & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{a}^\top \end{bmatrix} \mathbf{V}$ , which is defined as  $\mathbf{B} = \begin{bmatrix} \mathbf{J} \\ \mathbf{F} \\ \mathbf{y}^\top \end{bmatrix}$ :

$$\begin{array}{ccccc} \begin{bmatrix} j & j & j \\ f & f & f \\ 0 & f & f \\ 0 & 0 & f \\ y & y & y \end{bmatrix} & \xrightarrow{\mathbf{H}_1} & \begin{bmatrix} j & j & j \\ f & \mathbf{f} & f \\ 0 & f & f \\ 0 & \boxtimes & f \\ y & y & 0 \end{bmatrix} & \xrightarrow{G_1 \times} & \begin{bmatrix} j & j & j \\ f & f & f \\ \mathbf{0} & \mathbf{f} & f \\ \mathbf{0} & \mathbf{0} & f \\ y & y & 0 \end{bmatrix} \\ \mathbf{B} & & \mathbf{B}\mathbf{H}_1 & & \mathbf{G}_1\mathbf{B}\mathbf{H}_1 \\ & & & & \mathbf{G}_1\mathbf{B}\mathbf{H}_2\mathbf{H}_2 \\ & & & & \mathbf{G}_2\mathbf{G}_1\mathbf{B}\mathbf{H}_2\mathbf{H}_2 \end{array}.$$

In this part of the reduction,  $\mathbf{R}$  and  $\mathbf{x}^\top$  are not involved.

**Step 2: Triangularization using Givens rotations.** At this point, the matrix has the form:

$$\begin{bmatrix} \mathbf{R} & \tilde{\mathbf{j}} & \tilde{\mathbf{H}} \\ \mathbf{0} & \tilde{\mathbf{f}} & \tilde{\mathbf{N}} \\ \mathbf{x}^\top & \sigma & \mathbf{0} \end{bmatrix}.$$

This matrix is then reduced to upper triangular form using Givens rotations from the left, and the rank  $r$  is increased by 1. This procedure is similar to the QR decomposition using Givens rotations; see Section 3.12 for more details. The entire update process requires  $\mathcal{O}(n^2)$  floating-point operations.

#### 4.5. Application: Row Rank equals Column Rank via UTV

The UTV framework enables the proof of a fundamental theorem in linear algebra: the equality of the row rank and column rank of a matrix; Theorem 0.15. It is worth noting that when using the UTV decomposition for this proof, a slight adjustment is needed in framing the existence claim. For instance, in Theorem 4.1, the initial assumption about the matrix  $\mathbf{A}$  is that its rank is  $r$ . However, since having rank  $r$  inherently implies the equality of row and column ranks, a more precise assertion in this context would be to state that  $\mathbf{A}$  has **column rank  $r$**  in Theorem 4.1. For further discussion, see Lu (2021b).

**Proof** [of Theorem 0.15, second approach] 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>3</sup>. Let  $\mathbf{D} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ . Clearly, the row rank and column rank of  $\mathbf{D}$  are equal. If we can show that the column rank and row rank of  $\mathbf{A}$  are equal to those of  $\mathbf{D}$ , the proof is complete.

Let  $\mathbf{U} = \mathbf{U}_0^\top$  and  $\mathbf{V} = \mathbf{V}_0^\top$ . Then,  $\mathbf{D} = \mathbf{U}\mathbf{A}\mathbf{V}$ . The proof can be broken into two parts: first prove that the row rank and column rank of  $\mathbf{A}$  are equal to those of  $\mathbf{U}\mathbf{A}$ ; then prove that the row rank and column rank of  $\mathbf{U}\mathbf{A}$  are equal to those of  $\mathbf{U}\mathbf{A}\mathbf{V}$ .

**Row and column ranks of  $\mathbf{A}$  equal those of  $\mathbf{U}\mathbf{A}$ .** Let  $\mathbf{B} = \mathbf{U}\mathbf{A}$ . Partition  $\mathbf{A}$  and  $\mathbf{B}$  by columns:  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  and  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n]$ , where  $\mathbf{b}_i = \mathbf{U}\mathbf{a}_i$  for all  $i$ . If a linear combination  $x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \dots + x_n\mathbf{a}_n = \mathbf{0}$ , then premultiplying by  $\mathbf{U}$  gives

$$\mathbf{U}(x_1\mathbf{a}_1 + x_2\mathbf{a}_2 + \dots + x_n\mathbf{a}_n) = x_1\mathbf{b}_1 + x_2\mathbf{b}_2 + \dots + x_n\mathbf{b}_n = \mathbf{0}.$$

Thus, any independent subset of columns in  $\mathbf{A}$  corresponds to an independent subset of columns in  $\mathbf{B}$ , implying:  $\dim(\mathcal{C}(\mathbf{B})) \leq \dim(\mathcal{C}(\mathbf{A}))$ . Similarly, since  $\mathbf{A} = \mathbf{U}^\top \mathbf{B}$ , we have  $\dim(\mathcal{C}(\mathbf{A})) \leq \dim(\mathcal{C}(\mathbf{B}))$ . This proves  $\dim(\mathcal{C}(\mathbf{B})) = \dim(\mathcal{C}(\mathbf{A}))$ . Applying the same reasoning to  $\mathbf{B}^\top$  and  $\mathbf{A}^\top$  shows that  $\dim(\mathcal{C}(\mathbf{B}^\top)) = \dim(\mathcal{C}(\mathbf{A}^\top))$ . This implies the row rank and column rank of  $\mathbf{A}$  and  $\mathbf{B} = \mathbf{U}\mathbf{A}$  are the same.

**Row and column ranks of  $\mathbf{U}\mathbf{A}$  equal those of  $\mathbf{U}\mathbf{A}\mathbf{V}$ .** Using the same reasoning as above, applied to  $\mathbf{U}\mathbf{A}$  and  $\mathbf{U}\mathbf{A}\mathbf{V}$ , we conclude that their row and column ranks are equal. This completes the proof. ■

<sup>3</sup>. Instead of using the ULV decomposition, some texts use elementary transformations  $\mathbf{E}_1$  and  $\mathbf{E}_2$ , 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.

## 🌊 Chapter 4 Problems 🌊

1. Use the UTV decompositions of  $\mathbf{A}$  and  $\mathbf{B}$  to derive the UTV decomposition of  $\text{diag}(\mathbf{A}, \mathbf{B})$ .
  2. Prove rigorously that the four subspaces described in Equation (4.2) can be spanned using the UTV decomposition.
  3. Let  $\mathbf{A} = \mathbf{U}\mathbf{B}\mathbf{V}$  be given, where  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal matrices. Prove that  $\sum_{i,j} |a_{ij}|^2 = \sum_{i,j} |b_{ij}|^2$ . Hint: Verify this by showing that  $\text{tr}(\mathbf{A}^\top \mathbf{A}) = \text{tr}(\mathbf{B}^\top \mathbf{B})$ .
  4. Explain how to compute the UTV decomposition using either Householder reflectors or Givens rotations. Provide insights into the advantages and limitations of each approach.
  5. Present a detailed and rigorous proof of the URV decomposition, as outlined in Theorem 4.2.
  6. Let  $\mathbf{A}, \mathbf{B} \in \mathbb{C}^{n \times n}$  be given, where  $\mathbf{B}$  is nonsingular. Prove that there exist unitary matrices  $\mathbf{U}, \mathbf{V} \in \mathbb{C}^{n \times n}$  such that  $\mathbf{A} = \mathbf{U}\mathbf{T}_A\mathbf{V}$  and  $\mathbf{B} = \mathbf{U}\mathbf{T}_B\mathbf{V}$ , where  $\mathbf{T}_A$  and  $\mathbf{T}_B$  are upper triangular matrices. Furthermore, show that the main diagonal entries of  $\mathbf{T}_B^{-1}\mathbf{T}_A$  are the eigenvalues  $\mathbf{B}^{-1}\mathbf{A}$ . When these eigenvalues are real, show that all the underlying matrices can be chosen to be real, and  $\mathbf{U}$  and  $\mathbf{V}$  are orthogonal. Hint: Use the Schur decomposition for  $\mathbf{B}^{-1}\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^*$  (Theorem 9.12) and the QR decomposition for  $\mathbf{B}\mathbf{U}$ .
  7. Prove that the computational complexity of computing the triangular factor  $\mathbf{R}$  in (4.5) requires  $2r^2(n - r)$  flops.
  8. **Read Section 11.1 first.** Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ . Show that <sup>4</sup>
    - $\mathbf{A}$  and  $\mathbf{B}$  are orthogonally left equivalent if and only if  $\mathbf{A}^\top \mathbf{A} = \mathbf{B}^\top \mathbf{B}$ .
    - $\mathbf{A}$  and  $\mathbf{B}$  are orthogonally right equivalent if and only if  $\mathbf{A}\mathbf{A}^\top = \mathbf{B}\mathbf{B}^\top$ .
    - $\mathbf{A}$  and  $\mathbf{B}$  are orthogonally biequivalent if and only if  $\mathbf{A}$  and  $\mathbf{B}$  have the same singular values with the same multiplicity.
  9. Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ . Show that
    - The matrices  $\mathbf{A}$  and  $\mathbf{B}$  are left equivalent if and only if  $\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{B})$ .
    - The matrices  $\mathbf{A}$  and  $\mathbf{B}$  are right equivalent if and only if  $\mathcal{C}(\mathbf{A}) = \mathcal{C}(\mathbf{B})$ .
    - The matrices  $\mathbf{A}$  and  $\mathbf{B}$  are biequivalent if and only if  $\text{rank}(\mathbf{A}) = \text{rank}(\mathbf{B})$ .
  10. Let  $\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}$  be a UTV decomposition of  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . Show that the pseudo-inverse of  $\mathbf{A}$  is  $\mathbf{A}^+ = \mathbf{V}^\top \begin{bmatrix} \mathbf{R}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{U}^\top$ .
  11. Following the procedure outlined in Section 4.4 for appending a row to a rank-revealing URV decomposition, write out the complete pseudo-code for the algorithm and prove that it requires  $\mathcal{O}(n^2)$  flops.
  12. Recover the URV and SVD decompositions using the complete orthogonal decomposition.
- 
- 4.** Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ . Then  $\mathbf{A}$  and  $\mathbf{B}$  are *left equivalent* (resp., *orthogonally left equivalent*) if there exists a nonsingular (resp., orthogonal)  $\mathbf{Z}_1 \in \mathbb{R}^{m \times m}$  such that  $\mathbf{A} = \mathbf{Z}_1\mathbf{B}$ ;  $\mathbf{A}$  and  $\mathbf{B}$  are *right equivalent* (resp., *orthogonally right equivalent*) if there exists a nonsingular (resp., orthogonal) matrix  $\mathbf{Z}_2 \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A} = \mathbf{B}\mathbf{Z}_2$ ;  $\mathbf{A}$  and  $\mathbf{B}$  are *biequivalent* (resp., *orthogonally biequivalent*) if there exist nonsingular (resp., orthogonal) matrices  $\mathbf{Z}_1 \in \mathbb{R}^{m \times m}$  and  $\mathbf{Z}_2 \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A} = \mathbf{Z}_1\mathbf{B}\mathbf{Z}_2$ .

# **Part III**

# **Data Interpretation and Information Distillation**



## CR Decomposition

### 5.1. CR Decomposition

The CR decomposition of a matrix, introduced in [Strang \(2021\)](#); [Strang and Moler \(2022\)](#), offers valuable insights into the matrix's rank and the relationships between its columns and rows. As is customary, we begin by presenting the result, deferring the discussion of its existence and derivation to later sections.

**Theorem 5.1: (CR decomposition)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be a matrix of rank  $r$ . Then it can be factored as

$$\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{C}} \underset{r \times n}{\mathbf{R}},$$

where  $\mathbf{C}$  consists of the first  $r$  linearly independent columns of  $\mathbf{A}$ , and  $\mathbf{R}$  is an  $r \times n$  matrix that reconstructs all the columns of  $\mathbf{A}$  from  $\mathbf{C}$ . Specifically,  $\mathbf{R}$  corresponds to the *reduced row echelon form (RREF)* of  $\mathbf{A}$ , with the zero rows removed.

The storage required for this decomposition changes from  $mn$  floating-point numbers to  $r(m + n)$  floating-point numbers, which may either reduce or, in some cases, increase memory usage depending on the matrix dimensions and rank.

The CR decomposition offers several key advantages: It highlights the independent columns and rows of the matrix, which are crucial for understanding its rank and the structure of its column and row spaces. For large matrices, the decomposition can approximate the matrix by retaining only the most significant columns and rows, reducing computational complexity. It simplifies the solution of linear systems  $\mathbf{Ax} = \mathbf{b}$  by transforming the problem into a more manageable form using the matrices  $\mathbf{C}$  and  $\mathbf{R}$ . Since both  $\mathbf{C}$  and  $\mathbf{R}$  have full rank, the decomposition provides an efficient way to compute the (Moore-Penrose) pseudo-inverse:  $\mathbf{A}^+ = \mathbf{R}^+ \mathbf{C}^+$ , where  $\mathbf{R}^+$  and  $\mathbf{C}^+$  denote the pseudo-inverses of  $\mathbf{R}$  and  $\mathbf{C}$ , respectively (see Problem 5.4). The CR decomposition is also useful for analyzing the incidence matrix of a graph. It helps in studying conserved quantities, such as current flow in electrical circuits, as described by Kirchhoff's Current Law ([Strang and Drucker, 2021](#)).

### 5.2. Existence of CR Decomposition

The CR decomposition of a matrix can be obtained through the following steps. Given that the matrix  $\mathbf{A}$  has rank  $r$ , it contains  $r$  linearly independent columns. These columns can be identified and collected in the matrix  $\mathbf{C}$  as follows:

- If column 1 of  $\mathbf{A}$  is nonzero, include it as a column of  $\mathbf{C}$ .
- If column 2 of  $\mathbf{A}$  is not a scalar multiple of column 1, include it as a column of  $\mathbf{C}$ .
- If column 3 of  $\mathbf{A}$  is not a linear combination of columns 1 and 2, include it as a column of  $\mathbf{C}$ .
- Continue this process until  $r$  linearly independent columns have been selected. If  $r$  is not known in advance, continue until all linearly independent columns have been identified.

Once  $r$  linearly independent columns are extracted from  $\mathbf{A}$ , the CR decomposition can be constructed by interpreting matrix multiplication in terms of the column space. The product of two matrices,  $\mathbf{D} \in \mathbb{R}^{m \times k}$  and  $\mathbf{E} \in \mathbb{R}^{k \times n}$ , results in the matrix  $\mathbf{A} = \mathbf{DE}$ . This can be expressed as  $\mathbf{A} = \mathbf{D}[\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_n] = [\mathbf{De}_1, \mathbf{De}_2, \dots, \mathbf{De}_n]$ . In this interpretation, each column of  $\mathbf{A}$  is a linear combination of the columns of  $\mathbf{D}$ .

**Proof** [of Theorem 5.1] Since  $\mathbf{A}$  has rank  $r$  and  $\mathbf{C}$  is constructed from  $r$  linearly independent columns of  $\mathbf{A}$ , the column space of  $\mathbf{C}$  is the same as that of  $\mathbf{A}$ . Therefore, any other column  $\mathbf{a}_i$  of  $\mathbf{A}$  can be represented as a linear combination of the columns of  $\mathbf{C}$ , i.e., there exists a vector  $\mathbf{r}_i$  such that  $\mathbf{a}_i = \mathbf{Cr}_i, \forall i \in \{1, 2, \dots, n\}$ . By arranging these vectors  $\mathbf{r}_i$ 's as the columns of a matrix  $\mathbf{R}$ , we obtain

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] = [\mathbf{Cr}_1, \mathbf{Cr}_2, \dots, \mathbf{Cr}_n] = \mathbf{CR}.$$

Thus, the decomposition  $\mathbf{A} = \mathbf{CR}$  is established, completing the proof. ■

### 5.3. Reduced Row Echelon Form (RREF)

In Section 1.2 on Gaussian elimination, we introduced the elimination matrix (a lower triangular matrix; see (1.1)) and the permutation matrix to facilitate transforming  $\mathbf{A}$  into an upper triangular form. Let us now revisit the Gaussian elimination process for a  $4 \times 4$  square matrix, where  $\boxtimes$  denotes a value that is not necessarily zero, and **boldface** indicates the value has just been changed:

$$\begin{array}{cccc} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{E}_1} & \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{P}_1} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \boxtimes \\ \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{E}_2} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} \end{array} \right]. \\ \mathbf{A} & & \mathbf{E}_1\mathbf{A} & & \mathbf{P}_1\mathbf{E}_1\mathbf{A} & & \mathbf{E}_2\mathbf{P}_1\mathbf{E}_1\mathbf{A} \end{array}$$

Moreover, Gaussian elimination can also be applied to rectangular matrices. Below, we demonstrate the process for a  $4 \times 5$  matrix:

$$\begin{array}{ccc} \left[ \begin{array}{ccccc} \mathbf{2} & \boxtimes & 10 & 9 & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{E}_1} & \left[ \begin{array}{ccccc} \mathbf{2} & \boxtimes & 10 & 9 & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{5} & \mathbf{6} & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{2} & \mathbf{\boxtimes} & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{E}_2} \left[ \begin{array}{ccccc} \mathbf{2} & \boxtimes & 10 & 9 & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{5} & \mathbf{6} & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{3} & \boxtimes \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array} \right], \\ \mathbf{A} & & \mathbf{E}_1\mathbf{A} & & \mathbf{E}_2\mathbf{E}_1\mathbf{A} \end{array}$$

where the numbers highlighted in blue are *pivots*, as previously defined (Definition 1.6)<sup>1</sup>. The resulting matrix is referred to as the *row echelon form* of  $\mathbf{A}$ . In this example, the fourth row becomes a zero row. To continue, we perform additional row operations to ensure that

1. In the context of Gaussian elimination, a pivot element is the first nonzero element in a row when performing row operations to transform a matrix into its row echelon form or reduced row echelon form.

all entries above the pivots are zero:

$$\begin{array}{c} \left[ \begin{array}{ccccc} 2 & \boxtimes & 10 & 9 & \boxtimes \\ 0 & 0 & 5 & 6 & \boxtimes \\ 0 & 0 & 0 & 3 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{\mathbf{E}_3} \left[ \begin{array}{ccccc} 2 & \boxtimes & 0 & -3 & \boxtimes \\ 0 & 0 & 5 & 6 & \boxtimes \\ 0 & 0 & 0 & 3 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{\mathbf{E}_4} \left[ \begin{array}{ccccc} 2 & \boxtimes & 0 & 0 & \boxtimes \\ 0 & 0 & 5 & 0 & \boxtimes \\ 0 & 0 & 0 & 3 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right], \\ \mathbf{E}_2\mathbf{E}_1\mathbf{A} \qquad \qquad \qquad \mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A} \qquad \qquad \qquad \mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A} \end{array}$$

where  $\mathbf{E}_3$  subtracts twice the second row from the first row, while  $\mathbf{E}_4$  adds the third row to the first row and subtracts twice the third row from the second row. To achieve the full *reduced row echelon form (RREF)*, we need to ensure that all pivots are set to 1:

$$\begin{array}{c} \left[ \begin{array}{ccccc} 2 & \boxtimes & 0 & 0 & \boxtimes \\ 0 & 0 & 5 & 0 & \boxtimes \\ 0 & 0 & 0 & 3 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right] \xrightarrow{\mathbf{E}_5} \left[ \begin{array}{ccccc} 1 & \boxtimes & 0 & 0 & \boxtimes \\ 0 & 0 & 1 & 0 & \boxtimes \\ 0 & 0 & 0 & 1 & \boxtimes \\ 0 & 0 & 0 & 0 & 0 \end{array} \right], \\ \mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A} \qquad \qquad \qquad \mathbf{E}_5\mathbf{E}_4\mathbf{E}_3\mathbf{E}_2\mathbf{E}_1\mathbf{A} \end{array} \tag{5.1}$$

where  $\mathbf{E}_5$  scales the pivots so that they equal 1. Unlike the transformation matrices used in LU decomposition, which are generally lower triangular, the transformation matrices  $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_5$  may also include permutation matrices or other types of matrices. The resulting matrix is the *reduced row echelon form (RREF)* of  $\mathbf{A}$ , characterized by having pivots equal to 1 and zeros above the pivots.

For a general matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , let its row echelon form (with zeros above the pivots) be denoted as  $\mathbf{F}$ . To transform  $\mathbf{F}$  into the RREF of  $\mathbf{A}$ , we apply a sequence of transformations  $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_r$ , defined as:

$$\mathbf{E}_i = \mathbf{E}_{i;\eta} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & & \eta & \\ & & & & 1 \\ & & & & & \ddots \\ & & & & & & 1 \end{bmatrix} = \mathbf{I} + (\eta - 1)\mathbf{e}_i\mathbf{e}_i^\top, \quad \text{with } 1 \leq i \leq r,$$

where  $\mathbf{e}_i$  is the  $i$ -th unit basis vector,  $r$  is the rank of  $\mathbf{A}$  (i.e., the number of nonzero rows in  $\mathbf{F}$ ), and  $\eta$  is the inverse of the  $i$ -th pivot in  $\mathbf{F}$ . Each  $\mathbf{E}_i$  is invertible, with  $\mathbf{E}_{i;\eta}^{-1} = \mathbf{E}_{i;\eta^{-1}} = \mathbf{I} + (\frac{1}{\eta} - 1)\mathbf{e}_i\mathbf{e}_i^\top$ . The reduced row echelon form of  $\mathbf{A}$  can then be obtained by  $(\mathbf{E}_r\mathbf{E}_{r-1}\dots\mathbf{E}_1\mathbf{F})$ .

We formally define the reduced row echelon form (RREF) of a matrix as follows:

**Definition 5.2 (Reduced row echelon form, RREF).** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . The *row echelon form (REF)* of  $\mathbf{A}$  satisfies the following conditions:

1. The leading nonzero entry (called a pivot, as defined in Definition 1.6) of the  $(i+1)$ -th row appears to the right of the leading nonzero entry of the  $i$ -th row.
2. All entries below a pivot in a given column are zeros.
3. Any row that contains only zeros is positioned at the bottom of the matrix.

Although Property 2 is a consequence of Property 1, we include it here for emphasis. If a matrix in row echelon form satisfies the following additional conditions, it is said to be in *reduced row echelon form (RREF)*:

4. The leading nonzero entry (pivot) in each row is equal to 1.
5. All entries above each pivot are zeros.

**Exercise 5.3.** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be any matrix. Show that there exists a sequence of row transformations  $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_k$  such that  $\mathbf{B} = \mathbf{E}_k \mathbf{E}_{k-1} \dots \mathbf{E}_1 \mathbf{A}$  is in reduced row echelon form. Hint: Use induction.

The pivots (in its RREF) play a key role in estimating the rank of a matrix.

**Lemma 5.4: (Rank and pivots)** The rank of a matrix  $\mathbf{A}$  is equal to the number of pivots (in its reduced row echelon form). Consequently, the rank of a matrix is the same as the rank of its RREF.

**Proof** [of Lemma 5.4] According to Proposition 1.7, the row space of  $\mathbf{A}$  is identical to the row space of its RREF. Since the rank of the RREF is defined as the number of its pivots, it follows that the rank of  $\mathbf{A}$  is also equal to the number of pivots. ■

We now show that the RREF and the CR decomposition are closely related.

**Lemma 5.5: (RREF in CR)** The reduced row echelon form of the matrix  $\mathbf{A}$ , excluding zero rows, corresponds to the matrix  $\mathbf{R}$  in the CR decomposition.

**Proof** [Informal proof of Lemma 5.5] Informally, using the example provided earlier in (5.1), we express the matrix  $\mathbf{A}$  as:

$$\mathbf{E}_5 \mathbf{E}_4 \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} = \mathbf{R}_0 \quad \rightarrow \quad \mathbf{A} = (\mathbf{E}_5 \mathbf{E}_4 \mathbf{E}_3 \mathbf{E}_2 \mathbf{E}_1)^{-1} \mathbf{R}_0.$$

We observe that columns 1, 3, and 4 of  $\mathbf{R}_0$  each contain a single nonzero entry, which is equal to 1. This observation allows us to construct a matrix  $\mathbf{C}$  (identical to the “column matrix” in the CR decomposition) whose first three columns are equal to columns 1, 3, and 4 of  $\mathbf{A}$ , i.e.,  $\mathbf{C} = [\mathbf{a}_1, \mathbf{a}_3, \mathbf{a}_4]$ . Additionally, because the last row of  $\mathbf{R}_0$  consists entirely of zeros, the last row of  $\mathbf{R}_0$  can be safely disregarded in computations. Notably, this matrix  $\mathbf{C}$  is unique in its ability to reconstruct columns 1, 3, and 4 of  $\mathbf{A}$ , as the pivots of  $\mathbf{R}_0$  are all equal to 1. Thus, we obtain the CR decomposition:  $\mathbf{A} = \mathbf{C}\mathbf{R}$ . ■

Nest, we present a rigorous proof of the uniqueness of the RREF of a matrix.

**Theorem 5.6: (Uniqueness of RREF)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be any matrix of rank  $r$ . Suppose  $\mathbf{X}$  and  $\mathbf{Y}$  are two reduced row echelon forms of  $\mathbf{A}$ , obtained by applying two sequences of elementary row operations  $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_p$  and  $\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_q$ , respectively, where

$$\mathbf{X} = \mathbf{E}_p \dots \mathbf{E}_2 \mathbf{E}_1 \mathbf{A} = \mathbf{EA} \quad \text{and} \quad \mathbf{Y} = \mathbf{F}_q \dots \mathbf{F}_2 \mathbf{F}_1 \mathbf{A} = \mathbf{FA}.$$

Then, the two reduced row echelon forms are identical, i.e.,  $\mathbf{X} = \mathbf{Y}$  and  $\mathbf{E} = \mathbf{F}$ .

**Proof** [of Theorem 5.6] Let  $\mathbf{B} = \mathbf{EF}^{-1} = \mathbf{E}_p \dots \mathbf{E}_2 \mathbf{E}_1 \mathbf{F}_1^{-1} \mathbf{F}_2^{-1} \dots \mathbf{F}_q^{-1}$ . Then we have  $\mathbf{X} = \mathbf{BY}$  and  $\mathbf{Y} = \mathbf{B}^{-1}\mathbf{X}$ . The  $i$ -th column of  $\mathbf{X}$  and  $\mathbf{Y}$  can be expressed as  $\mathbf{x}_i = \mathbf{Xe}_i$  and  $\mathbf{y}_i = \mathbf{Ye}_i$ , respectively, where  $\mathbf{e}_i$  represents the  $i$ -th standard basis vector in  $\mathbb{R}^n$ .

**Zero columns match.** If  $\mathbf{x}_i = \mathbf{0}$ , then  $\mathbf{y}_i = \mathbf{B}^{-1}\mathbf{x}_i = \mathbf{0}$ . Similarly, if  $\mathbf{y}_i = \mathbf{0}$ , then  $\mathbf{x}_i = \mathbf{By}_i = \mathbf{0}$ . Thus, the zero columns in  $\mathbf{X}$  and  $\mathbf{Y}$  are aligned. Without loss of generality, we assume that  $\mathbf{X}$  and  $\mathbf{Y}$  contain no zero columns for the rest of the analysis.

**First column.** Since we assume  $\mathbf{X}$  and  $\mathbf{Y}$  do not contain zero columns, their first columns must be  $\mathbf{x}_1 = \mathbf{y}_1 = \mathbf{e}_1$ . This also implies the first column of  $\mathbf{B}$  is  $\mathbf{e}_1$ . We refer to columns in  $\mathbf{X}$  or  $\mathbf{Y}$  that contain pivots as *pivot columns*, and those do not as *non-pivot columns*.

**Non-pivot columns between the first and second pivot columns.** Suppose the indices of the pivot columns in  $\mathbf{X}$  are  $\{i_1, i_2, \dots, i_r\}$ , and the indices of the pivot columns in  $\mathbf{Y}$  are  $\{j_1, j_2, \dots, j_r\}$ . According to Lemma 5.4, there are  $r$  pivot columns. And we have already shown that  $i_1 = j_1 = 1$ . Then for  $k \in \{2, 3, \dots, j_2 - 1\}$ , we suppose  $\mathbf{y}_k = \lambda \mathbf{e}_1$  for some nonzero  $\lambda$ . We have

$$\mathbf{x}_k = \mathbf{B}\mathbf{y}_k = \mathbf{B}\lambda\mathbf{e}_1 = \lambda\mathbf{b}_1 = \lambda\mathbf{e}_1 \implies \mathbf{x}_k = \mathbf{y}_k, \quad \forall k \in \{2, 3, \dots, j_2 - 1\}.$$

Conversely, suppose  $\mathbf{x}_k = \lambda\mathbf{e}_1$  for some nonzero  $\lambda$  with  $k \in \{2, 3, \dots, i_2 - 1\}$ . In this case, we also find that  $\mathbf{x}_k = \mathbf{y}_k$ . This implies the non-pivot columns  $k = 2, 3, \dots, j_2 - 1$  of  $\mathbf{X}$  and  $\mathbf{Y}$  are the same, and moreover  $i_2 = j_2$  (the indices of the second pivot columns in  $\mathbf{X}$  and  $\mathbf{Y}$  are the same, and  $\mathbf{x}_{j_2} = \mathbf{y}_{j_2} = \mathbf{e}_2$ ).

To conclude, we have demonstrated that the first  $j_2$  columns of  $\mathbf{X}$  and  $\mathbf{Y}$  are identical.

**Non-pivot columns between the second and third pivot columns.** Since  $\mathbf{x}_{j_2} = \mathbf{y}_{j_2} = \mathbf{e}_2$ , we have  $\mathbf{x}_{j_2} = \mathbf{B}\mathbf{y}_{j_2} = \mathbf{B}\mathbf{e}_2 = \mathbf{b}_2 = \mathbf{e}_2$ , i.e., the second column of  $\mathbf{B}$  is  $\mathbf{e}_2$ . On the other hand, considering the non-pivot columns of  $k \in \{j_2 + 1, j_2 + 2, \dots, j_3 - 1\}$  in  $\mathbf{Y}$ , we assume  $\mathbf{y}_k = \lambda_1 \mathbf{e}_1 + \lambda_2 \mathbf{e}_2$ . Then we have

$$\begin{aligned} \mathbf{x}_k = \mathbf{B}\mathbf{y}_k &= \mathbf{B}(\lambda_1 \mathbf{e}_1 + \lambda_2 \mathbf{e}_2) = \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 = \lambda_1 \mathbf{e}_1 + \lambda_2 \mathbf{e}_2 \\ &\implies \mathbf{x}_k = \mathbf{y}_k, \quad \forall k \in \{j_2 + 1, j_2 + 2, \dots, j_3 - 1\}. \end{aligned}$$

Conversely, suppose  $\mathbf{x}_k = \lambda_1 \mathbf{e}_1 + \lambda_2 \mathbf{e}_2$  with  $k \in \{j_2 + 1, j_2 + 2, \dots, i_3 - 1\}$ , we also have  $\mathbf{x}_k = \mathbf{y}_k$ . Consequently, the non-pivot columns  $k = j_2 + 1, j_2 + 2, \dots, j_3 - 1$  of  $\mathbf{X}$  and  $\mathbf{Y}$  are the same, and also  $j_3 = i_3$  (the indices of the third pivot columns in  $\mathbf{X}$  and  $\mathbf{Y}$  are the same, and  $\mathbf{x}_{j_3} = \mathbf{y}_{j_3} = \mathbf{e}_3$ ).

By repeating this argument for all pivot positions, we show that all corresponding columns of  $\mathbf{X}$  and  $\mathbf{Y}$  are equal, completing the proof.  $\blacksquare$

**Exercise 5.7 (Determinant of RREF).** Show that the determinant of a matrix  $\mathbf{A}$  is nonzero if and only if its RREF is the identity matrix  $\mathbf{I}$ .

Using the above result and the multiplicative property of determinants (i.e.,  $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$ ), the determinant of  $\mathbf{A}$  can be determined by tracking the elementary row operations performed during the process of transforming  $\mathbf{A}$  into its RREF.

In summary, we begin by calculating the reduced row echelon form of matrix  $\mathbf{A}$ , denoted as  $rref(\mathbf{A})$ . Then, in the CR decomposition, the matrix  $\mathbf{C}$  is formed by selecting from  $\mathbf{A}$  only those columns that correspond to pivot columns in  $rref(\mathbf{A})$ . Simultaneously, the factor  $\mathbf{R}$  is obtained by removing all zero rows from  $rref(\mathbf{A})$ . This process represents a special case of rank decomposition (Theorem 5.8), but it is notable because it explicitly involves the RREF. Hence, we introduce it here due to its specific relevance.

An important property of  $\mathbf{R}$  is that a subset of its  $r$  columns, each containing a pivot, together form an  $r \times r$  identity matrix. It's worth reiterating that we can obtain this matrix  $\mathbf{R}$  simply by eliminating the zero rows from the RREF. As noted in Strang (2021), a notation for the RREF that retains the zero rows is denoted by  $\mathbf{R}_0$ :

$$\mathbf{R}_0 = rref(\mathbf{A}) = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_r & \mathbf{F} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{P},$$

where the  $n \times n$  permutation matrix  $\mathbf{P}$  arranges the columns of the  $r \times r$  identity matrix  $\mathbf{I}_r$  into their correct positions, aligning them with the first  $r$  linearly independent columns of the original matrix  $\mathbf{A}$ .

## 5.4. Rank Decomposition

We previously noted that the CR decomposition is a special case of rank decomposition. We formally prove that such a decomposition exists for any matrix.

**Theorem 5.8: (Rank decomposition)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be any matrix of rank  $r$ . Then  $\mathbf{A}$  can be factored into what is known as the *rank decomposition* as follows:

$$\mathbf{A}_{m \times n} = \mathbf{D}_{m \times r} \mathbf{F}_{r \times n},$$

where  $\mathbf{D} \in \mathbb{R}^{m \times r}$  and  $\mathbf{F} \in \mathbb{R}^{r \times n}$  both have (full) rank  $r$ .

**Proof** [of Theorem 5.8] From the ULV decomposition in Theorem 4.1, we can express  $\mathbf{A}$  as  $\mathbf{A} = \mathbf{U} \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}$ . Let  $\mathbf{U}_0 = \mathbf{U}_{:,1:r}$  and  $\mathbf{V}_0 = \mathbf{V}_{1:r,:}$ , i.e.,  $\mathbf{U}_0$  comprises the first  $r$  columns of  $\mathbf{U}$ , and  $\mathbf{V}_0$  consists of the first  $r$  rows of  $\mathbf{V}$ . Thus,  $\mathbf{A}$  can also be written as:  $\mathbf{A} = \mathbf{U}_0 \mathbf{L} \mathbf{V}_0$ , where  $\mathbf{U}_0 \in \mathbb{R}^{m \times r}$  and  $\mathbf{V}_0 \in \mathbb{R}^{r \times n}$ . This is also referred to as the reduced ULV decomposition. 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 obtain a valid rank decomposition of  $\mathbf{A}$ . ■

The rank decomposition is **not unique**. In fact, using elementary row and column operations, we can also write:  $\mathbf{A} = \mathbf{E}_1 \begin{bmatrix} \mathbf{Z} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{E}_2$ , where  $\mathbf{E}_1 \in \mathbb{R}^{m \times m}$  and  $\mathbf{E}_2 \in \mathbb{R}^{n \times n}$  are products of nonsingular elementary row and column operations, and  $\mathbf{Z} \in \mathbb{R}^{r \times r}$ . There exist many possible choices for  $\mathbf{E}_1$ ,  $\mathbf{E}_2$ , and  $\mathbf{Z}$ . When  $\mathbf{Z} = \mathbf{I}_r$ , where  $r$  is the rank of  $\mathbf{A}$ , this decomposition is known as the *Smith decomposition or Smith form* of  $\mathbf{A}$  (Bernstein, 2009). By using similar constructions as in the proof above, we can derive alternative rank decompositions from other matrix factorizations, such as SVD, URV, CR, and CUR. However, we can also establish a general relationship between different rank decompositions using the following corollary.

**Corollary 5.9: (Connection between rank decompositions)** Let  $\mathbf{A} = \mathbf{D}_1 \mathbf{F}_1 = \mathbf{D}_2 \mathbf{F}_2 \in \mathbb{R}^{m \times n}$  be two rank decompositions of  $\mathbf{A}$ . Then there exists a nonsingular matrix  $\mathbf{P}$  such that

$$\mathbf{D}_1 = \mathbf{D}_2 \mathbf{P} \quad \text{and} \quad \mathbf{F}_1 = \mathbf{P}^{-1} \mathbf{F}_2.$$

More generally, given  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{A}$  and  $\mathbf{B}$  are biequivalent <sup>a</sup> if and only if  $\mathbf{A}$  and  $\mathbf{B}$  share the same Smith form.

a.  $\mathbf{A}$  and  $\mathbf{B}$  are *biequivalent* if there exist nonsingular  $\mathbf{Z}_1 \in \mathbb{R}^{m \times m}$  and  $\mathbf{Z}_2 \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A} = \mathbf{Z}_1 \mathbf{B} \mathbf{Z}_2$ .

**Proof** [of Corollary 5.9] Given  $\mathbf{D}_1 \mathbf{F}_1 = \mathbf{D}_2 \mathbf{F}_2$ , postmultiplying by  $\mathbf{F}_1^\top$  yields  $\mathbf{D}_1 \mathbf{F}_1 \mathbf{F}_1^\top = \mathbf{D}_2 \mathbf{F}_2 \mathbf{F}_1^\top$ . Since  $\text{rank}(\mathbf{F}_1 \mathbf{F}_1^\top) = \text{rank}(\mathbf{F}_1) = r$ ,  $\mathbf{F}_1 \mathbf{F}_1^\top$  is a square matrix with full rank, hence nonsingular. Therefore, we have  $\mathbf{D}_1 = \mathbf{D}_2 \mathbf{F}_2 \mathbf{F}_1^\top (\mathbf{F}_1 \mathbf{F}_1^\top)^{-1}$ . 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  $\mathbf{F}_1 = \mathbf{P}^{-1} \mathbf{F}_2$ . The second part of the corollary can be proven similarly. ■

## 5.5. Application: Idempotent Matrix and Matrix Rank

The CR decomposition or rank decomposition plays a pivotal role in proving several essential theorems in linear algebra. For instance, it is instrumental in establishing the existence of the pseudo-inverse; it helps determine a basis for the four fundamental subspaces in linear algebra (Lu, 2021c).

Moreover, the CR factorization finds practical applications in data analysis and computational problem-solving. For instance, it proves valuable in solving least squares problems by reducing the system to a minimal set of variables, thereby eliminating redundancy.

The CR decomposition is also a powerful tool for analyzing the rank characteristics of idempotent matrices. Its utility in orthogonal projections is further explored in the Appendix of Lu (2021c).

**Proposition 5.10: (Rank and trace of an idempotent matrix)** Let  $\mathbf{A}$  be an idempotent matrix (i.e.,  $\mathbf{A}^2 = \mathbf{A}$ ). Then the rank of  $\mathbf{A}$  is equal to its trace.

**Proof** [of Proposition 5.10] Consider an  $n \times n$  idempotent matrix  $\mathbf{A}$  of rank  $r$ . By the CR decomposition, we can express  $\mathbf{A}$  as  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , where  $\mathbf{C} \in \mathbb{R}^{n \times r}$  and  $\mathbf{R} \in \mathbb{R}^{r \times n}$  are both of full rank  $r$ . Therefore,

$$\mathbf{A}^2 = \mathbf{A} \implies \mathbf{C}\mathbf{R}\mathbf{C}\mathbf{R} = \mathbf{C}\mathbf{R} \implies \mathbf{R}\mathbf{C}\mathbf{R} = \mathbf{R} \implies \mathbf{R}\mathbf{C} = \mathbf{I}_r,$$

where  $\mathbf{I}_r$  denotes the  $r \times r$  identity matrix. Consequently,  $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{C}\mathbf{R}) = \text{tr}(\mathbf{R}\mathbf{C}) = \text{tr}(\mathbf{I}_r) = r$ , which corresponds to the rank of  $\mathbf{A}$ . This equality holds due to the invariance of the trace under cyclic permutations. ■

On the other hand, we previously established a fundamental theorem in linear algebra using the UTV framework, which demonstrated that the row rank and column rank of any matrix are equal (Theorem 0.15). The CR decomposition offers an alternative explanation of this result.

**Proof** [of Theorem 0.15, the third way] Consider the CR decomposition of  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , where  $\mathbf{R}$  can be expressed as  $\mathbf{R} = [\mathbf{I}_r, \mathbf{F}]\mathbf{P}$ , and  $\mathbf{P}$  is an  $n \times n$  permutation matrix used to arrange the columns of the  $r \times r$  identity matrix  $\mathbf{I}_r$  in their appropriate positions. It is straightforward to verify that the  $r$  rows of  $\mathbf{R}$  are linearly independent due to the nonsingular submatrix  $\mathbf{I}_r$ . Therefore, the row rank of  $\mathbf{R}$  is  $r$ .

First, by the definition of the CR decomposition, the  $r$  columns of  $\mathbf{C}$  are selected from  $r$  linearly independent columns of  $\mathbf{A}$ , and the column rank of  $\mathbf{A}$  is  $r$ . Furthermore,

- Since  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , every row of  $\mathbf{A}$  can be represented as a linear combinations of the rows of  $\mathbf{R}$ . Hence, the row space of  $\mathbf{A}$  is contained within the row space of  $\mathbf{R}$ :  $\mathcal{C}(\mathbf{A}^\top) \subseteq \mathcal{C}(\mathbf{R}^\top)$ .
- From  $\mathbf{A} = \mathbf{C}\mathbf{R}$ , we also have  $(\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{C}\mathbf{R} = (\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{A}$ , simplifying to  $\mathbf{R} = (\mathbf{C}^\top\mathbf{C})^{-1}\mathbf{C}^\top\mathbf{A}$  (Because  $\mathbf{C}$  has full column rank  $r$ ,  $\mathbf{C}^\top\mathbf{C}$  is nonsingular). Consequently, the rows of  $\mathbf{R}$  are linear combinations of the rows of  $\mathbf{A}$ , meaning the row space of  $\mathbf{R}$  is contained within the row space of  $\mathbf{A}$ :  $\mathcal{C}(\mathbf{R}^\top) \subseteq \mathcal{C}(\mathbf{A}^\top)$ .

By this “sandwich” argument, the row spaces of  $\mathbf{A}$  and  $\mathbf{R}$  are equal, and thus their row ranks are equal:  $\mathcal{C}(\mathbf{A}^\top) = \mathcal{C}(\mathbf{R}^\top)$ .

Since the column rank of  $\mathbf{A}$  is also  $r$  by the definition of the CR decomposition, it follows that both the row rank and column rank of  $\mathbf{A}$  are equal to  $r$ . ■

## Chapter 5 Problems

1. Discuss what rank number  $r$  in Theorem 5.1 and Theorem 5.8 can reduce storage requirements.
2. Determine the reduced row echelon form and the CR decomposition for the matrix  $\mathbf{A} = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 7 & 6 \\ 4 & 5 & 8 \end{bmatrix}$ .
3. Apply the RREF process to the matrix  $\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 4 & 2 & 3 \\ 1 & 1 & 2 & -1 \\ -3 & -1 & 4 & 0 \end{bmatrix}$ .
4. Find the pseudo-inverse of a matrix  $\mathbf{A}$  using its CR decomposition.
5. Show that the solution of the linear system  $\mathbf{Ax} = \mathbf{b}$  remains unchanged if the same sequence of elementary row transformations is applied to both  $\mathbf{A}$  and  $\mathbf{b}$ . Consequently, the solution can be revealed by finding the RREF of the augmented matrix  $[\mathbf{A}, \mathbf{b}]$ .
6. Following Problem 5.5, show that the two linear systems  $\mathbf{A}_1\mathbf{x} = \mathbf{b}_1$  and  $\mathbf{A}_2\mathbf{x} = \mathbf{b}_2$  have the same set of solutions if and only if  $[\mathbf{A}_1, \mathbf{b}_1]$  and  $[\mathbf{A}_2, \mathbf{b}_2]$  have the same RREF.
7. Show that if a system of linear equations has two distinct solutions, then it must have infinitely many solutions.
8. Show that if a linear system  $\mathbf{Ax} = \mathbf{b}$  has more than one solution, then the corresponding homogeneous system  $\mathbf{Ax} = \mathbf{0}$  also has nontrivial solutions.
9. A system of linear equations with fewer equations than unknowns is sometimes referred to as an *underdetermined* system. Provide an example of an inconsistent underdetermined system of two equations in three unknowns (If there is at least one solution, the linear system is called *consistent*; otherwise, it is called *inconsistent*).
10. Suppose an underdetermined system is consistent. Explain why such a system must have an infinite number of solutions.
11. A system of linear equations with more equations than unknowns is sometimes referred to as an *overdetermined* system. Discuss the conditions under which such a system can be consistent.
12. Two matrices are called *row equivalent* if there is a sequence of elementary row operations that transforms one matrix into the other. Show that if matrices  $\mathbf{A}$  and  $\mathbf{B}$  are row equivalent, they have the same RREF.
13. Let  $\mathbf{Ax} = \mathbf{b}$  be a consistent system where  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . Show that  $\mathbf{A}$  has  $m$  pivot columns. Furthermore, let  $m = n$ ; show that the RREF of  $\mathbf{A}$  is the identity matrix.
14. Let  $\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$  be any  $2 \times 2$  nonsingular matrix. Show that there exists a nonsingular matrix  $\mathbf{S}$  such that  $\mathbf{SA} = \begin{bmatrix} 1 & 0 \\ 0 & ad - bc \end{bmatrix}$ , where  $\mathbf{S}$  is the product of at most four elementary matrices of the form  $\mathbf{E}_{i,j;\alpha} = \mathbf{I} + \alpha e_i e_j^\top \in \mathbb{R}^{n \times n}$ .
15. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be nonsingular. Show that there is a matrix  $\mathbf{S}$  such that  $\mathbf{SA} = \begin{bmatrix} \mathbf{I}_{n-1} & 0 \\ 0 & d \end{bmatrix}$ , where  $d = \det(\mathbf{A})$ , and  $\mathbf{S}$  is again a product of elementary matrices of the form  $\mathbf{E}_{i,j;\alpha} = \mathbf{I} + \alpha e_i e_j^\top \in \mathbb{R}^{n \times n}$ .

# 6

## Skeleton/CUR and Interpolative Decomposition

### 6.1. Skeleton/CUR Decomposition

The CR decomposition utilizes actual columns of a matrix, whereas the *skeleton decomposition* extends this concept by incorporating both actual columns and rows.

**Theorem 6.1: (Skeleton/CUR decomposition)** Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be decomposed as

$$\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{C}} \underset{r \times r}{\mathbf{U}^{-1}} \underset{r \times n}{\mathbf{R}},$$

where  $\mathbf{C}$  contains some  $r$  linearly independent columns of  $\mathbf{A}$ ,  $\mathbf{R}$  contains some  $r$  linearly independent rows of  $\mathbf{A}$ , and  $\mathbf{U}$  is the nonsingular submatrix formed by the intersection of these selected rows and columns.

- The storage requirement for this decomposition may be reduced (or potentially increased) compared to storing the full matrix, from  $mn$  floating-point numbers to  $r(m + n) + r^2$  floating-point numbers.
- Alternatively, if we only record the indices of the selected rows and columns, it requires  $mr$  and  $nr$  floating-point numbers for storing  $\mathbf{C}$  and  $\mathbf{R}$ , respectively. Additionally,  $2r$  integers are required to store the positions of the selected columns in  $\mathbf{C}$  and rows in  $\mathbf{R}$  within  $\mathbf{A}$ , which allows reconstruction of  $\mathbf{U}$  from  $\mathbf{C}$  and  $\mathbf{R}$ .

The skeleton decomposition is also referred to as the *CUR decomposition*, named after its component. Compared to the singular value decomposition (SVD), CUR offers significant advantages in terms of *reification* and *interpretability*. While SVD relies on artificial singular vectors that may not accurately reflect physical realities, CUR uses actual columns and rows from the original matrix, which makes it more interpretable and better aligned with the structure of the original data (Mahoney and Drineas, 2009). Moreover, CUR preserves sparsity in the underlying data, making it particularly suitable for applications involving sparse matrices.

On the other hand, like SVD, CUR is a versatile tool widely used across various domains for tasks like data compression, feature extraction, and data analysis. It provides a computationally efficient way to approximate matrices, making it well-suited for handling large-scale datasets (Mahoney and Drineas, 2009; An et al., 2012; Lee and Choi, 2008). For example, CUR reduces the storage and computational requirements by selecting only a subset of the original matrix's rows and columns. This results in a low-rank approximation that retains the essential information. It is particularly useful in numerical linear algebra

for tasks like solving linear systems, eigenvalue problems, and matrix inversion. CUR is also employed for image compression and analysis. By approximating the original image matrix with a lower-dimensional representation, CUR reduces storage costs while preserving key visual features. In machine learning, CUR is effective for dimensionality reduction, feature extraction, and data representation, which can enhance the efficiency of machine learning algorithms and reduce computational overhead. In collaborative filtering, CUR approximates large user-item interaction matrices in recommendation systems, improving scalability and efficiency. CUR decomposition can also be extended to higher-dimensional arrays (tensors) for applications in multi-linear algebra and data analysis, enabling the processing of complex datasets (Kishore Kumar and Schneider, 2017).

An illustration of CUR decomposition is shown in Figure 6.1, where **yellow** vectors denote the linearly independent columns of  $\mathbf{A}$ , and **green** vectors denote the linearly independent rows of  $\mathbf{A}$ . Specifically, given index vectors  $\mathbb{I}$  and  $\mathbb{J}$ , both of size  $r$ , containing the indices of rows and columns selected from  $\mathbf{A}$  to form  $\mathbf{R}$  and  $\mathbf{C}$ , respectively, the submatrix  $\mathbf{U}$  can be expressed as  $\mathbf{U} = \mathbf{A}[\mathbb{I}, \mathbb{J}]$  using Matlab-style notation.

$$\mathbf{A}_{m \times n} = \mathbf{C}_{m \times r} \times \mathbf{U}_{r \times r}^{-1} \times \mathbf{R}_{r \times n}$$

**Figure 6.1:** Demonstration of the skeleton (CUR) decomposition of a matrix.

## 6.2. Existence of the Skeleton Decomposition

In Theorem 0.15, we established that the row rank and column rank of a matrix are equal. In other words, the dimension of the column space is equal to the dimension of the row space. This property is essential for the existence of the skeleton decomposition. We now prove the existence of the skeleton decomposition. The proof is straightforward and relies on fundamental concepts from linear algebra.

**Proof** [of Theorem 6.1] The proof is based on demonstrating the existence of a nonsingular matrix  $\mathbf{U}$ , which is central to the skeleton decomposition.

**Existence of such a nonsingular matrix  $\mathbf{U}$ .** Since the matrix  $\mathbf{A}$  is of rank  $r$ , we can select  $r$  linearly independent columns from  $\mathbf{A}$ . Let these columns be  $\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots, \mathbf{a}_{ir}$  and form the  $m \times r$  matrix  $\mathbf{N} = [\mathbf{a}_{i1}, \mathbf{a}_{i2}, \dots, \mathbf{a}_{ir}] \in \mathbb{R}^{m \times r}$ . The column space of  $\mathbf{N}$  has dimension  $r$ , which implies, by Theorem 0.15, that its row space also has dimension  $r$ . From the rows of  $\mathbf{N}$ , we select  $r$  linearly independent rows  $\mathbf{n}_{j1}^\top, \mathbf{n}_{j2}^\top, \dots, \mathbf{n}_{jr}^\top$  to construct an  $r \times r$  matrix  $\mathbf{U} = [\mathbf{n}_{j1}^\top; \mathbf{n}_{j2}^\top; \dots; \mathbf{n}_{jr}^\top] \in \mathbb{R}^{r \times r}$ . Applying Theorem 0.15 again, the column space of  $\mathbf{U}$  also has dimension  $r$ , meaning  $\mathbf{U}$  has  $r$  linearly independent columns. Thus,  $\mathbf{U}$  is such a nonsingular matrix of size  $r \times r$ .

**Main proof.** Upon identifying a nonsingular  $r \times r$  matrix  $\mathbf{U}$  within  $\mathbf{A}$ , we proceed to establish the skeleton decomposition. Let  $\mathbf{U} = \mathbf{A}[\mathbb{I}, \mathbb{J}]$ , where  $\mathbb{I}$  and  $\mathbb{J}$  are index vectors of

size  $r$  representing the selected rows and columns. Since  $\mathbf{U}$  is a nonsingular matrix, its columns are linearly independent. Thus, the columns of the matrix  $\mathbf{C}$ , formed by selecting the same  $r$  columns from  $\mathbf{A}$  are also linearly independent:  $\mathbf{C} = \mathbf{A}[:, \mathbb{J}]$ . Here, the matrix  $\mathbf{C}$  is equivalent to the previously constructed  $\mathbf{N}$ .

Because the rank of  $\mathbf{A}$  is  $r$ , any column  $\mathbf{a}_i$  of  $\mathbf{A}$  can be expressed as a linear combination of the columns of  $\mathbf{C}$ . Specifically, there exists a vector  $\mathbf{x}$  such that  $\mathbf{a}_i = \mathbf{Cx}$ , for all  $i \in \{1, 2, \dots, n\}$ . Let  $r$  rows (entries) of  $\mathbf{a}_i \in \mathbb{R}^n$  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:

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n] \in \mathbb{R}^{m \times n} \quad \longrightarrow \quad \mathbf{A}[\mathbb{I}, :] = [\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_n] \in \mathbb{R}^{r \times n}.$$

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 states that  $\mathbf{x} = \mathbf{U}^{-1}\mathbf{r}_i$ . Thus, for every  $i \in \{1, 2, \dots, n\}$ , we have  $\mathbf{a}_i = \mathbf{CU}^{-1}\mathbf{r}_i$ . Combining 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 summary, the skeleton decomposition is constructed by identifying  $r$  linearly independent columns of  $\mathbf{A}$  and placing them into  $\mathbf{C} \in \mathbb{R}^{m \times r}$ . Subsequently, we extract an  $r \times r$  nonsingular submatrix  $\mathbf{U}$  from  $\mathbf{C}$ . The  $r$  rows of  $\mathbf{A}$ , corresponding to the entries of  $\mathbf{U}$ , contribute to reconstruct the columns of  $\mathbf{A}$ . This process is visually illustrated in Figure 6.1.

■

In the special case where  $\mathbf{A}$  is square and invertible, the skeleton decomposition simplifies to  $\mathbf{A} = \mathbf{CU}^{-1}\mathbf{R}$ , with  $\mathbf{C} = \mathbf{R} = \mathbf{U} = \mathbf{A}$ . Thus, the decomposition essentially reduces to  $\mathbf{A} = \mathbf{AA}^{-1}\mathbf{A}$ .

**CR decomposition vs skeleton decomposition.** The CR decomposition and skeleton decompositions share a similar structure and even comparable notation, with  $\mathbf{A} = \mathbf{CR}$  for the CR decomposition and  $\mathbf{A} = \mathbf{CU}^{-1}\mathbf{R}$  for the skeleton decomposition.

In both decompositions, we have the flexibility to select the **first**  $r$  linearly independent columns to form the matrix  $\mathbf{C}$  (denoted the same way in both decompositions). Consequently, the  $\mathbf{C}$  matrices in the CR and skeleton decompositions are identical when the same columns are selected. However, the distinction lies in the interpretation of  $\mathbf{R}$ : in the CR decomposition, it represents the reduced row echelon form without zero rows, while in the skeleton decomposition, it corresponds to  $r$  linearly independent rows selected directly from  $\mathbf{A}$ . This difference reflects a fundamental variation in how the two methods conceptualize  $\mathbf{R}$ .

To summarize, the construction of the skeleton decomposition involves selecting  $r$  linearly independent columns from  $\mathbf{A}$  to form the matrix  $\mathbf{C} \in \mathbb{R}^{m \times r}$ . Subsequently, we extract an  $r \times r$  nonsingular submatrix  $\mathbf{U}$  from  $\mathbf{C}$ . Finally, we identify the  $r$  rows of  $\mathbf{A}$  that correspond to the entries of  $\mathbf{U}$  to form the row matrix  $\mathbf{R} \in \mathbb{R}^{r \times n}$ . This naturally leads to the following question: If matrix  $\mathbf{A}$  has rank  $r$ , matrix  $\mathbf{C}$  contains  $r$  linearly independent columns of  $\mathbf{A}$ , and matrix  $\mathbf{R}$  contains  $r$  linearly independent rows of  $\mathbf{A}$ , is the  $r \times r$  “intersection” of  $\mathbf{C}$  and  $\mathbf{R}$  necessarily invertible? <sup>1</sup>

---

<sup>1</sup>. We express our gratitude to Gilbert Strang for raising this question.

**Corollary 6.2: (Nonsingular intersection)** If matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has rank  $r$ , matrix  $\mathbf{C}$  contains  $r$  linearly independent columns of  $\mathbf{A}$ , and matrix  $\mathbf{R}$  contains  $r$  linearly independent rows of  $\mathbf{A}$ , then the  $r \times r$  “intersection” matrix  $\mathbf{U}$  of  $\mathbf{C}$  and  $\mathbf{R}$  is invertible.

**Proof** [of Corollary 6.2] Let  $\mathbb{I}$  and  $\mathbb{J}$  be the indices of the rows and columns selected from  $\mathbf{A}$  to form  $\mathbf{R}$  and  $\mathbf{C}$ , respectively. Then,  $\mathbf{R}$  can be denoted as  $\mathbf{R} = \mathbf{A}[\mathbb{I}, :]$ ,  $\mathbf{C}$  can be represented as  $\mathbf{C} = \mathbf{A}[:, \mathbb{J}]$ , and  $\mathbf{U}$  can be denoted as  $\mathbf{U} = \mathbf{A}[\mathbb{I}, \mathbb{J}]$ .

Since  $\mathbf{C}$  contains  $r$  linearly independent columns of  $\mathbf{A}$ , any column  $\mathbf{a}_i$  of  $\mathbf{A}$  can be represented as  $\mathbf{a}_i = \mathbf{C}\mathbf{x}_i = \mathbf{A}[:, \mathbb{J}]\mathbf{x}_i$  for all  $i \in \{1, 2, \dots, n\}$ . This implies that the  $r$  entries of  $\mathbf{a}_i$  corresponding to the indices in  $\mathbb{I}$  can be represented by the columns of  $\mathbf{U}$  such that  $\mathbf{a}_i[\mathbb{I}] = \mathbf{U}\mathbf{x}_i \in \mathbb{R}^r$  for all  $i \in \{1, 2, \dots, n\}$ , i.e.,

$$\mathbf{a}_i = \mathbf{C}\mathbf{x}_i = \mathbf{A}[:, \mathbb{J}]\mathbf{x}_i \in \mathbb{R}^m \quad \longrightarrow \quad \mathbf{a}_i[\mathbb{I}] = \mathbf{A}[\mathbb{I}, \mathbb{J}]\mathbf{x}_i = \mathbf{U}\mathbf{x}_i \in \mathbb{R}^r.$$

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 the indices in  $\mathbb{J}$  (i.e., the  $r$  columns of  $\mathbf{U}$ ) are linearly independent.

Finally, by Theorem 0.15, the row space of  $\mathbf{U}$  also has dimension  $r$ . This implies that  $\mathbf{U}$  has  $r$  linearly independent rows, making it invertible.  $\blacksquare$

### 6.3. Interpolative Decomposition (ID)

A factorization closely related to the skeleton decomposition is the *interpolative decomposition (ID) framework*. We begin by discussing the *column interpolative decomposition*, which we will refer to simply as interpolative decomposition or ID when the context is clear.

The column interpolative decomposition (ID) factorizes a matrix into the product of two matrices: one consisting of selected columns from the original matrix, and the other containing a subset of columns that includes an identity matrix and entries whose magnitudes do not exceed 1. Formally, the details of the column ID are given in the following theorem.

**Theorem 6.3: (Column interpolative decomposition)** Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be decomposed as

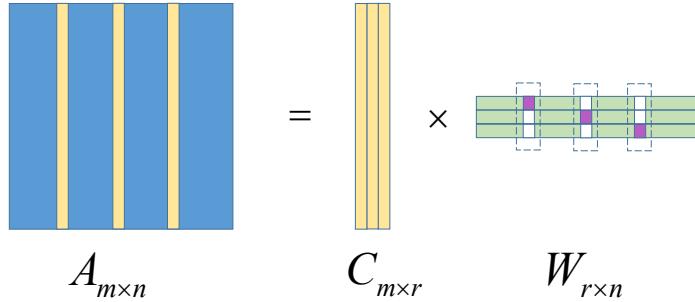
$$\underset{m \times n}{\mathbf{A}} = \underset{m \times r}{\mathbf{C}} \underset{r \times n}{\mathbf{W}},$$

where  $\mathbf{C} \in \mathbb{R}^{m \times r}$  contains  $r$  linearly independent columns of  $\mathbf{A}$ , and  $\mathbf{W} \in \mathbb{R}^{r \times n}$  is the matrix used to reconstruct  $\mathbf{A}$ . The factor  $\mathbf{W}$  contains an  $r \times r$  identity submatrix (under a mild column permutation) and satisfies:

$$\max |w_{ij}| \leq 1, \quad \forall i \in \{1, 2, \dots, r\}, j \in \{1, 2, \dots, n\}.$$

The storage requirements for this decomposition are reduced (or potentially increased) from  $mn$  floating-point numbers to  $mr$  and  $(n - r)r$  floating-point numbers for storing  $\mathbf{C}$  and  $\mathbf{W}$ , respectively. Additionally,  $r$  integers are needed to track the position of each column of  $\mathbf{C}$  within  $\mathbf{A}$ .

Figure 6.2 demonstrates a column ID of a matrix, where the yellow vectors represent some linearly independent columns of  $\mathbf{A}$ , and the purple entries in  $\mathbf{W}$  form an  $r \times r$  identity submatrix. The positions of these purple entries inside  $\mathbf{W}$  correspond to the positions of the



**Figure 6.2:** Demonstration of the column ID of a matrix, where the **yellow** vectors denote some linearly independent columns of  $\mathbf{A}$ , white entries denote zero, and **purple** entries denote one.

**yellow** vectors in  $\mathbf{A}$ . The column ID closely resembles the CR decomposition (Theorem 5.1): both methods select  $r$  linearly independent columns into the first factor, and the second factor contains an  $r \times r$  identity submatrix. However, in the CR decomposition, the first  $r$  linearly independent columns are specifically chosen, and the identity submatrix corresponds to the pivot columns (Definition 1.6). The second factor in the CR decomposition is derived from the reduced row echelon form (Lemma 5.5). As a result, column ID can be used in similar contexts as the CR decomposition. For example, it is useful in proving that the rank of a symmetric idempotent matrix equals its trace (Proposition 5.10), and in demonstrating the fundamental linear algebra theorem that the column rank equals the row rank of a matrix (Theorem 0.15). Moreover, column ID also serves as a special case of the rank decomposition (Theorem 5.8) and is apparently not unique. The relationships among different column IDs are described in Corollary 5.9.

**Notations for subsequent sections.** Using Matlab-style notation, let  $\mathbb{J}_s$  be an index vector of size  $r$ , containing the indices of columns selected from  $\mathbf{A}$  to form the matrix  $\mathbf{C}$ . Then,  $\mathbf{C}$  can be expressed as  $\mathbf{C} = \mathbf{A}[:, \mathbb{J}_s]$ . The matrix  $\mathbf{C}$  contains the “skeleton” columns of  $\mathbf{A}$ , hence the subscript  $s$  in  $\mathbb{J}_s$ . From the “skeleton” index vector  $\mathbb{J}_s$ , the  $r \times r$  identity submatrix inside  $\mathbf{W}$  can be recovered as

$$\mathbf{W}[:, \mathbb{J}_s] = \mathbf{I}_r \in \mathbb{R}^{r \times r}.$$

Let  $\mathbb{J}_r$  denote the indices of the remaining columns of  $\mathbf{A}$ , where

$$\mathbb{J}_s \cap \mathbb{J}_r = \emptyset \quad \text{and} \quad \mathbb{J}_s \cup \mathbb{J}_r = \{1, 2, \dots, n\}.$$

The remaining  $n - r$  columns of  $\mathbf{W}$  consist of an  $r \times (n - r)$  *expansion matrix*:

$$\mathbf{E} = \mathbf{W}[:, \mathbb{J}_r] \in \mathbb{R}^{r \times (n-r)},$$

where the entries of  $\mathbf{E}$  are called *expansion coefficients*. Finally, let  $\mathbf{P} \in \mathbb{R}^{n \times n}$  be a (column) permutation matrix (Definition 0.19) defined as  $\mathbf{P} = \mathbf{I}_n[:, (\mathbb{J}_s, \mathbb{J}_r)]$ , such that

$$\mathbf{AP} = \mathbf{A}[:, (\mathbb{J}_s, \mathbb{J}_r)] = [\mathbf{C}, \mathbf{A}[:, \mathbb{J}_r]],$$

and

$$\mathbf{WP} = \mathbf{W}[:, (\mathbb{J}_s, \mathbb{J}_r)] = [\mathbf{I}_r, \mathbf{E}] \quad \xrightarrow{\text{leads to}} \quad \mathbf{W} = [\mathbf{I}_r, \mathbf{E}] \mathbf{P}^\top. \quad (6.1)$$

## 6.4. Existence of the Column Interpolative Decomposition

**Cramer's rule.** The proof of the existence of the column ID relies on *Cramer's rule*, which we will briefly review here; see Problem 6.10~6.13 for more details. Consider a system of  $n$  linear equations in  $n$  unknowns, expressed in matrix form as:

$$\mathbf{M}\mathbf{x} = \mathbf{l},$$

where  $\mathbf{M} \in \mathbb{R}^{n \times n}$  is nonsingular, and  $\mathbf{x}, \mathbf{l} \in \mathbb{R}^n$ . Cramer's rule states that this system has a unique solution, where each unknown is 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 a more general setting, consider 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  $\mathbb{I} = [i_1, i_2, \dots, i_k]$  and  $\mathbb{J} = [j_1, j_2, \dots, j_k]$  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,  $\mathbf{X}[\mathbb{I}, \mathbb{J}]$  denotes a  $k \times k$  submatrix of  $\mathbf{X}$ . Let further  $\mathbf{M}_{\mathbf{L}}(\mathbb{I}, \mathbb{J})$  be the  $n \times n$  matrix formed by replacing the  $(i_s)$ -th column of  $\mathbf{M}$  with the  $(j_s)$ -th column of  $\mathbf{L}$  for all  $s \in \{1, 2, \dots, k\}$ . Then, we have:

$$\det(\mathbf{X}[\mathbb{I}, \mathbb{J}]) = \frac{\det(\mathbf{M}_{\mathbf{L}}(\mathbb{I}, \mathbb{J}))}{\det(\mathbf{M})}.$$

When  $\mathbb{I}$  and  $\mathbb{J}$  are of size 1, this simplifies to:

$$x_{ij} = \frac{\det(\mathbf{M}_{\mathbf{L}}(i, j))}{\det(\mathbf{M})}. \quad (6.2)$$

With this background, we are now ready to prove the existence of the column ID.

**Proof** [of Theorem 6.3] We mentioned that the proof relies on Cramer's rule. To complete the proof, we will show that the entries of  $\mathbf{W}$  can be expressed using the formula in Equation (6.2), where the absolute value of the numerator is less than or equal to the denominator. Note that the denominator in Equation (6.2) corresponds to the determinant of a square matrix. Here is the key idea.

**Step 1: column ID for a full row rank matrix.** First, consider a 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 to obtain). Determine the “skeleton” index vector  $\mathbb{J}_s$  by

$$\mathbb{J}_s = \arg \max_{\mathbb{J}} \{ |\det(\mathbf{A}[:, \mathbb{J}])| : \mathbb{J} \text{ is a subset of } \{1, 2, \dots, n\} \text{ with size } r = m \},$$

(6.3)

i.e.,  $\mathbb{J}_s$  is the index vector that is determined by maximizing the magnitude of the determinant of  $\mathbf{A}[:, \mathbb{J}_s]$ . From earlier discussion, there exists a (column) permutation matrix  $\mathbf{P}$  such that:

$$\mathbf{AP} = [\mathbf{A}[:, \mathbb{J}_s] \quad \mathbf{A}[:, \mathbb{J}_r]].$$

Since  $\mathbf{C} = \mathbf{A}[:, \mathbb{J}_s]$  has full column rank  $r = m$ , it is then nonsingular. Rewriting  $\mathbf{A}$ :

$$\begin{aligned}\mathbf{A} &= [\mathbf{A}[:, \mathbb{J}_s] \quad \mathbf{A}[:, \mathbb{J}_r]] \mathbf{P}^\top = \mathbf{A}[:, \mathbb{J}_s] \left[ \mathbf{I}_r \quad \mathbf{A}[:, \mathbb{J}_s]^{-1} \mathbf{A}[:, \mathbb{J}_r] \right] \mathbf{P}^\top \\ &= \mathbf{C} \underbrace{[\mathbf{I}_r \quad \mathbf{C}^{-1} \mathbf{A}[:, \mathbb{J}_r]]}_{\mathbf{W}} \mathbf{P}^\top,\end{aligned}$$

where the matrix  $\mathbf{W}$  is given by  $[\mathbf{I}_r \quad \mathbf{C}^{-1} \mathbf{A}[:, \mathbb{J}_r]] \mathbf{P}^\top = [\mathbf{I}_r \quad \mathbf{E}] \mathbf{P}^\top$ , from Equation (6.1). To prove the claim that the magnitude of  $\mathbf{W}$  is at most 1, it suffices to show that each entry of  $\mathbf{E} = \mathbf{C}^{-1} \mathbf{A}[:, \mathbb{J}_r] \in \mathbb{R}^{r \times (n-r)}$  satisfies  $|e_{kl}| \leq 1$  for all  $k \in \{1, 2, \dots, r\}, l \in \{1, 2, \dots, n-r\}$ .

Define the index vector  $[j_1, j_2, \dots, j_n]$  as a permutation of  $[1, 2, \dots, n]$  such that

$$[j_1, j_2, \dots, j_n] = [1, 2, \dots, n] \mathbf{P} = [\mathbb{J}_s, \mathbb{J}_r].$$

Thus, it follows from  $\mathbf{CE} = \mathbf{A}[:, \mathbb{J}_r]$  that

$$\underbrace{[\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}]}_{=\mathbf{C}=\mathbf{A}[:, \mathbb{J}_s]} \mathbf{E} = \underbrace{[\mathbf{a}_{j_{r+1}}, \mathbf{a}_{j_{r+2}}, \dots, \mathbf{a}_{j_n}]}_{=\mathbf{A}[:, \mathbb{J}_r]=\mathbf{B}},$$

where  $\mathbf{a}_i$  denotes the  $i$ -th column of  $\mathbf{A}$ , and we let  $\mathbf{B} = \mathbf{A}[:, \mathbb{J}_r]$ . Therefore, by Cramer's rule in Equation (6.2), we have

$$e_{kl} = \frac{\det(\mathbf{C}_B(k, l))}{\det(\mathbf{C})}, \quad (6.4)$$

where  $e_{kl}$  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  $\mathbf{C}$  with the  $l$ -th column of  $\mathbf{B}$ . For example,

$$\begin{aligned}e_{11} &= \frac{\det([\mathbf{a}_{j_{r+1}}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}, & e_{12} &= \frac{\det([\mathbf{a}_{j_{r+2}}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}, \\ e_{21} &= \frac{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_{r+1}}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}, & e_{22} &= \frac{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_{r+2}}, \dots, \mathbf{a}_{j_r}])}{\det([\mathbf{a}_{j_1}, \mathbf{a}_{j_2}, \dots, \mathbf{a}_{j_r}])}.\end{aligned}$$

Since  $\mathbb{J}_s$  was chosen to maximize  $\det(\mathbf{C})$  in Equation (6.3), it follows that

$$|e_{kl}| \leq 1, \quad \text{for all } k \in \{1, 2, \dots, r\}, l \in \{1, 2, \dots, n-r\}.$$

**Step 2: apply to general matrices.** To summarize, for any matrix  $\mathbf{F} \in \mathbb{R}^{r \times n}$  with full rank  $r \leq n$ , the column ID exists such that  $\mathbf{F} = \mathbf{C}_0 \mathbf{W}$ , where the entries of  $\mathbf{W}$  are bounded by 1 in absolute value.

For a general matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with rank  $r \leq \{m, n\}$ , the matrix admits a rank decomposition (Theorem 5.8) of the form:

$$\mathbf{A}_{m \times n} = \mathbf{D}_{m \times r} \mathbf{F}_{r \times n},$$

where  $\mathbf{D}$  and  $\mathbf{F}$  have full column rank  $r$  and full row rank  $r$ , respectively. Applying the column ID to  $\mathbf{F} = \mathbf{C}_0 \mathbf{W}$ , where  $\mathbf{C}_0 = \mathbf{F}[:, \mathbb{J}_s]$  contains  $r$  linearly independent columns of  $\mathbf{F}$ . We notice from  $\mathbf{A} = \mathbf{DF}$  such that

$$\mathbf{A}[:, \mathbb{J}_s] = \mathbf{DF}[:, \mathbb{J}_s],$$

i.e., the columns indexed by  $\mathbb{J}_s$  of  $(\mathbf{DF})$  can be obtained by  $\mathbf{DF}[:, \mathbb{J}_s]$ , which in turn are the columns of  $\mathbf{A}$  indexed by  $\mathbb{J}_s$ . This makes

$$\underbrace{\mathbf{A}[:, \mathbb{J}_s]}_C = \underbrace{\mathbf{DF}[:, \mathbb{J}_s]}_{DC_0},$$

and

$$\mathbf{A} = \mathbf{DF} = DC_0 \mathbf{W} = \underbrace{\mathbf{DF}[:, \mathbb{J}_s]}_C \mathbf{W} = CW.$$

This completes the proof. ■

The above proof provides an intuitive way to compute the “optimal” column ID of a matrix  $\mathbf{A}$ . However, any algorithm guaranteed to achieve such an optimally conditioned factorization necessarily involves combinatorial complexity due to the need to search for the best column subset  $\mathbb{J}_s$  that maximizes  $|\det(\mathbf{C})|$  (Martinsson, 2019; Lu, 2022a,b). To address this, randomized algorithms, along with approximations via column-pivoted QR (Section 3.8) and rank-revealing QR (Section 3.10), are commonly employed to obtain a relatively well-conditioned column ID decomposition. In these approaches, the matrix  $\mathbf{W}$  is designed to have a small norm rather than strictly ensure that all its entries are within the range  $[-1, 1]$ . On the other hand, Bayesian approaches can strictly constrain the entries of  $\mathbf{W}$  to lie within  $[-1, 1]$  (Lu, 2022a,b). However, these methods involve more advanced techniques and is beyond the scope of this discussion; and we will not elaborate on them here.

**Example 6.4 (Compute the column ID).** Let

$$\mathbf{A} = \begin{bmatrix} 56 & 41 & 30 \\ 32 & 23 & 18 \\ 80 & 59 & 42 \end{bmatrix}$$

be a rank-2 matrix. We now demonstrate the process of computing a column ID of  $\mathbf{A}$ . We begin by finding a rank decomposition of  $\mathbf{A}$ :

$$\mathbf{A} = \mathbf{DF} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} 56 & 41 & 30 \\ 32 & 23 & 18 \end{bmatrix}.$$

Since  $\text{rank}(\mathbf{A}) = 2$ , the index vector  $\mathbb{J}_s$  can take one of the following values:  $[1, 2], [0, 2], [0, 1]$ , where the absolute determinants of  $\mathbf{F}[:, \mathbb{J}_s]$  are 48, 48, and 24, respectively. We proceed with  $\mathbb{J}_s = [0, 2]$ , which yields:

$$\tilde{\mathbf{C}} = \mathbf{F}[:, \mathbb{J}_s] = \begin{bmatrix} 56 & 30 \\ 32 & 18 \end{bmatrix}, \quad \mathbf{M} = \mathbf{F}[:, \mathbb{J}_r] = \begin{bmatrix} 41 \\ 23 \end{bmatrix}.$$

Thus,

$$\mathbf{FP} = \mathbf{F}[:, (\mathbb{J}_s, \mathbb{J}_r)] = \mathbf{F}[:, (0, 2, 1)] \implies \mathbf{P} = \begin{bmatrix} 1 & & \\ & & 1 \\ & 1 & \end{bmatrix}.$$

In this example,  $\mathbf{E} \in \mathbb{R}^{2 \times 1}$ :

$$\begin{aligned} e_{11} &= \det \left( \begin{bmatrix} 41 & 30 \\ 23 & 18 \end{bmatrix} \right) / \det \left( \begin{bmatrix} 56 & 30 \\ 32 & 18 \end{bmatrix} \right) = 1; \\ e_{21} &= \det \left( \begin{bmatrix} 56 & 41 \\ 32 & 23 \end{bmatrix} \right) / \det \left( \begin{bmatrix} 56 & 30 \\ 32 & 18 \end{bmatrix} \right) = -\frac{1}{2}. \end{aligned}$$

This makes

$$\mathbf{E} = \begin{bmatrix} 1 \\ -\frac{1}{2} \end{bmatrix} \implies \mathbf{W} = [\mathbf{I}_2, \mathbf{E}] \mathbf{P}^\top = \begin{bmatrix} 1 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}.$$

The final selected columns and the resulting decomposition are:

$$\mathbf{C} = \mathbf{A}[:, \mathbb{J}_s] = \begin{bmatrix} 56 & 30 \\ 32 & 18 \\ 80 & 42 \end{bmatrix} \implies \mathbf{A} = \mathbf{C}\mathbf{W} = \begin{bmatrix} 56 & 30 \\ 32 & 18 \\ 80 & 42 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 \\ 0 & -\frac{1}{2} & 1 \end{bmatrix}.$$

As expected, the entries of  $\mathbf{W}$  have magnitudes no greater than 1.  $\square$

To conclude this section, we discuss the non-uniqueness of the column ID.

**Remark 6.5 (Non-uniqueness of the column ID).** The column ID is not unique, as illustrated in Example 6.4. Specifically, both  $\mathbf{F}[:, (1, 2)]$  and  $\mathbf{F}[:, (0, 2)]$  yield the maximum absolute determinant. Either choice results in a valid column ID for  $\mathbf{A}$ . Whilst, we only select one  $\mathbb{J}_s$  from  $[1, 2]$ ,  $[0, 2]$ , and  $[0, 1]$ . Additionally, when selecting the index set  $\mathbb{J}_s$ , any permutation of it is also valid. For example,  $\mathbb{J}_s = [0, 2]$  and  $\mathbb{J}_s = [2, 0]$  are both acceptable. This flexibility in selecting the column indices introduces non-uniqueness into the column ID.

## 6.5. Row ID and Two-Sided ID

The decomposition described above is called the column interpolative decomposition, a name that is not arbitrary—it is closely related to other types of interpolative decompositions, as explained below:

**Theorem 6.6: (The whole interpolative decomposition)** Any rank- $r$  matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  can be decomposed 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

- $\mathbf{C} = \mathbf{A}[:, \mathbb{J}_s] \in \mathbb{R}^{m \times r}$  contains  $r$  linearly independent columns of  $\mathbf{A}$ ,  $\mathbf{W} \in \mathbb{R}^{r \times n}$  is the matrix used to reconstruct  $\mathbf{A}$ , which contains an  $r \times r$  identity submatrix (under a mild column permutation):  $\mathbf{W}[:, \mathbb{J}_s] = \mathbf{I}_r$ ;
- $\mathbf{R} = \mathbf{A}[\mathbb{I}_s, :] \in \mathbb{R}^{r \times n}$  contains  $r$  linearly independent rows of  $\mathbf{R}$ ,  $\mathbf{Z} \in \mathbb{R}^{m \times r}$  is the matrix used to reconstruct  $\mathbf{A}$ , which contains an  $r \times r$  identity submatrix (under a mild row permutation):  $\mathbf{Z}[\mathbb{I}_s, :] = \mathbf{I}_r$ ;

- The entries in  $\mathbf{W}$  and  $\mathbf{Z}$  have values no larger than 1 in magnitude:  $\max |w_{ij}| \leq 1$  and  $\max |z_{ij}| \leq 1$ ;
- $\mathbf{U} = \mathbf{A}[\mathbb{I}_s, \mathbb{J}_s] \in \mathbb{R}^{r \times r}$  is the nonsingular submatrix at the intersection of  $\mathbf{C}$  and  $\mathbf{R}$ ;
- The three matrices  $\mathbf{C}$ ,  $\mathbf{R}$ , and  $\mathbf{U}$  in the boxed representations share the same notation and interpretation as in the skeleton decomposition (Theorem 6.1). Specifically,  $\mathbf{A} = \mathbf{C}\mathbf{U}^{-1}\mathbf{R}$  represents the skeleton decomposition.

The proof of the row ID follows similarly from the column ID by transposing. Assume the column ID of  $\mathbf{A}^\top$  is given as  $\mathbf{A}^\top = \mathbf{C}_0\mathbf{W}_0$ , where  $\mathbf{C}_0$  contains  $r$  linearly independent columns of  $\mathbf{A}^\top$  (i.e.,  $r$  linearly independent rows of  $\mathbf{A}$ ). Let  $\mathbf{R} = \mathbf{C}_0^\top$  and  $\mathbf{Z} = \mathbf{W}_0^\top$ . Then, the row ID is obtained as  $\mathbf{A} = \mathbf{Z}\mathbf{R}$ .

From the skeleton decomposition, where  $\mathbf{U}$  is the intersection of  $\mathbf{C}$  and  $\mathbf{R}$ , it follows that  $\mathbf{A} = \mathbf{C}\mathbf{U}^{-1}\mathbf{R}$ . Using the row ID, we get  $\mathbf{C}\mathbf{U}^{-1} = \mathbf{Z}$ , which implies  $\mathbf{C} = \mathbf{Z}\mathbf{U}$ . Substituting into the column ID yields  $\mathbf{A} = \mathbf{C}\mathbf{W} = \mathbf{Z}\mathbf{U}\mathbf{W}$ , thereby proving the existence of the two-sided ID.

**Data storage.** For each ID, the storage requirements are summarized as follows:

- *Column ID.* It requires  $mr$  and  $(n - r)r$  floating-point numbers 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  $(m - r)r$  floating-point numbers 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  $r^2$  floating-point numbers to store  $\mathbf{Z}$ ,  $\mathbf{W}$ , and  $\mathbf{U}$ , respectively. And an extra  $2r$  integers are required to store the indices of the selected rows and columns in  $\mathbf{A}$ .

**Storage reduction for sparse matrices.** For sparse matrices, further storage savings are possible. Consider the column ID:  $\mathbf{A} = \mathbf{C}\mathbf{W}$ , where  $\mathbf{C} = \mathbf{A}[:, \mathbb{J}_s]$ , and a good spanning row index set  $\mathbb{I}_s$  of  $\mathbf{C}$  exists such that:

$$\mathbf{A}[\mathbb{I}_s, :] = \mathbf{C}[\mathbb{I}_s, :] \mathbf{W}.$$

Since  $\mathbf{C}[\mathbb{I}_s, :] = \mathbf{A}[\mathbb{I}_s, \mathbb{J}_s] \in \mathbb{R}^{r \times r}$  is nonsingular, we can compute:

$$\mathbf{W} = (\mathbf{A}[\mathbb{I}_s, \mathbb{J}_s])^{-1} \mathbf{A}[\mathbb{I}_s, :].$$

Thus,  $\mathbf{W}$  does not need to be explicitly stored; only  $\mathbf{A}[\mathbb{I}_s, :]$  and  $(\mathbf{A}[\mathbb{I}_s, \mathbb{J}_s])^{-1}$  are required. Alternatively, If the inverse of  $\mathbf{A}[\mathbb{I}_s, \mathbb{J}_s]$  is computed dynamically, only  $r$  integers for  $\mathbb{J}_s$  are necessary, as  $\mathbf{A}[\mathbb{I}_s, \mathbb{J}_s]$  can be reconstructed from  $\mathbf{A}[\mathbb{I}_s, :]$ . This approach is particularly efficient for sparse matrices, where the storage of  $\mathbf{A}[\mathbb{I}_s, :]$  is economical.

## 6.6. Application: Low-Rank Approximation via Pseudoskeleton

We will explore singular value decomposition (SVD) in detail in Section 11.1. For now, we assume a basic understanding of SVD and demonstrate how it can be used to approximate skeleton decomposition. This section can be skipped during an initial reading.

Given a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , our goal is to construct a rank- $\gamma$  approximation of  $\mathbf{A}$ , where  $\gamma \leq \min(m, n)$ , using skeleton decomposition. Specifically, we approximate  $\mathbf{A}$  as  $\mathbf{A} \approx \mathbf{C}\mathbf{U}^{-1}\mathbf{R}$ , where  $\mathbf{C}$  and  $\mathbf{R}$  are matrices containing  $\gamma$  selected columns and rows, respectively, and  $\mathbf{U}$  is the submatrix formed by the intersection of these selected rows and columns. More precisely, if  $\mathbb{I}$  and  $\mathbb{J}$  denote the indices of the selected rows and columns, then  $\mathbf{U} = \mathbf{A}[\mathbb{I}, \mathbb{J}]$ . Note that  $\gamma$  does not necessarily equal the rank  $r$  of  $\mathbf{A}$ , thus forming a low-rank approximation.

Unlike standard skeleton decomposition, which selects  $r$  linearly independent columns from  $\mathbf{A}$ , we instead choose  $k$  random columns (where  $k > r$  or even  $k = \min\{m, n\}$ ) to form  $\mathbf{C}$ . The column indices  $\mathbb{J}$  determine  $\mathbf{C} = \mathbf{A}[:, \mathbb{J}] \in \mathbb{R}^{m \times k}$ . Simultaneously,  $k$  rows of  $\mathbf{A}$  are selected using the indices  $\mathbb{I}$ , forming  $\mathbf{R} = \mathbf{A}[\mathbb{I}, :]$ . These rows are chosen such that the intersection matrix  $\mathbf{U} = \mathbf{A}[\mathbb{I}, \mathbb{J}]$  has maximal volume; that is,  $\det(\mathbf{U})$  is maximized. While the matrix  $\mathbf{C}$  is selected randomly, the choice of  $\mathbf{R}$  is deterministic. This leads to the decomposition:

$$\mathbf{A} = \mathbf{C}_{m \times k} \mathbf{U}_{k \times k}^{-1} \mathbf{R}_{k \times n}.$$

However, the inverse of  $\mathbf{U}_{k \times k}$  can be numerically unstable due to the random selection of  $\mathbf{C}$ . To address this issue, we perform a full SVD on  $\mathbf{U}_{k \times k}$  (see Section 11.1 for the distinction between reduced 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, and  $\boldsymbol{\Sigma}_k$  is a diagonal matrix containing  $k$  singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k$ , some of which may be zero. We then select  $\gamma$  singular values greater than a threshold  $\epsilon$  and truncate  $\mathbf{U}_k$ ,  $\mathbf{V}_k$ , and  $\boldsymbol{\Sigma}_k$  accordingly, approximating  $\mathbf{U}_{k \times k}$  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 pseudo-inverse of  $\mathbf{U}_{k \times k}$  is

$$\mathbf{U}^+ = (\mathbf{U}_\gamma \boldsymbol{\Sigma}_\gamma \mathbf{V}_\gamma^\top)^+ = \mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1} \mathbf{U}_\gamma^\top.$$

Using this, we can approximate  $\mathbf{A}$  as a rank- $\gamma$  matrix:

$$\begin{aligned} \mathbf{A} &\approx \mathbf{C} \mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1} \mathbf{U}_\gamma^\top \mathbf{R} \\ &= \mathbf{C}_2 \mathbf{R}_2, \quad (\text{let } \mathbf{C}_2 = \mathbf{C} \mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1/2} \text{ and } \mathbf{R}_2 = \boldsymbol{\Sigma}_\gamma^{-1/2} \mathbf{U}_\gamma^\top \mathbf{R}) \end{aligned} \tag{6.5}$$

where  $\mathbf{C}_2$  and  $\mathbf{R}_2$  are rank- $\gamma$  matrices. For guidance on selecting the threshold  $\epsilon$ , refer to Goreinov et al. (1997) and Kishore Kumar and Schneider (2017). In the above approach,  $\mathbf{C}$  is chosen randomly, while  $\mathbf{R}$  is determined. Algorithms such as those in Zhu and Lin (2011), which select both  $\mathbf{C}$  and  $\mathbf{R}$  randomly, tend to yield more stable approximations.

Note that data interpretation methods like CR and skeleton decomposition help identify subsets of columns or rows that retain essential information in the matrix. While Equation (6.5) may not immediately reflect this capability, we also notice that  $\mathbf{C}_2 = \mathbf{C}(\mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1/2})$ , and the columns of  $\mathbf{V}_\gamma \boldsymbol{\Sigma}_\gamma^{-1/2}$  are mutually orthogonal. Thus, the columns of  $\mathbf{C}_2$  represent those of  $\mathbf{C}$  in a different coordinate system. Similarly, the rows of  $\mathbf{R}_2$  correspond to a transformed version of the rows of  $\mathbf{R}$ ; see Section 11.5.

## 6.7. Application: Feature Selection

In quantitative finance, ID and its enhanced variant, *intervened interpolative decomposition (IID)*, are useful tools for feature selection, particularly in the context of developing and optimizing algorithmic trading strategies (Lu, 2022a; Lu and Osterrieder, 2022).

Since ID can serve as a low-rank matrix approximation technique that reconstructs a full data matrix using only a subset of its columns, this allows it to extract key patterns or information from large volumes of market data while preserving sparsity and nonnegativity. IID builds on this idea by incorporating an importance score for each column (i.e., each feature), prioritizing those features considered most relevant or impactful to the model.

In quantitative investing, it's common to encounter thousands—or even millions—of alpha factors (signals that predict future stock returns). Standard ID algorithms can help

identify a small set of representative factors from this large pool. However, these selected factors may not necessarily be the ones with the strongest predictive performance. In contrast, the IID approach selects factors that are both representative—meaning they can reconstruct other alphas with minimal error—and desirable, such as those with high *RankIC scores*, which reflect strong predictive power.

Given the high dimensionality and potential multicollinearity among alpha factors in financial markets, including all available factors in a model can lead to overfitting and computational inefficiency. By applying ID or IID, practitioners can select a smaller but representative subset of alpha factors. This helps reduce overfitting while improving scalability and computational efficiency.

For example, experiments conducted by [Lu and Osterrieder \(2022\)](#) used data from ten assets across various sectors in the Chinese market, including banking, public utilities, and ETFs. Applying the ID or IID method to historical data enabled the identification of significant alpha factors, which were then used to construct effective trading strategies.

In summary, ID and IID provide quantitative analysts with powerful tools to identify the most influential variables when working with complex financial datasets. These techniques contribute to the development of more robust and effective trading strategies. They are especially valuable in handling high-dimensional data, as they reduce computational costs, improve model interpretability, and enhance predictive accuracy.

## ❖ Chapter 6 Problems ❖

1. Find the CUR decomposition for the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 7 & 6 \\ 4 & 5 & 8 \end{bmatrix}.$$

2. Using the CUR decomposition, compute the pseudo-inverse of the matrix  $\mathbf{A}$ .
3. Consider a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  in block form  $\mathbf{A} = \begin{bmatrix} \mathbf{K} & \mathbf{L} \\ \mathbf{M} & \mathbf{N} \end{bmatrix}$ , where  $\mathbf{K} \in \mathbb{R}^{r \times r}$  and  $\mathbf{N} \in \mathbb{R}^{(n-r) \times (n-r)}$ . Show that
  - If  $\mathbf{K}$  is nonsingular, then  $[\mathbf{K}, \mathbf{L}]$  and  $\begin{bmatrix} \mathbf{K} \\ \mathbf{M} \end{bmatrix}$  has full ranks.
  - If  $\text{rank}(\mathbf{A}) = \text{rank}([\mathbf{K}, \mathbf{L}]) = \text{rank}(\begin{bmatrix} \mathbf{K} \\ \mathbf{M} \end{bmatrix})$ , then  $\mathbf{K}$  is nonsingular.
4. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a symmetric or skew-symmetric matrix. Show that
  - $\text{rank}(\mathbf{A}[\mathbb{I}, :]) = \text{rank}(\mathbf{A}[:, \mathbb{I}])$  for any index set  $\mathbb{I} \subseteq \{1, 2, \dots, n\}$ .
  - The matrix  $\mathbf{A}$  is rank principal (i.e., it has a nonsingular  $r \times r$  principal submatrix; Definition 1.2).
5. Discuss algorithms for computing the CUR decomposition using Gaussian elimination or the Gram–Schmidt process. Determine the computational complexity of these algorithms.
6. Investigate how different column and row selection strategies affect the accuracy of the CUR decomposition. Generate a random matrix  $\mathbf{A} \in \mathbb{R}^{100 \times 100}$ , and implement different selection strategies for columns and rows (e.g., random, based on column norms, leverage scores). For each strategy, perform CUR decomposition/approximation with  $r = 10$ . Evaluate the approximation error for each strategy. Discuss which strategy provides the best approximation.

7. Compare the CUR decomposition with other matrix decomposition methods like SVD and QR decomposition in terms of accuracy and computational efficiency. Generate a random matrix  $\mathbf{A} \in \mathbb{R}^{100 \times 100}$ . Perform CUR decomposition, SVD, and QR decomposition on  $\mathbf{A}$ . For each method, compute the approximation error using an appropriate norm. Measure the computational time for each method. Discuss the trade-offs between accuracy and computational efficiency for each method.
8. Suppose that you have an  $n \times n$  matrix where the absolute value of every entry is at most 1. Show that the absolute value of the determinant of this matrix is also at most  $(n)^{n/2}$ . Additionally, provide an example of a  $2 \times 2$  matrix for which the determinant achieves this upper bound.
9. **Adjugate.** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Show that
- $\text{adj}(c\mathbf{A}) = c^{n-1}\text{adj}(\mathbf{A}) \implies \text{adj}(c\mathbf{I}) = c^{n-1}\mathbf{I}$ .
  - $\det(\text{adj}(\mathbf{A})) = (\det(\mathbf{A}))^{n-1}$ .
  - If  $\mathbf{A}$  is nonsingular,  $\mathbf{A}^{-1} = (\det(\mathbf{A}))^{-1}\text{adj}(\mathbf{A})$ .
  - If  $\mathbf{A}$  is nonsingular,  $\text{adj}(\mathbf{A}^{-1}) = \mathbf{A}/\det(\mathbf{A})$ .
  - If  $\mathbf{A}$  is singular and  $\text{rank}(\mathbf{A}) \leq n-2$ ,  $\text{adj}(\mathbf{A}) = \mathbf{0}$ .
  - If  $\mathbf{A}$  is singular and  $\text{rank}(\mathbf{A}) = n-1$ ,  $\text{rank}(\text{adj}(\mathbf{A})) = 1$ .
  - If  $\mathbf{A}, \mathbf{B}$  are nonsingular,  $\text{adj}(\mathbf{AB}) = \text{adj}(\mathbf{A})\text{adj}(\mathbf{B})$ . (This actually can be extended to all  $\mathbf{A}, \mathbf{B}$  due to continuity.)
  - If  $\mathbf{A}$  is nonsingular,  $\text{adj}(\text{adj}(\mathbf{A})) = (\det(\mathbf{A}))^{n-2}\mathbf{A}$ . (This actually can be extended to all  $\mathbf{A}$  due to continuity.)
  - If  $\mathbf{A}$  is nonsingular,  $\text{adj}(\mathbf{A}^\top) = \text{adj}(\mathbf{A})^\top$ . (This actually can be extended to all  $\mathbf{A}$  due to continuity.)
  - The adjugate is the transpose of the gradient of  $\det(\mathbf{A})$ :  $\text{adj}(\mathbf{A})_{ij} = \frac{\partial}{\partial a_{ji}} \det(\mathbf{A})$ .

10. **Cramer's rule.** Consider the linear system  $\mathbf{M}\mathbf{x} = \mathbf{l}$ , where  $\mathbf{M} \in \mathbb{R}^{n \times n}$ , and  $\mathbf{x}, \mathbf{l} \in \mathbb{R}^n$ . Let  $\mathbf{M}_l(i)$  represent the matrix formed by replacing the  $i$ -th column of  $\mathbf{M}$  with  $\mathbf{l}$ . Show that the  $i$ -th element of  $\text{adj}(\mathbf{M})\mathbf{l} \in \mathbb{R}^n$  (where  $\text{adj}(\mathbf{M})$  is the adjugate of  $\mathbf{M}$ ; see Definition 0.23) is

$$(\text{adj}(\mathbf{M})\mathbf{l})_i = \det(\mathbf{M}_l(i)), \quad i \in \{1, 2, \dots, n\}. \quad (6.6)$$

Now consider the linear system  $\mathbf{MX} = \mathbf{L}$ , where  $\mathbf{M} \in \mathbb{R}^{n \times n}$ , and  $\mathbf{X}, \mathbf{L} \in \mathbb{R}^{n \times m}$ . Let  $\mathbf{M}_L(i, j)$  be the matrix formed by replacing the  $i$ -th column of  $\mathbf{M}$  with the  $j$ -th column  $\mathbf{l}_j$  of  $\mathbf{L}$ . Show that the  $(i, j)$ -th element of  $\text{adj}(\mathbf{M})\mathbf{L} \in \mathbb{R}^{n \times m}$  is

$$(\text{adj}(\mathbf{M})\mathbf{L})_{ij} = \det(\mathbf{M}_L(i, j)), \quad i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}. \quad (6.7)$$

*Hint: Examine the definitions of determinant and adjugate (Definition 0.21, 0.23).*

11. **Cramer's rule.** In the same setting as Problem 6.10, represent the vector  $\text{adj}(\mathbf{M})\mathbf{l} \in \mathbb{R}^n$  and the matrix  $\text{adj}(\mathbf{M})\mathbf{L} \in \mathbb{R}^{n \times m}$  as follows:

$$\text{adj}(\mathbf{M})\mathbf{l} = [\det(\mathbf{M}_l(i))]_{i=1}^n \in \mathbb{R}^n, \quad \text{adj}(\mathbf{M})\mathbf{L} = [\det(\mathbf{M}_L(i, j))]_{i,j=1}^{n,m} \in \mathbb{R}^{n \times m},$$

i.e., the  $i$ -th element of the vector is  $\det(\mathbf{M}_l(i))$ , the  $(i, j)$ -th element of the matrix is  $\det(\mathbf{M}_L(i, j))$ . Show that

$$\begin{aligned} \mathbf{M}[\det(\mathbf{M}_l(i))]_{i=1}^n &= \mathbf{M}\text{adj}(\mathbf{M})\mathbf{l} = \det(\mathbf{M})\mathbf{l}; \\ \mathbf{M}[\det(\mathbf{M}_L(i, j))]_{i,j=1}^{n,m} &= \mathbf{M}\text{adj}(\mathbf{M})\mathbf{L} = \det(\mathbf{M})\mathbf{L}. \end{aligned} \quad (6.8)$$

*Hint: Examine the definition of adjugate (Definition 0.23) and Problem 6.9.*

12. **Cramer's rule.** In the same setting as Problem 6.10, assume further that  $\mathbf{M}$  is nonsingular. Show that the  $i$ -th element of the solution  $\mathbf{x}$  is

$$x_i = \frac{\det(\mathbf{M}_l(i))}{\det(\mathbf{M})}, \quad \forall i \in \{1, 2, \dots, n\}. \quad (6.9)$$

Similarly, show that the  $(i, j)$ -th element of the solution  $\mathbf{X}$  is

$$x_{ij} = \frac{\det(\mathbf{M}_{lj}(i, j))}{\det(\mathbf{M})}, \quad \forall i \in \{1, 2, \dots, n\}, j \in \{1, 2, \dots, m\}. \quad (6.10)$$

That is, Cramer's rule.

13. **Cramer's rule: the simple way.** In the same setting as Problem 6.10, assume further that  $\mathbf{M}$  is nonsingular. We notice that

$$\mathbf{M}\mathbf{I}_l(i) = \mathbf{M}_l(i), \quad \forall i \in \{1, 2, \dots, n\}, \quad (6.11)$$

where  $\mathbf{I}_l(i)$  represents the identity matrix with the  $i$ -th column replaced by  $\mathbf{l}$ . Taking the determinant yields:

$$\det(\mathbf{M}) \det(\mathbf{I}_l(i)) = \det(\mathbf{M}_l(i)). \quad (6.12)$$

Show that  $\det(\mathbf{I}_l(i)) = x_i$ , thereby verifying the result in (6.9).

14. **Determinant of inverses for subsets, Jacobi's equality.** Let  $\mathbf{M} \in \mathbb{R}^{n \times n}$ , and let  $\mathbb{I}, \mathbb{J} \subseteq \{1, 2, \dots, n\}$  be two index sets (their complementary sets are  $\mathbb{I}^C$  and  $\mathbb{J}^C$ , respectively). Show that

$$\det(\mathbf{M}^{-1}[\mathbb{I}^C, \mathbb{J}^C]) = (-1)^\gamma \frac{\det(\mathbf{M}[\mathbb{J}, \mathbb{I}])}{\det(\mathbf{M})}, \quad (6.13)$$

where  $\gamma = \sum_{i \in \mathbb{I}} i + \sum_{j \in \mathbb{J}} j$  is the sum of indices. When  $\mathbb{I} = \mathbb{J}$ , this also indicates

$$\det(\mathbf{M}^{-1}[\mathbb{I}^C, \mathbb{I}^C]) = \frac{\det(\mathbf{M}[\mathbb{I}, \mathbb{I}])}{\det(\mathbf{M})}, \quad (6.14)$$

which is known as *Jacobi's equality*. Hint: Examine the definitions of determinant and adjugate (Definitions 0.21 and 0.23). Alternatively, you may prove this using the Schur complement.

15. Determine the column ID for the matrix  $\mathbf{A} = \begin{bmatrix} 1 & 3 & 2 \\ 3 & 7 & 6 \\ 4 & 5 & 8 \end{bmatrix}$ .
16. **Matlab-style notation.** Consider a rectangular matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  of rank  $r$ , which admits a rank decomposition  $\mathbf{A} = \mathbf{D}\mathbf{F}$ . Let  $\mathbb{I}, \mathbb{J} \subseteq \{1, 2, \dots, m\}$  and  $\mathbb{K}, \mathbb{L} \subseteq \{1, 2, \dots, n\}$  be index sets with cardinality  $|\mathbb{I}| = |\mathbb{J}| = |\mathbb{K}| = |\mathbb{L}| = r$ . Then, we have  $\mathbf{A}[\mathbb{I}, \mathbb{K}] = \mathbf{D}[\mathbb{I}, :] \mathbf{F}[:, \mathbb{K}]$ . Show that
- $\mathbf{A}[\mathbb{I}, \mathbb{K}]$  is nonsingular if and only if  $\text{rank}(\mathbf{D}[\mathbb{I}, :]) = \text{rank}(\mathbf{F}[:, \mathbb{K}]) = r$ .
  - $\det(\mathbf{A}[\mathbb{I}, \mathbb{K}]) \det(\mathbf{A}[\mathbb{J}, \mathbb{L}]) = \det(\mathbf{A}[\mathbb{I}, \mathbb{L}]) \det(\mathbf{A}[\mathbb{J}, \mathbb{K}])$ .
17. Discuss algorithms for computing the column ID of a matrix using its column-pivoted QR decomposition.

# Part IV

## Reduction to Hessenberg, Tridiagonal, and Bidiagonal Form



## Hessenberg and Tridiagonal Decomposition

### Preliminary

In real applications, we often aim to factor a matrix  $\mathbf{A}$  into two orthogonal matrices such that  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , where  $\Lambda$  is either diagonal or upper triangular. Common examples include eigenanalysis via Schur decomposition and principal component analysis (PCA) using spectral decomposition. This type of factorization can be computed through 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 transformation can be difficult to handle in practice; for instance, when using Householder reflectors. Following the approach used in QR decomposition with Householder reflectors, the sequence of orthogonal similarity transformations can be constructed using Householder reflectors:

$$\begin{array}{ccc} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\mathbf{H}_1 \times} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\times \mathbf{H}_1^\top} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{array} \right], \\ \mathbf{A} & & \mathbf{H}_1 \mathbf{A} & & \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \end{array}$$

where the left Householder ( $\mathbf{H}_1 \times$ ) introduces zeros in the first column below the main diagonal (see Section 3.11), but unfortunately, the right Householder ( $\times \mathbf{H}_1^\top$ ) will undo the zeros created by the left Householder.

However, we can simplify the process by splitting the algorithm into two phases. In the first phase, we transform the matrix into a Hessenberg matrix (Definition 7.1) or a tridiagonal matrix (Definition 7.9). Then, in the second phase, we apply an iterative algorithm to convert the results from the first phase into the desired form. This leads to the following algorithmic structure:

$$\begin{array}{ccc} \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\mathbf{H}_1 \times} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \end{array} \right] & \xrightarrow{\times \mathbf{H}_1^\top} & \left[ \begin{array}{cccccc} \square & \square & \square & \square & \square \\ \square & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \\ 0 & \square & \square & \square & \square \end{array} \right] \dots \\ \mathbf{A} & & \mathbf{H}_1 \mathbf{A} & & \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \end{array}$$

In this case, the left Householder does not affect the first row, and the right Householder does not disturb the first column. A Phase 2 algorithm, typically iterative, for finding the triangular matrix is as follows:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\text{Phase 2}} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \textcolor{blue}{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \end{array} \right] \\ \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \mathbf{H}_3^\top \quad \Lambda \end{array}$$

As discussed above, to compute spectral decomposition, Schur decomposition, or singular value decomposition (SVD), we often make a trade-off. In the first phase, we reduce the matrix to Hessenberg, tridiagonal, or bidiagonal form. The second stage then completes the decomposition using an iterative method (Van Zee et al., 2012, 2014; Trefethen and Bau III, 1997).

## 7.1. Hessenberg Decomposition

The *Hessenberg decomposition* is a technique used to transform a matrix into an upper Hessenberg form. This transformation simplifies the matrix structure, making it an effective first step in various algorithms, as it reduces computational complexity. Let's begin with a formal definition of upper Hessenberg matrices.

**Definition 7.1 (Upper Hessenberg matrix).** An *upper Hessenberg matrix* (simply called Hessenberg matrix when the context is clear) is a square matrix in which all entries below the subdiagonal are zero. Similarly, a *lower Hessenberg matrix* is a square matrix in which all the entries above the superdiagonal are zero. The definition can be extended to rectangular matrices, where the structure is implied by the context.

Formally, for a matrix  $\mathbf{H} \in \mathbb{R}^{n \times n}$ , with elements  $h_{ij}$  for  $i, j \in \{1, 2, \dots, n\}$ ,  $\mathbf{H}$  is an upper Hessenberg matrix if  $h_{ij} = 0$  for all  $i \geq j + 2$ .

Additionally, if  $i$  is the smallest positive integer for which  $h_{i+1,i} = 0$  for  $i \in \{1, 2, \dots, n-1\}$ , then  $\mathbf{H}$  is called **unreduced** if  $i = n$ .

Consider a  $5 \times 5$  matrix. In an upper Hessenberg matrix, all elements below the first subdiagonal are zero:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \quad \text{or} \quad \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \textcolor{blue}{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes \end{array} \right] \\ \text{possibly unreduced} \quad \text{reduced} \end{array}$$

We now state the Hessenberg decomposition:

**Theorem 7.2: (Hessenberg decomposition)** Any  $n \times n$  square matrix  $\mathbf{A}$  can be decomposed as

$$\mathbf{A} = \mathbf{Q} \mathbf{H} \mathbf{Q}^\top,$$

where  $\mathbf{H}$  is an upper Hessenberg matrix, and  $\mathbf{Q}$  is an orthogonal matrix.

For a lower Hessenberg decomposition, the transpose of  $\mathbf{A}$ ,  $\mathbf{A}^\top$ , admits the decomposition  $\mathbf{A}^\top = \mathbf{Q} \mathbf{H}^\top \mathbf{Q}^\top$  if  $\mathbf{A}$  admits the Hessenberg decomposition  $\mathbf{A} = \mathbf{Q} \mathbf{H} \mathbf{Q}^\top$ . The Hessenberg

decomposition is conceptually similar to the QR decomposition in that both aim to reduce a matrix to a sparser form with zeros in the lower portion.

While the left orthogonal matrix  $\mathbf{Q}$  introduces zeros in  $\mathbf{H}$  (similar to the QR decomposition), the right orthogonal matrix  $\mathbf{Q}^\top$  does not simplify the matrix further. Then why use Hessenberg decomposition instead of QR decomposition, which achieves zeros even in the lower subdiagonal? The answer lies in the intended application. Hessenberg decomposition serves as a preparatory step (phase one) for more advanced factorizations like singular value decomposition (SVD) or UTV decomposition. A more aggressive transformation (e.g., QR decomposition) would introduce zeros in the subdiagonal but disrupt zeros during subsequent transformations.

Furthermore, the form  $\mathbf{A} = \mathbf{Q}\mathbf{H}\mathbf{Q}^\top$  is an *orthogonal similarity transformation* (Definition 7.3), preserving key properties of  $\mathbf{A}$ , such as its eigenvalues, rank, and trace (Proposition 7.4). Thus, studying  $\mathbf{H}$  provides a simplified way to understand the behavior of  $\mathbf{A}$ .

Moreover, let  $\mathbf{A} = \mathbf{Q}\mathbf{H}\mathbf{Q}^\top \in \mathbb{R}^{n \times n}$  be given. In certain scenarios, we may need to solve the linear system  $(\mathbf{A} + \gamma\mathbf{I})\mathbf{x} = \mathbf{b}$  for different values of  $\gamma \in \mathbb{R}$  and  $\mathbf{b} \in \mathbb{R}^n$ . The linear system can be equivalently expressed as  $(\mathbf{H} + \gamma\mathbf{I})\mathbf{Q}^\top\mathbf{x} = \mathbf{Q}^\top\mathbf{b}$ . Since  $\mathbf{H}$  is upper Hessenberg, the system can be solved efficiently using methods like forward and backward substitution.

## 7.2. (Orthogonal) Similarity Transformation

As mentioned earlier, the Hessenberg decomposition introduced in this section, the tridiagonal decomposition in the next section, the Schur decomposition (Theorem 9.7), and the spectral decomposition (Theorem 10.1) all share a common structure: they transform a matrix into another matrix that is similar to it. Below, we formally define similar matrices and similarity transformations.

**Definition 7.3 (Similar matrices and similarity transformation).** Two matrices  $\mathbf{A}$  and  $\mathbf{B}$  are said to be *similar matrices* if there exists a nonsingular matrix  $\mathbf{P}$  such that  $\mathbf{B} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}$ .

In simpler terms, given any nonsingular matrix  $\mathbf{P}$ , the matrices  $\mathbf{A}$  and  $\mathbf{P}\mathbf{A}\mathbf{P}^{-1}$  are similar. The transformation  $\mathbf{P}\mathbf{A}\mathbf{P}^{-1}$  is referred to as a *similarity transformation* of the matrix  $\mathbf{A}$ .

Furthermore, if  $\mathbf{P}$  is an orthogonal matrix, the transformation  $\mathbf{P}\mathbf{A}\mathbf{P}^\top$  is also known as an *orthogonal similarity transformation* of  $\mathbf{A}$ . Orthogonal similarity transformations are particularly significant because the condition number of the transformed matrix  $\mathbf{P}\mathbf{A}\mathbf{P}^\top$  is no worse than that of the original matrix  $\mathbf{A}$ . <sup>a</sup>

a. Note that two matrices  $\mathbf{A}$  and  $\mathbf{B}$  are referred to as *congruent* if  $\mathbf{B} = \mathbf{S}\mathbf{A}\mathbf{S}^\top$  for some nonsingular matrix  $\mathbf{S}$ . In this sense, an orthogonal similarity transformation is both a similarity transformation and a congruence transformation.

The distinction between similarity transformations and orthogonal similarity transformations will be further clarified in the context of coordinate transformations (Section 11.5). We now proceed to establish some important properties of similar matrices, which will prove useful in later discussions.

**Proposition 7.4: (Eigenvalue, trace, and rank of similar matrices)** Any eigenvalue of  $\mathbf{A}$  is also an eigenvalue of  $\mathbf{P}\mathbf{A}\mathbf{P}^{-1}$ , and vice versa. That is,  $\Lambda(\mathbf{A}) = \Lambda(\mathbf{B})$ , where  $\Lambda(\mathbf{X})$  denotes the spectrum of matrix  $\mathbf{X}$  (Definition 0.2).

Moreover, the trace and rank of  $\mathbf{A}$  are equal to those of  $\mathbf{PAP}^{-1}$  for any nonsingular matrix  $\mathbf{P}$ .

**Proof** [of Proposition 7.4] Let  $(\lambda, \mathbf{x})$  be any eigenpair of  $\mathbf{A}$  so that  $\mathbf{Ax} = \lambda\mathbf{x}$ . Then we have  $\lambda\mathbf{Px} = \mathbf{PAP}^{-1}\mathbf{Px}$  such that  $\mathbf{Px}$  is an eigenvector of  $\mathbf{PAP}^{-1}$  corresponding to  $\lambda$ . Conversely, for any eigenpair  $(\lambda, \mathbf{x})$  of  $\mathbf{PAP}^{-1}$ , we have  $\mathbf{PAP}^{-1}\mathbf{x} = \lambda\mathbf{x}$ . Then we have  $\mathbf{AP}^{-1}\mathbf{x} = \lambda\mathbf{P}^{-1}\mathbf{x}$  such that  $\mathbf{P}^{-1}\mathbf{x}$  is an eigenvector of  $\mathbf{A}$  corresponding to  $\lambda$ .

Next, consider the trace. Using the cyclic invariance property of the trace, we have:  $\text{tr}(\mathbf{PAP}^{-1}) = \text{tr}(\mathbf{AP}^{-1}\mathbf{P}) = \text{tr}(\mathbf{A})$ .

For the rank, we proceed in two steps:

**Rank claim 1:**  $\text{rank}(\mathbf{ZA}) = \text{rank}(\mathbf{A})$  if  $\mathbf{Z}$  is nonsingular. For any vector  $\mathbf{n}$  in the null space of  $\mathbf{A}$  (i.e.,  $\mathbf{An} = \mathbf{0}$ ), we have  $\mathbf{ZAn} = \mathbf{0}$ . Hence,  $\mathbf{n}$  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  $\mathbf{ZA}$  (i.e.,  $\mathbf{ZAm} = \mathbf{0}$ ), 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  $\mathcal{N}(\mathbf{ZA}) \subseteq \mathcal{N}(\mathbf{A})$ .

Combining both inclusions, we conclude:

$$\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{ZA}) \implies \text{rank}(\mathbf{ZA}) = \text{rank}(\mathbf{A}).$$

**Rank claim 2:**  $\text{rank}(\mathbf{AZ}) = \text{rank}(\mathbf{A})$  if  $\mathbf{Z}$  is nonsingular. Using the equality of row and column ranks (Theorem 0.15), we have  $\text{rank}(\mathbf{AZ}) = \text{rank}(\mathbf{Z}^\top \mathbf{A}^\top)$ . Since  $\mathbf{Z}^\top$  is nonsingular, applying claim 1 gives  $\text{rank}(\mathbf{Z}^\top \mathbf{A}^\top) = \text{rank}(\mathbf{A}^\top) = \text{rank}(\mathbf{A})$ , where the last equality follows again from the fact that the row rank is equal to the column rank for any matrix. This results in  $\text{rank}(\mathbf{AZ}) = \text{rank}(\mathbf{A})$ , as claimed.

Combining these results, and noting that both  $\mathbf{P}$  and  $\mathbf{P}^{-1}$  are nonsingular, we have  $\text{rank}(\mathbf{PAP}^{-1}) = \text{rank}(\mathbf{AP}^{-1}) = \text{rank}(\mathbf{A})$ . This completes the proof.  $\blacksquare$

### 7.3. Existence of Hessenberg Decomposition

We will demonstrate that any  $n \times n$  matrix can be transformed into Hessenberg form through a sequence of Householder transformations applied alternately from the left and the right. These transformations are performed in an interleaved manner. Previously, we used Householder reflectors to triangularize matrices by introducing zeros below the diagonal, as part of the QR decomposition process. A similar strategy can be employed to introduce zeros below the subdiagonal, enabling the transformation to Hessenberg form. Before delving into the mathematical construction of this decomposition, we highlight the following remark, which will prove essential for deriving the decomposition.

**Remark 7.5 (Left and right multiplied by a matrix with block identity).** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a square matrix, and let  $\mathbf{B} = \begin{bmatrix} \mathbf{I}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_{n-k} \end{bmatrix}$ , where  $\mathbf{I}_k$  is the a  $k \times k$  identity matrix. Then,  $\mathbf{BA}$  does not alter the first  $k$  rows of  $\mathbf{A}$ , and  $\mathbf{AB}$  does not alter the first  $k$  columns of  $\mathbf{A}$ .

#### First Step: Introduce Zeros for the First Column

Let  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$  be the column partition of  $\mathbf{A}$ , where 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 the vectors obtained by removing the first component in  $\mathbf{a}_i$ 's. Define

$$r_1 = \|\bar{\mathbf{a}}_1\|, \quad \mathbf{u}_1 = \frac{\bar{\mathbf{a}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{a}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \widetilde{\mathbf{H}}_1 = \mathbf{I} - 2\mathbf{u}_1 \mathbf{u}_1^\top \in \mathbb{R}^{(n-1) \times (n-1)},$$

where  $\mathbf{e}_1$  is the first unit basis in  $\mathbb{R}^{n-1}$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^{n-1}$ . To introduce zeros below the subdiagonal 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  $\mathbf{A}$  remains unchanged, as noted in Remark 7.5. Furthermore, it is straightforward to verify that both  $\mathbf{H}_1$  and  $\widetilde{\mathbf{H}}_1$  are symmetric and orthogonal matrices. To obtain the form in Theorem 7.2, we multiply  $\mathbf{H}_1\mathbf{A}$  on the right by  $\mathbf{H}_1^\top$ , resulting in  $\mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top$ . The multiplication on the right will not affect the first column of  $\mathbf{H}_1\mathbf{A}$ , preserving the zeros introduced in that 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:

$$\begin{array}{ccc} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_1 \times} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} & \xrightarrow{\times \mathbf{H}_1^\top} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} \\ \mathbf{A} & & \mathbf{H}_1\mathbf{A} & & \mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top \end{array}$$

### 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. The goal now is to introduce zeros in the second column below entry (3,2). Define  $\mathbf{B}_2 = \mathbf{B}_{2:n,2:n} = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{n-1}]$ . Let again  $\bar{\mathbf{b}}_1, \bar{\mathbf{b}}_2, \dots, \bar{\mathbf{b}}_{n-1} \in \mathbb{R}^{n-2}$  be the vectors obtained by removing the first component from each  $\mathbf{b}_i$ . We can again construct a Householder reflector:

$$r_1 = \|\bar{\mathbf{b}}_1\|, \quad \mathbf{u}_2 = \frac{\bar{\mathbf{b}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{b}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \widetilde{\mathbf{H}}_2 = \mathbf{I} - 2\mathbf{u}_2\mathbf{u}_2^\top \in \mathbb{R}^{(n-2) \times (n-2)}, \quad (7.1)$$

where  $\mathbf{e}_1$  is now the first unit basis in  $\mathbb{R}^{n-2}$ . To introduce zeros below the subdiagonal and operate on the submatrix  $\mathbf{B}_{3:n,1:n}$ , we extend 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 the  $2 \times 2$  identity matrix. We can see that the product  $\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top$  does not alter the first two rows of  $\mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top$ ; and since the Householder transformation cannot reflect a zero vector, the zeros in the first column are preserved. Again, applying  $\mathbf{H}_2^\top$  to the right of  $\mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top$  will not change the first two columns, thus preserving the previously introduced zeros.

Following the example of a  $5 \times 5$  matrix, the second step is shown as follows:

$$\begin{array}{ccc} \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} & \xrightarrow{\mathbf{H}_2 \times} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ 0 & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ 0 & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ 0 & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} & \xrightarrow{\times \mathbf{H}_2^\top} & \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \\ \mathbf{0} & \mathbf{0} & \mathbf{\boxtimes} & \mathbf{\boxtimes} & \mathbf{\boxtimes} \end{bmatrix} \\ \mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top & & \mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top & & \mathbf{H}_2\mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top \mathbf{H}_2^\top \end{array}$$

This process continues iteratively, and a total of  $n - 2$  such steps are required. In the end, the matrix will be transformed into Hessenberg form:

$$\mathbf{H} = \mathbf{H}_{n-2}\mathbf{H}_{n-3} \dots \mathbf{H}_1\mathbf{A}\mathbf{H}_1^\top \mathbf{H}_2^\top \dots \mathbf{H}_{n-2}^\top.$$

Since each  $\mathbf{H}_i$  is symmetric and orthogonal, this simplifies 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 that only  $n - 2$  stages are required, rather than  $n - 1$  or  $n$ . This can be verified using the full example for a  $5 \times 5$  matrix:

$$\begin{array}{c}
 \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{matrix} \xrightarrow{\mathbf{H}_1 \times} \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{matrix} \xrightarrow{\times \mathbf{H}_1^\top} \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{matrix} \\
 \mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top
 \end{array}$$
  

$$\begin{array}{c}
 \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{matrix} \xrightarrow{\mathbf{H}_2 \times} \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{matrix} \xrightarrow{\times \mathbf{H}_2^\top} \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{matrix} \\
 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \qquad \qquad \qquad \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top
 \end{array}$$
  

$$\begin{array}{c}
 \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{matrix} \xrightarrow{\mathbf{H}_3 \times} \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{matrix} \xrightarrow{\times \mathbf{H}_3^\top} \begin{matrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{matrix} \\
 \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \qquad \qquad \qquad \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{H}_1^\top \mathbf{H}_2^\top \mathbf{H}_3^\top
 \end{array}.$$

## 7.4. Properties of Hessenberg Decomposition

The Hessenberg decomposition is not unique, as there are multiple ways to construct the Householder reflectors (e.g., Equation (7.1)). However, under mild conditions, different decompositions exhibit a similar structure.

**Theorem 7.6: (Implicit Q theorem for Hessenberg decomposition)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a matrix with two Hessenberg decompositions,  $\mathbf{A} = \mathbf{U}\mathbf{H}\mathbf{U}^\top = \mathbf{V}\mathbf{G}\mathbf{V}^\top$ , 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  $\mathbf{U}$  and  $\mathbf{V}$ , respectively. Assume  $k$  is the smallest positive integer such that  $h_{k+1,k} = 0$ , where  $h_{ij}$  denotes the  $(i, j)$  entry 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  $k = n$ , the Hessenberg matrix  $\mathbf{H}$  is called *unreduced*. Otherwise, if  $k < n$ , then  $g_{k+1,k} = 0$ .

**Proof** [of Theorem 7.6] Define the orthogonal matrix  $\mathbf{Z} = \mathbf{V}^\top \mathbf{U}$ . We have:

$$\left. \begin{aligned} \mathbf{GZ} &= \mathbf{V}^\top \mathbf{AVV}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{AU} \\ \mathbf{ZH} &= \mathbf{V}^\top \mathbf{UU}^\top \mathbf{AU} = \mathbf{V}^\top \mathbf{AU} \end{aligned} \right\} \implies \mathbf{GZ} = \mathbf{ZH}.$$

For the  $(i-1)$ -th column, we have  $\mathbf{Gz}_{i-1} = \mathbf{Zh}_{i-1}$ , where  $\mathbf{z}_{i-1}$  and  $\mathbf{h}_{i-1}$  are the  $(i-1)$ -th columns of  $\mathbf{Z}$  and  $\mathbf{H}$ , respectively. Since  $h_{l,i-1} = 0$  for  $l \geq i+1$  (as per the definition of upper Hessenberg matrices),  $\mathbf{Zh}_{i-1}$  can be represented as

$$\mathbf{Zh}_{i-1} = \sum_{j=1}^i h_{j,i-1} \mathbf{z}_j = h_{i,i-1} \mathbf{z}_i + \sum_{j=1}^{i-1} h_{j,i-1} \mathbf{z}_j.$$

Combining results, we have  $h_{i,i-1} \mathbf{z}_i = \mathbf{Gz}_{i-1} - \sum_{j=1}^{i-1} h_{j,i-1} \mathbf{z}_j$ . A moment of reflexion reveals that  $[\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k]$  is upper triangular. Since  $\mathbf{Z}$  is orthogonal, it must be diagonal, and each value on the diagonal is in  $\{-1, 1\}$  for  $i \in \{2, \dots, k\}$ . Then  $\mathbf{z}_1 = \mathbf{e}_1$

and  $\mathbf{z}_i = \pm \mathbf{e}_i$  for  $i \in \{2, \dots, k\}$ . Additionally,  $\mathbf{z}_i = \mathbf{V}^\top \mathbf{u}_i$  and  $h_{i,i-1} = \mathbf{z}_i^\top (\mathbf{G}\mathbf{z}_{i-1} - \sum_{j=1}^{i-1} h_{j,i-1} \mathbf{z}_j) = \mathbf{z}_i^\top \mathbf{G}\mathbf{z}_{i-1}$ . Therefore, for  $i \in \{2, \dots, k\}$ ,  $\mathbf{z}_i^\top \mathbf{G}\mathbf{z}_{i-1}$  is just  $\pm g_{i,i-1}$ . It follows that  $|h_{i,i-1}| = |g_{i,i-1}|$  and  $\mathbf{u}_i = \pm \mathbf{v}_i, \forall i \in \{2, 3, \dots, k\}$ . This proves the first part. For the second part, if  $k < n$ ,

$$\begin{aligned} g_{k+1,k} &= \mathbf{e}_{k+1}^\top \mathbf{G}\mathbf{e}_k = \pm \mathbf{e}_{k+1}^\top \underbrace{\mathbf{G}\mathbf{Z}}_{\mathbf{Z}\mathbf{H}} \mathbf{e}_k = \pm \mathbf{e}_{k+1}^\top \underbrace{\mathbf{Z}\mathbf{H}\mathbf{e}_k}_{\text{k-th column of } \mathbf{Z}\mathbf{H}} \\ &= \pm \mathbf{e}_{k+1}^\top \mathbf{Z}\mathbf{h}_k = \pm \mathbf{e}_{k+1}^\top \sum_{j=1}^{k+1} h_{jk} \mathbf{z}_j = \pm \mathbf{e}_{k+1}^\top \sum_{j=1}^k h_{jk} \mathbf{z}_j = 0, \end{aligned}$$

where the penultimate equality is derived from the assumption that  $h_{k+1,k} = 0$ . This completes the proof.  $\blacksquare$

From the above theorem, we observe that if two Hessenberg decompositions of a matrix  $\mathbf{A}$  are both unreduced and share the same first column in their respective orthogonal matrices, the corresponding Hessenberg matrices  $\mathbf{H}$  and  $\mathbf{G}$  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 impose the condition that the elements on the lower subdiagonal of the Hessenberg matrix  $\mathbf{H}$  are 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 property is analogous to the uniqueness of the QR decomposition (as established in Corollary 3.10) and is crucial for simplifying the QR algorithm, which is widely used for computing the singular value decomposition or eigenvalues of a matrix (Golub and Van Loan, 2013; Lu, 2021c).

The next concept we introduce is that of a Krylov matrix, defined as follows:

**Definition 7.7 (Krylov matrix).** Given a 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 as:

$$\mathbf{K}(\mathbf{A}, \mathbf{q}, k) = [\mathbf{q}, \mathbf{A}\mathbf{q}, \dots, \mathbf{A}^{k-1}\mathbf{q}] \in \mathbb{R}^{n \times k}.$$

**Theorem 7.8: (Unreduced Hessenberg)** Suppose there exists an orthogonal matrix  $\mathbf{Q}$  such that a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  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 such that  $r_{kk} = 0$ , then  $k$  is also the smallest index satisfying  $h_{k,k-1} = 0$ .

**Proof** [of Theorem 7.8] Assume  $\mathbf{H}$  is an unreduced upper Hessenberg matrix. Write out the following matrix

$$\mathbf{R} = \mathbf{Q}^\top \mathbf{K}(\mathbf{A}, \mathbf{q}_1, n) = [e_1, \mathbf{H}e_1, \dots, \mathbf{H}^{n-1}e_1],$$

where, obviously,  $\mathbf{R}$  is upper triangular with  $r_{11} = 1$ . 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.

Conversely, assume  $\mathbf{R}$  is upper triangular and nonsingular. We observe the recurrence  $\mathbf{r}_{k+1} = \mathbf{H}\mathbf{r}_k$ , which implies that the  $(k+2 : n)$ -th rows of  $\mathbf{H}[:, 1:k]$  are zero and  $h_{k+1,k} \neq 0$  for  $k \in \{1, 2, \dots, n-1\}$ . Thus,  $\mathbf{H}$  is unreduced.

If  $\mathbf{R}$  is singular and  $k$  is the smallest index satisfying  $r_{kk} = 0$ , then

$$\left. \begin{array}{l} r_{k-1,k-1} = h_{21}h_{32} \dots h_{k-1,k-2} \neq 0 \\ r_{kk} = h_{21}h_{32} \dots h_{k-1,k-2}h_{kk-1} = 0 \end{array} \right\} \xrightarrow{\text{leads to}} h_{k,k-1} = 0,$$

from which the result follows.  $\blacksquare$

## 7.5. Hessenberg-Triangular Decomposition

A factorization that is closely related to the Hessenberg decomposition is called the *Hessenberg-triangular decomposition* for a pair of matrices. Given a matrix pair  $(\mathbf{A}, \mathbf{B})$ , where  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$ , a preprocessing step of the *QZ decomposition or generalized Schur decomposition* (Moler and Stewart, 1973) for solving the regular *generalized eigenvalue problem*  $(\mathbf{A} - \lambda \mathbf{B})\mathbf{x} = \mathbf{0}$  involves computing orthogonal matrices  $\mathbf{Q}, \mathbf{Z} \in \mathbb{R}^{n \times n}$  such that  $\mathbf{Q}^\top \mathbf{A} \mathbf{Z}$  is upper Hessenberg while  $\mathbf{Q}^\top \mathbf{B} \mathbf{Z}$  is upper triangular. This so-called *Hessenberg-triangular (HT) form* of the matrix pair  $(\mathbf{A}, \mathbf{B})$  significantly reduces the computational cost during the iterative part of the QZ algorithm, which in turn plays a crucial role in the computation of quadratic eigenvalue problems (Zhang, 2017).

The reduction to HT form begins by computing a QR decomposition  $\mathbf{B} = \mathbf{Q}_0 \mathbf{B}_0$ , where  $\mathbf{Q}_0$  is orthogonal and  $\mathbf{B}_0$  is upper triangular. The matrices  $\mathbf{A}$  and  $\mathbf{B}$  are then overwritten by  $\mathbf{Q}_0^\top \mathbf{A}$  and  $\mathbf{Q}_0^\top \mathbf{B} = \mathbf{B}_0$ , respectively. Thus, for the rest of this section, we assume that the matrix  $\mathbf{B}$  in the pair  $(\mathbf{A}, \mathbf{B})$  is already in upper triangular form. In the HT algorithm, the matrix  $\mathbf{A}$  is then reduced to Hessenberg form by applying a sequence of Givens rotations. The goal is to reduce  $\mathbf{A}$  to Hessenberg form while maintaining the triangular form of  $\mathbf{B}$ . This is achieved by premultiplying  $\mathbf{A}$  with Householder reflections or Givens rotations to annihilate elements below the first subdiagonal, and postmultiplying  $\mathbf{B}$  with a different set of Householder reflections or Givens rotations to preserve its triangular form (we use Givens rotations in Algorithm 9).

---

**Algorithm 9** Moler and Stewart's HT reduction (Moler and Stewart, 1973)

---

**Require:** A general matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and an upper triangular matrix  $\mathbf{B} \in \mathbb{R}^{n \times n}$ ;

**Ensure:** Orthogonal  $\mathbf{Q}, \mathbf{Z} \in \mathbb{R}^{n \times n}$  such that  $(\mathbf{H}, \mathbf{T}) = (\mathbf{Q}^\top \mathbf{A} \mathbf{Z}, \mathbf{Q}^\top \mathbf{B} \mathbf{Z})$  is in HT form;

- 1: **Remark:**  $\mathbf{L}_{i-1,i}, \mathbf{R}_{i,i-1} \in \mathbb{R}^{n \times n}$  denote Givens rotations (Section 3.12) acting on rows/- columns  $i-1$  and  $i$ .
  - 2: Initially set  $\mathbf{Q} \leftarrow \mathbf{I}_n$ ,  $\mathbf{Z} \leftarrow \mathbf{I}_n$ ,  $\mathbf{H} \leftarrow \mathbf{A}$ , and  $\mathbf{T} \leftarrow \mathbf{B}$ ;
  - 3: **for**  $j = 1, 2, \dots, n-2$  **do** ▷ Introduce zeros in the  $j$ -th column of  $\mathbf{A}$
  - 4:   **for**  $i = n, n-1, \dots, j+2$  **do**
  - 5:     Construct  $\mathbf{L}_{i-1,i}$  such that the  $(i, j)$ -th entry of  $\mathbf{L}_{i-1,i}^\top \mathbf{H}$  is zero.
  - 6:     Update  $\mathbf{H} \leftarrow \mathbf{L}_{i-1,i}^\top \mathbf{H}$ ,  $\mathbf{T} \leftarrow \mathbf{L}_{i-1,i}^\top \mathbf{T}$ ,  $\mathbf{Q} \leftarrow \mathbf{Q} \mathbf{L}_{i-1,i}$ .
  - 7:     Construct  $\mathbf{R}_{i,i-1}$  such that the fill-in  $(i, i-1)$  entry of  $\mathbf{T} \mathbf{R}_{i,i-1}$  is zero.
  - 8:     Update  $\mathbf{H} \leftarrow \mathbf{H} \mathbf{R}_{i,i-1}$ ,  $\mathbf{T} \leftarrow \mathbf{T} \mathbf{R}_{i,i-1}$ ,  $\mathbf{Z} \leftarrow \mathbf{Z} \mathbf{R}_{i,i-1}$ .
  - 9:   **end for**
  - 10: **end for**
  - 11: Output  $(\mathbf{H}, \mathbf{T}) = (\mathbf{Q}^\top \mathbf{A} \mathbf{Z}, \mathbf{Q}^\top \mathbf{B} \mathbf{Z})$ .
- 

An example of a  $7 \times 7$  matrix is shown as follows at  $i = 5$  and  $j = 2$ , where  $\boxtimes$  represents a value that is not necessarily zero, and **boldface** indicates the value has just been changed. The **blue** elements are introduced to zero from a nonzero value; while the **brown** elements

are modified to nonzero from a zero value:

$$\begin{array}{c} \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \textcolor{blue}{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \quad \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \textcolor{brown}{\boxtimes} & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \rightarrow \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 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \quad \left[ \begin{array}{cccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \textcolor{blue}{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & 0 & 0 & 0 & \boxtimes \end{array} \right]. \\ \mathbf{L}_{4,5}^\top \mathbf{H} \qquad \mathbf{L}_{4,5}^\top \mathbf{T} \qquad \mathbf{L}_{4,5}^\top \mathbf{H} \mathbf{R}_{5,4} \qquad \mathbf{L}_{4,5}^\top \mathbf{T} \mathbf{R}_{5,4} \end{array}$$

## 7.6. Tridiagonal Decomposition: Hessenberg in Symmetric Matrices

Similar to the Hessenberg decomposition, the *tridiagonal decomposition* simplifies matrices and serves as a preliminary step for other algorithms (e.g., diagonalization of a matrix), reducing their computational complexity. We begin by formally defining tridiagonal matrices.

**Definition 7.9 (Tridiagonal matrix).** A *tridiagonal matrix* is a square matrix, where all the entries below the subdiagonal and the entries above the superdiagonal are zero. In other words, a tridiagonal matrix is a special type of *band matrix*.

The concept of a tridiagonal matrix can also extend to rectangular matrices, with the form inferred from context.

Formally, consider a matrix  $\mathbf{T} \in \mathbb{R}^{n \times n}$  with entries  $t_{ij}$  for  $i, j \in \{1, 2, \dots, n\}$ . The matrix  $\mathbf{T}$  is tridiagonal if  $t_{ij} = 0$  for all  $i \geq j + 2$  and  $i \leq j - 2$ .

Additionally, let  $i$  denote the smallest positive integer such that  $h_{i+1,i} = 0$  for  $i \in \{1, 2, \dots, n-1\}$ . The matrix  $\mathbf{T}$  is termed **unreduced** if  $i = n$ .

For example, the following  $5 \times 5$  matrix is a tridiagonal matrix:

$$\begin{array}{c} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right] \quad \text{or} \quad \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & 0 \\ 0 & 0 & \textcolor{blue}{0} & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{array} \right]. \\ \text{possibly unreduced} \qquad \qquad \qquad \text{reduced} \end{array}$$

Clearly, a tridiagonal matrix is a special case of an upper Hessenberg matrix. This allows us to formulate the tridiagonal decomposition as follows:

**Theorem 7.10: (Tridiagonal decomposition)** Any  $n \times n$  symmetric matrix  $\mathbf{A}$  can be decomposed as

$$\mathbf{A} = \mathbf{Q} \mathbf{T} \mathbf{Q}^\top,$$

where  $\mathbf{T}$  is a *symmetric* tridiagonal matrix, and  $\mathbf{Q}$  is an orthogonal matrix.

The existence of the tridiagonal decomposition follows directly from applying the Hessenberg decomposition to the symmetric matrix  $\mathbf{A}$ .

## 7.7. Properties of Tridiagonal Decomposition

Like the Hessenberg decomposition, the tridiagonal decomposition is generally not unique. However, a similar implicit Q theorem can be stated.

**Theorem 7.11: (Implicit Q theorem for tridiagonal)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a symmetric matrix with two tridiagonal decompositions:  $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^\top = \mathbf{V}\mathbf{G}\mathbf{V}^\top$ , 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  $\mathbf{U}$  and  $\mathbf{V}$ , respectively. Suppose further that  $k$  is the smallest positive integer such that  $t_{k+1,k} = 0$ , 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  $k = n$ , the tridiagonal matrix  $\mathbf{T}$  is called unreduced. However, if  $k < n$ , then  $g_{k+1,k} = 0$ .

From the above theorem, we see that constraining the elements of the subdiagonal of  $\mathbf{T}$  to be positive (if possible) ensures that 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 is again analogous to the uniqueness of the QR decomposition (see Corollary 3.10).

Similarly, a reduced tridiagonal decomposition can be derived using the Krylov matrix (Definition 7.7).

**Theorem 7.12: (Unreduced 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 satisfying  $r_{kk} = 0$ , then  $k$  is also the smallest index such that  $t_{k,k-1} = 0$ .

## 🌊 Chapter 7 Problems 🌊

1. Show that if  $\lambda$  is a nonzero eigenvalue of  $\mathbf{AB}$ , then it is also a nonzero eigenvalue of  $\mathbf{BA}$ . Explain why this reasoning does not hold when  $\lambda = 0$ .
2. Show that if either  $\mathbf{A}$  or  $\mathbf{B}$  is invertible, then the matrices  $\mathbf{AB}$  and  $\mathbf{BA}$  are similar.
3. Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  be similar matrices. Show that  $\text{adj}(\mathbf{A})$  and  $\text{adj}(\mathbf{B})$  are also similar.
4. Let  $\mathbf{A}$  be given and  $\mathbf{P}$  be nonsingular. Show that if  $\mathbf{PAP}^{-1}$  is upper triangular, then the diagonal entries of  $\mathbf{PAP}^{-1}$  are the eigenvalues of  $\mathbf{A}$ .
5. **Power property of similar matrices.** Let  $\mathbf{B} = \mathbf{PAP}^{-1}$ . Show that  $\mathbf{B}^k = \mathbf{PA}^k\mathbf{P}^{-1}$  for  $k = 1, 2, \dots$ ; that is,  $\mathbf{B}^k$  and  $\mathbf{A}^k$  are similar if  $\mathbf{B}$  and  $\mathbf{A}$  are similar. If one of  $\mathbf{A}$  and  $\mathbf{B}$  is nonsingular, show that  $\mathbf{B}^{-1} = \mathbf{PA}^{-1}\mathbf{P}^{-1}$  also holds.
6. In the main section, we transform the given matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  into its orthogonal similarity transformation. Use Gaussian elimination matrices (see (1.1)) to transform into its similarity transformation. Discuss the complexity of your algorithm.
7. Show that if  $\mathbf{A} = \mathbf{ECE}^{-1}$  and  $\mathbf{B} = \mathbf{FCF}^{-1}$ , then  $\mathbf{A}$  and  $\mathbf{B}$  are similar matrices.
8. Show that the matrices  $\begin{bmatrix} 4 & 1 \\ -1 & 0 \end{bmatrix}$  and  $\begin{bmatrix} 1 & 1 \\ 0 & 3 \end{bmatrix}$  are similar.
9. **Polynomial.** Let  $\mathbf{A}$  and  $\mathbf{B}$  be similar, and consider a polynomial  $p(\mathbf{C}) = \gamma_n \mathbf{C}^n + \gamma_{n-1} \mathbf{C}^{n-1} + \dots + \gamma_0$ . Show that  $p(\mathbf{A})$  and  $p(\mathbf{B})$  are also similar.
10. **Polynomial.** Let  $\mathbf{A}$  and a nonsingular  $\mathbf{P}$  be given, and consider a polynomial  $p(\mathbf{C}) = \gamma_n \mathbf{C}^n + \gamma_{n-1} \mathbf{C}^{n-1} + \dots + \gamma_0$ . Show that  $p(\mathbf{PAP}^{-1}) = \mathbf{P}p(\mathbf{A})\mathbf{P}^{-1}$ .
11. **Similarity transformation.** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and let  $\mathbf{P} \in \mathbb{R}^{n \times n}$  be nonsingular. Show that  $\det(\mathbf{P}^{-1}\mathbf{AP} - \lambda\mathbf{I}) = \det(\mathbf{A} - \lambda\mathbf{I})$ . This again demonstrates that the eigenvalues remain unchanged under similarity transformations.

12. Let  $\mathbf{H} \in \mathbb{R}^{n \times n}$  be an unreduced upper Hessenberg matrix. Show that  $\text{rank}(\mathbf{H} - \lambda \mathbf{I}) \geq n - 1$  for any  $\lambda \in \mathbb{R}$ .
13. Let  $\mathbf{H} \in \mathbb{R}^{n \times n}$  be an unreduced upper Hessenberg matrix. Show that its geometric multiplicity is 1 for any eigenvalue (Definition 10.10).
14. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be given with a lower bandwidth of  $p$  (Definition 1.11). Provide an algorithm that computes the Hessenberg decomposition of  $\mathbf{A}$  using Householder reflectors or Givens rotations.
15. **Hessenberg LU.** Let  $\mathbf{H} \in \mathbb{R}^{n \times n}$  be upper Hessenberg. Show that there exists a set of Gaussian elimination matrices  $\mathbf{E}_1, \mathbf{E}_2, \dots, \mathbf{E}_{n-1}$  with entries bounded by unity (see Equation (1.1)) and a set of permutation matrices  $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_{n-1}$  such that  $\mathbf{E}_{n-1}\mathbf{P}_{n-1} \dots \mathbf{E}_2\mathbf{P}_2\mathbf{E}_1\mathbf{P}_1\mathbf{H}$  is upper triangular. Discuss the complexity of your algorithm.
16. **Hessenberg QR.** Let  $\mathbf{H} \in \mathbb{R}^{n \times n}$  be upper Hessenberg. Provide an algorithm that computes the QR decomposition of  $\mathbf{H}$  using Givens rotations with a complexity of  $\mathcal{O}(n^2)$  flops.
17. Let  $\mathbf{H} \in \mathbb{R}^{n \times n}$  be upper Hessenberg with an eigenpair  $(\lambda, \mathbf{v})$ . Provide an algorithm that computes an orthogonal matrix  $\mathbf{Q}$  such that  $\mathbf{Q}^\top \mathbf{H} \mathbf{Q} = \begin{bmatrix} \lambda & \mathbf{u}^\top \\ \mathbf{0} & \mathbf{H}_1 \end{bmatrix}$ , where  $\mathbf{H}_1 \in \mathbb{R}^{(n-1) \times (n-1)}$  is also upper Hessenberg. *Hint: Consider  $\mathbf{Q}$  as a product of Givens rotations.*
18. (Read Chapter 10 first) Consider a  $4 \times 4$  Hessenberg matrix:

$$\mathbf{H} = \begin{bmatrix} b_1 & c_1 & d_1 & e_1 \\ a_1 & b_2 & c_2 & d_2 \\ 0 & a_2 & b_3 & c_3 \\ 0 & 0 & a_3 & b_4 \end{bmatrix}.$$

Show that

- If  $a_1, a_2, a_3$  are all nonzero, and any eigenvalue  $\lambda$  of  $\mathbf{H}$  is a real number, then the geometric multiplicity (Definition 10.10) of  $\lambda$  must be equal to one.
  - If  $\mathbf{H}$  is similar to a symmetric matrix  $\mathbf{A}$ , and the algebraic multiplicity (Definition 10.10) of some eigenvalue  $\lambda$  of  $\mathbf{A}$  is greater than 1, then at least one of  $a_1, a_2, a_3$  must be zero.
19. Consult Kagström et al. (2008); Bujanovic et al. (2018) and derive the complexity of the Hessenberg-triangular decomposition.
20. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Show that  $\mathbf{A}$  is idempotent (i.e.,  $\mathbf{A}^2 = \mathbf{A}$ ) if and only if there exists an orthogonal matrix  $\mathbf{B} \in \mathbb{R}^{n \times n}$  such that  $\mathbf{A}$  and  $\mathbf{B}$  are similar.
21. Show that if  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is similar to an orthogonal matrix, then  $\mathbf{A}^{-1}$  is similar to  $\mathbf{A}^\top$ .
22. Show that all Householder reflection matrices are similar.
23. Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be a matrix with full column rank. The matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  is known as a projection matrix. Show that all projection matrices  $\mathbf{H}$  obtained by varying  $\mathbf{A}$  (but for particular values of  $m$  and  $n$ ) are similar. *Hint: Use the QR decomposition of  $\mathbf{A}$ .*
24. Show that all Givens matrices with the same rotation angle  $\theta$  are similar.
25. (Golub and Van Loan, 2013) Let  $\mathbf{A} = \mathbf{S} + \sigma \mathbf{u} \mathbf{u}^\top \in \mathbb{R}^{n \times n}$ , where  $\mathbf{S} \in \mathbb{R}^{n \times n}$  is skew-symmetric (satisfying  $\mathbf{A}^\top = -\mathbf{A}$ ),  $\mathbf{u} \in \mathbb{R}^n$ , and  $\sigma \in \mathbb{R}$ . Show that there exists an orthogonal matrix  $\mathbf{Q}$  such that  $\mathbf{Q}^\top \mathbf{A} \mathbf{Q} = \mathbf{T} + \sigma \mathbf{e}_1 \mathbf{e}_1^\top$ , where  $\mathbf{T}$  is tridiagonal and skew-symmetric.

26. Let  $\mathbf{H} \in \mathbb{R}^{n \times n}$  be upper Hessenberg. Provide an algorithm that computes the decomposition  $\mathbf{H}\mathbf{R} = \mathbf{RT}$ , where  $\mathbf{R}$  is unit upper triangular, and  $\mathbf{T}$  is tridiagonal.
27. Based on the proofs of Theorems 7.6 and 7.8, prove Theorems 7.11 and 7.12.
28. Let  $\gamma_0, \gamma_1, \dots, \gamma_n > 0$ . Show that the following  $n \times n$  tridiagonal matrix is positive definite:

$$\begin{bmatrix} \gamma_0 + \gamma_1 & -\gamma_1 & 0 & \dots & 0 \\ -\gamma_1 & \gamma_1 + \gamma_2 & -\gamma_2 & \dots & 0 \\ 0 & -\gamma_2 & \gamma_2 + \gamma_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \gamma_{n-1} + \gamma_n \end{bmatrix}.$$

*Hint: Consider the leading principal minors.*

29. (Higham, 2002a) Let  $\mathbf{T}_n(a, b, c) \in \mathbb{R}^{n \times n}$  be a tridiagonal matrix defined as:

$$(\text{toeplitz tridiagonal matrix}) : \quad \mathbf{T}_n(a, b, c) = \begin{bmatrix} b & c & 0 & \dots & 0 \\ a & b & c & \dots & 0 \\ 0 & a & b & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & b \end{bmatrix}.$$

Show that the eigenvalues of  $\mathbf{T}_n(a, b, c)$  are  $b + 2\sqrt{ac} \cos(\frac{k\pi}{n+1})$  for  $k \in \{1, 2, \dots, n\}$ .

30. (Noschese et al., 2013) Show that the matrix  $\mathbf{T}_n(a, b, c)$  is normal ( $\mathbf{T}_n^\top \mathbf{T}_n = \mathbf{T}_n \mathbf{T}_n^\top$ ) if and only if  $|a| = |c|$ .
31. Let  $\mathbf{T} \in \mathbb{R}^{n \times n}$  be an unreduced tridiagonal matrix. Show that  $\text{rank}(\mathbf{T} - \lambda \mathbf{I}) \geq n - 1$  for any  $\lambda \in \mathbb{R}$ .
32. Let  $\mathbf{T} \in \mathbb{R}^{n \times n}$  be an unreduced tridiagonal matrix. Show that its geometric multiplicity (Definition 10.10) is 1 for any eigenvalue.
33. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be tridiagonal. Show that if  $a_{i,i+1}a_{i+1,i} > 0$  for all  $i \in \{1, 2, \dots, n-1\}$ , then  $\mathbf{A}$  has  $n$  distinct real eigenvalues. Moreover, show that if  $a_{i,i+1}a_{i+1,i} \geq 0$  for all  $i \in \{1, 2, \dots, n-1\}$ , then all eigenvalues of  $\mathbf{A}$  are real. *Hint: Use the Jordan decomposition discussed in Chapter 9.*

# 8

## Bidiagonal Decomposition

### 8.1. Bidiagonal Decomposition

For a non-square symmetric matrix, reducing it to tridiagonal form is not straightforward. However, we can take an alternative approach by considering a decomposition that involves two distinct orthogonal matrices. To begin, we formally define upper bidiagonal matrices:

**Definition 8.1 (Upper bidiagonal matrix).** An *upper bidiagonal matrix* or simply *bidiagonal matrix* is a square matrix characterized by a banded structure, containing nonzero entries only along the *main diagonal* and the *superdiagonal* (i.e., the diagonal directly above the main diagonal). In this case, the matrix contains exactly two diagonals with nonzero entries.

If the nonzero entries instead appear on the diagonal directly below the main diagonal (i.e., the *subdiagonal*), the matrix is referred to as a *lower bidiagonal matrix*.

This definition can be naturally extended to rectangular matrices, where the bidiagonal structure can be implied based on the context.

As an example, consider a  $7 \times 5$  upper bidiagonal matrix. In such a matrix, all entries below the main diagonal and above the superdiagonal are zero:

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

We now state the following result regarding bidiagonal decomposition:

**Theorem 8.2: (Bidiagonal decomposition)** Any  $m \times n$  matrix  $\mathbf{A}$  can be decomposed as

$$\mathbf{A} = \mathbf{U}\mathbf{B}\mathbf{V}^\top,$$

where  $\mathbf{B}$  is an upper bidiagonal matrix, and  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  are orthogonal matrices.

The process of bidiagonalization shares structural similarities with the singular value decomposition (SVD). The key difference lies in the form of  $\mathbf{B}$ , which, in the bidiagonal

decomposition, contains nonzero entries specifically on the superdiagonal. This distinction plays an important role in the numerical computation of the singular value decomposition (Golub and Van Loan, 2013; Lu, 2021c).

## 8.2. Existence of Bidiagonal Decomposition: Three Approaches

In earlier discussions, we employed Householder reflectors to triangularize matrices, achieving the QR decomposition by introducing zeros below the main diagonal and the Hessenberg decomposition by introducing zeros below the subdiagonal. A similar strategy can be applied to compute the bidiagonal decomposition.

### 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 partition of  $\mathbf{A}$ , where each  $\mathbf{a}_i \in \mathbb{R}^m$ . We 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 standard basis vector in  $\mathbb{R}^m$ , i.e.,  $\mathbf{e}_1 = [1; 0; 0; \dots; 0] \in \mathbb{R}^m$ . The matrix  $\mathbf{H}_1$  is symmetric and orthogonal (from the definition of Householder reflectors). Applying  $\mathbf{H}_1$  to  $\mathbf{A}$  introduces zeros in the first column of  $\mathbf{A}$  below the  $(1, 1)$  entry, effectively reflecting  $\mathbf{a}_1$  to  $r_1 \mathbf{e}_1$ .

For example, consider a  $7 \times 5$  matrix  $\mathbf{A}$ . The transformation is illustrated below, where  $\boxtimes$  represents a potentially nonzero value, and **boldface** indicates entries modified by the transformation:

$$\begin{array}{c} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \xrightarrow{\mathbf{H}_1 \times} \left[ \begin{array}{cccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{A} \qquad \qquad \qquad \mathbf{H}_1 \mathbf{A} \end{array}.$$

At this stage, the process is similar to the steps used in the QR decomposition using Householder reflectors, as described in Section 3.11. To proceed, introducing zeros above the superdiagonal in  $\mathbf{H}_1 \mathbf{A}$  is equivalent to introducing zeros below the subdiagonal of  $(\mathbf{H}_1 \mathbf{A})^\top$ .

### First Step 1.2: Introduce Zeros for the First Row

Now, consider the transpose of  $\mathbf{H}_1 \mathbf{A}$ , denoted as  $(\mathbf{H}_1 \mathbf{A})^\top = \mathbf{A}^\top \mathbf{H}_1^\top \in \mathbb{R}^{n \times m}$ . 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  $\mathbf{z}_i \in \mathbb{R}^n$ . Let  $\bar{\mathbf{z}}_1, \bar{\mathbf{z}}_2, \dots, \bar{\mathbf{z}}_m \in \mathbb{R}^{n-1}$  represent the vectors obtained by removing the first component of each  $\mathbf{z}_i$ . We can construct the Householder reflector as follows:

$$r_1 = \|\bar{\mathbf{z}}_1\|, \quad \mathbf{v}_1 = \frac{\bar{\mathbf{z}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{z}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \tilde{\mathbf{L}}_1 = \mathbf{I} - 2\mathbf{v}_1 \mathbf{v}_1^\top \in \mathbb{R}^{(n-1) \times (n-1)},$$

where  $\mathbf{e}_1$  now denotes the first standard basis vector in  $\mathbb{R}^{n-1}$ . To introduce zeros below the subdiagonal and operate on the submatrix  $(\mathbf{A}^\top \mathbf{H}_1^\top)_{2:n, 1:m}$ , we extend the Householder reflector into  $\mathbf{L}_1 = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{L}}_1 \end{bmatrix}$ , where both  $\mathbf{L}_1$  and  $\tilde{\mathbf{L}}_1$  are orthogonal and symmetric (by the definition of Householder reflectors). In this case, multiplying  $(\mathbf{A}^\top \mathbf{H}_1^\top)$  by  $\mathbf{L}_1$  on the left introduces 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  $(\mathbf{A}^\top \mathbf{H}_1^\top)$  remains unchanged, as noted in Remark 7.5, ensuring that the zeros introduced in step (1.1) are preserved.

Returning to the original (untransposed) matrix  $\mathbf{H}_1 \mathbf{A}$ , multiplying on the right by  $\mathbf{L}_1^\top$  introduces zeros in the first row to the right of entry (1, 2). To illustrate, using the same  $7 \times 5$  matrix, the transformation is shown below:

$$\begin{array}{ccc} \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{L}_1 \times} & \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{(\cdot)^\top} & \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{A}^\top \mathbf{H}_1^\top & & \mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top & & \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \end{array}.$$

In summary,  $\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top$  achieves the first step by introducing zeros to both the first column and the first row of  $\mathbf{A}$ . The second step is similar, which introduces zeros to both the second column and the second 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 all entries in the first column below entry (1,1) and all entries in the first row to the right of (1,2) are zero. The goal of this step is to introduce zeros in the second column below the entry at (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 as follows:

$$r_1 = \|\mathbf{b}_1\|, \quad \mathbf{u}_2 = \frac{\mathbf{b}_1 - r_1 \mathbf{e}_1}{\|\mathbf{b}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \widetilde{\mathbf{H}}_2 = \mathbf{I} - 2\mathbf{u}_2 \mathbf{u}_2^\top \in \mathbb{R}^{(m-1) \times (m-1)},$$

where  $\mathbf{e}_1$  now is the first standard basis vector for  $\mathbb{R}^{m-1}$ . To introduce zeros below the main diagonal and operate on the submatrix  $\mathbf{B}_{2:m, 2:n}$ , we append the Householder reflector into  $\mathbf{H}_2 = \begin{bmatrix} 1 & 0 \\ \mathbf{0} & \widetilde{\mathbf{H}}_2 \end{bmatrix}$ . This transformation ensures that  $\mathbf{H}_2(\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top)$  does not affect the first row of  $(\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top)$ , as shown in Remark 7.5. Additionally, because a Householder reflector preserves zero vectors, the zeros in the first column remain unchanged.

Continuing from the previous example, the process applied to the  $7 \times 5$  matrix is illustrated below:

$$\begin{array}{ccc} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{H}_2 \times} & \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top & & \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \end{array}.$$

### Second Step 2.2: Introduce Zeros for the Second Row

Following step (1.2), consider the transpose of  $\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top$ , given by  $\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top \in \mathbb{R}^{n \times m}$ . Assume the column partition of  $\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top$  is  $\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]$ , where each  $\mathbf{x}_i \in \mathbb{R}^n$ . Let  $\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2, \dots, \bar{\mathbf{x}}_m \in \mathbb{R}^{n-2}$  denote the vectors obtained by removing the first two components of each  $\mathbf{x}_i$ . We construct the next Householder reflector:

$$r_1 = \|\bar{\mathbf{x}}_1\|, \quad \mathbf{v}_2 = \frac{\bar{\mathbf{x}}_1 - r_1 \mathbf{e}_1}{\|\bar{\mathbf{x}}_1 - r_1 \mathbf{e}_1\|}, \quad \text{and} \quad \widetilde{\mathbf{L}}_2 = \mathbf{I} - 2\mathbf{v}_2 \mathbf{v}_2^\top \in \mathbb{R}^{(n-2) \times (n-2)},$$

where  $\mathbf{e}_1$  now is the first standard basis vector for  $\mathbb{R}^{n-2}$ . To introduce zeros below the subdiagonal 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  $\mathbf{L}_2 = \begin{bmatrix} \mathbf{I}_2 & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{L}}_2 \end{bmatrix}$ , where  $\mathbf{I}_2$  is the  $2 \times 2$  identity matrix. By definition,  $\mathbf{L}_2$  and  $\tilde{\mathbf{L}}_2$  are both symmetric and orthogonal matrices. 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  $(\mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top)$  remain unaffected and kept unchanged, as noted in Remark 7.5. Furthermore, its first column will be kept unchanged as well.

Returning to the *untransposed* matrix  $\mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top$ , multiplying on the right by  $\mathbf{L}_2^\top$  introduces zeros in the second row to the right of entry (2,3). As before, the transformation for the  $7 \times 5$  matrix is illustrated below:

$$\begin{array}{ccc} \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{L}_2 \times} & \left[ \begin{array}{cccccc} \boxtimes & 0 & 0 & 0 & 0 & 0 & 0 \\ \boxtimes & \boxtimes & 0 & 0 & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{(\cdot)^\top} & \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \mathbf{0} & \mathbf{0} \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top & & \mathbf{L}_2 \mathbf{L}_1 \mathbf{A}^\top \mathbf{H}_1^\top \mathbf{H}_2^\top & & \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \mathbf{L}_2^\top \end{array} .$$

Thus,  $\mathbf{H}_2(\mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top) \mathbf{L}_2^\top$  completes the second step by introducing zeros into the second column and row of  $\mathbf{A}$ .

This process can be continued iteratively. It is important to observe that there are  $n$  left reflectors, denoted as  $\mathbf{H}_i$ , and  $n - 2$  right reflectors, denoted as  $\mathbf{L}_i$  (suppose  $m > n$  for simplicity). This alternating application of left and right reflectors is commonly referred to as the *Golub–Kahan bidiagonalization* (Golub and Kahan, 1965). Ultimately, this procedure yields the following bidiagonalized form:

$$\mathbf{B} = \mathbf{H}_n \mathbf{H}_{n-1} \dots \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \mathbf{L}_2^\top \dots \mathbf{L}_{n-2}^\top.$$

Since all  $\mathbf{H}_i$ 's and  $\mathbf{L}_i$ 's are symmetric and orthogonal by definition, this can also be expressed as:

$$\mathbf{B} = \mathbf{H}_n \mathbf{H}_{n-1} \dots \mathbf{H}_1 \mathbf{A} \mathbf{L}_1 \mathbf{L}_2 \dots \mathbf{L}_{n-2}.$$

The complete procedure for the  $7 \times 5$  matrix is shown as follows:

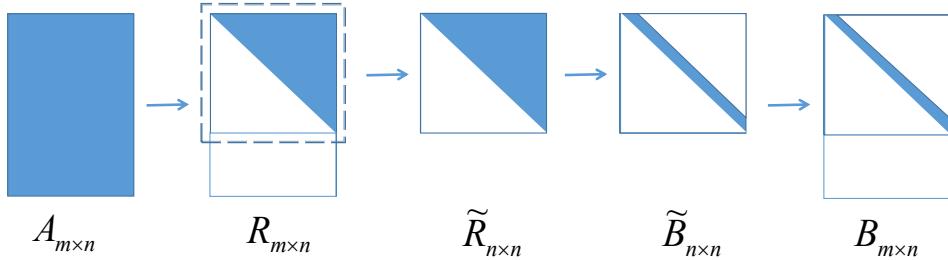
$$\begin{array}{ccc} \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\mathbf{H}_1 \times} & \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\times \mathbf{L}_1^\top} & \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{array} \right] \\ \mathbf{A} & & \mathbf{H}_1 \mathbf{A} & & \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \\ \xrightarrow{\mathbf{H}_2 \times} & & \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{x} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \\ 0 & \mathbf{0} & \boxtimes & \boxtimes & \boxtimes \end{array} \right] & \xrightarrow{\times \mathbf{L}_2^\top} & \left[ \begin{array}{ccccc} \boxtimes & \boxtimes & 0 & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \mathbf{0} & \mathbf{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{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top & & & & \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \mathbf{L}_2^\top \end{array}$$

$$\begin{array}{c}
\begin{array}{ccc}
& \xrightarrow{\mathbf{H}_3 \times} & \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 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \\
& \xrightarrow{\mathbf{L}_3^\top} & \begin{bmatrix} \boxtimes & \boxtimes & 0 & 0 & 0 \\ \boxtimes & \boxtimes & \boxtimes & 0 & 0 \\ 0 & \boxtimes & \boxtimes & \boxtimes & \mathbf{0} \\ 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{bmatrix} \\
& \mathbf{H}_3 \mathbf{H}_2 \mathbf{H}_1 \mathbf{A} \mathbf{L}_1^\top \mathbf{L}_2^\top & \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}{ccc}
& \xrightarrow{\mathbf{H}_4 \times} & \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 & \mathbf{0} & \boxtimes \\ 0 & 0 & 0 & \mathbf{0} & \boxtimes \\ 0 & 0 & 0 & \mathbf{0} & \boxtimes \\ 0 & 0 & 0 & \mathbf{0} & \boxtimes \end{bmatrix} \\
& \xrightarrow{\mathbf{H}_5 \times} & \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 & \mathbf{0} \\ 0 & 0 & 0 & 0 & \boxtimes \\ 0 & 0 & 0 & 0 & \mathbf{0} \\ 0 & 0 & 0 & 0 & \mathbf{0} \end{bmatrix} \\
& \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 & \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}$$

In our implementation, each right Householder reflector  $\mathbf{L}_i$  follows immediately after its corresponding left reflector  $\mathbf{H}_i$ . A common mistake is to apply all the left reflectors first, followed by all the right reflectors, which essentially combines a QR decomposition with a Hessenberg decomposition. However, this method is problematic because applying the right reflector  $\mathbf{L}_1$  after all left reflectors would undo the zeros introduced by the latter. To preserve the structure, the left and right reflectors must be applied in an interleaved fashion to maintain and reinforce the zero patterns.

Although the Golub–Kahan bidiagonalization is effective, it is not the most computationally efficient approach for calculating a bidiagonal decomposition. For an  $m \times n$  matrix with  $m > n$ , the method requires  $\sim 4mn^2 - \frac{4}{3}n^3$  flops to compute a bidiagonal decomposition. Furthermore, if the explicit computation of the orthogonal matrices  $\mathbf{U}$  and  $\mathbf{V}$  is also required, an additional  $\sim 4m^2n - 2mn^2 + 2n^3$  flops are needed (Lu, 2021c).

**LHC Bidiagonalization.** Nevertheless, when  $m \gg n$ , we can extract a square triangular matrix through QR decomposition and then apply the Golub–Kahan bidiagonalization to the resulting  $n \times n$  square triangular matrix. This procedure, known as the *Lawson-Hanson-Chan (LHC) bidiagonalization* (Lawson and Hanson, 1995; Chan, 1982), is illustrated in Figure 8.1. The LHC bidiagonalization begins with the full QR decomposition of  $\mathbf{A}$ , ex-



**Figure 8.1:** Illustration of the LHC bidiagonalization process for a matrix.

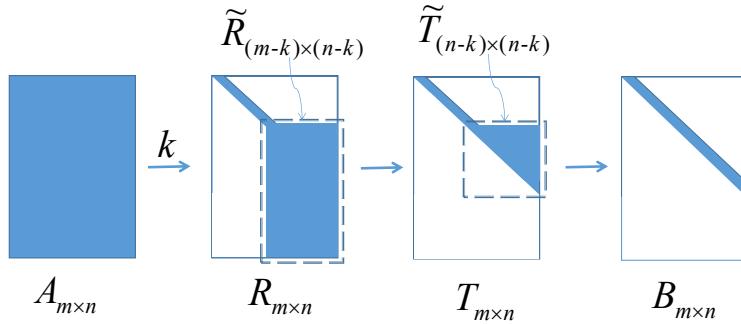
pressed as  $\mathbf{A} = \mathbf{Q}\mathbf{R}$ , where  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  is orthogonal and  $\mathbf{R} \in \mathbb{R}^{m \times n}$  is upper triangular. Next, the Golub–Kahan process is applied to the square  $n \times n$  triangular submatrix  $\tilde{\mathbf{R}}$  within  $\mathbf{R}$ , resulting in  $\tilde{\mathbf{R}} = \tilde{\mathbf{U}}\tilde{\mathbf{B}}\mathbf{V}^\top$ . The matrices  $\tilde{\mathbf{U}}$  and  $\tilde{\mathbf{B}}$  are then appended to form

$$\mathbf{U}_0 = \begin{bmatrix} \tilde{\mathbf{U}} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{m-n} \end{bmatrix} \in \mathbb{R}^{m \times m} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} \tilde{\mathbf{B}} \\ \mathbf{0}_{(m-n) \times n} \end{bmatrix} \in \mathbb{R}^{m \times n},$$

which gives  $\mathbf{R} = \mathbf{U}_0 \mathbf{B} \mathbf{V}^\top$  and  $\mathbf{A} = \mathbf{Q} \mathbf{U}_0 \mathbf{B} \mathbf{V}^\top$ . Let  $\mathbf{U} = \mathbf{Q} \mathbf{U}_0$ , we obtain the desired bidiagonal decomposition of  $\mathbf{A}$ . The computational cost of the QR decomposition is  $\sim 2mn^2 - \frac{2}{3}n^3$  flops, while the Golub–Kahan process applied to the  $n \times n$  submatrix  $\tilde{\mathbf{R}}$  requires  $\sim \frac{8}{3}n^3$  (Lu, 2021c). Therefore, the total computational cost for obtaining the bidiagonal matrix  $\mathbf{B}$  through the LHC bidiagonalization is approximately

$$\text{LHC bidiagonalization: } \sim 2mn^2 + 2n^3 \text{ flops.}$$

The LHC process creates zeros and then destroys them again in the lower triangle of the upper  $n \times n$  square of  $\mathbf{R}$ . However, the zeros in the lower  $(m-n) \times n$  rectangular submatrix of  $\mathbf{R}$  remain unaffected. Consequently, when  $m-n$  is sufficiently large (i.e.,  $m \gg n$ ), this approach achieves a net computational gain. In fact, simple analysis shows that the LHC bidiagonalization becomes more efficient than the standard Golub–Kahan method when  $m > \frac{5}{3}n$ .



**Figure 8.2:** Illustration of the Three-Step bidiagonalization process for a matrix.

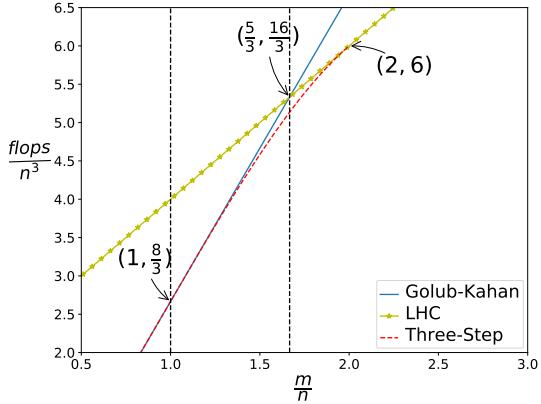
**Three-Step Bidiagonalization.** While the LHC method is advantageous when  $m > \frac{5}{3}n$ , an alternative approach is to apply the QR decomposition at an intermediate stage rather than at the beginning (Trefethen and Bau III, 1997). This modified process—known as the *Three-Step bidiagonalization* and illustrated in Figure 8.2—begins with the application of the first  $k$  steps of left and right Householder reflectors, as in the Golub–Kahan process, while leaving the bottom-right  $(m-k) \times (n-k)$  submatrix “unreflected.” The LHC procedure is then applied to this submatrix to produce the final bidiagonal decomposition. This adjustment reduces computational complexity in cases where  $n < m < 2n$ .

The computational costs of the three bidiagonalization methods are summarized as follows:

$$\begin{cases} \text{Golub–Kahan: } \sim 4mn^2 - \frac{4}{3}n^3 \text{ flops,} \\ \text{LHC: } \sim 2mn^2 + 2n^3 \text{ flops,} \\ \text{Three-Step: } \sim 2mn^2 + 2m^2n - \frac{2}{3}m^3 - \frac{2}{3}n^3 \text{ flops.} \end{cases}$$

When  $m > 2n$ , the LHC method is preferable; when  $n < m < 2n$ , the Three-Step method offers marginal improvements, as shown in Figure 8.3, which plots the operation counts of the three methods as a function of  $\frac{m}{n}$ . Note that the above complexity estimates do not include the cost of computing the orthogonal matrices  $\mathbf{U}$  and  $\mathbf{V}$ . These additional costs are omitted here for simplicity.

**Figure 8.3:** Comparison of the computational complexities of the three bidiagonalization methods. When  $m > 2n$ , the LHC method is preferred; when  $n < m < 2n$ , the Three-Step method is slightly more efficient.



### 8.3. Connection to Tridiagonal Decomposition

To see the connection to tridiagonal decomposition, we begin by illustrating the relationship between tridiagonal and bidiagonal decompositions using the following lemma, which explains how to construct a tridiagonal matrix from a bidiagonal one.

**Lemma 8.3: (Construct tridiagonal from bidiagonal)** Let  $B \in \mathbb{R}^{n \times n}$  be an upper bidiagonal matrix. Then,  $T_1 = B^\top B$  and  $T_2 = BB^\top$  are symmetric tridiagonal matrices.

This lemma reveals a key property: if  $A = UBV^\top$  is the bidiagonal decomposition of  $A$ , then the symmetric matrix  $AA^\top$  admits a tridiagonal decomposition:

$$AA^\top = UBV^\top V B^\top U^\top = UBB^\top U^\top.$$

Similarly, the symmetric matrix  $A^\top A$  also admits a tridiagonal decomposition:

$$A^\top A = VB^\top U^\top UBV^\top = VB^\top BV^\top.$$

As a final result in this section, we present a theorem that provides the tridiagonal decomposition of a symmetric matrix with nonnegative eigenvalues.

**Theorem 8.4: (Tridiagonal decomposition for nonnegative eigenvalues)** Let  $A$  be an  $n \times n$  symmetric matrix with nonnegative eigenvalues. Then, there exists a matrix  $Z$  such that

$$A = ZZ^\top.$$

Furthermore, the tridiagonal decomposition of  $A$  reduces to finding the bidiagonal decomposition of  $Z = UBV^\top$ , such that the tridiagonal decomposition of  $A$  is given by

$$A = ZZ^\top = UBB^\top U^\top.$$

**Proof [of Theorem 8.4]** The eigenvectors of a symmetric matrix can be chosen to be orthogonal (Lemma 10.3), allowing  $A$  to be decomposed as  $A = Q\Lambda Q^\top$  (spectral theorem 10.1), where  $\Lambda$  is a diagonal matrix containing the eigenvalues of  $A$ . When the eigenvalues are nonnegative,  $\Lambda$  can be factored as  $\Lambda = \Lambda^{1/2}\Lambda^{1/2}$ . Setting  $Z = Q\Lambda^{1/2}$ , we obtain  $A = ZZ^\top$ . Combining these results leads to the desired conclusion. ■

## Chapter 8 Problems

1. Prove Lemma 8.3.
2. We discussed the bidiagonalization for a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  with  $m \geq n$  in the main section. Provide an algorithm to compute the bidiagonalization when  $m < n$ , and analyze its computational complexity. Alternatively, discuss the algorithm for computing  $\mathbf{A} = \mathbf{U}\mathbf{B}\mathbf{V}^\top$  with orthogonal  $\mathbf{U}, \mathbf{V}$  and lower bidiagonal  $\mathbf{B}$  when  $m \geq n$ .
3. Prove in detail that the LHC bidiagonalization method is more efficient when  $m > \frac{5}{3}n$  compared to the Golub–Kahan bidiagonalization.
4. Prove in detail that the Three-Step bidiagonalization method is more efficient when  $n \leq m < 2n$  compared to the Golub–Kahan and LHC bidiagonalization methods.
5. (Read Section 11.1 first) Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be upper bidiagonal with a repeated singular value. Show that  $\mathbf{A}$  must have a zero on its diagonal or superdiagonal.
6. **Singular values of bidiagonal** (read Section 11.1 first (Bernstein, 2008; Mathias and Hogben, 2014)). Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be upper bidiagonal with the main diagonal values  $\{a_1, a_2, \dots, a_n\}$  and the superdiagonal values  $\{b_1, b_2, \dots, b_{n-1}\}$ , and let  $\mathbf{B} \in \mathbb{R}^{n \times n}$  be bidiagonal. Show that
  - (a) The singular values of  $\mathbf{A}$  are distinct.
  - (b) If  $|\mathbf{B}| = |\mathbf{A}|$ , where  $|\cdot|$  denotes the element-wise absolute value of a matrix, then  $\mathbf{A}$  and  $\mathbf{B}$  have the same singular values.
  - (c) If  $|\mathbf{A}| \preceq |\mathbf{B}|$  (i.e.,  $|\mathbf{B}| - |\mathbf{A}|$  is PSD) and  $|\mathbf{A}| \neq |\mathbf{B}|$ , then  $\sigma_{\max}(\mathbf{A}) < \sigma_{\max}(\mathbf{B})$ .
  - (d) If  $|\mathbf{I} \circledast \mathbf{A}| \preceq |\mathbf{I} \circledast \mathbf{B}|$  and  $|\mathbf{I} \circledast \mathbf{A}| \neq |\mathbf{I} \circledast \mathbf{B}|$ , where  $\circledast$  denotes the Hadamard product, then  $\sigma_{\min}(\mathbf{A}) < \sigma_{\min}(\mathbf{B})$ .
  - (e) If  $|\mathbf{I}_{\text{up}} \circledast \mathbf{A}| \preceq |\mathbf{I}_{\text{up}} \circledast \mathbf{B}|$ , and  $|\mathbf{I}_{\text{up}} \circledast \mathbf{A}| \neq |\mathbf{I}_{\text{up}} \circledast \mathbf{B}|$ , where  $\mathbf{I}_{\text{up}}$  denotes the matrix with all entries on the superdiagonal equal to 1 and all other entries equal to 0, then  $\sigma_{\min}(\mathbf{B}) < \sigma_{\min}(\mathbf{A})$ .
7. Explore the process of bidiagonalization using Givens rotations. What happens if the matrix is upper triangular or tridiagonal?
8. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be upper bidiagonal with  $a_{nn} = 0$ . Show how to construct orthogonal matrices  $\mathbf{U}$  and  $\mathbf{V}$  (as products of Givens rotations) such that  $\mathbf{U}^\top \mathbf{A} \mathbf{V}$  is upper bidiagonal with the  $n$ -th column being zero.
9. Show that the matrix  $\mathbf{V}$  in the bidiagonal decomposition (Theorem 8.2) has the structure  $\mathbf{V} = \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix}$ , where  $\mathbf{Q} \in \mathbb{R}^{(n-1) \times (n-1)}$  is orthogonal.

# Part V

# Eigenvalue Problem



# 9

## Eigenvalue, Jordan, and Schur Decomposition

### 9.1. Eigenvalue Decomposition

**Theorem 9.1: (Eigenvalue decomposition)** Any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with linearly independent eigenvectors can be decomposed as

$$\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1},$$

where  $\mathbf{X}$  contains the eigenvectors of  $\mathbf{A}$  as its columns, and  $\Lambda$  is a diagonal matrix  $\text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ , with  $\lambda_1, \lambda_2, \dots, \lambda_n$  representing the eigenvalues of  $\mathbf{A}$ .

This decomposition is known as the *eigenvalue decomposition (EVD)*, or sometimes as *diagonalizing* the matrix  $\mathbf{A}$ . If all the eigenvalues of  $\mathbf{A}$  are distinct, then its eigenvectors are guaranteed to be linearly independent, and hence  $\mathbf{A}$  can be diagonalized. Note that without  $n$  linearly independent eigenvectors, diagonalization is not possible. In Section 10.4, we will explore further conditions under which a matrix has linearly independent eigenvectors.

**Proof** [of Theorem 9.1] Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$  be the linearly independent eigenvectors of  $\mathbf{A}$ . Clearly, we have

$$\mathbf{A}\mathbf{x}_1 = \lambda_1\mathbf{x}_1, \quad \mathbf{A}\mathbf{x}_2 = \lambda_2\mathbf{x}_2, \quad \dots, \quad \mathbf{A}\mathbf{x}_n = \lambda_n\mathbf{x}_n.$$

Stacking these equations in matrix form yields:

$$\mathbf{AX} = [\mathbf{Ax}_1, \mathbf{Ax}_2, \dots, \mathbf{Ax}_n] = [\lambda_1\mathbf{x}_1, \lambda_2\mathbf{x}_2, \dots, \lambda_n\mathbf{x}_n] = \mathbf{X}\Lambda.$$

Since the eigenvectors are assumed to be linearly independent, the matrix  $\mathbf{X}$  has full rank and is invertible. Therefore, we obtain  $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$ . This completes the proof. ■

In the spectral decomposition chapter (Chapter 10), we will discuss similar forms of eigenvalue decomposition, where the matrix  $\mathbf{A}$  is required to be symmetric, and  $\mathbf{X}$  is not only nonsingular but also orthogonal. Alternatively,  $\mathbf{A}$  may be a *simple matrix*, meaning that the algebraic and geometric multiplicities of its eigenvalues are equal. In this case,  $\mathbf{X}$  will be a trivial nonsingular matrix. The decomposition also has a geometric interpretation, which we will explore in Section 11.5.

A matrix decomposition of the form  $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$  has a notable property, allowing for efficient computation of the  $m$ -th power of  $\mathbf{A}$ .

**Remark 9.2 ( $m$ -th power).** The  $m$ -th power of  $\mathbf{A}$  is given by  $\mathbf{A}^m = \mathbf{X}\boldsymbol{\Lambda}^m\mathbf{X}^{-1}$  if  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{X}\boldsymbol{\Lambda}\mathbf{X}^{-1}$ . Computing  $\boldsymbol{\Lambda}^m$  is easy because we can apply this operation individually to each diagonal element. Moreover, if  $\mathbf{A} = \mathbf{X}\boldsymbol{\Lambda}\mathbf{X}^{-1}$ , then the eigenvalues of  $\mathbf{A}^m$  are precisely the  $m$ -th powers of the eigenvalues of  $\mathbf{A}$ .

We observe that for the eigenvalue decomposition to exist, the matrix  $\mathbf{A}$  must have a complete set of linearly independent eigenvectors. This condition is naturally satisfied under certain circumstances.

**Lemma 9.3: (Different eigenvalues)** If the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  of a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  are all distinct, then the corresponding eigenvectors are linearly independent. In other words, any square matrix with distinct eigenvalues can be diagonalized.

**Proof** [of Lemma 9.3] Assume that the eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  are distinct, but that the eigenvectors  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$  are linearly dependent. Without loss of generality, assume 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

$$\mathbf{A}\mathbf{x}_n = \mathbf{A} \left( \sum_{i=1}^{n-1} c_i \mathbf{x}_i \right) = c_1 \lambda_1 \mathbf{x}_1 + c_2 \lambda_2 \mathbf{x}_2 + \dots + c_{n-1} \lambda_{n-1} \mathbf{x}_{n-1}.$$

and

$$\mathbf{A}\mathbf{x}_n = \lambda_n (\mathbf{x}_1 + \mathbf{x}_2 + \dots + \mathbf{x}_{n-1}).$$

Equating these two expressions gives:  $\sum_{i=1}^{n-1} (\lambda_n - \lambda_i) c_i \mathbf{x}_i = \mathbf{0}$ . This leads to a contradiction since  $\lambda_n \neq \lambda_i$  for all  $i \in \{1, 2, \dots, n-1\}$ , thus proving that the eigenvectors are linearly independent. ■

There are also several limitations to the eigenvalue decomposition, which will be addressed in the following chapters:

- The eigenvectors in  $\mathbf{X}$  are generally not orthogonal, and there may not be enough eigenvectors (i.e., some eigenvalues are repeated).
- To compute the eigenvalues and eigenvectors,  $\mathbf{A}\mathbf{x} = \lambda\mathbf{x}$ ,  $\mathbf{A}$  must be square. Rectangular matrices cannot be diagonalized using the eigenvalue decomposition.

## 9.2. Jordan Decomposition

In eigenvalue decomposition, we assume that the matrix  $\mathbf{A}$  has  $n$  linearly independent eigenvectors. However, this assumption does not hold for all square matrices. To address this limitation, we introduce a generalized form of eigenvalue decomposition, known as the *Jordan decomposition* or *Jordan canonical form*, named after *Camille Jordan* (Jordan, 1870).

To describe the Jordan decomposition, we begin by defining *Jordan blocks* and the *Jordan form*.

**Definition 9.4 (Jordan block).** An  $m \times m$  upper triangular matrix  $B(\lambda, m)$  is called a *Jordan block* if all its  $m$  diagonal elements are equal to  $\lambda$ , and all superdiagonal elements

are 1. Mathematically,

$$B(\lambda, m) = \begin{bmatrix} \lambda & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \lambda & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & \lambda & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & \lambda \end{bmatrix}_{m \times m}.$$

**Definition 9.5 (Jordan form).** Given an  $n \times n$  matrix  $\mathbf{A}$ , a Jordan form  $\mathbf{J}$  of  $\mathbf{A}$  is a block diagonal matrix of the form:

$$\mathbf{J} = \text{diag}(B(\lambda_1, m_1), B(\lambda_2, m_2), \dots, B(\lambda_k, m_k)),$$

where  $\lambda_1, \lambda_2, \dots, \lambda_k$  are eigenvalues of  $\mathbf{A}$  (with possible repetitions), and  $m_1 + m_2 + \dots + m_k = n$ .

Although not all matrices can be decomposed using eigenvalue decomposition, they can be factored using Jordan decomposition. A non-diagonalizable matrix  $\mathbf{A}$  with multiple eigenvalues can be reduced to its Jordan canonical form through a similarity transformation.

**Theorem 9.6: (Jordan decomposition)** Any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be decomposed 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 its columns, and  $\mathbf{J}$  is a Jordan form matrix represented as  $\text{diag}(\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_k)$ . Each block  $\mathbf{J}_i \in \mathbb{R}^{m_i \times m_i}$  is defined as:

$$\mathbf{J}_i = \begin{bmatrix} \lambda_i & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & \lambda_i & 1 & \dots & 0 & 0 & 0 \\ 0 & 0 & \lambda_i & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_i & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & \lambda_i & 1 \\ 0 & 0 & 0 & \dots & 0 & 0 & \lambda_i \end{bmatrix}_{m_i \times m_i},$$

where  $\lambda_i$  is an eigenvalue of  $\mathbf{A}$ , and  $m_1 + m_2 + \dots + m_k = n$ . These blocks  $\mathbf{J}_i$  are referred to as Jordan blocks. Furthermore, the nonsingular matrix  $\mathbf{X}$  is called the *matrix of generalized eigenvectors* of  $\mathbf{A}$ .

For example, a Jordan form  $\mathbf{J}$  can take the following structure:

$$\mathbf{J} = \text{diag}(B(\lambda_1, m_1), \dots, B(\lambda_k, m_k)) = \begin{bmatrix} \begin{bmatrix} \lambda_1 & 1 & 0 \\ 0 & \lambda_1 & 1 \\ 0 & 0 & \lambda_1 \end{bmatrix} & & & & & \\ & \begin{bmatrix} \lambda_2 & & & & & \\ & \begin{bmatrix} \lambda_3 & 1 \\ 0 & \lambda_3 \end{bmatrix} & & & & \\ & & \ddots & & & \\ & & & \begin{bmatrix} \lambda_k & 1 \\ 0 & \lambda_k \end{bmatrix} & & \end{bmatrix}.$$

Note that zeros can appear on the superdiagonal of  $\mathbf{J}$ , and the first column is always a vector containing only eigenvalues of  $\mathbf{A}$  in each block. Although Jordan decomposition

is theoretically significant, it is rarely used in practice due to its extreme sensitivity to perturbations. Even small random changes to a matrix can render it diagonalizable (van de Geijn and Myers, 2020). As a result, no major mathematical software libraries or tools provide direct support for computing the Jordan decomposition. Additionally, its proof spans dozens of pages and is beyond the scope of this discussion. Interested readers are encouraged to explore the references for further details (Gohberg and Goldberg, 1996; Hales and Passi, 1999; Lu, 2021c).

### 9.3. Schur Decomposition

The eigenvalue decomposition is a special case of the *Schur decomposition*. The latter generalizes the eigenvalue decomposition to all square matrices, even those that are not diagonalizable. It uses an orthogonal similarity transformation to transform an arbitrary square matrix into an upper triangular matrix. This transformation allows many properties of the original matrix to be analyzed using the simpler structure of the upper triangular form.

**Theorem 9.7: (Schur decomposition)** Any **real** square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with **real** eigenvalues can be decomposed as

$$\mathbf{A} = \mathbf{Q}\mathbf{U}\mathbf{Q}^\top,$$

where  $\mathbf{Q}$  is a (real) orthogonal matrix, and  $\mathbf{U}$  is a (real) upper triangular matrix. In other words, any real square matrix  $\mathbf{A}$  with real eigenvalues can be *triangularized*.

The first columns of  $\mathbf{AQ}$  and  $\mathbf{QU}$  are given by  $\mathbf{Aq}_1$  and  $u_{11}\mathbf{q}_1$ , respectively. Consequently,  $u_{11}$  corresponds to an eigenvalue of  $\mathbf{A}$ , while  $\mathbf{q}_1$  serves as its associated eigenvector. However, the remaining columns of  $\mathbf{Q}$  are not necessarily eigenvectors of  $\mathbf{A}$ .

**Schur decomposition for symmetric matrices.** For a symmetric matrix  $\mathbf{A} = \mathbf{A}^\top$ , the relation  $\mathbf{QUQ}^\top = \mathbf{QU}^\top\mathbf{Q}^\top$  holds. In this case,  $\mathbf{U}$  must be diagonal, and this diagonal matrix contains the eigenvalues of  $\mathbf{A}$ . Furthermore, all columns of  $\mathbf{Q}$  are eigenvectors of  $\mathbf{A}$ . Thus, we conclude that all symmetric matrices are diagonalizable, even in the presence of repeated eigenvalues; see Chapter 10 for more discussions.

To validate Theorem 9.7, we rely on the following lemmas.

**Lemma 9.8: (Determinant intermezzo)** The determinant of a matrix (Definition 0.21) satisfies the following properties:

- The determinant of the product of two matrices is given by  $\det(\mathbf{AB}) = \det(\mathbf{A})\det(\mathbf{B})$ ;
- The determinant of the transpose of a matrix is the same as the determinant of the original matrix:  $\det(\mathbf{A}^\top) = \det(\mathbf{A})$ ;
- If matrix  $\mathbf{A}$  has an eigenvalue  $\lambda$ , then  $\det(\mathbf{A} - \lambda\mathbf{I}) = 0$ ;
- The determinant of an identity matrix is 1;
- For an orthogonal matrix  $\mathbf{Q}$ , the determinant satisfies:

$$\det(\mathbf{Q}) = \det(\mathbf{Q}^\top) = \pm 1, \quad \text{since } \det(\mathbf{Q}^\top)\det(\mathbf{Q}) = \det(\mathbf{Q}^\top\mathbf{Q}) = \det(\mathbf{I}) = 1;$$

- For any square matrix  $\mathbf{A}$  and an orthogonal matrix  $\mathbf{Q}$ , the determinant relation holds:

$$\det(\mathbf{A}) = \det(\mathbf{Q}^\top)\det(\mathbf{A})\det(\mathbf{Q}) = \det(\mathbf{Q}^\top\mathbf{A}\mathbf{Q});$$

- For a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , the determinant of  $-\mathbf{A}$  is given by  $\det(-\mathbf{A}) = (-1)^n \det(\mathbf{A})$ .

**Lemma 9.9: (Submatrix with same eigenvalue)** Let  $\mathbf{A}_{k+1} \in \mathbb{R}^{(k+1) \times (k+1)}$  be a square matrix with real eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_{k+1}$ . Then, we can construct a  $k \times k$  matrix  $\mathbf{A}_k$  with eigenvalues  $\lambda_2, \lambda_3, \dots, \lambda_{k+1}$  as follows:

$$\mathbf{A}_k = \begin{bmatrix} -\mathbf{p}_2^\top & - \\ -\mathbf{p}_3^\top & - \\ \vdots & \\ -\mathbf{p}_{k+1}^\top & - \end{bmatrix} \mathbf{A}_{k+1} \begin{bmatrix} \mathbf{p}_2 & \mathbf{p}_3 & \dots & \mathbf{p}_{k+1} \end{bmatrix},$$

where  $\mathbf{p}_1$  is a unit-norm eigenvector of  $\mathbf{A}_{k+1}$  corresponding to the eigenvalue  $\lambda_1$ , and  $\mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_{k+1}$  denote any mutually orthonormal vectors orthogonal to  $\mathbf{p}_1$ , i.e.,  $\mathbf{p}_1 \in \mathcal{C}^\perp([\mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_{k+1}])$ .

**Proof** [of Lemma 9.9] Let  $\mathbf{P}_{k+1} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{k+1}]$ . It follows that  $\mathbf{P}_{k+1}^\top \mathbf{P}_{k+1} = \mathbf{I}$ , 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  $\lambda \in \{\lambda_2, \lambda_3, \dots, \lambda_{k+1}\}$ , by Lemma 9.8, we have

$$\begin{aligned} \det(\mathbf{A}_{k+1} - \lambda \mathbf{I}) &= \det(\mathbf{P}_{k+1}^\top (\mathbf{A}_{k+1} - \lambda \mathbf{I}) \mathbf{P}_{k+1}) = \det(\mathbf{P}_{k+1}^\top \mathbf{A}_{k+1} \mathbf{P}_{k+1} - \lambda \mathbf{P}_{k+1}^\top \mathbf{P}_{k+1}) \\ &= \det \left( \begin{bmatrix} \lambda_1 - \lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k - \lambda \mathbf{I} \end{bmatrix} \right) = (\lambda_1 - \lambda) \det(\mathbf{A}_k - \lambda \mathbf{I}). \end{aligned}$$

Since  $\lambda$  is an eigenvalue of  $\mathbf{A}$  and  $\lambda \neq \lambda_1$ , it follows that  $\det(\mathbf{A}_{k+1} - \lambda \mathbf{I}) = (\lambda_1 - \lambda) \det(\mathbf{A}_k - \lambda \mathbf{I}) = 0$ , which implies that  $\lambda$  is also an eigenvalue of  $\mathbf{A}_k$ . ■

We now establish the existence of the Schur decomposition using an inductive proof.

**Proof** [of Theorem 9.7: Existence of Schur decomposition] We begin by noting that the theorem is trivial when  $n = 1$ , as we can simply set  $Q = 1$  and  $U = A$ . Now, suppose the theorem holds true for  $n = k$  for some  $k > 1$ . To complete the proof, we must show that the theorem also holds for  $n = k + 1$ . Assume for  $n = k$ , the theorem is valid, i.e., any matrix  $\mathbf{A}_k \in \mathbb{R}^{k \times k}$  can be expressed as  $\mathbf{A}_k = \mathbf{Q}_k \mathbf{U}_k \mathbf{Q}_k^\top$ , where  $\mathbf{Q}_k$  is orthogonal, and  $\mathbf{U}_k$  is upper triangular.

For  $n = k + 1$ , let  $\mathbf{A}_{k+1}$  be a matrix with eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_{k+1}$ . Using Lemma 9.9, construct an orthogonal matrix  $\mathbf{P}_{k+1} = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{k+1}]$ , where  $\mathbf{p}_1$  is a unit-norm eigenvector of  $\mathbf{A}_{k+1}$  corresponding to the eigenvalue  $\lambda_1$ , and  $\mathbf{p}_2, \dots, \mathbf{p}_{k+1}$  are mutually orthonormal vectors orthogonal to  $\mathbf{p}_1$ . Since we assume the theorem is true for  $n = k$ , we can find a matrix  $\mathbf{A}_k \in \mathbb{R}^{k \times k}$  with eigenvalues  $\lambda_2, \lambda_3, \dots, \lambda_{k+1}$ , satisfying  $\mathbf{A}_k = \mathbf{Q}_k \mathbf{U}_k \mathbf{Q}_k^\top$ . By Lemma 9.9, the following properties hold:

$$\mathbf{P}_{k+1}^\top \mathbf{A}_{k+1} \mathbf{P}_{k+1} = \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \end{bmatrix} \quad \Rightarrow \quad \mathbf{A}_{k+1} \mathbf{P}_{k+1} = \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \end{bmatrix}.$$

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

$$\begin{aligned} \mathbf{A}_{k+1} \mathbf{Q}_{k+1} &= \mathbf{A}_{k+1} \mathbf{P}_{k+1} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} = \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \end{bmatrix} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} = \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_k \mathbf{Q}_k \end{bmatrix} \\ &= \mathbf{P}_{k+1} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \mathbf{U}_k \end{bmatrix} = \mathbf{P}_{k+1} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{Q}_k \end{bmatrix} \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_k \end{bmatrix} = \mathbf{Q}_{k+1} \mathbf{U}_{k+1}, \end{aligned}$$

where we let  $\mathbf{U}_{k+1} = \begin{bmatrix} \lambda_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_k \end{bmatrix}$ . Therefore,  $\mathbf{A}_{k+1} = \mathbf{Q}_{k+1}\mathbf{U}_{k+1}\mathbf{Q}_{k+1}^\top$ , 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. This completes the inductive step and proves the existence of the Schur decomposition.  $\blacksquare$

## 9.4. Other Forms of Schur Decomposition

In the proof of the Schur decomposition, the upper triangular matrix  $\mathbf{U}_{k+1}$  is constructed by appending the eigenvalue  $\lambda_1$  to  $\mathbf{U}_k$ . This ensures that the diagonal elements consistently represent the eigenvalues of the underlying matrix. Consequently, the upper triangular matrix can be decomposed into two distinct components.

**Corollary 9.10: (Form 2 of Schur decomposition)** Any real matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with real eigenvalues can be decomposed 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 in which all diagonal and lower-triangular entries are zero. Another way to understand this decomposition is by noting that  $\mathbf{A}$  and  $\mathbf{U}$  (where  $\mathbf{U} = \mathbf{Q}^\top \mathbf{A} \mathbf{Q}$ ) are similar matrices, and therefore share the same eigenvalues (Proposition 7.4). Moreover, the eigenvalues of any upper triangular matrices are located on its diagonal. To see this, consider 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 standard basis vector in  $\mathbb{R}^n$ . Thus, we can decompose  $\mathbf{U}$  into the sum of  $\mathbf{\Lambda}$  and  $\mathbf{T}$ .

**Remark 9.11 (m-th power).** The above observation also implies that the eigenvalues of the  $m$ -th power  $\mathbf{A}^m$  are simply the  $m$ -th powers of the eigenvalues of  $\mathbf{A}$ .

A final observation about the second form of the Schur decomposition is as follows. From the equation  $\mathbf{A}\mathbf{Q} = \mathbf{Q}(\mathbf{\Lambda} + \mathbf{T})$ , it follows that  $\mathbf{A}\mathbf{q}_k = \lambda_k\mathbf{q}_k + \sum_{i=1}^{k-1} t_{ik}\mathbf{q}_i$ , where  $t_{ik}$  is the  $(i, k)$ -th entry of  $\mathbf{T}$ . The form is quite similar to the eigenvalue decomposition. However, instead of being eigenvectors, the columns of  $\mathbf{Q}$  form an orthonormal basis that is interrelated.

In the main result of Theorem 9.7, we focus on real matrices with real eigenvalues. However, this restriction may not always be practical in various applications. A more general version is presented in the following theorem. This decomposition is attributed to *Issai Schur* (1875–1941), a Russian mathematician who spent most of his professional life in Germany.

**Theorem 9.12: (Complex Schur decomposition)** Let  $\mathbf{A} \in \mathbb{C}^{n \times n}$  be any **complex** square matrix. Then, it can be decomposed as

$$\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{U}^*,$$

where  $\mathbf{U} \in \mathbb{C}^{n \times n}$  is a unitary matrix, and  $\mathbf{T} \in \mathbb{C}^{n \times n}$  is an upper triangular matrix (not necessarily real).

**Proof** See Lu (2021c). ■

## 9.5. Application: Computing Fibonacci Numbers

Eigenvalue decomposition offers a powerful method for computing Fibonacci numbers (Strang, 2009). The Fibonacci sequence is defined recursively: each term  $F_{k+2}$  is the sum of the two preceding terms,  $F_{k+1} + F_k$ . The sequence begins as  $0, 1, 1, 2, 3, 5, 8, \dots$ . A natural question arises: What is the value of  $F_{100}$ ? Eigenvalue decomposition allows us to derive a general formula for the Fibonacci sequence.

Let  $\mathbf{u}_k = \begin{bmatrix} F_{k+1} \\ F_k \end{bmatrix}$ . By the definition of the Fibonacci sequence, we have  $\mathbf{u}_{k+1} = \begin{bmatrix} F_{k+2} \\ F_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \mathbf{u}_k$ . Define  $\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ . It follows that  $\mathbf{u}_{100} = \mathbf{A}^{100} \mathbf{u}_0$ , where  $\mathbf{u}_0 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ .

The eigenvalues of  $\mathbf{A}$  are found by solving  $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$ , where  $\lambda$  is an eigenvalue of  $\mathbf{A}$  (Remark 0.25). Solving the characteristic equation  $\det(\mathbf{A} - \lambda \mathbf{I}) = \lambda^2 - \lambda + 1 = 0$ , we obtain the eigenvalues and their corresponding eigenvectors:

$$(\lambda_1, \mathbf{x}_1) = \left( \frac{1 + \sqrt{5}}{2}, \begin{bmatrix} \lambda_1 \\ 1 \end{bmatrix} \right) \quad \text{and} \quad (\lambda_2, \mathbf{x}_2) = \left( \frac{1 - \sqrt{5}}{2}, \begin{bmatrix} \lambda_2 \\ 1 \end{bmatrix} \right).$$

As per Remark 9.2, we can express  $\mathbf{A}^{100} = \mathbf{X} \boldsymbol{\Lambda}^{100} \mathbf{X}^{-1} = \mathbf{X} \begin{bmatrix} \lambda_1^{100} & 0 \\ 0 & \lambda_2^{100} \end{bmatrix} \mathbf{X}^{-1}$ , 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  $\mathbf{u}_{100} = \mathbf{A}^{100} \mathbf{u}_0$  corresponds to the first column of  $\mathbf{A}^{100}$ , which can be represented as:

$$\mathbf{u}_{100} = \begin{bmatrix} F_{101} \\ F_{100} \end{bmatrix} = \begin{bmatrix} \frac{\lambda_1^{101} - \lambda_2^{101}}{\lambda_1 - \lambda_2} \\ \frac{\lambda_1^{100} - \lambda_2^{100}}{\lambda_1 - \lambda_2} \end{bmatrix}.$$

Upon a simple check of the calculation, we have  $F_{100} = 3.542248481792631e + 20$ . Or more generally, we can express  $\mathbf{u}_K$  as follows:

$$\mathbf{u}_K = \begin{bmatrix} F_{K+1} \\ F_K \end{bmatrix} = \begin{bmatrix} \frac{\lambda_1^{K+1} - \lambda_2^{K+1}}{\lambda_1 - \lambda_2} \\ \frac{\lambda_1^K - \lambda_2^K}{\lambda_1 - \lambda_2} \end{bmatrix},$$

where the general form of  $F_K$  is given by  $F_K = \frac{\lambda_1^K - \lambda_2^K}{\lambda_1 - \lambda_2}$ .

## 9.6. Application: Matrix Polynomials

We previously demonstrated in Problems 7.9 and 7.10 that  $f(\mathbf{P}\mathbf{A}\mathbf{P}^{-1}) = \mathbf{P}f(\mathbf{A})\mathbf{P}^{-1}$  if  $f(\mathbf{C}) = \gamma_m \mathbf{C}^m + \gamma_{m-1} \mathbf{C}^{m-1} + \dots + \gamma_0$  is a polynomial. Let  $\mathbf{A} = \mathbf{X}\mathbf{J}\mathbf{X}^{-1}$  be the

Jordan decomposition of  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , where  $\mathbf{X} \in \mathbb{R}^{n \times n}$  is a nonsingular matrix containing the *generalized eigenvectors* of  $\mathbf{A}$  as its columns, and  $\mathbf{J} \in \mathbb{R}^{n \times n}$  is a Jordan form matrix  $\text{diag}(\mathbf{J}_{m_1}(\lambda_1), \mathbf{J}_{m_2}(\lambda_2), \dots, \mathbf{J}_{m_k}(\lambda_k)) = \text{diag}(\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_k)$ , where  $\mathbf{J}_{m_i}(\lambda_i) \in \mathbb{R}^{m_i \times m_i}$  and  $\sum_{i=1}^k m_i = n$ . Using this decomposition, we have:

$$f(\mathbf{A}) = \mathbf{X} f(\mathbf{J}) \mathbf{X}^{-1} = \mathbf{X} \text{diag}(f(\mathbf{J}_1), f(\mathbf{J}_2), \dots, f(\mathbf{J}_k)) \mathbf{X}^{-1}, \quad (9.1)$$

where

$$f(\mathbf{J}_i) = \begin{bmatrix} f(\lambda_i) & f'(\lambda_i) & \frac{1}{2!} f''(\lambda_i) & \dots & \frac{1}{(m_i-1)!} f^{(m_i-1)}(\lambda_i) \\ 0 & f(\lambda_i) & f'(\lambda_i) & \dots & \frac{1}{(m_i-2)!} f^{(m_i-2)}(\lambda_i) \\ 0 & 0 & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & f(\lambda_i) & f'(\lambda_i) \\ 0 & 0 & 0 & \dots & f(\lambda_i) \end{bmatrix}, \quad (9.2)$$

and  $f^{(k)}(x)$  denotes the  $k$ -th derivative of  $f(x)$ . This representation allows us to extend the concept of matrix functions to many common functions that can also be expressed as power series (Zhang, 2017). Below are several important examples:

**Powers of a matrix.**  $\mathbf{A}^m = \mathbf{X} \mathbf{J}^m \mathbf{X}^{-1} = \mathbf{X} f(\mathbf{J}) \mathbf{X}^{-1}$ , where  $f(x) = x^m$  for all  $m = 1, 2, \dots$

**Matrix logarithm.** Let  $f(x) = \ln(1 + x)$ . Then,

$$\ln(\mathbf{I} + \mathbf{A}) = \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{i} \mathbf{A}^i = \mathbf{X} \left( \sum_{i=1}^{\infty} \frac{(-1)^{i-1}}{i} \mathbf{J}^i \right) \mathbf{X}^{-1} = \mathbf{X} f(\mathbf{J}) \mathbf{X}^{-1}. \quad (9.3)$$

**Sine and cosine functions.** Let  $f_1(x) = \sin(x)$  and  $f_2(x) = \cos(x)$ . Then,

$$\sin(\mathbf{A}) = \sum_{i=0}^{\infty} \frac{(-1)^i}{(2i+1)!} \mathbf{A}^{2i+1} = \mathbf{X} \left( \sum_{i=0}^{\infty} \frac{(-1)^i}{(2i+1)!} \mathbf{J}^{2i+1} \right) \mathbf{X}^{-1} = \mathbf{X} f_1(\mathbf{J}) \mathbf{X}^{-1}; \quad (9.4)$$

$$\cos(\mathbf{A}) = \sum_{i=0}^{\infty} \frac{(-1)^i}{(2i)!} \mathbf{A}^{2i} = \mathbf{X} \left( \sum_{i=0}^{\infty} \frac{(-1)^i}{(2i)!} \mathbf{J}^{2i} \right) \mathbf{X}^{-1} = \mathbf{X} f_2(\mathbf{J}) \mathbf{X}^{-1}. \quad (9.5)$$

**Matrix exponentials.** Let  $f_1(x) = e^x$  and  $f_2(x) = e^{-x}$ . Then,

$$e^{\mathbf{A}} = \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{A}^i = \mathbf{X} \left( \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{J}^i \right) \mathbf{X}^{-1} = \mathbf{X} f_1(\mathbf{J}) \mathbf{X}^{-1}; \quad (9.6)$$

$$e^{-\mathbf{A}} = \sum_{i=0}^{\infty} \frac{1}{i!} (-1)^i \mathbf{A}^i = \mathbf{X} \left( \sum_{i=0}^{\infty} \frac{1}{i!} (-1)^i \mathbf{J}^i \right) \mathbf{X}^{-1} = \mathbf{X} f_2(\mathbf{J}) \mathbf{X}^{-1}. \quad (9.7)$$

**Matrix exponential functions.** Let  $f_1(x) = e^{xt}$  and  $f_2(x) = e^{-xt}$ . Then,

$$e^{\mathbf{A}t} = \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{A}^i t^i = \mathbf{X} \left( \sum_{i=0}^{\infty} \frac{1}{i!} \mathbf{J}^i t^i \right) \mathbf{X}^{-1} = \mathbf{X} f_1(\mathbf{J}) \mathbf{X}^{-1}; \quad (9.8)$$

$$e^{-\mathbf{A}t} = \sum_{i=0}^{\infty} \frac{1}{i!} (-1)^i \mathbf{A}^i t^i = \mathbf{X} \left( \sum_{i=0}^{\infty} \frac{1}{i!} (-1)^i \mathbf{J}^i t^i \right) \mathbf{X}^{-1} = \mathbf{X} f_2(\mathbf{J}) \mathbf{X}^{-1}. \quad (9.9)$$

## 9.7. Applications and Properties of Schur Decomposition

A few results can be easily proved using the Schur decomposition. For example, to prove the existence of the spectral decomposition (Theorem 10.1), to prove the trace of a matrix is equal to the sum of eigenvalues, to prove the existence of the block-diagonalization, and to prove the Schur inequality (Lu, 2021c). In this section, we present additional results derived from the Schur decomposition.

### Cayley–Hamilton Theorem

We now provide a rigorous proof of the *Cayley–Hamilton theorem*.

**Theorem 9.13: (Cayley–Hamilton Theorem)** A matrix satisfies its own characteristic equation. That is, given a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , it holds that  $p_{\mathbf{A}}(\lambda) = \det(\lambda\mathbf{I} - \mathbf{A}) = \prod_{i=1}^n (\lambda - \lambda_i)$  and  $p_{\mathbf{A}}(\lambda) = 0$  if  $\lambda$  is an eigenvalue of  $\mathbf{A}$ . Then,  $\mathbf{A}$  also satisfies this characteristic equation:  $p_{\mathbf{A}}(\mathbf{A}) = \prod_{i=1}^n (\mathbf{A} - \lambda_i \mathbf{I}) = \mathbf{0}$ .

**Proof** [of Theorem 9.13] Suppose  $\mathbf{A}$  admits the Schur decomposition  $\mathbf{A} = \mathbf{Q}\mathbf{U}\mathbf{Q}^\top$ . Then,

$$p_{\mathbf{A}}(\mathbf{A}) = \prod_{i=1}^n (\mathbf{Q}\mathbf{U}\mathbf{Q}^\top - \lambda_i \mathbf{I}) = \mathbf{Q} \cdot p_{\mathbf{A}}(\mathbf{U}) \cdot \mathbf{Q}^\top.$$

Therefore, it suffices to show that  $p_{\mathbf{A}}(\mathbf{U}) = \prod_{i=1}^n (\mathbf{U} - \lambda_i \mathbf{I}) = \mathbf{0}$ . We observe that the upper left 2-by-2 block of  $(\mathbf{U} - \lambda_1 \mathbf{I})(\mathbf{U} - \lambda_2 \mathbf{I})$  is zero. This again invokes the upper left 3-by-3 block  $(\mathbf{U} - \lambda_1 \mathbf{I})(\mathbf{U} - \lambda_2 \mathbf{I})(\mathbf{U} - \lambda_3 \mathbf{I})$  to be zero. Continuing this process, the result follows. ■

### Computation of Inverses

We have shown in Remark 9.2 that the eigenvalue decomposition can help identify the  $m$ -th power of a matrix easily. The Cayley–Hamilton theorem can be used to express the  $m$ -th power of a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  as a linear combination of  $\mathbf{I}, \mathbf{A}, \mathbf{A}^2, \dots, \mathbf{A}^{m-1}$ , i.e., as a linear combination of lower power values. Let the characteristic polynomial of  $\mathbf{A}$  be given by  $p_{\mathbf{A}}(\lambda) = \det(\lambda\mathbf{I} - \mathbf{A}) = \lambda^n + \gamma_{n-1}\lambda^{n-1} + \dots + \gamma_1\lambda + \gamma_0$ . Then,

$$\mathbf{A}^n = -\gamma_{n-1}\mathbf{A}^{n-1} - \dots - \gamma_1\mathbf{A} - \gamma_0\mathbf{I}. \quad (9.10)$$

This also implies that

$$\mathbf{I} = -\frac{1}{\gamma_0}(\mathbf{A}^{n-1} + \gamma_{n-1}\mathbf{A}^{n-2} + \dots + \gamma_1)\mathbf{A}. \quad (9.11)$$

If  $\mathbf{A}$  is nonsingular, then multiplying (9.11) by  $\mathbf{A}^{-1}$  yields

$$\mathbf{A}^{-1} = -\left(\frac{1}{\gamma_0}\mathbf{A}^{n-1} + \frac{\gamma_{n-1}}{\gamma_0}\mathbf{A}^{n-2} + \dots + \frac{\gamma_1}{\gamma_0}\mathbf{I}\right).$$

That is, the inverse of an invertible  $n \times n$  matrix  $\mathbf{A}$  can be expressed as a polynomial of  $\mathbf{A}$  of degree at most  $(n - 1)$ .

### Sylvester's Theorem

Given  $\mathbf{A}, \mathbf{X} \in \mathbb{R}^{n \times n}$ , matrices  $\mathbf{A}$  and  $\mathbf{X}$  are said to *commute* if  $\mathbf{AX} = \mathbf{XA}$ . More generally, consider the equation  $\mathbf{AX} = \mathbf{XB}$ , where  $\mathbf{A} \in \mathbb{R}^{n \times n}$ ,  $\mathbf{B} \in \mathbb{R}^{m \times m}$ , and  $\mathbf{X} \in \mathbb{R}^{n \times m}$ . The Cayley–Hamilton theorem indicates (see Problem 9.20):

$$p(\mathbf{A})\mathbf{X} = \mathbf{X}p(\mathbf{B}), \quad \text{for any polynomial } p(\lambda). \quad (9.12)$$

This relationship leads to Sylvester's Theorem.

**Theorem 9.14: (Sylvester's theorem)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and  $\mathbf{B} \in \mathbb{R}^{m \times m}$ .

- If  $\Lambda(\mathbf{A}) \cup \Lambda(\mathbf{B}) = \emptyset$  (i.e., the intersection of the spectrum sets is empty), the equation  $\mathbf{AX} - \mathbf{XB} = \mathbf{0}$  is satisfied only when  $\mathbf{X} = \mathbf{0} \in \mathbb{R}^{n \times m}$ .
- More generally, *Sylvester's equation*  $\mathbf{AX} - \mathbf{XB} = \mathbf{C}$  has a unique solution  $\mathbf{X} \in \mathbb{R}^{n \times m}$  for each  $\mathbf{C} \in \mathbb{R}^{n \times m}$  if and only if  $\Lambda(\mathbf{A}) \cup \Lambda(\mathbf{B}) = \emptyset$ . <sup>a</sup>

a. If  $\mathbf{A}$  and  $\mathbf{B}$  are complex, then there is a unique complex solution  $\mathbf{X}$  for each  $\mathbf{C} \in \mathbb{C}^{n \times m}$ .

**Proof** [of Theorem 9.14] The second part is a direct result of the first part; so we only prove the first part. For the first part, it suffices to show that  $p_{\mathbf{B}}(\mathbf{A})\mathbf{X} = \mathbf{X}p_{\mathbf{B}}(\mathbf{B}) = \mathbf{0}$  due to (9.12). Suppose  $\mathbf{B}$  has eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n$  and admits the characteristic polynomial  $p_{\mathbf{B}}(\lambda) = \prod_{i=1}^n (\lambda - \lambda_i)$  and  $p_{\mathbf{B}}(\mathbf{A}) = \prod_{i=1}^n (\mathbf{A} - \lambda_i \mathbf{I})$ . If  $\Lambda(\mathbf{A}) \cup \Lambda(\mathbf{B}) = \emptyset$ , then each component  $(\mathbf{A} - \lambda_i \mathbf{I})$  is nonsingular, and  $p_{\mathbf{B}}(\mathbf{A})$  is nonsingular. Therefore,  $p_{\mathbf{B}}(\mathbf{A})\mathbf{X} = \mathbf{0}$  if and only if  $\mathbf{X} = \mathbf{0}$ . Conversely, if  $p_{\mathbf{B}}(\mathbf{A})\mathbf{X} = \mathbf{0}$  has a nontrivial solution, then at least one component  $(\mathbf{A} - \lambda_i \mathbf{I})$  must be singular. Thus,  $\Lambda(\mathbf{A}) \cup \Lambda(\mathbf{B}) \neq \emptyset$ . ■

The existence of the Schur decomposition reveals the eigenvalues of  $\mathbf{B}^{-1}\mathbf{A}$  (when  $\mathbf{B}$  is nonsingular) from the upper triangular matrices.

**Corollary 9.15: (Eigenvalues from Schur)** Suppose  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  admit decompositions  $\mathbf{A} = \mathbf{QT}_A\mathbf{V}^\top$  and  $\mathbf{B} = \mathbf{QT}_B\mathbf{V}^\top$ , respectively, where  $\mathbf{Q}, \mathbf{V}$  are orthogonal and  $\mathbf{T}_A, \mathbf{T}_B$  are upper triangular. Then, the diagonal elements of  $\mathbf{T}_B^{-1}\mathbf{T}_A$  are the eigenvalues of  $\mathbf{B}^{-1}\mathbf{A}$  (we assume all the eigenvalues are real).

**Proof** [of Corollary 9.15] The proof relies on the Schur decomposition (Theorem 9.7), and we assume all the eigenvalues discussed are real for simplicity. Suppose  $\mathbf{B}^{-1}\mathbf{A}$  admits a Schur decomposition  $\mathbf{B}^{-1}\mathbf{A} = \mathbf{V}\mathbf{U}\mathbf{V}^\top$  ( $\mathbf{V}$  is orthogonal,  $\mathbf{U}$  is upper triangular)<sup>1</sup>, and  $\mathbf{BV}$  admits a QR decomposition  $\mathbf{BV} = \mathbf{QT}_B \implies \mathbf{B} = \mathbf{QT}_B\mathbf{V}^\top$  ( $\mathbf{Q}$  is orthogonal,  $\mathbf{T}_B$  is upper triangular). Then,  $\mathbf{A} = \mathbf{BV}\mathbf{U}\mathbf{V}^\top = \mathbf{Q} \underbrace{(\mathbf{T}_B\mathbf{U})}_{=\mathbf{T}_A} \mathbf{V}^\top$ , where  $\mathbf{T}_A = \mathbf{T}_B\mathbf{U}$  is upper triangular.

This completes the proof. ■

**Exercise 9.16.** Discuss the connection between the decompositions in Corollary 9.15 and the UTV decomposition (Section 4.1).

## Chapter 9 Problems

1. In the corollary, if we don't assume real eigenvalues, then  $\mathbf{U}$  can be upper quasi-triangular. And  $\mathbf{T}_A$  shown below is also upper quasi-triangular.

1. Show that if a matrix  $\mathbf{A}$  satisfies  $\mathbf{A}^2 = 4\mathbf{I}$ , then all eigenvalues of  $\mathbf{A}$  are 2 and -2.
2. Given a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  where all entries are equal to 1, find the  $n$  eigenvalues of  $\mathbf{A}$ .
3. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be an idempotent matrix (i.e.,  $\mathbf{A}^2 = \mathbf{A}$ ). Show that the matrices  $\mathbf{BA}$  and  $\mathbf{ABA}$  share the same eigenvalues.
4. Consider a Householder transformation matrix  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top \in \mathbb{R}^{n \times n}$ , where  $\|\mathbf{u}\| = 1$ . Show that  $\mathbf{u}$  is an eigenvector of  $\mathbf{H}$  and determine its corresponding eigenvalue. Provide a geometric interpretation of the eigenvalues of  $\mathbf{H}$ . Suppose further that  $\mathbf{v}^\top \mathbf{u} = 0$ , where  $\mathbf{v}$  is a nonzero vector. Show that  $\mathbf{v}$  is also an eigenvector of  $\mathbf{H}$  and find its corresponding eigenvalue.
5. Let  $\lambda$  be an eigenvalue of  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Show that  $\lambda - \mu$  is an eigenvalue of  $\mathbf{A} - \mu\mathbf{I}$ .
6. Let  $\lambda$  be an eigenvalue of a nonsingular matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Show that  $\lambda^{-1}$  is an eigenvalue of  $\mathbf{A}^{-1}$ .
7. Derive the general formula for  $\mathbf{u}_K = \mathbf{A}^K \mathbf{u}_0$ , where  $\mathbf{A}$  is a general  $2 \times 2$  matrix.
8. Consider the matrix  $\mathbf{A} = \mathbf{1}\mathbf{1}^\top \in \mathbb{R}^{n \times n}$ , where all entries are equal to 1, and  $\mathbf{1} \in \mathbb{R}^n$  is the vector of all ones. Find  $n$  linearly independent eigenvectors of  $\mathbf{A}$ , and determine the corresponding eigenvalues. Hint: Consider  $\mathbf{x}_i = \mathbf{1} - n\mathbf{e}_i$  and  $\mathbf{1}$ .
9. What are the eigenvalues of the matrix  $\mathbf{A} = \begin{bmatrix} 5 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & 5 \end{bmatrix}$ ?
10. **Generalized eigenproblem.** Many scientific packages address the generalized eigenproblem  $\mathbf{Ax} = \lambda \mathbf{Bx}$ , where  $\mathbf{B}$  is nonsingular. If  $\mathbf{A}$  is symmetric and  $\mathbf{B}$  is PD with the Cholesky decomposition  $\mathbf{B} = \mathbf{R}^\top \mathbf{R}$ , show that the eigenvalue  $\lambda$  is a (standard) eigenvalue of  $\mathbf{C} = (\mathbf{R}^{-1})^\top \mathbf{A} \mathbf{R}^{-1}$ , corresponding to the eigenvector  $\mathbf{Rx}$ .
11. **Generalized eigenproblem (Teukolsky et al., 1992).** Suppose  $\mathbf{A}\lambda^2 + \mathbf{B}\lambda + \mathbf{C} = \mathbf{0}$ . Show that  $\lambda$  can be solved by a standard eigenproblem. Hint: Let  $\mathbf{y} = \lambda\mathbf{x}$ , and consider the matrix  $\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{A}^{-1}\mathbf{C} & -\mathbf{A}^{-1}\mathbf{B} \end{bmatrix}$ .
12. **Matrix exponentials.** Given the definition of matrix exponentials in (9.6), let  $\mathbf{A}$  and  $\mathbf{B}$  commute, i.e.,  $\mathbf{AB} = \mathbf{BA}$ . Show that  $e^{\mathbf{A}+\mathbf{B}} = e^{\mathbf{A}} \cdot e^{\mathbf{B}}$ .
13. **Matrix exponentials.** Let  $f(t) : \mathbb{R} \rightarrow \mathbb{R}^n$  be a function satisfying  $f'(t) = \mathbf{Af}(t)$  with  $f(0) = \mathbf{x}_0 \in \mathbb{R}^n$  and  $\mathbf{A} \in \mathbb{R}^{n \times n}$ . Show that the unique solution is  $f(t) = e^{\mathbf{At}} \mathbf{x}_0$ .
14. **Matrix exponentials.** Consider the matrix exponential function in (9.8). Let  $\mathbf{X}^{-1}\mathbf{AX} = \mathbf{J} = \text{diag}(\mathbf{J}_1, \mathbf{J}_2, \dots, \mathbf{J}_k)$  be the Jordan form of  $\mathbf{A}$ , where  $\mathbf{J}_i \in \mathbb{R}^{m_i \times m_i}$  contains the eigenvalue  $\lambda_i$  along the diagonal. Show that

$$e^{\mathbf{J}_i t} = e^{\lambda_i t} \begin{bmatrix} 1 & t & \frac{t^2}{2!} & \cdots & \frac{t^{m_i-1}}{(m_i-1)!} \\ 0 & 1 & t & \cdots & \frac{t^{m_i-2}}{(m_i-2)!} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \ddots & t \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}.$$

Rewrite the matrix equation in Problem 9.13 in the form  $g'(t) = \mathbf{J}g(t)$  and determine the explicit form of  $g(t)$ . Hint: Decompose every Jordan block as a sum of a diagonal matrix and a nilpotent matrix.<sup>2</sup>

15. We have presented several important results regarding the determinant of a matrix in Lemma 9.8. Given  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , show that

<sup>2</sup> A matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is *nilpotent* if there exists a  $k$  such that  $\mathbf{A}^k = \mathbf{0}$ .

- $\det(c\mathbf{A}) = c^n \det(\mathbf{A})$ ; (*Hint: Use induction.*)
  - $\det(\mathbf{A}^{-1}) = 1/\det(\mathbf{A})$ ;
  - $\det(\mathbf{A}^m) = \det(\mathbf{A})^m$ ;
  - $\det(\mathbf{I} + \mathbf{u}\mathbf{v}^\top) = 1 + \mathbf{u}^\top \mathbf{v}$ .
16. Given  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , for  $n = 2$ , show that
- $\det(\mathbf{I} + \mathbf{A}) = 1 + \det(\mathbf{A}) + \text{tr}(\mathbf{A})$ .
- For  $n = 3$ , show that
- $\det(\mathbf{I} + \mathbf{A}) = 1 + \det(\mathbf{A}) + \text{tr}(\mathbf{A}) + \frac{1}{2}\text{tr}(\mathbf{A})^2 - \frac{1}{2}\text{tr}(\mathbf{A}^2)$ .
17. Given  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , and let  $\mathbf{B}$  be the matrix obtained by interchanging two rows of  $\mathbf{A}$ . Prove that  $\det(\mathbf{B}) = -\det(\mathbf{A})$ . *Hint: Use induction.*
18. Given  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , and let  $\mathbf{B}$  be the matrix obtained by multiplying a row of  $\mathbf{A}$  by a nonnegative scalar  $\gamma$ . Prove that  $\det(\mathbf{B}) = \gamma \det(\mathbf{A})$ .
19. Prove Theorem 9.12 rigorously.
20. Given any polynomial  $p(\lambda)$ , show that  $p(\mathbf{A})\mathbf{X} = \mathbf{X}p(\mathbf{B})$  if  $\mathbf{A}\mathbf{X} = \mathbf{X}\mathbf{B}$ .
21. Given any polynomial  $p(\lambda)$ , show that  $\mathbf{A}\mathbf{B}p(\mathbf{AB}) = \mathbf{A}p(\mathbf{BA})\mathbf{B}$  if  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{B} \in \mathbb{R}^{n \times m}$ .
22. **Diagonalization of upper triangular matrices.** Let  $\mathbf{U} \in \mathbb{R}^{n \times n}$  be an upper triangular matrix whose  $(i, j)$ -th entry is denoted by  $u_{ij}$ , and let  $\mathbf{D}_t = \text{diag}(t, t^2, \dots, t^n)$  be a diagonal matrix. Show that the similarity transformation on  $\mathbf{U}$  takes the following form:
- $$\mathbf{D}_t \mathbf{U} \mathbf{D}_t^{-1} = \begin{bmatrix} u_{11} & t^{-1}u_{12} & t^{-2}u_{13} & \dots & t^{-n+1}u_{1n} \\ 0 & u_{22} & t^{-1}u_{23} & \dots & t^{-n+2}u_{2n} \\ 0 & 0 & u_{33} & \dots & t^{-n+3}u_{3n} \\ 0 & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & u_{nn} \end{bmatrix}.$$
- Thus, when  $t$  is sufficiently large, the off-diagonal values can be made arbitrarily small.
23. Show that  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is nilpotent if and only if  $\text{tr}(\mathbf{A}^k) = 0$  for all  $k \in \{1, 2, \dots, n\}$ . *Hint: Use the m-th power eigenvalues, Remark 9.11.*
24. **Rank-one perturbation of Schur decomposition.** Suppose  $\mathbf{A} \in \mathbb{C}^{n \times n}$  has eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{C}$ , where  $\mathbf{Ax} = \lambda_1 \mathbf{x}$ . Show that, for any vector  $\mathbf{v} \in \mathbb{C}^n$ , the eigenvalues of  $\mathbf{A} + \mathbf{x}\mathbf{v}^*$  are  $\lambda_1 + \mathbf{v}^*\mathbf{x}, \lambda_2, \lambda_3, \dots, \lambda_n$ . Show that the Schur decomposition of  $\mathbf{A} + \mathbf{x}\mathbf{v}^*$  can be obtained efficiently if the Schur decomposition of  $\mathbf{A}$  is known.

# 10

## Spectral Decomposition (Theorem)

### 10.1. Spectral Decomposition (Theorem)

The *spectral theorem*, also known as the *spectral decomposition* for symmetric matrices, states that symmetric matrices have real eigenvalues and can be diagonalized using a (real) orthonormal basis<sup>1</sup>. In the following theorem, we will present the main result and defer detailed discussions.

**Theorem 10.1: (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  $\mathbf{\Lambda}$  such that

$$\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top,$$

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  $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$  are the corresponding eigenvalues of  $\mathbf{A}$ , which are real. Specifically, the following properties hold:

1. A symmetric matrix has only **real eigenvalues**.
2. The eigenvectors are orthogonal and can be chosen to be **orthonormal** by normalization.
3. The rank of  $\mathbf{A}$  is equal to the number of nonzero eigenvalues.
4. If the eigenvalues are distinct, the eigenvectors are linearly independent.

In the eigenvalue decomposition (Theorem 9.1), we require the matrix  $\mathbf{A}$  to be square and its eigenvectors to be linearly independent. In contrast, the spectral theorem applies to any symmetric matrix, and the eigenvectors are chosen to be orthonormal. On the other hand, analogous to eigenvalue decomposition, the spectral decomposition enables efficient computation of the  $m$ -th power of a matrix  $\mathbf{A}$ : If  $\mathbf{A}$  admits a spectral decomposition  $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^\top$ , then the  $m$ -th power of  $\mathbf{A}$  can be computed as  $\mathbf{A}^m = \mathbf{Q}\mathbf{\Lambda}^m\mathbf{Q}^\top$ .

In Proposition 7.4, we proved that similar matrices have the same eigenvalues. From the spectral decomposition, we observe that  $\mathbf{A}$  and  $\mathbf{\Lambda}$  are similar matrices and thus share the same eigenvalues. For any diagonal matrix, the eigenvalues are simply the entries on the main diagonal.<sup>2</sup> To verify this, observe that

$$\mathbf{\Lambda}\mathbf{e}_i = \lambda_i \mathbf{e}_i,$$

---

1. Note that for *Hermitian matrices*, the spectral decomposition states that they also have real eigenvalues and can be diagonalized using a complex orthonormal basis.  
2. In fact, in the previous section, we showed that the diagonal entries of triangular matrices are their eigenvalues.

where  $\mathbf{e}_i$  is the  $i$ -th standard basis vector. Therefore, the diagonal matrix  $\Lambda$  contains the eigenvalues of  $\mathbf{A}$ .

## 10.2. Existence of Spectral Decomposition

We will prove the theorem in several steps. We begin by showing that all eigenvalues of a symmetric matrix are real.

Symmetric Matrix Property 1 of 4

**Lemma 10.2: (Real eigenvalues)** All eigenvalues of a symmetric matrix are real.

**Proof** [of Lemma 10.2] Suppose  $\lambda$  is a complex eigenvalue of a symmetric matrix  $\mathbf{A}$ , expressed as  $\lambda = a + ib$ , where  $a$  and  $b$  are real numbers. Its complex conjugate is  $\bar{\lambda} = a - ib$ . Similarly, for the corresponding complex eigenvector  $\mathbf{x} = \mathbf{c} + i\mathbf{d}$ , its complex conjugate is  $\bar{\mathbf{x}} = \mathbf{c} - i\mathbf{d}$ , where  $\mathbf{c}$  and  $\mathbf{d}$  are real vectors. The following properties hold:

$$\mathbf{Ax} = \lambda\mathbf{x} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}\bar{\mathbf{x}} = \bar{\lambda}\bar{\mathbf{x}} \quad \xrightarrow{\text{transpose to}} \quad \bar{\mathbf{x}}^\top \mathbf{A} = \bar{\lambda}\bar{\mathbf{x}}^\top.$$

Taking the dot product of the first equation with  $\bar{\mathbf{x}}$  and the last equation with  $\mathbf{x}$ , we get:

$$\bar{\mathbf{x}}^\top \mathbf{Ax} = \lambda\bar{\mathbf{x}}^\top \mathbf{x} \quad \text{and} \quad \bar{\mathbf{x}}^\top \mathbf{A}\mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}.$$

Equating these, we find  $\lambda\bar{\mathbf{x}}^\top \mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}$ . Since  $\bar{\mathbf{x}}^\top \mathbf{x} = (\mathbf{c} - i\mathbf{d})^\top (\mathbf{c} + i\mathbf{d}) = \mathbf{c}^\top \mathbf{c} + \mathbf{d}^\top \mathbf{d}$ , which is a real number. Therefore, the imaginary part of  $\lambda$  must be zero, implying  $\lambda$  is real. ■

Symmetric Matrix Property 2 of 4

**Lemma 10.3: (Orthogonal eigenvectors)** The eigenvectors corresponding to distinct eigenvalues of any symmetric matrix are orthogonal. Therefore, these eigenvectors can be normalized to form an orthonormal set because  $\mathbf{Ax} = \lambda\mathbf{x}$  leads to  $\mathbf{A}\frac{\mathbf{x}}{\|\mathbf{x}\|} = \lambda\frac{\mathbf{x}}{\|\mathbf{x}\|}$ , which preserves the eigenvalue  $\lambda$ .

**Proof** [of Lemma 10.3] Let eigenvalues  $\lambda_1$  and  $\lambda_2$  correspond to eigenvectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , respectively, such that  $\mathbf{Ax}_1 = \lambda_1\mathbf{x}_1$  and  $\mathbf{Ax}_2 = \lambda_2\mathbf{x}_2$ . The following equalities hold:

$$\mathbf{Ax}_1 = \lambda_1\mathbf{x}_1 \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_1^\top \mathbf{A} = \lambda_1\mathbf{x}_1^\top \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_1^\top \mathbf{A}\mathbf{x}_2 = \lambda_1\mathbf{x}_1^\top \mathbf{x}_2,$$

and

$$\mathbf{Ax}_2 = \lambda_2\mathbf{x}_2 \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_2^\top \mathbf{A} = \lambda_2\mathbf{x}_2^\top \quad \xrightarrow{\text{leads to}} \quad \mathbf{x}_2^\top \mathbf{A}\mathbf{x}_2 = \lambda_2\mathbf{x}_2^\top \mathbf{x}_2.$$

Equating these, we get  $\lambda_1\mathbf{x}_1^\top \mathbf{x}_2 = \lambda_2\mathbf{x}_2^\top \mathbf{x}_1$ . Since eigenvalues  $\lambda_1 \neq \lambda_2$ , the eigenvectors are orthogonal. ■

In Lemma 10.3, we proved that eigenvectors corresponding to distinct eigenvalues of a symmetric matrix are orthogonal. More generally, we now prove a stronger result: eigenvectors corresponding to distinct eigenvalues of any square matrix are linearly independent.

**Theorem 10.4: (Independent eigenvector theorem)** If a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  has  $k$  distinct eigenvalues, then any set of  $k$  corresponding (nonzero) eigenvectors are linearly independent.

**Proof** [of Theorem 10.4] We prove the theorem by induction. First, we show that any two eigenvectors corresponding to distinct eigenvalues are linearly independent. Let eigenvectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$  correspond to distinct eigenvalues  $\lambda_1$  and  $\lambda_2$ , respectively. Assume, for contradiction, that  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are linearly dependent. Then there exists a nonzero vector  $\mathbf{x} = [x_1, x_2] \neq \mathbf{0}$  such that

$$x_1\mathbf{v}_1 + x_2\mathbf{v}_2 = \mathbf{0}. \quad (10.1)$$

Multiplying (10.1) on the left by  $\mathbf{A}$  gives:

$$x_1\lambda_1\mathbf{v}_1 + x_2\lambda_2\mathbf{v}_2 = \mathbf{0}. \quad (10.2)$$

Similarly, multiplying (10.1) by  $\lambda_2$  yields:

$$x_1\lambda_2\mathbf{v}_1 + x_2\lambda_2\mathbf{v}_2 = \mathbf{0}. \quad (10.3)$$

Subtracting these two equations yields:

$$x_1(\lambda_2 - \lambda_1)\mathbf{v}_1 = \mathbf{0}.$$

Since  $\lambda_2 \neq \lambda_1$  and  $\mathbf{v}_1 \neq \mathbf{0}$ , we must have  $x_1 = 0$ . Since  $\mathbf{v}_2 \neq \mathbf{0}$ , substituting  $x_1 = 0$  back into (10.1) shows that  $x_2 = 0$ , leading to a contradiction. Thus,  $\mathbf{v}_1$  and  $\mathbf{v}_2$  are linearly independent.

Suppose that any set of  $j < k$  eigenvectors corresponding to distinct eigenvalues is linearly independent. We aim to show that any  $j+1$  eigenvectors are also linearly independent. Let  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j$  be linearly independent eigenvectors corresponding to distinct eigenvalues  $\lambda_1, \lambda_2, \dots, \lambda_j$ . Assume, for contradiction, that an additional eigenvector  $\mathbf{v}_{j+1}$ , corresponding to a different eigenvalue  $\lambda_{j+1}$ , is linearly dependent on  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j$ . Then there exist scalars  $x_1, x_2, \dots, x_j$ , not all zero, such that:

$$\mathbf{v}_{j+1} = x_1\mathbf{v}_1 + x_2\mathbf{v}_2 + \dots + x_j\mathbf{v}_j. \quad (10.4)$$

Multiplying (10.4) on the left by  $\mathbf{A}$  gives:

$$\lambda_{j+1}\mathbf{v}_{j+1} = x_1\lambda_1\mathbf{v}_1 + x_2\lambda_2\mathbf{v}_2 + \dots + x_j\lambda_j\mathbf{v}_j. \quad (10.5)$$

Similarly, multiplying (10.4) by  $\lambda_{j+1}$  yields:

$$\lambda_{j+1}\mathbf{v}_{j+1} = x_1\lambda_{j+1}\mathbf{v}_1 + x_2\lambda_{j+1}\mathbf{v}_2 + \dots + x_j\lambda_{j+1}\mathbf{v}_j. \quad (10.6)$$

Subtracting the two equations gives:

$$x_1(\lambda_{j+1} - \lambda_1)\mathbf{v}_1 + x_2(\lambda_{j+1} - \lambda_2)\mathbf{v}_2 + \dots + x_j(\lambda_{j+1} - \lambda_j)\mathbf{v}_j = \mathbf{0}.$$

From the 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. Thus, the eigenvectors  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j, \mathbf{v}_{j+1}$  are linearly independent. By induction, any  $k$  eigenvectors corresponding to  $k$  distinct eigenvalues are linearly independent.  $\blacksquare$

An immediate consequence of the above theorem is the following:

**Corollary 10.5: (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 10.6: (Orthonormal eigenvectors for duplicate eigenvalue)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be symmetric. If  $\mathbf{A}$  has a repeated eigenvalue  $\lambda_i$  with multiplicity  $k \geq 2$ , then there exist  $k$  orthonormal eigenvectors corresponding to  $\lambda_i$ .

a. Multiplicity is rigorously defined in Definition 10.10.

**Proof** [of Lemma 10.6] First, note that there exists at least one eigenvector  $\mathbf{x}_{i1}$  corresponding to  $\lambda_i$ . For this eigenvector  $\mathbf{x}_{i1}$ , we can always find  $n - 1$  additional orthonormal vectors  $\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n$ , such that the set  $\{\mathbf{x}_{i1}, \mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n\}$  forms an orthonormal basis for  $\mathbb{R}^n$ . Define the matrices  $\mathbf{Y}_1$  and  $\mathbf{P}_1$  as follows:

$$\mathbf{Y}_1 = [\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n] \quad \text{and} \quad \mathbf{P}_1 = [\mathbf{x}_{i1}, \mathbf{Y}_1].$$

Since  $\mathbf{A}$  is symmetric, we then have  $\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 = \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 \end{bmatrix}$ . Since  $\mathbf{P}_1$  is nonsingular and orthogonal, it follows that  $\mathbf{A}$  and  $\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1$  are similar matrices such that they share the same eigenvalues (see Proposition 7.4), whence we have

$$\det(\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 - \lambda \mathbf{I}_n) = ^3 (\lambda_i - \lambda) \det(\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 - \lambda \mathbf{I}_{n-1}).$$

If  $\lambda_i$  has multiplicity  $k \geq 2$ , then the term  $(\lambda_i - \lambda)$  appears  $k$  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 appears  $k - 1$  times in the polynomial from  $\det(\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 - \lambda \mathbf{I}_{n-1})$ . Thus,  $\det(\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1 - \lambda_i \mathbf{I}_{n-1}) = 0$ , and  $\lambda_i$  is an eigenvalue of  $\mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1$  with multiplicity  $k - 1$ .

Define  $\mathbf{B} = \mathbf{Y}_1^\top \mathbf{A} \mathbf{Y}_1$ . Since  $\det(\mathbf{B} - \lambda_i \mathbf{I}_{n-1}) = 0$ , the null space of  $\mathbf{B} - \lambda_i \mathbf{I}_{n-1}$  is nontrivial. Suppose  $(\mathbf{B} - \lambda_i \mathbf{I}_{n-1})\mathbf{n} = \mathbf{0}$ , i.e.,  $\mathbf{B}\mathbf{n} = \lambda_i \mathbf{n}$ , where  $\mathbf{n}$  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

$$\mathbf{A} \mathbf{P}_1 \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} = [\lambda_i \mathbf{x}_{i1}, \mathbf{A} \mathbf{Y}_1] \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} = \lambda_i z \mathbf{x}_{i1} + \mathbf{A} \mathbf{Y}_1 \mathbf{n}. \quad (10.7)$$

From the right side of the equation, we have

$$\begin{aligned} \mathbf{P}_1 \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} &= [\mathbf{x}_{i1} \quad \mathbf{Y}_1] \begin{bmatrix} \lambda_i & \mathbf{0} \\ \mathbf{0} & \mathbf{B} \end{bmatrix} \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} = [\lambda_i \mathbf{x}_{i1} \quad \mathbf{Y}_1 \mathbf{B}] \begin{bmatrix} z \\ \mathbf{n} \end{bmatrix} \\ &= \lambda_i z \mathbf{x}_{i1} + \mathbf{Y}_1 \mathbf{B} \mathbf{n} = \lambda_i z \mathbf{x}_{i1} + \lambda_i \mathbf{Y}_1 \mathbf{n}, \end{aligned} \quad (10.8)$$

where the last equality follows from  $\mathbf{B}\mathbf{n} = \lambda_i \mathbf{n}$ . Combining Equations (10.8) and (10.7), we obtain

$$\mathbf{A} \mathbf{Y}_1 \mathbf{n} = \lambda_i \mathbf{Y}_1 \mathbf{n},$$

3. 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  $\det(\mathbf{M}) = \det(\mathbf{A}) \det(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})$ .

which means  $\mathbf{Y}_1\mathbf{n}$  is an eigenvector of  $\mathbf{A}$  corresponding to the eigenvalue  $\lambda_i$  (the same eigenvalue corresponding to  $\mathbf{x}_{i1}$ ). Since  $\mathbf{Y}_1\mathbf{n}$  is a linear combination of  $\mathbf{y}_2, \mathbf{y}_3, \dots, \mathbf{y}_n$ , which are orthonormal to  $\mathbf{x}_{i1}$ , it can be chosen to be orthonormal to  $\mathbf{x}_{i1}$  by scaling  $\mathbf{n}$ .

To summarize, if there exists an eigenvector,  $\mathbf{x}_{i1}$ , corresponding to the eigenvalue  $\lambda_i$ , with multiplicity  $k \geq 2$ , we can construct a second eigenvector by selecting a vector from the null space of  $(\mathbf{B} - \lambda_i \mathbf{I}_{n-1})$ , as outlined above. Assume that we have constructed this second eigenvector,  $\mathbf{x}_{i2}$ , which is orthonormal to  $\mathbf{x}_{i1}$ . With these eigenvectors,  $\mathbf{x}_{i1}$  and  $\mathbf{x}_{i2}$ , we can always find  $n - 2$  additional orthonormal vectors  $\mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n$ , such that the set  $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n\}$  forms an orthonormal basis for  $\mathbb{R}^n$ . Arrange these vectors  $\mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n$  into matrix  $\mathbf{Y}_2$  and  $\{\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n\}$  into matrix  $\mathbf{P}_2$ :

$$\mathbf{Y}_2 = [\mathbf{y}_3, \mathbf{y}_4, \dots, \mathbf{y}_n] \quad \text{and} \quad \mathbf{P}_2 = [\mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{Y}_1].$$

Since  $\mathbf{A}$  is symmetric, 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  $\det(\mathbf{P}_2^\top \mathbf{A} \mathbf{P}_2 - \lambda_i \mathbf{I}_n) = (\lambda_i - \lambda)^2 \det(\mathbf{C} - \lambda_i \mathbf{I}_{n-2})$ . If the multiplicity of  $\lambda_i$  is  $k \geq 3$ , then  $\det(\mathbf{C} - \lambda_i \mathbf{I}_{n-2}) = 0$ , implying the null space of  $\mathbf{C} - \lambda_i \mathbf{I}_{n-2}$  is non-empty. From the null space of  $\mathbf{C} - \lambda_i \mathbf{I}_{n-2}$ , we can select a vector  $\mathbf{n}$  such that  $\mathbf{C}\mathbf{n} = \lambda_i \mathbf{n}$ . Using this vector, we construct  $\begin{bmatrix} z_1 \\ z_2 \\ \mathbf{n} \end{bmatrix} \in \mathbb{R}^n$ , where  $z_1$  and  $z_2$  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  $\mathbf{A}$ , orthogonal to  $\mathbf{x}_{i1}$  and  $\mathbf{x}_{i2}$ . This eigenvector can also be normalized to ensure orthonormality with the first two eigenvectors.

By iterating this process, we can construct  $k$  orthonormal eigenvectors corresponding to the eigenvalue  $\lambda_i$ .

Finally, the dimension of the null space of  $\mathbf{P}_1^\top \mathbf{A} \mathbf{P}_1 - \lambda_i \mathbf{I}_n$  equals the multiplicity  $k$ . This confirms that if  $\lambda_i$  has multiplicity  $k$ , there cannot be more than  $k$  orthonormal eigenvectors corresponding to  $\lambda_i$ . Otherwise, we would obtain more than  $n$  orthogonal eigenvectors in  $\mathbb{R}^n$ , which leads to a contradiction.  $\blacksquare$

The existence of the spectral decomposition can be straightforwardly established using the lemmas presented above. Alternatively, we can prove its existence by applying the Schur decomposition.

**Proof** [of Theorem 10.1: Existence of spectral decomposition, alternative proof] According to Theorem 9.7, a symmetric matrix  $\mathbf{A} = \mathbf{A}^\top$  can be decomposed as  $\mathbf{Q} \mathbf{U} \mathbf{Q}^\top = \mathbf{Q} \mathbf{U}^\top \mathbf{Q}^\top$ , where  $\mathbf{Q}$  is orthogonal and  $\mathbf{U}$  is upper triangular. This implies that  $\mathbf{U}$  must be a diagonal matrix whose diagonal entries are the eigenvalues of  $\mathbf{A}$ . Consequently, the columns of  $\mathbf{Q}$  are the corresponding eigenvectors of  $\mathbf{A}$ . Therefore, we conclude that every symmetric matrix

can be orthogonally diagonalized, even when eigenvalues are repeated. ■

For a more concise proof, Lu (2021c) presents an approach utilizing the Gram–Schmidt process combined with mathematical induction.

For a symmetric matrix  $\mathbf{A}^\top \mathbf{A}$ , the rank remains the same as that of  $\mathbf{A}$ , a property we will utilize in proving the singular value decomposition in the next chapter. However, in general, the rank of a product of two matrices does not exceed the rank of either matrix; see also Exercises 0.11 and 0.12.

**Lemma 10.7: (Rank of  $\mathbf{AB}$ )** Given matrices  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{B} \in \mathbb{R}^{n \times k}$ , the rank of their product  $\mathbf{AB} \in \mathbb{R}^{m \times k}$  satisfies  $\text{rank}(\mathbf{AB}) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\}$ .

**Proof** [of Lemma 10.7] Considering the matrix product  $\mathbf{AB}$ :

- Each row of  $\mathbf{AB}$  is a linear combination of the rows of  $\mathbf{B}$ , implying that the row space of  $\mathbf{AB}$  is contained within that of  $\mathbf{B}$ . Therefore,  $\text{rank}(\mathbf{AB}) \leq \text{rank}(\mathbf{B})$ .
- Similarly, each column of  $\mathbf{AB}$  is a linear combination of the columns of  $\mathbf{A}$ , so the column space of  $\mathbf{AB}$  is contained within that of  $\mathbf{A}$ . Hence,  $\text{rank}(\mathbf{AB}) \leq \text{rank}(\mathbf{A})$ .

Combining these observations, we conclude that  $\text{rank}(\mathbf{AB}) \leq \min\{\text{rank}(\mathbf{A}), \text{rank}(\mathbf{B})\}$ . ■

#### Symmetric Matrix Property 4 of 4

**Lemma 10.8: (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  $\mathbf{A}$ . Furthermore, the column space  $\mathcal{C}(\mathbf{A})$  is the linear subspace spanned by the eigenvectors of  $\mathbf{A}$  corresponding to its nonzero eigenvalues.

**Proof** [of Lemma 10.8] For any symmetric matrix  $\mathbf{A}$ , it can be expressed in its spectral form as  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , where  $\mathbf{Q}$  is an orthogonal matrix and  $\Lambda$  is a diagonal matrix containing the eigenvalues of  $\mathbf{A}$ . Using Lemma 10.7, we proceed as follows:

- 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  $\text{rank}(\Lambda) \leq \text{rank}(\mathbf{Q}^\top \mathbf{A}) \leq \text{rank}(\mathbf{A})$ .

This implies  $\text{rank}(\mathbf{A}) = \text{rank}(\Lambda)$ , which is equal to the total number of nonzero eigenvalues of  $\mathbf{A}$ . ■

### 10.3. Uniqueness of Spectral Decomposition

It's important to note that spectral decomposition of a matrix is generally not unique. This is primarily due to the presence of repeated eigenvalues. When two or more eigenvalues  $\lambda_i$  and  $\lambda_j$  (for  $1 \leq i, j \leq n$ ) are identical, swapping their corresponding eigenvectors in the orthogonal matrix  $\mathbf{Q}$  results in a different decomposition that is still mathematically valid and equivalent.

However, the *eigenspaces* associated with each eigenvalue—specifically, the null spaces  $\mathcal{N}(\mathbf{A} - \lambda_i \mathbf{I})$  for each eigenvalue  $\lambda_i$ —remain fixed. This means that while the choice of eigenvectors within each eigenspace can vary, leading to different decompositions, the de-

composition in terms of eigenspaces is unique. In other words, any orthonormal basis for these eigenspaces can be used without affecting the overall spectral decomposition.

## 10.4. Other Forms, Connecting Eigenvalue Decomposition\*

In this section, we examine various types of spectral decomposition under different conditions. To support this discussion, we begin with a formal definition of the characteristic polynomial of a square matrix.

**Definition 10.9 (Characteristic polynomial).** For any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , its *characteristic polynomial* is defined as:

$$\det(\lambda\mathbf{I} - \mathbf{A}) = \lambda^n + \gamma_{n-1}\lambda^{n-1} + \dots + \gamma_1\lambda + \gamma_0 = (\lambda - \lambda_1)^{k_1}(\lambda - \lambda_2)^{k_2} \dots (\lambda - \lambda_m)^{k_m},$$

where  $\lambda_1, \lambda_2, \dots, \lambda_m$  are the distinct roots of  $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$ , which are also the eigenvalues of  $\mathbf{A}$ . The sum of the multiplicities satisfies  $k_1 + k_2 + \dots + k_m = n$ , indicating that  $\det(\lambda\mathbf{I} - \mathbf{A})$  is a polynomial of degree  $n$  for any matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  (see proof of Lemma 10.6). The equation  $\det(\lambda\mathbf{I} - \mathbf{A}) = 0$  is referred to as the *characteristic equation* of  $\mathbf{A}$ .

The characteristic polynomial is essential in defining two key concepts: algebraic multiplicity and geometric multiplicity.

**Definition 10.10 (Algebraic multiplicity and geometric multiplicity).** Given the characteristic polynomial of a 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., it equals the multiplicity of the corresponding root in the characteristic polynomial.

The *eigenspace associated with the eigenvalue  $\lambda_i$*  is the null space of  $(\mathbf{A} - \lambda_i\mathbf{I})$ , denoted by  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$ . And the dimension of the eigenspace associated with  $\lambda_i$ ,  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$ , is known as the *geometric multiplicity* of  $\lambda_i$ .

For brevity, we denote the algebraic multiplicity of  $\lambda_i$  by  $alg(\lambda_i)$  and its geometric multiplicity by  $geo(\lambda_i)$ .

**Remark 10.11 (Geometric multiplicity).** For a matrix  $\mathbf{A}$  and its eigenspace  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$  corresponding to an eigenvalue  $\lambda_i$ , the dimension of the eigenspace reflects the number of linearly independent eigenvectors of  $\mathbf{A}$  associated with  $\lambda_i$ . This means that while there are infinitely many eigenvectors associated with each eigenvalue  $\lambda_i$ , they form a subspace that can be described using a finite set of basis vectors. In other words, the geometric multiplicity indicates the maximum number of linearly independent eigenvectors available for  $\lambda_i$ .

By definition, the sum of the algebraic multiplicities of all eigenvalues equals  $n$ , whereas the sum of the geometric multiplicities can be strictly smaller.

**Corollary 10.12: (Multiplicity in similar matrices)** Similar matrices share the same algebraic and geometric multiplicities for their eigenvalues.

**Proof** [of Corollary 10.12] From Proposition 7.4, we know that similar matrices have identical eigenvalues, which implies they also share the same algebraic multiplicities.

Consider two similar matrices  $\mathbf{A}$  and  $\mathbf{B} = \mathbf{P}\mathbf{A}\mathbf{P}^{-1}$ , where  $\mathbf{P}$  is nonsingular. Suppose the geometric multiplicity of an eigenvalue  $\lambda$  of  $\mathbf{A}$  is  $k$ . This means there exist  $k$  linearly independent eigenvectors  $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$  forming a basis for the eigenspace  $\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})$  such that  $\mathbf{A}\mathbf{v}_i = \lambda\mathbf{v}_i$  for each  $i \in \{1, 2, \dots, k\}$ . Then,  $\mathbf{w}_i = \mathbf{P}\mathbf{v}_i$ 's are the eigenvectors of  $\mathbf{B}$  associated with  $\lambda$ . Since  $\mathbf{P}$  is nonsingular, these  $\mathbf{w}_i$ 's are also linearly independent. Thus, the dimension of the eigenspace  $\mathcal{N}(\mathbf{B} - \lambda\mathbf{I})$  is at least  $k$ , implying  $\dim(\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})) \leq \dim(\mathcal{N}(\mathbf{B} - \lambda\mathbf{I}))$ .

Conversely, if we start with a set of  $k$  linearly independent eigenvectors  $\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_k$  for  $\mathbf{B}$  corresponding to  $\lambda$ , then the vectors  $\mathbf{v}_i = \mathbf{P}^{-1}\mathbf{w}_i$  for all  $i \in \{1, 2, \dots, k\}$  are eigenvectors of  $\mathbf{A}$  associated with  $\lambda$ . This gives us  $\dim(\mathcal{N}(\mathbf{B} - \lambda\mathbf{I})) \leq \dim(\mathcal{N}(\mathbf{A} - \lambda\mathbf{I}))$ .

By combining both inequalities, we conclude that  $\dim(\mathcal{N}(\mathbf{A} - \lambda\mathbf{I})) = \dim(\mathcal{N}(\mathbf{B} - \lambda\mathbf{I}))$ , establishing the equality of geometric multiplicities for similar matrices. ■

**Lemma 10.13: (Bounded geometric multiplicity)** For any matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and its eigenvalue  $\lambda_i$ , the geometric multiplicity is bounded by the algebraic multiplicity:

$$\text{geo}(\lambda_i) \leq \text{alg}(\lambda_i).$$

**Proof** [of Lemma 10.13] Suppose  $\mathbf{P}_1 = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k]$  contains a set of linearly independent eigenvectors of  $\mathbf{A}$  associated with  $\lambda_i$ . That is, the  $k$  vectors form a basis for the eigenspace  $\mathcal{N}(\mathbf{A} - \lambda_i\mathbf{I})$ , and the geometric multiplicity associated with  $\lambda_i$  is  $k$ . Extend  $\mathbf{P}_1$  to a full basis  $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2] = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k, \mathbf{v}_{k+1}, \dots, \mathbf{v}_n]$ , where  $\mathbf{P}$  is nonsingular. Then we have  $\mathbf{AP} = [\lambda_i\mathbf{P}_1, \mathbf{AP}_2]$ .

Now 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,  $\mathbf{P}^{-1}\mathbf{AP} = \mathbf{B}$ , and therefore  $\mathbf{A}$  and  $\mathbf{B}$  are similar matrices. Such matrices  $\mathbf{C}$  and  $\mathbf{D}$  always exist because the vectors  $\mathbf{v}_i$  are linearly independent vectors spanning the entire space  $\mathbb{R}^n$ , and any column of  $\mathbf{AP}_2$  belongs to the column space of  $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ . Therefore,

$$\begin{aligned} \det(\mathbf{A} - \lambda\mathbf{I}) &= \det(\mathbf{P}^{-1}) \det(\mathbf{A} - \lambda\mathbf{I}) \det(\mathbf{P}) = \det(\mathbf{P}^{-1}(\mathbf{A} - \lambda\mathbf{I})\mathbf{P}) = \det(\mathbf{B} - \lambda\mathbf{I}) \\ &= \det\left(\begin{bmatrix} (\lambda_i - \lambda)\mathbf{I}_k & \mathbf{C} \\ \mathbf{0} & \mathbf{D} - \lambda\mathbf{I} \end{bmatrix}\right) = (\lambda_i - \lambda)^k \det(\mathbf{D} - \lambda\mathbf{I}). \end{aligned}$$

This shows that the algebraic multiplicity of  $\lambda_i$  is at least  $k$ , which is the geometric multiplicity. Therefore,  $\text{geo}(\lambda_i) \leq \text{alg}(\lambda_i)$ . And we complete the proof. ■

Building on the proof of Lemma 10.6, it becomes evident that for symmetric matrices, the algebraic and geometric multiplicities of all eigenvalues are equal. Such matrices are called *simple matrices*.

**Definition 10.14 (Simple matrix).** A square matrix is called *simple* if, for each of its eigenvalues, the algebraic multiplicity equals the geometric multiplicity.

**Definition 10.15 (Diagonalizable).** A square matrix  $\mathbf{A}$  is said to be *diagonalizable* if there exists a nonsingular matrix  $\mathbf{P}$  and a diagonal matrix  $\mathbf{D}$  such that  $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ .

Diagonal matrices have a particularly simple structure, which makes computations such as determinants and inverses more straightforward. The eigenvalue decomposition (Theorem 9.1) and the spectral decomposition (Theorem 10.1) are examples of diagonalization techniques applicable to specific classes of matrices.

**Lemma 10.16: (Simple matrices are diagonalizable)** A matrix is simple if and only if it is diagonalizable.

**Proof** [of Lemma 10.16] Suppose that  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is a simple matrix, meaning 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  $\mathcal{N}(\mathbf{A} - \lambda_i \mathbf{I})$ . In other words,  $\{\mathbf{v}_1^i, \mathbf{v}_2^i, \dots, \mathbf{v}_{k_i}^i\}$  is a set of linearly independent eigenvectors of  $\mathbf{A}$  associated with  $\lambda_i$ , where  $k_i$  is the algebraic or geometric multiplicity of  $\lambda_i$ :  $alg(\lambda_i) = geo(\lambda_i) = k_i$ . 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. Consider a linear combination of these eigenvectors:

$$\mathbf{z} = \sum_{j=1}^{k_1} x_j^1 \mathbf{v}_j^1 + \sum_{j=1}^{k_2} x_j^2 \mathbf{v}_j^2 + \dots + \sum_{j=1}^{k_m} x_j^m \mathbf{v}_j^m = \mathbf{0}. \quad (10.9)$$

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 with  $\lambda_i$  or the zero vector. Therefore,  $\mathbf{z} = \sum_{i=1}^m \mathbf{w}^i$  is a sum of either zero vectors or eigenvectors 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 with  $\lambda_i$  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:

$$\{\mathbf{v}_1^1, \mathbf{v}_2^1, \dots, \mathbf{v}_{k_1}^1\}, \{\mathbf{v}_1^2, \mathbf{v}_2^2, \dots, \mathbf{v}_{k_2}^2\}, \dots, \{\mathbf{v}_1^m, \mathbf{v}_2^m, \dots, \mathbf{v}_{k_m}^m\}.$$

According to the eigenvalue decomposition presented in Theorem 9.1,  $\mathbf{A}$  is diagonalizable.

Conversely, suppose  $\mathbf{A}$  is diagonalizable. That is, there exists a nonsingular matrix  $\mathbf{P}$  and a diagonal matrix  $\mathbf{D}$  such that  $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ . Then  $\mathbf{A}$  and  $\mathbf{D}$  are similar matrices, and therefore they have the same eigenvalues (Proposition 7.4), the same algebraic multiplicities, and the same geometric multiplicities (Corollary 10.12). It can be readily verified that a diagonal matrix has equal algebraic and geometric multiplicities. Therefore,  $\mathbf{A}$  is a simple matrix. ■

From Theorem 10.4, which states that any eigenvectors corresponding to different eigenvalues are linearly independent, and Remark 10.11, which explains that the geometric multiplicity is the dimension of the corresponding eigenspace, we can conclude the following: if, for a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , the geometric multiplicity is equal to the algebraic multiplicity (for all eigenvalues), the eigenspaces can span the entire space  $\mathbb{R}^n$ . Hence, the above lemma equivalently claims that if the eigenspaces span the entire space  $\mathbb{R}^n$ , then  $\mathbf{A}$  can be diagonalized.

**Corollary 10.17** A square matrix  $\mathbf{A}$  is considered simple if it has a complete set of linearly independent eigenvectors. Alternatively, any symmetric matrix  $\mathbf{A}$  also qualifies as a simple matrix by definition.

The proof of this corollary follows directly from the eigenvalue decomposition given in Theorem 9.1 and the spectral theorem presented in Theorem 10.1.

We now present an alternative expression for the spectral decomposition:

**Theorem 10.18: (Spectral decomposition: the second form)** For a **simple matrix**  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , it can be expressed as a weighted sum of idempotent matrices:

$$\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{A}_i,$$

where each  $\lambda_i$ , for  $i \in \{1, 2, \dots, n\}$ , represents an eigenvalue of  $\mathbf{A}$  (potentially repeated). The idempotent matrices  $\mathbf{A}_i$  satisfy 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 10.18] Given that  $\mathbf{A}$  is a simple matrix, according to Lemma 10.16, there exists a nonsingular matrix  $\mathbf{P}$  and a diagonal matrix  $\Lambda$  such that  $\mathbf{A} = \mathbf{P} \Lambda \mathbf{P}^{-1}$ , where  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ ,  $\lambda_i$ 's are eigenvalues of  $\mathbf{A}$ , and the columns of  $\mathbf{P}$  consist of the corresponding eigenvectors. Let

$$\mathbf{P} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_n] \quad \text{and} \quad \mathbf{P}^{-1} = [\mathbf{w}_1^\top; \mathbf{w}_2^\top; \dots; \mathbf{w}_n^\top]$$

denote the column and row partitions of  $\mathbf{P}$  and  $\mathbf{P}^{-1}$ , respectively. Then, we can rewrite  $\mathbf{A}$  as

$$\mathbf{A} = \mathbf{P} \Lambda \mathbf{P}^{-1} = [\mathbf{v}_1 \ \mathbf{v}_2 \ \dots \ \mathbf{v}_n] \Lambda \begin{bmatrix} \mathbf{w}_1^\top \\ \mathbf{w}_2^\top \\ \vdots \\ \mathbf{w}_n^\top \end{bmatrix} = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{w}_i^\top.$$

By defining  $\mathbf{A}_i = \mathbf{v}_i \mathbf{w}_i^\top$ , we obtain  $\mathbf{A} = \sum_{i=1}^n \lambda_i \mathbf{A}_i$ . It follows from  $\mathbf{P}^{-1} \mathbf{P} = \mathbf{I}$  that

$$\begin{cases} \mathbf{w}_i^\top \mathbf{v}_j = 1, & \text{if } i = j; \\ \mathbf{w}_i^\top \mathbf{v}_j = 0, & \text{if } i \neq j. \end{cases}$$

Thus,

$$\mathbf{A}_i \mathbf{A}_j = \mathbf{v}_i \mathbf{w}_i^\top \mathbf{v}_j \mathbf{w}_j^\top = \begin{cases} \mathbf{v}_i \mathbf{w}_i^\top = \mathbf{A}_i, & \text{if } i = j; \\ \mathbf{0}, & \text{if } i \neq j. \end{cases}$$

This confirms both the idempotency and orthogonality of the matrices  $\mathbf{A}_i$ . Moreover, we have  $\sum_{i=1}^n \mathbf{A}_i = \mathbf{P} \mathbf{P}^{-1} = \mathbf{I}$ , which verifies their additivity. Finally, the rank-additivity property holds trivially because  $\text{rank}(\mathbf{A}_i) = 1$  for all  $i \in \{1, 2, \dots, n\}$ . ■

This form of the decomposition is closely related to *Cochran's theorem* and is widely used in the distribution theory of linear models (Lu, 2021c,d).

Going further, suppose we have  $k$  distinct eigenvalues. Then we have the following result.

**Theorem 10.19: (Spectral decomposition: the third form)** For a **simple matrix**  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with  $k$  **distinct eigenvalues**, it can be expressed as a weighted sum of a set

of idempotent matrices:

$$\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i,$$

where each  $\lambda_i$ , for  $i \in \{1, 2, \dots, k\}$ , represents one of the distinct eigenvalues of  $\mathbf{A}$ . The idempotent matrices  $\mathbf{A}_i$  satisfy 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 10.19] Building on Theorem 10.18, we can express  $\mathbf{A}$  as  $\mathbf{A} = \sum_{j=1}^n \beta_j \mathbf{B}_j$ , where  $\beta'_j$ 's are the eigenvalues and  $\mathbf{B}'_j$ 's are the corresponding idempotent matrices from the second form of the spectral decomposition. Assume without loss of generality that the eigenvalues are ordered such that  $\beta_1 \leq \beta_2 \leq \dots \leq \beta_n$ , allowing for duplicates. Let  $\{\lambda_1, \lambda_2, \dots, \lambda_k\}$  denote the set of  $k$  distinct eigenvalues, and let  $\mathbf{A}_i$  represent the sum of the  $\mathbf{B}_j$  matrices associated with  $\lambda_i$ . Suppose the multiplicity of  $\lambda_i$  is  $m_i$ , and the set of  $\mathbf{B}_j$  matrices associated with  $\lambda_i$  can be denoted by  $\{\mathbf{B}_1^i, \mathbf{B}_2^i, \dots, \mathbf{B}_{m_i}^i\}$ . Thus,  $\mathbf{A}_i$  can be defined as  $\mathbf{A}_i = \sum_{j=1}^{m_i} \mathbf{B}_j^i$ . Consequently, we have  $\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i$ .

**Idempotency.**  $\mathbf{A}_i^2 = (\mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i)(\mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i) = \mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i = \mathbf{A}_i$  due to the idempotency and orthogonality of the  $\mathbf{B}_j^i$  matrices.

**Ortognality.**  $\mathbf{A}_i \mathbf{A}_j = (\mathbf{B}_1^i + \mathbf{B}_2^i + \dots + \mathbf{B}_{m_i}^i)(\mathbf{B}_1^j + \mathbf{B}_2^j + \dots + \mathbf{B}_{m_j}^j) = \mathbf{0}$  due to the orthogonality of the  $\mathbf{B}_j^i$  matrices.

**Additivity.** It is evident 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  $\text{rank}(\mathbf{A}_1) + \text{rank}(\mathbf{A}_2) + \dots + \text{rank}(\mathbf{A}_k) = m_1 + m_2 + \dots + m_k = n$ . ■

The reverse implication of the above theorem also holds true.

**Theorem 10.20: (Spectral decomposition: backward implication)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a matrix with  $k$  distinct eigenvalues. If  $\mathbf{A}$  can be decomposed as a linear combination of a set of idempotent matrices

$$\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i,$$

where each  $\lambda_i$ , for  $i \in \{1, 2, \dots, k\}$ , represents one of the distinct eigenvalues of  $\mathbf{A}$ , and the matrices  $\mathbf{A}_i$  satisfy the four conditions outlined in Theorem 10.19, then  $\mathbf{A}$  is a simple matrix.

**Proof** [of Theorem 10.20] Assume that  $\text{rank}(\mathbf{A}_i) = r_i$  for all  $i \in \{1, 2, \dots, k\}$ . By the ULV decomposition given in Theorem 4.1, each  $\mathbf{A}_i$  can be decomposed as  $\mathbf{A}_i = \mathbf{U}_i \begin{bmatrix} \mathbf{L}_i & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}_i$ , where  $\mathbf{L}_i \in \mathbb{R}^{r_i \times r_i}$  is lower triangular, and  $\mathbf{U}_i \in \mathbb{R}^{n \times n}$  and  $\mathbf{V}_i \in \mathbb{R}^{n \times n}$  are orthogonal. Define

$$\mathbf{X}_i = \mathbf{U}_i \begin{bmatrix} \mathbf{L}_i \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \mathbf{V}_i = \begin{bmatrix} \mathbf{Y}_i \\ \mathbf{Z}_i \end{bmatrix},$$

where  $\mathbf{X}_i$  is of size  $\mathbb{R}^{n \times r_i}$ , and  $\mathbf{Y}_i \in \mathbb{R}^{r_i \times n}$  consists of the first  $r_i$  rows of  $\mathbf{V}_i$ . Consequently, we have  $\mathbf{A}_i = \mathbf{X}_i \mathbf{Y}_i$ . This can be seen as a *reduced* ULV decomposition of  $\mathbf{A}_i$ . Concatenating the  $\mathbf{X}_i$ 's and  $\mathbf{Y}_i$ 's into matrices  $\mathbf{X}$  and  $\mathbf{Y}$ :

$$\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k], \quad \text{and} \quad \mathbf{Y} = [\mathbf{Y}_1; \mathbf{Y}_2; \dots; \mathbf{Y}_k],$$

where  $\mathbf{X} \in \mathbb{R}^{n \times n}$  and  $\mathbf{Y} \in \mathbb{R}^{n \times n}$  (by rank-additivity). Using block matrix multiplication and leveraging the additivity property of the  $\mathbf{A}_i$ '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 we also have

$$\mathbf{YX} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_k \end{bmatrix} [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k] = \begin{bmatrix} \mathbf{Y}_1 \mathbf{X}_1 & \mathbf{Y}_1 \mathbf{X}_2 & \dots & \mathbf{Y}_1 \mathbf{X}_k \\ \mathbf{Y}_2 \mathbf{X}_1 & \mathbf{Y}_2 \mathbf{X}_2 & \dots & \mathbf{Y}_2 \mathbf{X}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}_k \mathbf{X}_1 & \mathbf{Y}_k \mathbf{X}_2 & \dots & \mathbf{Y}_k \mathbf{X}_k \end{bmatrix} = \mathbf{I},$$

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 conclude that

$$\mathbf{AX} = \mathbf{A}[\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_k] = [\lambda_1 \mathbf{X}_1, \lambda_2 \mathbf{X}_2, \dots, \lambda_k \mathbf{X}_k] = \mathbf{X}\Lambda,$$

where  $\Lambda = \text{diag}(\lambda_1 \mathbf{I}_{r_1}, \lambda_2 \mathbf{I}_{r_2}, \dots, \lambda_k \mathbf{I}_{r_k})$  is a diagonal matrix. This implies  $\mathbf{A}$  can be diagonalized, and by Lemma 10.16,  $\mathbf{A}$  is indeed a simple matrix. ■

Combining Theorem 10.19 and Theorem 10.20, we can claim that a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is a simple matrix with  $k$  distinct eigenvalues if and only if it can be decomposed as a sum of a set of idempotent matrices

$$\mathbf{A} = \sum_{i=1}^k \lambda_i \mathbf{A}_i,$$

where each  $\lambda_i$ , for  $i \in \{1, 2, \dots, k\}$ , represents one of the distinct eigenvalues of  $\mathbf{A}$ , and the matrices  $\mathbf{A}_i$  satisfy the four conditions outlined in Theorem 10.19.

## 10.5. Skew-Symmetric Matrix and its Properties\*

We presented the spectral decomposition of symmetric matrices. Another significant class of matrices related to symmetry is known as *skew-symmetric matrices*.

**Definition 10.21 (Skew-symmetric matrix).** A matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is called a *skew-symmetric matrix* if it satisfies the condition  $\mathbf{A}^\top = -\mathbf{A}$ . Under this definition, the diagonal entries  $a_{ii}$  for all  $i \in \{1, 2, \dots, n\}$  must satisfy the equation  $a_{ii} = -a_{ii}$ , which implies that all diagonal entries are zero.

Previously, in Lemma 10.2, we established that the eigenvalues of symmetric matrices are real. Similarly, it can be shown that all eigenvalues of skew-symmetric matrices are either purely imaginary or zero.

**Lemma 10.22: (Imaginary eigenvalues)** The eigenvalues of any skew-symmetric matrix are either purely imaginary or zero.

**Proof** [of Lemma 10.22] Suppose the eigenvalue  $\lambda$  of the skew-symmetric matrix  $\mathbf{A}$  is a complex number  $\lambda = a + ib$ , where  $a$  and  $b$  are real numbers. Its complex conjugate is  $\bar{\lambda} = a - ib$ . Similarly, for the corresponding complex eigenvector  $\mathbf{x} = \mathbf{c} + i\mathbf{d}$ , its complex conjugate is  $\bar{\mathbf{x}} = \mathbf{c} - i\mathbf{d}$ , where  $\mathbf{c}$  and  $\mathbf{d}$  are real vectors. The eigenvalue equation and its conjugate can be written as:

$$\mathbf{Ax} = \lambda\mathbf{x} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A}\bar{\mathbf{x}} = \bar{\lambda}\bar{\mathbf{x}} \quad \xrightarrow{\text{transpose to}} \quad \bar{\mathbf{x}}^\top \mathbf{A}^\top = \bar{\lambda}\bar{\mathbf{x}}^\top.$$

Taking the dot product of the first equation with  $\bar{\mathbf{x}}$  and the last equation with  $\mathbf{x}$ :

$$\bar{\mathbf{x}}^\top \mathbf{Ax} = \lambda\bar{\mathbf{x}}^\top \mathbf{x} \quad \text{and} \quad \bar{\mathbf{x}}^\top \mathbf{A}^\top \mathbf{x} = \bar{\lambda}\bar{\mathbf{x}}^\top \mathbf{x}.$$

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, the real part of  $\lambda$  must be zero, indicating  $\lambda$  is either purely imaginary or zero.  $\blacksquare$

**Lemma 10.23: (Odd skew-symmetric determinant)** For any skew-symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , if  $n$  is odd, then  $\det(\mathbf{A}) = 0$ .

**Proof** [of Lemma 10.23] 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  $\det(\mathbf{A}) = 0$ .  $\blacksquare$

**Theorem 10.24: (Block-diagonalization of skew-symmetric matrices)** A real skew-symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be decomposed 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 of 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 10.24] The proof follows from a recursive construction. As usual, we denote the entry  $(i, j)$  of a matrix  $\mathbf{A}$  by  $a_{ij}$ .

**Case 1).** Suppose the first row of  $\mathbf{A}$  is nonzero. Note that  $\mathbf{EAE}^\top$  is skew-symmetric for any matrix  $\mathbf{E}$  if  $\mathbf{A}$  is skew-symmetric. Therefore, both the diagonals of  $\mathbf{A}$  and  $\mathbf{EAE}^\top$  zero. The upper-left  $2 \times 2$  submatrix of  $\mathbf{EAE}^\top$  takes 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 0.19), such that we will exchange the nonzero value, say  $a$ , in the first row to the second column of  $\mathbf{PAP}^\top$ . The upper-left  $2 \times 2$  submatrix of  $\mathbf{PAP}^\top$  becomes

$$(\mathbf{PAP}^\top)_{1:2,1:2} = \begin{bmatrix} 0 & a \\ -a & 0 \end{bmatrix}.$$

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

$$(\mathbf{MPAP}^\top \mathbf{M}^\top)_{1:2,1:2} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

This completes the block-diagonalization of the upper-left  $2 \times 2$  block. Next, if there exists a nonzero value, say  $b$ , in the first row of  $(\mathbf{MPAP}^\top \mathbf{M}^\top)$  at position  $(1, j)$  for some  $j > 2$ , we can construct a nonsingular matrix  $\mathbf{L} = \mathbf{I} - b \cdot \mathbf{E}_{j2}$ , where  $\mathbf{E}_{j2}$  is an all-zero matrix except that the entry  $(j, 2)$  is 1, such that  $\mathbf{L}(\mathbf{MPAP}^\top \mathbf{M}^\top)\mathbf{L}^\top$  will set the entry with value  $b$  to 0.

### A Trivial Example

For example, suppose  $\mathbf{MPAP}^\top \mathbf{M}^\top$  is a  $3 \times 3$  matrix with the following value

$$\mathbf{MPAP}^\top \mathbf{M}^\top = \begin{bmatrix} 0 & 1 & b \\ -1 & 0 & \times \\ \times & \times & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{L} = \mathbf{I} - b \cdot \mathbf{E}_{j2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -b & 1 \end{bmatrix},$$

where  $j = 3$  for this specific example. This results in

$$\mathbf{LMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -b & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & \textcolor{blue}{b} \\ -1 & 0 & \times \\ \times & \times & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & \textcolor{blue}{0} \\ -1 & 0 & \times \\ \times & \times & 0 \end{bmatrix}.$$

Similarly, if the second row of  $\mathbf{LMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top$  contains a nonzero value, say  $c$ , we can construct a nonsingular matrix  $\mathbf{K} = \mathbf{I} + c \cdot \mathbf{E}_{j1}$  such that  $\mathbf{K}(\mathbf{LMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top)\mathbf{K}^\top$  will introduce a zero for the entry with value  $c$ .

### A Trivial Example

For example, suppose  $\mathbf{LMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top$  is a  $3 \times 3$  matrix with the following value

$$\mathbf{LMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & c \\ \times & \times & 0 \end{bmatrix}, \quad \text{and} \quad \mathbf{K} = \mathbf{I} + c \cdot \mathbf{E}_{j1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c & 0 & 1 \end{bmatrix},$$

where  $j = 3$  for this specific example. This results in

$$\mathbf{KLMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top \mathbf{K}^\top = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ c & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & \textcolor{blue}{c} \\ \times & \times & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & \textcolor{blue}{0} \\ \times & \times & 0 \end{bmatrix}.$$

Since we have shown that  $\mathbf{KLMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top \mathbf{K}^\top$  is also skew-symmetric, then it simplifies to

$$\mathbf{KLMPAP}^\top \mathbf{M}^\top \mathbf{L}^\top \mathbf{K}^\top = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

so we do not need to address the first two columns further.

Apply this process iteratively to the bottom-right  $(n-2) \times (n-2)$  submatrix can complete the block-diagonalization.

**Case 2).** If the first row of  $\mathbf{A}$  is zero, we can use a permutation matrix to move the first row to the last row and then proceed with the process described in Case 1 to complete the proof. ■

The block-diagonalization of skew-symmetric matrices, as discussed earlier, demonstrates that the rank of a skew-symmetric matrix is always even. Moreover, we can prove that the determinant of a skew-symmetric matrix of even order is nonnegative, as stated in the following lemma:

**Lemma 10.25: (Even skew-symmetric determinant)** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a skew-symmetric matrix. If  $n$  is even, then  $\det(\mathbf{A}) \geq 0$ .

**Proof** [of Lemma 10.25] Applying Theorem 10.24,  $\mathbf{A}$  can be block-diagonalized as  $\mathbf{A} = \mathbf{ZDZ}^\top$ , resulting in  $\det(\mathbf{A}) = \det(\mathbf{ZDZ}^\top) = \det(\mathbf{Z})^2 \det(\mathbf{D}) \geq 0$ . This completes the proof. ■

## 10.6. Applications in Optimization, Linear Algebra, Machine Learning

### 10.6.1 Application: Variable Separation for Optimization

Consider the quadratic function  $f(\mathbf{x}) = \mathbf{x}^\top \mathbf{Ax} + \mathbf{b}^\top \mathbf{x} + c$ . Unless the symmetric matrix  $\mathbf{A}$  is diagonal, the resulting function contains cross terms of the form  $x_i x_j$ . These are known as *interacting terms*, and they commonly appear in real-world quadratic functions.

It is worth noting that any multivariate quadratic function can be transformed into an *additively separable* function (i.e., one without interacting terms) by applying a suitable linear transformation to the input variables. Additively separable functions are significantly easier to optimize, as the optimization problem can be decomposed into smaller, independent subproblems involving individual variables. For instance, a multivariate quadratic function can be rewritten as a simple sum of univariate quadratic functions, each of which is straightforward to minimize (Aggarwal, 2020). We begin by formally defining the concept of separability:

**Definition 10.26 (Additively separable functions).** A function  $F(x_1, x_2, \dots, x_n)$  of  $n$  variables is said to be *additively separable* if it can be expressed in the following form for appropriately chosen univariate functions  $f_1(\cdot), f_2(\cdot), \dots, f_n(\cdot)$ :

$$F(x_1, x_2, \dots, x_n) = \sum_{i=1}^n f_i(x_i).$$

Now consider the (symmetric) quadratic function defined on an  $n$ -dimensional vector  $\mathbf{x}$ :

$$f(\mathbf{x}) = \mathbf{x}^\top \mathbf{A} \mathbf{x} + \mathbf{b}^\top \mathbf{x} + c.$$

Since  $\mathbf{A}$  is an  $n \times n$  symmetric matrix, it can be diagonalized as  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , and we can perform a variable transformation  $\mathbf{z} = \mathbf{Q}^\top \mathbf{x}$ . Substituting this transformation into the original function yields a new function  $g(\mathbf{z}) = f(\mathbf{Qz})$ , which represents the same function expressed in a different basis. It can be shown that the transformed function becomes:

$$g(\mathbf{z}) = f(\mathbf{Qz}) = \mathbf{z}^\top \Lambda \mathbf{z} + \mathbf{b}^\top \mathbf{Qz} + c.$$

Because  $\Lambda$  is a diagonal matrix, the function  $g(\mathbf{z})$  becomes additively separable. This allows us to solve for  $\mathbf{z}$  using univariate optimization methods and then recover the original variable  $\mathbf{x}$  via  $\mathbf{x} = \mathbf{Qz}$ .

While this method simplifies the optimization process, a key drawback is that computing eigenvectors (as required for diagonalization) can be computationally expensive. To mitigate this, one can generalize the approach by seeking a matrix  $\mathbf{Q}$  (not necessarily orthogonal) such that  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$  for some diagonal matrix  $\Lambda$ . Note that this would not constitute a true diagonalization unless the columns of  $\mathbf{Q}$  are orthonormal and  $\mathbf{Q}^\top = \mathbf{Q}^{-1}$ ; see Definition 10.15 for more details. Nevertheless, such a decomposition is sufficient for constructing a separable quadratic function.

### 10.6.2 Application: Eigenvalue of Projection Matrices

In Section 11.6, we will demonstrate how the QR, UTV, SVD decompositions can be applied to solve the least squares problem. Specifically, we consider the overdetermined system  $\mathbf{Ax} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the data matrix, and  $\mathbf{b} \in \mathbb{R}^m$  is the observation vector, with  $m \geq n$ . Typically,  $\mathbf{A}$  is assumed to have full column rank, as real-world data is often sufficiently diverse to ensure linear independence, or the data can be made linearly independent after preprocessing.

Since  $\mathbf{A}$  has full column rank,  $\mathbf{A}^\top \mathbf{A}$  is invertible, and  $\text{rank}(\mathbf{A}^\top \mathbf{A}) = \text{rank}(\mathbf{A})$ . Therefore, the least squares solution is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ , minimizing  $\|\mathbf{Ax} - \mathbf{b}\|^2$ . The recovered observation vector is  $\hat{\mathbf{b}} = \mathbf{Ax}_{LS} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ . While the observed vector  $\mathbf{b}$  may not lie in the column space of  $\mathbf{A}$ , the recovered vector  $\hat{\mathbf{b}}$  does.

We define the matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  as the (orthogonal) projection matrix, which projects  $\mathbf{b}$  onto the column space of  $\mathbf{A}$ . This matrix is also known as the *hat matrix* because it “puts a hat” on  $\mathbf{b}$ . It is straightforward to verify that  $\mathbf{H}$  is both symmetric ( $\mathbf{H} = \mathbf{H}^\top$ ) and idempotent ( $\mathbf{H}^2 = \mathbf{H}$ ).

**Remark 10.27 (Column space of projection matrices).** The hat matrix  $\mathbf{H} = \mathbf{A}(\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top$  projects any vector in  $\mathbb{R}^m$  onto the column space of  $\mathbf{A}$ , i.e.,  $\mathbf{Hy} \in \mathcal{C}(\mathbf{A})$ . Notably,  $\mathbf{Hy}$  is a linear combination of the columns of  $\mathbf{H}$ , which implies  $\mathcal{C}(\mathbf{H}) = \mathcal{C}(\mathbf{A})$ .

More generally, for any projection matrix  $\mathbf{H}$  that projects vectors onto a subspace  $\mathcal{V}$ , it holds that  $\mathcal{C}(\mathbf{H}) = \mathcal{V}$ . This property can be formally established using the singular value decomposition (Section 11.1).

We now show that any projection matrix has specific eigenvalues.

**Proposition 10.28: (Eigenvalue of projection matrix)** The eigenvalues of a projection matrix are restricted to 0 and 1.

**Proof** [of Proposition 10.28] Since  $\mathbf{H}$  is symmetric, it has a spectral decomposition  $\mathbf{H} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . Using the idempotent property of  $\mathbf{H}$ , we have:

$$(\mathbf{Q}\Lambda\mathbf{Q}^\top)^2 = \mathbf{Q}\Lambda\mathbf{Q}^\top \implies \mathbf{Q}\Lambda^2\mathbf{Q}^\top = \mathbf{Q}\Lambda\mathbf{Q}^\top \implies \Lambda^2 = \Lambda \implies \lambda_i^2 = \lambda_i,$$

Thus, each eigenvalue satisfies  $\lambda_i \in \{0, 1\}$ . ■

This property is significant in the analysis of distribution theory for linear models; see, for example, Lu (2021d). Building on the eigenvalues of the projection matrix, we can also define the *orthogonal complement projection matrix*  $\mathbf{I} - \mathbf{H}$ .

**Proposition 10.29: (Project onto  $\mathcal{V}^\perp$ )** Let  $\mathcal{V}$  be a subspace, and  $\mathbf{H}$  be the projection matrix onto  $\mathcal{V}$ . Then,  $\mathbf{I} - \mathbf{H}$  serves as the projection matrix onto  $\mathcal{V}^\perp$ .

**Proof** [of Proposition 10.29] 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 symmetric. Furthermore, it follows that

$$(\mathbf{I} - \mathbf{H})^2 = \mathbf{I}^2 - \mathbf{I}\mathbf{H} - \mathbf{H}\mathbf{I} + \mathbf{H}^2 = \mathbf{I} - \mathbf{H},$$

which shows that  $\mathbf{I} - \mathbf{H}$  is idempotent. Thus,  $\mathbf{I} - \mathbf{H}$  qualifies as a projection matrix. Using the spectral theorem, write  $\mathbf{H} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . Then,  $\mathbf{I} - \mathbf{H} = \mathbf{Q}\mathbf{Q}^\top - \mathbf{Q}\Lambda\mathbf{Q}^\top = \mathbf{Q}(\mathbf{I} - \Lambda)\mathbf{Q}^\top$ . Consequently, 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 10.28), which aligns with  $\mathcal{V}^\perp$ . ■

For a more detailed discussion of projection matrices and their applications, refer to Lu (2021c). While these results are important, they extend beyond the primary focus of matrix decomposition techniques, and thus will not be repeated here.

### 10.6.3 Application: An Alternative Definition of PD and PSD of Matrices

In Definition 2.2, positive definite (PD) and positive semidefinite (PSD) matrices are defined based on their quadratic forms. Here, we establish that a symmetric matrix is positive definite (resp., positive semidefinite) if and only if all its eigenvalues are positive (resp., nonnegative).

**Lemma 10.30: (Eigenvalues of PD and PSD matrices i.e., the eigenvalue characterization theorem)** A symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive definite (PD) if and only if all eigenvalues of  $\mathbf{A}$  are positive. And a symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is positive semidefinite (PSD) if and only if all eigenvalues of  $\mathbf{A}$  are nonnegative.

**Proof** [of Lemma 10.30] 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  $\lambda > 0$ .

Conversely, suppose all eigenvalues of  $\mathbf{A}$  are positive, and consider the spectral decomposition of  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , where  $\mathbf{Q}$  is orthogonal and  $\Lambda$  is diagonal. Let  $\mathbf{x}$  be any nonzero vector, and let  $\mathbf{y} = \mathbf{Q}^\top \mathbf{x}$ . We have:

$$\mathbf{x}^\top \mathbf{A}\mathbf{x} = \mathbf{x}^\top (\mathbf{Q}\Lambda\mathbf{Q}^\top)\mathbf{x} = (\mathbf{x}^\top \mathbf{Q})\Lambda(\mathbf{Q}^\top \mathbf{x}) = \mathbf{y}^\top \Lambda \mathbf{y} = \sum_{i=1}^n \lambda_i y_i^2 > 0.$$

Thus,  $\mathbf{A}$  is PD. The proof for the PSD case follows similarly. ■

**Theorem 10.31: (Nonsingular factor of PSD and PD matrices)** A real symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is PSD if and only if  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ , where  $\mathbf{P} \in \mathbb{R}^{n \times n}$ ; and it is PD if and only if  $\mathbf{P}$  is nonsingular.

**Proof** [of Theorem 10.31] Suppose  $\mathbf{A}$  is PSD. From its spectral decomposition  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , we can decompose  $\Lambda = \Lambda^{1/2}\Lambda^{1/2}$  (since the eigenvalues of any PSD matrix are nonnegative). Let  $\mathbf{P} = \Lambda^{1/2}\mathbf{Q}^\top$ . Then,  $\mathbf{A}$  can be decomposed as  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ .

Conversely, suppose  $\mathbf{A}$  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

$$\lambda = \frac{\mathbf{v}^\top \mathbf{A} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} = \frac{\mathbf{v}^\top \mathbf{P}^\top \mathbf{P} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}} = \frac{\|\mathbf{P}\mathbf{v}\|^2}{\|\mathbf{v}\|^2} \geq 0.$$

Therefore,  $\mathbf{A}$  is PSD by Lemma 10.30.

Similarly, we can prove the second part for PD matrices, where the positive definiteness will result in the nonsingular  $\mathbf{P}$ ; and the nonsingularity of  $\mathbf{P}$  implies the positivity of the eigenvalues. ■

#### 10.6.4 Proof for Semidefinite Rank-Revealing Decomposition

In this section, we provide an alternative proof for Theorem 2.13, which establishes the existence of a rank-revealing decomposition for positive semidefinite matrices.

**Proof** [of Theorem 2.13] The proof is based on two key results: the nonsingular factorization of PSD matrices (Theorem 10.31) and the column-pivoted QR decomposition (Theorem 3.2).

By Theorem 10.31, any PSD matrix  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{Z}^\top \mathbf{Z}$ , 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 10.8, the rank of  $\mathbf{A}$  equals the number of its nonzero eigenvalues, which corresponds to the positive eigenvalues for a PSD matrix. Consequently, only  $r$  diagonal elements of  $\Lambda^{1/2}$  are nonzero, making  $\mathbf{Z} = \Lambda^{1/2}\mathbf{Q}^\top$  a rank- $r$  matrix with  $r$  linearly independent columns. Applying the column-pivoted QR decomposition to  $\mathbf{Z}$ , we obtain  $\mathbf{Z}\mathbf{P} = \mathbf{Q} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ , 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,

$$\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{P}^\top \mathbf{Z}^\top \mathbf{Z} \mathbf{P} = \begin{bmatrix} \mathbf{R}_{11}^\top & \mathbf{0} \\ \mathbf{R}_{12}^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Let  $\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$ . Thus, the rank-revealing decomposition of the PSD matrix  $\mathbf{A}$  is:  $\mathbf{P}^\top \mathbf{A} \mathbf{P} = \mathbf{R}^\top \mathbf{R}$ . ■

This decomposition is obtained using complete pivoting, where at each step the algorithm selects the largest diagonal element in the active submatrix as the pivot. This strategy is conceptually similar to the partial pivoting technique discussed in Section 1.7.1.

#### 10.6.5 Application: Cholesky via QR and Spectral Decompositions

In this section, we present an alternative proof for the existence of the Cholesky decomposition using the nonsingular factor of PD matrices.

**Proof** [of Theorem 2.1] From Theorem 10.31, the PD matrix  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ , where  $\mathbf{P}$  is a nonsingular matrix. Applying the QR decomposition to  $\mathbf{P}$ , we write  $\mathbf{P} = \mathbf{QR}$ , which implies:

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

This result closely resembles the Cholesky decomposition, with the exception that  $\mathbf{R}$  is not explicitly required to have positive diagonal entries. However, by considering the CGS algorithm for computing the QR decomposition (discussed in Section 3.5), it can be observed that the diagonal entries of  $\mathbf{R}$  are nonnegative. Moreover, if  $\mathbf{P}$  is nonsingular, these diagonal entries are strictly positive.  $\blacksquare$

The proof above relies on the existence of both the QR decomposition and the spectral decomposition. Thus, in this context, the existence of the Cholesky decomposition can be demonstrated using these two fundamental decomposition methods.

### 10.6.6 Application: Unique Power Decomposition of PD Matrices

In this section, we present a *unique power decomposition* for positive definite matrices using their spectral decomposition.

**Theorem 10.32: (Unique power decomposition of PD matrices)** Any  $n \times n$  positive definite matrix  $\mathbf{A}$  can be **uniquely** decomposed as the square of another positive definite matrix  $\mathbf{B}$ , that is,  $\mathbf{A} = \mathbf{B}^2$ .

**Proof** [of Theorem 10.32] We first prove the existence of a positive definite matrix  $\mathbf{B}$  satisfying  $\mathbf{A} = \mathbf{B}^2$  and then demonstrate its uniqueness.

**Existence.** Since  $\mathbf{A}$  symmetric and positive definite, its spectral decomposition is given by  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ . By Lemma 10.30, all eigenvalues of a PD matrix are strictly positive. Hence, the square root of  $\Lambda$  exists, allowing us to define  $\mathbf{B} = \mathbf{Q}\Lambda^{1/2}\mathbf{Q}^\top$ . It follows that  $\mathbf{A} = \mathbf{B}^2$ . Since  $\mathbf{B}$  is symmetric with positive eigenvalues, it is also positive definite.

**Uniqueness.** Suppose the factorization is not unique. Then, there exist two positive definite matrices  $\mathbf{B}_1$  and  $\mathbf{B}_2$  such that

$$\mathbf{A} = \mathbf{B}_1^2 = \mathbf{B}_2^2,$$

where both  $\mathbf{B}_1$  and  $\mathbf{B}_2$  are PD. Their spectral decompositions 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  $\Lambda_2^2$  contain 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

$$\mathbf{Q}_1 \Lambda \mathbf{Q}_1^\top = \mathbf{Q}_2 \Lambda \mathbf{Q}_2^\top \quad \xrightarrow{\text{leads to}} \quad \mathbf{Q}_2^\top \mathbf{Q}_1 \Lambda = \Lambda \mathbf{Q}_2^\top \mathbf{Q}_1.$$

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  $\Lambda$  (Lu, 2021c). 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.  $\blacksquare$

Similarly, we can prove the unique decomposition of a PSD matrix  $\mathbf{A}$  such that  $\mathbf{A} = \mathbf{B}^2$ , where  $\mathbf{B}$  is PSD (Koeber and Schäfer, 2006).

**Decompositions for PD matrices.** To summarize, a PD matrix  $\mathbf{A}$  can be decomposed in several ways: 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 the Cholesky decomposition;  $\mathbf{A} = \mathbf{P}^\top \mathbf{P}$ , where  $\mathbf{P}$  is nonsingular in Theorem 10.31; and  $\mathbf{A} = \mathbf{B}^2$ , where  $\mathbf{B}$  is PD in Theorem 10.32.

### 10.6.7 Application: Feature Engineering for Scatter Matrices

Consider an  $n \times p$  data matrix  $\mathbf{X}$ , where each row  $\mathbf{x}_i$  represents a data point. In machine learning, an  $n \times n$  symmetric *scatter matrix* or *kernel matrix*  $\mathbf{S}$  can be defined among the  $n$  data points as follows:

$$s_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j), \quad \forall i, j,$$

where  $k(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})^\top \phi(\mathbf{y})$  is called a *kernel function*, and  $\phi(\mathbf{x})$  is the associated *basis function*. Thus, the scatter matrix  $\mathbf{S}$  contains all pairwise kernel evaluations between data points. It can be easily shown that  $\mathbf{S}$  must be symmetric and positive semidefinite (Lu, 2021d).

**Exercise 10.33 (Properties of scatter matrices).** Let  $\mathbf{X}$  be an  $n \times p$  data matrix, where each row  $\mathbf{x}_i$  denotes a data point. And let  $\phi(\cdot) : \mathbb{R}^p \rightarrow \mathbb{R}^k$  be a basis function. Show that

- $\mathbf{S}$  is a symmetric matrix, i.e.,  $k(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_j, \mathbf{x}_i)$ .
- $\mathbf{S}$  is positive semidefinite.

At first glance, it may appear that  $k(\mathbf{x}, \mathbf{x}')$  can be any arbitrary function of  $\mathbf{x}$  and  $\mathbf{x}'$ . However, the requirement for the scatter matrix to be positive semidefinite constrains the form of valid kernel functions. This constraint ensures that every valid kernel corresponds to an implicit inner product in some (possibly infinite-dimensional) feature space. The following are examples of widely used kernel functions:

1. *Linear kernel.*  $k(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$ .
2. *Polynomial kernel.*  $k(\mathbf{x}, \mathbf{x}') = (\eta + \gamma \mathbf{x}^\top \mathbf{x}')^Q$  with  $\gamma > 0, \eta \geq 0$ .
3. *Gaussian kernel.*  $k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$ . We now show that the Gaussian kernel corresponds to an infinite-dimensional feature mapping. Without loss of generality, let  $\gamma = 1$ . Then,

$$\begin{aligned} k(\mathbf{x}, \mathbf{x}') &= \exp\{-\|\mathbf{x} - \mathbf{x}'\|^2\} = \exp\{-\mathbf{x}^\top \mathbf{x}\} \exp\{-\mathbf{x}'^\top \mathbf{x}'\} \exp\{2\mathbf{x}^\top \mathbf{x}'\} \\ &\stackrel{\substack{\text{Taylor} \\ \text{expansion}}}{=} \exp\{-\mathbf{x}^\top \mathbf{x}\} \exp\{-\mathbf{x}'^\top \mathbf{x}'\} \exp\left\{\sum_{i=0}^{\infty} \frac{(2\mathbf{x}^\top \mathbf{x}')^i}{i!}\right\} \\ &= \sum_{i=0}^{\infty} \left( \exp\{-\mathbf{x}^\top \mathbf{x}\} \exp\{-\mathbf{x}'^\top \mathbf{x}'\} \sqrt{\frac{2^i}{i!}} \sqrt{\frac{2^i}{i!}} (\mathbf{x})^i \cdot (\mathbf{x}')^i \right) \\ &= \sum_{i=0}^{\infty} \left( \exp\{-\mathbf{x}^\top \mathbf{x}\} \sqrt{\frac{2^i}{i!}} (\mathbf{x})^i \cdot \exp\{-\mathbf{x}'^\top \mathbf{x}'\} \sqrt{\frac{2^i}{i!}} (\mathbf{x}')^i \right) \\ &= \phi(\mathbf{x})^\top \phi(\mathbf{x}'), \end{aligned}$$

where  $\phi(\mathbf{x}) = \sum_{i=0}^{\infty} \exp\{-\mathbf{x}^\top \mathbf{x}\} \sqrt{\frac{2^i}{i!}} (\mathbf{x})^i$ . This shows that the Gaussian kernel maps inputs from a finite-dimensional space to an infinite-dimensional space. A similar derivation holds for general  $\gamma > 0$ .

Given the data matrix  $\mathbf{X}$  and the basis function  $\phi$ , it is easy to transform the data matrix  $\mathbf{X}$  into its corresponding scatter matrix  $\mathbf{S}$ . However, recovering the original data  $\mathbf{X}$  from the scatter matrix  $\mathbf{S}$  is more complex. The recovery process cannot be unique due to the invariance of dot products under rotation and reflection. For example, consider a  $p \times p$  orthogonal matrix  $\mathbf{U}$ , which acts as a rotation/reflection matrix. Then, the rotated/reflected version of  $\mathbf{X}$  is  $\tilde{\mathbf{X}} = \mathbf{X}\mathbf{U}$ . Consequently, the scatter matrix  $\tilde{\mathbf{S}}$  using  $\tilde{\mathbf{X}}$  can be shown to be equal to  $\mathbf{S}$  as follows:

$$\tilde{\mathbf{S}} = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^\top = (\mathbf{X}\mathbf{U})(\mathbf{X}\mathbf{U})^\top = \mathbf{X}(\mathbf{U}\mathbf{U}^\top)\mathbf{X}^\top = \mathbf{S}.$$

A *symmetric factorization* of an  $n \times n$  matrix is a factorization of  $\mathbf{S}$  into two  $n \times k$  matrices of the form  $\mathbf{S} = \mathbf{W}\mathbf{W}^\top$ . For exact factorization, the value of  $k$  will be equal to the rank of the scatter matrix  $\mathbf{S}$ . The  $i$ -th row of  $\mathbf{W}$  in any symmetric factorization  $\mathbf{W}\mathbf{W}^\top$  of  $\mathbf{S}$  yields a valid set of features of the  $i$ -th data point. The representation of  $\mathbf{W}$  is important because it enables the use of many machine learning algorithms—such as *support vector machines (SVMs)* or *logistic regression*—that operate on multidimensional data.

There are three common methods for performing symmetric factorization:

- Spectral decomposition:  $\mathbf{S} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$ . Since the eigenvalues of a positive semidefinite matrix are nonnegative, we can represent the diagonal matrix as  $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^2$ :  $\mathbf{S} = \mathbf{Q}\boldsymbol{\Sigma}^2\mathbf{Q}^\top = (\mathbf{Q}\boldsymbol{\Sigma})(\mathbf{Q}\boldsymbol{\Sigma})^\top = \mathbf{W}\mathbf{W}^\top$ . Spectral decomposition of the scatter matrix provides one of infinitely many possible representations that can be derived from factorizing  $\mathbf{S}$ . Among these, it is also one of the most compact in terms of the number of nonzero columns. The compactness can be further improved by discarding eigenvectors corresponding to small eigenvalues.
- Symmetric square-root matrix, which can also be extracted from the spectral decomposition as  $\mathbf{S} = \mathbf{Q}\boldsymbol{\Sigma}^2\mathbf{Q}^\top = (\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^\top)(\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^\top)^\top = (\sqrt{\mathbf{S}})^2$ . In this case, we set  $\mathbf{W}$  to be  $\mathbf{Q}\boldsymbol{\Sigma}\mathbf{Q}^\top$ .
- Cholesky factorization:  $\mathbf{S} = \mathbf{L}\mathbf{L}^\top$ , and we set  $\mathbf{W} = \mathbf{L}$ .

In all cases, the  $i$ -th row of  $\mathbf{W}$  contains the *embedded representation* (also referred to as the *hidden or latent representation*) of the  $i$ -th data point. Choosing any of these representations will not affect the predictions made by machine learning algorithms that rely on dot products (or Euclidean distances), since these quantities remain unchanged regardless of whether we use spectral decomposition, Cholesky factorization, or the square-root matrix. For example, see its application in large language model compression (Lu et al., 2025), and in generalized least squares models (Lu, 2021d).

**Kernel clustering.** The kernel representation of  $\mathbf{W}$  is crucial because it enables the use of various machine learning algorithms. Consider a scenario where we have an  $n \times n$  scatter matrix  $\mathbf{S}$  for  $n$  data points, and we aim to cluster these points into similar groups. Using the spectral decomposition as an example, the approach of explicit feature engineering involves diagonalizing the scatter matrix as follows:

1. Diagonalize  $\mathbf{S} = \mathbf{Q}\boldsymbol{\Sigma}^2\mathbf{Q}^\top$ .
2. Extract the  $n$ -dimensional embeddings from the rows of  $\mathbf{Q}\boldsymbol{\Sigma}$ .
3. Remove any zero columns from  $\mathbf{Q}\boldsymbol{\Sigma}$  to form  $\mathbf{Q}_0\boldsymbol{\Sigma}_0$ .
4. Apply a clustering algorithm (e.g., Bayesian GMM, K-Means (Lu, 2021a)) on the rows of  $\mathbf{Q}_0\boldsymbol{\Sigma}_0$ .

In this process, the columns of  $\mathbf{Q}_0$  contain the nonzero eigenvectors, and the  $n$  rows of  $\mathbf{Q}_0\boldsymbol{\Sigma}_0$  represent the latent features of the  $n$  data points.

**Kernel clustering for adjacency matrices.** In addition to its application to scatter matrices, the concept of kernel clustering can also be applied to the *adjacency matrix* of an *undirected graph*.

A *graph*, sometimes referred to as a *network*, is a mathematical structure used to represent “relationships” (i.e., *edges* in the graph) among objects (i.e., *vertices* or *nodes* in the graph). The objects can be of any type—such as web pages, individuals in a social network, or chemical elements—while the relationships depend on the specific application; examples include hyperlinks between web pages, friendships in social networks, or chemical bonds between molecules.

A graph is considered *undirected* when its edges do not have a direction. For example, a Facebook friendship link is undirected. In contrast, a graph is *directed* when its edges do have direction, as in follower-followee relationships on Twitter.

An adjacency matrix  $\mathbf{A}$  is a square matrix used to represent the structure of an undirected graph. Each entry  $a_{ij}$  of the matrix indicates whether there is an edge between node  $i$  and node  $j$ : it takes the value 1 or a positive weight  $w_{ij}$  if such an edge exists, and 0 otherwise. This definition extends naturally to directed graphs. Note that for a directed graph with  $q$  edges, the adjacency matrix will contain exactly  $q$  nonzero entries. In contrast, for an undirected graph with  $q$  edges, the adjacency matrix will have  $2q$  nonzero entries due to symmetry.

Since the adjacency matrix of an undirected graph is symmetric (and possibly indefinite), kernel clustering methods based on spectral decomposition can be directly applied to it. This process, commonly known as *spectral clustering*, allows us to extract similarity information between nodes. However, because the adjacency matrix of a directed graph is asymmetric, these methods cannot be directly extended. We will explore this issue further in Section 12.13.

**Kernel outlier detection.** For outlier detection tasks, the scatter matrix can be utilized in the following way:

1. Diagonalize  $\mathbf{S} = \mathbf{Q}\boldsymbol{\Sigma}^2\mathbf{Q}^\top$ .
2. Extract the  $n$ -dimensional embeddings from the rows of  $\mathbf{Q}\boldsymbol{\Sigma}$ .
3. Remove any zero columns from  $\mathbf{Q}\boldsymbol{\Sigma}$  to form  $\mathbf{Q}_0\boldsymbol{\Sigma}_0$ .
4. Calculate the outlier score for each row of  $\mathbf{Q}_0$  as the  $\ell_2$  distance from the mean of all rows in  $\mathbf{Q}_0$ .

It is important to note that we use  $\mathbf{Q}_0$  instead of  $\mathbf{Q}_0\boldsymbol{\Sigma}_0$  to compute the outlier score for each point. This distinction is critical in outlier detection since outliers often manifest in the deviations along lower-order eigenvectors. If we were to multiply by  $\boldsymbol{\Sigma}_0$ , it would scale down these deviations, potentially making outliers harder to detect (Aggarwal, 2020).

## ~~~~ Chapter 10 Problems ~~~~

1. True or False?
  - If a (square) matrix has all zero eigenvalues, then it must be the zero matrix.
  - If a **symmetric** matrix has all zero eigenvalues, then it must be the zero matrix.
2. Show that the determinant of any diagonalizable matrix equals the product of its eigenvalues.
3. Let  $\mathbf{A}$  be a square and diagonalizable matrix. Consider a situation in which we add  $\lambda$  to each diagonal entry of  $\mathbf{A}$  to create  $\mathbf{B}$ . Show that  $\mathbf{B}$  has the same eigenvectors as  $\mathbf{A}$ , and its eigenvalues are related to  $\mathbf{A}$  by a difference of  $\lambda$ .

4. Show that the eigenvalues of a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  are the same as those of its transpose  $\mathbf{A}^\top$ . *Hint: Use the characteristic polynomial of  $\mathbf{A}$  to prove this.*
5. Let  $\mathbf{A}$  and  $\mathbf{B}$  be symmetric positive definite matrices. Show that the product  $\mathbf{AB}$  may not be symmetric, but its eigenvalues remain positive. *Hint: Take the product of  $\mathbf{B}\mathbf{x}$  and  $\mathbf{AB}\mathbf{x} = \lambda\mathbf{x}$ .*
6. Given a symmetric positive definite matrix  $\mathbf{A}$ , show that  $\mathbf{B}^\top \mathbf{AB}$  is positive definite if  $\mathbf{B}$  contains linearly independent columns.
7. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a symmetric positive definite matrix with eigenvalues ordered as  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ .
  - Find the eigenvalues of  $\lambda\mathbf{I} - \mathbf{A}$ .
  - Prove that  $\lambda\mathbf{I} - \mathbf{A}$  is positive semidefinite.
  - Show that  $\lambda_1 \mathbf{x}^\top \mathbf{x} \geq \mathbf{x}^\top \mathbf{Ax}$  for all  $\mathbf{x} \in \mathbb{R}^n$ .
  - Determine the maximum value of  $\mathbf{x}^\top \mathbf{Ax}/\mathbf{x}^\top \mathbf{x}$ .
8. Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  satisfy  $\mathbf{A}^2 - \mathbf{A} = 2\mathbf{I}$ . Show that  $\mathbf{A}$  is diagonalizable.
9. Suppose that  $\mathbf{A}$  is a diagonalizable matrix, i.e.,  $\mathbf{A}$  can be expressed as  $\mathbf{A} = \mathbf{P}\Lambda\mathbf{P}^{-1}$ . Show that the matrix  $\lim_{k \rightarrow \infty} (\mathbf{I} + \frac{\mathbf{A}}{k})^k$  exists with finite entries. *Hint: Use the fact that  $\lim_{k \rightarrow \infty} (1 + \frac{x}{k})^k = e^x$ .*<sup>4</sup>
10. What can you claim about  $\mathbf{P} \in \mathbb{R}^{m \times n}$  with  $m \geq n$  in Theorem 10.31?
11. Show that two normal matrices are similar if and only if they have the same characteristic polynomial.
12. **Symmetric idempotent.** Let  $\mathbf{A} = \mathbf{A}^\top = \mathbf{A}^2 \in \mathbb{R}^{n \times n}$  with  $\text{rank}(\mathbf{A}) = r$ . Show that there exists an orthogonal matrix  $\mathbf{Q}$  such that  $\mathbf{Q}^\top \mathbf{AQ} = \text{diag}(\mathbf{I}_r, \mathbf{0})$ .
13. **Skew-symmetric.** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be skew-symmetric. Show that  $\text{tr}(\mathbf{A}) = 0$ . Additionally, if  $\mathbf{B} \in \mathbb{R}^{n \times n}$  is symmetric, show that  $\text{tr}(\mathbf{AB}) = 0$ .
14. Show that when  $\mathbf{A}$  and  $\mathbf{B}$  are positive semidefintie, then the condition  $\text{tr}(\mathbf{AB}) = 0$  is equivalent to  $\mathbf{AB} = \mathbf{0}$ . *Hint: The trace is invariant under cyclic permutations, and write out the trace using the elements of matrices from the spectral decomposition.*
15. **Fan's inequality (Fan, 1949; Borwein and Lewis, 2006).** Let  $\mathbf{A}, \mathbf{B}$  be real symmetric, and let  $\lambda^\downarrow(\mathbf{A})$  be the vector containing the eigenvalues of  $\mathbf{A}$  in nonincreasing order. Show that  $\text{tr}(\mathbf{AB}) \leq \lambda^\downarrow(\mathbf{A})^\top \lambda^\downarrow(\mathbf{B})$ . The equality holds if and only if  $\mathbf{A}$  and  $\mathbf{B}$  admit spectral decompositions  $\mathbf{A} = \mathbf{Q}\lambda^\downarrow(\mathbf{A})\mathbf{Q}^\top$  and  $\mathbf{B} = \mathbf{Q}\lambda^\downarrow(\mathbf{B})\mathbf{Q}^\top$  (called *simultaneous ordered spectral decomposition*).<sup>5</sup>
16. **Hardy-Littlewood-Pólya inequality (Borwein and Lewis, 2006).** Let  $[\mathbf{x}]^\downarrow$  denote the vector with the same components of  $\mathbf{x}$  permuted into nonincreasing order. Show that  $\mathbf{x}^\top \mathbf{y} \leq [\mathbf{x}]^\downarrow \cdot [\mathbf{y}]^\downarrow$ . *Hint: Apply Fan's inequality to diagonal matrices.*
17. **Mirsky's theorem.** Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  be symmetric. Let further  $\lambda^\downarrow(\mathbf{A})$  and  $\lambda^\downarrow(\mathbf{B})$  be the vectors containing the eigenvalues of  $\mathbf{A}$  and  $\mathbf{B}$ , respectively, in nonincreasing order. Show that  $\|\text{diag}(\lambda^\downarrow(\mathbf{A})) - \text{diag}(\lambda^\downarrow(\mathbf{B}))\| \leq \|\mathbf{A} - \mathbf{B}\|$  if the matrix norm  $\|\cdot\|$  is *unitarily/orthogonally invariant* (i.e.,  $\|\mathbf{UAV}\| = \|\mathbf{A}\|$  for all orthogonal  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  and for all  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ).
18. Let  $\mathbf{H} = \mathbf{I} - 2\frac{\mathbf{uu}^\top}{\mathbf{u}^\top \mathbf{u}} \in \mathbb{R}^{n \times n}$  be a Householder reflector. Show that the eigenvalue  $\lambda_1 = 1$  has multiplicity  $n - 1$ ; and the eigenvalue  $\lambda_2 = -1$  has multiplicity 1. This implies  $\det(\mathbf{H}) = -1$ .
19. **Eigenvalues of reverse product.** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{B} \in \mathbb{R}^{n \times m}$  with  $m \leq n$ . Show that the  $n$  eigenvalues  $\mathbf{BA}$  are the eigenvalues of  $\mathbf{AB}$  together with  $n - m$

<sup>4</sup>. The result also holds for any square matrix.

<sup>5</sup>. Fan's inequality is a refinement of the Cauchy-Schwarz inequality for symmetric matrices.

- zeros. Hint: Show that  $\begin{bmatrix} \mathbf{AB} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} \end{bmatrix}$  and  $\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{B} & \mathbf{AB} \end{bmatrix}$  are similar, and use Proposition 7.4 to discuss the eigenvalues of the two matrices.
20. **Eigenvalues of rank decomposition.** Consider the rank decomposition of  $\mathbf{A} = \mathbf{DF} \in \mathbb{R}^{n \times n}$  with rank  $r$  (Theorem 5.8). Show that the eigenvalues of  $\mathbf{A}$  are the same as those of  $\mathbf{FD}$  together with  $n - r$  zeros. How does this result change if the decomposition satisfies  $\mathbf{D} \in \mathbb{R}^{n \times k}$  and  $\mathbf{F} \in \mathbb{R}^{k \times n}$  with  $k > r$ ? Hint: Use Problem 10.19.
  21. **Subspace in symmetric.** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be symmetric. Show that  $\mathcal{C}(\mathbf{A}) = \mathcal{C}(\mathbf{A}^k)$  and  $\mathcal{N}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^k)$  for all integers  $k \geq 2$ .
  22. Show that  $\mathbf{A}$  is symmetric  $\iff \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A} & \mathbf{0} \end{bmatrix}$  is symmetric  $\iff \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ -\mathbf{A} & \mathbf{0} \end{bmatrix}$  is skew-symmetric.
  23. We introduced kernel clustering and kernel outlier detection in the main text. Discuss how this process can be applied or adapted for use in classification or regression tasks.
  24. **Sigmoid kernel.** Is the “sigmoid kernel” a valid kernel function:  $k(\mathbf{x}, \mathbf{x}') = \tanh(\kappa \cdot \mathbf{x}^\top \mathbf{x}' - \sigma)$ , where  $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ , and  $\kappa$  and  $\sigma$  are scalars?

# 11

## Singular Value Decomposition (SVD)

### 11.1. Singular Value Decomposition (SVD)

In eigenvalue decomposition, a matrix is typically factored into a diagonal matrix. However, this is not always possible. If the underlying matrix lacks linearly independent eigenvectors, diagonalization cannot be performed. The *singular value decomposition (SVD)* overcomes this limitation. Instead of decomposing a matrix into an eigenvector matrix, SVD breaks it into two orthogonal matrices. We present the result of the SVD in the following theorem and will discuss its existence in later sections.

**Theorem 11.1: (Reduced SVD for rectangular matrices)** Given any real  $m \times n$  matrix  $\mathbf{A}$  of rank  $r$ , the matrix  $\mathbf{A}$  can be decomposed as

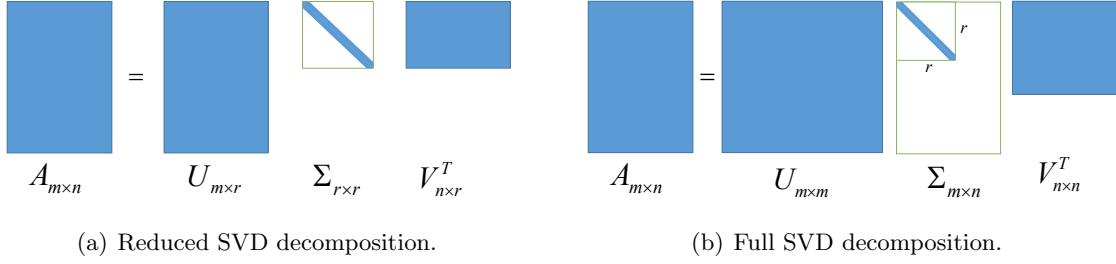
$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top,$$

where  $\Sigma \in \mathbb{R}^{r \times r}$  is a diagonal matrix,  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$  with  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$  and

- The values  $\sigma_i$  are the nonzero *singular values* of  $\mathbf{A}$ ; in the meantime, they are also the (positive) square roots of the nonzero *eigenvalues* of both  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A} \mathbf{A}^\top$ .
- The 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  $\mathbf{A} \mathbf{A}^\top$ .
- The 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  $\mathbf{A}^\top \mathbf{A}$ .
- Additionally, the columns of  $\mathbf{U}$  and  $\mathbf{V}$  are called the *left and right singular vectors* of  $\mathbf{A}$ , respectively.
- Moreover, the columns of both  $\mathbf{U}$  and  $\mathbf{V}$  are mutually orthonormal (by spectral theorem 10.1).

In particular, we can express the matrix decomposition as a 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 represents a sum of  $r$  rank-one matrices.

If we append  $m - r$  additional orthonormal columns to  $\mathbf{U}$ , orthogonal to the  $r$  eigenvectors of  $\mathbf{A} \mathbf{A}^\top$  (similar to the silent columns in the QR decomposition; see Section 3.6), we obtain an orthogonal matrix  $\mathbf{U} \in \mathbb{R}^{m \times m}$ . The same principle applies to the columns of  $\mathbf{V}$ , yielding the *full SVD*. A comparison between the reduced and full SVD is shown in Figure 11.1, where white entries represent zero, and blue entries are not necessarily zero.



**Figure 11.1:** Comparison between the reduced and full SVD.

## 11.2. Existence of the SVD

To prove the existence of the SVD, we need to use the following lemmas. As previously mentioned, the singular values of a matrix  $\mathbf{A}$  are defined as the square roots of the eigenvalues of  $\mathbf{A}^\top \mathbf{A}$ . Since negative values do not have real square roots, it is essential that the eigenvalues of  $\mathbf{A}^\top \mathbf{A}$  must be nonnegative.

**Lemma 11.2: (Nonnegative eigenvalues of  $\mathbf{A}^\top \mathbf{A}$ )** For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , the matrix  $\mathbf{A}^\top \mathbf{A}$  has nonnegative eigenvalues.

**Proof** [of Lemma 11.2] Let  $\lambda$  be an eigenvalue of  $\mathbf{A}^\top \mathbf{A}$  with the corresponding eigenvector  $\mathbf{x}$ . We have  $\mathbf{A}^\top \mathbf{A} \mathbf{x} = \lambda \mathbf{x} \implies \mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} = \lambda \mathbf{x}^\top \mathbf{x}$ . Since  $\mathbf{x}^\top \mathbf{A}^\top \mathbf{A} \mathbf{x} = \|\mathbf{A} \mathbf{x}\|^2 \geq 0$  and  $\mathbf{x}^\top \mathbf{x} > 0$ , we have  $\lambda \geq 0$ . ■

Since  $\mathbf{A}^\top \mathbf{A}$  has nonnegative eigenvalues, we can define the *singular value*  $\sigma \geq 0$  of  $\mathbf{A}$ , such that  $\sigma^2$  is the eigenvalue of  $\mathbf{A}^\top \mathbf{A}$ . In other words,  $\boxed{\mathbf{A}^\top \mathbf{A} \mathbf{v} = \sigma^2 \mathbf{v}}$ . This is a key condition for the existence of the SVD.

We also showed in Lemma 10.7 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 special in that its rank is equal to the rank of  $\mathbf{A}$ . We now prove this result.

**Lemma 11.3: (Rank of  $\mathbf{A}^\top \mathbf{A}$ )** The matrices  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A}$  have same rank. Extending this observation to  $\mathbf{A}^\top$ , we can also prove that  $\mathbf{AA}^\top$  and  $\mathbf{A}$  share the same rank.

**Proof** [of Lemma 11.3] Let  $\mathbf{x} \in \mathcal{N}(\mathbf{A})$ . Then  $\mathbf{Ax} = \mathbf{0} \implies \mathbf{A}^\top \mathbf{Ax} = \mathbf{0}$ , i.e.,  $\mathbf{x} \in \mathcal{N}(\mathbf{A}) \implies \mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ . This shows that  $\mathcal{N}(\mathbf{A}) \subseteq \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ .

Next, let  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$ . We obtain  $\mathbf{A}^\top \mathbf{Ax} = \mathbf{0} \implies \mathbf{x}^\top \mathbf{A}^\top \mathbf{Ax} = 0 \implies \|\mathbf{Ax}\|^2 = 0 \implies \mathbf{Ax} = \mathbf{0}$ . Thus,  $\mathbf{x} \in \mathcal{N}(\mathbf{A}^\top \mathbf{A})$  implies  $\mathbf{x} \in \mathcal{N}(\mathbf{A})$ , and therefore,  $\mathcal{N}(\mathbf{A}^\top \mathbf{A}) \subseteq \mathcal{N}(\mathbf{A})$ .

By combining both inclusions, we conclude 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 (Theorem 0.17), it follows that  $\mathbf{A}^\top \mathbf{A}$  and  $\mathbf{A}$  have the same rank.

Applying the observation to  $\mathbf{A}^\top$ , we can also conclude 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 SVD, we claim that the matrix  $\mathbf{A}$  is a sum of  $r$  rank-one matrices, where  $r$  denotes the number of nonzero singular values. This count of nonzero singular values is, in fact, equal to the rank of the matrix.

**Lemma 11.4: (The number of nonzero singular values vs the rank)** The number of nonzero singular values of a matrix  $\mathbf{A}$  is equal to its rank.

**Proof** [of Lemma 11.4] By Lemma 10.8, the rank of any symmetric matrix (such as  $\mathbf{A}^\top \mathbf{A}$ ) is equal to the number of nonzero eigenvalues (counting multiplicities). Thus, the number of nonzero singular values of  $\mathbf{A}$  equals the rank of  $\mathbf{A}^\top \mathbf{A}$ . By Lemma 11.3, the number of nonzero singular values is therefore also equal to the rank of  $\mathbf{A}$ . ■

We are now ready to prove the existence of the SVD.

**Proof** [of Theorem 11.1: Existence of the reduced SVD] Since  $\mathbf{A}^\top \mathbf{A}$  is a symmetric matrix, by the spectral theorem 10.1 and Lemma 11.2, there exists a semi-orthogonal matrix  $\mathbf{V} \in \mathbb{R}^{n \times r}$  such that  $\mathbf{A}^\top \mathbf{A} = \mathbf{V} \Sigma^2 \mathbf{V}^\top$ , where  $\Sigma$  is a diagonal matrix containing the  $r$  nonzero singular values of  $\mathbf{A}$ , i.e.,  $\Sigma^2$  contains the corresponding nonzero eigenvalues of  $\mathbf{A}^\top \mathbf{A}$ . Specifically,  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r)$ , and the set  $\{\sigma_1^2, \sigma_2^2, \dots, \sigma_r^2\}$  represents the nonzero eigenvalues of  $\mathbf{A}^\top \mathbf{A}$ , where  $r = \text{rank}(\mathbf{A})$ . Now we proceed with the core of the proof.

Starting from the equation  $\boxed{\mathbf{A}^\top \mathbf{A} \mathbf{v}_i = \sigma_i^2 \mathbf{v}_i}$ ,  $\forall i \in \{1, 2, \dots, r\}$ , i.e., the eigenvector  $\mathbf{v}_i$  of  $\mathbf{A}^\top \mathbf{A}$  is corresponding to the eigenvalue  $\sigma_i^2$ :

1. Multiplying 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. Multiplying 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 a unit norm.

These vectors  $\mathbf{u}_i$  are mutually orthonormal because  $(\mathbf{A} \mathbf{v}_i)^\top (\mathbf{A} \mathbf{v}_j) = \mathbf{v}_i^\top \mathbf{A}^\top \mathbf{A} \mathbf{v}_j = \sigma_j^2 \mathbf{v}_i^\top \mathbf{v}_j = 0$  if  $i \neq j$ . Thus, we conclude that  $\mathbf{A} \mathbf{A}^\top = \mathbf{U} \Sigma^2 \mathbf{U}^\top$ , where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r]$ . Since  $\mathbf{A} \mathbf{v}_i = \sigma_i \mathbf{u}_i$ , we have

$$[\mathbf{A} \mathbf{v}_1, \mathbf{A} \mathbf{v}_2, \dots, \mathbf{A} \mathbf{v}_r] = [\sigma_1 \mathbf{u}_1, \sigma_2 \mathbf{u}_2, \dots, \sigma_r \mathbf{u}_r] \quad \xrightarrow{\text{leads to}} \quad \mathbf{A} \mathbf{V} = \mathbf{U} \Sigma. \quad (11.1)$$

At this point, since  $\mathbf{V} \mathbf{V}^\top \neq \mathbf{I}$ , we cannot directly obtain the reduced SVD. However, by appending  $\mathbf{V}$  with additional orthogonal columns, we can construct an orthogonal matrix  $\tilde{\mathbf{V}} = [\mathbf{V}, \mathbf{V}_2]$ , and similarly append  $\mathbf{U}$  with orthonormal columns to form  $\tilde{\mathbf{U}} = [\mathbf{U}, \mathbf{U}_2]$ . This leads to the full SVD (since  $\tilde{\mathbf{V}} \tilde{\mathbf{V}}^\top = \mathbf{I}$ ):

$$\mathbf{A} \tilde{\mathbf{V}} = \tilde{\mathbf{U}} \tilde{\Sigma}, \quad \text{where} \quad \tilde{\Sigma} = \begin{bmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad \xrightarrow{\text{leads to}} \quad \mathbf{A} = \tilde{\mathbf{U}} \tilde{\Sigma} \tilde{\mathbf{V}}^\top.$$

Finally, simplifying the product, we have  $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^\top + \mathbf{U}_2 \cdot \mathbf{0} \cdot \mathbf{V}_2^\top = \mathbf{U} \Sigma \mathbf{V}^\top$ , which is the reduced SVD. This completes the proof. ■

The proof also shows that if  $\mathbf{A} = \mathbf{U} \Sigma \mathbf{V}^\top$  is the reduced SVD of  $\mathbf{A}$ , it follows from (11.1) that  $\mathbf{A} \mathbf{V} \mathbf{V}^\top = \mathbf{A}$ . This implies that  $\mathbf{V} \mathbf{V}^\top$  (where  $\mathbf{V} \in \mathbb{R}^{n \times r}$  is semi-orthogonal) is an (orthogonal) projection matrix that maps each row of  $\mathbf{A}$  onto itself (a projection matrix onto the row space of  $\mathbf{A}$ ).

**SVD-related orthogonal projections.** In the context of the SVD, several important orthogonal projections arise from the four fundamental subspaces. In simple terms, an orthogonal projection matrix has two key properties: it is symmetric and idempotent; see Sections 3.5 and 10.6.2. Such a projection matrix projects any vector onto its column space. Idempotency means that applying the projection twice is the same as applying it once. Symmetry has a geometric interpretation: the projection minimizes the distance between the original vector and its projection, where the projection lies within the column space of the projection matrix. Now suppose  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$  is the full SVD of  $\mathbf{A}$  with rank  $r$ . Consider 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  $\mathbf{V}_r$  consist of the first  $r$  columns of  $\mathbf{U}$  and  $\mathbf{V}$ , respectively. The four orthogonal projections can then be written as:

$$\begin{aligned} \mathbf{V}_r\mathbf{V}_r^\top &= \text{projection onto } \mathcal{C}(\mathbf{A}^\top), & \mathbf{V}_n\mathbf{V}_n^\top &= \text{projection onto } \mathcal{N}(\mathbf{A}), \\ \mathbf{U}_r\mathbf{U}_r^\top &= \text{projection onto } \mathcal{C}(\mathbf{A}), & \mathbf{U}_m\mathbf{U}_m^\top &= \text{projection onto } \mathcal{N}(\mathbf{A}^\top). \end{aligned}$$

These projection matrices allow us to cleanly map vectors onto the four fundamental subspaces of  $\mathbf{A}$ : its column space, row space, null space, and left null space.

**Spectral decomposition of  $\mathbf{AA}^\top$ .** An additional result from the above proof is that the spectral decomposition of  $\mathbf{A}^\top\mathbf{A} = \mathbf{V}\Sigma^2\mathbf{V}^\top$  naturally leads to the spectral decomposition of  $\mathbf{AA}^\top = \mathbf{U}\Sigma^2\mathbf{U}^\top$ , with the same eigenvalues.

**Corollary 11.5: (Eigenvalues of  $\mathbf{A}^\top\mathbf{A}$  and  $\mathbf{AA}^\top$ )** The nonzero eigenvalues of  $\mathbf{A}^\top\mathbf{A}$  and  $\mathbf{AA}^\top$  are identical.

We have shown in Lemma 11.2 that the eigenvalues of  $\mathbf{A}^\top\mathbf{A}$  are nonnegative. Therefore, the eigenvalues of  $\mathbf{AA}^\top$  must also be nonnegative.

**Corollary 11.6: (Nonnegative eigenvalues of  $\mathbf{A}^\top\mathbf{A}$  and  $\mathbf{AA}^\top$ )** The eigenvalues of both  $\mathbf{A}^\top\mathbf{A}$  and  $\mathbf{AA}^\top$  are nonnegative.

Extending Lemma 11.4, the existence of the SVD is also crucial for defining the effective rank of a matrix.

**Definition 11.7 (Effective rank vs exact rank).** The *effective rank*, also known as the *numerical rank*, is defined as follows: From Lemma 11.4, we know that the number of nonzero singular values of a matrix is equal to its rank. Let the  $i$ -th largest singular value of  $\mathbf{A}$  be denoted as  $\sigma_i(\mathbf{A})$ . If  $\sigma_r(\mathbf{A}) \gg \sigma_{r+1}(\mathbf{A}) \approx 0$ , then  $r$  is called the numerical rank of  $\mathbf{A}$ . In contrast, if  $\sigma_i(\mathbf{A}) > \sigma_{r+1}(\mathbf{A}) = 0$ , we say that  $\mathbf{A}$  has *exact rank*  $r$ , as discussed in most of our previous examples.

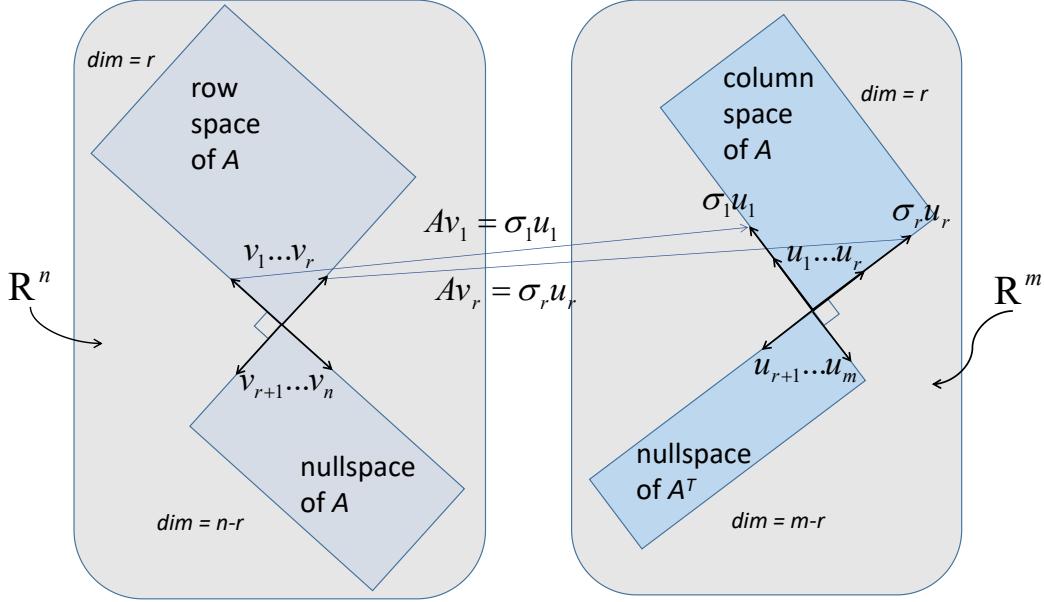
### 11.3. Properties of the SVD

#### 11.3.1 Four Subspaces in SVD

For any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , the following properties hold:

- The null space  $\mathcal{N}(\mathbf{A})$  is the orthogonal complement of the row space  $\mathcal{C}(\mathbf{A}^\top)$  in  $\mathbb{R}^n$ :  $\dim(\mathcal{N}(\mathbf{A})) + \dim(\mathcal{C}(\mathbf{A}^\top)) = n$ .
- The left null space  $\mathcal{N}(\mathbf{A}^\top)$  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 result is known as the fundamental theorem of linear algebra, also referred to as the rank-nullity theorem (Theorem 0.17). Using the SVD, we can identify an orthonormal basis for each of these subspaces.



**Figure 11.2:** Orthonormal bases that diagonalize  $\mathbf{A}$  using the SVD.

**Proposition 11.8: (Four orthonormal bases)** Given the full SVD of a 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  $\mathbf{V}$ , respectively, the following properties hold:

- $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  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 among these four subspaces is shown in Figure 11.2, where  $\mathbf{A}$  maps the row basis  $\mathbf{v}_i$  to the column basis  $\mathbf{u}_i$  via the equation  $\sigma_i \mathbf{u}_i = \mathbf{A} \mathbf{v}_i$  for all  $i \in \{1, 2, \dots, r\}$ .

**Proof [of Proposition 11.8]** From Lemma 10.8, for the symmetric matrix  $\mathbf{A}^\top \mathbf{A}$ , the subspace  $\mathcal{C}(\mathbf{A}^\top \mathbf{A})$  is spanned by the eigenvectors, thus  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  forms an orthonormal basis for  $\mathcal{C}(\mathbf{A}^\top \mathbf{A})$ . We proceed as follows:

1. Since  $\mathbf{A}^\top \mathbf{A}$  is symmetric, the row space of  $\mathbf{A}^\top \mathbf{A}$  is equal to its column space.
2. All rows of  $\mathbf{A}^\top \mathbf{A}$  are linear combinations of the rows of  $\mathbf{A}$ , meaning 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 11.3, we then have:  
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 for  $\mathcal{C}(\mathbf{A}^\top)$ .

Moreover, the space spanned by  $\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}$  is the orthogonal complement to the space spanned by  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$ . Hence,  $\{\mathbf{v}_{r+1}, \mathbf{v}_{r+2}, \dots, \mathbf{v}_n\}$  forms an orthonormal basis for  $\mathcal{N}(\mathbf{A})$ .

Applying a similar argument to  $\mathbf{A}\mathbf{A}^\top$  proves the remaining claims. Alternatively, we can observe that  $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r\}$  forms a basis for the column space of  $\mathbf{A}$  by Lemma 0.16<sup>1</sup>, since  $\mathbf{u}_i = \frac{\mathbf{A}\mathbf{v}_i}{\sigma_i}$  for all  $i \in \{1, 2, \dots, r\}$ . ■

### 11.3.2 Relationship between Singular Values and Determinant

Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be a square matrix, and let its SVD be 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 are nonzero, then  $\det(\mathbf{A}) \neq 0$ . That is,  $\mathbf{A}$  is **nonsingular**. If at least one singular value is zero, say  $\sigma_i = 0$ , then  $\det(\mathbf{A}) = 0$ , implying that  $\mathbf{A}$  does not have full rank and is not invertible. In this case,  $\mathbf{A}$  is called **singular**. This explains why the values  $\sigma_i$  are referred to as the *singular values* of  $\mathbf{A}$ .

### 11.3.3 Orthogonally Equivalence

We have defined in Definition 7.3 that  $\mathbf{A}$  and  $\mathbf{P}\mathbf{A}\mathbf{P}^{-1}$  are similar matrices for any nonsingular matrix  $\mathbf{P}$ . The concept of *orthogonally equivalence* is defined in a similar way.

**Definition 11.9 (Orthogonally equivalent matrices).** Given two orthogonal matrices  $\mathbf{U}$  and  $\mathbf{V}$ , the matrices  $\mathbf{A}$  and  $\mathbf{UAV}$  are called *orthogonally equivalent matrices*. In the complex domain, when  $\mathbf{U}$  and  $\mathbf{V}$  are unitary matrices, the matrices are called *unitarily equivalent*.

We now state the following property for orthogonally equivalent matrices:

**Lemma 11.10: (Orthogonally equivalent matrices)** If matrices  $\mathbf{A}$  and  $\mathbf{B}$  are orthogonally equivalent, then they have the same singular values.

**Proof** [of Lemma 11.10] Since  $\mathbf{A}$  and  $\mathbf{B}$  are orthogonally equivalent, there exist orthogonal matrices  $\mathbf{U}$  and  $\mathbf{V}$  such 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  $\mathbf{BB}^\top$  and  $\mathbf{AA}^\top$  are similar matrices. By Proposition 7.4, the eigenvalues of similar matrices are the same, which implies that the singular values of  $\mathbf{A}$  and  $\mathbf{B}$  are the same. ■

### 11.3.4 SVD for QR

**Lemma 11.11: (SVD for QR)** Suppose the full QR decomposition of a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , with  $m \geq n$ , is given by  $\mathbf{A} = \mathbf{QR}$ , 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.

1. For any matrix  $\mathbf{A}$ , if  $\{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_r\}$  forms a basis for the row space, then  $\{\mathbf{Ar}_1, \mathbf{Ar}_2, \dots, \mathbf{Ar}_r\}$  forms a basis for the column space of  $\mathbf{A}$ .

**Proof** [of Lemma 11.11] We observe 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. Consequently,  $\mathbf{A}$  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 implies that the SVD of a matrix can be derived from its QR decomposition. 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  $\mathbf{R} = \mathbf{U}_0 \boldsymbol{\Sigma} \mathbf{V}^\top$ . Then, the SVD of  $\mathbf{A}$  can be expressed as:  $\mathbf{A} = \underbrace{\mathbf{Q}\mathbf{U}_0}_{\mathbf{U}} \boldsymbol{\Sigma} \mathbf{V}^\top$ .

## 11.4. Polar Decomposition

A decomposition closely related to the SVD is the *polar form* or *polar decomposition* of a matrix. In the context of continuum mechanics, it is imperative to distinguish between stretching and rotation. The polar decomposition factors any matrix into an orthogonal matrix (which corresponds to a rotation or reflection) and a symmetric PSD matrix (which corresponds to stretching or compression, see Section 11.5).

**Theorem 11.12: (Polar decomposition)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$ . Then  $\mathbf{A}$  can be factored as

- **Case  $m > n$ : left polar decomposition.**  $\mathbf{A} = \mathbf{Q}_l \mathbf{S}_l$ , where  $\mathbf{S}_l^2 = \mathbf{A}^\top \mathbf{A}$  is PSD and is **uniquely** determined. The factor  $\mathbf{Q}_l$  has orthonormal columns, and it is **uniquely** determined if  $\text{rank}(\mathbf{A}) = n$ .
- **Case  $m < n$ : right polar decomposition.**  $\mathbf{A} = \mathbf{S}_r \mathbf{Q}_r$ , where  $\mathbf{S}_r^2 = \mathbf{A} \mathbf{A}^\top$  is PSD and is **uniquely** determined. The factor  $\mathbf{Q}_r$  has orthonormal rows, and it is **uniquely** determined if  $\text{rank}(\mathbf{A}) = m$ .
- **Case  $m = n$ : left/right polar decomposition.**  $\mathbf{A} = \mathbf{Q} \mathbf{S}_l = \mathbf{S}_r \mathbf{Q}$ , where  $\mathbf{S}_l^2 = \mathbf{A}^\top \mathbf{A}$  and  $\mathbf{S}_r^2 = \mathbf{A} \mathbf{A}^\top$  are PSD and are **uniquely** determined. The factor  $\mathbf{Q}$  is orthonoal, and it is the same for both the left and right polar decompositions.  $\mathbf{Q}$  is **uniquely** determined if  $\mathbf{A}$  is nonsingular (i.e.,  $\text{rank}(\mathbf{A}) = n$ ).

Note in all cases, the PSD factors ( $\mathbf{S}_l$  or  $\mathbf{S}_r$ ) are uniquely determined, and become PD if  $\mathbf{A}$  has full rank (full row or column rank). The semi-orthogonal factors  $\mathbf{Q}_l$ ,  $\mathbf{Q}_r$ , and  $\mathbf{Q}$  are uniquely determined only when  $\mathbf{A}$  has full rank. <sup>a</sup>

- a. When  $\mathbf{A}$  is complex, then the orthogonal (resp., semi-orthogonal) matrices become unitary (resp., semi-unitary) matrices, and the PSD matrices become complex Hermitian and PSD matrices.

**Proof** [of Theorem 11.12] Let the SVD of  $\mathbf{A}$  be  $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = (\mathbf{U}\mathbf{V}^\top)(\mathbf{V}\boldsymbol{\Sigma}\mathbf{V}^\top) = \mathbf{Q}_l \mathbf{S}_l$  such that  $\mathbf{S}_l^2 = \mathbf{V}\boldsymbol{\Sigma}^2\mathbf{V}^\top = \mathbf{A}^\top \mathbf{A}$ . Since  $\mathbf{A}^\top \mathbf{A}$  is PSD,  $\mathbf{S}_l$  is uniquely determined (Theorem 10.32). If further  $\text{rank}(\mathbf{A}) = n$ , i.e.,  $\mathbf{A}$  has full (column) rank,  $\mathbf{A}^\top \mathbf{A}$  is PD and  $\mathbf{S}_l$  has full rank (Theorem 10.32) such that  $\mathbf{Q}_l = \mathbf{A}\mathbf{S}_l^{-1}$ , implying  $\mathbf{Q}_l$  is uniquely determined.

The second case can be similarly proved such that  $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top = (\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^\top)(\mathbf{U}\mathbf{V}^\top) = \mathbf{S}_r \mathbf{Q}_r$ . Since  $\mathbf{A}\mathbf{A}^\top$  is PSD,  $\mathbf{S}_r$  is uniquely determined. If further  $\text{rank}(\mathbf{A}) = m$ ,  $\mathbf{A}\mathbf{A}^\top$  is PD and  $\mathbf{S}_r$  has full rank such that  $\mathbf{Q}_r = \mathbf{S}_r^{-1}\mathbf{A}$  is uniquely determined.

The third case is a combination of the previous two. This completes the proof. ■

**Exercise 11.13 (Trace of PSD in polar decomposition).** Show that the trace of the PSD matrices in the polar decomposition,  $\text{tr}(\mathbf{S}_r)$  or  $\text{tr}(\mathbf{S}_l)$ , is equal to the sum of the singular values of  $\mathbf{A}$ .

**Exercise 11.14 (Normal from polar).** Let  $\mathbf{A} \in \mathbb{R}^{n \times n}$  be nonsingular, and suppose it admits the polar decomposition  $\mathbf{A} = \mathbf{S}_r \mathbf{Q}$ , where  $\mathbf{S}_r$  is PD and  $\mathbf{Q}$  is orthogonal. Show that  $\mathbf{A}$  is normal if and only if  $\mathbf{S}_r \mathbf{Q} = \mathbf{Q} \mathbf{S}_r$ .

**Exercise 11.15.** Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$  be orthogonal, and let  $\mathbf{A} + \mathbf{B}$  be nonsingular. Show that the orthogonal factor in the polar decomposition of  $\mathbf{A} + \mathbf{B}$  is  $\mathbf{A}(\mathbf{A}^\top \mathbf{B})^{1/2}$ .

## 11.5. Coordinate Transformation in Matrix Decomposition

Consider a vector  $\mathbf{v} \in \mathbb{R}^3$  with elements  $\mathbf{v} = [3, 7, 2]^\top$ . It is essential to clarify the significance of these values: In the Cartesian coordinate system, they represent a component of 3 along the  $x$ -axis, a component of 7 along the  $y$ -axis, and a component of 2 along the  $z$ -axis. These scalar values are the *coordinates* of  $\mathbf{v}$  with respect to the basis of the Cartesian system. Matrix multiplication, on the other hand, gains significance when applied in high-dimensional spaces.

**Coordinate defined by a nonsingular matrix.** Suppose we have a  $3 \times 3$  nonsingular matrix  $\mathbf{B}$ , which is invertible and possesses linearly independent columns. Consequently, the three columns of  $\mathbf{B}$  collectively form a basis for the  $\mathbb{R}^3$  space. Taking a step further, the three columns of  $\mathbf{B}$  can serve as the basis for a **new coordinate system**, referred to as the  **$B$  coordinate system**.

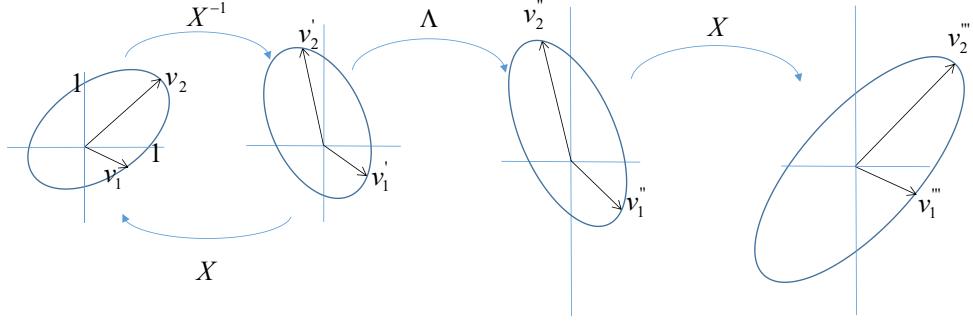
Returning to the Cartesian coordinate system, we also have a set of three vectors forming a basis, denoted by  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ . If we arrange the three vectors as columns in a matrix, this matrix will be the identity matrix. Therefore, when we multiply a vector  $\mathbf{v}$  by the identity matrix, denoted by  $\mathbf{I}\mathbf{v}$ , we are essentially performing a coordinate transformation that leaves  $\mathbf{v}$  in the same coordinate system. In other words,  $\mathbf{I}\mathbf{v} = \mathbf{v}$  means **transferring  $\mathbf{v}$  from the Cartesian coordinate system into the Cartesian coordinate system**, the same coordinate.

Similarly, when we multiply a vector  $\mathbf{v}$  by the matrix  $\mathbf{B}$ , denoted by  $\mathbf{B}\mathbf{v}$ , we are **transforming  $\mathbf{v}$  from the Cartesian coordinate system into the  $B$  coordinate system**. To illustrate this with a specific example, consider  $\mathbf{v} = [3, 7, 2]^\top$  and  $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3]$ . In this case, we have  $\mathbf{u} = \mathbf{B}\mathbf{v} = 3\mathbf{b}_1 + 7\mathbf{b}_2 + 2\mathbf{b}_3$ , i.e., vector  $\mathbf{u}$  contains 3 units of the first basis  $\mathbf{b}_1$  of  $\mathbf{B}$ , 7 units of the second basis  $\mathbf{b}_2$  of  $\mathbf{B}$ , and 2 units of the third basis  $\mathbf{b}_3$  of  $\mathbf{B}$ . Now, if we wish to transform the vector  $\mathbf{u}$  from the  $B$  coordinate system back to the Cartesian coordinate system, we can achieve this by multiplying  $\mathbf{u}$  by the inverse of  $\mathbf{B}$ , denoted by  $\mathbf{B}^{-1}$ . This operation results in  $\mathbf{B}^{-1}\mathbf{u} = \mathbf{v}$ .

**Coordinate defined by an orthogonal matrix.** A  $3 \times 3$  orthogonal matrix  $\mathbf{Q}$  defines a “better” coordinate system since its three columns, forming the basis, are mutually orthonormal (same as those in the Cartesian coordinate system). The operation  $\mathbf{Q}\mathbf{v}$  facilitates the transition of  $\mathbf{v}$  from the Cartesian coordinate system to the one defined by the orthogonal matrix. Since the basis vectors from the orthogonal matrix exhibit orthonormality, just like the three vectors  $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3$  in the Cartesian coordinate system, the transformation induced by the orthogonal matrix involves rotating or reflecting the Cartesian system. To revert to the Cartesian coordinate system, one can utilize  $\mathbf{Q}^{-1} = \mathbf{Q}^\top$ .

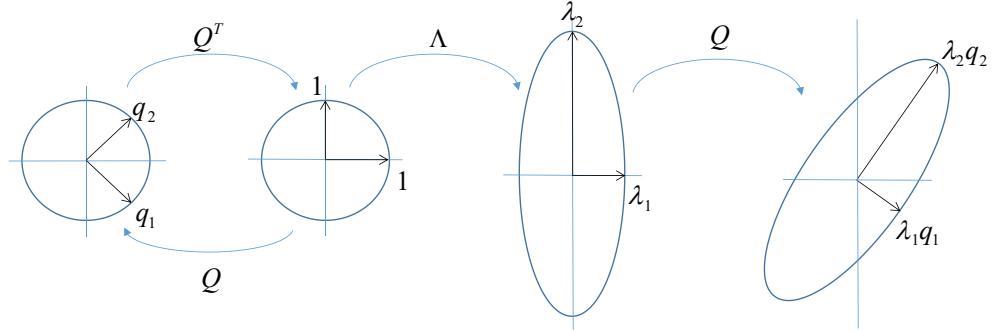
## Eigenvalue Decomposition

A 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  $\mathbf{X}$  and  $\mathbf{X}^{-1}$  are nonsingular so that they define a system transformation



**Figure 11.3:** Eigenvalue decomposition  $\mathbf{A} = \mathbf{X}\Lambda\mathbf{X}^{-1}$ :  $\mathbf{X}^{-1}$  undergoes a transformation into a different coordinate system, followed by stretching with  $\Lambda$ , and then transforming back with  $\mathbf{X}$ .  $\mathbf{X}^{-1}$  and  $\mathbf{X}$  are nonsingular, which will change the basis of the system, and the angle between the vectors  $v_1$  and  $v_2$  will **not** be preserved. In other words, the angle between  $v_1$  and  $v_2$  is **different** from the angle between  $v_1'$  and  $v_2'$ . The lengths of  $v_1$  and  $v_2$  are also **not** preserved; that is,  $\|v_1\|_2 \neq \|v_1'\|_2$  and  $\|v_2\|_2 \neq \|v_2'\|_2$ .

inherently. The operation  $\mathbf{A}\mathbf{u} = \mathbf{X}\Lambda\mathbf{X}^{-1}\mathbf{u}$  firstly transfers  $\mathbf{u}$  into the coordinate system defined by  $\mathbf{X}^{-1}$ , which we shall refer to as the *eigen coordinate system*. Subsequently, the operation  $\Lambda(\cdot)$  stretches each component of the vector in the eigen system by the length of the corresponding eigenvalue. Finally,  $\mathbf{X}$  facilitates the transformation of the resultant vector back to the Cartesian coordinate system. The overall result is an *anisotropic* scaling in  $n$  eigenvector directions. A visual representation of the coordinate system transformation via eigenvalue decomposition is presented in Figure 11.3, where  $v_1$  and  $v_2$  are two linearly independent eigenvectors of  $\mathbf{A}$  such that they form a basis for  $\mathbb{R}^2$ .



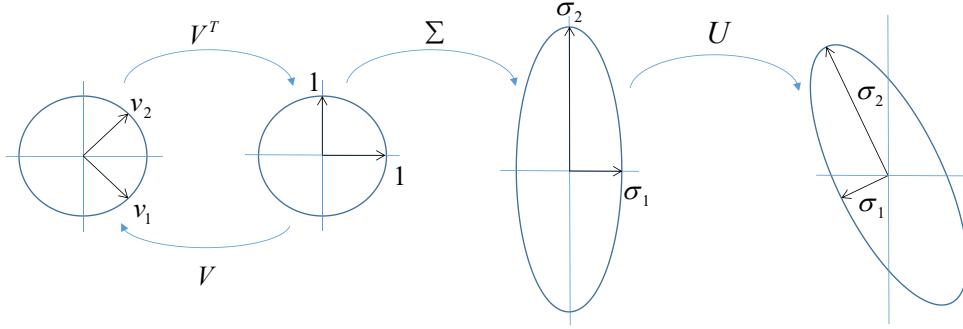
**Figure 11.4:** Spectral decomposition  $\mathbf{Q}\Lambda\mathbf{Q}^\top$ :  $\mathbf{Q}^\top$  rotates or reflects,  $\Lambda$  stretches the cycle to an ellipse, and  $\mathbf{Q}$  rotates or reflects back. Orthogonal matrices  $\mathbf{Q}^\top$  and  $\mathbf{Q}$  only change the basis of the system. However, they preserve both the angle between the vectors  $q_1$  and  $q_2$ , and their lengths.

### Spectral Decomposition

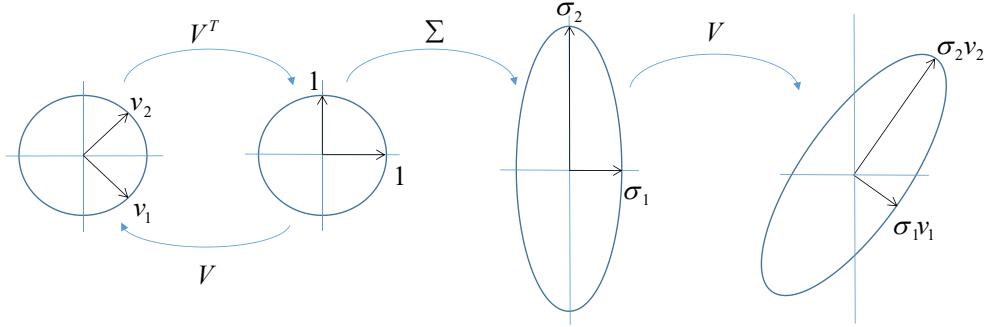
A symmetric matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be decomposed as  $\mathbf{A} = \mathbf{Q}\Lambda\mathbf{Q}^\top$ , where  $\mathbf{Q}$  and  $\mathbf{Q}^\top$  are orthogonal matrices so that they define a coordinate system transformation inherently as well. The operation  $\mathbf{A}\mathbf{u} = \mathbf{Q}\Lambda\mathbf{Q}^\top\mathbf{u}$  firstly rotates or reflects  $\mathbf{u}$  into the coordinate system defined by  $\mathbf{Q}^\top$ , which we shall refer to as the *spectral coordinate system*. The operation

$\Lambda(\cdot)$  stretches each component of the vector in the spectral system by the length of the corresponding eigenvalue. Subsequently,  $\mathbf{Q}$  facilitates the rotation or reflection of the resultant vector back to the original coordinate system. Once again, the overall result is an anisotropic scaling in  $n$  eigenvector directions. A demonstration of how the spectral decomposition transforms between coordinate systems in the  $\mathbb{R}^2$  space is shown in Figure 11.4, where  $\mathbf{q}_1$  and  $\mathbf{q}_2$  represent 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 in the eigenvalue decomposition, with the distinction that in the spectral decomposition, orthogonal vectors transformed by  $\mathbf{Q}^\top$  remain 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 while preserving the angle (inner product) between vectors:  $\mathbf{u}^\top \mathbf{v} = (\mathbf{Q}\mathbf{u})^\top (\mathbf{Q}\mathbf{v})$ . The invariance of the angle between vectors also relies on the invariance of their lengths:  $\|\mathbf{Q}\mathbf{u}\| = \|\mathbf{u}\|$ .

## SVD



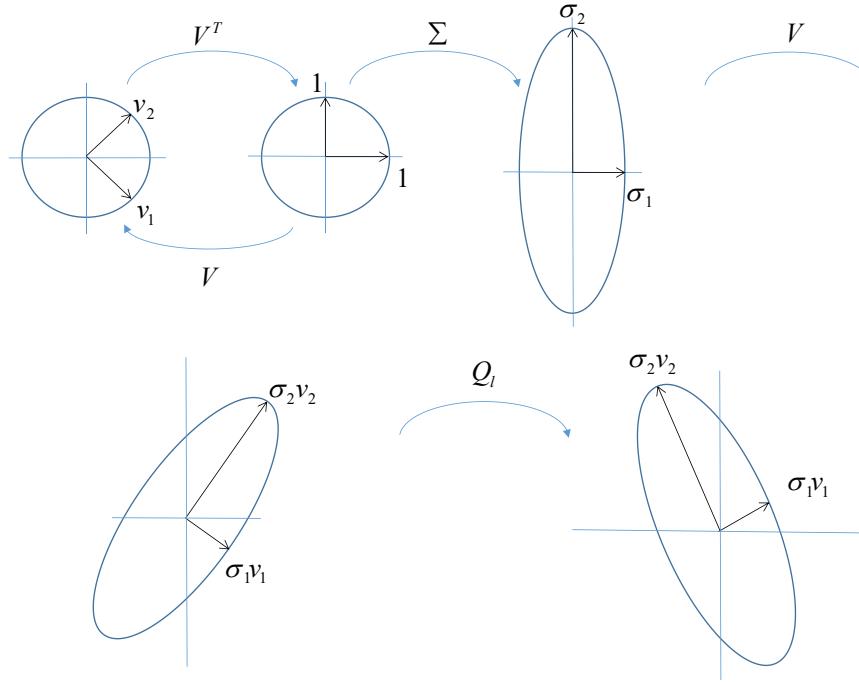
**Figure 11.5:** SVD  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ :  $\mathbf{V}^\top$  and  $\mathbf{U}$  rotate or reflect,  $\Sigma$  stretches the circle to an ellipse. Orthogonal matrices  $\mathbf{V}^\top$  and  $\mathbf{U}$  only change the basis of the system. However, they preserve both the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , and their lengths.



**Figure 11.6:**  $\mathbf{V}\Sigma\mathbf{V}^\top$  from SVD or polar decomposition:  $\mathbf{V}^\top$  rotates or reflects,  $\Sigma$  stretches the cycle to an ellipse, and  $\mathbf{V}$  rotates or reflects back. Orthogonal matrices  $\mathbf{V}^\top$  and  $\mathbf{V}$  only change the basis of the system. However, they preserve both the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , and their lengths.

Any  $m \times n$  matrix of rank  $r$  can be factored as  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , which represents the SVD. The operation  $\mathbf{A}\mathbf{u} = \mathbf{U}\Sigma\mathbf{V}^\top\mathbf{u}$  then firstly rotates or reflects vector  $\mathbf{u}$  into the system

defined by  $\mathbf{V}^\top$ , which we refer to as the  $V$  coordinate system.  $\Sigma$  stretches the first  $r$  components of the resulting vector in the  $V$  system by the lengths of the singular values. If  $n \geq m$ , then  $\Sigma$  only keeps  $m - r$  additional components, which are scaled to zero, while removing the final  $n - m$  components. If  $m > n$ , then  $\Sigma$  scales  $n - r$  components to zero and also adds  $m - n$  additional zero components. Finally,  $\mathbf{U}$  rotates or reflects the resulting vector into the  $U$  coordinate system defined by  $\mathbf{U}$ . A visual demonstration of how the SVD transforms in a  $2 \times 2$  example is shown in Figure 11.5. Further, Figure 11.6 demonstrates the transformation of  $\mathbf{V}\Sigma\mathbf{V}^\top$  by 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 but preserve the angle between vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ .



**Figure 11.7:** Polar decomposition  $\mathbf{A} = \mathbf{Q}_l \mathbf{S}$ :  $\mathbf{V}^\top$  rotates or reflects,  $\Sigma$  stretches the cycle to an ellipse, and  $\mathbf{V}$  rotates or reflects back. Orthogonal matrices  $\mathbf{V}^\top$ ,  $\mathbf{V}$ , and  $\mathbf{Q}_l$  only change the basis of the system. However, they preserve both the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ , and their lengths.

## Polar Decomposition

Any square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  can be factored as the left polar decomposition  $\mathbf{A} = (\mathbf{U}\mathbf{V}^\top)(\mathbf{V}\Sigma\mathbf{V}^\top) = \mathbf{Q}_l \mathbf{S}$ . Similarly, the operation  $\mathbf{A}\mathbf{u} = \mathbf{Q}_l(\mathbf{V}\Sigma\mathbf{V}^\top)\mathbf{u}$  transforms  $\mathbf{u}$  into the system defined by  $\mathbf{V}^\top$ , and stretch each component by the lengths of the corresponding singular values. Subsequently, the resulting 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 right polar decomposition carries a similar interpretation. Similar to the spectral decomposition, orthogonal matrices  $\mathbf{V}^\top$ ,  $\mathbf{V}$ , and  $\mathbf{Q}_l$  only change the basis of the system but preserve the angle between the vectors  $\mathbf{v}_1$  and  $\mathbf{v}_2$ .

## 11.6. Application: LS via Cholesky, QR, UTV, SVD, Bidiagonalization

In 1801, Gauss predicted the orbit of the asteroid Ceres using the method of *least squares* (*LS*, or *ordinary least squares*). Since then, the principle of least squares has become the standard procedure for the analysis of scientific data, which is also the genesis of a vast array of models in machine learning. We now show how to solve the least squares problems using various decompositional approaches (Lu, 2021d).

### Least Squares via Cholesky Decomposition

Let us consider the overdetermined system  $\mathbf{A}\mathbf{x} = \mathbf{b}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the data matrix, and  $\mathbf{b} \in \mathbb{R}^m$  is the observation vector, with  $m \geq n$ . In most real-world applications,  $\mathbf{A}$  typically has full column rank, either naturally or after preprocessing. The least squares solution is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ , which minimizes  $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$ , 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 classical method for solving a linear least squares problem  $\min_{\mathbf{x}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$  is to form and solve the symmetric *normal equation*  $\mathbf{A}^\top \mathbf{A}\mathbf{x} = \mathbf{A}^\top \mathbf{b}$ , which is derived from minimizing the objective function (the root of the objective function). If  $\text{rank}(\mathbf{A}) = n$ , then  $\mathbf{x} \neq \mathbf{0}$  implies that  $\mathbf{A}\mathbf{x} \neq \mathbf{0}$ . Hence,

$$\mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} > 0, \quad \forall \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{x} \neq \mathbf{0}, \quad (11.2)$$

and  $\mathbf{A}^\top \mathbf{A}$  is positive definite. Conversely, any symmetric positive definite matrix is non-singular. If it were singular, there would be a vector  $\mathbf{x} \neq \mathbf{0}$  such that  $\mathbf{A}\mathbf{x} = \mathbf{0}$ , leading to  $\mathbf{x}^\top \mathbf{A}^\top \mathbf{A}\mathbf{x} = 0$ , which contradicts the positive definiteness.

Substituting the Cholesky factorization  $\mathbf{A}^\top \mathbf{A} = \mathbf{R}^\top \mathbf{R}$  into the normal equation yields  $\mathbf{R}^\top \mathbf{R}\mathbf{x} = \boldsymbol{\alpha}$ , where  $\boldsymbol{\alpha} = \mathbf{A}^\top \mathbf{b}$ . Hence, the solution is obtained by solving two triangular systems:

$$\mathbf{R}^\top \mathbf{u} = \boldsymbol{\alpha}, \quad \mathbf{R}\mathbf{x} = \mathbf{u}. \quad (11.3)$$

This method is easy to implement and often faster than other direct solution methods, e.g., using Gradient descent methods (Lu, 2021d).

When solving a least squares problem, it is often preferable to work with the Cholesky factorization of the cross-product of the augmented matrix  $[\mathbf{A}, \mathbf{b}]$ :

$$\begin{bmatrix} \mathbf{A}^\top \\ \mathbf{b}^\top \end{bmatrix} [\mathbf{A} \quad \mathbf{b}] = \begin{bmatrix} \mathbf{A}^\top \mathbf{A} & \mathbf{A}^\top \mathbf{b} \\ \mathbf{b}^\top \mathbf{A} & \mathbf{b}^\top \mathbf{b} \end{bmatrix}. \quad (11.4)$$

If  $\text{rank}(\mathbf{A}) = n$ , then the Cholesky factor of the cross-product (11.4) takes the following form:

$$\mathbf{S} = \begin{bmatrix} \mathbf{R} & \mathbf{v} \\ \mathbf{0} & \rho \end{bmatrix}, \quad (11.5)$$

which exists even when  $\rho = 0$  (see Theorem 2.11). Forming  $\mathbf{S}^\top \mathbf{S}$  shows that

$$\mathbf{A}^\top \mathbf{A} = \mathbf{R}^\top \mathbf{R}, \quad \mathbf{R}^\top \mathbf{v} = \mathbf{A}^\top \mathbf{b}, \quad \mathbf{b}^\top \mathbf{b} = \mathbf{v}^\top \mathbf{v} + \rho^2.$$

Hence,  $\mathbf{R}$  is the Cholesky factor of  $\mathbf{A}^\top \mathbf{A}$ , and the least squares solution is obtained from  $\mathbf{R}\mathbf{x} = \mathbf{v}$ . Since  $\mathbf{e} = \mathbf{b} - \mathbf{A}\mathbf{x}$  is orthogonal to  $\mathbf{A}\mathbf{x}$  (by orthogonal projections; see Section 10.6.2), we have

$$\|\mathbf{A}\mathbf{x}\|_2^2 = (\mathbf{e} + \mathbf{A}\mathbf{x})^\top \mathbf{A}\mathbf{x} = \mathbf{b}^\top \mathbf{A}\mathbf{x} = \mathbf{b}^\top \mathbf{A}\mathbf{R}^{-1} \mathbf{R}^{-\top} \mathbf{A}^\top \mathbf{b} = \mathbf{v}^\top \mathbf{v},$$

and therefore the residual term satisfies  $\|\mathbf{e}\|_2^2 = \rho^2 = \mathbf{b}^\top \mathbf{b} - \mathbf{v}^\top \mathbf{v}$  and  $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 = \rho$ .

**Cholesky QR factorization.** On the other hand, suppose  $\mathbf{A} \in \mathbb{R}^{m \times n}$  have full column rank, and let  $\mathbf{A}^\top \mathbf{A} = \mathbf{R}^\top \mathbf{R}$  be its Cholesky factorization. Define  $\mathbf{Q}_1 = \mathbf{A}\mathbf{R}^{-1} \in \mathbb{R}^{m \times n}$ . Then,

$$\mathbf{A} = \mathbf{Q}_1 \mathbf{R} \quad \text{and} \quad \mathbf{Q}_1^\top \mathbf{Q}_1 = \mathbf{I}_n \quad (11.6)$$

is the *Cholesky QR factorization* of  $\mathbf{A}$ . The semi-orthogonal factor  $\mathbf{Q}_1$  can be obtained as the unique solution of the lower triangular matrix equation  $\mathbf{R}^\top \mathbf{Q}_1^\top = \mathbf{A}^\top$  using forward substitution. In this setting, the normal equation simplifies to  $\mathbf{R}^\top \mathbf{Q}_1^\top \mathbf{Q}_1 \mathbf{R} \mathbf{x} = \mathbf{R}^\top \mathbf{R} \mathbf{x} = \mathbf{R}^\top \mathbf{Q}_1^\top \mathbf{b}$  or  $\mathbf{R} \mathbf{x} = \mathbf{Q}_1^\top \mathbf{b}$ .

In real arithmetic, the computational cost of this Cholesky QR algorithm is  $\sim 2mn^2 + n^3/3$  flops. More accurate methods for computing the QR factorization (11.6) directly from  $\mathbf{A}$  are described in Sections 3.5, 3.6, and Lu (2021c).

### Least Squares via Full QR Decomposition

Since computing the inverse of a matrix can be computationally expensive, as an alternative, we can use the QR decomposition to find the least squares solution. This approach is more efficient and numerically stable. The method is summarized in the following theorem:

**Theorem 11.16: (LS via QR for full column rank matrix)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , with full rank and  $m \geq n$ , be the data matrix, and  $\mathbf{b} \in \mathbb{R}^m$  be the observation vector. And let  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  be its full QR decomposition, where  $\mathbf{Q} \in \mathbb{R}^{m \times m}$  is orthogonal, and  $\mathbf{R} \in \mathbb{R}^{m \times n}$  is upper triangular, with  $m - n$  additional rows of zeros appended at the bottom:  $\mathbf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}$  and  $\mathbf{R}_1 \in \mathbb{R}^{n \times n}$  is the square upper triangular part of  $\mathbf{R}$ . Then, the LS solution to  $\mathbf{Ax} = \mathbf{b}$  is given by

$$\mathbf{x}_{LS} = \mathbf{R}_1^{-1} \mathbf{c}, \quad \text{where} \quad \mathbf{Q}^\top \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}.$$

**Proof** [of Theorem 11.16] Since  $\mathbf{A} = \mathbf{Q}\mathbf{R}$  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 3.4. It follows that

$$\|\mathbf{Ax} - \mathbf{b}\|_2^2 \stackrel{!}{=} \left\| \mathbf{Q}^\top \mathbf{Ax} - \mathbf{Q}^\top \mathbf{b} \right\|_2^2 = \left\| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \mathbf{x} - \mathbf{Q}^\top \mathbf{b} \right\|_2^2 = \|\mathbf{R}_1 \mathbf{x} - \mathbf{c}\|_2^2 + \|\mathbf{d}\|_2^2,$$

where the equality (!) follows from the invariance of norms under orthogonal transformations. Here,  $\mathbf{c}$  represents the first  $n$  components of  $\mathbf{Q}^\top \mathbf{b}$ , and  $\mathbf{d}$  represents the last  $m - n$  components. The least squares solution is obtained by solving the upper triangular system  $\mathbf{R}_1 \mathbf{x} = \mathbf{c}$ , which can be expressed as  $\mathbf{x}_{LS} = \mathbf{R}_1^{-1} \mathbf{c}$ . ■

In the least squares problem, the rows of  $\mathbf{A}$  represent data samples, and the number of columns,  $n$ , corresponds to the dimension of the variables. In some applications, we may want to add or remove a data point (a row in  $\mathbf{A}$ ), or alternatively, add or remove a variable (a column in  $\mathbf{A}$ ); for instance, when performing variable selection using an *F*-test (Lu, 2021d). By utilizing the update methods described in Sections 3.17 and 3.18, the QR decomposition of the modified matrix  $\mathbf{A}$  can be efficiently updated, rather than recomputed from scratch. Therefore, QR decomposition is particular useful for *online LS problems*, where the solution is updated sequentially as new data arrives, as well as for *feature selection problems*, where variables are added or removed dynamically.

### Least Squares via ULV/URV for Rank-Deficient Matrices

In the previous sections, we introduced the least squares method using the full QR decomposition and the Cholesky decomposition for matrices with full column rank. However, in practice, many matrices are rank-deficient, meaning they do not have full rank. If  $\mathbf{A}$  does not have full column rank,  $\mathbf{A}^\top \mathbf{A}$  is not invertible. To handle such cases, we can use the ULV/URV decomposition to find the least squares solution, as stated in the following theorem.

**Theorem 11.17: (LS via ULV/URV for rank-deficient matrix)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be a matrix of rank  $r$  and  $m \geq n$ . Suppose  $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{V}$  is its full ULV/URV decomposition, where  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  are orthogonal 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 either a lower or upper triangular matrix. Given  $\mathbf{b} \in \mathbb{R}^m$ , the LS solution with the minimal  $\ell_2$  norm to  $\mathbf{Ax} = \mathbf{b}$  is given by

$$\mathbf{x}_{LS} = \mathbf{V}^\top \begin{bmatrix} \mathbf{T}_{11}^{-1} \mathbf{c} \\ \mathbf{0} \end{bmatrix}, \quad \text{where } \mathbf{U}^\top \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}.$$

**Proof** [of Theorem 11.17] Since  $\mathbf{A} = \mathbf{U}\mathbf{T}\mathbf{V}$  is the full UTV decomposition of  $\mathbf{A}$  and  $m \geq n$ , we can write:

$$\begin{aligned} \|\mathbf{Ax} - \mathbf{b}\|^2 &= \left\| \mathbf{U}^\top \mathbf{Ax} - \mathbf{U}^\top \mathbf{b} \right\|^2 = \left\| \mathbf{U}^\top \mathbf{U}\mathbf{T}\mathbf{V}\mathbf{x} - \mathbf{U}^\top \mathbf{b} \right\|^2 \\ &= \left\| \mathbf{T}\mathbf{V}\mathbf{x} - \mathbf{U}^\top \mathbf{b} \right\|^2 = \|\mathbf{T}_{11}\mathbf{e} - \mathbf{c}\|^2 + \|\mathbf{d}\|^2, \end{aligned}$$

where  $\mathbf{c}$  is the first  $r$  components of  $\mathbf{U}^\top \mathbf{b}$ ,  $\mathbf{d}$  is the last  $m - r$  components of  $\mathbf{U}^\top \mathbf{b}$ ,  $\mathbf{e}$  is the first  $r$  components of  $\mathbf{V}\mathbf{x}$ , and  $\mathbf{f}$  is the last  $n - r$  components of  $\mathbf{V}\mathbf{x}$ :

$$\mathbf{U}^\top \mathbf{b} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \quad \mathbf{V}\mathbf{x} = \begin{bmatrix} \mathbf{e} \\ \mathbf{f} \end{bmatrix}.$$

The least squares solution is obtained by performing backward/forward substitution of the upper/lower triangular system  $\mathbf{T}_{11}\mathbf{e} = \mathbf{c}$ , i.e.,  $\mathbf{e} = \mathbf{T}_{11}^{-1}\mathbf{c}$ . To ensure that the solution  $\mathbf{x}$  has the minimal  $\ell_2$  norm,  $\mathbf{f}$  must be zero. Thus,  $\mathbf{x}_{LS} = \mathbf{V}^\top \begin{bmatrix} \mathbf{T}_{11}^{-1}\mathbf{c} \\ \mathbf{0} \end{bmatrix}$ . This completes the proof. ■

**Note on the minimal  $\ell_2$  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 a convex set. 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 the proof above, if we do not set  $\mathbf{f} = \mathbf{0}$ , we can still find other least squares solutions. However, the minimal  $\ell_2$  norm least squares solution is unique. For the full-rank case, as discussed in the previous sections, the least squares solution is always unique and must have the minimal  $\ell_2$  norm (Foster, 2003; Golub and Van Loan, 2013; Lu, 2021d).

### Least Squares via SVD for Rank-Deficient Matrices

In addition to the UTV decomposition, the singular value decomposition (SVD) can also be used to solve rank-deficient least squares problems.

**Theorem 11.18: (LS via SVD for rank-deficient matrix)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be a matrix of rank  $r$  and  $m \geq n$ . Suppose  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$  is its full SVD, where  $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m] \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n] \in \mathbb{R}^{n \times n}$  are orthogonal. Given  $\mathbf{b} \in \mathbb{R}^m$ , the LS solution with the minimal  $\ell_2$  norm to  $\mathbf{Ax} = \mathbf{b}$  is given by

$$\mathbf{x}_{LS} = \sum_{i=1}^r \frac{\mathbf{u}_i^\top \mathbf{b}}{\sigma_i} \mathbf{v}_i = \mathbf{V}\Sigma^+\mathbf{U}^\top \mathbf{b}, \quad (11.7)$$

where the upper-left side of  $\Sigma^+ \in \mathbb{R}^{n \times m}$  is a diagonal matrix, structured as:  $\Sigma^+ = \begin{bmatrix} \Sigma_1^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$  with  $\Sigma_1^+ = \text{diag}(\frac{1}{\sigma_1}, \frac{1}{\sigma_2}, \dots, \frac{1}{\sigma_r})$ .

**Proof** [of Theorem 11.18] To minimize the squared error, we begin by expanding:

$$\begin{aligned} \|\mathbf{Ax} - \mathbf{b}\|^2 &\stackrel{\dagger}{=} \left\| \mathbf{U}^\top \mathbf{Ax} - \mathbf{U}^\top \mathbf{b} \right\|^2 = \left\| \mathbf{U}^\top \mathbf{A} \mathbf{V} \mathbf{V}^\top \mathbf{x} - \mathbf{U}^\top \mathbf{b} \right\|^2 \\ &\stackrel{+}{=} \left\| \Sigma \boldsymbol{\alpha} - \mathbf{U}^\top \mathbf{b} \right\|^2 \stackrel{*}{=} \sum_{i=1}^r (\sigma_i \alpha_i - \mathbf{u}_i^\top \mathbf{b})^2 + \sum_{i=r+1}^m (\mathbf{u}_i^\top \mathbf{b})^2, \end{aligned}$$

where the equality ( $\dagger$ ) follows from the invariance of the norm under orthogonal transformations, the equality (+) follows by setting  $\boldsymbol{\alpha} = \mathbf{V}^\top \mathbf{x}$ , and the equality (\*) follows because  $\sigma_{r+1} = \sigma_{r+2} = \dots = \sigma_m = 0$ . Since  $\mathbf{x}$  only appears in  $\boldsymbol{\alpha}$ , we minimize the expression by setting  $\alpha_i = \mathbf{u}_i^\top \mathbf{b} / \sigma_i$  for all  $i \in \{1, 2, \dots, r\}$ . Any value assigned to  $\alpha_{r+1}, \alpha_{r+2}, \dots, \alpha_n$  will not affect the error term. From the regularization point of view (or to obtain the solution with the smallest  $\ell_2$  norm), we can set them to 0. This yields 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  $\mathbf{A}$ . ■

### Bidiagonal Least Squares and LGK Bidiagonalization

We consider an overdetermined linear system  $\mathbf{Cx} = \mathbf{b}$ , where  $\mathbf{C} \in \mathbb{R}^{m \times (n-1)}$  and  $m \geq (n-1)$ .<sup>2</sup> We are interested in the bidiagonalization (Theorem 8.2) of the augmented matrix  $\mathbf{A} = [\mathbf{b}, \mathbf{C}] \in \mathbb{R}^{m \times n}$ , which admits the following bidiagonal decomposition:

$$\begin{aligned} \mathbf{A} &= \mathbf{UBV}^\top = \mathbf{UB} \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & Q \end{bmatrix}^\top \\ \implies \mathbf{B} &= \mathbf{U}^\top [\mathbf{b}, \mathbf{C}] \begin{bmatrix} 1 & \mathbf{0} \\ \mathbf{0} & Q \end{bmatrix} = [\mathbf{U}^\top \mathbf{b}, \mathbf{U}^\top \mathbf{C}Q] = \begin{bmatrix} b_{11}\mathbf{e}_1 & \mathbf{B}_2 \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \end{aligned} \quad (11.8)$$

<sup>2</sup>. Note we set the matrix dimensions to  $m \times (n-1)$  since we consider the bidiagonal decomposition of an  $m \times n$  augmented matrix. Generally, we can also consider  $\mathbf{C} \in \mathbb{R}^{m \times n}$  with  $m \geq n$ .

where  $b_{11}$  represents the (1,1) entry of  $\mathbf{B}$ ,  $\mathbf{B}_2 = \mathbf{B}[1:n, 2:n] \in \mathbb{R}^{n \times (n-1)}$ , and  $\mathbf{Q} = \mathbf{V}[2:n, 2:n] \in \mathbb{R}^{(n-1) \times (n-1)}$  is orthogonal (see Problem 8.9). We then have

$$\|\mathbf{b} - \mathbf{Cx}\|_2 = \left\| [\mathbf{b}, \mathbf{C}] \begin{bmatrix} 1 \\ -\mathbf{x} \end{bmatrix} \right\|_2 = \left\| \mathbf{U}^\top [\mathbf{b}, \mathbf{C}] \mathbf{V} \mathbf{V}^\top \begin{bmatrix} 1 \\ -\mathbf{x} \end{bmatrix} \right\|_2.$$

Let  $\mathbf{d} = \mathbf{Q}^\top \mathbf{x}$ . Then,

$$\|\mathbf{b} - \mathbf{Cx}\|_2 = \left\| \mathbf{BV}^\top \begin{bmatrix} 1 \\ -\mathbf{x} \end{bmatrix} \right\|_2 = \|b_{11}\mathbf{e}_1 - \mathbf{B}_2 \mathbf{d}\|_2.$$

Therefore, the least squares problem of  $\|\mathbf{b} - \mathbf{Cx}\|_2$  then can be equivalently recovered by finding the least squares solution of  $\|b_{11}\mathbf{e}_1 - \mathbf{B}_2 \mathbf{d}\|_2$  in terms of the variable  $\mathbf{d}$ .

### LGK Bidiagonalization

We now express  $\mathbf{B}_2 = \mathbf{B}[1:n, 2:n]$  as follows:

$$\mathbf{B}_2 = \begin{bmatrix} b_{12} & & \cdots & & \\ b_{22} & b_{23} & & \cdots & \\ & \ddots & \ddots & & \vdots \\ & & b_{n-1,n-1} & b_{n-1,n} & \\ & & & b_{nn} & \end{bmatrix} \implies \mathbf{B}_2^\top = \begin{bmatrix} b_{12} & b_{22} & \cdots & & \\ 0 & b_{23} & b_{33} & & \\ & \ddots & b_{n-1,n-1} & & \\ & & b_{n-1,n} & b_{nn} & \end{bmatrix} \in \mathbb{R}^{(n-1) \times n}.$$

**First step.** From Equation (11.8), we find that  $b_{11} = \|\mathbf{b}\|_2 = \|\mathbf{a}_1\|_2$  if  $b_{11}$  is nonnegative, where  $\mathbf{b} = \mathbf{a}_1$  is the first column of  $\mathbf{A}$ . Additionally, we have:

$$\mathbf{CQ} = \mathbf{U} \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{0} \end{bmatrix} \implies \mathbf{C}^\top [\mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_n] = \mathbf{Q} \mathbf{B}_2^\top, \quad \text{where } \mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m].$$

Let  $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_{n-1}]$  be the column partition of  $\mathbf{Q}$ , and let  $\mathbf{q}_0 = \mathbf{0}$ . We then have

$$\begin{cases} \mathbf{C}^\top \mathbf{u}_i = b_{ii} \mathbf{q}_{i-1} + b_{i,i+1} \mathbf{q}_i \implies b_{i,i+1} \mathbf{q}_i = \mathbf{C}^\top \mathbf{u}_i - b_{ii} \mathbf{q}_{i-1}, & \forall i \in \{1, 2, \dots, n-1\}; \\ \mathbf{C}^\top \mathbf{u}_n = b_{nn} \mathbf{q}_{n-1}. \end{cases}$$

If  $b_{ii}$  and  $\mathbf{q}_{i-1}$  are known,  $b_{i,i+1}$  can be determined as the norm of the right-hand side equation in the above equality:

$$b_{i,i+1} = \pm \left\| \mathbf{C}^\top \mathbf{u}_i - b_{ii} \mathbf{q}_{i-1} \right\|_2, \quad \forall i \in \{1, 2, \dots, n-1\}; \quad (11.9)$$

$$\mathbf{q}_i = \frac{\mathbf{C}^\top \mathbf{u}_i - b_{ii} \mathbf{q}_{i-1}}{b_{i,i+1}}, \quad \text{if } b_{i,i+1} \neq 0, \quad \forall i \in \{1, 2, \dots, n-1\}; \quad (11.10)$$

$$b_{ii} = \pm \left\| \mathbf{C}^\top \mathbf{u}_i - b_{i,i+1} \mathbf{q}_i \right\|_2, \quad \forall i \in \{2, 3, \dots, n-1\}. \quad (11.11)$$

**Second step.** Similarly, from Equation (11.8), we have  $\mathbf{u}_1 = \mathbf{b}/b_{11} \equiv \mathbf{a}_1/b_{11}$  and  $\mathbf{CQ} = \mathbf{U} \begin{bmatrix} \mathbf{B}_2 \\ \mathbf{0} \end{bmatrix}$ . This leads to:

$$\begin{aligned} \mathbf{Cq}_i &= b_{i,i+1} \mathbf{u}_i + b_{i+1,i+1} \mathbf{u}_{i+1}, \quad \forall i \in \{1, 2, \dots, n-1\} \\ \implies \mathbf{u}_{i+1} &= \frac{\mathbf{Cq}_i - b_{i,i+1} \mathbf{u}_i}{b_{i+1,i+1}}, \quad \text{if } b_{i+1,i+1} \neq 0, \quad \forall i \in \{1, 2, \dots, n-1\}. \end{aligned} \quad (11.12)$$

The two steps described above form a recursive algorithm for computing the bidiagonal decomposition of the matrix  $\mathbf{A}$ , and is known as the *LGK bidiagonalization*. The derivation

above is valid when  $m > n$ . A similar approach can be applied when  $n \geq m$ . Simple calculations can show the complexity is  $\sim 4mn^2$  flops to obtain all  $\mathbf{B}$ ,  $\mathbf{U}$ , and  $\mathbf{V}$ , which is more efficient than the standard Golub–Kahan bidiagonalization; see Section 8.2.

---

**Algorithm 10** LGK Bidiagonal Decomposition

---

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

- 1: Initially set  $b_{11} \leftarrow \|\mathbf{a}_1\|_2$ ,  $\mathbf{u}_1 \leftarrow \mathbf{a}_1/b_{11}$ ,  $\mathbf{q}_0 \leftarrow \mathbf{0}$ ;
  - 2: **for**  $i = 1$  to  $n - 1$  **do**
  - 3:      $b_{i,i+1} \leftarrow \pm \|\mathbf{C}^\top \mathbf{u}_i - b_{ii} \mathbf{q}_{i-1}\|_2$  by Equation (11.9);
  - 4:      $\mathbf{q}_i \leftarrow \frac{\mathbf{C}^\top \mathbf{u}_i - b_{ii} \mathbf{q}_{i-1}}{b_{i,i+1}}$  by Equation (11.10);
  - 5:      $\mathbf{u}_{i+1} \leftarrow \frac{\mathbf{C} \mathbf{q}_i - b_{i,i+1} \mathbf{u}_i}{b_{i+1,i+1}}$  by Equation (11.12);
  - 6:      $b_{jj} \leftarrow \pm \|\mathbf{C}^\top \mathbf{u}_j - b_{j,j+1} \mathbf{q}_j\|_2$  by Equation (11.11), where  $j = i + 1$ ;
  - 7: **end for**
  - 8: Output  $\mathbf{B}$ ,  $\mathbf{U}$ , and  $\mathbf{V}$ .
- 

The algorithm breaks down if any  $b_{i,i+1}$  or  $b_{jj}$  is equal to zero. However, in the context of solving least squares problems, these cases can be handled with special treatment; see Björck (2004). Another issue arises is that, in floating-point arithmetic, the columns in  $\mathbf{U}$  and  $\mathbf{V}$  can lose orthogonality as the recursion proceeds (similar to the loss of orthogonality seen in the CGS and MGS methods for computing the QR decomposition; see Section 3.5).

**Approximate least squares.** We further explore the approximation of the least squares problem  $\min_{\mathbf{x}} \|\mathbf{Cx} - \mathbf{b}\|_2$ . Denote  $\mathbf{Q}_k = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k]$ ,  $\mathbf{U}_k = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k]$ , and  $\mathbf{U} = [\mathbf{U}_{k+1}, \mathbf{U}_\perp]$ . Additionally, let  $\mathbf{B}_k$  be the upper-left  $k \times (k - 1)$  submatrix of  $\mathbf{B}_2$ . Once again, referring to Equation (11.8), we have:

$$\mathbf{CQ}_k = \mathbf{U}_{k+1} \mathbf{B}_{k+1}.$$

Note that the variable  $\mathbf{x}$  lies in  $\mathbb{R}^{n-1}$ , and the vectors  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$  are mutually orthonormal in  $\mathbb{R}^{n-1}$ . Approximately, we can estimate  $\mathbf{x}$  using a linear combination of the  $k$  vectors, i.e., there exists a vector  $\mathbf{y}$  such that  $\mathbf{x} \approx \mathbf{Q}_k \mathbf{y}$ . Assume we want to find the optimal approximate solution within the subspace spanned by the  $k$  vectors  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$ , i.e., solving the following problem in terms of  $\mathbf{y}$ :

$$\min_{\mathbf{y}} \|\mathbf{CQ}_k \mathbf{y} - \mathbf{b}\|_2, \quad (11.13)$$

where  $\mathbf{y} \in \mathbb{R}^k$  (it can be shown that  $\mathbf{y} = \mathbf{Q}_k^\top \mathbf{x} \in \mathbb{R}^k$ ). Based on the preceding discussion, the optimization problem is equivalent to:

$$\begin{aligned} \min_{\mathbf{y}} \|\mathbf{U}_{k+1} \mathbf{B}_{k+1} \mathbf{y} - \mathbf{b}\|_2 &= \min_{\mathbf{y}} \left\| \mathbf{U}^\top (\mathbf{U}_{k+1} \mathbf{B}_{k+1} \mathbf{y} - \mathbf{b}) \right\|_2 \\ &= \min_{\mathbf{y}} \left\| \begin{bmatrix} \mathbf{B}_{k+1} \mathbf{y} \\ \mathbf{0} \end{bmatrix} - \begin{bmatrix} b_{11} \mathbf{e}_1 \\ \mathbf{0} \end{bmatrix} \right\|_2 = \min_{\mathbf{y}} \|\mathbf{B}_{k+1} \mathbf{y} - b_{11} \mathbf{e}_1\|_2. \end{aligned}$$

Thus, the approximate least squares problem becomes  $\min_{\mathbf{y}} \|\mathbf{B}_{k+1} \mathbf{y} - b_{11} \mathbf{e}_1\|_2$ , where  $b_{11} = \|\mathbf{b}\|_2$ . Due to the bidiagonal structure, the problem can be solved in  $\sim n$  flops (Eldén, 2007).

**Reduced-rank model.** The problem in Equation (11.13) is known as the least squares problem associated with the *reduced-rank model*. Instead of considering the full model  $\min_{\mathbf{y}} \|\mathbf{Cx} - \mathbf{b}\|_2$ , we introduce an approximate orthogonal basis of low dimension in  $\mathbb{R}^{n-1}$

where the solution  $\mathbf{x}$  lies (i.e.,  $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_k\}$ ). This approach helps reduce the ill-conditioning of the original problem and makes the solution less sensitive to perturbations in the data (Eldén, 2007).

### 11.7. Application: PCA via Spectral Decomposition and SVD

An important application of SVD is its use in *principal component analysis (PCA)*. PCA is widely employed to identify patterns in data and to analyze the variance-covariance structure of the data. It serves two primary purposes:

1. *Data reduction.* Reducing the dimensionality of the data by selecting a smaller number of *principal components*.
2. *Interpretation.* Uncovering relationships within the data that were previously unobserved.

Given a data set of  $n$  observations  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ , where each  $\mathbf{x}_i \in \mathbb{R}^p$  for all  $i \in \{1, 2, \dots, n\}$ , the goal is to project the data into a lower-dimensional space of dimension  $m$  ( $m < p$ ). To do this, we first compute the sample mean vector and the 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  $\mathbf{S}$  ensures it is an unbiased and consistent estimator of the covariance matrix (Lu, 2021d). Alternatively, 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 still a consistent estimator of the covariance matrix <sup>3</sup>.

Each data point  $\mathbf{x}_i$  is then projected onto a scalar value using a vector  $\mathbf{u}_1$  (see discussion below), such that the projection is given by  $\mathbf{u}_1^\top \mathbf{x}_i$ . The mean of the projected data is obtained by  $\mathbb{E}[\mathbf{u}_1^\top \mathbf{x}_i] = \mathbf{u}_1^\top \bar{\mathbf{x}}$ , and the variance of the projected data is given by

$$\text{Cov}[\mathbf{u}_1^\top \mathbf{x}_i] = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{u}_1^\top \mathbf{x}_i - \mathbf{u}_1^\top \bar{\mathbf{x}})^2 = \frac{1}{n-1} \sum_{i=1}^n \mathbf{u}_1^\top (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top \mathbf{u}_1 = \mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1.$$

To retain as much information as possible in the projection, we maximize the projected variance  $\mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1$  with respect to  $\mathbf{u}_1$ . To prevent  $\mathbf{u}_1$  from scaling indefinitely, a constraint is imposed:  $\|\mathbf{u}_1\|^2 = \mathbf{u}_1^\top \mathbf{u}_1 = 1$ . Using the method of Lagrange multipliers (see, for example, Bishop (2006); Boyd et al. (2004)), the optimization problem becomes:

$$\mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1 + \lambda_1(1 - \mathbf{u}_1^\top \mathbf{u}_1).$$

Solving this yields the equation:

$$\mathbf{S} \mathbf{u}_1 = \lambda_1 \mathbf{u}_1 \quad \xrightarrow{\text{leads to}} \quad \mathbf{u}_1^\top \mathbf{S} \mathbf{u}_1 = \lambda_1.$$

This shows that  $\mathbf{u}_1$  is an eigenvector of  $\mathbf{S}$  corresponding to the eigenvalue  $\lambda_1$ . The direction of maximum variance,  $\mathbf{u}_1$ , corresponds to the largest eigenvalue of  $\mathbf{S}$ . The eigenvector  $\mathbf{u}_1$  is referred to as the *first principal axis*.

The subsequent principal axes are defined by the remaining eigenvectors of  $\mathbf{S}$ , arranged in descending order of their eigenvalues. By selecting the top  $m$  principal components,

---

3. 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  $n \rightarrow \infty$ .

the dimensionality of the data can be effectively reduced. This process is known as the *maximum-variance formulation* of PCA (Hotelling, 1933; Bishop, 2006; Shlens, 2014). Alternative perspectives on the maximum-variance formulation, such as from data reconstruction, data projection, and autoencoders, are discussed in Lu (2021c). Another approach, known as the *minimum-error formulation* of PCA, is discussed in Pearson (1901); Bishop (2006).

**PCA via the spectral decomposition.** Now, let's assume that the data are already centered, meaning the sample mean vector  $\bar{\mathbf{x}}$  is the zero vector. Alternatively, we can centralize the data by setting  $\mathbf{x}_i := \mathbf{x}_i - \bar{\mathbf{x}}$ , which involves subtracting the mean from each data point. Let the data matrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$  contain the centered data, with each row representing one observation. The covariance matrix is symmetric, and its spectral decomposition is given by

$$\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1} = \mathbf{U} \Lambda \mathbf{U}^\top, \quad (11.14)$$

where  $\mathbf{U}$  is an orthogonal matrix of eigenvectors (the columns of  $\mathbf{U}$  are the eigenvectors of  $\mathbf{S}$ ), and  $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_p)$  is a diagonal matrix containing the corresponding eigenvalues (ordered such that  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p$ ). As discussed above, the eigenvectors are called the *principal axes* of the data, and they *decorrelate* the covariance matrix. Projections of the original data onto the principal axes are called the *principal components*. Specifically, the  $i$ -th principal component is given by the  $i$ -th column of  $\mathbf{X}\mathbf{U}$ . If our objective is to reduce the dimension from  $p$  to  $m$ , we simply select the first  $m$  columns of  $\mathbf{X}\mathbf{U}$ , i.e.,  $\mathbf{X}\mathbf{U}[:, 1:m]$ .

**PCA via SVD.** If the SVD of  $\mathbf{X}$  is given by  $\mathbf{X} = \mathbf{P} \Sigma \mathbf{Q}^\top$ , then the covariance matrix can be expressed as

$$\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1} = \mathbf{Q} \frac{\Sigma^2}{n-1} \mathbf{Q}^\top, \quad (11.15)$$

where  $\mathbf{Q} \in \mathbb{R}^{p \times p}$  is an orthogonal matrix containing the right singular vectors of  $\mathbf{X}$ , and the upper-left part of  $\Sigma$  is a diagonal matrix containing the singular values  $\text{diag}(\sigma_1, \sigma_2, \dots)$ , ordered such that  $\sigma_1 \geq \sigma_2 \geq \dots$ . The number of singular values is equal to  $\min\{n, p\}$ , which will not be larger than  $p$ , and some of these values may be zero.

By comparing Equation (11.15) with Equation (11.14), we can see that Equation (11.15) also represents a spectral decomposition of  $\mathbf{S}$ . This is because both the eigenvalues in  $\Lambda$  and the singular values in  $\Sigma$  are ordered in descending order, and the spectral decomposition in terms of the eigenspaces is unique (as discussed in Section 10.3).

In other words, the right singular vectors  $\mathbf{Q}$  can also serve as the principal axes, which decorrelate the covariance matrix. The singular values are related to the eigenvalues of the covariance matrix through the relationship:  $\lambda_i = \frac{\sigma_i^2}{n-1}$  for each  $i$ . To reduce the dimensionality of the data from  $p$  to  $m$ , we select the largest  $m$  singular values and their corresponding right singular vectors. This process is related to the truncated SVD (TSVD), where:  $\mathbf{X}_m = \sum_{i=1}^m \sigma_i \mathbf{p}_i \mathbf{q}_i^\top$ , where  $\mathbf{p}_i$ 's and  $\mathbf{q}_i$ 's are the columns of  $\mathbf{P}$  and  $\mathbf{Q}$ , respectively.

**A byproduct of PCA via SVD for high-dimensional data.** For a principal axis  $\mathbf{u}_i$  of  $\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1}$ , we have  $\frac{\mathbf{X}^\top \mathbf{X}}{n-1} \mathbf{u}_i = \lambda_i \mathbf{u}_i$ . Multiplying both sides by  $\mathbf{X}$  on the left, we obtain:

$$\frac{\mathbf{X} \mathbf{X}^\top}{n-1} (\mathbf{X} \mathbf{u}_i) = \lambda_i (\mathbf{X} \mathbf{u}_i),$$

which implies that  $\lambda_i$  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 relationship is also discussed in the proof of Theorem 11.1, which

establishes the existence of the SVD. When the number of features  $p$  is much larger than the number of samples  $n$  (i.e.,  $p \gg n$ ), instead of finding the eigenvectors of  $\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1}$ , i.e., the principal axes of  $\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1}$ , we can find the eigenvectors of  $\frac{\mathbf{X}\mathbf{X}^\top}{n-1}$ . This reduces the computational complexity from  $\mathcal{O}(p^3)$  to  $\mathcal{O}(n^3)$ , which is more efficient when  $p \gg n$ .

Now, returning to the principal axes of  $\mathbf{S} = \frac{\mathbf{X}^\top \mathbf{X}}{n-1}$ , suppose the eigenvector of  $\frac{\mathbf{X}\mathbf{X}^\top}{n-1}$  is  $\mathbf{v}_i$ , corresponding to a nonzero eigenvalue  $\lambda_i$ :  $\frac{\mathbf{X}\mathbf{X}^\top}{n-1}\mathbf{v}_i = \lambda_i\mathbf{v}_i$ . Multiplying both sides by  $\mathbf{X}^\top$ , we obtain

$$\frac{\mathbf{X}^\top \mathbf{X}}{n-1}(\mathbf{X}^\top \mathbf{v}_i) = \mathbf{S}(\mathbf{X}^\top \mathbf{v}_i) = \lambda_i(\mathbf{X}^\top \mathbf{v}_i),$$

which shows that 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  $\lambda_i$ . Note that a further normalization step is required to ensure that  $\|\mathbf{u}_i\| = 1$ . Thus, when  $p \gg n$ , we can efficiently compute the principal axes using the spectral decomposition of  $\frac{\mathbf{X}\mathbf{X}^\top}{n-1}$ , instead of directly computing the eigenvectors of  $\mathbf{S}$ .

**Data whitening.** PCA is commonly used for feature preprocessing in machine learning. It first reduces the dimensionality of the data and then normalizes the newly transformed features so that the variance along each direction in the transformed space is equal. Let  $\mathbf{U}_m$  be the  $p \times m$  matrix containing the top- $m$  eigenvectors obtained from PCA. The first step is to transform the mean-centered data matrix  $\mathbf{X}$  into an  $m$ -dimensional representation using  $\mathbf{U}_m$ , as follows:

$$\widetilde{\mathbf{X}} = \mathbf{X}\mathbf{U}_m.$$

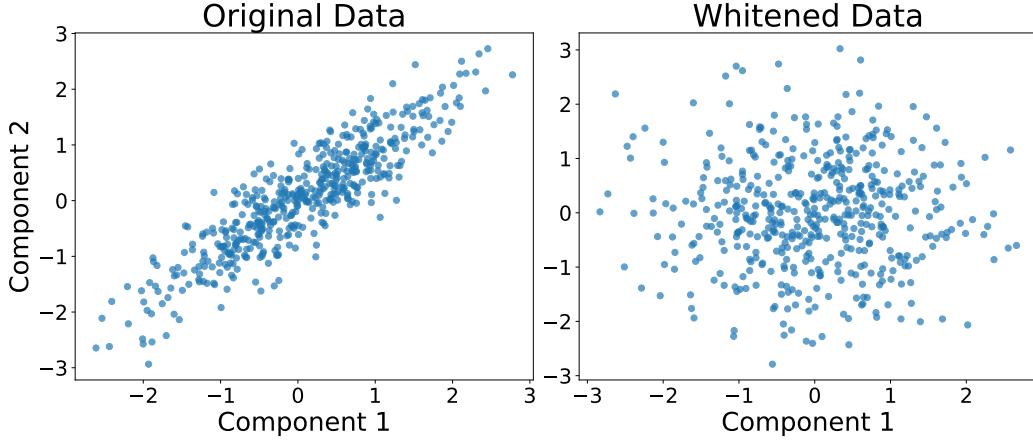
The next step involves scaling each column of  $\widetilde{\mathbf{X}}$  by its standard deviation. This process transforms the original data distribution into one that is approximately spherical in shape. This technique is known as *whitening*.

Whitened data often leads to better performance in gradient-based optimization algorithms (Lu, 2025). This is because large differences in variance across features can cause the loss function to have varying curvature in different directions, which slows down convergence. By normalizing the variance, whitening reduces ill-conditioning of the loss function, allowing gradient descent to converge faster. Additionally, it prevents certain features from dominating the learning process due to their scale.

Whitening is especially valuable in unsupervised learning tasks such as outlier detection, where no labels are available to guide the relative importance of different directions in the data. In such cases, ensuring that all directions are treated equally becomes even more critical. An illustration of how PCA whitens an ellipsoidal data distribution is shown in Figure 11.8, where the resulting distribution becomes approximately spherical.

## 11.8. Application: Low-Rank Approximation

In the context of low-rank approximation, two types of problems arise due to the interplay between rank and approximation error: *fixed-precision approximation problem* and *fixed-rank approximation problem*. In the fixed-precision approximation problem, given a matrix  $\mathbf{A}$  and a tolerance  $\epsilon$ , the goal is 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 seeks a matrix  $\mathbf{B}$  with a fixed rank  $k$  that minimizes the error  $\|\mathbf{A} - \mathbf{B}\|$ . This section focuses on the latter.



**Figure 11.8:** An example of whitening an ellipsoidal data distribution using principal component analysis.

To approximate a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  of rank  $r$  with a rank- $k$  matrix  $\mathbf{B}$  ( $k < r$ ), the approximation can be evaluated using the spectral norm (Definition 0.28):

$$\mathbf{B} = \underset{\text{rank}(\mathbf{B})=k}{\arg \min} \|\mathbf{A} - \mathbf{B}\|_2. \quad (11.16)$$

Then, we can recover the optimal rank- $k$  approximation by the following theorem.

**Theorem 11.19: (Eckart–Young–Mirsky theorem w.r.t. spectral norm)** Given a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $1 \leq k \leq \text{rank}(\mathbf{A}) = r$ , and let  $\mathbf{A}_k$  be the *truncated SVD (TSVD)* of  $\mathbf{A}$  with the largest  $k$  singular terms, i.e.,  $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  from the 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 optimal rank- $k$  approximation to  $\mathbf{A}$  in terms of the spectral norm. <sup>a</sup>

<sup>a</sup>. Note that  $\mathbf{A}_k$  can be stored using  $(m + n)k + k$  entries, as opposed to  $mn$  entries.

**Proof** [of Theorem 11.19] For any matrix  $\mathbf{B}$  with  $\text{rank}(\mathbf{B}) = k$ , we aim to show that  $\|\mathbf{A} - \mathbf{B}\|_2 \geq \|\mathbf{A} - \mathbf{A}_k\|_2$ .

Since  $\text{rank}(\mathbf{B}) = k$ ,  $\dim(\mathcal{N}(\mathbf{B})) = n - k$ . Thus, any set of  $k + 1$  basis vectors in  $\mathbb{R}^n$  intersects  $\mathcal{N}(\mathbf{B})$ . From Proposition 11.8, the singular vectors  $\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r\}$  form an orthonormal basis for  $\mathcal{C}(\mathbf{A}^\top) \subset \mathbb{R}^n$ ; so we can choose the first  $k + 1$  singular vectors  $\mathbf{v}_i$  as a basis for a  $(k + 1)$ -dimensional subspace of  $\mathbb{R}^n$ . Let  $\mathbf{V}_{k+1} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k+1}]$ . Then there exists a unit vector  $\mathbf{x}$  such that

$$\mathbf{x} \in \mathcal{N}(\mathbf{B}) \cap \mathcal{C}(\mathbf{V}_{k+1}), \quad \text{s.t. } \|\mathbf{x}\|_2 = 1.$$

That is, the vector  $\mathbf{x}$  can be expressed as  $\mathbf{x} = \sum_{i=1}^{k+1} a_i \mathbf{v}_i$  with  $\left\| \sum_{i=1}^{k+1} a_i \mathbf{v}_i \right\|_2 = \sum_{i=1}^{k+1} a_i^2 = 1$ , and we also have  $\mathbf{Bx} = \mathbf{0}$ . Using these properties, the following chain of inequalities holds:

$$\begin{aligned} \|\mathbf{A} - \mathbf{B}\|_2^2 &\geq \|(\mathbf{A} - \mathbf{B})\mathbf{x}\|_2^2 / \|\mathbf{x}\|_2^2 \stackrel{\dagger}{=} \|\mathbf{Ax}\|_2^2 \stackrel{\ddagger}{=} \sum_{i=1}^{k+1} \sigma_i^2 (\mathbf{v}_i^\top \mathbf{x})^2 \\ &\stackrel{*}{\geq} \sigma_{k+1}^2 \sum_{i=1}^{k+1} (\mathbf{v}_i^\top \mathbf{x})^2 \stackrel{\ddagger}{\geq} \sigma_{k+1}^2 \sum_{i=1}^{k+1} a_i^2 = \sigma_{k+1}^2, \end{aligned}$$

where the first inequality follows from the definition of the spectral norm, the equality ( $\dagger$ ) follows from the fact that  $\mathbf{x}$  lies in null space of  $\mathbf{B}$ , the equality (+) follows from the fact that  $\mathbf{x}$  is orthogonal to  $\mathbf{v}_{k+2}, \dots, \mathbf{v}_r$ , the inequality (\*) follows from  $\sigma_{k+1} \leq \sigma_k \leq \dots \leq \sigma_1$ , and the inequality ( $\ddagger$ ) follows from  $\mathbf{v}_i^\top \mathbf{x} = a_i$ . On the other hand, it is evident that  $\|\mathbf{A} - \mathbf{A}_k\|_2^2 = \left\| \sum_{i=k+1}^r \sigma_i \mathbf{u}_i \mathbf{v}_i^\top \right\|_2^2 = \sigma_{k+1}^2$ . Thus,  $\|\mathbf{A} - \mathbf{A}_k\|_2 \leq \|\mathbf{A} - \mathbf{B}\|_2$ , which completes the proof.  $\blacksquare$

Moreover, it can also be shown that  $\mathbf{A}_k$  is the optimal rank- $k$  approximation to  $\mathbf{A}$  in terms of the Frobenius norm (Definition 0.27). The minimal error is given by the Euclidean norm of the singular values that have been zeroed out in the process:  $\|\mathbf{A} - \mathbf{A}_k\|_F = \sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \dots + \sigma_r^2}$ .

**Exercise 11.20.** Given the definition of the Frobenius norm, show that the truncated SVD  $\mathbf{A}_k = \sum_{i=1}^k \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$  is also the optimal rank- $k$  approximation to  $\mathbf{A}$  in terms of the Frobenius norm.

**Application in the text domain.** Low-rank approximation is an important technique with significant applications in text analysis. In this context, each document is represented as a row in a matrix, where the number of columns corresponds to the number of unique words (features). The value of each entry in the matrix represents the frequency of a specific word in the corresponding document. It is worth noting that such matrices are typically very sparse, making them a standard use case for SVD. For example, a word-frequency matrix  $\mathbf{A}$  might have dimensions  $m = 10^6$  documents by  $n = 10^5$  words. In such cases, truncated SVD often provides excellent approximations of the original matrix using a rank as small as  $k \approx 400$  (Aggarwal, 2020), which results in a substantial reduction in dimensionality. This application of SVD in text analysis is commonly referred to as *latent semantic analysis*, due to its ability to uncover hidden (latent) topics represented by the rank-1 components of the SVD.

## ~~~~ Chapter 11 Problems ~~~~

1. Show that  $(\mathbf{A}\mathbf{A}^\top)^{1/2}\mathbf{A} = \mathbf{A}(\mathbf{A}^\top\mathbf{A})^{1/2}$ .
2. Given a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , show that the trace of  $\mathbf{A}^\top\mathbf{A}$  is equal to the sum of the squares of all its entries, i.e.,  $\text{tr}(\mathbf{A}^\top\mathbf{A}) = \sum_{i,j=1}^{m,n} a_{ij}^2$ .
3. **Matrix factorization via spanning subspaces.** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be a matrix of rank  $r$ . Suppose the columns of  $\mathbf{B} \in \mathbb{R}^{m \times r}$  span the column space of  $\mathbf{A}$ , and the columns of  $\mathbf{C} \in \mathbb{R}^{n \times r}$  span the row space of  $\mathbf{A}$ . Show that the matrix  $\mathbf{A}$  can be factored as  $\mathbf{A} = \mathbf{B}\mathbf{E}\mathbf{C}^\top$ , where  $\mathbf{E}$  is an  $r$  by  $r$  nonsingular matrix.
4. Consider a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  of rank  $r$ , and the  $(2n) \times (2n)$  symmetric matrix  $\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{0} \end{bmatrix}$ . If  $\mathbf{A}$  admits the full SVD  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ :
  - Show that  $\sigma_k$  is an eigenvalue of  $\mathbf{B}$  corresponding to the eigenvector  $\begin{bmatrix} \mathbf{v}_k \\ \mathbf{u}_k \end{bmatrix}$  for any  $k \in \{1, 2, \dots, n\}$ , and that  $-\sigma_k$  is an eigenvalue of  $\mathbf{B}$  corresponding to the eigenvector  $\begin{bmatrix} \mathbf{v}_k \\ -\mathbf{u}_k \end{bmatrix}$  for any  $k \in \{1, 2, \dots, n\}$ .
  - Show that the  $2n$  eigenvectors are pairwise orthogonal.

5. Consider a rectangular matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  of rank  $r$ , and the  $(m+n) \times (m+n)$  symmetric matrix  $\mathbf{B} = \begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{A}^\top & \mathbf{0} \end{bmatrix}$ . If  $\mathbf{A}$  admits the full SVD  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ :
- Show that  $\sigma_k$  is an eigenvalue of  $\mathbf{B}$  corresponding to the eigenvector  $\begin{bmatrix} \mathbf{v}_k \\ \mathbf{u}_k \end{bmatrix}$  for any  $k \in \{1, 2, \dots, r\}$ , and that  $-\sigma_k$  is an eigenvalue of  $\mathbf{B}$  corresponding to the eigenvector  $\begin{bmatrix} \mathbf{v}_k \\ -\mathbf{u}_k \end{bmatrix}$  for any  $k \in \{1, 2, \dots, r\}$ .
  - Show that the remaining  $m+n-2r$  eigenvectors of  $\mathbf{B}$  are corresponding to the eigenvalue 0.
  - Show that the  $m+n$  eigenvectors are pairwise orthogonal.
6. Given two nonzero vectors  $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ , and let  $\mathbf{A} = \mathbf{u}\mathbf{v}^\top$ . Show that the nonzero singular value of  $\mathbf{A}$  is  $\|\mathbf{u}\| \cdot \|\mathbf{v}\|$ .
7. For a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  with singular values  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n$ , show that  $\sigma_1^3, \sigma_2^3, \dots, \sigma_n^3$  are the singular values of  $\mathbf{A}\mathbf{A}^\top$ .
8. Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be a rectangular matrix, and let  $\mathbf{B} \in \mathbb{R}^{\hat{m} \times \hat{n}}$  be a submatrix of  $\mathbf{A}$ , where  $\hat{m} \leq m$  and  $\hat{n} \leq n$ . Show that the largest singular value of  $\mathbf{B}$  is less than or equal to the largest singular value of  $\mathbf{A}$ .
9. For a positive definite matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , show that the singular values and the eigenvalues of  $\mathbf{A}$  are the same.
10. Given a matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  and a positive definite matrix  $\mathbf{B} \in \mathbb{R}^{n \times n}$ , prove that the singular values of  $\mathbf{B}\mathbf{A}$  are the same as those of  $\mathbf{A}$ . Discuss the relationship between the left and right singular vectors of  $\mathbf{B}\mathbf{A}$  and  $\mathbf{A}$ .
11. We have shown in Lemma 11.10 that orthogonally equivalent matrices share the same singular values. Prove the reverse implication: if two matrices have the same singular values, then they are orthogonally equivalent.
12. In this chapter, we focus on the SVD of real matrices, expressed as  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ . Show that if  $\mathbf{A}$  is real, then the matrices  $\mathbf{U}$  and  $\mathbf{V}$  are also real.
13. Given a Householder transformation matrix  $\mathbf{H} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^\top \in \mathbb{R}^{n \times n}$ , where  $\|\mathbf{u}\| = 1$ , determine the eigenvalues, determinant, and singular values of  $\mathbf{H}$ .
14. Given the nonzero singular values  $\sigma_1, \sigma_2, \dots, \sigma_r$  of  $\mathbf{A}$ , discuss the singular values of  $\mathbf{A}^\top$ ,  $\gamma\mathbf{A}$  with  $\gamma > 0$ , and  $\mathbf{A}^{-1}$  (if  $\mathbf{A}$  is nonsingular).
15. Given a square and real matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , show that  $\mathbf{A} = \mathbf{0}$  if and only if  $\mathbf{A}$  has only zero eigenvalues.
16. Given a square matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$ , show that  $\mathbf{A}^\top\mathbf{A}$  and  $\mathbf{A}\mathbf{A}^\top$  are similar (Definition 7.3). *Hint: Proceeding with the SVD of  $\mathbf{A}$ .*
17. Show that all eigenvalues of a square matrix are less than or equal to its largest singular value  $\sigma_1$ .
18. Suppose  $\mathbf{x}$  is an eigenvector of  $\mathbf{A}^\top\mathbf{A}$  corresponding to a nonzero eigenvalue. Discuss the corresponding eigenvector of  $\mathbf{A}\mathbf{A}^\top$ . *Hint: Premultiply by  $\mathbf{A}$ .*
19. Given the SVD of a nonsingular square matrix  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \in \mathbb{R}^{n \times n}$ , determine the singular values of  $\mathbf{A}^\top\mathbf{A}$ .
20. Find the optimal rank-one approximation (in terms of the spectral norm) for the matrix:  $\mathbf{A} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$ .
21. **Skew-symmetric.** Given a skew-symmetric and tridiagonal matrix  $\mathbf{S} \in \mathbb{R}^{n \times n}$ , show that it can be decomposed as:  $\mathbf{P}^\top \mathbf{S} \mathbf{P} = \begin{bmatrix} \mathbf{0} & \mathbf{B}^\top \\ \mathbf{B} & \mathbf{0} \end{bmatrix}$ , where  $\mathbf{B} \in \mathbb{R}^{m \times m}$ ,  $n = 2m$ , and  $\mathbf{P}$

is a permutation matrix. Given further the SVD of  $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^\top$ , find the eigenvalues and eigenvectors of  $\mathbf{S}$ .

22. Discuss the uniqueness of the polar decomposition for the matrix:  $\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ .
23. Let  $\mathbf{A}$  be a negative semidefinite matrix. Show that the singular value decomposition of  $\mathbf{A}$  is of the form  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$ , where  $\mathbf{U} = -\mathbf{V}$ .
24. **Block diagonal structure of padded SVD.** Let  $\mathbf{B}$  be a  $p \times p$  matrix obtained by padding the  $m \times n$  matrix  $\mathbf{A}$  with either zero rows or zero columns, where  $p = \max\{m, n\}$ . Depending on whether  $m$  is greater than  $n$  or vice versa, show that the SVD  $\mathbf{B} = \mathbf{U}\Sigma\mathbf{V}^\top$  takes one of the following forms:

$$\text{When } n < m : \quad \mathbf{B} = [\mathbf{A}, \mathbf{0}] = \mathbf{U} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{V}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{V}_2 \end{bmatrix}^\top;$$

$$\text{When } m < n : \quad \mathbf{B} = \begin{bmatrix} \mathbf{A} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{U}_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \mathbf{V}^\top.$$

Here, the matrices  $\mathbf{U}$ ,  $\mathbf{V}$ , and  $\Sigma$  are all square matrices of size  $p \times p$ . The matrix  $\mathbf{V}_1$  is of size  $n \times n$ , and  $\mathbf{U}_1$  is of size  $m \times m$ . The matrices  $\mathbf{V}_2$  and  $\mathbf{U}_2$  are of sizes  $(p-n) \times (p-n)$  and  $(p-m) \times (p-m)$ , respectively. The matrix  $\Sigma_1$  is of size  $\min\{m, n\} \times \min\{m, n\}$ .

25. **Two-way to three-way SVD.** Let  $\mathbf{A} = \mathbf{P}\mathbf{Q}^\top \in \mathbb{R}^{m \times n}$  be a decomposition of matrix  $\mathbf{A}$ , where the columns of  $\mathbf{P} \in \mathbb{R}^{m \times k}$  and  $\mathbf{Q} \in \mathbb{R}^{n \times k}$  are orthogonal (not necessarily orthonormal, i.e., having unit norms), and  $k \leq \min\{m, n\}$ . Provide a way to obtain the SVD of  $\mathbf{A}$ .
26. **Push-through identity.** Use SVD to prove the push-through identity:

$$\mathbf{C}^\top (\lambda \mathbf{I}_m + \mathbf{C}\mathbf{C}^\top)^{-1} = (\lambda \mathbf{I}_n + \mathbf{C}^\top \mathbf{C})^{-1} \mathbf{C}^\top, \quad (11.17)$$

where  $\lambda > 0$ , and  $\mathbf{C} \in \mathbb{R}^{m \times n}$ .

27. **Shared SVD from identical scatter matrices.** Consider two data matrices  $\mathbf{A}_1$  and  $\mathbf{A}_2$  that have identical scatter matrices  $\mathbf{A}_1^\top \mathbf{A}_1 = \mathbf{A}_2^\top \mathbf{A}_2$ , but are otherwise distinct. Show that both  $\mathbf{A}_1$  and  $\mathbf{A}_2$  can be decomposed using a partially shared singular value decomposition, such that  $\mathbf{A}_1 = \mathbf{U}_1 \Sigma \mathbf{V}^\top$  and  $\mathbf{A}_2 = \mathbf{U}_2 \Sigma \mathbf{V}^\top$ . Use this fact to show that  $\mathbf{A}_2 = \mathbf{Q}_{12} \mathbf{A}_1$ , where  $\mathbf{Q}_{12}$  is an orthogonal matrix.
28. **Frobenius norm.** Let  $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ . Show that the squared Frobenius norm of  $\mathbf{A} - \mathbf{B}$  is

$$\|\mathbf{A} - \mathbf{B}\|_F^2 = \|\mathbf{A}\|_F + \|\mathbf{B}\|_F - 2\text{tr}(\mathbf{A}^\top \mathbf{B}).$$

29. Is there any coordinate transformation involved in the QR or LQ decomposition?
30. Given the SVD of the matrix

$$\mathbf{A} = \begin{bmatrix} 2 & 2 \\ -1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2\sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ -1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} = \mathbf{U}\Sigma\mathbf{V}^\top,$$

illustrate the coordinate transformation of this decomposition in a two-dimensional figure.

31. (Horn and Johnson, 2012) Let  $\mathbf{Q} \in \mathbb{R}^{n \times n}$  be an orthogonal matrix. Show that  $\mathbf{Q}$  can be decomposed as  $\mathbf{Q} = \mathbf{U}_1 \mathbf{U}_2 \dots \mathbf{U}_N \mathbf{D}$ , where  $\mathbf{D} = \text{diag}(1, 1, \dots, 1, \det(\mathbf{Q}))$ , each  $\mathbf{U}_i$  represents a plane rotation (Definition 2.15), and  $N = n(n-1)/2$ . Hint: Use the result in Problem 3.12.

# **Part VI**

## **Special Topics**



# 12

## Alternating Least Squares (ALS)

It is evident that any given matrix can be factorized in infinitely many ways. However, certain types of factorizations are particularly valuable because of the specific properties they offer. Two main types of such properties are commonly sought in matrix decompositions:

- *Linear algebra properties with exact decomposition.* In this type of decomposition, the goal is to break down a matrix into components that possess particular linear algebraic or geometric characteristics, such as orthogonality or triangular form. These properties make the decomposition useful for various tasks in linear algebra, including the construction of (orthogonal) bases, as discussed in previous chapters. So far, we have studied several decompositions that fall into this category, including LU decomposition, CR decomposition, QR decomposition, and singular value decomposition (SVD).
- *Optimization and compression properties with approximate decomposition.* This type focuses on approximating a large matrix by factoring it into smaller matrices. A well-known example is truncated SVD. Consider a matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  that is approximated by a rank- $k$  matrix using the following factorization:

$$\mathbf{A} \approx \mathbf{U}_k \boldsymbol{\Sigma}_k \mathbf{V}_k^\top, \quad (12.1)$$

where  $\mathbf{U}_k$  is an  $m \times k$  semi-orthogonal matrix,  $\boldsymbol{\Sigma}_k$  is a  $k \times k$  diagonal matrix with nonnegative entries, and  $\mathbf{V}_k$  is an  $n \times k$  semi-orthogonal matrix. The total number of entries across all three matrices is  $(m + n + k)k$  or  $(m + n + 1)k$  if counting only nonzero entries, which is often significantly smaller than the  $mn$  entries in the original matrix when  $m$  and  $n$  are both large.

As discussed previously, singular value decomposition is unique in that it offers advantages from both perspectives: it provides strong linear algebra properties when used exactly and useful compression properties when truncated; see Theorem 11.19. The value  $k$  is referred to as the rank of the approximation. The optimization-based view of matrix factorization, where we approximate  $\mathbf{A} \approx \mathbf{W}\mathbf{Z}$ , is especially valuable in machine learning. This approach involves defining  $\mathbf{A}$ ,  $\mathbf{W}$ , and  $\mathbf{Z}$  in different ways depending on the application. Below are two important examples:

1. A *rating* is a numerical score that a user assigns to an item, such as a movie <sup>1</sup>. *Recommender systems* collect these ratings to predict how users might rate items

---

<sup>1</sup>. For example, see the top 250 movies rated by different websites: <https://www.imdb.com/list/ls027618268/>.

they haven't yet rated. When  $\mathbf{A}$  represents a *user-item rating matrix*—where rows correspond to items, columns to users, and entries contain the observed ratings—the matrix factorization  $\mathbf{A} \approx \mathbf{W}\mathbf{Z}$  is performed using only the known ratings. In this case, the rows of  $\mathbf{W}$  represent latent features of items, and the columns of  $\mathbf{Z}$  represent latent features of users. The product  $\mathbf{W}\mathbf{Z}$  reconstructs the full rating matrix, including predictions for missing entries.

2. When  $\mathbf{A}$  is a *term-document matrix*, representing the frequency of words (rows of  $\mathbf{A}$ ) in documents (columns of  $\mathbf{A}$ ), the rows of  $\mathbf{W}$  provide latent representations of words, and the columns of  $\mathbf{Z}$  provide latent representations of documents (see Chapter 13).

From an optimization perspective, additional constraints can be imposed on the matrices involved in the factorization—such as requiring their entries to be nonnegative (as discussed in Chapter 13). These constraints often enhance the usefulness of the decomposition in practical applications.

In this chapter, we will focus on the first application mentioned above (recommender systems). In the next chapter, we will discuss the second (topic modeling via term-document matrices using *nonnegative matrix factorization (NMF)*).

### 12.1. Preliminary: Least Squares Approximations

The linear model is a fundamental technique in regression analysis, relying on the least squares approximation, which aims to minimize the sum of squared errors (refer to Section 11.6). This method naturally emerges when trying to identify the regression function that minimizes the corresponding expected squared error. Over the past several decades, linear models have found extensive applications across diverse domains, including decision-making (Dawes and Corrigan, 1974), time series analysis (Christensen, 1991; Lu, 2017), quantitative finance (Menchero et al., 2011), and various other fields such as production science, social science, and soil science (Fox, 1997; Lane, 2002; Schaeffer, 2004; Mrode, 2014).

To be more concrete, consider an overdetermined system represented by  $\mathbf{b} = \mathbf{Ax}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$  represents the *input data matrix* (also known as the *predictor variables*),  $\mathbf{b} \in \mathbb{R}^m$  is the *observation vector* (or *target/response vector*), and the number of samples  $m$  exceeds the number of predictors  $n$ . The vector  $\mathbf{x}$  represents the *weights* (or *coefficients*) of the linear model. Typically, it is assumed that  $\mathbf{A}$  has full column rank, as real-world data is often uncorrelated or can be preprocessed to meet this condition. In practical scenarios, a *bias term* (a.k.a., an *intercept*) is added to the first column of  $\mathbf{A}$ . This adjustment enables the least squares method to solve equations of the form:

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}} = [\mathbf{1}, \mathbf{A}] \begin{bmatrix} x_0 \\ \mathbf{x} \end{bmatrix} = \mathbf{b}. \quad (12.2)$$

However, it is common for the equation  $\mathbf{b} = \mathbf{Ax}$  to have no exact solution (the system is *inconsistent*) because it is overdetermined—that is, there are more equations than unknowns. Define the column space of  $\mathbf{A}$  as  $\{\mathbf{A}\gamma : \forall \gamma \in \mathbb{R}^n\}$ , denoted by  $\mathcal{C}(\mathbf{A})$ . In essence, when we say  $\mathbf{b} = \mathbf{Ax}$  has no solution, it implies that  $\mathbf{b}$  lies outside the column space of  $\mathbf{A}$ . In other words, the error  $\mathbf{e} = \mathbf{b} - \mathbf{Ax}$  cannot be reduced to zero. The objective then becomes minimizing the error, which is typically measured using the mean squared error. The resulting solution  $\mathbf{x}_{LS}$ , which minimizes  $\|\mathbf{b} - \mathbf{Ax}_{LS}\|_2^2$ , is referred to as the *least squares solution*. The least squares method is a cornerstone of mathematical sciences, and a wealth of resources are dedicated to its study and application, including works by Trefethen and Bau III (1997); Strang (2019, 2021); Lu (2021d).

**Least squares by calculus.** When  $\|\mathbf{b} - \mathbf{Ax}\|_2^2$  is differentiable and the parameter space of  $\mathbf{x}$  spans the entire space  $\mathbb{R}^n$  (i.e., an unconstrained optimization problem)<sup>2</sup>, the least squares estimate corresponds to the root of the gradient of  $\|\mathbf{b} - \mathbf{Ax}\|_2^2$ . This leads us to the following lemma.<sup>3</sup>

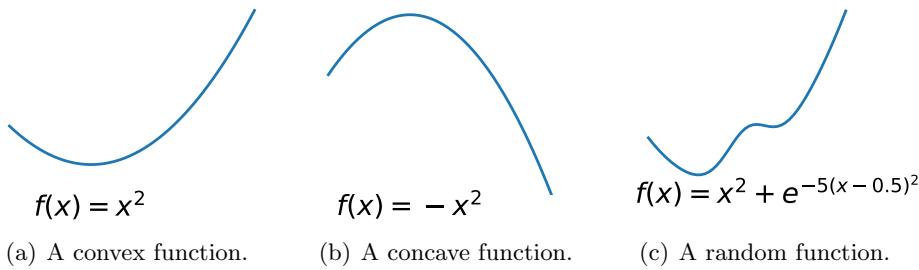
**Lemma 12.1: (Least squares by calculus)** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be a fixed data matrix with full rank and  $m \geq n$  (i.e., its columns are linearly independent)<sup>a</sup>. For the overdetermined system  $\mathbf{b} = \mathbf{Ax}$ , the least squares solution, obtained by setting the partial derivatives in every direction of  $\|\mathbf{b} - \mathbf{Ax}\|_2^2$  to zero (i.e., the gradient vanishes), is given by  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ <sup>b</sup>. The value,  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ , is commonly referred to as the *ordinary least squares (OLS)* estimate or simply the *least squares (LS)* estimate of  $\mathbf{x}$ .

- a. Relaxations of this condition using the pseudo-inverse are discussed in Problems 12.1~12.3.
- b. This is known as the *first-order optimality condition* for local optima points. Note that the proof of the first-order optimality condition for multivariate functions strongly relies on the first-order optimality conditions for univariate functions, which is also known as *Fermat's theorem*. See Problem 12.18.

To prove the lemma above, we must show that  $\mathbf{A}^\top \mathbf{A}$  is invertible. Given that  $\mathbf{A}$  has full rank and  $m \geq n$ , the matrix  $\mathbf{A}^\top \mathbf{A} \in \mathbb{R}^{n \times n}$  is invertible if it has rank  $n$ , which matches the rank of  $\mathbf{A}$ . This claim is verified in Lemma 11.3.

**Proof** [of Lemma 12.1] Using calculus, a function  $f(\mathbf{x})$  attains a minimum at  $\mathbf{x}_{LS}$  when its gradient  $\nabla f(\mathbf{x}) = \mathbf{0}$ . The gradient of  $\|\mathbf{b} - \mathbf{Ax}\|_2^2$  is given by  $2\mathbf{A}^\top \mathbf{Ax} - 2\mathbf{A}^\top \mathbf{b}$ .  $\mathbf{A}^\top \mathbf{A}$  is invertible since we assume  $\mathbf{A}$  is fixed and has full rank with  $m \geq n$  (Lemma 11.3). Consequently, the OLS solution for  $\mathbf{x}$  is  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ , which completes the proof. ■

**Definition 12.2 (Normal equation).** The condition for the gradient of  $\|\mathbf{b} - \mathbf{Ax}\|_2^2$  to be zero can be expressed as  $\mathbf{A}^\top \mathbf{Ax} = \mathbf{A}^\top \mathbf{b}$ . This is called the *normal equation*. Under the assumption that  $\mathbf{A}$  has full rank with  $m \geq n$ , the matrix  $\mathbf{A}^\top \mathbf{A}$  is invertible, leading to the solution  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$ .



**Figure 12.1:** Three types of functions.

However, it is not immediately evident whether the least squares estimate derived in Lemma 12.1 corresponds to a minimum, maximum, or neither. An example illustrating this ambiguity is shown in Figure 12.1. What we can assert with confidence is the existence of at least one root for the gradient of the function  $f(\mathbf{x}) = \|\mathbf{b} - \mathbf{Ax}\|_2^2$ . This root represents a necessary condition for a minimum point, but not necessarily a sufficient one. The following remark provides further clarification on this matter.

2. In this context, the *domain* of the optimization problem  $\min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_2^2$  is the entire space  $\mathbb{R}^n$ .

3. Variants of the least squares problem are explored in Problems 12.8~12.14.

**Remark 12.3 (Verification of least squares solution).** Why does a zero gradient imply the least mean squared error? The usual explanation relies on convex analysis, which we will explore shortly. However, here we directly confirm that the OLS solution minimizes the mean squared error. For any  $\mathbf{x} \neq \mathbf{x}_{LS}$ , we have

$$\begin{aligned}\|\mathbf{b} - \mathbf{Ax}\|_2^2 &= \|\mathbf{b} - \mathbf{Ax}_{LS} + \mathbf{Ax}_{LS} - \mathbf{Ax}\|_2^2 = \|\mathbf{b} - \mathbf{Ax}_{LS} + \mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|_2^2 \\ &= \|\mathbf{b} - \mathbf{Ax}_{LS}\|_2^2 + \|\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|_2^2 + 2(\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x}))^\top (\mathbf{b} - \mathbf{Ax}_{LS}) \\ &= \|\mathbf{b} - \mathbf{Ax}_{LS}\|_2^2 + \|\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|_2^2 + 2(\mathbf{x}_{LS} - \mathbf{x})^\top (\mathbf{A}^\top \mathbf{b} - \mathbf{A}^\top \mathbf{Ax}_{LS}),\end{aligned}$$

where the third term is zero because of the normal equation, and  $\|\mathbf{A}(\mathbf{x}_{LS} - \mathbf{x})\|_2^2 \geq 0$ . Therefore,  $\|\mathbf{b} - \mathbf{Ax}\|_2^2 \geq \|\mathbf{b} - \mathbf{Ax}_{LS}\|_2^2$ . Thus, we have demonstrated that the OLS estimate corresponds to a minimum, not a maximum or a saddle point <sup>a</sup>. As a matter of fact, this condition from the least squares estimate is also known as the *sufficiency of stationarity under convexity*. When  $\mathbf{x}$  is defined over the entire space  $\mathbb{R}^n$ , this condition is also known as the *necessity of stationarity under convexity*.

---

<sup>a</sup> A *saddle point* is a point at which the gradient vanishes (a *stationary point*), but the objective function increases in some directions and decreases in others.

Another natural question is: Why does the normal equation appear to “magically” provide solutions for  $\mathbf{x}$ ? A simple analogy can help illustrate this concept. The equation  $x^2 = -1$  has no real solution. However,  $x \cdot x^2 = x \cdot (-1)$  does have a real solution  $\hat{x} = 0$ , in which case,  $\hat{x}$  minimizes the difference between  $x^2$  and  $-1$ , making them as close as possible.

**Example 12.4 (Altering the solution set by left multiplication).** Consider the data matrix and target vector:  $\mathbf{A} = \begin{bmatrix} -3 & -4 \\ 4 & 6 \\ 1 & 1 \end{bmatrix}$  and  $\mathbf{b} = \begin{bmatrix} 1 \\ -1 \\ 0 \end{bmatrix}$ . It can be easily verified that the system  $\mathbf{Ax} = \mathbf{b}$  has no solution for  $\mathbf{x}$ . However, if we multiply both sides on the left by  $\mathbf{B} = \begin{bmatrix} 0 & -1 & 6 \\ 0 & 1 & -4 \end{bmatrix}$ , then the solution  $\mathbf{x}_{LS} = [1/2, -1/2]^\top$  satisfies  $\mathbf{B}\mathbf{Ax} = \mathbf{B}\mathbf{b}$ . This example illustrates why the normal equation can lead to the least squares solution. Multiplying a linear system on the left alters the solution set, effectively projecting the problem into a different subspace where a least squares solution exists.  $\square$

**Rank-deficiency.** In this discussion, we assume that the matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  has full rank with  $m \geq n$ , ensuring that  $\mathbf{A}^\top \mathbf{A}$  is invertible. However, if two or more columns of  $\mathbf{A}$  are perfectly correlated, the matrix  $\mathbf{A}$  becomes deficient, and  $\mathbf{A}^\top \mathbf{A}$  becomes singular. To address this issue, one can select the vector  $\mathbf{x}$  that minimizes  $\|\mathbf{x}_{LS}\|_2^2$  while satisfying the normal equation. That is, we choose the least squares solution with the smallest magnitude. In Section 11.6, we briefly discussed how to use UTV decomposition and SVD to solve such rank-deficient least squares problems. See Problems 12.1~12.3 or the following paragraph for further insights.

**Regularizations and stability.** A common issue that arises in the ordinary least square solution is the near-singularity of the matrix  $\mathbf{A}$ . Let the SVD of  $\mathbf{A}$  be  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top \in \mathbb{R}^{m \times n}$ , where  $\mathbf{U} \in \mathbb{R}^{m \times m}$  and  $\mathbf{V} \in \mathbb{R}^{n \times n}$  are orthogonal, and the main diagonal of  $\Sigma \in \mathbb{R}^{m \times n}$  contains the singular values of  $\mathbf{A}$ . Consequently,  $\mathbf{A}^\top \mathbf{A} = \mathbf{V}(\Sigma^\top \Sigma)\mathbf{V}^\top = \mathbf{V}\mathbf{S}\mathbf{V}^\top$ , where  $\mathbf{S} = \Sigma^\top \Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_n^2) \in \mathbb{R}^{n \times n}$  contains the squared singular values of  $\mathbf{A}$ . When  $\mathbf{A}$  is nearly singular,  $\sigma_n^2 \approx 0$ , making the inverse operation  $(\mathbf{A}^\top \mathbf{A})^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{V}^\top$  numerically unstable. As a result, the least squares solution  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  may become highly sensitive or even diverge. To address this instability, an  $\ell_2$  regularization

term is typically added, leading to the solution of the following optimization problem:

$$\mathbf{x}_{Tik} = \arg \min_{\mathbf{x}} \|\mathbf{b} - \mathbf{Ax}\|_2^2 + \lambda \|\mathbf{x}\|_2^2. \quad (12.3)$$

This approach is known as the *Tikhonov regularization method* (or simply the  $\ell_2$  regularized method) (Tikhonov, 1963). The gradient of the problem is  $2(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})\mathbf{x} - 2\mathbf{A}^\top \mathbf{b}$ . Thus, the least squares solution is given by  $\mathbf{x}_{Tik} = (\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{b}$ . The inverse operation becomes  $(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})^{-1} = \mathbf{V}(\mathbf{S} + \lambda \mathbf{I})^{-1} \mathbf{V}^\top$ , where  $\mathbf{S} = (\mathbf{S} + \lambda \mathbf{I}) = \text{diag}(\sigma_1^2 + \lambda, \sigma_2^2 + \lambda, \dots, \sigma_n^2 + \lambda)$ . The solutions for OLS and Tikhonov regularized LS are given, respectively, by

$$\begin{aligned} \mathbf{x}_{LS} &= (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{V}(\mathbf{S}^{-1} \Sigma) \mathbf{U}^\top \mathbf{b}; \\ \mathbf{x}_{Tik} &= (\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{b} = \mathbf{V}((\mathbf{S} + \lambda \mathbf{I})^{-1} \Sigma) \mathbf{U}^\top \mathbf{b}, \end{aligned} \quad (12.4)$$

where the main diagonals of  $(\mathbf{S}^{-1} \Sigma)$  are  $\text{diag}(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \dots, \frac{1}{\sigma_n^2})$ ; and the main diagonals of  $((\mathbf{S} + \lambda \mathbf{I})^{-1} \Sigma)$  are  $\text{diag}(\frac{\sigma_1^2}{\sigma_1^2 + \lambda}, \frac{\sigma_2^2}{\sigma_2^2 + \lambda}, \dots, \frac{\sigma_n^2}{\sigma_n^2 + \lambda})$ . The latter solution is more stable if  $\lambda$  is greater than the smallest nonzero squared singular value. The *condition number*, which measures the sensitivity of the problem to perturbations, becomes smaller if the smallest singular value  $\sigma_n$  is close to zero:

$$\kappa(\mathbf{A}^\top \mathbf{A}) = \frac{\sigma_1^2}{\sigma_n^2} \quad \rightarrow \quad \kappa(\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I}) = \frac{\lambda + \sigma_1^2}{\lambda + \sigma_n^2}.$$

Thus, Tikhonov regularization effectively prevents divergence in the least squares solution  $\mathbf{x}_{LS} = (\mathbf{A}^\top \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{b}$  when the matrix  $\mathbf{A}$  is nearly singular or even rank-deficient. This improvement enhances the convergence properties of both the LS algorithm and its variants, such as alternating least squares, while addressing identifiability issues in various settings (see Section 12.3). As a result, Tikhonov regularization has become a widely applied technique.

**Exercise 12.5.** Use SVD to show that the optimum solution  $\mathbf{x}_{Tik} = (\mathbf{A}^\top \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^\top \mathbf{b}$  has non-increasing norm with increasing  $\lambda$ .

**Data least squares.** While the OLS method accounts for errors in the response variable  $\mathbf{b}$ , the *data least squares (DLS)* method considers errors in the predictor variables:

$$\mathbf{x}_{DLS} = \arg \min_{\mathbf{x}, \tilde{\mathbf{A}}} \left\| \tilde{\mathbf{A}} \right\|_F^2, \quad \text{s.t. } \mathbf{b} \in \mathcal{C}(\mathbf{A} + \tilde{\mathbf{A}}), \quad (12.5)$$

where  $\tilde{\mathbf{A}}$  represents a perturbation in the matrix  $\mathbf{A}$  (i.e., a noise in the predictor variables). That is,  $(\mathbf{A} + \tilde{\mathbf{A}})\mathbf{x}_{DLS} = \mathbf{b}$ , assuming the measured response  $\mathbf{b}$  is noise-free. The Lagrangian function and its gradient w.r.t.  $\mathbf{x}$  are, respectively, given by

$$\begin{aligned} L(\mathbf{x}, \tilde{\mathbf{A}}, \boldsymbol{\lambda}) &= \text{tr}(\tilde{\mathbf{A}} \tilde{\mathbf{A}}^\top) + \boldsymbol{\lambda}^\top (\mathbf{A}\mathbf{x} + \tilde{\mathbf{A}}\mathbf{x} - \mathbf{b}); \\ \nabla_{\tilde{\mathbf{A}}} L(\mathbf{x}, \tilde{\mathbf{A}}, \boldsymbol{\lambda}) &= \tilde{\mathbf{A}} + \boldsymbol{\lambda}\mathbf{x}^\top = \mathbf{0} \implies \tilde{\mathbf{A}} = -\boldsymbol{\lambda}\mathbf{x}^\top, \end{aligned}$$

where  $\boldsymbol{\lambda} \in \mathbb{R}^m$  is a vector of Lagrange multipliers. Substituting the value of the vanishing gradient into  $(\mathbf{A} + \tilde{\mathbf{A}})\mathbf{x} = \mathbf{b}$  yields  $\boldsymbol{\lambda} = \frac{\mathbf{A}\mathbf{x} - \mathbf{b}}{\mathbf{x}^\top \mathbf{x}}$  and  $\tilde{\mathbf{A}} = -\frac{(\mathbf{A}\mathbf{x} - \mathbf{b})\mathbf{x}^\top}{\mathbf{x}^\top \mathbf{x}}$ . Therefore, using the invariance of the trace under cyclic permutations, the objective function becomes

$$\arg \min_{\mathbf{x}} \frac{(\mathbf{A}\mathbf{x} - \mathbf{b})^\top (\mathbf{A}\mathbf{x} - \mathbf{b})}{\mathbf{x}^\top \mathbf{x}}.$$

**Total least squares.** Similar to data least squares, the *total least squares (TLS)* method accounts for errors in both the predictor variables and the response variables. The TLS problem can be formulated as:

$$\mathbf{x}_{TLS} = \arg \min_{\mathbf{x}, \tilde{\mathbf{A}}, \tilde{\mathbf{b}}} \left\| [\tilde{\mathbf{A}}, \tilde{\mathbf{b}}] \right\|_F^2, \quad \text{s.t. } (\mathbf{b} + \tilde{\mathbf{b}}) \in \mathcal{C}(\mathbf{A} + \tilde{\mathbf{A}}), \quad (12.6)$$

where  $\tilde{\mathbf{A}}$  and  $\tilde{\mathbf{b}}$  represent perturbations in the predictor variables and the response variable, respectively. To simplify, define  $\mathbf{C} = [\mathbf{A}, \mathbf{b}] \in \mathbb{R}^{m \times (n+1)}$ ,  $\mathbf{D} = [\tilde{\mathbf{A}}, \tilde{\mathbf{b}}] \in \mathbb{R}^{m \times (n+1)}$ , and  $\mathbf{y} \in \begin{bmatrix} \mathbf{x} \\ -1 \end{bmatrix}$ , the problem can be equivalently stated as

$$\mathbf{x}_{TLS} = \arg \min_{\mathbf{y}, \mathbf{D}} \|\mathbf{D}\|_F^2, \quad \text{s.t. } \mathbf{Dy} = -\mathbf{Cy}, \quad (12.7)$$

## 12.2. Netflix Recommender and Matrix Factorization

The rapid advancements in sensor technology and computer hardware have led to an explosion in the volume of data, presenting new challenges for data analysis. This data is often vast, noisy, and distorted, necessitating preprocessing to enable effective scientific inference. For instance, signals captured by antenna arrays are frequently contaminated by noise and other forms of degradation. To analyze such data effectively, it is essential to reconstruct or represent it in a manner that reduces inaccuracies while adhering to feasibility conditions.

In many cases, data collected from complex systems arises from multiple interrelated variables acting in unison. When these variables are not well-defined, the original data may contain overlapping or ambiguous information. By constructing a simplified system model, it is possible to achieve a level of accuracy comparable to that of the original system. A common approach to noise reduction, model simplification, data compression, and reconstruction is to replace the original data with a lower-dimensional representation obtained through subspace approximation. As a result, *low-rank matrix approximations (LRMA)* or *low-rank matrix decompositions* play a central role in many applications, such as data compression, feature selection, and noise filtering.<sup>4</sup>

Low-rank matrix decomposition is a powerful tool in machine learning and data mining for expressing a given matrix as the product of two or more matrices with lower dimensions. It captures the essential structure of a matrix while filtering out noise and redundancies. Common methods for low-rank matrix decomposition include singular value decomposition (SVD), principal component analysis (PCA), multiplicative update nonnegative matrix factorization (NMF), and the alternating least squares (ALS) approach, which will be introduced in this section.

### Example: The Netflix Prize

For example, in the Netflix Prize competition (Bennett et al., 2007), the goal is to predict the ratings of users for different movies, given the existing ratings (resp., interaction) of those users for other movies (resp., items). We index  $M$  movies with  $m = 1, 2, \dots, M$

---

<sup>4</sup>. Strictly speaking, the term “approximation” usually refers to representing a matrix  $\mathbf{A}$  as  $\mathbf{A} \approx \mathbf{WZ}$ , where  $\mathbf{W}$  and  $\mathbf{Z}$  are matrices whose product approximates  $\mathbf{A}$ . Conversely, the term “decomposition” typically implies that  $\mathbf{A}$  is exactly represented as  $\mathbf{A} = \mathbf{WZ}$ . In this context, however, we use the terms approximation and decomposition interchangeably to refer to both exact and approximate matrix representations.

and  $N$  users with  $n = 1, 2, \dots, N$ . (In the matrix approximation context, lowercase letters e.g.,  $m, n, k$ , are used for the subscripts in running indices, while uppercase letters  $M, N, K$  denote the upper bound of an index.) We denote the rating of the  $n$ -th user for the  $m$ -th movie by  $a_{mn}$ . Define  $\mathbf{A}$  as an  $M \times N$  rating matrix (a *movie-by-user matrix*) with columns  $\{\mathbf{a}_n\} \in \mathbb{R}^M$ , each representing the ratings provided by the  $n$ -th user (also referred to as the *preference matrix*). Note that many ratings  $\{a_{mn}\}$  are missing, and our goal is to predict these missing ratings accurately, i.e., to complete the matrix.

It is clear that without some inherent structure in the matrix, and consequently in the way users rate items, there would be no relationship between the observed and unobserved entries. This would mean there is no unique method to complete the matrix. Therefore, it is crucial to impose some structure on the matrix. A common structural assumption is that of low rank: we aim to fill in the missing entries of matrix  $\mathbf{A}$ , assuming  $\mathbf{A}$  is a low-rank matrix. This assumption makes the problem well-posed and allows for a unique solution to some extent, as the low-rank structure establishes connections between the matrix entries (i.e., a *matrix completion* problem). Consequently, the unobserved entries can no longer be independent of the observed values.<sup>5</sup> It is important to note that, except for very special data structures, a matrix cannot be compressed/decomposed without incurring some compression error, since a low-rank matrix representation is only an approximation of the original matrix. This procedure, often known as *collaborative filtering*, seeks to exploit co-occurring patterns in the observed behaviors across users in order to predict future behaviors of users.

### Matrix Completion Formulation

Consider the *mask matrix*  $\mathbf{M} \in \{0, 1\}^{M \times N}$ , where  $m_{mn} \in \{0, 1\}$  indicates whether user  $n$  has rated movie  $m$  or not. Then the low-rank matrix completion problem can be formulated as

$$\tilde{\mathbf{A}} = \arg \min_{\mathbf{X} \in \mathbb{R}^{M \times N}} \sum_{m,n=1}^{M,N} (x_{mn} - a_{mn})^2 \cdot m_{mn} \quad \text{s.t.} \quad \text{rank}(\mathbf{X}) \leq K. \quad (12.8)$$

However, this problem is NP-hard (non-deterministic polynomial) (Hardt et al., 2014). While it can be equivalently written (proof from singular value decomposition) in the following unconstrained form:

$$\tilde{\mathbf{A}} = \widetilde{\mathbf{W}} \widetilde{\mathbf{Z}} = \arg \min_{\substack{\mathbf{W} \in \mathbb{R}^{M \times K} \\ \mathbf{Z} \in \mathbb{R}^{K \times N}}} \sum_{m,n=1}^{M,N} ((\mathbf{W} \mathbf{Z})_{mn} - a_{mn})^2 \cdot m_{mn}, \quad (12.9)$$

which allows for indirect solution or approximation using alternate algorithms.

We then 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  $\mathbf{Z}$ . Typically,  $K$  is selected to be smaller than both  $M$  and  $N$ , ensuring that  $\mathbf{W}$  and  $\mathbf{Z}$  have reduced dimensions compared to the original matrix  $\mathbf{A}$ . This reduction in dimensionality results in a compressed representation of the original data matrix. An appropriate decision on the

<sup>5</sup> It is worth noting that the low-rank assumption can be quite strong. For example, consider a rank- $r$  matrix  $\mathbf{A} = \sum_{i=1}^r \mathbf{e}_i \tilde{\mathbf{e}}_j^\top$ , where  $\mathbf{e}_i$  and  $\tilde{\mathbf{e}}_j$  are the standard bases for  $\mathbb{R}^M$  and  $\mathbb{R}^N$ , respectively. Such a matrix contains only  $r$  nonzero entries. In real-world recommendation systems, we typically observe only a small fraction of matrix entries, which introduces the possibility that some entries may never be observed. This poses a significant challenge for matrix completion, but this topic is beyond the scope of this book.

value of  $K$  is critical in practice; but the choice of  $K$  is very often problem-dependent. The factorization is significant in the sense that if  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]$  and  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$  are the column partitions of  $\mathbf{A}$  and  $\mathbf{Z}$ , respectively, then we have  $\mathbf{a}_n = \mathbf{W}\mathbf{z}_n$ . This means 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, the columns of  $\mathbf{W}$  can be thought of as containing the column basis (*template columns*, or the approximation of the column basis) of  $\mathbf{A}$ ; and  $\mathbf{z}_n$  indicates the coordinates (or *activations*) of  $\mathbf{a}_n$  in the basis  $\mathbf{W}$ . This concept is similar to the factorization methods discussed in the data interpretation part (Part III). The key difference is that we do not restrict  $\mathbf{W}$  to consist of exact columns from  $\mathbf{A}$ .

---

**Algorithm 11** 2-Block Coordinate Descent: Framework of Most ALS and NMF Algorithms

---

**Require:** A loss function for a variable with two blocks  $\mathbf{X} = (\mathbf{W}, \mathbf{Z})$ :  $f(\mathbf{X}) = f(\mathbf{W}, \mathbf{Z})$ , and data matrix  $\mathbf{A}$ ;

**Ensure:** Constraint on  $\mathbf{W}$  and  $\mathbf{Z}$ ;

- 1: Generate some initial matrices  $\mathbf{W}^{(0)}$  and  $\mathbf{Z}^{(0)}$ ;
  - 2: **for**  $t = 1, 2, \dots$  **do**
  - 3:      $\mathbf{W}^{(t)} \leftarrow \text{update}(\mathbf{A}, \mathbf{Z}^{(t-1)}, \mathbf{W}^{(t-1)})$ ;
  - 4:      $\mathbf{Z}^{(t)} \leftarrow \text{update}(\mathbf{A}, \mathbf{W}^{(t)}, \mathbf{Z}^{(t-1)})$ ;
  - 5: **end for**
- 

However, in most cases, the resulting factorization problem has no exact solution, thus requiring optimization procedures to find suitable numerical approximations. The problem is usually solved using a *two-block coordinate descent (2-BCD)* approach (see Algorithm 11 for a general illustration). In order to obtain the approximation  $\mathbf{A} \approx \mathbf{WZ}$ , we must establish a loss function such that the distance between  $\mathbf{A}$  and  $\mathbf{WZ}$  can be measured. In our discussion, the chosen loss function is the Frobenius norm (a.k.a., the Euclidean distance, Definition 0.27) between two matrices, which vanishes to zero if  $\mathbf{A} = \mathbf{WZ}$ , and its advantages will become evident shortly.

To simplify the problem, let's first assume that there are no missing ratings. We project the data vectors  $\mathbf{a}_n \in \mathbb{R}^M$  into a lower dimension  $\mathbf{z}_n \in \mathbb{R}^K$  with  $K < \min\{M, N\}$  in a way that the *reconstruction error* (a.k.a., *criterion function*, *objective function*, *cost function*, or *loss function*) as measured by the Frobenius norm (a.k.a., sum of squared loss) is minimized (assume  $K$  is known):

$$L(\mathbf{W}, \mathbf{Z}) = D(\mathbf{A}, \mathbf{WZ}) = \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2 = \frac{1}{2} \|\mathbf{WZ} - \mathbf{A}\|_F^2, \quad (12.10)$$

where  $\mathbf{W} = [\mathbf{w}_1^\top; \mathbf{w}_2^\top; \dots; \mathbf{w}_M^\top] \in \mathbb{R}^{M \times K}$  and  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N] \in \mathbb{R}^{K \times N}$  contain  $\mathbf{w}_m$ 's and  $\mathbf{z}_n$ 's as **rows** and **columns**, respectively. In (12.10),  $L(\mathbf{W}, \mathbf{Z})$  indicates that it is a loss function w.r.t.  $\mathbf{W}$  and  $\mathbf{Z}$ , and  $D(\mathbf{A}, \mathbf{WZ})$  implies it is a distance/divergence<sup>7</sup> between  $\mathbf{A}$  and  $\mathbf{WZ}$  (we will use the two terms interchangeably when necessary).

- 
6. Note that we include a scaling factor of  $\frac{1}{2}$  for easier discussion of gradients. Minimizing over  $\frac{1}{2} \|\mathbf{WZ} - \mathbf{A}\|_F^2$  is equivalent to minimizing over  $\|\mathbf{WZ} - \mathbf{A}\|_F^2$  or  $\|\mathbf{WZ} - \mathbf{A}\|_F$ . The choice of the Frobenius norm assumes i.i.d. Gaussian noise on the data ( $\mathbf{A} = \mathbf{WZ} + \mathbf{N}$ , where each entry of  $\mathbf{N}$  follows i.i.d. Gaussian noise) and leads to a smooth optimization via least squares. When the loss is measured by the  $\ell_1$  matrix norm, one obtains a robust low-rank matrix factorization; and the noise is assumed i.i.d. Laplace. See Lu (2021c) for more details.
  7. In words, the *distance*  $D(\mathbf{E}, \mathbf{F})$  indicates  $D(\mathbf{E}, \mathbf{F}) = D(\mathbf{F}, \mathbf{E}) \geq 0$  and the equality holds if and only if  $\mathbf{E} = \mathbf{F}$ ; while the *divergence* holds that  $D(\mathbf{E}, \mathbf{F}) \neq D(\mathbf{F}, \mathbf{E}) \geq 0$  and the equality holds if and only if  $\mathbf{E} = \mathbf{F}$ .

Moreover, the loss function  $L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|_F^2$  is convex<sup>8</sup> concerning  $\mathbf{Z}$  when  $\mathbf{W}$  is held constant, and analogously, convex with respect to  $\mathbf{W}$  when  $\mathbf{Z}$  is fixed. This characteristic motivates an alternating algorithm that alternately fixes one of the variables and optimizes over the other. Therefore, we can first minimize the loss with respect to  $\mathbf{Z}$  while keeping  $\mathbf{W}$  fixed, and subsequently minimize it with respect to  $\mathbf{W}$  with  $\mathbf{Z}$  fixed. This leads to two optimization subproblems, denoted by ALS1 and ALS2, respectively:

$$\begin{cases} \mathbf{Z} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}); & (\text{ALS1}) \\ \mathbf{W} \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W}, \mathbf{Z}). & (\text{ALS2}) \end{cases}$$

This approach is known as the *two-block coordinate descent (2-BCD) algorithm* as mentioned previously, where we alternate between optimizing the least squares with respect to  $\mathbf{W}$  and  $\mathbf{Z}$ . Hence, it is also referred to as the *alternating least squares (ALS)* algorithm (Comon et al., 2009; Takács and Tikk, 2012; Giampouras et al., 2018). Convergence is guaranteed if the loss function  $L(\mathbf{W}, \mathbf{Z})$  decreases at each iteration, and we shall discuss this further in the sequel.

**Remark 12.6 (Convexity and global minimum).** Although the loss function defined by the Frobenius norm  $\frac{1}{2} \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|_F^2$  is convex either with respect to  $\mathbf{W}$  when  $\mathbf{Z}$  is fixed or vice versa (called *marginally convex*), it is not *jointly convex* in both variables simultaneously. Therefore, locating the global minimum is generally infeasible. However, the algorithm is guaranteed to converge to a local minimum.

More generally, let  $D(\mathbf{A}, \mathbf{B})$  be convex in the second argument  $\mathbf{B}$ . Then,  $D(\mathbf{A}, \mathbf{W}\mathbf{Z})$  is convex in  $\mathbf{W}$  when  $\mathbf{Z}$  is fixed, and vice versa; see Problem 12.5.

### Given $\mathbf{W}$ , Optimizing $\mathbf{Z}$

Now, let's examine 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 refer to it as the *least squares* minimizer of  $\arg \min_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z})$ . With  $\mathbf{W}$  fixed,  $L(\mathbf{W}, \mathbf{Z})$  can be represented as  $L(\mathbf{Z}|\mathbf{W})$  (or more compactly, as  $L(\mathbf{Z})$ ) to emphasize its dependence on  $\mathbf{Z}$ :

$$2L(\mathbf{Z}|\mathbf{W}) = \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|_F^2 = \|\mathbf{W}[\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N] - [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]\|^2 = \left\| \begin{bmatrix} \mathbf{W}\mathbf{z}_1 - \mathbf{a}_1 \\ \mathbf{W}\mathbf{z}_2 - \mathbf{a}_2 \\ \vdots \\ \mathbf{W}\mathbf{z}_N - \mathbf{a}_N \end{bmatrix} \right\|_F^2.$$

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 ordinary least squares problem for minimizing  $\|\widetilde{\mathbf{W}}\widetilde{\mathbf{z}} - \widetilde{\mathbf{a}}\|_2^2$  with respect to  $\widetilde{\mathbf{z}}$ . And the solution is given by  $\widetilde{\mathbf{z}} = (\widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}})^{-1} \widetilde{\mathbf{W}}^\top \widetilde{\mathbf{a}}$ . However, it is not advisable to obtain the result using this approach, as computing the

8. A set  $\mathbb{S} \subseteq \mathbb{R}^n$  is *convex* if for all  $\mathbf{x}, \mathbf{y} \in \mathbb{S}$  and  $\lambda \in [0, 1]$ , the point  $(1 - \lambda)\mathbf{x} + \lambda\mathbf{y}$  also belongs to  $\mathbb{S}$ .

A function  $f : \mathbb{S} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$  is *convex* on a convex set  $\mathbb{S}$  if for all  $\mathbf{x}, \mathbf{y} \in \mathbb{S}$  and  $\lambda \in [0, 1]$ , it holds that  $f((1 - \lambda)\mathbf{x} + \lambda\mathbf{y}) \leq (1 - \lambda)f(\mathbf{x}) + \lambda f(\mathbf{y})$ . If the inequality is strict for all  $\mathbf{x} \neq \mathbf{y}$  and  $\lambda \in (0, 1)$ , then  $f$  is *strictly convex*.

inverse of  $\widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}}$  requires  $2(KN)^3$  flops (Lu, 2021c). Alternatively, a more direct way to solve the (ALS1) problem is to find the gradient of  $L(\mathbf{Z}|\mathbf{W})$  with respect to  $\mathbf{Z}$  (assuming all partial derivatives of this function exist):

$$\nabla_{\mathbf{Z}} L(\mathbf{Z}|\mathbf{W}) = \frac{1}{2} \frac{\partial \text{tr}((\mathbf{W}\mathbf{Z} - \mathbf{A})(\mathbf{W}\mathbf{Z} - \mathbf{A})^\top)}{\partial \mathbf{Z}} \stackrel{*}{=} \mathbf{W}^\top (\mathbf{W}\mathbf{Z} - \mathbf{A}) \in \mathbb{R}^{K \times N}, \quad (12.11)$$

where the first equality arises from the definition of the Frobenius norm (Definition 0.27) such that  $\|\mathbf{A}\|_F = \sqrt{\sum_{m=1,n=1}^{M,N} (a_{mn})^2} = \sqrt{\text{tr}(\mathbf{A}\mathbf{A}^\top)}$ , and the equality  $(*)$  is a consequence of 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 can determine the least squares solution using differential calculus. Since we optimize over an open set  $\mathbb{R}^{K \times N}$ , any minimum of the function  $L(\mathbf{Z}|\mathbf{W})$  must satisfy the condition:

$$\nabla_{\mathbf{Z}} L(\mathbf{Z}|\mathbf{W}) = \mathbf{0}.$$

Solving this equation yields the “candidate” update for  $\mathbf{Z}$  that minimizes  $L(\mathbf{Z}|\mathbf{W})$ :

$$(\text{“Candidate” update for } \mathbf{Z}) : \quad \mathbf{Z} = (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{A} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{Z}|\mathbf{W}). \quad (12.12)$$

This computation requires  $2K^3$  flops to compute the inverse of  $\mathbf{W}^\top \mathbf{W}$ , compared to  $2(KN)^3$  flops to get the inverse of  $\widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}}$ . Prior to confirming that a root of the equation above is indeed a minimizer (as opposed to a maximizer, hence the term “candidate” update), it is imperative to establish the convexity of the function. For a twice continuously differentiable function, this verification can be equivalently achieved by confirming (see Problem 12.19 for more details):

$$\nabla_{\mathbf{Z}}^2 L(\mathbf{Z}|\mathbf{W}) \succ 0. \quad ^9$$

That is, the Hessian matrix is positive definite (Definition 2.2; see, for example, Beck (2014)). To demonstrate this, we explicitly express the Hessian matrix as

$$\nabla_{\mathbf{Z}}^2 L(\mathbf{Z}|\mathbf{W}) = \widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}} \in \mathbb{R}^{KN \times KN}, \quad ^{10}$$

which maintains full rank if  $\mathbf{W} \in \mathbb{R}^{M \times K}$  has full rank and  $K < M$  (Lemma 11.3).

**Remark 12.7 (Positive definite Hessian if  $\mathbf{W}$  has full rank).** We claim that if  $\mathbf{W} \in \mathbb{R}^{M \times K}$  has full rank  $K$  with  $K < M$ , then  $\nabla_{\mathbf{Z}}^2 L(\mathbf{Z}|\mathbf{W})$  is positive definite. This can be demonstrated by confirming that when  $\mathbf{W}$  has full rank, the equation  $\mathbf{W}\mathbf{x} = \mathbf{0}$  holds true only when  $\mathbf{x} = \mathbf{0}$ , since the null space of  $\mathbf{W}$  has dimension zero. Therefore,

$$\mathbf{x}^\top (\mathbf{W}^\top \mathbf{W}) \mathbf{x} > 0, \quad \text{for any nonzero vector } \mathbf{x} \in \mathbb{R}^K.$$

And this in turn implies  $\widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}} \succ \mathbf{0}$ .

- 
- 9. In short, a twice continuously differentiable function  $f$  over an open convex set  $\mathbb{S}$  is called *convex* if and only if  $\nabla^2 f(\mathbf{x}) \succeq \mathbf{0}$  for any  $\mathbf{x} \in \mathbb{S}$  (sufficient and necessary for convex); and called *strictly convex* if  $\nabla^2 f(\mathbf{x}) \succ \mathbf{0}$  for any  $\mathbf{x} \in \mathbb{S}$  (only sufficient for strictly convex, e.g.,  $f(x) = x^6$  is strictly convex, but  $f''(x) = 30x^4$  is equal to zero at  $x = 0$ ). And when the convex function  $f$  is a continuously differentiable function over a convex set  $\mathbb{S}$ , the stationary point  $\nabla f(\mathbf{x}^*) = \mathbf{0}$  of  $\mathbf{x}^* \in \mathbb{S}$  is a *global minimizer* of  $f$  over  $\mathbb{S}$ . In our context, when given  $\mathbf{W}$  and updating  $\mathbf{Z}$ , the function is defined over the entire space  $\mathbb{R}^{K \times N}$ .
  - 10. A block-diagonal matrix whose block matrix on the diagonal is  $\mathbf{W}^\top \mathbf{W}$ . And it can be equivalently denoted as  $\nabla_{\mathbf{Z}}^2 L(\mathbf{Z}|\mathbf{W}) = \text{diag}(\mathbf{W}, \mathbf{W}, \dots, \mathbf{W})^\top \text{diag}(\mathbf{W}, \mathbf{W}, \dots, \mathbf{W})$ . Using the Kronecker product “ $\otimes$ ”, this can be equivalently written as  $\nabla_{\mathbf{Z}}^2 L(\mathbf{Z}|\mathbf{W}) = \mathbf{I}_N \otimes (\mathbf{W}^\top \mathbf{W})$ , where  $\mathbf{I}_N$  is the  $N \times N$  identity matrix.

Now, the problem becomes **whether  $\mathbf{W}$  has full rank so that the Hessian of  $L(\mathbf{Z}|\mathbf{W})$  is positive definite**; otherwise, we cannot claim the update of  $\mathbf{Z}$  in Equation (12.12) reduces the loss (due to convexity) so that the matrix decomposition progressively improves the approximation of the original matrix  $\mathbf{A}$  by  $\mathbf{W}\mathbf{Z}$  in each iteration. We will address the positive definiteness of the Hessian matrix shortly, relying on the following lemma.

**Lemma 12.8: (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  $K < M$  (i.e.,  $K < M \leq N$ ). 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 (12.12) has full rank.

**Proof** [of Lemma 12.8] Since  $\mathbf{W}^\top \mathbf{W} \in \mathbb{R}^{K \times K}$  has full rank if  $\mathbf{W}$  has full rank (Lemma 11.3), it follows that  $(\mathbf{W}^\top \mathbf{W})^{-1}$  has full rank.

Suppose  $\mathbf{W}^\top \mathbf{x} = \mathbf{0}$ . This implies that  $(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{x} = \mathbf{0}$ . Thus, the following two null spaces satisfy:  $\mathcal{N}(\mathbf{W}^\top) \subseteq \mathcal{N}((\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top)$ . Moreover, suppose  $\mathbf{x}$  lies in the null space of  $(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top$  such that  $(\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{x} = \mathbf{0}$ . And since  $(\mathbf{W}^\top \mathbf{W})^{-1}$  is invertible, it implies  $\mathbf{W}^\top \mathbf{x} = (\mathbf{W}^\top \mathbf{W})\mathbf{0} = \mathbf{0}$ , leading to  $\mathcal{N}((\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top) \subseteq \mathcal{N}(\mathbf{W}^\top)$ . Consequently, through “sandwiching,” it follows that

$$\mathcal{N}(\mathbf{W}^\top) = \mathcal{N}((\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top). \quad (12.14)$$

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  $\mathbf{A}^\top \mathbf{T}^\top \mathbf{x} = \mathbf{0}$ , yielding  $\mathcal{N}(\mathbf{T}^\top) \subseteq \mathcal{N}(\mathbf{A}^\top \mathbf{T}^\top)$ . 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 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  $\dim(\mathcal{N}(\mathbf{A}^\top)) = M - \dim(\mathcal{C}(\mathbf{A})) = 0$ . Consequently,  $\mathbf{x}$  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, we obtain

$$\mathcal{N}(\mathbf{T}^\top) = \mathcal{N}(\mathbf{A}^\top \mathbf{T}^\top). \quad (12.15)$$

Since  $\mathbf{T}^\top$  has full rank  $K < M \leq N$ , it follows that  $\dim(\mathcal{N}(\mathbf{T}^\top)) = \dim(\mathcal{N}(\mathbf{A}^\top \mathbf{T}^\top)) = 0$ . Therefore,  $\mathbf{Z}^\top = \mathbf{A}^\top \mathbf{T}^\top$  has full rank  $K$ . We complete the proof. ■

### Given $\mathbf{Z}$ , Optimizing $\mathbf{W}$

The matrix factorization problem exhibits symmetry:  $\mathbf{A} = \mathbf{W}\mathbf{Z}$  if and only if  $\mathbf{A}^\top = \mathbf{Z}^\top \mathbf{W}^\top$  such that  $D(\mathbf{A}, \mathbf{W}\mathbf{Z}) = D(\mathbf{A}^\top, \mathbf{Z}^\top \mathbf{W}^\top)$ . The analysis of optimizing  $\mathbf{W}$  given  $\mathbf{Z}$  directly follows from the previously discussed methodology. Below, we provide a brief outline of the results. With  $\mathbf{Z}$  fixed,  $L(\mathbf{W}, \mathbf{Z})$  can be expressed as  $L(\mathbf{W}|\mathbf{Z})$  (or more compactly, as  $L(\mathbf{W})$ ) to emphasize the dependence on  $\mathbf{W}$ :  $L(\mathbf{W}|\mathbf{Z}) = \frac{1}{2} \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|_F^2$ . To solve the optimization problem (ALS2) directly, we compute the gradient of  $L(\mathbf{W}|\mathbf{Z})$  with respect to  $\mathbf{W}$ :

$$\nabla_{\mathbf{W}} L(\mathbf{W}|\mathbf{Z}) = \frac{1}{2} \frac{\partial \text{tr}((\mathbf{W}\mathbf{Z} - \mathbf{A})(\mathbf{W}\mathbf{Z} - \mathbf{A})^\top)}{\partial \mathbf{W}} = (\mathbf{W}\mathbf{Z} - \mathbf{A})\mathbf{Z}^\top \in \mathbb{R}^{M \times K}. \quad (12.16)$$

Similarly, the “candidate” update for  $\mathbf{W}$  can be obtained by identifying the root of the gradient  $\nabla_{\mathbf{W}} L(\mathbf{W}|\mathbf{Z})$ :

$$(\text{“Candidate” update for } \mathbf{W}): \quad \mathbf{W}^\top = (\mathbf{Z}\mathbf{Z}^\top)^{-1} \mathbf{Z}\mathbf{A}^\top \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W}|\mathbf{Z}). \quad (12.17)$$

Once more, we emphasize that the update is merely a “candidate” update. Further validation is necessary to ascertain the positive definiteness of the Hessian matrix. The Hessian matrix is given by:

$$\nabla_{\mathbf{W}}^2 L(\mathbf{W}|\mathbf{Z}) = \tilde{\mathbf{Z}} \tilde{\mathbf{Z}}^\top \in \mathbb{R}^{KM \times KM}, \quad (12.18)$$

where  $\tilde{\mathbf{Z}} = \text{diag}(\mathbf{Z}, \mathbf{Z}, \dots, \mathbf{Z}) \in \mathbb{R}^{KM \times NM}$  is defined analogously to  $\tilde{\mathbf{W}}$  in (12.13). Therefore, by similar reasoning, if  $\mathbf{Z}$  has full rank with  $K < N$ , the Hessian matrix is positive definite.

In Lemma 12.8, we proved that  $\mathbf{Z}$  has full rank under certain conditions, ensuring that the Hessian matrix in Equation (12.18) is positive definite, and the update in Equation (12.17) exists. We now prove that  $\mathbf{W}$  also has full rank under certain conditions, such that the Hessian in Equation (12.13) is positive definite, and the update in Equation (12.12) exists.

**Lemma 12.9: (Rank of  $\mathbf{W}$  after updating)** Suppose  $\mathbf{A} \in \mathbb{R}^{M \times N}$  has full rank with  $M \geq N$  and  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  has full rank with  $K < N$  (i.e.,  $K < N \leq M$ ). Then the update of  $\mathbf{W}^\top = (\mathbf{Z} \mathbf{Z}^\top)^{-1} \mathbf{Z} \mathbf{A}^\top$  in Equation (12.17) has full rank.

The proof of Lemma 12.9 follows the same reasoning as that of Lemma 12.8, so we omit the details.

**Key observation.** Combining the observations from Lemmas 12.8 and 12.9, as long as we initialize  $\mathbf{Z}$  and  $\mathbf{W}$  to have full rank, the updates in Equations (12.12) and (12.17) are well-defined since the Hessians in Equations (12.13) and (12.18) are positive definite. Note that we need an additional condition to satisfy both Lemma 12.8 and Lemma 12.9:  $M = N$ , meaning there must be an equal number of movies and users. We will relax this condition in the next section through regularization. (Alternatively, Problems 12.1~12.3 relax this condition using the pseudo-inverse.) We summarize the process in Algorithm 12. Since the loss  $\frac{1}{2} \|\mathbf{A} - \mathbf{W}\mathbf{Z}\|_F^2$  in each iteration is monotonically nonincreasing and bounded below, it converges (Lu, 2021c; Gillis, 2020). In particular,  $\nabla_{\mathbf{Z}} L(\mathbf{Z}|\mathbf{W})$  and  $\nabla_{\mathbf{W}} L(\mathbf{W}|\mathbf{Z})$  approach zero when  $\text{iter} \rightarrow \infty$ .

---

### Algorithm 12 Alternating Least Squares

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$  with  $M = N$ ;

- 1: Initialize  $\mathbf{W} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  with full rank and  $K < M = N$ ;
  - 2: Choose a stop criterion on the approximation error  $\delta$ ;
  - 3: Choose the maximal number of iterations  $C$ ;
  - 4:  $\text{iter} = 0$ ; ▷ Count for the number of iterations
  - 5: **while**  $\|\mathbf{A} - \mathbf{W}\mathbf{Z}\|_F > \delta$  and  $\text{iter} < C$  **do**
  - 6:    $\text{iter} = \text{iter} + 1$ ;
  - 7:    $\mathbf{Z} \leftarrow (\mathbf{W}^\top \mathbf{W})^{-1} \mathbf{W}^\top \mathbf{A} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{Z}|\mathbf{W})$ ;
  - 8:    $\mathbf{W}^\top \leftarrow (\mathbf{Z} \mathbf{Z}^\top)^{-1} \mathbf{Z} \mathbf{A}^\top \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W}|\mathbf{Z})$ ;
  - 9: **end while**
  - 10: Output  $\mathbf{W}, \mathbf{Z}$ .
- 

### 12.3. Regularization and Identifiability: Extension to General Matrices

*Regularization* is a machine learning technique employed to prevent overfitting and improve the generalization of models. Overfitting occurs when a model is overly complex and fits

the training data too closely, resulting in poor performance on unseen data. To mitigate this issue, regularization introduces a constraint or a penalty term into the loss function used for model optimization, discouraging the development of overly complex models. This creates a trade-off between having a simple, generalizable model and fitting the training data well. Common types of regularization include  $\ell_1$  regularization,  $\ell_2$  regularization (Tikhonov regularization), and elastic net regularization (a combination of  $\ell_1$  and  $\ell_2$  regularizations). Regularization finds extensive applications in machine learning algorithms such as linear regression, logistic regression, and neural networks.

In the context of the alternating least squares problem, we can incorporate an  $\ell_2$  regularization term to minimize the following regularized loss function:

$$L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|_F^2 + \frac{1}{2}\lambda_w \|\mathbf{W}\|_F^2 + \frac{1}{2}\lambda_z \|\mathbf{Z}\|_F^2, \quad \lambda_w > 0, \lambda_z > 0, \quad (12.19)$$

where the gradient with respect to  $\mathbf{Z}$  and  $\mathbf{W}$  are given, respectively, by

$$\begin{cases} \nabla_{\mathbf{Z}} L(\mathbf{Z}|\mathbf{W}) = \mathbf{W}^\top (\mathbf{W}\mathbf{Z} - \mathbf{A}) + \lambda_z \mathbf{Z} \in \mathbb{R}^{K \times N}; \\ \nabla_{\mathbf{W}} L(\mathbf{W}|\mathbf{Z}) = (\mathbf{W}\mathbf{Z} - \mathbf{A})\mathbf{Z}^\top + \lambda_w \mathbf{W} \in \mathbb{R}^{M \times K}. \end{cases} \quad (12.20)$$

The corresponding Hessian matrices are given, respectively, by

$$\begin{cases} \nabla_{\mathbf{Z}}^2 L(\mathbf{Z}|\mathbf{W}) = \widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}} + \lambda_z \mathbf{I} \in \mathbb{R}^{KN \times KN}; \\ \nabla_{\mathbf{W}}^2 L(\mathbf{W}|\mathbf{Z}) = \widetilde{\mathbf{Z}} \widetilde{\mathbf{Z}}^\top + \lambda_w \mathbf{I} \in \mathbb{R}^{KM \times KM}, \end{cases}$$

which are positive definite due to the perturbation by the regularization:

$$\begin{cases} \mathbf{x}^\top (\widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}} + \lambda_z \mathbf{I}) \mathbf{x} = \underbrace{\mathbf{x}^\top \widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}} \mathbf{x}}_{\geq 0} + \lambda_z \|\mathbf{x}\|_2^2 > 0, & \text{for nonzero } \mathbf{x}; \\ \mathbf{x}^\top (\widetilde{\mathbf{Z}} \widetilde{\mathbf{Z}}^\top + \lambda_w \mathbf{I}) \mathbf{x} = \underbrace{\mathbf{x}^\top \widetilde{\mathbf{Z}} \widetilde{\mathbf{Z}}^\top \mathbf{x}}_{\geq 0} + \lambda_w \|\mathbf{x}\|_2^2 > 0, & \text{for nonzero } \mathbf{x}. \end{cases}$$

**The regularization ensures that the Hessian matrices remain positive definite, even if  $\mathbf{W}$  and  $\mathbf{Z}$  are rank-deficient.** Consequently, matrix decomposition can be extended to any matrix, regardless of whether  $M > N$  or  $M < N$ . In rare cases,  $K$  even can be chosen as  $K > \max\{M, N\}$  to obtain a *high-rank approximation* of  $\mathbf{A}$ . However, in most scenarios, we aim to find a *low-rank approximation* of  $\mathbf{A}$  with  $K < \min\{M, N\}$ . Therefore, the minimizers can be determined by identifying the roots of the gradients:

$$\mathbf{Z} = (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{A} \quad \text{and} \quad \mathbf{W}^\top = (\mathbf{Z} \mathbf{Z}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z} \mathbf{A}^\top. \quad (12.21)$$

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 selection of these parameters is typically problem-dependent and can be determined through *cross-validation* (CS). Again, we summarize the regularized ALS procedure in Algorithm 13. We will also introduce the *alternating direction methods of multipliers* (ADMM) for solving matrix factorization problems with  $\ell_2$  or  $\ell_1$  regularization in Section 13.5, where the method can be extended to other types of regularizations and constraints, such as nonnegativity constraints.

The  $\ell_2$  (or  $\ell_1$ ) regularizations can be applied to generalize the ALS problem to general matrices. However, we will consider the case where some entries of the matrix  $\mathbf{A}$  are missing. This leads to the matrix completion problem. In this sense, the  $\ell_1$  and  $\ell_2$  regularizations

are not the only applicable regularizations; for example, the *nuclear norm*<sup>11</sup> of  $\mathbf{WZ}$  (the sum of singular values of the matrix) can be applied, for which the *Soft-Impute for matrix completion* algorithm guarantees the recovery of the matrix when the number of observed entries  $z$  satisfies  $z \geq Crn \log n$ , where the underlying matrix  $\mathbf{A}$  is of size  $\mathbb{R}^{n \times n}$  and  $C > 0$  is a fixed universal constant (Gross, 2011; Hastie et al., 2015). However, the  $\ell_2$  regularization on  $\mathbf{W}$  and  $\mathbf{Z}$  can somehow be reformulated into the nuclear norm form (see Problem 12.22).

---

**Algorithm 13** Alternating Least Squares with Regularization

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 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 the maximal number of iterations  $C$ ;
- 5:  $iter = 0$ ; ▷ Count for the number of iterations
- 6: **while**  $\|\mathbf{A} - \mathbf{WZ}\|_F > \delta$  and  $iter < C$  **do**
- 7:    $iter = iter + 1$ ;
- 8:    $\mathbf{Z} \leftarrow (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{A} \leftarrow \arg \min_{\mathbf{Z}} L(\mathbf{Z} | \mathbf{W})$ ;
- 9:    $\mathbf{W}^\top \leftarrow (\mathbf{ZZ}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z} \mathbf{A}^\top \leftarrow \arg \min_{\mathbf{W}} L(\mathbf{W} | \mathbf{Z})$ ;
- 10: **end while**
- 11: Output  $\mathbf{W}, \mathbf{Z}$ .

---

**Regularization as constraints and identifiability.** Regularization terms, such as  $\lambda_w \|\mathbf{W}\|_F^2$  in (12.19), can be interpreted as constraints like  $\|\mathbf{W}\|_F \leq C$ , where  $C$  is a constant, via Lagrangian multipliers (see, for example, Boyd et al. (2004) or Section 12.7). Different constraints can be placed on the factors  $\mathbf{W}$  and  $\mathbf{Z}$ . For example, the nonnegativity constraint discussed in Chapter 13 and the sparsity constraint discussed in Section 12.7. Moreover, the two matrices  $\mathbf{W} \in \mathbb{R}^{M \times K}$  and  $\mathbf{Z} \in \mathbb{R}^{K \times N}$  have  $(M + N)K$  degrees of freedom. However, due to the scaling degree of freedom of the columns of  $\mathbf{W}$  and rows of  $\mathbf{Z}$  in  $\mathbf{A} = \mathbf{WZ}$ , the factorization  $\mathbf{WZ}$  has  $(M + N - 1)K$  degrees of freedom:  $\mathbf{W}[:, k]\mathbf{Z}[k, :] = (\gamma \mathbf{W}[:, k])(\frac{1}{\gamma} \mathbf{Z}[k, :])$  for any scalar  $\gamma \neq 0$  and  $k \in \{1, 2, \dots, K\}$ . Therefore, the factorization is not identifiable. Regularization helps reduce overfitting and addresses the issue of identifiability by incorporating prior information through constraints.

## 12.4. Missing Entries and Rank-One Update

Matrix decomposition via ALS is extensively used in the context of Netflix-style recommender data, where a substantial number of entries are missing due to users not having watched certain movies or choosing not to rate them for various reasons. In this scenario, the low-rank matrix decomposition problem is also known as *matrix completion* that can help recover unobserved entries (Jain et al., 2017). To model this, we can introduce an additional mask matrix  $\mathbf{M} \in \{0, 1\}^{M \times N}$ , where each entry  $m_{mn} \in \{0, 1\}$  indicates whether user  $n$  has rated movie  $m$  or not. Using this mask, the loss function can be defined as:

$$L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \|\mathbf{M} \circledast \mathbf{A} - \mathbf{M} \circledast (\mathbf{WZ})\|_F^2,$$

where  $\circledast$  represents the *Hadamard product* between matrices. The above formulation concisely expresses our goal of finding a completion of the ratings matrix that is both of low

---

<sup>11</sup> Also called the *Schatten 1-norm* or *trace norm*.

rank and consistent with observed user ratings. To find the solution to this problem, we decompose the updates in Equation (12.21) into:

$$\begin{cases} \mathbf{z}_n = (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{a}_n, & \text{for } n \in \{1, 2, \dots, N\}; \\ \mathbf{w}_m = (\mathbf{Z} \mathbf{Z}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z} \mathbf{b}_m, & \text{for } m \in \{1, 2, \dots, M\}, \end{cases} \quad (12.22)$$

where  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$  and  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]$  represent the column partitions of  $\mathbf{Z}$  and  $\mathbf{A}$ , respectively. Similarly,  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]$  and  $\mathbf{A}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M]$  are the column partitions of  $\mathbf{W}^\top$  and  $\mathbf{A}^\top$ , respectively. This decomposition of the updates indicates that the updates can be performed in a column-by-column fashion (the rank-one updates).

**Given  $\mathbf{W}$ .** Let  $\mathbf{o}_n \in \{0, 1\}^M$  represent 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 using the Matlab-style notation as  $\mathbf{a}_n[\mathbf{o}_n]$ . And we want to approximate the existing entries of the  $n$ -th column by  $\mathbf{a}_n[\mathbf{o}_n] \approx \mathbf{W}[\mathbf{o}_n, :] \mathbf{z}_n$ , which is indeed a rank-one least squares problem:

$$\mathbf{z}_n = \left( \mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{W}[\mathbf{o}_n, :] + \lambda_z \mathbf{I} \right)^{-1} \mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{a}_n[\mathbf{o}_n], \quad \text{for } n \in \{1, 2, \dots, N\}. \quad (12.23)$$

Moreover, the loss function with respect to  $\mathbf{z}_n$  and  $\mathbf{Z}$  can be described, respectively, by

$$L(\mathbf{z}_n | \mathbf{W}) = \sum_{m \in \mathbf{o}_n} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2 \quad \text{and} \quad L(\mathbf{Z} | \mathbf{W}) = \sum_{n=1}^N \sum_{m \in \mathbf{o}_n} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2.$$

**Given  $\mathbf{Z}$ .** Similarly, if  $\mathbf{p}_m \in \{0, 1\}^N$  denotes the users who have rated movie  $m$ , with  $p_{mn} = 1$  if movie  $m$  has been rated by user  $n$ , and  $p_{mn} = 0$  otherwise. Then the  $m$ -th row of  $\mathbf{A}$  without missing entries can be denoted by the Matlab-style notation as  $\mathbf{b}_m[\mathbf{p}_m]$ . We want to approximate the existing entries of the  $m$ -th row by  $\mathbf{b}_m[\mathbf{p}_m] \approx \mathbf{Z}[:, \mathbf{p}_m]^\top \mathbf{w}_m$ , <sup>12</sup> which is again a rank-one least squares problem:

$$\mathbf{w}_m = (\mathbf{Z}[:, \mathbf{p}_m] \mathbf{Z}[:, \mathbf{p}_m]^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z}[:, \mathbf{p}_m] \mathbf{b}_m[\mathbf{p}_m], \quad \text{for } m \in \{1, 2, \dots, M\}. \quad (12.24)$$

Similarly, the loss function with respect to  $\mathbf{w}_m$  and  $\mathbf{W}$  can be described, respectively, by

$$L(\mathbf{w}_m | \mathbf{Z}) = \sum_{n \in \mathbf{p}_m} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2 \quad \text{and} \quad L(\mathbf{W} | \mathbf{Z}) = \sum_{m=1}^M \sum_{n \in \mathbf{p}_m} \left( a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n \right)^2.$$

The procedure is once again presented in Algorithm 14. Other approaches, such as *singular value projection (SVP)*, also exist to address the matrix completion problem. At a high level, SVP is a type of projected gradient descent (PGD) method that updates iteratively via gradient descent, projecting the updated matrix into a low-rank form through singular value decomposition at each step. However, the alternating least squares approach generally outperforms SVP in the context of matrix completion, so we will not delve into SVP here. For more details, refer to Jain et al. (2017) and the references therein.

---

<sup>12</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.

**Algorithm 14** Alternating Least Squares with Missing Entries and Regularization

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 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 the maximal number of iterations  $C$ ;
- 6:  $iter = 0$ ; ▷ Count for the number of iterations
- 7: **while**  $\|\mathbf{M} \circledast \mathbf{A} - \mathbf{M} \circledast (\mathbf{W}\mathbf{Z})\|_F^2 > \delta$  and  $iter < C$  **do**
- 8:    $iter = iter + 1$ ;
- 9:   **for**  $n = 1, 2, \dots, N$  **do**
- 10:      $\mathbf{z}_n \leftarrow (\mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{W}[\mathbf{o}_n, :] + \lambda_z \mathbf{I})^{-1} \mathbf{W}[\mathbf{o}_n, :]^\top \mathbf{a}_n[\mathbf{o}_n]$ ; ▷  $n$ -th column of  $\mathbf{Z}$
- 11:     **end for**
- 12:     **for**  $m = 1, 2, \dots, M$  **do**
- 13:        $\mathbf{w}_m \leftarrow (\mathbf{Z}[:, \mathbf{p}_m] \mathbf{Z}[:, \mathbf{p}_m]^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z}[:, \mathbf{p}_m] \mathbf{b}_m[\mathbf{p}_m]$ ; ▷  $m$ -th column of  $\mathbf{W}^\top$
- 14:     **end for**
- 15: **end while**
- 16: Output  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M], \mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ .

---

## 12.5. Vector Inner Product and Hidden Vectors

We observe that the ALS algorithm seeks to find lower-dimensional matrices  $\mathbf{W}$  and  $\mathbf{Z}$  such that their product  $\mathbf{W}\mathbf{Z}$  can approximate  $\mathbf{A} \approx \mathbf{W}\mathbf{Z}$  in terms of the squared loss:  $\min_{\mathbf{W}, \mathbf{Z}} \sum_{n=1}^N \sum_{m=1}^M (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2$ . That is, each entry  $a_{mn}$  in  $\mathbf{A}$  can be approximated as the inner product of two vectors:  $\mathbf{w}_m^\top \mathbf{z}_n$ . The geometric interpretation of the vector inner product is given by

$$\mathbf{w}_m^\top \mathbf{z}_n = \|\mathbf{w}_m\|_2 \cdot \|\mathbf{z}_n\|_2 \cos \theta,$$

where  $\theta$  represents the angle between the vectors  $\mathbf{w}_m$  and  $\mathbf{z}_n$ . Thus, if the vector norms of  $\mathbf{w}_m$  and  $\mathbf{z}_n$  are determined, a smaller angle between them results in a larger inner product.

In the context of Netflix-style recommendation systems, movie ratings typically range from 0 to 5, with higher ratings indicating a stronger user preference for the movie. If  $\mathbf{w}_m$  and  $\mathbf{z}_n$  fall sufficiently “close” in direction, the value of  $\mathbf{w}_m^\top \mathbf{z}_n$  becomes larger. This reflects a stronger match between the user’s preferences and the movie’s characteristics.

This concept elucidates the essence of ALS, where  $\mathbf{w}_m$  represents the features or attributes of movie  $m$ , while  $\mathbf{z}_n$  encapsulates the features or preferences of user  $n$ . In other words, ALS associates each user with a *latent vector of preference* and each movie with a *latent vector of attributes*. Furthermore, each element in  $\mathbf{w}_m$  and  $\mathbf{z}_n$  signifies a specific feature. For example, it could be that the second feature  $w_{m2}$  ( $w_{m2}$  denotes the second element of the vector  $\mathbf{w}_m$ ) represents whether the movie is an action movie or not, and  $z_{n2}$  might denote whether user  $n$  has a preference for action movies. When this holds true, then the inner product  $\mathbf{w}_m^\top \mathbf{z}_n$  becomes large and provides a good approximation of the observed rating  $a_{mn}$ .

In the matrix decomposition  $\mathbf{A} \approx \mathbf{W}\mathbf{Z}$ , it is established that 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. Nevertheless, the explicit meanings of the rows in  $\mathbf{W}$  or the columns in  $\mathbf{Z}$  remain undisclosed. Although they might correspond to categories or genres of the movies, fostering

underlying connections between users and movies, their precise nature remains uncertain. It is precisely this ambiguity that gives rise to the terminology “latent” or “hidden.”

## 12.6. Gradient Descent

In Algorithms 12, 13, and 14, we minimize the loss function through the inversion of matrices (e.g., using LU decomposition). The reality, however, is frequently far from straightforward, particularly in the big data era of today. As data volumes explode, the size of the inversion matrix will grow at a pace proportional to the cube of the number of samples, which poses a great challenge to the storage and computational resources. This complexity has led to the ongoing development of gradient-based optimization techniques. Among these, the *gradient descent (GD)* method and its variant, the *stochastic gradient descent (SGD)* method, are among the simplest, fastest, and most efficient methods (Lu, 2022c). These methods are particularly effective for solving convex optimization problems. We now provide a more detailed explanation of their underlying principles.

In Equation (12.22), we derived the column-by-column update rules directly from the full matrix approach outlined in Equation (12.21) (with regularization taken into account). To understand the underlying concept, consider the loss function with regularization, as given by Equation (12.19). When minimizing the loss in (12.19) with respect to  $\mathbf{z}_n$ , we can break down the loss as follows:

$$L(\mathbf{z}_n) = \frac{1}{2} \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|_F^2 + \frac{1}{2}\lambda_w \|\mathbf{W}\|_F^2 + \frac{1}{2}\lambda_z \|\mathbf{Z}\|_F^2 = \frac{1}{2} \|\mathbf{W}\mathbf{z}_n - \mathbf{a}_n\|_2^2 + \frac{1}{2}\lambda_z \|\mathbf{z}_n\|_2^2 + C_{z_n}, \quad (12.25)$$

where  $C_{z_n}$  is a constant with respect to  $\mathbf{z}_n$ , and  $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$  and  $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N]$  represent the column partitions of  $\mathbf{Z}$  and  $\mathbf{A}$ , respectively. The gradient and the root are given, respectively, by

$$\nabla_{\mathbf{z}_n} L(\mathbf{z}_n) = \mathbf{W}^\top \mathbf{W} \mathbf{z}_n - \mathbf{W}^\top \mathbf{a}_n + \lambda_z \mathbf{z}_n \implies \mathbf{z}_n = (\mathbf{W}^\top \mathbf{W} + \lambda_z \mathbf{I})^{-1} \mathbf{W}^\top \mathbf{a}_n, \forall n.$$

This solution corresponds to the first update rule in the column-wise updates of Equation (12.22). Similarly, when minimizing the loss with respect to  $\mathbf{w}_m$ , we have:

$$L(\mathbf{w}_m) = \frac{1}{2} \|\mathbf{Z}^\top \mathbf{W} - \mathbf{A}^\top\|_F^2 + \frac{1}{2}\lambda_w \|\mathbf{W}^\top\|_F^2 + \frac{1}{2}\lambda_z \|\mathbf{Z}\|_F^2 = \frac{1}{2} \|\mathbf{Z}^\top \mathbf{w}_m - \mathbf{b}_n\|_2^2 + \frac{1}{2}\lambda_w \|\mathbf{w}_m\|_2^2 + C_{w_m}, \quad (12.26)$$

where  $C_{w_m}$  is a constant with respect to  $\mathbf{w}_m$ , and  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]$  and  $\mathbf{A}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M]$  represent the column partitions of  $\mathbf{W}^\top$  and  $\mathbf{A}^\top$ , respectively. Analogously, taking the gradient with respect to  $\mathbf{w}_m$ , it follows that

$$\nabla_{\mathbf{w}_m} L(\mathbf{w}_m) = \mathbf{Z} \mathbf{Z}^\top \mathbf{w}_m - \mathbf{Z} \mathbf{b}_n + \lambda_w \mathbf{w}_m \implies \mathbf{w}_m = (\mathbf{Z} \mathbf{Z}^\top + \lambda_w \mathbf{I})^{-1} \mathbf{Z} \mathbf{b}_n, \forall m.$$

This solution corresponds to the second update rule in the column-wise updates of Equation (12.22):

Now suppose we express the iteration number ( $t = 1, 2, \dots$ ) as the superscript, and we want to find the updates  $\{\mathbf{z}_n^{(t+1)}, \mathbf{w}_m^{(t+1)}\}$  at the  $(t+1)$ -th iteration base on  $\{\mathbf{Z}^{(t)}, \mathbf{W}^{(t)}\}$  from the  $t$ -th iteration:

$$\mathbf{z}_n^{(t+1)} \leftarrow \arg \min_{\mathbf{z}_n^{(t)}} L(\mathbf{z}_n^{(t)}) \quad \text{and} \quad \mathbf{w}_m^{(t+1)} \leftarrow \arg \min_{\mathbf{w}_m^{(t)}} L(\mathbf{w}_m^{(t)}).$$

For simplicity, we will only derive for  $\mathbf{z}_n^{(t+1)} \leftarrow \arg \min_{\mathbf{z}_n^{(t)}} L(\mathbf{z}_n^{(t)})$ , and the derivation for the update on  $\mathbf{w}_m^{(t+1)}$  follows a similar approach.

**Approximation by linear update.** Suppose we want to approximate  $\mathbf{z}_n^{(t+1)}$  using a *linear update* based on  $\mathbf{z}_n^{(t)}$ :

$$(\text{Linear Update}) : \quad \mathbf{z}_n^{(t+1)} = \mathbf{z}_n^{(t)} + \eta \mathbf{v}.$$

The problem now becomes finding the solution of  $\mathbf{v}$  such that

$$\mathbf{v} = \arg \min_{\mathbf{v}} L(\mathbf{z}_n^{(t)} + \eta \mathbf{v}).$$

By Taylor's formula,  $L(\mathbf{z}_n^{(t)} + \eta \mathbf{v})$  can be approximated by

$$L(\mathbf{z}_n^{(t)} + \eta \mathbf{v}) \approx L(\mathbf{z}_n^{(t)}) + \eta \mathbf{v}^\top \nabla L(\mathbf{z}_n^{(t)}),$$

where  $\eta$  is a small value, and  $\nabla L(\mathbf{z}_n^{(t)})$  represents the gradient of  $L(\mathbf{z})$  evaluated at  $\mathbf{z}_n^{(t)}$ . To find  $\mathbf{v}$  under the constraint  $\|\mathbf{v}\|_2 = 1$  for a positive  $\eta$ , we perform the following minimization:

$$\mathbf{v} = \arg \min_{\|\mathbf{v}\|_2=1} L(\mathbf{z}_n^{(t)} + \eta \mathbf{v}) \approx \arg \min_{\|\mathbf{v}\|_2=1} \left\{ L(\mathbf{z}_n^{(t)}) + \eta \mathbf{v}^\top \nabla L(\mathbf{z}_n^{(t)}) \right\}.$$

This strategy is known as *greedy search*. The optimal  $\mathbf{v}$  can be obtained by

$$\mathbf{v} = -\nabla L(\mathbf{z}_n^{(t)}) / \|\nabla L(\mathbf{z}_n^{(t)})\|_2,$$

which means that  $\mathbf{v}$  points in the opposite direction to the gradient  $\nabla L(\mathbf{z}_n^{(t)})$ . Therefore, it is reasonable to update  $\mathbf{z}_n^{(t+1)}$  as follows:

$$\mathbf{z}_n^{(t+1)} = \mathbf{z}_n^{(t)} + \eta \mathbf{v} = \mathbf{z}_n^{(t)} - \eta \nabla L(\mathbf{z}_n^{(t)}) / \|\nabla L(\mathbf{z}_n^{(t)})\|_2,$$

which is commonly referred to as *gradient descent* (GD). Similarly, the gradient descent update for  $\mathbf{w}_m^{(t+1)}$  is given by

$$\mathbf{w}_m^{(t+1)} = \mathbf{w}_m^{(t)} + \eta \mathbf{v} = \mathbf{w}_m^{(t)} - \eta \nabla L(\mathbf{w}_m^{(t)}) / \|\nabla L(\mathbf{w}_m^{(t)})\|_2.$$

The revised procedure for Algorithm 13 employing a gradient descent approach is presented in Algorithm 15.

It's noteworthy that the ALS without GD (Algorithm 13) does not involve explicit parameters such as step size  $\eta$ . This characteristic can be both advantageous and disadvantageous. On one hand, it absolves the user from the time-consuming task of fine-tuning parameters, making the method more accessible and less demanding. On the other hand, this absence of adjustable parameters also restricts the user's control to directly influence the progression of the algorithm, leaving the convergence of ALS entirely contingent upon the inherent structure of the optimization problem at hand.

In practical applications, it is customary to alternate between the pure ALS iterations outlined in Algorithm 13 and the modified, gradient-descent variants discussed in this section. These descent-based adaptations offer the user a degree of control through a tunable step length parameter, allowing for a more customized approach to the optimization process.

## Geometric Interpretation of Gradient Descent

**Algorithm 15** Alternating Least Squares with Full Entries and Gradient Descent

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 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  $\lambda_w, \lambda_z$ , and step sizes  $\eta_w, \eta_z$ ;
- 4: Choose the maximal number of iterations  $C$ ;
- 5:  $iter = 0$ ; ▷ Count for the number of iterations
- 6: **while**  $\|\mathbf{A} - (\mathbf{W}\mathbf{Z})\|_F^2 > \delta$  and  $iter < C$  **do**
- 7:    $iter = iter + 1$ ;
- 8:   **for**  $n = 1, 2, \dots, N$  **do**
- 9:      $\mathbf{z}_n^{(t+1)} \leftarrow \mathbf{z}_n^{(t)} - \eta_z \nabla L(\mathbf{z}_n^{(t)}) / \|\nabla L(\mathbf{z}_n^{(t)})\|_2$ ; ▷  $n$ -th column of  $\mathbf{Z}$
- 10:   **end for**
- 11:   **for**  $m = 1, 2, \dots, M$  **do**
- 12:      $\mathbf{w}_m^{(t+1)} \leftarrow \mathbf{w}_m^{(t)} - \eta_w \nabla L(\mathbf{w}_m^{(t)}) / \|\nabla L(\mathbf{w}_m^{(t)})\|_2$ ; ▷  $m$ -th column of  $\mathbf{W}^\top$
- 13:   **end for**
- 14: **end while**
- 15: Output  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M], \mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ .

---

**Lemma 12.10: (Direction of gradients)** The gradient of a function at a given point is perpendicular to the level curve (or level surface in higher dimensions) passing through that point.

**Proof** [of Lemma 12.10, the informal proof] This proof involves showing that the gradient is orthogonal to the tangent vector of the level curve. For simplicity, let's start with the two-dimensional case. Suppose the level curve takes the form  $f(x, y) = c$ . This implicitly establishes a relationship between  $x$  and  $y$  such that  $y = y(x)$ , where  $y$  can be regarded as a function of  $x$ <sup>13</sup>. Therefore, the level curve can be expressed as  $f(x, y(x)) = c$ . Applying the chain rule, we get:

$$\underbrace{\frac{\partial f}{\partial x} \frac{dx}{dx}}_{=1} + \frac{\partial f}{\partial y} \frac{dy}{dx} = 0 \quad \Rightarrow \quad \left\langle \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right\rangle \cdot \left\langle \frac{dx}{dx}, \frac{dy}{dx} \right\rangle = 0.$$

That is, the gradient is perpendicular to the tangent.

In full generality, consider 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 parameter  $t$  on the level curve  $f(\mathbf{x}) = c$ :  $f(x_1(t), x_2(t), \dots, x_n(t)) = c$ . Differentiating the equation with respect to  $t$  using the chain rule:

$$\frac{\partial f}{\partial x_1} \frac{dx_1}{dt} + \frac{\partial f}{\partial x_2} \frac{dx_2}{dt} + \dots + \frac{\partial f}{\partial x_n} \frac{dx_n}{dt} = 0.$$

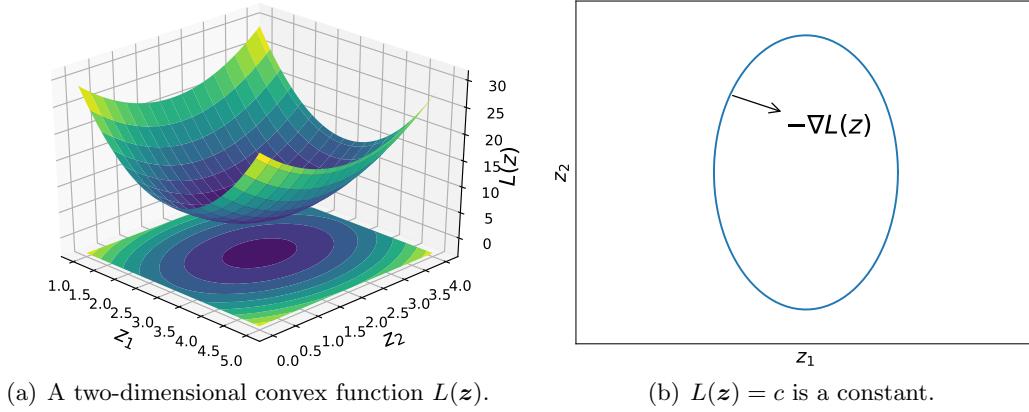
Thus, the gradient is perpendicular to the tangent in the  $n$ -dimensional case:

$$\left\langle \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right\rangle \cdot \left\langle \frac{dx_1}{dt}, \frac{dx_2}{dt}, \dots, \frac{dx_n}{dt} \right\rangle = 0.$$

This completes the proof. ■

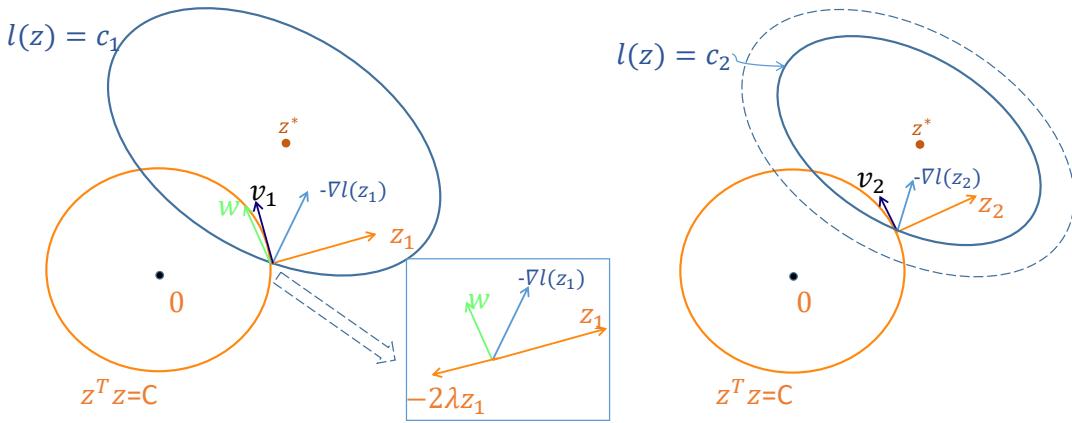
<sup>13</sup>. This is known as the implicit function theorem, provided that the partial derivative is nonzero and the function is smooth.

This lemma provides a key geometric insight into gradient descent. When minimizing a convex function  $L(\mathbf{z})$ , gradient descent moves in the direction opposite to the gradient, which corresponds to the steepest descent direction. This direction ensures a decrease in the value of the loss function. Figure 12.2 illustrates this concept in two dimensions, where the vector  $-\nabla L(\mathbf{z})$  points in the direction of maximum decrease of the convex function  $L(\mathbf{z})$ .



**Figure 12.2:** Figure 12.2(a) shows surface and contour plots for a specific function (blue=low, yellow=high), where the upper graph is the surface plot, and the lower one is its projection (i.e., contour). Figure 12.2(b):  $-\nabla L(\mathbf{z})$  pushes the loss to decrease for the convex function  $L(\mathbf{z})$ .

## 12.7. Regularization: A Geometric Interpretation



**Figure 12.3:** Constrained gradient descent with  $\mathbf{z}^T \mathbf{z} \leq C$ . The green vector  $\mathbf{w}$  represents the projection of  $\mathbf{v}_1$  onto the set  $\mathbf{z}^T \mathbf{z} \leq C$ , where  $\mathbf{v}_1$  is the component of  $-\nabla l(\mathbf{z})$  that is perpendicular to  $\mathbf{z}_1$ . The image on the right illustrates the next step after the update in the left image.  $\mathbf{z}^*$  denotes the optimal solution of  $\{\min l(\mathbf{z})\}$ .

In Section 12.3, we discussed how regularization can extend the ALS algorithm to general matrices. Gradient descent offers a geometric interpretation of regularization. To avoid

confusion, we denote the loss function without regularization as  $l(\mathbf{z})$  and the loss function with regularization as  $L(\mathbf{z}) = l(\mathbf{z}) + \lambda_z \|\mathbf{z}\|_2^2$ , where  $l(\mathbf{z}) : \mathbb{R}^n \rightarrow \mathbb{R}$ . When minimizing  $l(\mathbf{z})$ , a descent method typically searches for a solution in  $\mathbb{R}^n$ . However, in machine learning, searching across the entire space  $\mathbb{R}^n$  can lead to overfitting. One way to mitigate this is to restrict the search to a subset of the vector space, such as searching in  $\mathbf{z}^\top \mathbf{z} < C$  for some constant  $C$ . This can be formulated as the constrained optimization problem:

$$\arg \min_{\mathbf{z}} l(\mathbf{z}), \quad \text{s.t.,} \quad \mathbf{z}^\top \mathbf{z} \leq C.$$

As demonstrated above, a standard gradient descent method updates  $\mathbf{z}$  by moving in the direction of steepest descent, i.e., update  $\mathbf{z}$  as  $\mathbf{z} \leftarrow \mathbf{z} - \eta \nabla l(\mathbf{z})$  for a small step size  $\eta$ . When the level curve is  $l(\mathbf{z}) = c_1$  and the current position of parameter  $\mathbf{z}$  is  $\mathbf{z} = \mathbf{z}_1$ , where  $\mathbf{z}_1$  lies at 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 image of Figure 12.3 (by Lemma 12.10). However, if we further restrict that the optimal value must lie within  $\mathbf{z}^\top \mathbf{z} \leq C$ , the standard descent direction  $-\nabla l(\mathbf{z}_1)$  will lead the update  $\mathbf{z}_2 = \mathbf{z}_1 - \eta \nabla l(\mathbf{z}_1)$  beyond the boundary of  $\mathbf{z}^\top \mathbf{z} \leq C$ . One 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  $a\mathbf{z}_1$  represents 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  $\mathbf{z}^\top \mathbf{z} = C$ . By keeping only the step  $\mathbf{v}_1$ , the update becomes

$$\mathbf{z}_2 = \text{project}(\mathbf{z}_1 + \eta \mathbf{v}_1) = \text{project}\left(\mathbf{z}_1 + \eta \underbrace{(-\nabla l(\mathbf{z}_1) - a\mathbf{z}_1)}_{\mathbf{v}_1}\right), \quad \text{14}$$

which will lead to a smaller loss from  $l(\mathbf{z}_1)$  to  $l(\mathbf{z}_2)$  while maintaining the constraint  $\mathbf{z}^\top \mathbf{z} \leq C$ . This approach is known as *projection gradient descent (PGD)*. It is not hard to see that the update  $\mathbf{z}_2 = \text{project}(\mathbf{z}_1 + \eta \mathbf{v}_1)$  can be understood as finding a vector  $\mathbf{w}$  (represented by the green vector in the left image of Figure 12.3) such that  $\mathbf{z}_2 = \mathbf{z}_1 + \mathbf{w}$  lies within the constraint set  $\mathbf{z}^\top \mathbf{z} \leq C$ . Mathematically, the vector  $\mathbf{w}$  can be determined as  $-\nabla l(\mathbf{z}_1) - 2\lambda \mathbf{z}_1$  for some  $\lambda$ , as illustrated in the middle image of Figure 12.3. This corresponds precisely to the negative gradient of the regularized loss function  $L(\mathbf{z}) = l(\mathbf{z}) + \lambda \|\mathbf{z}\|_2^2$ , so that

$$\mathbf{w} = -\nabla L(\mathbf{z}) = -\nabla l(\mathbf{z}) - 2\lambda \mathbf{z} \implies \mathbf{z}_2 = \mathbf{z}_1 + \mathbf{w} = \mathbf{z}_1 - \nabla L(\mathbf{z}).$$

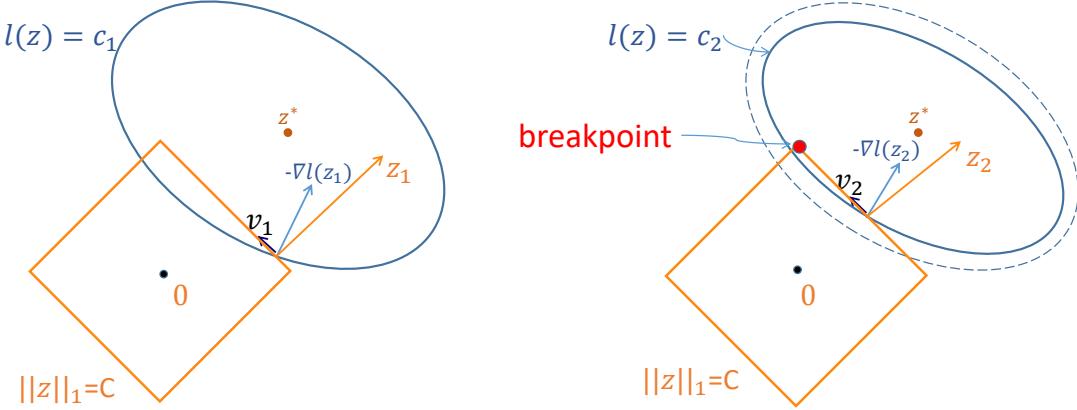
And in practice, using a small step size  $\eta$  prevents the trajectory from moving outside the constraint set  $\mathbf{z}^\top \mathbf{z} \leq C$ :

$$\mathbf{z}_2 = \mathbf{z}_1 - \eta \nabla L(\mathbf{z}),$$

which aligns with the regularization term discussed in Section 12.3.

**Sparsity.** In certain scenarios, we seek to identify a sparse solution  $\mathbf{z}$  such that  $l(\mathbf{z})$  is minimized. For example, in facial feature extraction, sparsity leads to more localized features, meaning that fewer features are used to reconstruct each input image. Regularization to be constrained in  $\|\mathbf{z}\|_1 \leq C$  exists to this purpose, where  $\|\cdot\|_1$  denotes the  $\ell_1$  norm of a vector or a matrix. Similar to the previous case, the  $\ell_1$  constrained optimization pushes the gradient descent towards the border of the level set  $\|\mathbf{z}\|_1 = C$ . The situation in the two-dimensional case is shown in Figure 12.4. In high-dimensional cases, many elements in  $\mathbf{z}$  will be driven towards the breakpoint of  $\|\mathbf{z}\|_1 = C$ , promoting sparsity in the solution, as shown in the right image of Figure 12.4.

14. where the operation  $\text{project}(\mathbf{x})$  will project the vector  $\mathbf{x}$  to the closest point inside  $\mathbf{z}^\top \mathbf{z} \leq C$ . Notice here the unprojected update  $\mathbf{z}_2 = \mathbf{z}_1 + \eta \mathbf{v}_1$  can still make  $\mathbf{z}_2$  fall outside the curve of  $\mathbf{z}^\top \mathbf{z} \leq C$ .



**Figure 12.4:** Constrained gradient descent with  $\|z\|_1 \leq C$ , where the red dot denotes the breakpoint in the  $\ell_1$  norm. The right image illustrates the next step after the update in the left image.  $z^*$  denotes the optimal solution of  $\{\min l(z)\}$ .

## 12.8. Stochastic Gradient Descent

The gradient descent method is a valuable optimization algorithm; however, it exhibits certain limitations in practical applications. To understand these limitations, consider the mean squared error (MSE) derived from Equation (12.10):

$$\frac{1}{MN} \min_{\mathbf{W}, \mathbf{Z}} \sum_{n=1}^N \sum_{m=1}^M (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2. \quad (12.27)$$

The MSE requires calculating the residual  $e_{mn} = (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2$  for each observed entry  $a_{mn}$ , representing the squared difference between the predicted and actual values. The total sum of squared residuals is denoted by  $e = \sum_{m,n=1}^{MN} e_{mn}$ . When the number of training entries is large (i.e.,  $MN$  is large), computing the full gradient over all entries becomes computationally expensive and slow. Moreover, gradients from different samples may cancel each other out, leading to small net updates and slow convergence. To address these issues, researchers have enhanced the gradient descent method with the *stochastic gradient descent (SGD)* method (see, for example, Lu (2022c)). In the SGD algorithm, instead of calculating the full gradient of the objective function with respect to the parameters across all samples in the data set, which can be computationally expensive, the algorithm takes a more efficient approach. It randomly chooses one sample and calculates the gradient of the objective function with respect to the parameters using only this single sample. This gradient estimate is then used to update the parameters in the direction that minimizes the objective function. By using a single sample at each iteration, the SGD algorithm provides a fast and often sufficient approximation of the full gradient, making it particularly well-suited for large-scale data sets.

In particular, we consider again the per-example loss:

$$L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \sum_{n=1}^N \sum_{m=1}^M (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2 + \frac{1}{2} \lambda_w \sum_{m=1}^M \|\mathbf{w}_m\|_2^2 + \frac{1}{2} \lambda_z \sum_{n=1}^N \|\mathbf{z}_n\|_2^2.$$

As we iteratively minimize the loss term  $l(\mathbf{w}_m, \mathbf{z}_n) = \frac{1}{2} (a_{mn} - \mathbf{w}_m^\top \mathbf{z}_n)^2 + \frac{1}{2} \lambda_w \|\mathbf{w}_m\|_2^2 + \frac{1}{2} \lambda_z \|\mathbf{z}_n\|_2^2$  for all  $m \in \{1, 2, \dots, M\}, n \in \{1, 2, \dots, N\}$  (referred to as the per-example loss

term), the overall loss  $L(\mathbf{W}, \mathbf{Z})$  decreases accordingly. This approach is also known as *stochastic coordinate descent*. The gradients with respect to  $\mathbf{w}_m$  and  $\mathbf{z}_n$ , and their roots are given, respectively, by

$$\begin{cases} \nabla_{\mathbf{z}_n} l(\mathbf{z}_n) = \mathbf{w}_m \mathbf{w}_m^\top \mathbf{z}_n + \lambda_z \mathbf{z}_n - a_{mn} \mathbf{w}_m & \Rightarrow \mathbf{z}_n = a_{mn} (\mathbf{w}_m \mathbf{w}_m^\top + \lambda_z \mathbf{I})^{-1} \mathbf{w}_m; \\ \nabla_{\mathbf{w}_m} l(\mathbf{w}_m) = \mathbf{z}_n \mathbf{z}_n^\top \mathbf{w}_m + \lambda_w \mathbf{w}_m - a_{mn} \mathbf{z}_n & \Rightarrow \mathbf{w}_m = a_{mn} (\mathbf{z}_n \mathbf{z}_n^\top + \lambda_w \mathbf{I})^{-1} \mathbf{z}_n. \end{cases}$$

Alternatively, the update can be performed using gradient descent for the per-example loss. Since we update based on the per-example loss, this approach is thus known as the *stochastic gradient descent (SGD)*:

$$\mathbf{z}_n \leftarrow \mathbf{z}_n - \eta_z \frac{\nabla_{\mathbf{z}_n} l(\mathbf{z}_n)}{\|\nabla_{\mathbf{z}_n} l(\mathbf{z}_n)\|_2} \quad \text{and} \quad \mathbf{w}_m \leftarrow \mathbf{w}_m - \eta_w \frac{\nabla_{\mathbf{w}_m} l(\mathbf{w}_m)}{\|\nabla_{\mathbf{w}_m} l(\mathbf{w}_m)\|_2}.$$

The stochastic gradient descent update for ALS is formulated in Algorithm 16. It is possible that the gradient descent or stochastic gradient descent algorithm may fail to converge. In such cases, it is advisable to re-run the algorithm using a smaller step size. And in practice, the indices  $m$  and  $n$  in the algorithm can be randomly generated, which is why the method is termed “stochastic.” <sup>15</sup>

---

**Algorithm 16** Alternating Least Squares with Full Entries and SGD

---

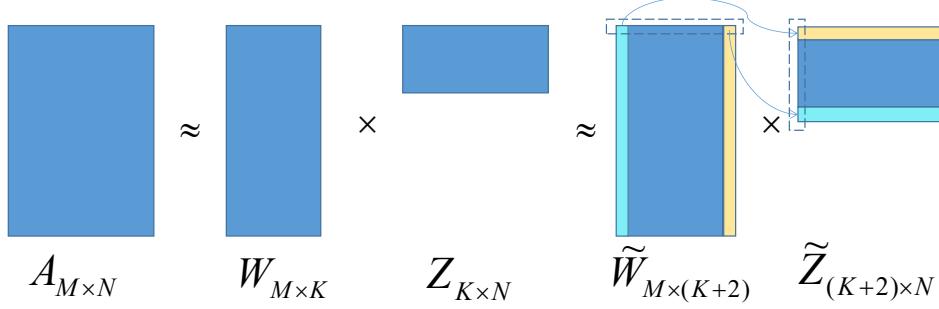
**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 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  $\lambda_w, \lambda_z$ , and step sizes  $\eta_w, \eta_z$ ;
  - 4: Choose the maximal number of iterations  $C$ ;
  - 5:  $iter = 0$ ; ▷ Count for the number of iterations
  - 6: **while**  $\|\mathbf{A} - (\mathbf{W}\mathbf{Z})\|_F^2 > \delta$  and  $iter < C$  **do**
  - 7:    $iter = iter + 1$ ;
  - 8:   **for**  $n = 1, 2, \dots, N$  **do**
  - 9:     **for**  $m = 1, 2, \dots, M$  **do** ▷ in practice,  $m, n$  can be randomly produced
  - 10:        $\mathbf{z}_n \leftarrow \mathbf{z}_n - \eta_z \nabla l(\mathbf{z}_n) / \|\nabla l(\mathbf{z}_n)\|_2$ ; ▷  $n$ -th column of  $\mathbf{Z}$
  - 11:        $\mathbf{w}_m \leftarrow \mathbf{w}_m - \eta_w \nabla l(\mathbf{w}_m) / \|\nabla l(\mathbf{w}_m)\|_2$ ; ▷  $m$ -th column of  $\mathbf{W}^\top$
  - 12:     **end for**
  - 13:   **end for**
  - 14: **end while**
  - 15: Output  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M], \mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N]$ .
- 

## 12.9. Bias Term

In ordinary least squares models, a bias term is usually incorporated into the raw matrix to improve model performance, as illustrated in Equation (12.2). A similar approach can be applied to the ALS algorithm. Specifically, a fixed column filled with all ones can be appended to the **last column** of matrix  $\mathbf{W}$ . To account for this, an extra row should be added to the last row of matrix  $\mathbf{Z}$  to fit the features introduced by the bias term in  $\mathbf{W}$ .

<sup>15</sup>. When we iteratively choose the values of  $m$  and  $n$  from  $\{1, 2, \dots, M\}$  and  $\{1, 2, \dots, N\}$  in a deterministic cyclic order, respectively, the stochastic method can be referred to as “*incremental gradient descent*.”



**Figure 12.5:** Bias terms in alternating least squares, where the yellow entries denote ones (which are fixed), and the cyan entries denote the added features to fit the bias terms. The dotted boxes provide an example of how the bias terms work.

Analogously, a fixed row with all ones can be added to the **first row** of  $\mathbf{Z}$ , and an extra column in the first column of  $\mathbf{W}$  can be added to fit the features. This configuration is illustrated in Figure 12.5.

Given the loss function with respect to the columns of  $\mathbf{Z}$  in Equation (12.25), let  $\tilde{\mathbf{z}}_n = \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix} \in \mathbb{R}^{K+2}$  be the  $n$ -th column of  $\tilde{\mathbf{Z}}$ . Then we have:

$$\begin{aligned} 2L(\mathbf{z}_n) &= \left\| \tilde{\mathbf{W}} \tilde{\mathbf{Z}} - \mathbf{A} \right\|_F^2 + \lambda_w \left\| \tilde{\mathbf{W}} \right\|_F^2 + \lambda_z \left\| \tilde{\mathbf{Z}} \right\|_F^2 = \left\| \tilde{\mathbf{W}} \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix} - \mathbf{a}_n \right\|_2^2 + \underbrace{\lambda_z \|\tilde{\mathbf{z}}_n\|_2^2}_{=\lambda_z \|\mathbf{z}_n\|_2^2 + \lambda_z} + C_{z_n} \\ &= \left\| [\bar{\mathbf{w}}_0 \quad \bar{\mathbf{W}}] \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix} - \mathbf{a}_n \right\|_2^2 + \lambda_z \|\mathbf{z}_n\|_2^2 + C_{z_n} = \left\| \bar{\mathbf{W}} \mathbf{z}_n - (\underbrace{\mathbf{a}_n - \bar{\mathbf{w}}_0}_{\bar{\mathbf{a}}_n}) \right\|_2^2 + \lambda_z \|\mathbf{z}_n\|_2^2 + C_{z_n}, \end{aligned} \quad (12.28)$$

where  $\bar{\mathbf{w}}_0$  represents the first column of  $\tilde{\mathbf{W}}$ ,  $\bar{\mathbf{W}}$  denotes the remaining  $K+1$  columns of  $\tilde{\mathbf{W}}$  (i.e.,  $\tilde{\mathbf{W}} = [\bar{\mathbf{w}}_0, \bar{\mathbf{W}}]$ ), 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 for  $\mathbf{z}_n$  is just similar to the one in Equation (12.25), with the gradient given by

$$\nabla_{\mathbf{z}_n} L(\mathbf{z}_n) = \bar{\mathbf{W}}^\top \bar{\mathbf{W}} \mathbf{z}_n - \bar{\mathbf{W}}^\top \bar{\mathbf{a}}_n + \lambda_z \mathbf{z}_n.$$

Therefore, the update for  $\mathbf{z}_n$  is given by determining the root of the gradient above:

$$(\text{update for } \tilde{\mathbf{z}}_n) : \quad \mathbf{z}_n = (\bar{\mathbf{W}}^\top \bar{\mathbf{W}} + \lambda_z \mathbf{I})^{-1} \bar{\mathbf{W}}^\top \bar{\mathbf{a}}_n \quad \Rightarrow \quad \tilde{\mathbf{z}}_n = \begin{bmatrix} 1 \\ \mathbf{z}_n \end{bmatrix}, \forall n.$$

Similarly, following the loss with respect to each row of  $\mathbf{W}$  in Equation (12.26), let  $\tilde{\mathbf{w}}_m = \begin{bmatrix} \mathbf{w}_m \\ 1 \end{bmatrix} \in \mathbb{R}^{K+2}$  be the  $m$ -th row of  $\tilde{\mathbf{W}}$  (or  $m$ -th column of  $\tilde{\mathbf{W}}^\top$ ). Then we have:

$$\begin{aligned} 2L(\mathbf{w}_m) &= \left\| \tilde{\mathbf{Z}}^\top \tilde{\mathbf{W}}^\top - \mathbf{A}^\top \right\|_F^2 + \lambda_w \left\| \tilde{\mathbf{W}}^\top \right\|_F^2 + \lambda_z \left\| \tilde{\mathbf{Z}} \right\|_F^2 = \left\| \tilde{\mathbf{Z}}^\top \tilde{\mathbf{w}}_m - \mathbf{b}_m \right\|_2^2 + \underbrace{\lambda_w \|\tilde{\mathbf{w}}_m\|_2^2}_{=\lambda_w \|\mathbf{w}_m\|_2^2 + \lambda_w} + C_{w_m} \\ &= \left\| [\bar{\mathbf{Z}}^\top \quad \bar{\mathbf{z}}_0] \begin{bmatrix} \mathbf{w}_m \\ 1 \end{bmatrix} - \mathbf{b}_m \right\|_2^2 + \lambda_w \|\mathbf{w}_m\|_2^2 + C_{w_m} = \left\| \bar{\mathbf{Z}}^\top \mathbf{w}_m - (\mathbf{b}_m - \bar{\mathbf{z}}_0) \right\|_2^2 + \lambda_w \|\mathbf{w}_m\|_2^2 + C_{w_m}, \end{aligned} \quad (12.29)$$

where  $\bar{\mathbf{z}}_0$  represents the last column of  $\tilde{\mathbf{Z}}^\top$ ,  $\bar{\mathbf{Z}}^\top$  contains the remaining  $K+1$  columns of  $\tilde{\mathbf{Z}}^\top$  (i.e.,  $\tilde{\mathbf{Z}}^\top = [\bar{\mathbf{Z}}^\top, \bar{\mathbf{z}}_0]$ ), and  $C_{w_m}$  is a constant with respect to  $\mathbf{w}_m$ .  $\mathbf{W}^\top = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_M]$  and  $\mathbf{A}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M]$  are the column partitions of  $\mathbf{W}^\top$  and  $\mathbf{A}^\top$ , respectively. Let

$\bar{\mathbf{b}}_m = \mathbf{b}_m - \bar{\mathbf{z}}_0$ . The update for  $\mathbf{w}_m$  is again just similar to the one in Equation (12.26), with the gradient given by

$$\nabla_{\mathbf{w}_m} L(\mathbf{w}_m) = \bar{\mathbf{Z}} \cdot \bar{\mathbf{Z}}^\top \mathbf{w}_m - \bar{\mathbf{Z}} \cdot \bar{\mathbf{b}}_m + \lambda_w \mathbf{w}_m.$$

Therefore, the update for  $\mathbf{w}_m$  is given by the root of the gradient above:

$$(\text{update for } \tilde{\mathbf{w}}_m) : \quad \mathbf{w}_m = (\bar{\mathbf{Z}} \cdot \bar{\mathbf{Z}}^\top + \lambda_w \mathbf{I})^{-1} \bar{\mathbf{Z}} \cdot \bar{\mathbf{b}}_m \quad \Rightarrow \quad \tilde{\mathbf{w}}_m = \begin{bmatrix} \mathbf{w}_m \\ 1 \end{bmatrix}, \forall m.$$

Similar updates can be derived using gradient descent, taking into account the bias terms and handling missing entries (see Section 12.6 and 12.4 for a reference).

## 12.10. Low-Rank Hadamard Decomposition

In the fields of linear algebra and data analysis, matrix decomposition techniques are essential for extracting meaningful information from complex datasets. As discussed above, one common objective is to approximate a given matrix using a lower-rank representation, which simplifies the data while preserving its key characteristics. The Hadamard product, also known as the element-wise product, provides an alternative to traditional matrix multiplication in matrix decomposition.

As discussed previously, the alternating least squares (ALS) algorithm is an iterative method used to find a suboptimal low-rank approximation of a matrix by decomposing it into two or more matrices. ALS is particularly advantageous for large-scale problems, such as those found in recommender systems, where the goal is to predict missing entries in a user-item interaction matrix. During each iteration, the ALS algorithm alternates between updating one matrix while keeping the other fixed, thereby minimizing the reconstruction error at every step. Nonnegative matrix factorization (NMF), introduced in Chapter 13, is a variant of matrix factorization where both the original matrix and the resulting factorized matrices have nonnegative entries. This constraint makes NMF especially suitable for applications where the data represents quantities that cannot be negative, such as images, audio signals, or document-term matrices in text mining.

Ws further explore the Hadamard decomposition of a matrix  $\mathbf{A}$ , where  $\mathbf{A}$  can be expressed as the Hadamard product of two low-rank matrices:  $\mathbf{A} = \mathbf{A}_1 \circledast \mathbf{A}_2$ . This type of decomposition is advantageous when the data exhibits multiplicative relationships, and a low-rank approximation is desired to reduce complexity or enhance interpretability.

**Non-Factorizability Issue.** When  $\mathbf{A}_1 \in \mathbb{R}^{n^2 \times n^2}$  and  $\mathbf{A}_2 \in \mathbb{R}^{n^2 \times n^2}$  share the same rank  $n$ , the Hadamard product  $\mathbf{A}_1 \circledast \mathbf{A}_2$  can achieve a maximum rank of  $n^2$  (Problem 12.23). However, not all matrices  $\mathbf{A} \in \mathbb{R}^{n^2 \times n^2}$  of rank  $n^2$  can be represented as the Hadamard product of two lower-rank matrices:

- The Hadamard decomposition  $\mathbf{A} = \mathbf{A}_1 \circledast \mathbf{A}_2$ , where  $\mathbf{A}_1$  and  $\mathbf{A}_2$  are rank- $n$  factors, encodes a system of nonlinear equations.
- This system comprises  $n^2 \times n^2 = n^4$  equations (one per entry of  $\mathbf{A}$ ) and, due to the low-rank constraint on the two Hadamard factors  $\mathbf{A}_1$  and  $\mathbf{A}_2$ , only  $(n^2n + nn^2) = 2n^3$  variables exist.
- For  $n > 2$ , there are more equations than variables, suggesting that all the equations will be simultaneously satisfied only in special cases. For example, if the matrix  $\mathbf{A}$  includes a row or a column with all but a single entry being zero, then not all the equations in the system can be satisfied (Ciaperoni et al., 2024).

Therefore, we focus on solving the low-rank reconstruction problem for the Hadamard decomposition. Assuming that  $\mathbf{A}_1$  and  $\mathbf{A}_2$  share the same rank  $K$ , our aim is to reconstruct the design matrix  $\mathbf{A}$  through the Hadamard product  $\mathbf{A}_1 \circledast \mathbf{A}_2$ . Building upon the matrix factorization method used in alternating least squares (Section 12.2), we now concentrate on algorithms for solving the *low-rank Hadamard decomposition* problem:

- Given a real matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ , find matrix factors  $\mathbf{A}_1 \in \mathbb{R}^{M \times N}$  and  $\mathbf{A}_2 \in \mathbb{R}^{M \times N}$  such that:

$$\min L(\mathbf{C}_1, \mathbf{D}_1, \mathbf{C}_2, \mathbf{D}_2) = \|\mathbf{A}_1 \circledast \mathbf{A}_2 - \mathbf{A}\|_F^2 = \|(\mathbf{C}_1 \mathbf{D}_1) \circledast (\mathbf{C}_2 \mathbf{D}_2) - \mathbf{A}\|_F^2, \quad (12.30)$$

where  $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{M \times K}$ , and  $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{K \times N}$ :  $\mathbf{A}_1 = \mathbf{C}_1 \mathbf{D}_1$  and  $\mathbf{A}_2 = \mathbf{C}_2 \mathbf{D}_2$  such that  $\mathbf{A}_1$  and  $\mathbf{A}_2$  are rank- $K$  matrices.

Low-rank (Hadamard) decomposition is often necessary because many natural phenomena exhibit multiplicative or conjunctive relationships (Ciaperoni et al., 2024). For instance, consider a study on risk factors for a disease with two predictors: smoking status (yes/no) and alcohol consumption (yes/no). The multiplicative model would account not only for the individual effects of smoking and alcohol consumption but also for their interaction. The (low-rank) Hadamard decomposition offers an alternative approach to modeling such relationships.

Following the alternating descent framework using gradient descent, at each iteration, the matrices  $\mathbf{C}_1, \mathbf{D}_1, \mathbf{C}_2$ , and  $\mathbf{D}_2$  are updated sequentially by taking a step in the direction opposite to the gradient of the objective function. It then can be shown that

$$\nabla L(\mathbf{C}_1) = \nabla L(\mathbf{C}_1 | \mathbf{D}_1, \mathbf{C}_2, \mathbf{D}_2) = 2((\mathbf{C}_1 \mathbf{D}_1) \circledast (\mathbf{C}_2 \mathbf{D}_2) - \mathbf{A}) \circledast (\mathbf{C}_2 \mathbf{D}_2) \mathbf{D}_1^\top.$$

**Proof** For simplicity, we derive the gradient of  $\mathbf{E}$  for  $f(\mathbf{E}) = \|\mathbf{E}\mathbf{F} \circledast \mathbf{C} - \mathbf{D}\|_F^2$ . We have

$$\begin{aligned} f(\mathbf{E}) &= \|\mathbf{E}\mathbf{F} \circledast \mathbf{C} - \mathbf{D}\|_F^2 = \text{tr} \left( (\mathbf{E}\mathbf{F} \circledast \mathbf{C} - \mathbf{D})^\top (\mathbf{E}\mathbf{F} \circledast \mathbf{C} - \mathbf{D}) \right) \\ &= \text{tr} \left( (\mathbf{E}\mathbf{F} \circledast \mathbf{C})^\top (\mathbf{E}\mathbf{F} \circledast \mathbf{C}) \right) - 2\text{tr} \left( (\mathbf{E}\mathbf{F} \circledast \mathbf{C})^\top \mathbf{D} \right) + \text{tr}(\mathbf{D}^\top \mathbf{D}). \end{aligned}$$

Considering the first term, we get

$$\frac{\partial \text{tr} ((\mathbf{E}\mathbf{F} \circledast \mathbf{C})^\top (\mathbf{E}\mathbf{F} \circledast \mathbf{C}))}{\partial \mathbf{E}} = 2(\mathbf{E}\mathbf{F}) \circledast \mathbf{C} \circledast \mathbf{C} \cdot \mathbf{F}^\top. \quad 16$$

For the second term, it follows that

$$-2 \frac{\partial \text{tr} ((\mathbf{E}\mathbf{F} \circledast \mathbf{C})^\top \mathbf{D})}{\partial \mathbf{E}} = -2\mathbf{D} \circledast \mathbf{C} \cdot \frac{\partial \mathbf{E}\mathbf{F}}{\partial \mathbf{E}} = -2\mathbf{D} \circledast \mathbf{C} \cdot \mathbf{F}^\top. \quad 17$$

The third term is a constant w.r.t. to  $\mathbf{E}$ . Therefore,  $\frac{\partial f(\mathbf{E})}{\partial \mathbf{E}} = 2(\mathbf{E}\mathbf{F}) \circledast \mathbf{C} \circledast \mathbf{C} \cdot \mathbf{F}^\top - 2\mathbf{D} \circledast \mathbf{C} \cdot \mathbf{F}^\top = 2((\mathbf{E}\mathbf{F}) \circledast \mathbf{C} - \mathbf{D}) \circledast \mathbf{C} \cdot \mathbf{F}^\top$ . Substituting  $\mathbf{E} = \mathbf{C}_1$ ,  $\mathbf{F} = \mathbf{D}_1$ ,  $\mathbf{C} = \mathbf{C}_2 \mathbf{D}_2$ , and  $\mathbf{D} = \mathbf{A}$  completes the proof. ■

The gradients with respect to  $\mathbf{D}_1, \mathbf{C}_2$ , and  $\mathbf{D}_2$  can be derived analogously. Thus, the alternating descent method for obtaining the low-rank approximation of Hadamard decomposition can be described by Algorithm 17.

16. Use the fact that  $\frac{\partial \text{tr} ((\mathbf{E}\mathbf{F} \circledast \mathbf{C})^\top (\mathbf{E}\mathbf{F} \circledast \mathbf{C}))}{\partial \mathbf{E}} = 2\mathbf{E} \circledast \mathbf{C} \circledast \mathbf{C}$ , which can be derived element-wise.

17. Use the fact that  $\frac{\partial \text{tr} ((\mathbf{E}\mathbf{F} \circledast \mathbf{C})^\top \mathbf{D})}{\partial \mathbf{E}} = \mathbf{D} \circledast \mathbf{C}$ , which can be derived element-wise. Since  $\text{tr}((\mathbf{E} \circledast \mathbf{C})^\top \mathbf{D}) = \sum_{i,j} d_{ij} a_{ij} c_{ij}$  and thus  $\frac{\partial \text{tr} ((\mathbf{E}\mathbf{F} \circledast \mathbf{C})^\top \mathbf{D})}{\partial a_{ij}} = d_{ij} c_{ij}$ .

**Algorithm 17** Alternating Descent with Gradient Descent for Low-Rank Hadamard Decomposition: A regularization can also be added into the gradient descent update (see Section 12.3).

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 1: Initialize  $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{M \times K}$ , and  $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{K \times N}$ ;
- 2: Choose a stopping criterion on the approximation error  $\delta$ ;
- 3: Choose step size  $\eta$ ;
- 4: Choose the maximum number of iterations  $C$ ;
- 5:  $iter = 0$ ; ▷ Count for the number of iterations
- 6: **while**  $\|(\mathbf{C}_1 \mathbf{D}_1) \circledast (\mathbf{C}_2 \mathbf{D}_2) - \mathbf{A}\|_F^2 > \delta$  and  $iter < C$  **do**
- 7:    $iter = iter + 1$ ;
- 8:    $\Delta \leftarrow ((\mathbf{C}_1 \mathbf{D}_1) \circledast (\mathbf{C}_2 \mathbf{D}_2) - \mathbf{A})$ ;
- 9:    $\mathbf{C}_1 \leftarrow \mathbf{C}_1 - \eta \nabla L(\mathbf{C}_1) = \mathbf{C}_1 - \eta \cdot 2(\Delta \circledast (\mathbf{C}_2 \mathbf{D}_2)) \mathbf{D}_1^\top$ ;
- 10:    $\mathbf{D}_1 \leftarrow \mathbf{D}_1 - \eta \nabla L(\mathbf{D}_1) = \mathbf{D}_1 - \eta \cdot 2\{(\Delta^\top \circledast (\mathbf{C}_2 \mathbf{D}_2)^\top) \mathbf{C}_1\}^\top$ ;
- 11:    $\mathbf{C}_2 \leftarrow \mathbf{C}_2 - \eta \nabla L(\mathbf{C}_2) = \mathbf{C}_2 - \eta \cdot 2(\Delta \circledast (\mathbf{C}_1 \mathbf{D}_1)) \mathbf{D}_2^\top$ ;
- 12:    $\mathbf{D}_2 \leftarrow \mathbf{D}_2 - \eta \nabla L(\mathbf{D}_2) = \mathbf{D}_2 - \eta \cdot 2\{(\Delta^\top \circledast (\mathbf{C}_1 \mathbf{D}_1)^\top) \mathbf{C}_2\}^\top$ ;
- 13: **end while**
- 14: Output  $\mathbf{C}_1, \mathbf{D}_1, \mathbf{C}_2, \mathbf{D}_2$ .

---

### 12.10.1 Rank-One Update

Following the rank-one update approach used in ALS (Section 12.4), we consider updating the  $n$ -th column  $\mathbf{d}_{1,n}$  of  $\mathbf{D}_1$ ,  $n \in \{1, 2, \dots, N\}$ . Analogously, the gradient with respect to  $\mathbf{d}_{1,n}$  can be derived as:

$$\begin{aligned} \nabla L(\mathbf{d}_{1,n}) &= \frac{\partial L(\mathbf{d}_{1,n})}{\partial \mathbf{d}_{1,n}} = 2\mathbf{C}_1^\top ((\mathbf{C}_1 \mathbf{d}_{1,n}) \circledast \mathbf{a}_{2,n} \circledast \mathbf{a}_{2,n}) - 2\mathbf{C}_1^\top (\mathbf{a}_n \circledast \mathbf{a}_{2,n}) \\ &= 2\mathbf{C}_1^\top [((\mathbf{C}_1 \mathbf{d}_{1,n}) \circledast \mathbf{a}_{2,n} - \mathbf{a}_n) \circledast \mathbf{a}_{2,n}], \quad n \in \{1, 2, \dots, N\}, \end{aligned} \quad (12.31)$$

where  $\mathbf{a}_{2,n}$  denotes the  $n$ -th column of  $\mathbf{A}_2 = \mathbf{C}_2 \mathbf{D}_2$ . The gradients for the columns of  $\mathbf{D}_2$  can be computed in a similar manner.

Suppose further that  $\mathbf{C}_1^\top = [\mathbf{c}_{1,1}, \mathbf{c}_{1,2}, \dots, \mathbf{c}_{1,M}] \in \mathbb{R}^{K \times M}$ ,  $\mathbf{B} = \mathbf{A}^\top = [\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_M] \in \mathbb{R}^{N \times M}$ , and  $\mathbf{B}_2 = \mathbf{A}_2^\top = (\mathbf{C}_2 \mathbf{D}_2)^\top = [\mathbf{b}_{2,1}, \mathbf{b}_{2,2}, \dots, \mathbf{b}_{2,M}] \in \mathbb{R}^{N \times M}$ , i.e., the row partitions of  $\mathbf{C}_1$ ,  $\mathbf{A}$ , and  $\mathbf{A}_2 = (\mathbf{C}_2 \mathbf{D}_2)$ , respectively. Then, the gradient with respect to  $\mathbf{c}_{1,m}$  is given by:

$$\nabla L(\mathbf{c}_{1,m}) = \frac{\partial L(\mathbf{c}_{1,m})}{\partial \mathbf{c}_{1,m}} = 2\mathbf{D}_1 \left( [(\mathbf{D}_1^\top \mathbf{c}_{1,m}) \circledast \mathbf{b}_{2,m} - \mathbf{b}_m] \circledast \mathbf{b}_{2,m} \right), \quad m \in \{1, 2, \dots, M\}. \quad (12.32)$$

The gradient for the rows of  $\mathbf{C}_2$  can be obtained analogously. Therefore, Algorithm 17 can be adapted to update the columns of  $\mathbf{D}_1, \mathbf{D}_2$  and the rows of  $\mathbf{C}_1, \mathbf{C}_2$  iteratively (referred to as rank-one updates).

### 12.10.2 Missing Entries

The rank-one update framework can be extended to settings like the Netflix problem, in which case many entries of  $\mathbf{A} \in \mathbb{R}^{M \times N}$  are missing. Assuming  $\mathbf{A}$  is a low-rank matrix, we aim to fill in the missing entries of matrix  $\mathbf{A}$  (where  $M$  represents the number of movies, and  $N$  represents the number of users).

Let  $\mathbf{o}_n \in \{0, 1\}^M, n \in \{1, 2, \dots, N\}$ , represent the movies rated by user  $n$ , where  $o_{nm} = 1$  if user  $n$  has rated movie  $m$ , and  $o_{nm} = 0$  otherwise. Similarly, let  $\mathbf{p}_m \in \{0, 1\}^N, m \in \{1, 2, \dots, M\}$  denote the users who have rated movie  $m$ , with  $p_{mn} = 1$  if the movie  $m$  has been rated by user  $n$ , and  $p_{mn} = 0$  otherwise. Then, Equations (12.31) and (12.32) become

$$\nabla L(\mathbf{d}_{1,n}) = 2\mathbf{C}_1[\mathbf{o}_n, :]^\top ((\mathbf{C}_1[\mathbf{o}_n, :] \mathbf{d}_{1,n}) \circledast \mathbf{a}_{2,n}[\mathbf{o}_n] - \mathbf{a}_n[\mathbf{o}_n]) \circledast \mathbf{a}_{2,n}[\mathbf{o}_n], \\ n \in \{1, 2, \dots, N\}; \quad (12.33)$$

$$\nabla L(\mathbf{c}_{1,m}) = 2\mathbf{D}_1[:, \mathbf{p}_m] \left( (\mathbf{D}_1[:, \mathbf{p}_m]^\top \mathbf{c}_{1,m}) \circledast \mathbf{b}_{2,m}[\mathbf{p}_m] - \mathbf{b}_m[\mathbf{p}_m] \right) \circledast \mathbf{b}_{2,m}[\mathbf{p}_m], \\ m \in \{1, 2, \dots, M\}. \quad (12.34)$$

Since the Hadamard product commutes, the gradients for  $L(\mathbf{d}_{2,n}), n \in \{1, 2, \dots, N\}$  and  $L(\mathbf{c}_{2,m}), m \in \{1, 2, \dots, M\}$  can be obtained similarly due to symmetry. The complete procedure for predicting missing entries in  $\mathbf{A}$  using low-rank Hadamard decomposition is summarized in Algorithm 18.

---

**Algorithm 18** Alternating Descent with Gradient Descent for Hadamard Decomposition with Missing Entries: A regularization can also be added into the gradient descent update (see Section 12.3).

---

**Require:** Matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ;

- 1: Initialize  $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{M \times K}$ , and  $\mathbf{D}_1, \mathbf{D}_2 \in \mathbb{R}^{K \times N}$ ;
- 2: Choose a stopping criterion on the approximation error  $\delta$ ;
- 3: Choose step size  $\eta$ ;
- 4: Choose the maximum number of iterations  $C$ ;
- 5:  $iter = 0$ ; ▷ Count for the number of iterations
- 6: **while**  $\|(\mathbf{C}_1 \mathbf{D}_1) \circledast (\mathbf{C}_2 \mathbf{D}_2) - \mathbf{A}\|_F^2 > \delta$  and  $iter < C$  **do**
- 7:    $iter \leftarrow iter + 1$ ;
- 8:   **for**  $n = 1, 2, \dots, N$  **do**
- 9:      $\mathbf{d}_{1,n} \leftarrow \mathbf{d}_{1,n} - \eta \nabla L(\mathbf{d}_{1,n})$ ; ▷ Equation (12.33)
- 10:      $\mathbf{d}_{2,n} \leftarrow \mathbf{d}_{2,n} - \eta \nabla L(\mathbf{d}_{2,n})$ ;
- 11:   **end for**
- 12:   **for**  $m = 1, 2, \dots, M$  **do**
- 13:      $\mathbf{c}_{1,m} \leftarrow \mathbf{c}_{1,m} - \eta \nabla L(\mathbf{c}_{1,m})$ ; ▷ Equation (12.34)
- 14:      $\mathbf{c}_{2,m} \leftarrow \mathbf{c}_{2,m} - \eta \nabla L(\mathbf{c}_{2,m})$ ;
- 15:   **end for**
- 16: **end while**
- 17: Output  $\mathbf{C}_1, \mathbf{D}_1, \mathbf{C}_2, \mathbf{D}_2$ .

---

## 12.11. Application: Movie Recommender

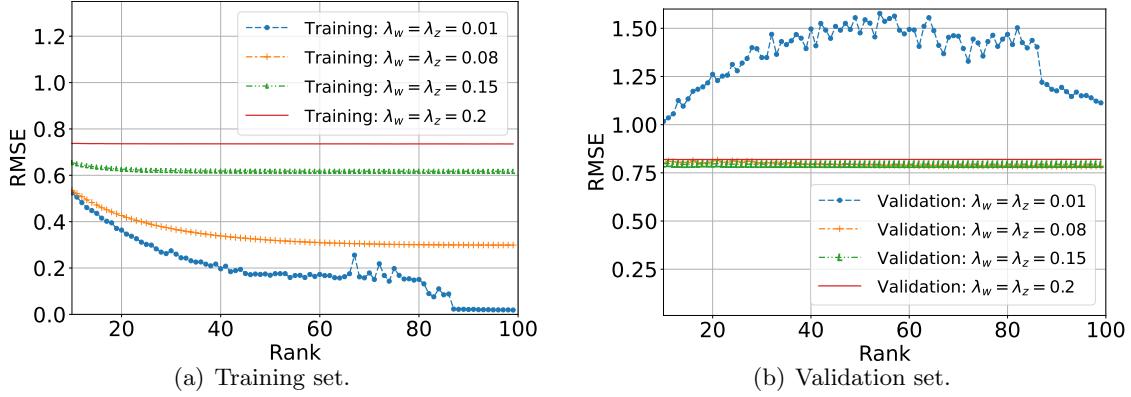
The ALS algorithm has been extensively developed for movie recommendation systems. To illustrate its application, we use the ‘‘MovieLens 100K’’ data set from MovieLens (Harper and Konstan, 2015)<sup>18</sup>. This data set is widely recognized and used in the field of recommender systems research due to its comprehensive set of user ratings for movies. It consists of 100,000 ratings from 943 users for 1,682 movies, with rating values ranging from 0 to 5. The data was collected through the MovieLens website over a seven-month period from

---

<sup>18</sup>. <http://grouplens.org>

September 19th, 1997 to 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, our focus will solely be on the raw rating matrix to evaluate how well the low-rank ALS approach can capture the underlying structure of the data, leading to accurate and meaningful recommendations.

The data set is split into training and validation set, comprising approximately 95,015 and 4,985 ratings, respectively, for fitting the ALS algorithm. The error is quantified using the *root mean squared error (RMSE)*. The RMSE is a common measure of the difference between actual and predicted values. For a set of values  $\{x_1, x_2, \dots, x_n\}$  and their predictions  $\{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_n\}$ , the RMSE can be described as  $\text{RMSE}(\mathbf{x}, \hat{\mathbf{x}}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2}$ . For evaluating the ALS algorithm, the minimum RMSE for the validation set is achieved with  $K = 62$  and  $\lambda_w = \lambda_z = 0.15$ , resulting in an RMSE of 0.806 (less than 1), as shown in Figure 12.6. Given that ratings range from 0 to 5, the ALS algorithm can predict whether a user is likely to enjoy a movie (e.g., ratings of 4 to 5) or not (e.g., ratings of 0 to 2) on average due to the RMSE score.



**Figure 12.6:** Comparison of training and validation error for the “MovieLens 100K” data set with different reduction dimensions and regularization parameters.

**Recommender 1.** A simple recommender system suggests movie  $m$  to user  $n$  if  $a_{mn} \geq 4$  and user  $n$  has not yet rated movie  $m$ .

**Recommender 2.** Alternatively, we can recommend similar movies to those highly rated by the user. Suppose user  $n$  has rated movie  $m$  with a 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 12.5 on vector inner products), the solution involves identifying the most similar movies that user  $n$  has not rated (or watched), to movie  $m$ . Mathematically, this is expressed as:

$$\arg \max_{\mathbf{w}_i} \text{similarity}(\mathbf{w}_i, \mathbf{w}_m), \quad \text{for all } i \notin \mathbf{o}_n,$$

where  $\mathbf{w}_i$ 's are the rows of  $\mathbf{W}$ , each representing the hidden features of movie  $i$ , and  $\mathbf{o}_n$  represents a mask vector, indicating the movies that user  $n$  has already rated.

The method described above relies on a similarity function applied to two vectors. The *cosine similarity* is the most commonly used measure. It is defined as the cosine of the

angle between the two vectors:

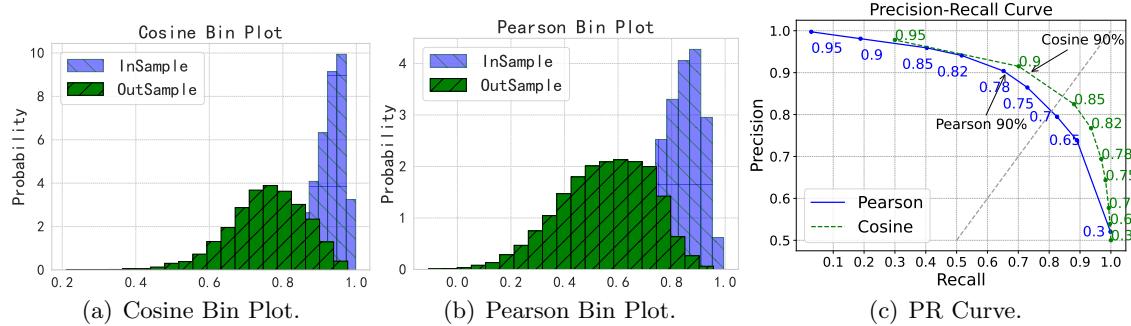
$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\|_2 \cdot \|\mathbf{y}\|_2},$$

where the value ranges from  $-1$  to  $1$ , with  $-1$  representing perfectly dissimilar and  $1$  being perfectly similar. Based on this definition, it follows that the cosine similarity depends only on the angle between the two nonzero vectors, but not on their magnitudes since it can be regarded as the inner product between the normalized versions of these vectors. Another measure for calculating similarity is the *Pearson similarity*:

$$\text{Pearson}(\mathbf{x}, \mathbf{y}) = \frac{\text{Cov}(\mathbf{x}, \mathbf{y})}{\sigma_x \cdot \sigma_y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}.$$

It is calculated as the ratio between the covariance of two variables and the product of their standard deviations, whose range varies between  $-1$  and  $1$ , where  $-1$  is perfectly dissimilar,  $1$  is perfectly similar, and  $0$  indicates no linear relationship. Pearson similarity is commonly used to measure the linear correlation between two sets of data.

Both Pearson correlation and cosine similarity are widely used in machine learning and data analysis. Pearson correlation is often used in regression analysis, while cosine similarity is commonly used in recommendation systems and information retrieval tasks. In our context, cosine similarity performs better in *precision-recall (PR) curve* analysis.



**Figure 12.7:** Distribution of the insample and outsample using cosine and Pearson similarities, and the Precision-Recall curves for both.

Building upon the previous example using the MovieLens 100K data set, we set  $\lambda_w = \lambda_z = 0.15$  for regularization and a rank of 62 to minimize RMSE. We aim to analyze the similarity between different movie hidden vectors, and the goal of Recommender 2 is to see whether the matrix factorization can help differentiate high-rated from low-rated movies, thereby recommending movies correlated with the user's high-rated ones. Define further the term “insample” as the similarity between the movies having rates 5 for each user, and “outsample” as the similarity between the movies having rates 5 and 1 for each user. Figure 12.7(a) and 12.7(b) depict the bin plots of the distributions of insample and outsample under cosine and Pearson similarities, respectively. In both scenarios, a clear distinction is observed between the distributions of the “insample” and “outsample” data, indicating that ALS decomposition can actually find the hidden features of different movies for each user. Figure 12.7(c) displays the *precision-recall (PR) curve* for these scenarios, where we find cosine similarity outperforms Pearson similarity, achieving over 73% recall with 90% precision. However, Pearson similarity can identify only about 64% of the high-rated movies with the same precision. In practice, other measures, such as *negative Euclidean distance*,

can also be explored. The Euclidean distance measures the “dissimilarity” between two vectors; and a negative value thus represents their similarity.

The ALS method for recommendation discussed here is designed for *explicit data*, where the ratings provided by each user have a clear hierarchical meaning. In contrast, there are also recommendation systems for *implicit data*, where the system automatically infers users’ preferences by tracking their actions, such as which items they viewed, where they clicked, which products they purchased, or how long they spent on a web page. In such cases, ALS can be extended to more complex models, such as using a dictionary matrix to transform the explicit data into user and item latent vectors (He et al., 2017), incorporating multinomial prior into a variational auto-encoder, and enhancing the model’s ability to handle implicit feedback by leveraging probabilistic modeling techniques (Liang et al., 2018).

## 12.12. Application: Outlier Detection

We can also use matrix decomposition algorithms to detect outlier entries in a matrix. Given an  $M \times N$  input matrix  $\mathbf{A}$ , the objective is to identify entries that significantly deviate from their reconstructed counterparts based on a low-rank approximation. Begin by selecting a rank  $K$  such that  $K < \min(M, N)$ , which captures the dominant structure of the data while filtering out noise or anomalies. Then, we employ a matrix factorization method such as ALS, SVD, nonnegative matrix factorization (NMF; see Chapter 13), or another suitable decomposition technique to approximate  $\mathbf{A}$  as  $\tilde{\mathbf{A}} = \mathbf{W}\mathbf{Z}$ , where  $\mathbf{W} \in \mathbb{R}^{M \times K}$  and  $\mathbf{Z} \in \mathbb{R}^{K \times N}$ . Subsequently, reconstruct the approximation  $\tilde{\mathbf{A}} = \mathbf{W}\mathbf{Z}$  and compute the element-wise reconstruction error matrix  $\mathbf{E} \in \mathbb{R}^{M \times N}$  defined by  $e_{ij} = |a_{ij} - \tilde{a}_{ij}|$  for all  $i, j$ . Establish a threshold  $\tau$  either through statistical means—such as mean absolute deviation or quantile-based methods—or via domain-specific criteria. Finally, we can declare an entry  $a_{ij}$  as an outlier if  $e_{ij} > \tau$ . The identified outliers correspond to those indices where the deviation between the original and reconstructed values exceeds the expected variability captured by the low-rank model. This approach exploits the ability of matrix factorization to capture the main patterns in the data. As a result, large reconstruction errors often indicate potential anomalies or unusual behavior.

## 12.13. Application: Spectral Clustering and Link Prediction

In Section 10.6.7, we discussed how spectral decomposition can be applied to perform spectral clustering on adjacency matrices of undirected graphs. However, adjacency matrices of directed graphs are asymmetric. For example, in a social network like Twitter, a teenager might follow a famous rock star, but the rock star may not follow the teenager in return.

Given an adjacency matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  of a directed graph, ALS can be used to find the low-dimensional representation for each node such that  $\mathbf{A} = \mathbf{U}\mathbf{V}^\top$ , where  $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{n \times k}$ . Unlike the symmetric case, the  $k$ -dimensional row vectors of  $\mathbf{U}$  and  $\mathbf{V}$  can be concatenated to form  $2k$ -dimensional embedded representations for each node.

These  $2k$ -dimensional embeddings can then be used for spectral clustering of the nodes, similar to the approach outlined in Section 10.6.7. Alternatively, once the factorization  $\mathbf{A} = \mathbf{U}\mathbf{V}^\top$  is obtained, the reconstructed matrix  $\mathbf{U}\mathbf{V}^\top$  can be used to predict links—just as in the Netflix recommendation problem. In such cases, when the connections between nodes are binary (i.e., an implicit data set), *logistic matrix decomposition* using ALS can also be applied to model the probability of a link; see Problem 12.16.

1. **Least squares for rank-deficiency** (Lu, 2021d). Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{b} \in \mathbb{R}^m$ . Show that the least squares problem  $L(\mathbf{x}) = \|\mathbf{Ax} - \mathbf{b}\|_2^2$  has a minimizer  $\mathbf{x}^* \in \mathbb{R}^n$  if and only if there exists a vector  $\mathbf{y} \in \mathbb{R}^n$  such that  $\mathbf{x}^* = \mathbf{A}^+ \mathbf{b} + (\mathbf{I} - \mathbf{A}^+ \mathbf{A}) \mathbf{y}$ , where  $\mathbf{A}^+$  denotes the pseudo-inverse of  $\mathbf{A}$ . Show that:

- The least squares has a **unique** minimizer of  $\mathbf{x}^* = \mathbf{A}^+ \mathbf{b}$  only when  $\mathbf{A}^+$  is a left inverse of  $\mathbf{A}$  (i.e.,  $\mathbf{A}^+ \mathbf{A} = \mathbf{I}$ ). The solution in Lemma 12.1 is a special case.
- The optimal value is  $L(\mathbf{x}^*) = \mathbf{b}^\top (\mathbf{I} - \mathbf{A} \mathbf{A}^+) \mathbf{b}$ .
- If  $\mathbf{y} \neq \mathbf{0}$ , then  $\|\mathbf{A}^+ \mathbf{b}\|_2 \leq \|\mathbf{A}^+ \mathbf{b} + (\mathbf{I} - \mathbf{A}^+ \mathbf{A}) \mathbf{y}\|_2$ .

*Hint: See LS via SVD in Section 11.6.*

2. **Least squares for rank-deficiency.** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{B} \in \mathbb{R}^{m \times p}$ . Show that the least squares problem  $L(\mathbf{X}) = \|\mathbf{AX} - \mathbf{B}\|_F^2$  has a minimizer  $\mathbf{X}^* = \mathbf{A}^+ \mathbf{B} \in \mathbb{R}^{n \times p}$ . Determine all the minimizers using Problem 12.1.
3. **Least squares for rank-deficiency.** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  and  $\mathbf{B} \in \mathbb{R}^{p \times n}$ . Show that the least squares problem  $L(\mathbf{X}) = \|\mathbf{XA} - \mathbf{B}\|_F^2$  has a minimizer  $\mathbf{X}^* = \mathbf{BA}^+ \in \mathbb{R}^{p \times m}$ .
4. Prove Lemma 12.9.
5. **Marginally convex.** Let  $D(\mathbf{A}, \mathbf{B})$  be convex in the second argument  $\mathbf{B}$ . Show that  $D(\mathbf{A}, \mathbf{WZ})$  is convex in  $\mathbf{W}$  for a fixed  $\mathbf{Z}$ , and vice versa.
6. Derive the column-by-column update rules for Algorithm 13.
7. **Orthogonal and projective matrix factorization.** Consider the optimization  $\min_{\mathbf{W}} \|\mathbf{A} - \mathbf{WZ}\|_F^2$  such that  $\mathbf{ZZ}^\top = \mathbf{I}_K$ , where  $\mathbf{A} \in \mathbb{R}^{M \times N}$ ,  $\mathbf{W} \in \mathbb{R}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}^{K \times N}$ , and  $K \leq \min\{M, N\}$ . Show that the optimal value  $\mathbf{W}^*$  given  $\mathbf{Z}$  is  $\mathbf{AZ}^\top$ . This indicates that the matrix factorization optimization can be equivalently stated as  $\min_{\mathbf{ZZ}^\top = \mathbf{I}_K} \|\mathbf{A} - \mathbf{AZ}^\top \mathbf{Z}\|_F^2$ . And the relaxed problem is called the *projective matrix factorization* (Yuan and Oja, 2005; Yang and Oja, 2010):

$$\min_{\mathbf{Z}} \|\mathbf{A} - \mathbf{AZ}^\top \mathbf{Z}\|_F^2,$$

where each row of  $\mathbf{A}$  is projected onto a  $K$ -dimensional subspace, hence the name. The interpretations of orthogonal and projective matrix factorizations are further discussed in Problem 13.11.

8. **Regularized least squares (RLS).** Given  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$ ,  $\mathbf{B} \in \mathbb{R}^{p \times n}$ , and  $\lambda \in \mathbb{R}_{++}$ , we consider the regularized least squares (RLS) problem:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{Bx}\|_2^2.$$

Show that this regularized least squares problem has a unique solution if and only if  $\mathcal{N}(\mathbf{A}) \cap \mathcal{N}(\mathbf{B}) = \{\mathbf{0}\}$ .

9. **Denoising via RLS.** Consider a noisy measurement of a signal  $\mathbf{x} \in \mathbb{R}^n$ :  $\mathbf{y} = \mathbf{x} + \mathbf{e}$ , where  $\mathbf{y}$  is the observed measurement, and  $\mathbf{e}$  is the noise vector. We want to find an estimate  $\mathbf{x}$  of the observed measurement  $\mathbf{y}$  such that  $\mathbf{x} \approx \mathbf{y}$ :  $\min \|\mathbf{x} - \mathbf{y}\|_2^2$ . Apparently, the optimal solution of this optimization is given by  $\mathbf{x} = \mathbf{y}$ ; however, it is meaningless. To improve the estimate, we can add a penalty term for the differences between consecutive observations:  $R(\mathbf{x}) = \sum_{i=1}^{n-1} (x_i - x_{i+1})^2$ . Then,

- Find the regularized least squares representation for this problem and derive the regularized least squares solution.
- Find some applications of this denoising problem. For example, when we model the profit and loss signal of a financial asset, the two observations over consecutive days of the underlying asset should exhibit smooth transitions rather than abrupt changes.

10. **Weighted least squares (WLS).** Building upon the assumptions in Lemma 12.1, we consider further that each data point  $i \in \{1, 2, \dots, m\}$  (i.e., each row of  $\mathbf{A}$ ) has a weight  $w_i$ . This means some data points may carry greater significance than others, and we can produce approximate minimizers that reflect this. Show that the value  $\mathbf{x}_{WLS} = (\mathbf{A}^\top \mathbf{W}^2 \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{W}^2 \mathbf{b}$  serves as the *weighted least squares (WLS)* estimate of  $\mathbf{x}$ , where  $\mathbf{W} = \text{diag}(w_1, w_2, \dots, w_m) \in \mathbb{R}^{m \times m}$ . Hint: Find the normal equation for this problem.
11. **Positive definite weighted least squares (PDWLS).** Building upon the assumptions in Lemma 12.1, we consider further the matrix equation  $\mathbf{Ax} + \mathbf{e} = \mathbf{b}$ , where  $\mathbf{e}$  is an error vector. Define the weighted error squared sum  $E_w = \mathbf{e}^\top \mathbf{We}$ , where the weighting matrix  $\mathbf{W}$  is positive definite. Show that the positive definite weighted least squares solution is  $\mathbf{x}^* = (\mathbf{A}^\top \mathbf{WA})^{-1} \mathbf{A}^\top \mathbf{Wb}$ . Hint: Compute the gradient of  $E_w = (\mathbf{b} - \mathbf{Ax})^\top \mathbf{W}(\mathbf{b} - \mathbf{Ax})$ .
12. **Weighted color noise least squares.** Building upon the assumptions in Lemma 12.1, we consider the matrix equation  $\mathbf{Ax} + \mathbf{e} = \mathbf{b}$ , where  $\mathbf{e}$  is an additive color noise vector satisfying the conditions  $\mathbb{E}[\mathbf{e}] = \mathbf{0}$  and  $\mathbb{E}[\mathbf{ee}^\top] = \Sigma$ , where  $\Sigma$  is known. Use the weighting error function  $E_w = \mathbf{e}^\top \mathbf{We}$  as the loss function for finding the optimal estimate  $\mathbf{x}^*$ . Show that  $\mathbf{x}^* = (\mathbf{A}^\top \mathbf{WA})^{-1} \mathbf{A}^\top \mathbf{Wb}$ , where the optimal choice of the weighting matrix  $\mathbf{W}$  is  $\mathbf{W}^* = \Sigma^{-1}$ . Hint: Compute the gradient of  $E_w = (\mathbf{b} - \mathbf{Ax})^\top \mathbf{W}(\mathbf{b} - \mathbf{Ax})$ .
13. **Transformed least squares (TLS).** Building upon the assumptions in Lemma 12.1, we consider further the restriction  $\mathbf{x} = \mathbf{C}\gamma + \mathbf{c}$ , where  $\mathbf{C} \in \mathbb{R}^{n \times k}$  is a known matrix such that  $\mathbf{AC}$  has full rank,  $\mathbf{c}$  is a known vector, and  $\gamma$  is an unknown vector. Show that the value  $\mathbf{x}_{TLS} = \mathbf{C}(\mathbf{C}^\top \mathbf{A}^\top \mathbf{AC})^{-1}(\mathbf{C}^\top \mathbf{A}^\top)(\mathbf{b} - \mathbf{Ac}) + \mathbf{c}$  serves as the *transformed least squares (TLS)* estimate of  $\mathbf{x}$ .
14. Find the transformed weighted least squares estimate.
15. **ALS for weighted matrix decomposition.** Let  $\mathbf{A} \in \mathbb{R}^{m \times n}$  be the data matrix and  $\mathbf{M} \in \mathbb{R}^{m \times n}$  be the corresponding weight matrix, where each entry of  $\mathbf{M}$  represents the weight associated with the corresponding entry in  $\mathbf{A}$ . Derive the ALS algorithm of the *weighted matrix decomposition* problem:

$$L(\mathbf{W}, \mathbf{Z}) = \|\mathbf{M} \circledast (\mathbf{A} - \mathbf{WZ})\|_F^2. \quad (12.35)$$

In the context of *implicit feedback data*, each entry of  $\mathbf{A}$  takes on a binary value; for example, 1 if the user rates a movie with 3, 4, or 5 stars, and 0 if the rating is 0, 1, or 2 stars. Suppose that matrix  $\mathbf{B} \in \mathbb{R}^{m \times n}$  contains the original raw ratings of the matrix (ranging from 0 to 5). A relaxation of this implicit feedback representation can be achieved using a weight matrix  $\mathbf{M}$ , defined as:

$$m_{ij} = 1 + \gamma \cdot b_{ij}, \quad \forall i, j,$$

where the parameter  $\gamma$  is typically set to a large value, such as  $\gamma = 40$ . Discuss the relationship between this formulation of weighted matrix decomposition and the standard matrix decomposition used for implicit feedback data.

16. **ALS for logistic matrix decomposition.** Let  $\mathbf{A} \in \{0, 1\}^{m \times n}$  be the observed binary data matrix. The logistic matrix decomposition aims to factor  $\mathbf{A}$  into  $\mathbf{W} \in \mathbb{R}^{m \times k}$  and  $\mathbf{Z} \in \mathbb{R}^{k \times n}$  such that

$$L(\mathbf{W}, \mathbf{Z}) = \|\sigma(\mathbf{WZ}) - \mathbf{A}\|_F^2$$

is minimized, where  $\sigma(x) = 1/(1 + \exp\{-x\})$  denotes the logistic sigmoid function, applied element-wise to the matrix  $\mathbf{WZ}$ . Alternatively, since we want to learn  $\mathbf{W}$

and  $\mathbf{Z}$  such that  $\sigma(\mathbf{W}\mathbf{Z})_{ij}$  has a large value when  $a_{ij}$  is 1, and a small value when  $a_{ij}$  is 0, we can use the log-likelihood loss function:

$$L(\mathbf{W}, \mathbf{Z}) = - \sum_{i,j=1}^{m,n} [a_{ij} \ln (\sigma(\mathbf{w}_i, \mathbf{z}_j)) + (1 - a_{ij}) \ln (1 - \sigma(\mathbf{w}_i, \mathbf{z}_j))], \quad (12.36)$$

where  $\mathbf{w}_i$  and  $\mathbf{z}_j$  denote the  $i$ -th row of  $\mathbf{W}$  and the  $j$ -th column of  $\mathbf{Z}$ , respectively. Derive the ALS steps for this loss function, and discuss how it can be applied to the implicit feedback data introduced in Problem 12.15.

17. **ALS for shared matrix decomposition.** In the main text, we consider a matrix,  $\mathbf{A} \in \mathbb{R}^{m \times n}$ , representing the user-movie matrix, where rows correspond to movies, columns to users, and entries contain the observed ratings. The product  $\mathbf{W}\mathbf{Z}$  reconstructs the full rating matrix, including predictions for missing entries. Suppose we are further given a user-book matrix  $\mathbf{B} \in \mathbb{R}^{p \times n}$ , where the rows correspond to book items, and the columns contain the same set of users. To leverage both types of data (movies and books), we consider the *shared matrix decomposition* problem:

$$\min_{\mathbf{W}, \mathbf{Y}, \mathbf{Z}} L(\mathbf{W}, \mathbf{Y}, \mathbf{Z}) = \|\mathbf{A} - \mathbf{W}\mathbf{Z}\|_F^2 + \|\mathbf{B} - \mathbf{Y}\mathbf{Z}\|_F^2 + \lambda(\|\mathbf{W}\|_F^2 + \|\mathbf{Y}\|_F^2 + \|\mathbf{Z}\|_F^2). \quad (12.37)$$

Since the columns of  $\mathbf{Z}$  represent latent features of users, these representations gain information from both the movie and book data. Derive the ALS steps for this problem.

18. **First-order optimality condition for local optima points.** Consider *Fermat's theorem*: for a one-dimensional function  $g(\cdot)$  defined and differentiable over an interval  $(a, b)$ , if a point  $x^* \in (a, b)$  is a local maximum or minimum, then  $g'(x^*) = 0$ . Prove the first-order optimality conditions for multivariate functions based on Fermat's theorem for one-dimensional functions. That is, let  $f : \mathbb{S} \rightarrow \mathbb{R}$  be a function defined on a set  $\mathbb{S} \subseteq \mathbb{R}^n$ . Suppose that  $\mathbf{x}^* \in \text{int}(\mathbb{S})$ , i.e., in the interior point of the set, is a local optimum point and that all the partial derivatives of  $f$  exist at  $\mathbf{x}^*$ . Then  $\nabla f(\mathbf{x}^*) = \mathbf{0}$ , i.e., the gradient vanishes at all local optimum points. (Note that, this optimality condition is a necessary condition but not sufficient; however, there could be vanished points which are not local maximum or minimum points.) *Hint: Consider the one-dimensional function  $g(t) = f(\mathbf{x}^* + t\mathbf{e}_i)$  for  $i \in \{1, 2, \dots, n\}$ .*
19. **Global minimum point of convex functions.** Let the function  $f$  be a twice continuously differentiable function defined over  $\mathbb{R}^n$ . Suppose that the Hessian  $\nabla^2 f(\mathbf{x}) \succeq 0$  for any  $\mathbf{x} \in \mathbb{R}^n$  (i.e., the Hessian is always positive semidefinite <sup>19</sup>). This property is also referred to as the *convexity*. Show that  $\mathbf{x}^*$  is a global minimum point of  $f$  if  $\nabla f(\mathbf{x}^*) = \mathbf{0}$ . *Hint: Use the linear approximation theorem from Taylor's expansion.*
20. **Two-sided matrix least squares (Friedland and Torokhti, 2007; Aggarwal, 2020).** Let  $\mathbf{B}$  be an  $M \times K$  matrix and  $\mathbf{C}$  be a  $P \times N$  matrix. Find the  $K \times P$  matrix  $\mathbf{X}$  such that  $L(\mathbf{X}) = \|\mathbf{A} - \mathbf{B}\mathbf{X}\mathbf{C}\|_F^2$  is minimized, where  $\mathbf{A} \in \mathbb{R}^{M \times N}$  is known.
- Derive the derivative of  $L$  with respect to  $\mathbf{X}$  and the optimality conditions.
  - Show that one possible solution to the optimality conditions is  $\mathbf{X}^* = \mathbf{B}^+ \mathbf{A} \mathbf{C}^+$ , where  $\mathbf{B}^+$  and  $\mathbf{C}^+$  are the pseudo-inverses of  $\mathbf{B}$  and  $\mathbf{C}$ , respectively.

Similarly, consider the optimization with  $\text{rank}(\mathbf{X}) \leq p$ :  $L(\mathbf{X}) = \|\mathbf{A} - \mathbf{B}\mathbf{X}\mathbf{C}\|_F^2$ , s.t.  $\text{rank}(\mathbf{X}) \leq p$ . Show that

---

<sup>19</sup> Instead, if we assume the Hessian is positive semidefinite at a given point, then the point is a local minimum point.

- One possible solution to this is  $\mathbf{X}^* = \mathbf{B}^+ \mathbf{A}_p \mathbf{C}^+$ , where  $\mathbf{A}_p$  a truncated SVD of  $\mathbf{B}\mathbf{B}^+ \mathbf{A}\mathbf{C}^+\mathbf{C}$  by replacing all but the  $p$  largest singular values by zero.
  - $\mathbf{X}^*$  also minimizes  $\|\mathbf{X}\|_F$ , i.e., has the smallest magnitude among all solutions.
  - $\mathbf{X}^*$  is the **unique** solution if and only if either  $\text{rank}(\mathbf{B}\mathbf{B}^+ \mathbf{A}\mathbf{C}^+\mathbf{C}) \leq p$  or both  $\text{rank}(\mathbf{B}\mathbf{B}^+ \mathbf{A}\mathbf{C}^+\mathbf{C}) \geq p$  and  $\sigma_{p+1}(\mathbf{B}\mathbf{B}^+ \mathbf{A}\mathbf{C}^+\mathbf{C}) < \sigma_p(\mathbf{B}\mathbf{B}^+ \mathbf{A}\mathbf{C}^+\mathbf{C})$ .
21. **Monotonic progress of gradient descent.** Consider the gradient descent for a differentiable function  $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$  that is  $L$ -strongly smooth <sup>20</sup>. Suppose the iterate  $\mathbf{x}^{(t+1)}$  is obtained from iterate  $\mathbf{x}^{(t)}$  by  $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla f(\mathbf{x}^{(t)})$ . Show that
- If the step size  $\eta \leq \frac{2}{L}$ , the function value  $f$  is nonincreasing:  $f(\mathbf{x}^{(t+1)}) \leq f(\mathbf{x}^{(t)})$ .
  - If the step size  $\eta \in [\frac{1}{2L}, \frac{1}{L}]$ , the gradient satisfies  $\|\nabla f(\mathbf{x}^{(t)})\|_2 \leq \epsilon$  after  $T = \mathcal{O}(\frac{1}{\epsilon^2})$  steps.
22. (Rennie and Srebro, 2005; Mazumder et al., 2010) Consider the nuclear norm <sup>21</sup>  $\|\mathbf{A}\|_n$  of any matrix  $\mathbf{A} \in \mathbb{R}^{m \times n}$  of rank  $r$ . Show that
- $$\|\mathbf{A}\|_n = \min_{\substack{\mathbf{W} \in \mathbb{R}^{m \times r} \\ \mathbf{Z} \in \mathbb{R}^{r \times n}}} \frac{1}{2} (\|\mathbf{W}\|_F^2 + \|\mathbf{Z}\|_F^2) \quad \text{s.t.} \quad \mathbf{A} = \mathbf{W}\mathbf{Z}.$$
23. Let  $\mathbf{A}_1, \mathbf{A}_2 \in \mathbb{R}^{m \times n}$  be any  $m \times n$  matrices of rank  $r_1$  and rank  $r_2$ , respectively. Show that their Hadamard product  $\mathbf{A}_1 \circledast \mathbf{A}_2$  has rank at most  $r_1 \cdot r_2$ :  $\text{rank}(\mathbf{A}_1 \circledast \mathbf{A}_2) \leq \text{rank}(\mathbf{A}_1)\text{rank}(\mathbf{A}_2)$ .
24. **Modified LS.** Consider a modified least squares problem of minimizing  $\|\mathbf{Ax} - \mathbf{b}\|_2^2 + \mathbf{c}^\top \mathbf{x}$ , where  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ , and  $\mathbf{b} \in \mathbb{R}^m$ . Show that the problem can be reduced to the standard least squares problem as long as  $\mathbf{c}$  lies in the row space of  $\mathbf{A}$ . What happens when  $\mathbf{c}$  does not lie in the row space of  $\mathbf{A}$ ? Hint: First examine the univariate version of this problem.

<sup>20</sup> A continuously differentiable function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is called *L-strongly smooth (SS)* if, for every  $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ , it follows that  $f(\mathbf{y}) - f(\mathbf{x}) - f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) \leq \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|^2$ .

<sup>21</sup> The nuclear norm is defined as the sum of singular values of a matrix and provides the tightest convex envelope of the rank function of a matrix.



# 13

## Nonnegative Matrix Factorization (NMF)

### 13.1. Nonnegative Matrix Factorization

In the era of big data, extracting meaningful patterns and latent structures from high-dimensional data sets has become a central challenge in various scientific and technological fields. Singular value decomposition (SVD) is supported by strong theoretical foundations and is applicable in a wide range of contexts. However, it has certain limitations; for example, when applied to a nonnegative matrix<sup>1</sup>, SVD may produce negative values in the resulting factors, which can be difficult to interpret meaningfully. To overcome this limitation, *nonnegative matrix factorization (NMF)* has emerged as a powerful and interpretable tool for dimensionality reduction, feature extraction, and discovering latent structures within complex data. Early consideration of the NMF problem was due to Paatero and Tapper (1994); Cohen and Rothblum (1993), who referred to it as *positive matrix factorization*. Later, Lee and Seung (2001) popularized the problem with the introduction of the *multiplicative update* rule.

Following the discussion of matrix factorization using the alternating least squares (ALS) method, we now turn to algorithms for solving the NMF problem:

- Given a nonnegative matrix  $\mathbf{A} \in \mathbb{R}_+^{M \times N}$  of rank  $r$ , find nonnegative matrix factors  $\mathbf{W} \in \mathbb{R}_+^{M \times K}$  and  $\mathbf{Z} \in \mathbb{R}_+^{K \times N}$  such that:  $\mathbf{A} \approx \mathbf{W}\mathbf{Z}$ .

As discussed in the ALS section, a fundamental challenge in linear data analysis involves transforming or decomposing a high-dimensional data vector into a linear combination of lower-dimensional vectors. This transformation captures the essential characteristics of the original data, making it suitable for tasks such as pattern recognition. Consequently, these lower-dimensional vectors are often referred to as “*hidden vectors*,” “*pattern vectors*,” or “*feature vectors*.” When conducting data analysis, building models, and processing information, two primary requirements for a pattern vector are essential:

- Interpretability.* Each component of a pattern vector should possess clear physical or physiological significance, allowing for a meaningful interpretation of the underlying data.
- Statistical fidelity.* In cases where the data are reliable and contain minimal error or noise, the components of a pattern vector should effectively capture the variability within the data, reflecting its primary structure and distribution of information.

<sup>1</sup>. Nonnegative matrices possess unique properties in linear algebra and are crucial for theoretical analysis; see Problems 13.12~13.17.

The NMF approach addresses these issues in various applications. For example:

- In document collections, documents are represented as vectors, with each vector element indicating the frequency (often weighted) of a specific term within the document. Arranging these document vectors sequentially forms a nonnegative term-by-document matrix, which provides a numerical representation of the entire document collection.
- In image collections, each image is depicted by a vector, where each vector element represents a pixel. The value of each element, a nonnegative number, reflects the intensity and color of the corresponding pixel, leading to a nonnegative pixel-by-image matrix.
- In gene expression analysis, observations from gene sequences under different experimental conditions are compiled into gene-by-experiment matrices. These matrices encapsulate the variations in gene expression across experiments.
- For item sets or recommendation systems, customer purchase histories or ratings for a selection of items are recorded in a nonnegative sparse matrix. This matrix efficiently captures the sparse nature of user interactions with a large number of potential items.

Unlike arbitrary linear combinations, the linear combinations in the NMF context involve only nonnegative weights of nonnegative *template vectors* (or *basis vectors*, i.e., the columns of  $\mathbf{W}$ ). This prevents phenomena such as *destructive interference*, where a positive component could be canceled out by adding a negative component. Instead, data vectors must be explained using purely constructive methods, involving only positive components. The nonnegativity constraint inherently imposes **sparsity**, enabling the factorization to capture additive features, which is especially advantageous in applications where parts-based representations are meaningful. This property has led to its widespread use in fields such as text mining, image processing, document analysis, and bioinformatics, where the identified components often correspond to distinct parts or features. For example, in image processing, NMF has proven valuable for tasks such as object detection, image segmentation, and facial recognition (Lee and Seung, 2001; Gillis, 2014, 2020). The decomposition into nonnegative components aligns with the intuitive notion that images are composed of identifiable parts. In the topic recovery problem, each column of  $\mathbf{A}$  denotes a document; NMF aligns with a soft clustering approach, where each column of  $\mathbf{W}$  represents a topic, and the positive entries of each column of  $\mathbf{Z}$  represent the positive weights of each document for those topics (Shahnaz et al., 2006). On the other hand, a nonnegative matrix factorization  $\mathbf{A} \approx \mathbf{WZ}$  can be applied directly for clustering algorithms. Specifically, the data vector  $\mathbf{a}_j$  is assigned to cluster  $i$  if  $z_{ij}$  is the largest element in column  $j$  of  $\mathbf{Z}$  (Brunet et al., 2004; Gao and Church, 2005). For further applications, see the survey by Berry et al. (2007). In conclusion, the popularity of NMF stems from its ability to automatically extract sparse and easily interpretable factors.

To measure the quality of the approximation, we evaluate the loss by computing the Frobenius norm of the difference between the original matrix and its reconstruction:

$$L(\mathbf{W}, \mathbf{Z}) = D(\mathbf{A}, \mathbf{WZ}) = \frac{1}{2} \|\mathbf{WZ} - \mathbf{A}\|_F^2, \quad ^2 \tag{13.1}$$

where  $L(\mathbf{W}, \mathbf{Z})$  indicates it is a loss function w.r.t.  $\mathbf{W}$  and  $\mathbf{Z}$ , and  $D(\mathbf{A}, \mathbf{WZ})$  implies it is a distance/divergence between  $\mathbf{A}$  and  $\mathbf{WZ}$  (we will use the two notations interchangeably as needed). The Frobenius norm is arguably the most widely used norm for NMF because it corresponds to Gaussian additive noise, which is reasonable in many situations and allows

---

<sup>2</sup>. Note that the factor  $\frac{1}{2}$  is included for analytical convenience in derivative calculations.

for the design of particularly efficient algorithms. For nonnegative data, Gaussian noise can be interpreted as a truncated version of standard Gaussian noise (Lu, 2023). In later sections, we will extend this approach to include more general  $\beta$ -divergences (Section 13.8).

When we want to find two nonnegative matrices  $\mathbf{W} \in \mathbb{R}_+^{M \times r}$  and  $\mathbf{Z} \in \mathbb{R}_+^{r \times N}$  such that  $\mathbf{A} = \mathbf{W}\mathbf{Z}$ , the problem is known as the *Exact NMF* of  $\mathbf{A}$  of size  $r$ . However, exact NMF is NP-hard (Vavasis, 2010; Gillis, 2020). Therefore, we focus on the approximate NMF formulation in this discussion. In the context of collaborative filtering, it is recognized that NMF via multiplicative updates can result in overfitting despite favorable convergence properties. The overfitting issue can be partially mitigated through regularization, but its out-of-sample performance may still be limited. Bayesian optimization through the use of generative models, on the other hand, can effectively prevent overfitting in nonnegative matrix factorization (Brouwer et al., 2017; Lu and Ye, 2022; Lu, 2023).

In the following sections, we introduce several methods for solving NMF problems and provide a brief overview of their applications.

---

**Algorithm 19** Projected Gradient Descent Method

---

**Require:** A function  $f(\mathbf{x})$  and a set  $\mathbb{S}$ ;

- 1: **for**  $t = 1, 2, \dots$  **do**
  - 2:     Pick a step size  $\eta_t$ ;
  - 3:     Set  $\mathbf{x}^{(t+1)} \leftarrow \mathcal{P}_{\mathbb{S}}(\mathbf{x}^{(t)} - \eta_t \nabla f(\mathbf{x}^{(t)}))$ ;
  - 4: **end for**
  - 5: Output final  $\mathbf{x}$ ;
- 

## 13.2. NMF via Alternating Projected Gradient Descent (APGD)

The projected gradient descent (PGD, Algorithm 19) is designed to minimize a function over a constraint set  $\mathbb{S}$ :

$$\min_{\mathbf{x} \in \mathbb{S}} f(\mathbf{x}).$$

The *orthogonal projection* onto  $\mathbb{S}$  is defined as  $\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = \arg \min_{\mathbf{y} \in \mathbb{S}} \|\mathbf{y} - \mathbf{x}\|_2$ . When  $\mathbb{S}$  is the nonnegative orthant, the projection  $\mathcal{P}_{\mathbb{S}}(\mathbf{x})$  simplifies to  $\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = \max\{\mathbf{0}, \mathbf{x}\}$ , where the max operator is applied componentwise.

Therefore, the *alternating PGD (APGD)* approach for NMF updates the factored components iteratively by

$$\mathbf{Z} \leftarrow \max \left\{ \mathbf{0}, \arg \min_{\mathbf{Z} \in \mathbb{R}^{K \times N}} \|\mathbf{WZ} - \mathbf{A}\|_F \right\} \quad \text{and} \quad \mathbf{W} \leftarrow \max \left\{ \mathbf{0}, \arg \min_{\mathbf{W} \in \mathbb{R}^{M \times K}} \|\mathbf{WZ} - \mathbf{A}\|_F \right\},$$

where each update can be solved using a least squares method followed by projection onto the nonnegative orthant. However, due to the projection, the solution may not be properly scaled. A closed-form scaling factor  $\gamma$  can be applied at each iteration to improve the approximation:

$$\gamma^* = \arg \min_{\gamma \geq 0} \|\gamma \mathbf{WZ} - \mathbf{A}\|_F = \frac{\langle \mathbf{A}, \mathbf{WZ} \rangle}{\langle \mathbf{WZ}, \mathbf{WZ} \rangle} = \frac{\langle \mathbf{AZ}^\top, \mathbf{W} \rangle}{\langle \mathbf{W}^\top \mathbf{W}, \mathbf{Z} \mathbf{Z}^\top \rangle}.$$

While it is generally not advised to use APGD due to its convergence challenges, APGD can be quite effective as an initialization method. This approach involves running a few iterations of APGD before switching to a different NMF algorithm, which is particularly beneficial for sparse matrices (Gillis, 2014).

### 13.3. NMF via Alternating Nonnegative Least Squares (ANLS)

A fundamental component of the ALS approach is the least squares problem (Lemma 12.1). For NMF, we focus on the *nonnegative least squares (NNLS)* problem:

$$\min_{\mathbf{x} \geq \mathbf{0}} f(\mathbf{x}) = \min_{\mathbf{x} \geq \mathbf{0}} \frac{1}{2} \|\mathbf{b} - \mathbf{M}\mathbf{x}\|_2^2 \quad \text{with } \mathbf{M} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m, \mathbf{x} \in \mathbb{R}_+^n. \quad (13.2)$$

The KKT conditions imply the complementary slackness condition  $\lambda_i x_i^* = 0, \forall i$ , where  $\lambda_i$  is the Lagrangian multiplier; and the optimal condition  $\nabla f(\mathbf{x}^*) - \sum_i \lambda_i \mathbf{e}_i = \mathbf{0}$ , where  $\mathbf{x}^*$  denotes the optimal solution of the NNLS problem. Together, the complementary slackness and the optimal condition indicate that:

$$\nabla f(\mathbf{x}^*) = \sum_{i: x_i^* = 0} \lambda_i \mathbf{e}_i.$$

From this, we derive the following equivalent KKT conditions for NNLS:

$$(\text{KKT of NNLS}) \quad \mathbf{x}^* \geq \mathbf{0}, \quad \nabla f(\mathbf{x}^*) \geq \mathbf{0}, \quad \text{and} \quad x_i^*(\nabla f(\mathbf{x}^*))_i = 0, \forall i. \quad (13.3)$$

These conditions imply sparsity when the nonnegative constraint is applied, meaning the NNLS or NMF problem inherently imposes a **sparsity constraint**.

Assume we are given the inactive set  $\mathbb{I} \subseteq \{1, 2, \dots, n\}$ :

$$\mathbb{I} = \{i \mid x_i^* > 0, \forall i \in \{1, 2, \dots, n\}\}.$$

The complement of  $\mathbb{I}$  is the so-called *active set*, where the corresponding constraints are active. That is, the active set contains indices  $i$  such that  $x_i^* = 0$ . The nonzero entries of  $\mathbf{x}^*$  can be determined by solving the following reduced linear system:

$$[\nabla_{\mathbf{x}} f(\mathbf{x})]_{\mathbb{I}} = \mathbf{0} \iff [\mathbf{M}^\top (\mathbf{M}\mathbf{x} - \mathbf{b})]_{\mathbb{I}} = \mathbf{0} \iff \mathbf{M}[:, \mathbb{I}]^\top \mathbf{M}[:, \mathbb{I}] \mathbf{x}[\mathbb{I}] = \mathbf{M}[:, \mathbb{I}]^\top \mathbf{b}.$$

This is precisely the normal equation for the unconstrained least squares problem w.r.t.  $\mathbf{x}[\mathbb{I}]$ , that is,

$$\min_{\mathbf{x}[\mathbb{I}]} \frac{1}{2} \|\mathbf{b} - \mathbf{M}[:, \mathbb{I}] \mathbf{x}[\mathbb{I}]\|_2^2.$$

This observation forms the basis of the *active-set method*, which iteratively updates the active set through pivoting (that is, entering and removing variables from the active set) to ensure the objective function decreases (Lawson and Hanson, 1995); see Algorithm 20.

**Alternating nonnegative least squares (ANLS).** Once we have the active-set method for NNLS problems, NMF can be achieved by replacing OLS in ALS algorithms with NNLS, known as *alternating nonnegative least squares (ANLS)* (Kim and Park, 2011). Given a fixed  $\mathbf{W}$ , the NMF objective can be solved for each column of  $\mathbf{Z}$  separately:

$$\frac{1}{2} \|\mathbf{A} - \mathbf{WZ}\|_F^2 = \frac{1}{2} \sum_{n=1}^N \|\mathbf{a}_n - \mathbf{Wz}_n\|_2^2,$$

where each subproblem  $\min_{\mathbf{z}_n \geq \mathbf{0}} \|\mathbf{a}_n - \mathbf{Wz}_n\|_2^2$  can be solved using NNLS. Since the NMF problem is symmetric:  $\mathbf{A} = \mathbf{WZ}$  if and only if  $\mathbf{A}^\top = \mathbf{Z}^\top \mathbf{W}^\top$  such that  $D(\mathbf{A}, \mathbf{WZ}) = D(\mathbf{A}^\top, \mathbf{Z}^\top \mathbf{W}^\top)$ . The analysis of optimizing  $\mathbf{W}$  given  $\mathbf{Z}$  follows directly from the previous methodology. We should also note that since the initial guess of  $\mathbf{W}$  and  $\mathbf{Z}$  typically offers a poor approximation of  $\mathbf{A}$ , solving the NNLS subproblems exactly in the early stages of the alternating algorithms is often unnecessary. Instead, it can be more efficient to use ANLS as a refinement step within a less computationally expensive NMF algorithm, such as APGD or MU (discussed in later sections)

**Algorithm 20** Nonnegative Least Squares (NNLS) via Active-Set Method

**Require:** A real-valued matrix  $M \in \mathbb{R}^{m \times n}$ , a real-valued vector  $b \in \mathbb{R}^m$ ;

- ```

1: Initialize index sets  $\mathbb{I} = \emptyset$  and  $\mathbb{J} = \{1, \dots, n\}$ ;
2: Initialize unknown  $\mathbf{x} \in \mathbb{R}^n$  to an all-zero vector and let  $\mathbf{w} \leftarrow \mathbf{M}^\top(\mathbf{b} - \mathbf{M}\mathbf{x})$ ;
3: Let  $\mathbf{w}[\mathbb{J}]$  denote the sub-vector with indices from  $\mathbb{J}$ ;
4: Choose a stopping criterion on the approximation error  $\delta$ ;
5: Choose the maximal number of iterations  $C$ ;
6:  $iter = 0$ ; ▷ Count for the number of iterations
7: while  $\mathbb{J} \neq \emptyset$  and  $\max(\mathbf{w}[\mathbb{J}]) > \delta$  and  $iter < C$  do
8:    $iter = iter + 1$ ;
9:   Let  $j$  in  $\mathbb{J}$  be the index of  $\max(\mathbf{w}[\mathbb{J}])$  in  $\mathbf{w}$ :  $j = \arg \max_{j \in \mathbb{J}} w_j$ ;
10:  Add  $j$  to  $\mathbb{I}$  and remove  $j$  from  $\mathbb{J}$  such that  $\mathbb{I} \cup \mathbb{J} = \{1, 2, \dots, n\}$ ;
11:  Let  $\mathbf{M}[:, \mathbb{I}]$  be  $\mathbf{M}$  restricted to the variables/columns included in  $\mathbb{I}$ ;
12:  Let  $\mathbf{s}$  be vector of same length as  $\mathbf{x}$ ; Let  $\mathbf{s}[\mathbb{I}]$  denote the sub-vector with indices from  $\mathbb{I}$ , and let  $\mathbf{s}[\mathbb{J}]$  denote the sub-vector with indices from  $\mathbb{J}$ ;
13:  Set  $\mathbf{s}[\mathbb{I}] \leftarrow ((\mathbf{M}[:, \mathbb{I}])^\top \mathbf{M}[:, \mathbb{I}])^{-1} (\mathbf{M}[:, \mathbb{I}])^\top \mathbf{b}$  and  $\mathbf{s}[\mathbb{J}]$  to zero;
14:  while  $\min(\mathbf{s}[\mathbb{I}]) \leq 0$  do
15:    Let  $\alpha \leftarrow \min \frac{x_i}{x_i - s_i}$  for  $i$  in  $\mathbb{I}$  where  $s_i \leq 0$ ;
16:    Set  $\mathbf{x} \leftarrow \mathbf{x} + \alpha(\mathbf{s} - \mathbf{x})$ ;
17:    Move to  $\mathbb{J}$  all indices  $j$  in  $\mathbb{I}$  such that  $x_j \leq 0$ ;
18:    Set  $\mathbf{s}[\mathbb{I}] \leftarrow ((\mathbf{M}[:, \mathbb{I}])^\top \mathbf{M}[:, \mathbb{I}])^{-1} (\mathbf{M}[:, \mathbb{I}])^\top \mathbf{b}$ ;
19:  end while
20:  Set  $\mathbf{s}[\mathbb{J}]$  to zero;
21:  Set  $\mathbf{x} \leftarrow \mathbf{s}$ ;
22:  Set  $\mathbf{w} \leftarrow \mathbf{M}^\top(\mathbf{b} - \mathbf{M}\mathbf{x})$ ;
23: end while
24: Output  $\mathbf{x}$ .

```

13.4. NMF via Hierarchical Alternating Nonnegative Least Squares

Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}_+^n$ be two nonnegative vectors. The *univariate NNLS* problem can be formulated as

$$\min_{x>0} \| \mathbf{a} - x\mathbf{b} \|_2^2.$$

This problem admits a closed-form solution: $x = \max\{0, \frac{\mathbf{b}^\top \mathbf{a}}{\|\mathbf{b}\|_2^2}\}$ if $\|\mathbf{b}\|_2 \neq 0$. With this univariate NNLS solution in mind, considering the k -th row of \mathbf{Z} for $k \in \{1, 2, \dots, K\}$, the subproblem in NMF is

$$\min_{Z[k,:] \geq 0} \left\| \underbrace{\left(A - \sum_{p \neq k}^K W[:,p] Z[p,:]\right) - W[:,k] Z[k,:]}_{=A_k} \right\|_F^2, \quad \forall k, \quad (13.4)$$

which indicates the entries in a row of \mathbf{Z} do not interact (similarly, entries in a column of \mathbf{W} do not interact). Therefore, the optimization of each entry in a row of \mathbf{Z} can be decoupled. Let $\mathbf{A}_k = (\mathbf{A} - \sum_{p \neq k}^K \mathbf{W}[:, p] \mathbf{Z}[p, :])$. Then, the NMF problem becomes a set of rank-one

3. This subproblem is convex and is L -Lipschitz gradient continuous/ L -strongly smooth (definition in Problem 12.21); see Problem 13.1.

updates on \mathbf{A}_k , for $k \in \{1, 2, \dots, K\}$. The solution is

$$\mathbf{Z}^*[k, :] = \arg \min_{\mathbf{Z}[k, :] \geq \mathbf{0}} \|\mathbf{A}_k - \mathbf{W}[:, k] \mathbf{Z}[k, :]\|_F^2 = \max \left(\mathbf{0}, \frac{\mathbf{W}[:, k]^\top \mathbf{A}_k}{\|\mathbf{W}[:, k]\|_2^2} \right), \quad \forall k,$$

where the max operator is applied componentwise. This derivation leads to the *hierarchical ANLS (Hi-ANLS)* solution for NMF problems, which iteratively solves a univariate NNLS problem. The procedure is described in Algorithm 21, where we note that $\mathbf{Z}[k, :]^\top = \mathbf{Z}^\top[:, k]$. In the algorithm, we update the k -th row of \mathbf{Z} and k -th column of \mathbf{W} in an interleaved manner. Gillis and Glineur (2012) show that updating \mathbf{Z} several times before updating \mathbf{W} can significantly improve the performance since this reuses the results of $\mathbf{W}^\top \mathbf{A}$ and $\mathbf{W}^\top \mathbf{W}$.

Algorithm 21 NMF via Hierarchical Alternating Nonnegative Least Squares (Hi-ANLS)

Require: Matrix $\mathbf{A} \in \mathbb{R}_+^{M \times N}$;

```

1: Initialize  $\mathbf{W} \in \mathbb{R}_{++}^{M \times K}$ ,  $\mathbf{Z} \in \mathbb{R}_{++}^{K \times N}$  randomly with positive entries;
2: Choose a stop criterion on the approximation error  $\delta$ ;
3: Choose maximal number of iterations  $C$ ;
4:  $iter = 0$ ; ▷ Count for the number of iterations
5: while  $\|\mathbf{A} - (\mathbf{W}\mathbf{Z})\|_F^2 > \delta$  and  $iter < C$  do
6:    $iter = iter + 1$ ;
7:   for  $k = 1$  to  $K$  do
8:      $\mathbf{Z}[k, :] \leftarrow \max \left( \mathbf{0}, \frac{\mathbf{W}[:, k]^\top \mathbf{A}_k}{\|\mathbf{W}[:, k]\|_2^2} \right)$ ; ▷  $\mathbf{A}_k = (\mathbf{A} - \sum_{p \neq k}^K \mathbf{W}[:, p] \mathbf{Z}[p, :])$ 
9:      $\mathbf{W}[:, k] \leftarrow \max \left( \mathbf{0}, \frac{\mathbf{A}_k \mathbf{Z}[k, :]^\top}{\|\mathbf{Z}[k, :]\|_2^2} \right)$ ;
10:  end for
11: end while
12: Output  $\mathbf{W}, \mathbf{Z}$ .

```

13.5. NMF via Alternating Direction Methods of Multipliers (ADMM)

We briefly introduce the *alternating direction methods of multipliers (ADMM)* method and then discuss its applications in matrix factorization and NMF.

ADMM. ADMM is designed to solve convex optimization problems of the form:

$$\min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}), \quad \text{s.t.} \quad \mathbf{Dx} + \mathbf{Ez} = \mathbf{f}. \quad (13.5)$$

Given a penalty parameter $\rho > 0$, the *augmented Lagrangian* of (13.5) is

$$L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{l}) = f(\mathbf{x}) + g(\mathbf{z}) + \langle \mathbf{l}, \mathbf{Dx} + \mathbf{Ez} - \mathbf{f} \rangle + \frac{\rho}{2} \|\mathbf{Dx} + \mathbf{Ez} - \mathbf{f}\|_2^2. \quad (13.6)$$

When $\rho = 0$, the augmented Lagrangian function reduces to the Lagrangian function; when $\rho > 0$, the augmented Lagrangian function acts as a penalized version of the Lagrangian function. The *augmented Lagrangian method* solves the problem iteratively. At the $(t+1)$ -th iteration, it performs the following updates:

$$\text{augmented Lagrangian: } \begin{cases} (\mathbf{x}^{(t+1)}, \mathbf{z}^{(t+1)}) \in \arg \min_{\mathbf{x}, \mathbf{z}} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{l}); \\ \mathbf{l}^{(t+1)} = \mathbf{l}^{(t)} + \rho(\mathbf{Dx}^{(t+1)} + \mathbf{Ez}^{(t+1)} - \mathbf{f}), \end{cases}$$

where the update on $\mathbf{l}^{(t+1)}$ is derived from the *conjugate subgradient theorem* (see, for example, Bach et al. (2011)), and the symbol ‘ \in ’ indicates that the minimum points may not be uniquely determined. One source of difficulty is the coupling term between the \mathbf{x} and \mathbf{z} variables, which is of the form $\rho(\mathbf{x}^\top \mathbf{D}^\top \mathbf{E} \mathbf{z})$. ADMM tackles this difficulty by replacing the exact minimization of (\mathbf{x}, \mathbf{z}) with one iteration of the alternating minimization method. To be more specific, at the $(t + 1)$ -iteration, ADMM performs the following updates:

$$\text{ADMM: } \begin{cases} \mathbf{x}^{(t+1)} \in \arg \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \frac{\rho}{2} \left\| \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{z}^{(t)} - \mathbf{f} + \frac{1}{\rho} \mathbf{l}^{(t)} \right\|_2^2 \right\}; \\ \mathbf{z}^{(t+1)} \in \arg \min_{\mathbf{z}} \left\{ g(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{D}\mathbf{x}^{(t+1)} + \mathbf{E}\mathbf{z} - \mathbf{f} + \frac{1}{\rho} \mathbf{l}^{(t)} \right\|_2^2 \right\}; \\ \mathbf{l}^{(t+1)} = \mathbf{l}^{(t)} + \rho(\mathbf{D}\mathbf{x}^{(t+1)} + \mathbf{E}\mathbf{z}^{(t+1)} - \mathbf{f}). \end{cases} \quad (13.7)$$

By defining $\tilde{\mathbf{l}} = \frac{1}{\rho} \mathbf{l}$, this can be equivalently stated as (this form will be used in the sequel):

$$\text{ADMM: } \begin{cases} \mathbf{x}^{(t+1)} \in \arg \min_{\mathbf{x}} \left\{ f(\mathbf{x}) + \frac{\rho}{2} \left\| \mathbf{D}\mathbf{x} + \mathbf{E}\mathbf{z}^{(t)} - \mathbf{f} + \tilde{\mathbf{l}}^{(t)} \right\|_2^2 \right\}; \\ \mathbf{z}^{(t+1)} \in \arg \min_{\mathbf{z}} \left\{ g(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{D}\mathbf{x}^{(t+1)} + \mathbf{E}\mathbf{z} - \mathbf{f} + \tilde{\mathbf{l}}^{(t)} \right\|_2^2 \right\}; \\ \tilde{\mathbf{l}}^{(t+1)} = \tilde{\mathbf{l}}^{(t)} + (\mathbf{D}\mathbf{x}^{(t+1)} + \mathbf{E}\mathbf{z}^{(t+1)} - \mathbf{f}). \end{cases} \quad (13.8)$$

That is, ADMM alternately updates \mathbf{x} , \mathbf{z} , and \mathbf{l} (or the scaled dual variable $\tilde{\mathbf{l}}$).

ADMM applied to matrix factorization. We return to the problem discussed in ALS (Equation (12.10), i.e., matrix factorization with Frobenius norm; not necessarily a NMF problem), along with a regularization function $r(\mathbf{Z})$:

$$\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{A} - \mathbf{W}\mathbf{Z}\|_F^2 + r(\mathbf{Z}).$$

The problem can be equivalently stated with an auxiliary variable $\tilde{\mathbf{Z}} \in \mathbb{R}^{K \times N}$:

$$\min_{\mathbf{Z}} \frac{1}{2} \|\mathbf{A} - \mathbf{W}\mathbf{Z}\|_F^2 + r(\tilde{\mathbf{Z}}), \quad \text{s.t.} \quad \mathbf{Z} = \tilde{\mathbf{Z}}. \quad (13.9)$$

Following (13.8), let a. $\{\mathbf{x} \leftarrow \mathbf{Z}, \mathbf{z} \leftarrow \tilde{\mathbf{Z}}, \tilde{\mathbf{l}} \leftarrow \mathbf{L}, \mathbf{D} = -\mathbf{I}, \mathbf{E} = \mathbf{I}\}$ or b. $\{\mathbf{x} \leftarrow \mathbf{Z}, \mathbf{z} \leftarrow \tilde{\mathbf{Z}}, \tilde{\mathbf{l}} \leftarrow \mathbf{L}, \mathbf{D} = \mathbf{I}, \mathbf{E} = -\mathbf{I}\}$, the resulting ADMM updates for (13.9) are:

$$\begin{cases} \mathbf{Z} \xleftarrow{(a)} (\mathbf{W}^\top \mathbf{W} + \rho \mathbf{I})^{-1} [\mathbf{W}^\top \mathbf{A} + \rho(\tilde{\mathbf{Z}} + \mathbf{L})] & \xleftarrow{(b)} (\mathbf{W}^\top \mathbf{W} + \rho \mathbf{I})^{-1} [\mathbf{W}^\top \mathbf{A} + \rho(\tilde{\mathbf{Z}} - \mathbf{L})]; \\ \tilde{\mathbf{Z}} \xleftarrow{(a)} \arg \min_{\tilde{\mathbf{Z}}} r(\tilde{\mathbf{Z}}) + \frac{\rho}{2} \left\| -\mathbf{Z} + \tilde{\mathbf{Z}} + \mathbf{L} \right\|_F^2 & \xleftarrow{(b)} \arg \min_{\tilde{\mathbf{Z}}} r(\tilde{\mathbf{Z}}) + \frac{\rho}{2} \left\| \mathbf{Z} - \tilde{\mathbf{Z}} + \mathbf{L} \right\|_F^2 \\ \mathbf{L} \xleftarrow{(a)} \mathbf{L} - \mathbf{Z} + \tilde{\mathbf{Z}} & \xleftarrow{(b)} \mathbf{L} + \mathbf{Z} - \tilde{\mathbf{Z}}. \end{cases} \quad (13.10)$$

In practice, the Cholesky decomposition of $(\mathbf{W}^\top \mathbf{W} + \rho \mathbf{I})$ can be calculated such that the update can be obtained by forward and backward substitutions. The update for \mathbf{W} can be obtained similarly due to symmetry. In the following discussion, we adopt setting (a) from (13.10).

ADMM applied to ℓ_1 regularization. We may also consider the ℓ_1 regularization (see Section 12.3): $r(\tilde{\mathbf{Z}}) = \lambda \|\tilde{\mathbf{Z}}\|_1$. The update for each element (k, n) of $\tilde{\mathbf{Z}}$ is $\tilde{z}_{kn} \leftarrow \max(0, 1 - \frac{\lambda}{\rho} |h_{kn}|^{-1})h_{kn}$ for all $k \in \{1, 2, \dots, K\}$ and $n \in \{1, 2, \dots, N\}$, where $h_{kn} = z_{kn} - l_{kn}$ (i.e., the elements of $\mathbf{H} = \mathbf{Z} - \mathbf{L}$).

ADMM applied to smoothness/denoising regularization. A smoothness regularization on \mathbf{Z} can be defined as $r(\tilde{\mathbf{Z}}) = \frac{\lambda}{2} \|\mathbf{T}\tilde{\mathbf{Z}}^\top\|_F^2$, where \mathbf{T} is an $N \times N$ tridiagonal matrix with 2 on the main diagonal and -1 on the superdiagonal and subdiagonal. This regularization ensures the proximal components in each row of $\tilde{\mathbf{Z}}$ is smooth (see Problem 12.9). The update for $\tilde{\mathbf{Z}}$ becomes $\tilde{\mathbf{Z}} \leftarrow \rho \mathbf{Z} (\lambda \mathbf{T}^\top \mathbf{T} + \rho \mathbf{I})^{-1}$ (Huang et al., 2016).

ADMM applied to NMF. The NMF with ADMM is achieved simply by replacing $r(\mathbf{Z})$ with an indicator function. The update for $\tilde{\mathbf{Z}}$ becomes $\max(\mathbf{0}, \mathbf{Z} - \mathbf{L})$, where the max operator is applied componentwise. However, unlike the methods discussed earlier (such as NNLS) or the MU approach introduced in the next section, ADMM updates are generally not monotonically nonincreasing in terms of the objective function. This is an important consideration when monitoring convergence.

13.6. NMF via Multiplicative Update (MU)

We consider an alternative alternating update approach for NMF. The hidden features in \mathbf{W} and \mathbf{Z} are modeled as nonnegative vectors in a low-dimensional space. These latent vectors are randomly initialized and iteratively updated via an alternating *multiplicative update* rule to minimize the Frobenius norm distance between the observed and modeled matrices. Following Section 12.2, we consider the low-rank with K components; given $\mathbf{W} \in \mathbb{R}_+^{M \times K}$, we aim to update $\mathbf{Z} \in \mathbb{R}_+^{K \times N}$. The gradient of the loss function $L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \|\mathbf{A} - \mathbf{WZ}\|_F^2$ with respect to \mathbf{Z} is given by Equation (12.11): $\nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}) = \mathbf{W}^\top (\mathbf{WZ} - \mathbf{A}) \in \mathbb{R}^{K \times N}$. Applying the gradient descent idea discussed in Section 12.6, a straightforward update for \mathbf{Z} is:

$$(\text{GD on } \mathbf{Z}) \quad \mathbf{Z} \leftarrow \mathbf{Z} - \eta (\nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z})) = \mathbf{Z} - \eta \nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}),$$

where η represents a small positive step size (learning rate).

Multiplicative update (MU). If we allow a different step size for each entry of \mathbf{Z} , the update can be written as:

$$(\text{GD}' \text{ on } \mathbf{Z}) \quad z_{kn} \leftarrow z_{kn} - \eta_{kn} (\nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}))_{kn} = z_{kn} - \eta_{kn} (\mathbf{W}^\top \mathbf{WZ} - \mathbf{W}^\top \mathbf{A})_{kn}, \quad \forall k, n,$$

where z_{kn} denotes the (k, n) -th entry of \mathbf{Z} . To proceed, we further rescale the step size:

$$\eta_{kn} = \frac{z_{kn}}{(\mathbf{W}^\top \mathbf{WZ})_{kn}}.$$

Then we obtain the update rule:

$$(\text{MU on } \mathbf{Z}) \quad \mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{[\mathbf{W}^\top \mathbf{A}]}{[\mathbf{W}^\top \mathbf{WZ}]} \stackrel{*}{=} \mathbf{Z} - \frac{[\mathbf{Z}]}{[\mathbf{W}^\top \mathbf{WZ}]} \circledast \nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}), \quad (13.11)$$

where $\frac{[\cdot]}{[\cdot]}$ represents the componentwise division between two matrices. This is known as the *multiplicative update (MU)*, and is first developed in Lee and Seung (2001) for NMF problems. Analogously, the multiplicative update for \mathbf{W} can be obtained by

$$(\text{MU on } \mathbf{W}) \quad \mathbf{W} \leftarrow \mathbf{W} \circledast \frac{[\mathbf{AZ}^\top]}{[\mathbf{WZZ}^\top]} \stackrel{*}{=} \mathbf{W} - \frac{[\mathbf{W}]}{[\mathbf{WZZ}^\top]} \circledast \nabla_{\mathbf{W}} L(\mathbf{W}, \mathbf{Z}). \quad (13.12)$$

The factors $\frac{(\mathbf{W}^\top \mathbf{A})_{kn}}{(\mathbf{W}^\top \mathbf{WZ})_{kn}}$ and $\frac{(\mathbf{AZ}^\top)_{mk}}{(\mathbf{WZZ}^\top)_{mk}}$ for all m, k, n in (13.11) and (13.12) are called *multiplicative factors*. When $\mathbf{A} = \mathbf{WZ}$, these multiplicative factors reduce to one, indicating that the corresponding gradients vanish.

MU vs gradient descent. The above derivation shows that multiplicative update algorithms are fundamentally similar to gradient descent algorithms, differing primarily in step size selection. With an appropriate choice of step size, the multiplicative algorithm can transform the subtraction update rule of the standard gradient descent method into a multiplicative update rule.

In the gradient descent algorithm, a fixed or adaptive step length is typically used, and this step length is independent of the specific variable being updated. In other words, the step size may vary over time, but at any given update step, all entries of the matrix variable are updated using the same step size. In contrast, the multiplicative algorithm uses different step sizes (η_{kn} above) for different entries of the factor matrix. This means that the step length is adaptive to each matrix entry. This adaptability is a key reason why the multiplicative algorithm can outperform the gradient descent algorithm in NMF algorithms.

KKT conditions for NMF with Frobenius norm. The KKT conditions indicate that (see derivation in (13.3)):

$$\begin{aligned} \mathbf{Z} \geq \mathbf{0}, \quad \nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}) &\geq \mathbf{0}, \quad \langle \mathbf{Z}, \nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}) \rangle = \mathbf{0}_{K \times N}; \\ \mathbf{W} \geq \mathbf{0}, \quad \nabla_{\mathbf{W}} L(\mathbf{W}, \mathbf{Z}) &\geq \mathbf{0}, \quad \langle \mathbf{W}, \nabla_{\mathbf{W}} L(\mathbf{W}, \mathbf{Z}) \rangle = \mathbf{0}_{M \times K}. \end{aligned} \quad (13.13)$$

This also implies

$$\min\{\mathbf{Z}, \nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z})\} = \mathbf{0}_{K \times N} \quad \text{and} \quad \min\{\mathbf{W}, \nabla_{\mathbf{W}} L(\mathbf{W}, \mathbf{Z})\} = \mathbf{0}_{M \times K}, \quad (13.14)$$

where the min operator $\min\{\cdot, \cdot\}$ is applied componentwise. Any pair (\mathbf{W}, \mathbf{Z}) satisfying the KKT conditions is a stationary point of the NMF problem in (13.1).

Problems in MU. The equality $(*)$ in (13.11) indicates a rescaled gradient descent update in the MU rules, which also implies

$$\frac{[\mathbf{W}^\top \mathbf{A}]_{kn}}{[\mathbf{W}^\top \mathbf{WZ}]_{kn}} \geq 1 \iff (\nabla_{\mathbf{Z}} L(\mathbf{W}, \mathbf{Z}))_{kn} \leq 0, \quad \forall k, n.$$

Therefore, the MU algorithm induces three-fold rules: (i) Increase if its partial derivative is negative; (ii) Decrease it if its partial derivative is positive; (iii) Keep it unchanged if its partial derivative is zero. However, if an element of \mathbf{Z} is equal to zero, the MU rule cannot modify it. Therefore, it is possible for an entry of \mathbf{Z} to be zero while its partial derivative is negative, which would violate the KKT conditions in (13.13). As a result, the iterates from the MU rule are not guaranteed to converge to a stationary point. There are several ways to address this issue, such as: (i) Using a small positive lower bound for the entries of \mathbf{Z} and \mathbf{W} (Gillis and Glineur, 2012). (ii) Using the MU rule while reinitializing zero entries of \mathbf{Z} and \mathbf{W} to a small positive constant when their partial derivatives become negative (Chi and Kolda, 2012).

We now prove that the MU rule monotonically decreases the loss function.

Theorem 13.1: (Monotonically nonincreasing of multiplicative update) The loss $L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \|\mathbf{WZ} - \mathbf{A}\|_F^2$ remains nonincreasing under the following multiplicative

update rules: ^a

$$\mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{[\mathbf{W}^\top \mathbf{A}]}{[\mathbf{W}^\top \mathbf{W} \mathbf{Z}]} \quad \text{and} \quad \mathbf{W} \leftarrow \mathbf{W} \circledast \frac{[\mathbf{A} \mathbf{Z}^\top]}{[\mathbf{W} \mathbf{Z} \mathbf{Z}^\top]},$$

where $\mathbf{A} \in \mathbb{R}_+^{M \times N}$, $\mathbf{W} \in \mathbb{R}_+^{M \times K}$, and $\mathbf{Z} \in \mathbb{R}_+^{K \times N}$. The operator $\frac{[\cdot]}{[\cdot]}$ represents the componentwise division between two matrices, and \circledast denotes the Hadamard product between two matrices.

The MU update requires that \mathbf{Z} and \mathbf{W} should be initialized with positive (nonzero) entries; otherwise, zeros will persist due to the multiplicative nature of the update.

^a More general results for β -divergences are discussed in Theorem 13.8.

The MU method sparked significant interest in NMF and has since become a cornerstone in the field, due to several advantages: (i) The update rules are extremely easy to implement; (ii) In practice, the convergence is relatively faster compared to many other methods; (iii) Nonnegativity is automatically preserved during updates. To prove the monotonicity of the MU rules, we use the auxiliary function framework.

Definition 13.2 (Auxiliary function (majorizer)). $G(\mathbf{x}, \tilde{\mathbf{x}})$ is called an *auxiliary function* for $F(\mathbf{x})$ (or a majorizer of F at $\tilde{\mathbf{x}}$) if the conditions ^a

$$G(\mathbf{x}, \tilde{\mathbf{x}}) \geq F(\mathbf{x}) \quad \text{and} \quad G(\mathbf{x}, \mathbf{x}) = F(\mathbf{x})$$

are satisfied. In other words, the auxiliary function $G(\mathbf{x}, \tilde{\mathbf{x}})$ is an upper bound of $F(\mathbf{x})$, and the bound is tight when $\tilde{\mathbf{x}} = \mathbf{x}$.

^a \mathbf{x} can be scalars, vectors, or matrices.

Lemma 13.3: (Nonincreasing in auxiliary functions) If G is an auxiliary function for F , then F is nonincreasing under the update

$$\mathbf{x}^{(t+1)} = \arg \min_{\mathbf{x}} G(\mathbf{x}, \mathbf{x}^{(t)}). \quad (13.15)$$

Proof [of Lemma 13.3] The definition of the auxiliary function indicates that $F(\mathbf{x}^{(t+1)}) \leq G(\mathbf{x}^{(t+1)}, \mathbf{x}^{(t)}) \leq G(\mathbf{x}^{(t)}, \mathbf{x}^{(t)}) = F(\mathbf{x}^{(t)})$. ■

Note that $F(\mathbf{x}^{(t+1)}) = F(\mathbf{x}^{(t)})$ only if $\mathbf{x}^{(t)}$ is a local minimum of $G(\mathbf{x}, \mathbf{x}^{(t)})$ w.r.t. \mathbf{x} . If the partial derivatives of F exist and are continuous in a small neighborhood of $\mathbf{x}^{(t)}$, this also implies that the gradient $\nabla F(\mathbf{x}^{(t)}) = \mathbf{0}$. Thus, by iterating the update in (13.15), we obtain a sequence of estimates that converge to a local minimum $\mathbf{x}_{\min} = \arg \min_{\mathbf{x}} F(\mathbf{x})$ of the objective function:

$$F(\mathbf{x}^{(0)}) \geq F(\mathbf{x}^{(1)}) \geq F(\mathbf{x}^{(2)}) \geq \dots \geq F(\mathbf{x}^{(t)}) \geq F(\mathbf{x}^{(t+1)}) \geq \dots \geq F(\mathbf{x}_{\min}). \quad (13.16)$$

Definition 13.2 finds a majorizer G of F , and Lemma 13.3 shows the minimization property in G , hence the algorithm is often referred to as the *majorization-minimization (MM) framework*. The update benefits when the global minimizer of G has a closed-form solution or can be computed efficiently.

Therefore, if we can construct an appropriate auxiliary function $G(\mathbf{x}, \mathbf{x}^{(t)})$ for both variables in $\|\mathbf{A} - \mathbf{W}\mathbf{Z}\|_F$, the update rules in Theorem 13.1 follow from (13.15). To apply

the auxiliary function method to the NMF problem, we focus on a single column of \mathbf{A} or \mathbf{Z} : $\mathbf{a} = \mathbf{a}_n$ and $\mathbf{z} = \mathbf{z}_n$ in the following lemma, where $n \in \{1, 2, \dots, N\}$.

Lemma 13.4: (Auxiliary function for NMF) Let $\mathbf{W} \in \mathbb{R}^{K \times N}$, $\mathbf{a} \in \mathbb{R}^M$, and $\mathbf{z} \in \mathbb{R}^K$. Let further $\mathbf{D} \in \mathbb{R}^{K \times K}$ be a diagonal matrix with the (k, k) -th entry being $d_{kk} = \frac{(\mathbf{W}^\top \mathbf{W} \mathbf{z})_k}{z_k} = \frac{\mathbf{w}_k^\top \mathbf{W} \mathbf{z}}{z_k} = \frac{\sum_{j=1}^K (\mathbf{W}^\top \mathbf{W})_{kj} z_j}{z_k}$, $\forall k \in \{1, 2, \dots, K\}$, where \mathbf{w}_k is the k -th column of \mathbf{W} and z_k is the k -th component of \mathbf{z} . Then, the following function is an auxiliary function for $F(\mathbf{z}) = \frac{1}{2} \|\mathbf{a} - \mathbf{W}\mathbf{z}\|_2^2$:

$$G(\mathbf{z}, \mathbf{z}^{(t)}) = F(\mathbf{z}^{(t)}) + (\mathbf{z} - \mathbf{z}^{(t)})^\top \nabla F(\mathbf{z}^{(t)}) + \frac{1}{2} (\mathbf{z} - \mathbf{z}^{(t)})^\top \mathbf{D} (\mathbf{z} - \mathbf{z}^{(t)}).$$

Proof [of Lemma 13.4] Since the third-order partial derivatives of $F(\mathbf{z})$ vanish (see Problem 13.3), $F(\mathbf{z})$ can be factored as

$$F(\mathbf{z}) = F(\mathbf{z}^{(t)}) + (\mathbf{z} - \mathbf{z}^{(t)})^\top \nabla F(\mathbf{z}^{(t)}) + \frac{1}{2} (\mathbf{z} - \mathbf{z}^{(t)})^\top \mathbf{W}^\top \mathbf{W} (\mathbf{z} - \mathbf{z}^{(t)}).$$

Apparently, $G(\mathbf{z}, \mathbf{z}) = F(\mathbf{z})$. To complete the proof, we need to show that $G(\mathbf{z}, \mathbf{z}^{(t)}) \geq F(\mathbf{z})$; that is, $\mathbf{D} - \mathbf{W}^\top \mathbf{W}$ is positive semidefinite. To prove this, consider the matrix $\mathbf{M} \in \mathbb{R}^{K \times K}$ whose entries are $m_{ij} = z_i (\mathbf{D} - \mathbf{W}^\top \mathbf{W})_{ij} z_j$ for all $i, j \in \{1, 2, \dots, K\}$, which is a rescaling of the components of $\mathbf{D} - \mathbf{W}^\top \mathbf{W}$. Then $\mathbf{D} - \mathbf{W}^\top \mathbf{W}$ is positive semidefinite if and only if \mathbf{M} is:

$$\begin{aligned} \mathbf{x}^\top \mathbf{M} \mathbf{x} &= \sum_{i,j=1}^{K,K} x_i m_{ij} x_j \stackrel{*}{=} \sum_{i,j=1}^{K,K} \left\{ (\mathbf{W}^\top \mathbf{W})_{ij} z_i z_j x_i^2 - (\mathbf{W}^\top \mathbf{W})_{ij} z_i z_j x_i x_j \right\} \\ &\stackrel{\dagger}{=} \sum_{i,j=1}^{K,K} (\mathbf{W}^\top \mathbf{W})_{ij} z_i z_j \left(\frac{1}{2} x_i^2 + \frac{1}{2} x_j^2 - x_i x_j \right) = \sum_{i,j=1}^{K,K} (\mathbf{W}^\top \mathbf{W})_{ij} z_i z_j \frac{1}{2} (x_i - x_j)^2 \geq 0, \end{aligned}$$

where the equality (\dagger) follows from the symmetry of \mathbf{M} , and the equality ($*$) follows from the diagonality of \mathbf{D} :

$$\sum_{i,j=1}^{K,K} x_i z_i d_{ij} z_j x_j = \sum_{i=1}^K x_i z_i d_{ii} z_i x_i = \sum_{i=1}^K x_i^2 z_i^2 \frac{\sum_{j=1}^K (\mathbf{W}^\top \mathbf{W})_{ij} z_j}{z_i} = \sum_{i,j=1}^{K,K} (\mathbf{W}^\top \mathbf{W})_{ij} z_i z_j x_i^2.$$

This completes the proof. ■

The proof of the monotonicity of MU updates in Theorem 13.1 follows directly from the above lemmas. Clearly, the approximations \mathbf{W} and \mathbf{Z} remain nonnegative during the updates. It is generally better to update \mathbf{W} and \mathbf{Z} “simultaneously” rather than “sequentially,” i.e., updating each matrix completely before the other. In this case, after updating a row of \mathbf{Z} , we update the corresponding column of \mathbf{W} . In the implementation, it is advisable to introduce a small positive quantity, say the square root of the machine precision, to the denominators in the approximations of \mathbf{W} and \mathbf{Z} at each iteration. And a trivial value like $\epsilon = 10^{-9}$ suffices. The full procedure is shown in Algorithm 22. In practice, the algorithm can also be accelerated by updating \mathbf{W} several times before updating \mathbf{Z} , during which process we can reuse the result of $\mathbf{A}\mathbf{Z}^\top$ and $\mathbf{Z}\mathbf{Z}^\top$, and vice versa.

Algorithm 22 NMF via Multiplicative Updates

Require: Matrix $\mathbf{A} \in \mathbb{R}_+^{M \times N}$;

- 1: Initialize $\mathbf{W} \in \mathbb{R}_{++}^{M \times K}$, $\mathbf{Z} \in \mathbb{R}_{++}^{K \times N}$ randomly with positive entries;
- 2: Choose a stop criterion on the approximation error δ ;
- 3: Choose maximal number of iterations C ;
- 4: $iter = 0$; ▷ Count for the number of iterations
- 5: **while** $\|\mathbf{A} - (\mathbf{W}\mathbf{Z})\|_F^2 > \delta$ and $iter < C$ **do**
- 6: $iter = iter + 1$;
- 7: $\mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{[\mathbf{W}^\top \mathbf{A}]}{[\mathbf{W}^\top \mathbf{W}\mathbf{Z}] + \epsilon}$;
- 8: $\mathbf{W} \leftarrow \mathbf{W} \circledast \frac{[\mathbf{A}\mathbf{Z}^\top]}{[\mathbf{W}\mathbf{Z}\mathbf{Z}^\top] + \epsilon}$;
- 9: **end while**
- 10: Output \mathbf{W}, \mathbf{Z} .

13.6.1 Regularization

As mentioned in (13.3), the NNLS or NMF problem implicitly imposes a **sparsity constraint**. On the other hand, similar to the ALS method with regularization discussed in Section 12.3 (recall that the regularization can help extend the applicability of ALS to general matrices), a regularization term can be incorporated into the NMF framework to enhance its performance:

$$L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \|\mathbf{W}\mathbf{Z} - \mathbf{A}\|_F^2 + \frac{1}{2} \lambda_w \|\mathbf{W}\|_F^2 + \frac{1}{2} \lambda_z \|\mathbf{Z}\|_F^2, \quad \lambda_w > 0, \lambda_z > 0,$$

where the employed matrix norm is still the Frobenius norm. The gradient with respect to \mathbf{Z} given \mathbf{W} is the same as that in Equation (12.20):

$$\frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} = \mathbf{W}^\top (\mathbf{W}\mathbf{Z} - \mathbf{A}) + \lambda_z \mathbf{Z} \in \mathbb{R}^{K \times N}.$$

The gradient descent update can be obtained by

$$(\text{GD on } \mathbf{Z}) \quad \mathbf{Z} \leftarrow \mathbf{Z} - \eta \left(\frac{\partial L(\mathbf{Z}|\mathbf{W})}{\partial \mathbf{Z}} \right) = \mathbf{Z} - \eta \left(\mathbf{W}^\top \mathbf{W}\mathbf{Z} - \mathbf{W}^\top \mathbf{A} + \lambda_z \mathbf{Z} \right),$$

Analogously, if we assume a different step size for each entry of \mathbf{Z} , the update can be obtained by

$$(\text{GD}' \text{ on } \mathbf{Z}) \quad z_{kn} = z_{kn} - \eta_{kn} (\mathbf{W}^\top \mathbf{W}\mathbf{Z} - \mathbf{W}^\top \mathbf{A} + \lambda_z \mathbf{Z})_{kn}, \quad \forall k, n.$$

We again rescale the step size: $\eta_{kn} = \frac{z_{kn}}{(\mathbf{W}^\top \mathbf{W}\mathbf{Z})_{kn}}$. Then we obtain the MU rules for \mathbf{Z} and \mathbf{W} (due to symmetry):

$$\mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{[\mathbf{W}^\top \mathbf{A} - \lambda_z \mathbf{Z}]}{[\mathbf{W}^\top \mathbf{W}\mathbf{Z}]} \quad \text{and} \quad \mathbf{W} \leftarrow \mathbf{W} \circledast \frac{[\mathbf{A}\mathbf{Z}^\top - \lambda_w \mathbf{W}]}{[\mathbf{W}\mathbf{Z}\mathbf{Z}^\top]}.$$

Modified MU. Since the update for the above regularized NMF can result in negative values, a modified MU can be applied such that

$$(\text{MMU1}): \quad \mathbf{Z} \leftarrow \left[\mathbf{Z} \circledast \frac{[\mathbf{W}^\top \mathbf{A} - \lambda_z \mathbf{Z}]}{[\mathbf{W}^\top \mathbf{W}\mathbf{Z}]} \right]_+ \quad \text{and} \quad \mathbf{W} \leftarrow \left[\mathbf{W} \circledast \frac{[\mathbf{A}\mathbf{Z}^\top - \lambda_w \mathbf{W}]}{[\mathbf{W}\mathbf{Z}\mathbf{Z}^\top]} \right]_+,$$

where $[x]_+ = \max\{x, \epsilon\}$. The parameter ϵ is usually a very small positive number that prevents the emergence of negative update. That is, we add a small lower bound for entries of \mathbf{W} and \mathbf{Z} . An alternative applies the nonnegativity constraint only to the numerator:

$$(\text{MMU2}): \quad \mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{[\mathbf{W}^\top \mathbf{A} - \lambda_z \mathbf{Z}]_+}{[\mathbf{W}^\top \mathbf{W} \mathbf{Z}]} \quad \text{and} \quad \mathbf{W} \leftarrow \mathbf{W} \circledast \frac{[\mathbf{A} \mathbf{Z}^\top - \lambda_w \mathbf{W}]_+}{[\mathbf{W} \mathbf{Z} \mathbf{Z}^\top]}.$$

13.7. NMF with Three Factors

The NMF method, when extended to incorporate three factor matrices, is referred to as *nonnegative matrix trifactorization (tri-NMF)*. This approach introduces an additional factor:

$$\mathbf{A} \approx \mathbf{W} \mathbf{U} \mathbf{Z}, \quad (13.17)$$

where $\mathbf{W} \in \mathbb{R}_+^{M \times K}$, $\mathbf{U} \in \mathbb{R}_+^{K \times J}$, and $\mathbf{Z} \in \mathbb{R}_+^{J \times N}$. Consider the item-by-user matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, where each element is a binary number $\{0, 1\}$. This type of data is referred to as *implicit feedback*, in contrast to *explicit feedback* (such as numerical ratings) used in other contexts.⁴ Standard NMF on this matrix provides a sum of K rank-one matrices $\mathbf{A} \approx \sum_{k=1}^K \mathbf{W}[:, k] \mathbf{Z}[k, :]$. In the context of implicit data, each rank-one matrix can be interpreted as finding a subset of users and a subset of items (e.g., movies) that interact strongly with each other. In contrast, tri-NMF yields the following approximation:

$$\mathbf{A} \approx \sum_{k=1}^K \sum_{j=1}^J \mathbf{W}[:, k] \mathbf{U}[k, j] \mathbf{Z}[j, :].$$

This formulation can be interpreted as identifying separately J subsets of movies that are watched together (the rows of \mathbf{Z}) and K subset of users that behave similarly (the columns of \mathbf{W}); while the matrix \mathbf{U} tells us how these subsets interact together. If $u_{kj} > 0$, then the k -th subset of users (corresponding to the positive entries of $\mathbf{W}[:, k]$) watches the movies from the j -th subset of movies (corresponding to the positive entries of $\mathbf{Z}[j, :]$).

In other words, tri-NMF identifies groups of users who exhibit similar behavior (by watching the same movies) and groups of movies that are similar (because they are watched by the same users), while connecting these groups through the nonnegative interaction matrix \mathbf{U} . This model is also applicable in text mining, where it can identify groups of documents that contain similar sets of words (columns of \mathbf{W}) and groups of words that commonly appear together in the same documents (rows of \mathbf{Z}), with \mathbf{U} encoding the relationships between these groups (Brouwer et al., 2017; Gillis, 2020; Lu, 2023).

13.8. β -Divergence, Alternative Perspectives of MU

As mentioned previously, the sum of squared loss, as given in (12.10) or (13.1), is convex when one of the factors is held constant, leading to a smooth optimization process. This type of loss function falls under a broader class of distance/divergence estimators known as β -divergence in the context of NMF. Given two nonnegative scalars x and y , the β -divergence

⁴. For example, in datasets like Netflix or MovieLens, ratings above 4 can be mapped to 1, while ratings below 1 can be mapped to 0 to obtain an implicit data set.

between x and y is defined as follows:

$$d_\beta(x, y) = \begin{cases} \frac{x}{y} - \ln \frac{x}{y} - 1, & \text{if } \beta = 0; \\ x \ln \frac{x}{y} - x + y, & \text{if } \beta = 1; \\ \frac{1}{\beta^2 - \beta} (x^\beta + (\beta - 1)y^\beta - \beta xy^{\beta-1}), & \text{otherwise.} \end{cases} \quad (13.18)$$

The β -divergence is continuous in β since $\lim_{\beta \rightarrow 0} (x^\beta - y^\beta)/\beta = \ln(x/y)$. When $\beta = 0, 1$, and 2 , the β -divergences are also known as the *Itakura-Saito (IS)*, *KL*, and *Frobenius/Euclidean distances/divergences*, respectively. The β -divergence between two matrices \mathbf{B} and \mathbf{C} is

$$D_\beta(\mathbf{B}, \mathbf{C}) = \sum_j d_\beta(\mathbf{b}_j, \mathbf{c}_j) = \sum_{i,j} d_\beta(b_{ij}, c_{ij}). \quad (13.19)$$

The analysis of β -divergence is complex. When the first argument is fixed at 1, smaller values are less penalized as the β value increases; however, when the first argument is 2, smaller values are more penalized as the β value increases. In both cases, larger values are more heavily penalized as the β value increases. See Figure 13.1.

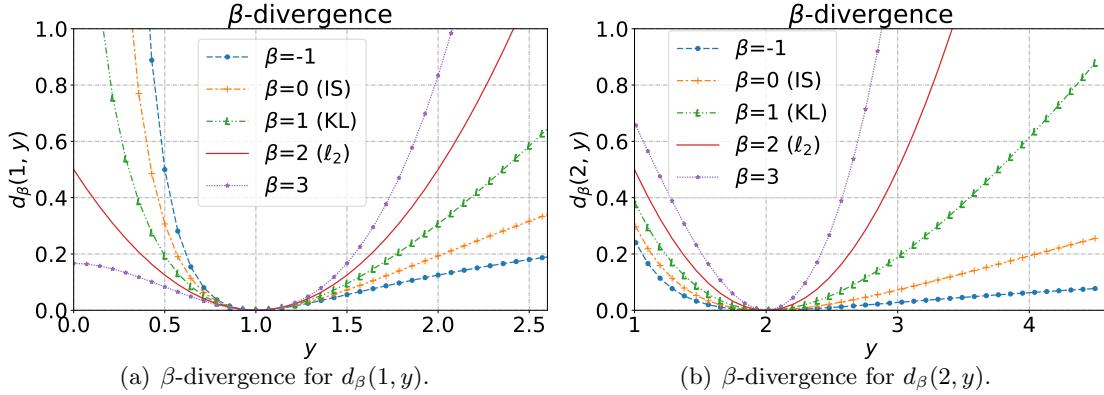


Figure 13.1: The analysis of β -divergence is complex. When the first argument is fixed at 1, smaller values are less penalized as the β value increases; however, when the first argument is 2, smaller values are more penalized as the β value increases. In both cases, larger values are more heavily penalized as the β value increases.

Convexity of β -divergence. When $\beta \in [1, 2]$, the function $d_\beta(x, y)$ is convex in the second argument y . This implies $D_\beta(\mathbf{A}, \mathbf{WZ})$ is convex in \mathbf{W} when fixing \mathbf{Z} , and vice versa (Problem 12.5). Therefore, coordinate descent algorithms can be effectively applied for NMF using the β -divergence.

Scaling in arguments. Let $\gamma > 0$ be a scale factor, then

$$d_\beta(\gamma x, \gamma y) = \gamma^\beta d_\beta(x, y). \quad (13.20)$$

This indicates that the larger the β , the more sensitive the β -divergence is to large values of x or y ; on the contrary, β -divergence with small $\beta < 0$ values relies more heavily on the smallest data values. However, when $\beta = 0$ (called *Itakura-Saito divergence, IS divergence*), the β -divergence is not related to the β value, and thus it is invariant to scaling. What matters is the ratio between x and y ; see (13.18).

Gradient. Since we consider a nonnegative matrix \mathbf{A} for NMF problems, we should note that $d_\beta(x, \cdot)$ for $x = 0$ is not defined for all values of β :

$$d_\beta(0, y) = \begin{cases} \text{not defined,} & \text{if } \beta \leq 0; \\ \frac{1}{\beta} y^\beta, & \text{if } \beta > 0, \end{cases} \implies d'_\beta(0, y) = \begin{cases} \text{not defined,} & \text{if } \beta \leq 0; \\ y^{\beta-1}, & \text{if } \beta > 0, \end{cases}$$

where the derivative $d'_\beta(0, y)$ corresponds to the second variable y . Therefore, the algorithm developed in the following sections can only be applied to positive matrices when $\beta \leq 0$. Table 13.1 and Table 13.2 present the domains of $d_\beta(x, \cdot)$ and $d'_\beta(x, \cdot)$, respectively, for different values of β and x .

Table 13.1: Domain of $d_\beta(x, \cdot)$.

	$\beta \leq 0$	$\beta \in (0, 1]$	$\beta > 1$		$\beta \leq 0$	$\beta \in (0, 1)$	$\beta \in [1, 2]$	$\beta \geq 2$
$x = 0$	\emptyset	\mathbb{R}_+	\mathbb{R}_+	$x = 0$	\emptyset	\mathbb{R}_{++}	\mathbb{R}_+	\mathbb{R}_+
$x > 0$	\mathbb{R}_{++}	\mathbb{R}_{++}	\mathbb{R}_+	$x > 0$	\mathbb{R}_{++}	\mathbb{R}_{++}	\mathbb{R}_{++}	\mathbb{R}_+

Table 13.2: Domain of $d'_\beta(x, \cdot)$.

On the other hand, the gradients of $D_\beta(\mathbf{A}, \mathbf{WZ})$ w.r.t. \mathbf{Z} and \mathbf{W} are given by (if exist)

$$\begin{aligned} \nabla_{\mathbf{Z}} D_\beta(\mathbf{A}, \mathbf{WZ}) &= \mathbf{W}^\top ((\mathbf{WZ})^{\beta-2} \circledast (\mathbf{WZ} - \mathbf{A})); \\ \nabla_{\mathbf{W}} D_\beta(\mathbf{A}, \mathbf{WZ}) &= ((\mathbf{WZ})^{\beta-2} \circledast (\mathbf{WZ} - \mathbf{A})) \mathbf{Z}^\top, \end{aligned} \quad (13.21)$$

where $(\mathbf{WZ})^{\beta-2}$ denotes the componentwise exponent. When $\beta = 2$, the gradient reduces to the one in (12.11) and (12.16).

Decomposition of β -divergence. The β -divergence can be divided into three parts: convex, concave, and constant terms. We should note that this decomposition is not unique since any affine term is both convex and concave. We follow the convention in Févotte and Idier (2011):

$$d_\beta(x, y) = \check{d}_\beta(x, y) + \hat{d}_\beta(x, y) + \bar{d}_\beta(x, y), \quad (13.22)$$

where $\check{d}_\beta(x, y)$ is convex in y , $\hat{d}_\beta(x, y)$ is concave in y , and $\bar{d}_\beta(x, y)$ is constant in y ; see Table 13.3 for different β values.

	$\check{d}_\beta(x, y)/\check{d}'_\beta(x, y)$, convex	$\hat{d}_\beta(x, y)/\hat{d}'_\beta(x, y)$, concave	$\bar{d}_\beta(x, y)$, constant
$\beta < 1, \beta \neq 0$	$-\frac{1}{\beta-1} xy^{\beta-1} / -xy^{\beta-2}$	$\frac{1}{\beta} y^\beta / y^{\beta-1}$	$\frac{1}{\beta(\beta-1)} x^\beta$
$\beta = 0$	$xy^{-1} / -xy^{-2}$	$\ln y / y^{-1}$	$x(\ln x - 1)$
$1 \leq \beta \leq 2$	$d_\beta(x, y)/d'_\beta(x, y)$	$0/0$	0
$\beta > 2$	$\frac{1}{\beta} y^\beta / y^{\beta-1}$	$-\frac{1}{\beta-1} xy^{\beta-1}$	$\frac{1}{\beta(\beta-1)} x^\beta$

Table 13.3: Scalar convex-concave-constant decomposition of $d_\beta(x, y)$ with respect to the second variable y , and the corresponding derivatives with respect to the second variable y .

KKT conditions for NMF with β -divergence. The KKT conditions indicate that (see derivation in (13.3)):

$$\begin{aligned} \mathbf{Z} \geq \mathbf{0}, \quad \nabla_{\mathbf{Z}} D_\beta(\mathbf{A}, \mathbf{WZ}) &\geq \mathbf{0}, \quad \langle \mathbf{Z}, \nabla_{\mathbf{Z}} D_\beta(\mathbf{A}, \mathbf{WZ}) \rangle = \mathbf{0}_{K \times N}; \\ \mathbf{W} \geq \mathbf{0}, \quad \nabla_{\mathbf{W}} D_\beta(\mathbf{A}, \mathbf{WZ}) &\geq \mathbf{0}, \quad \langle \mathbf{W}, \nabla_{\mathbf{W}} D_\beta(\mathbf{A}, \mathbf{WZ}) \rangle = \mathbf{0}_{M \times K}. \end{aligned} \quad (13.23)$$

This also implies

$$\min\{\mathbf{Z}, \nabla_{\mathbf{Z}} D_{\beta}(\mathbf{A}, \mathbf{WZ})\} = \mathbf{0}_{K \times N} \quad \text{and} \quad \min\{\mathbf{W}, \nabla_{\mathbf{W}} D_{\beta}(\mathbf{A}, \mathbf{WZ})\} = \mathbf{0}_{M \times K}, \quad (13.24)$$

where the min operator $\min\{\cdot, \cdot\}$ is applied componentwise.

13.8.1 MU for β -Divergence Obtained by Gradient Ratio Heuristic

We have shown that the MU update for the Frobenius norm can be derived from rescaled gradient descent. For brevity, let $\nabla_{\mathbf{Z}} = \nabla_{\mathbf{Z}} D_{\beta}(\mathbf{A}, \mathbf{WZ}) = \nabla_{\mathbf{Z}}^+ - \nabla_{\mathbf{Z}}^-$, where

$$\nabla_{\mathbf{Z}}^+ = \mathbf{W}^\top ((\mathbf{WZ})^{\beta-1}) \quad \text{and} \quad \nabla_{\mathbf{Z}}^- = \mathbf{W}^\top ((\mathbf{WZ})^{\beta-2} \circledast \mathbf{A}). \quad (13.25)$$

When $z_{kn} > 0, \forall k, n$, the KKT conditions show that $(\nabla_{\mathbf{Z}}^+)_k = (\nabla_{\mathbf{Z}}^-)_k$. The rule from gradient descent (i.e., $\mathbf{Z}^{(t+1)} = \mathbf{Z}^{(t)} - \eta \nabla_{\mathbf{Z}}$) indicates a small decrease (resp., increase) of z_{kn} will lead to a decrease of the loss function if $(\nabla_{\mathbf{Z}})_k > 0$ (resp., < 0). Therefore, it is reasonable to update z_{kn} using the componentwise ratio between $\nabla_{\mathbf{Z}}^-$ and $\nabla_{\mathbf{Z}}^+$:

$$\mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{[\nabla_{\mathbf{Z}}^-]}{[\nabla_{\mathbf{Z}}^+]}, \quad (13.26)$$

where $\frac{[\cdot]}{[\cdot]}$ represents the componentwise ratio of two matrices. This update rule also corresponds to a multiplicative update (MU). When $\beta = 2$, the MU algorithm reduces to the case in Theorem 13.1. When $\beta = 1$, the loss function becomes the KL divergence, and the update for \mathbf{Z} is

$$(\beta = 1) : \quad \mathbf{Z} \leftarrow \mathbf{Z} \circledast \frac{[\mathbf{W}^\top \frac{[\mathbf{A}]}{[\mathbf{WZ}]})]}{[\mathbf{W}^\top \mathbf{1}_{M \times N}]}.$$

It can be shown that when $\beta \in [1, 2]$, the MU algorithms derived for β -divergence will monotonically decrease $D_{\beta}(\mathbf{A}, \mathbf{WZ})$.

13.8.2 MU for β -Divergence Obtained by Rescaled PGD

As discussed in Section 13.2, the PGD approach involves projecting the gradient descent update onto the feasible set (Lu, 2025). Consider a standard GD update on $f(\mathbf{x})$: $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \eta \nabla f(\mathbf{x}^{(t)})$, where η is a step size and $-\nabla f(\mathbf{x}^{(t)})$ is a *descent direction* (\mathbf{g} is a descent direction if $\mathbf{g}^\top \nabla f(\mathbf{x}^{(t)}) < 0$). Consider further a diagonal \mathbf{D} such that $-\eta \nabla f(\mathbf{x}^{(t)}) \rightarrow -\mathbf{D} \nabla f(\mathbf{x}^{(t)})$ is also a descent direction (replacing the step size by a diagonal matrix)⁵. In this case, if the feasible set of \mathbf{x} is nonnegative, then the PGD is useful: $\mathbf{x}^{(t+1)} = \mathcal{P}(\mathbf{x}^{(t)} - \mathbf{D} \nabla f(\mathbf{x}^{(t)}))$, where $\mathcal{P}(x) = \max\{x, 0\}$ ⁶. If we further decompose the gradient into positive and negative parts: $\nabla f(\mathbf{x}^{(t)}) = \nabla^+ f(\mathbf{x}^{(t)}) - \nabla^- f(\mathbf{x}^{(t)})$ with $\nabla^+ f(\mathbf{x}^{(t)}) > 0$ and $\nabla^- f(\mathbf{x}^{(t)}) > 0$, taking $\mathbf{D} = \text{diag}(\frac{[\mathbf{x}^{(t)}]}{[\nabla^+ f(\mathbf{x}^{(t)})]})$, the rescaled PGD update becomes a MU rule:

$$\mathbf{x}^{(t+1)} = \mathcal{P}\left(\mathbf{x}^{(t)} - \text{diag}\left(\frac{[\mathbf{x}^{(t)}]}{[\nabla^+ f(\mathbf{x}^{(t)})]}\right) \nabla f(\mathbf{x}^{(t)})\right) = \mathcal{P}\left(\mathbf{x}^{(t)} \circledast \frac{[\nabla^- f(\mathbf{x}^{(t)})]}{[\nabla^+ f(\mathbf{x}^{(t)})]}\right). \quad (13.27)$$

If we use the decomposition of gradient in (13.25), the rescaled PGD becomes the MU update for NMF in (13.26). If we further incorporate a step size η in the rescaled PGD

5. \mathbf{D} can be relaxed to any positive definite matrices.

6. see, for example, Beck (2017).

update, it becomes

$$\mathbf{x}^{(t+1)} = \mathcal{P}\left((1 - \eta)\mathbf{x}^{(t)} + \eta\mathbf{x}^{(t)} \circledast \frac{[\nabla^- f(\mathbf{x}^{(t)})]}{[\nabla^+ f(\mathbf{x}^{(t)})]}\right). \quad (13.28)$$

Since $-\mathbf{D}\nabla f(\mathbf{x}^{(t)})$ is a descent direction, the step size $\eta \in (0, 1)$ can ensure that the update is monotonically nonincreasing. Note that the projection operator can be omitted since all updates are nonnegative.

13.8.3 MU for β -Divergence Obtained by MM Framework

The β -divergence between two matrices can be defined columnwise (Equation (13.19)), and the β -divergence can be divided into three parts (convex, concave, and constant, Equation (13.22)). Thus, the loss function in NMF can be decomposed into (note the loss function can be further divided componentwise):

$$D_\beta(\mathbf{A}, \mathbf{WZ}) = \sum_{n=1}^N d_\beta(\mathbf{a}_n, \mathbf{Wz}_n) = \sum_{n=1}^N \left(\check{d}_\beta(\mathbf{a}_n, \mathbf{Wz}_n) + \hat{d}_\beta(\mathbf{a}_n, \mathbf{Wz}_n) + \bar{d}_\beta(\mathbf{a}_n, \mathbf{Wz}_n) \right).$$

For each column n , the MM framework involves finding auxiliary functions for the three components separately. To see this, we need the following lemma:

Lemma 13.5: (Auxiliary function by parts) Let $F(\mathbf{x}) = \sum_{i=1}^n F_i(\mathbf{x})$, and let $G_i(\mathbf{x}, \tilde{\mathbf{x}})$ be an auxiliary function for $F_i(\mathbf{x})$ at $\tilde{\mathbf{x}}$ for all i . Then, $G(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{i=1}^n G_i(\mathbf{x}, \tilde{\mathbf{x}})$ is an auxiliary function for $F(\mathbf{x})$ at $\tilde{\mathbf{x}}$.

This lemma indicates that if the auxiliary function is constructed separately for each component, it allows us to decouple the optimization.

Constant part. There is no need to find an auxiliary function for the constant term $\bar{d}_\beta(\mathbf{a}_n, \mathbf{Wz}_n)$, since it does not influence the minimization of $d_\beta(\mathbf{a}_n, \mathbf{Wz}_n)$ with respect to \mathbf{z}_n .

Concave part. Any concave function can be upper-bounded using linearization (the tangent plane):

$$\hat{d}_\beta(x, y) \leq \hat{d}_\beta(x, \tilde{y}) + (y - \tilde{y})\hat{d}'_\beta(x, \tilde{y}),$$

where $\hat{d}'_\beta(x, \tilde{y})$ denotes the gradient of $\hat{d}_\beta(x, \tilde{y})$ with respect to the second component \tilde{y} . Therefore, for any $\tilde{\mathbf{z}}_n \in \mathbb{R}^K$, the auxiliary function for the concave component $\hat{d}_\beta(\mathbf{a}_n, \mathbf{Wz}_n)$ can be constructed by

$$\hat{G}(\mathbf{z}_n, \tilde{\mathbf{z}}_n) = \hat{d}_\beta(\mathbf{a}_n, \mathbf{Wz}_n) + (\mathbf{Wz}_n - \mathbf{Wz}_n) \circledast \hat{d}'_\beta(\mathbf{a}_n, \mathbf{Wz}_n).$$

Convex part. The auxiliary function for the convex part follows from the convexity inequality ⁷. Construct a matrix $\mathbf{P} \in \mathbb{R}^{M \times K}$ as follows:

$$p_{mk} = \frac{w_{mk}\tilde{z}_{kn}}{\sum_j w_{mj}\tilde{z}_{jn}} = \frac{w_{mk}\tilde{z}_{kn}}{\mathbf{W}[m, :]\tilde{\mathbf{z}}_n} \implies \mathbf{P} \geq \mathbf{0} \text{ and } \mathbf{P}\mathbf{1} = \mathbf{1}. \quad (13.29)$$

⁷. Let $f : \mathbb{S} \rightarrow \mathbb{R}$ be a convex function, and let $p \geq 2$ be any integer. Then, $f\left(\sum_{i=1}^p \lambda_i \mathbf{x}_i\right) \leq \sum_{i=1}^p \lambda_i f(\mathbf{x}_i)$, if $\lambda_i \geq 0$ and $\sum_{i=1}^p \lambda_i = 1$.

That is, each row of \mathbf{P} belongs to the unit simplex in \mathbb{R}^K . Therefore, we have

$$\begin{aligned}\check{d}_\beta(a_{mn}, \mathbf{W}[m, :] \mathbf{z}_n) &= \check{d}_\beta(a_{mn}, \sum_{k=1}^K w_{mk} z_{kn}) = \check{d}_\beta(a_{mn}, \sum_{k=1}^K p_{mk} \frac{w_{mk} z_{kn}}{p_{mk}}) \\ &\leq \sum_{k=1}^K p_{mk} \check{d}_\beta(a_{mn}, \frac{w_{mk} z_{kn}}{p_{mk}}).\end{aligned}$$

This decomposition finds an auxiliary function for $D_\beta(\mathbf{A}, \mathbf{WZ})$ w.r.t. \mathbf{Z} .

Theorem 13.6: (Auxiliary function for $D_\beta(\mathbf{A}, \mathbf{WZ})$ w.r.t. \mathbf{Z}) Let $\tilde{\mathbf{a}}_n = \mathbf{W}\tilde{\mathbf{z}}_n$ with $\tilde{a}_{mn} = \mathbf{W}[m, :] \tilde{\mathbf{z}}_n$ for all m, n , where $\tilde{\mathbf{z}}_n$ is any vector in \mathbb{R}^K . Then, $G(\mathbf{Z}, \tilde{\mathbf{Z}}) = \sum_{n=1}^N G_n(\mathbf{z}_n, \tilde{\mathbf{z}}_n) = \sum_{n=1}^N \sum_{m=1}^M G_{mn}$ is an auxiliary function for $D_\beta(\mathbf{A}, \mathbf{WZ})$ w.r.t. \mathbf{Z} , where

$$G_{mn} = \bar{d}_\beta(a_{mn}, \tilde{a}_{mn}) + \hat{d}_\beta(a_{mn}, \tilde{a}_{mn}) + \sum_{k=1}^K w_{mk} (z_{kn} - \tilde{z}_{kn}) \hat{d}'_\beta(a_{mn}, \tilde{a}_{mn}) + \sum_{k=1}^K \frac{w_{mk} \tilde{z}_{kn}}{\tilde{a}_{mn}} \check{d}_\beta(a_{mn}, \frac{\tilde{a}_{mn} z_{kn}}{\tilde{z}_{kn}}).$$

Exercise 13.7 (Gradient and Hessian of auxiliary functions). Consider the setting and notations in Theorem 13.6. Let $G_n(\mathbf{z}_n, \tilde{\mathbf{z}}_n) = \sum_{k=1}^K G_k(\mathbf{z}_{kn}, \tilde{\mathbf{z}}_n) + C(\mathbf{z}_n)$ where $C(\mathbf{z}_n)$ is a constant w.r.t. \mathbf{z}_n . That is,

$$G_k(\mathbf{z}_{kn}, \tilde{\mathbf{z}}_n) = \sum_{m=1}^M w_{mk} z_{kn} \hat{d}'_\beta(a_{mn}, \tilde{a}_{mn}) + \sum_{m=1}^M \frac{w_{mk} \tilde{z}_{kn}}{\tilde{a}_{mn}} \check{d}_\beta(a_{mn}, \frac{\tilde{a}_{mn} z_{kn}}{\tilde{z}_{kn}}).$$

Show that the gradient of the auxiliary function is

$$\nabla_{z_{kn}} G_n(\mathbf{z}_n, \tilde{\mathbf{z}}_n) = \sum_{m=1}^M w_{mk} \left(\hat{d}'_\beta(a_{mn}, \tilde{a}_{mn}) + \check{d}'_\beta(a_{mn}, \frac{\tilde{a}_{mn} z_{kn}}{\tilde{z}_{kn}}) \right),$$

and the Hessian matrix is diagonal with entries

$$\nabla_{z_{kn}}^2 G_n(\mathbf{z}_n, \tilde{\mathbf{z}}_n) = \sum_{m=1}^M w_{mk} \frac{\tilde{a}_{mn}}{\tilde{z}_{kn}} \left(\check{d}_\beta''(a_{mn}, \frac{\tilde{a}_{mn} z_{kn}}{\tilde{z}_{kn}}) \right).$$

Note in all cases, the first-order derivative or the second-order derivative corresponds to the second argument of $d_\beta(\cdot, \cdot)$.

Since $\check{d}_\beta(\cdot, \cdot)$ is convex in the second argument, the Hessian is positive definite. Thus, the auxiliary function is convex. These constructions result in the following theorem by minimizing the auxiliary function obtained in Theorem 13.6.

Theorem 13.8: (Nonincreasing of MU for β -divergence (Févotte and Idier, 2011; Gillis, 2020)) Let $\mathbf{A} \in \mathbb{R}_+^{M \times N}$, $\mathbf{W} \in \mathbb{R}_{++}^{M \times K}$, and $\mathbf{Z} \in \mathbb{R}_{++}^{K \times N}$. The loss $D_\beta(\mathbf{A}, \mathbf{WZ})$ remains nonincreasing under the following multiplicative update rules:

$$\mathbf{Z} \leftarrow \mathbf{Z} \circledast \left(\frac{[\mathbf{W}^\top \{(\mathbf{WZ})^{(\beta-2)} \circledast \mathbf{A}\}]}{[\mathbf{W}^\top (\mathbf{WZ})^{(\beta-1)}]} \right)^{m(\beta)}, \text{ and } \mathbf{W} \leftarrow \mathbf{W} \circledast \left(\frac{[\{(\mathbf{WZ})^{(\beta-2)} \circledast \mathbf{A}\} \mathbf{Z}^\top]}{[(\mathbf{WZ})^{(\beta-1)} \mathbf{Z}^\top]} \right)^{m(\beta)},$$

where

$$m(\beta) = \begin{cases} \frac{1}{2-\beta}, & \text{if } \beta < 1; \\ 1, & \text{if } 1 \leq \beta \leq 2; \\ \frac{1}{\beta-1}, & \text{if } \beta > 1. \end{cases}$$

When $\beta = 2$, the result reduces to Theorem 13.1. When $1 \leq \beta \leq 2$, the MU obtained via the MM framework coincides with heuristic described in Section 13.8.1.

The update in Theorem 13.8 ensures nonnegativity of the parameter updates, provided they are initialized with positive values.

Choice of β for NMF. The choice of β -divergence for NMF is problem-dependent. Févotte et al. (2009) present results of decomposing a piano power spectrogram using $\beta = 0$ and demonstrate that components corresponding to very low residual noise and hammer strikes on the strings are extracted with great accuracy; these components are either ignored or severely degraded when using Euclidean or KL distances/divergences. FitzGerald et al. (2009) show that $\beta = 0.5$ is optimal for music source separation problems.

Convergence. An algorithm is said to be *convergent* if it produces a sequence of iterates $\{\mathbf{Z}^{(t)}\}_{t \geq 1}$ or $\{\mathbf{W}^{(t)}\}_{t \geq 1}$ that converges to a limit point \mathbf{W}^* or \mathbf{Z}^* satisfying the KKT conditions in (13.23). Monotonic nonincreasingness does not imply convergence in general, and neither is monotonicity necessary for convergence. Proving convergence of the MU methods is beyond the scope of this book; we refer the readers to Gillis (2020); Févotte and Idier (2011) and references therein for more details.

13.8.4 Initialization of NMF

A significant challenge in NMF is the lack of guaranteed convergence to a global minimum. Often, the convergence process is slow, and the algorithm may reach a suboptimal approximation. In the preceding discussion, we initialized \mathbf{W} and \mathbf{Z} randomly. To mitigate this issue, 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 reference:

- *Clustering techniques.* Apply some clustering methods to the columns of \mathbf{A} , set 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 that \mathbf{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 approach.* Suppose the optimal rank- K approximation of \mathbf{A} is $\mathbf{A} = \sum_{i=1}^K \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$, 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 and \mathbf{v}_i , and nonnegative σ_i . Denote $[x]_+ = \max(x, 0)$, we notice

$$\mathbf{u}_i \mathbf{v}_i^\top = [\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top + [-\mathbf{u}_i]_+ [-\mathbf{v}_i]_+^\top - [-\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top - [\mathbf{u}_i]_+ [-\mathbf{v}_i]_+^\top,$$

where the first two rank-one factors in this decomposition are nonnegative. Then, either $[\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top$ or $[-\mathbf{u}_i]_+ [-\mathbf{v}_i]_+^\top$ can be selected to replace the factor $\mathbf{u}_i \mathbf{v}_i^\top$. Boutsidis and Gallopoulos (2008) suggest to replace each rank-one factor in $\sum_{i=1}^K \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$ with either $[\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top$ or $[-\mathbf{u}_i]_+ [-\mathbf{v}_i]_+^\top$, selecting the one with the larger norm and scaling it properly. In other words, if we select $[\mathbf{u}_i]_+ [\mathbf{v}_i]_+^\top$, then $\sigma_i \cdot [\mathbf{u}_i]_+$ can be initialized as the i -th column of \mathbf{W} , and $[\mathbf{v}_i]_+^\top$ can be chosen as the i -th row of \mathbf{Z} .

However, these techniques are not guaranteed to yield better performance theoretically. We recommend referring to the aforementioned papers for more detailed information.

13.9. Movie Recommender Context

Both NMF and ALS methods approximate a matrix and reconstruct its entries using a set of basis/template vectors. The key difference lies in the nature of these basis vectors and how the approximation is carried out. The basis in NMF is composed of vectors with nonnegative elements while the basis vectors in ALS can have positive or negative values. In NMF, each vector is reconstructed as a nonnegative summation of the basis vectors with “relatively” small components in the direction of each basis vector. In contrast, in the ALS approximation, the data is modeled as a linear combination of the basis vector such that we 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.

Movie recommender context. In the context of a movie recommender system, the rows of \mathbf{W} represent the hidden features of movies, while the columns of \mathbf{Z} represent the hidden features of users. For example, in NMF, a movie might be described as 0.5 comedy, 0.002 action, and 0.09 romantic. However, in the ALS approach, we can get combinations such as 4 comedy, -0.05 action, and -3 drama, indicating positive or negative contributions to each feature.

Implicit hierarchy. Both ALS and NMF do not rank the importance of each basis vector hierarchically. In contrast, singular value decomposition (SVD) ranks the importance of each basis vector based on the corresponding singular value. In the SVD representation 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, followed by the second set, and so on. This creates an implicit hierarchy in the SVD basis that doesn’t happen in the ALS or the NMF approach.

Interpretability of basis vectors. In SVD, the basis vectors can be statistically interpreted as the directions of maximum variance, but many of these directions lack a clear visual or intuitive interpretation due to the presence of zero, positive, and negative entries. When these basis vectors are used in a linear combination, the combination involves complex cancellations of positive and negative values, which can obscure the intuitive physical meaning of individual basis vectors. As a result, many basis vectors do not provide a meaningful explanation for nonnegative data, such as pixel values in a color image. On one hand, the entries of a nonnegative pattern vector should all be nonnegative values. On the other hand, mutually orthogonal singular vectors must contain negative entries. For example, if all entries of the singular vector \mathbf{u}_1 corresponding to the maximum singular value are non-negative, then any other singular vector orthogonal to \mathbf{u}_1 must contain at least one negative entry; otherwise, the orthogonality condition $\mathbf{u}_1^\top \mathbf{u}_j = 0$ for $j \neq 1$ cannot be satisfied. This indicates that mutually orthogonal singular vectors are not suitable as pattern vectors or basis vectors in nonnegative data analysis.

13.10. Other Applications

Music spectral reconstruction. To illustrate the application of NMF, we demonstrate how this technique can decompose a spectrogram of a music recording into components

that carry musical significance (Müller, 2015). As an example, let's examine the opening measures of *Frédéric Chopin's Prélude Op. 28, No. 4*. Figure 13.2 presents the musical notation alongside a piano-roll visualization that is synchronized with an audio recording of the piece. For clarity, all information pertaining to the note numbered $p = 71$ are emphasized with red rectangular frames.



Figure 13.2: Musical score and piano-roll representation. Figure is adapted from Müller (2015).

Regarding the original data matrix \mathbf{A} , we utilize the magnitude STFT (see, for example, López-Serrano et al. (2019)), which consists of a series of spectral vectors. By applying NMF, this matrix can be decomposed into two nonnegative matrices, \mathbf{W} and \mathbf{Z} . Ideally, \mathbf{W} encapsulates the spectral patterns corresponding to the pitches of the notes present in the musical piece, whereas \mathbf{Z} indicates the temporal points at which these patterns appear in the audio recording. Figure 13.3 illustrates such a decomposition applied to the Chopin prelude.

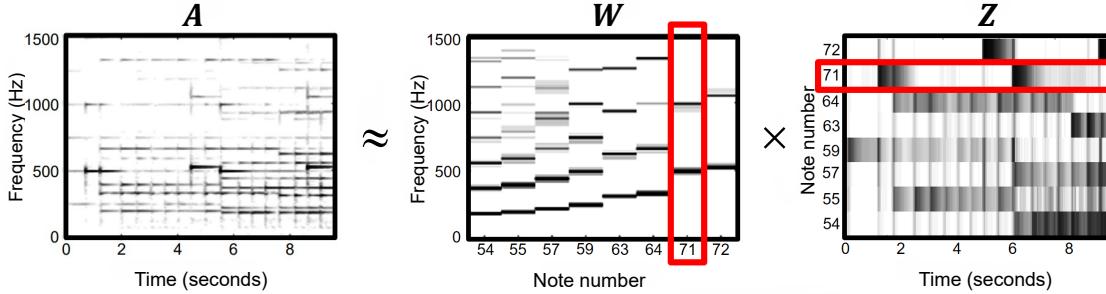


Figure 13.3: Ideal NMF of the spectrogram using NMF. Figure is adapted from Müller (2015).

In this scenario, each template represented by the matrix \mathbf{W} corresponds to the spectral manifestation of a specific pitch within \mathbf{A} , and the activation matrix \mathbf{Z} resembles the piano-roll representation of the musical score. Therefore, the advantages of NMF over general matrix factorization are evident:

- **Nonnegativity constraint.** NMF enforces nonnegativity on both the factorization matrices \mathbf{W} and \mathbf{Z} . This constraint makes the resulting matrices more interpretable because they can be directly related to physical or perceptual quantities in the domain of interest. In the case of music, the nonnegative factors correspond to meaningful musical elements like notes or chords.
- **Interpretability.** In NMF, the matrix \mathbf{W} represents the spectral patterns (timbres) of the notes present in the music piece, and \mathbf{Z} indicates the temporal activations of these patterns. This leads to a more interpretable decomposition compared to unconstrained matrix factorization methods, where the factors might not have a clear physical or musical interpretation.

$$\begin{array}{c}
 \mathbf{A}[:, n] \approx \mathbf{W} \times \mathbf{Z}[:, n] = \mathbf{WZ}[:, n]
 \end{array}$$

n-th facial image 49 basis vectors positive weights of basis vectors reconstruction of n-th facial image

Figure 13.4: NMF of the CBCL face database with $K = 49$. The basis vectors in \mathbf{W} are reshaped into 19×19 images. Facial features can be observed from these reshaped basis vectors, e.g., eyes, noses, nasolabial folds, and lips. Figure is adapted from Lee and Seung (1999); Gillis (2014).

Facial feature extraction and reconstruction. Suppose each column of the data matrix $\mathbf{A} \in \mathbb{R}_+^{M \times N}$ represents a vectorized grayscale image of a face, where the (m, n) -th entry of \mathbf{A} corresponds to the intensity of the m -th pixel in the n -th face. NMF decomposes \mathbf{A} into two nonnegative matrices \mathbf{W} and \mathbf{Z} such that each image a_n can be approximated by a linear combination of the columns of \mathbf{W} . Since \mathbf{W} is nonnegative, its columns can be interpreted as images, referred to as *template images* or *basis images*, which are vectors of pixel intensities. The nonnegative weights in \mathbf{Z} ensure that these template images are combined additively to reconstruct each original face image. Given that the number of basis images K is typically much smaller than the number of images N , the basis images should capture localized and sparse features that are common across multiple images. For facial images, these basis images often represent features such as eyes, noses, nasolabial folds, and lips (see Figure 13.4, NMF for the CBCL face data ⁸), while the columns of \mathbf{Z} indicate the presence of these features in each image (Lee and Seung, 1999; Gillis, 2014).

On the other hand, if each column of \mathbf{A} indicates a facial image of a single person, the NMF approach can be utilized for *face recognition*. NMF has been shown to be more robust to *occlusions* compared to PCA or ALS, which generates dense factors. For instance, if a new face with occlusions (e.g., wearing sunglasses or distortions) needs to be mapped into the NMF basis, the non-occluded parts (e.g., the lips or the noise) can still be accurately approximated (Jain et al., 2017).

Topic recovery. As mentioned at the very beginning of this chapter, NMF can be effectively applied to topic recovery problems. Typically, this application involves constructing a term-document matrix \mathbf{A} , where the rows correspond to terms (words or phrases) and the columns correspond to documents. Each entry a_{mn} in the matrix represents the weight or importance of term m in document n . This weight could be binary (presence/absence), *term frequency (TF)*, or *term frequency-inverse document frequency (TF-IDF)* (Shahnaz et al., 2006). In this framework, each column of \mathbf{W} can be interpreted as a topic, while each column of \mathbf{Z} indicates the positive weight of each topic for the given document; NMF in this context aligns with a *soft clustering* approach where each data point can belong to multiple clusters. NMF is particularly well-suited for topic recovery because it captures the additive nature of topics in text data and produces interpretable results. However, the choice of the number of topics K and the initialization of \mathbf{W} and \mathbf{Z} can significantly affect

⁸. <http://cbcl.mit.edu/software-datasets/FaceData2.html>

the quality of the results. Additionally, NMF is sensitive to the scaling of the input data, so it's important to preprocess the term-document matrix appropriately.

Decorative wavy lines flanking the chapter title

1. **L -strongly smooth and PGD in Hi-ANLS problems.** A function $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ is called an L -Lipschitz gradient continuous function (a.k.a., a L -strongly smooth function) if it satisfies that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2$ for all \mathbf{x}, \mathbf{y} . Show that the subproblem (13.4) in Hi-ANLS is L -strongly smooth with constant $L = \|\mathbf{W}[:, k]\|_2^2$. Therefore, the subproblem can be treated as a *projected gradient descent (PGD)* update with a step size $\eta = \frac{1}{L}$, i.e., using gradient descent update with a step size $\eta = \frac{1}{L}$ first and projecting the update onto the nonnegative orthant afterwards (Lu, 2025).
2. **Descent lemma for L -strongly smooth functions.** Let $f : \mathbb{S} \rightarrow (-\infty, \infty]$ be a function defined over a convex set \mathbb{S} such that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2$ for all \mathbf{x} and \mathbf{y} . Show that $f(\mathbf{y}) \leq f(\mathbf{x}) + \nabla f(\mathbf{x})^\top (\mathbf{y} - \mathbf{x}) + \frac{L}{2} \|\mathbf{x} - \mathbf{y}\|_2^2$. Hint: Use the fundamental theorem of calculus: $\nabla f(\mathbf{x} + \alpha \mathbf{d}) - \nabla f(\mathbf{x}) = \int_0^\alpha \nabla^2 f(\mathbf{x} + t\mathbf{d}) dt$.
3. Let $\mathbf{a} \in \mathbb{R}^M$, $\mathbf{z} \in \mathbb{R}^K$, and $\mathbf{W} \in \mathbb{R}^{K \times L}$. Show that the third-order partial derivatives of $F(\mathbf{z}) = \frac{1}{2} \|\mathbf{a} - \mathbf{Wz}\|_2^2$ vanish.
4. **MM applied to L -strongly smooth functions.** Let $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ be a L -strongly smooth function such that $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq L \|\mathbf{x} - \mathbf{y}\|_2$ for all \mathbf{x}, \mathbf{y} . Show that $g(\mathbf{x}, \tilde{\mathbf{x}}) = f(\tilde{\mathbf{x}}) + \nabla f(\tilde{\mathbf{x}})^\top (\mathbf{x} - \tilde{\mathbf{x}}) + \frac{L}{2} \|\mathbf{x} - \tilde{\mathbf{x}}\|_2^2$ is an auxiliary function for $f(\mathbf{x})$. Find the update rule for this problem.
5. Derive the gradients and gradient descent updates for the tri-NMF problem in (13.17).
6. **Projection property-O.** Let $\mathbb{S} \subset \mathbb{R}^n$ be **any set** and $\mathbf{y} \in \mathbb{R}^n$ such that $\tilde{\mathbf{y}} = \mathcal{P}_{\mathbb{S}}(\mathbf{y})$ is the projection of \mathbf{y} onto set \mathbb{S} . Show that for all $\mathbf{x} \in \mathbb{S}$, we have $\|\tilde{\mathbf{y}} - \mathbf{y}\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2$.
7. **Projection property-I.** Let $\mathbb{S} \subset \mathbb{R}^n$ be a **convex set** and $\mathbf{y} \in \mathbb{R}^n$ such that $\tilde{\mathbf{y}} = \mathcal{P}_{\mathbb{S}}(\mathbf{y})$. Show that for all $\mathbf{x} \in \mathbb{S}$, we have $\langle \mathbf{x} - \tilde{\mathbf{y}}, \mathbf{y} - \tilde{\mathbf{y}} \rangle \leq 0$, i.e., the angle between the two vectors is greater than 90° .
8. **Projection property-II.** Let $\mathbb{S} \subset \mathbb{R}^n$ be a **convex set** and $\mathbf{y} \in \mathbb{R}^n$ such that $\tilde{\mathbf{y}} = \mathcal{P}_{\mathbb{S}}(\mathbf{y})$. Show that for all $\mathbf{x} \in \mathbb{S}$, we have $\|\tilde{\mathbf{y}} - \mathbf{x}\|_2 \leq \|\mathbf{y} - \mathbf{x}\|_2$ and $\|\tilde{\mathbf{y}} - \mathbf{x}\|_2^2 \leq \|\mathbf{y} - \mathbf{x}\|_2^2 - \|\mathbf{y} - \tilde{\mathbf{y}}\|_2^2$ (the latter is related to the Pythagorean theorem). Hint: Examine $\|\mathbf{y} - \mathbf{x}\|_2^2 = \|(\tilde{\mathbf{y}} - \mathbf{x}) - (\tilde{\mathbf{y}} - \mathbf{y})\|_2^2$ and Problem 13.7.
9. **Linear feasibility.** Let $\mathbb{S} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} = \mathbf{b}\}$ with full row rank \mathbf{A} . Show that $\mathcal{P}_{\mathbb{S}}(\mathbf{x}) = \mathbf{x} - \mathbf{A}^\top (\mathbf{AA}^\top)^{-1}(\mathbf{Ax} - \mathbf{b})$.
10. **AB divergence (Amari and Nagaoka, 2000).** Let the α - β (AB) divergence be given as follows:

$$d_{\alpha, \beta}(x, y) = \begin{cases} -\frac{1}{\alpha\beta}(x^\alpha y^\beta - \frac{\alpha}{\alpha+\beta} x^{\alpha+\beta} - \frac{\beta}{\alpha+\beta} y^{\alpha+\beta}), & \alpha, \beta, \alpha + \beta \neq 0; \\ \frac{1}{\alpha^2}(x^\alpha \ln(\frac{x^\alpha}{y^\alpha}) - x^\alpha + y^\alpha), & \alpha \neq 0, \beta = 0; \\ \frac{1}{\alpha^2}(\ln(\frac{y^\alpha}{x^\alpha}) + (\frac{y^\alpha}{x^\alpha})^{-1} - 1), & \alpha = -\beta \neq 0; \\ \frac{1}{\beta^2}(y^\beta \ln(\frac{y^\beta}{x^\beta}) - y^\beta + x^\beta), & \alpha = 0, \beta \neq 0; \\ \frac{1}{2}(\ln(x) - \ln(y))^2, & \alpha = 0, \beta = 0. \end{cases}$$

When $\alpha + \beta = 1$, it is called the α -divergence. Discuss under what conditions it reduces to the β -divergence. Show that $d_{\alpha, \beta}(x, y) \geq 0$ and the equality holds if and only if $x = y$.

11. **Orthogonal and projective NMF, and clustering.** Consider the same setting as the orthogonal or projective matrix factorization in Problem 12.7, and suppose

further that \mathbf{A} , \mathbf{W} , and \mathbf{Z} are nonnegative. Show that there is only one positive entry in each column of \mathbf{Z} in this case. How is this related to the K-means problem? When each column of \mathbf{A} represents a data point, discuss the interpretation of z_{kn} (the (k, n) -th entry of \mathbf{Z}) as the importance of the k -th cluster to the n -th data point in the projective NMF case; that is, each data point can belong to several clusters.

12. Suppose $\mathbf{A} \geq \mathbf{0}_n$ is nonnegative ⁹ and has a positive row. If $|\mathbf{Ax}| = \mathbf{A}|\mathbf{x}|$, where $\mathbf{x} \in \mathbb{C}^n$, then there exists a real $\theta \in [0, 2\pi)$ such that $e^{-i\theta}\mathbf{x} = |\mathbf{x}|$, where $e^{-i\theta}\mathbf{x}$ indicates a complex vector with j -th element being $e^{-i\theta}x_j$. Hint: Use triangle inequality $|\mathbf{Ax}| \leq |\mathbf{A}||\mathbf{x}|$, and examine the positive row. In the polar coordinate notation, $e^{i\theta} = \cos \theta + i \sin \theta$ and $|e^{i\theta}\mathbf{x}| = |\mathbf{x}| \implies \|e^{i\theta}\mathbf{x}\|_2 = \|\mathbf{x}\|_2$.
13. **Nonnegative algebra.** A bounty of results can be harvested from nonnegative conditions. We investigate several of them in this problem. Given square matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D} \in \mathbb{R}^{n \times n}$, show that

- **Triangle inequality.** $|\mathbf{AB}| \leq |\mathbf{A}||\mathbf{B}|$.
- **Nonexpansiveness.** $|\mathbf{A}^k| \leq |\mathbf{A}|^k$, for all $k = \{1, 2, \dots\}$.
- **Equal norm.** $\|\mathbf{A}\|_F = \|\mathbf{|A|}\|_F$.
- If $|\mathbf{B}| \geq |\mathbf{A}|$, then $\|\mathbf{B}\|_F \geq \|\mathbf{A}\|_F$.
- If $\mathbf{B} \geq \mathbf{A} \geq \mathbf{0}$ and $\mathbf{D} \geq \mathbf{C} \geq \mathbf{0}$, then $\mathbf{BD} \geq \mathbf{AC} \geq \mathbf{0}$.
- If $\mathbf{B} \geq \mathbf{A} \geq \mathbf{0}$, then $\mathbf{B}^k \geq \mathbf{A}^k \geq \mathbf{0}$, for all $k = \{1, 2, \dots\}$,

where $\mathbf{B} \geq \mathbf{A}$ indicates that $\mathbf{B} - \mathbf{A}$ is a nonnegative matrix. Given rectangular matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$, show that

- $|\mathbf{A} + \mathbf{B}| \leq |\mathbf{A}| + |\mathbf{B}|$.

14. * **Eigenvalue interlacing in nonnegative matrices.** Let $\mathbf{B} - |\mathbf{A}| \in \mathbb{R}_+^{n \times n}$ be nonnegative. Show that

$$\rho(\mathbf{A}) \leq \rho(|\mathbf{A}|) \leq \rho(\mathbf{B}),$$

where $\rho(\mathbf{X})$ represents the spectral radius of \mathbf{X} (Definition 0.2). Hint: Use Problem 13.13 and Gelfand formula; show that $\|\mathbf{A}^k\|_F \leq \|\mathbf{|A|}^k\|_F \leq \|\mathbf{B}^k\|_F$.

15. Use Problem 13.14 to show that $\rho(\mathbf{B}) \geq \rho(\mathbf{A})$ if $\mathbf{B} \geq \mathbf{A} \geq \mathbf{0}$.
16. Let $\mathbf{A} \in \mathbb{R}_+^{n \times n}$ be nonnegative, let $\mathbf{B} = \mathbf{A}[1:k, 1:k]$, $\forall k \in \{1, 2, \dots, n\}$ (i.e., any leading principal submatrix of \mathbf{A} , Definition 1.3), and let $\mathbf{C} \in \mathbb{R}^{k \times k}$, $\forall k \in \{1, 2, \dots, n\}$ be any principal submatrix of \mathbf{A} (Definition 1.2). Show that

- $\rho(\begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}) \leq \rho(\mathbf{A}) \implies \rho(\mathbf{B}) \leq \rho(\mathbf{A})$.
- Use the first result to prove $\rho(\mathbf{C}) \leq \rho(\mathbf{A})$. Hint: Use permutation transformations.
- $\max_{i=1,2,\dots,n} a_{ii} \leq \rho(\mathbf{A})$.

17. * Let $\mathbf{A} \in \mathbb{R}_+^{n \times n}$ be nonnegative. Show that

$$\text{Row sum: } \min_{1 \leq i \leq n} \sum_{j=1}^n a_{ij} \leq \rho(\mathbf{A}) \leq \max_{1 \leq i \leq n} \sum_{j=1}^n a_{ij};$$

$$\text{Column sum: } \min_{1 \leq j \leq n} \sum_{i=1}^n a_{ij} \leq \rho(\mathbf{A}) \leq \max_{1 \leq j \leq n} \sum_{i=1}^n a_{ij}.$$

⁹. $\mathbf{A} \geq \mathbf{0}_n$ indicates that \mathbf{A} is an $n \times n$ nonnegative matrix, and $\mathbf{A} \geq \mathbf{0}_{m,n}$ indicates that \mathbf{A} is an $m \times n$ nonnegative matrix. Note that $\mathbf{C} = |\mathbf{A}|$ is defined as the matrix obtained by setting each entry of \mathbf{C} as the absolute value of $\mathbf{A} \in \mathbb{R}^{n \times n}$.

14

Biconjugate Decomposition

14.1. Existence of the Biconjugate Decomposition

The concept of *biconjugate decomposition* was introduced by Chu et al. (1995). However, its underlying principle—the rank-diminishing operator—on the other hand, has roots in the work of Egervary (1960); Householder (1964); Stewart (1973). A variety of matrix decomposition methods can be unified through this biconjugate decomposition. In Section 14.3, biconjugate decomposition is put into perspective by providing connections with standard decompositional methods, namely LDU, Cholesky, QR, and SVD decompositions. The existence of the biconjugate decomposition is supported by the rank-one reduction theorem, as presented below.

Theorem 14.1: ((Wedderburn's) rank-one reduction) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be an $m \times n$ matrix of rank r , and let $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ be a pair of vectors such that $w = \mathbf{y}^\top \mathbf{A} \mathbf{x} \neq 0$. Then the matrix

$$\mathbf{B} = \mathbf{A} - w^{-1} \mathbf{A} \mathbf{x} \mathbf{y}^\top \mathbf{A} \quad (14.1)$$

has rank $r - 1$, which is exactly one less than the rank of \mathbf{A} , i.e., $\text{rank}(\mathbf{B}) = \text{rank}(\mathbf{A}) - 1$.

A generalization of the rank-one reduction is discussed in Problem 14.4.

Proof [of Theorem 14.1] To prove the theorem, it suffices to show that the dimension of the null space of \mathbf{B} is one greater than that of \mathbf{A} , indicating that \mathbf{B} has a 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 have $\mathbf{B}\mathbf{n} = \mathbf{A}\mathbf{n} - w^{-1} \mathbf{A} \mathbf{x} \mathbf{y}^\top \mathbf{A} \mathbf{n} = \mathbf{0}$, implying that $\mathcal{N}(\mathbf{A}) \subseteq \mathcal{N}(\mathbf{B})$.

Now, consider any vector $\mathbf{m} \in \mathcal{N}(\mathbf{B})$, i.e., $\mathbf{B}\mathbf{m} = \mathbf{0}$. We have $\mathbf{B}\mathbf{m} = \mathbf{A}\mathbf{m} - w^{-1} \mathbf{A} \mathbf{x} \mathbf{y}^\top \mathbf{A} \mathbf{m} = \mathbf{0}$.

Let $k = w^{-1} \mathbf{y}^\top \mathbf{A} \mathbf{m}$, which is a scalar. Therefore, $\mathbf{B}\mathbf{m} = \mathbf{A}(\mathbf{m} - k\mathbf{x}) = \mathbf{0}$, 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 $\mathbf{A}\mathbf{x} \neq \mathbf{0}$ based on 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 one. Consequently, the dimension of $\mathcal{N}(\mathbf{A})$ is smaller than the dimension of $\mathcal{N}(\mathbf{B})$ by one, which completes the proof. ■

The converse of the above theorem is also true, as stated in the following corollary.

Corollary 14.2: (Rank-one reduction, (Egervary, 1960)) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any $m \times n$ matrix, and let $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$ be two vectors. Then, the rank of the matrix $\mathbf{B} = \mathbf{A} - \sigma^{-1}\mathbf{u}\mathbf{v}^\top$ is less than that of \mathbf{A} if and only if there exist vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$ such that $\mathbf{u} = \mathbf{A}\mathbf{x}$, $\mathbf{v} = \mathbf{A}^\top\mathbf{y}$, and $\sigma = \mathbf{y}^\top\mathbf{A}\mathbf{x} \neq 0$. In this case, it holds that $\text{rank}(\mathbf{B}) = \text{rank}(\mathbf{A}) - 1$.

More generally, the rank-one reduction can be extended to reductions involving matrices of higher rank.

Corollary 14.3: (Rank- k reduction, (Cline and Funderlic, 1979)) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any $m \times n$ matrix. Let further $\mathbf{P} \in \mathbb{R}^{m \times k}$, $\mathbf{U} \in \mathbb{R}^{k \times k}$ be nonsingular, and $\mathbf{Q} \in \mathbb{R}^{n \times k}$. Then,

$$\text{rank}(\mathbf{A} - \mathbf{P}\mathbf{U}^{-1}\mathbf{Q}^\top) = \text{rank}(\mathbf{A}) - \text{rank}(\mathbf{P}\mathbf{U}^{-1}\mathbf{Q}^\top)$$

if and only if there exist $\mathbf{X} \in \mathbb{R}^{n \times k}$ and $\mathbf{Y} \in \mathbb{R}^{m \times k}$ such that

$$\mathbf{P} = \mathbf{A}\mathbf{X}, \quad \mathbf{Q} = \mathbf{A}^\top\mathbf{Y}, \quad \text{and} \quad \mathbf{U} = \mathbf{Y}^\top\mathbf{A}\mathbf{X}.$$

Suppose a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ has rank r . We can define a rank-reducing process to generate a sequence of matrices $\{\mathbf{A}_k\}$, known as *Wedderburn matrices* or *Wedderburn sequence*:

$$\mathbf{A}_1 = \mathbf{A} \quad \text{and} \quad \mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1}\mathbf{A}_k\mathbf{x}_k\mathbf{y}_k^\top\mathbf{A}_k, \quad \forall k \in \{1, 2, \dots, r\}, \quad (14.2)$$

where $\mathbf{x}_k \in \mathbb{R}^n$ and $\mathbf{y}_k \in \mathbb{R}^m$ are any vectors satisfying $w_k = \mathbf{y}_k^\top\mathbf{A}_k\mathbf{x}_k \neq 0$. The operator in Equation (14.1) is known as a *rank-diminishing operator*, and the process described by Equation (14.2) is referred to the *rank-reducing process*. And the sets $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r\}$ and $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r\}$ are called the *vectors associated with the rank-reducing process*. Alternatively, if we let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r]$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r]$, then the pair (\mathbf{X}, \mathbf{Y}) is said to *effect a rank-reducing process* for \mathbf{A} .

The sequence will terminate after r steps since the rank of \mathbf{A}_k decreases by exactly one at each step. The sequence can be written out as follows:

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{A}, \\ \mathbf{A}_1 - \mathbf{A}_2 &= w_1^{-1}\mathbf{A}_1\mathbf{x}_1\mathbf{y}_1^\top\mathbf{A}_1, \\ \mathbf{A}_2 - \mathbf{A}_3 &= w_2^{-1}\mathbf{A}_2\mathbf{x}_2\mathbf{y}_2^\top\mathbf{A}_2, \\ \mathbf{A}_3 - \mathbf{A}_4 &= w_3^{-1}\mathbf{A}_3\mathbf{x}_3\mathbf{y}_3^\top\mathbf{A}_3, \\ &\vdots = \vdots \\ \mathbf{A}_{r-1} - \mathbf{A}_r &= w_{r-1}^{-1}\mathbf{A}_{r-1}\mathbf{x}_{r-1}\mathbf{y}_{r-1}^\top\mathbf{A}_{r-1}, \\ \mathbf{A}_r - \mathbf{0} &= w_r^{-1}\mathbf{A}_r\mathbf{x}_r\mathbf{y}_r^\top\mathbf{A}_r. \end{aligned}$$

By summing up the sequence, we obtain

$$\begin{aligned} (\text{Rank-reducing}) : \quad & (\mathbf{A}_1 - \mathbf{A}_2) + (\mathbf{A}_2 - \mathbf{A}_3) + \dots + (\mathbf{A}_{r-1} - \mathbf{A}_r) + (\mathbf{A}_r - \mathbf{0}) \\ &= \mathbf{A}_1 = \mathbf{A} = \sum_{i=1}^r w_i^{-1}\mathbf{A}_i\mathbf{x}_i\mathbf{y}_i^\top\mathbf{A}_i. \end{aligned} \quad (14.3)$$

Therefore, we can derive the following decomposition directly from this rank-reducing process.

Theorem 14.4: (Biconjugate decomposition: form 1) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ any matrix of rank r . This equality (14.3), derived from the 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

$$\phi_k = \mathbf{A}_k \mathbf{x}_k \quad \text{and} \quad \psi_k = \mathbf{A}_k^\top \mathbf{y}_k, \quad \forall k \in \{1, 2, \dots, r\}.$$

Thus, different choices of the vectors \mathbf{x}_k and \mathbf{y}_k will result in different biconjugate factorizations, making this factorization quite general and versatile. In the following sections, we will explore its connections to several well-known matrix factorizations.

Remark 14.5. Regarding the vectors \mathbf{x}_k and \mathbf{y}_k in the Wedderburn sequence, the following orthogonality properties hold:

$$\begin{aligned} \mathbf{x}_k \in \mathcal{N}(\mathbf{A}_{k+1}) &\implies \mathbf{x}_k \perp \mathcal{C}(\mathbf{A}_{k+1}^\top), \\ \mathbf{y}_k \in \mathcal{N}(\mathbf{A}_{k+1}^\top) &\implies \mathbf{y}_k \perp \mathcal{C}(\mathbf{A}_{k+1}). \end{aligned}$$

To verify this, observe that:

$$\begin{aligned} \mathbf{A}_{k+1} \mathbf{x}_k &= (\mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k) \mathbf{x}_k \\ &= \mathbf{A}_k (\mathbf{x}_k - \frac{\mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k}{\mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k} \mathbf{x}_k) = \mathbf{0}, \end{aligned}$$

since $w_k = \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k \neq 0$. Hence, $\mathbf{x}_k \in \mathcal{N}(\mathbf{A}_{k+1})$. Similarly, it can be shown that $\mathbf{A}_{k+1}^\top \mathbf{y}_k = \mathbf{0}$.

Lemma 14.6: (General term formula of Wedderburn sequence: V1) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix of rank r , and let $\mathbf{A}_1 = \mathbf{A}$. For each matrix in the sequence defined by $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$ ($k \in \{1, 2, \dots, r-1\}$), the matrix \mathbf{A}_{k+1} can be expressed as

$$\mathbf{A}_{k+1} = \mathbf{A} - \sum_{i=1}^k w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}, \quad \forall k \in \{1, 2, \dots, r-1\},$$

where

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{w_i} \mathbf{u}_i \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\mathbf{y}_k^\top \mathbf{A} \mathbf{u}_i}{w_i} \mathbf{v}_i.$$

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r]$, $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r]$, $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r]$, and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r]$ be the column partitions for each set of vectors. Then, the rank-reducing process can be viewed as transforming the matrix pair (\mathbf{X}, \mathbf{Y}) into the pair (\mathbf{U}, \mathbf{V}) . ^a

a. It can be shown that if \mathbf{A} is symmetric and $\mathbf{X} = \mathbf{Y}$, then $\mathbf{U} = \mathbf{V}$.

The proof of this lemma is deferred to Section 14.4. We notice that $w_i = \mathbf{y}_i^\top \mathbf{A}_i \mathbf{x}_i$ in the general term formula is related to \mathbf{A}_i , which means the expression is not the true general term formula. We will later reformulate w_i in terms of the original matrix \mathbf{A} rather than

\mathbf{A}_i . From the general term formula of the Wedderburn sequence, we have:

$$\begin{aligned}\mathbf{A}_{k+1} &= \mathbf{A} - \sum_{i=1}^k w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}, \\ \mathbf{A}_k &= \mathbf{A} - \sum_{i=1}^{k-1} w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}.\end{aligned}$$

Subtracting these two equations yields: $\mathbf{A}_{k+1} - \mathbf{A}_k = -w_k^{-1} \mathbf{A} \mathbf{u}_k \mathbf{v}_k^\top \mathbf{A}$. Since the sequence is defined as $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$, we can deduce that $w_k^{-1} \mathbf{A} \mathbf{u}_k \mathbf{v}_k^\top \mathbf{A} = w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$. Consequently, it follows that

$$\begin{aligned}\mathbf{A} \mathbf{u}_k &= \mathbf{A}_k \mathbf{x}_k, \\ \mathbf{v}_k^\top \mathbf{A} &= \mathbf{y}_k^\top \mathbf{A}_k.\end{aligned}\tag{14.4}$$

Let $z_{k,i} = \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{w_i}$, which is a scalar. Referring to the definitions of \mathbf{u}_k and \mathbf{v}_k in the lemma above, we can express them explicitly as follows:

- $\mathbf{u}_1 = \mathbf{x}_1$;
- $\mathbf{u}_2 = \mathbf{x}_2 - z_{2,1} \mathbf{u}_1 \implies \mathbf{x}_2$ is a linear combination of \mathbf{u}_1 and \mathbf{u}_2 ;
- $\mathbf{u}_3 = \mathbf{x}_3 - z_{3,1} \mathbf{u}_1 - z_{3,2} \mathbf{u}_2 \implies \mathbf{x}_3$ is a linear combination of $\mathbf{u}_1, \mathbf{u}_2$, and \mathbf{u}_3 ;
-

Each coefficient $z_{k,i}$ ($i < k$) encodes the component of \mathbf{x}_k in that of \mathbf{u}_i . This process bears resemblance to the Gram–Schmidt process (Section 3.3). However, in this process, we do not perform an **orthogonal projection** of \mathbf{x}_2 onto \mathbf{x}_1 to find the vector component of \mathbf{x}_2 along \mathbf{x}_1 , as we would do in an orthogonal projection (Section 3.5). Instead, the vector of \mathbf{x}_2 along \mathbf{x}_1 is now defined by $z_{2,1}$ (i.e., an **oblique projection**; see Section 3.5). This process is illustrated in Figure 14.1.

In Figure 14.1(a), \mathbf{u}_2 is **not** perpendicular to \mathbf{u}_1 (in the Gram–Schmidt process, \mathbf{u}_2 would be perpendicular to \mathbf{u}_1 via orthogonal projections). Nevertheless, \mathbf{u}_2 does not lie on the same line as \mathbf{u}_1 , so $\{\mathbf{u}_1, \mathbf{u}_2\}$ can still span a \mathbb{R}^2 subspace. Similarly, in Figure 14.1(b), $\mathbf{u}_3 = \mathbf{x}_3 - z_{3,1} \mathbf{u}_1 - z_{3,2} \mathbf{u}_2$ does not lie in the subspace spanned by $\{\mathbf{u}_1, \mathbf{u}_2\}$, allowing $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$ to 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 $\{\mathbf{u}_2, \mathbf{u}_1\}$. This equivalence extends to the \mathbf{v}_i vectors and \mathbf{y}_i vectors as well. We can express this property as follows:

$$\begin{cases} \text{span}\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_j\} = \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_j\}, & \forall j \in \{1, 2, \dots, r\}; \\ \text{span}\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_j\} = \text{span}\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_j\}, & \forall j \in \{1, 2, \dots, r\}. \end{cases}\tag{14.5}$$

Furthermore, from the rank-reducing property of the Wedderburn sequence, we have the following nested inclusions:

$$\begin{cases} \mathcal{C}(\mathbf{A}_1) \supset \mathcal{C}(\mathbf{A}_2) \supset \mathcal{C}(\mathbf{A}_3) \supset \dots; \\ \mathcal{N}(\mathbf{A}_1^\top) \subset \mathcal{N}(\mathbf{A}_2^\top) \subset \mathcal{N}(\mathbf{A}_3^\top) \subset \dots. \end{cases}$$

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 $j < k + 1$. This also holds true for $\mathbf{x}_{k+1}^\top \mathbf{A}_{k+1}^\top \mathbf{y}_j = 0$ for all $j < k + 1$. From

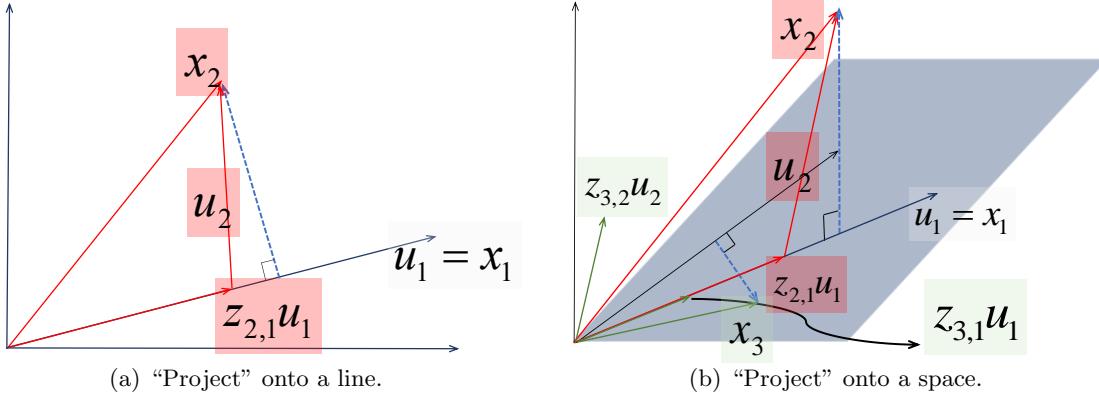


Figure 14.1: “Project” a vector onto a line and onto a space. Compare with the Gram–Schmidt process in Figure 3.2.

Equation (14.4), we also have $\mathbf{u}_{k+1}^\top \mathbf{A}^\top \mathbf{y}_j = 0$ for all $j < k + 1$. Following Equation (14.5), we obtain

$$\mathbf{x}_{k+1}^\top \mathbf{A}_{k+1}^\top \mathbf{y}_j = 0 \stackrel{(14.4)}{\implies} \mathbf{u}_{k+1}^\top \mathbf{A}^\top \mathbf{y}_j = 0 \stackrel{(14.5)}{\implies} \mathbf{v}_j^\top \mathbf{A} \mathbf{u}_{k+1} = 0 \quad \text{for all } j < k + 1. \quad (14.6)$$

Similarly, we can prove

$$\mathbf{v}_{k+1}^\top \mathbf{A} \mathbf{u}_j = 0 \quad \text{for all } j < k + 1. \quad (14.7)$$

Moreover, since $w_k = \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k$, according to Equation (14.4), we can express w_k as:

$$\begin{aligned} w_k &= \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k = \mathbf{v}_k^\top \mathbf{A} \mathbf{x}_k \\ &= \mathbf{v}_k^\top \mathbf{A} (\mathbf{u}_k + \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{w_i} \mathbf{u}_i) \quad (\text{by the definition of } \mathbf{u}_k \text{ in Lemma 14.6}) \\ &= \mathbf{v}_k^\top \mathbf{A} \mathbf{u}_k, \quad (\text{by } \mathbf{v}_k^\top \mathbf{A} \mathbf{u}_j = 0 \text{ for all } j < k) \end{aligned}$$

which can be utilized to substitute the value of w_k in Lemma 14.6. We then have the full version of the general term formula of the Wedderburn sequence. In this form, the formula no longer depends on matrices \mathbf{A}_k (in the form of w_k 's):

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{\mathbf{v}_i^\top \mathbf{A} \mathbf{u}_i} \mathbf{u}_i \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\mathbf{y}_k^\top \mathbf{A} \mathbf{u}_i}{\mathbf{v}_i^\top \mathbf{A} \mathbf{u}_i} \mathbf{v}_i. \quad (14.8)$$

Gram–Schmidt process from Wedderburn sequence. Suppose the matrices $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r] \in \mathbb{R}^{n \times r}$ and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_r] \in \mathbb{R}^{n \times r}$ effect a rank-reducing process for $\mathbf{A} \in \mathbb{R}^{n \times n}$. If $\mathbf{A} = \mathbf{I} \in \mathbb{R}^{n \times n}$ is the identity matrix and $(\mathbf{X} = \mathbf{Y})$ are identical, containing the vectors for which an orthogonal basis is desired (i.e., we aim to obtain an orthogonal basis spanning the same column space of $\mathbf{X} = \mathbf{Y}$), then the result of the rank-reducing process, $(\mathbf{U} = \mathbf{V})$, gives the resultant orthogonal basis (but not an orthonormal basis as that in the Gram–Schmidt process):

- To see this, we follow the computation of the Gram–Schmidt process as given in Equation (3.3), where we replace the data matrix with $\mathbf{X} = \mathbf{Y} \in \mathbb{R}^{m \times n}$ (assuming \mathbf{X} has full rank for simplicity), and we use the matrix $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_n]$ to denote

the orthonormal basis. Then, for any $k \in \{1, 2, \dots, n\}$, the Gram–Schmidt process admits

$$\text{Gram–Schmidt process} = \begin{cases} \mathbf{x}_k^\perp = \left(\mathbf{I} - \sum_{i=1}^{k-1} \mathbf{q}_i \mathbf{q}_i^\top \right) \mathbf{x}_k; \\ \mathbf{q}_k = \frac{\mathbf{x}_k^\perp}{\|\mathbf{x}_k^\perp\|}. \end{cases} \quad (14.9)$$

- Considering the rank-reducing process with $\mathbf{A} = \mathbf{I}$, $\mathbf{X} = \mathbf{Y}$. Since \mathbf{A} is symmetric and $\mathbf{X} = \mathbf{Y}$, we obtain $\mathbf{U} = \mathbf{V}$. We can therefore focus on the analysis of \mathbf{U} . For any $k \in \{1, 2, \dots, n\}$, the rank-reducing process yields

$$\text{Rank-reducing process} = \begin{cases} \mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_k}{\mathbf{v}_i^\top \mathbf{A} \mathbf{u}_i} \mathbf{u}_i = \left(\mathbf{I} - \sum_{i=1}^{k-1} \frac{\mathbf{u}_i \mathbf{u}_i^\top}{\mathbf{u}_i^\top \mathbf{u}_i} \right) \mathbf{x}_k; \\ \tilde{\mathbf{u}}_k = \frac{\mathbf{u}_k}{\|\mathbf{u}_k\|}. \end{cases} \quad (14.10)$$

Therefore, by comparing Equations (14.9) and (14.10), we can find the equivalence between \mathbf{q}_k and $\tilde{\mathbf{u}}_k$ for $k \in \{1, 2, \dots, n\}$. Thus, the Wedderburn sequence effects a Gram–Schmidt process when $\mathbf{X} = \mathbf{Y}$ is the data matrix and $\mathbf{A} = \mathbf{I}$.

“Projection” notations. The expressions for \mathbf{u}_k and \mathbf{v}_k in Equation (14.8) closely resemble the projection onto the perpendicular space in the Gram–Schmidt process, as shown in Equation (3.1). We then define the bilinear form $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{y}^\top \mathbf{A} \mathbf{x}$ to explicitly mimic the form of projection in Equation (3.1).

We consolidate the results established so far into the following lemma, which provides a concise overview of what we have been working on. These results will be extensively utilized in the sequel.

Lemma 14.7: (Properties of Wedderburn sequence) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix of rank r , and let $\mathbf{A}_1 = \mathbf{A}$. For each matrix in the sequence defined by $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$ ($k \in \{1, 2, \dots, r-1\}$), the matrix \mathbf{A}_{k+1} can be expressed 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

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{x}_k, \mathbf{v}_i \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{u}_i \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{u}_i, \mathbf{y}_k \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{v}_i. \quad (14.11)$$

Furthermore, we can observe the following properties:

$$\begin{aligned} \mathbf{A} \mathbf{u}_k &= \mathbf{A}_k \mathbf{x}_k; \\ \mathbf{v}_k^\top \mathbf{A} &= \mathbf{y}_k^\top \mathbf{A}_k; \end{aligned} \quad (14.12)$$

$$\langle \mathbf{u}_k, \mathbf{v}_j \rangle = \langle \mathbf{u}_j, \mathbf{v}_k \rangle = 0 \text{ for all } j < k; \quad (14.13)$$

$$w_k = \mathbf{y}_k^\top \mathbf{A}_k \mathbf{x}_k = \langle \mathbf{u}_k, \mathbf{v}_k \rangle. \quad (14.14)$$

By substituting Equation (14.12) into Form 1 of the biconjugate decomposition and using Equation (14.14), which implies $w_k = \mathbf{v}_k^\top \mathbf{A} \mathbf{u}_k$, we obtain the Form 2 and Form 3 of this decomposition:

Theorem 14.8: (Biconjugate decomposition: Form 2 and Form 3) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix of rank r . This equality (14.3), which results from the rank-reducing process, implies the following matrix decomposition:

$$\mathbf{A} = \mathbf{A} \mathbf{U}_r \boldsymbol{\Omega}_r^{-1} \mathbf{V}_r^\top \mathbf{A}, \quad (14.15)$$

where $\boldsymbol{\Omega}_r = \text{diag}(w_1, w_2, \dots, w_r)$, $\mathbf{U}_r = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \in \mathbb{R}^{n \times r}$, and $\mathbf{V}_r = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_r] \in \mathbb{R}^{m \times r}$ with

$$\mathbf{u}_k = \mathbf{x}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{x}_k, \mathbf{v}_i \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{u}_i \quad \text{and} \quad \mathbf{v}_k = \mathbf{y}_k - \sum_{i=1}^{k-1} \frac{\langle \mathbf{u}_i, \mathbf{y}_k \rangle}{\langle \mathbf{u}_i, \mathbf{v}_i \rangle} \mathbf{v}_i. \quad (14.16)$$

Additionally, for any $\gamma \leq r$, the following decomposition holds:

$$\mathbf{V}_\gamma^\top \mathbf{A} \mathbf{U}_\gamma = \boldsymbol{\Omega}_\gamma, \quad (14.17)$$

where $\boldsymbol{\Omega}_\gamma = \text{diag}(w_1, w_2, \dots, w_\gamma)$, $\mathbf{U}_\gamma = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_\gamma] \in \mathbb{R}^{n \times \gamma}$, and $\mathbf{V}_\gamma = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_\gamma] \in \mathbb{R}^{m \times \gamma}$. Note the difference between the subscripts r and γ employed here, where $\gamma \leq r$.

Note that Equation (14.15) is derived from (14.12), and Equation (14.17) is a consequence of (14.13). Importantly, these two forms of the biconjugate decomposition no longer depend on the intermediate Wedderburn matrices $\{\mathbf{A}_k\}$.

Notation. In the following discussion, we will use subscripts to indicate the dimensions of matrices to avoid ambiguity. For example, the use of r and γ in the above theorem highlights the size of the constructed matrices.

14.2. Properties of the Biconjugate Decomposition

The following corollary establishes a connection between matrices \mathbf{U}_γ and \mathbf{X}_γ through unique unit upper triangular matrices derived from the Wedderburn sequence.

Corollary 14.9: (Connection of \mathbf{U}_γ and \mathbf{X}_γ) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix of rank $r \geq \gamma$. 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 exist 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}_γ and \mathbf{V}_γ are matrices whose columns are derived from the Wedderburn sequence, as described in Equation (14.17).

Proof [of Corollary 14.9] The proof follows directly from the definitions of \mathbf{u}_k and \mathbf{v}_k in Equations (14.11) or (14.16). We construct the j -th columns of $\mathbf{R}_\gamma^{(x)}$ and $\mathbf{R}_\gamma^{(y)}$ as follows:

$$\left[\frac{\langle \mathbf{x}_j, \mathbf{v}_1 \rangle}{\langle \mathbf{u}_1, \mathbf{v}_1 \rangle}, \frac{\langle \mathbf{x}_j, \mathbf{v}_2 \rangle}{\langle \mathbf{u}_2, \mathbf{v}_2 \rangle}, \dots, \frac{\langle \mathbf{x}_j, \mathbf{v}_{j-1} \rangle}{\langle \mathbf{u}_{j-1}, \mathbf{v}_{j-1} \rangle}, 1, 0, 0, \dots, 0 \right]^\top,$$

and

$$\left[\frac{\langle \mathbf{u}_1, \mathbf{y}_j \rangle}{\langle \mathbf{u}_1, \mathbf{v}_1 \rangle}, \frac{\langle \mathbf{u}_2, \mathbf{y}_j \rangle}{\langle \mathbf{u}_2, \mathbf{v}_2 \rangle}, \dots, \frac{\langle \mathbf{u}_{j-1}, \mathbf{y}_j \rangle}{\langle \mathbf{u}_{j-1}, \mathbf{v}_{j-1} \rangle}, 1, 0, 0, \dots, 0 \right]^\top.$$

And the uniqueness stems from the fact that the matrices \mathbf{U}_γ and \mathbf{V}_γ have independent columns from the rank-reducing process. This completes the proof. ■

The pair $(\mathbf{U}_\gamma, \mathbf{V}_\gamma) \in \mathbb{R}^{m \times \gamma} \times \mathbb{R}^{n \times \gamma}$ in Theorem 14.8 is called a **biconjugate pair** with respect to \mathbf{A} if Ω_γ is nonsingular and diagonal. Furthermore, suppose the pair $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma) \in \mathbb{R}^{n \times \gamma} \times \mathbb{R}^{m \times \gamma}$ effects a rank-reducing process for \mathbf{A} . Then, the pair $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$ is said to be **biconjugatable** and can be **biconjugated into a biconjugate pair** of matrices $(\mathbf{U}_\gamma, \mathbf{V}_\gamma)$, if there exist unit upper triangular matrices $\mathbf{R}_\gamma^{(x)}$ and $\mathbf{R}_\gamma^{(y)}$ such that $\mathbf{X}_\gamma = \mathbf{U}_\gamma \mathbf{R}_\gamma^{(x)}$ and $\mathbf{Y}_\gamma = \mathbf{V}_\gamma \mathbf{R}_\gamma^{(y)}$.

14.3. Connection to Well-Known Decomposition Methods

In this section, we demonstrate how biconjugate decomposition relates to well-known matrix factorization methods.

14.3.1 LDU Decomposition

Theorem 14.10: (LDU, Chu et al. (1995) Theorem 2.4) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix of rank $r \geq \gamma$. Let further $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma) \in \mathbb{R}^{n \times \gamma} \times \mathbb{R}^{m \times \gamma}$ with $\gamma \in \{1, 2, \dots, r\}$. Then, the pair $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$ is biconjugatable if and only if $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma$ admits an LDU decomposition.

Proof [of Theorem 14.10] Suppose \mathbf{X}_γ and \mathbf{Y}_γ are biconjugatable. Then there exist 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. 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 LDU decomposition of $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma$. This expression can be regarded as the **fourth form of biconjugate decomposition**.

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. Since the inverses \mathbf{R}_1^{-1} and \mathbf{R}_2^{-1} are also unit upper triangular matrices, the pair $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$ can be biconjugated into $(\mathbf{X}_\gamma \mathbf{R}_1^{-1}, \mathbf{Y}_\gamma \mathbf{R}_2^{-1})$. This completes the proof. ■

Corollary 14.11: (Determinant) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix of rank $r \geq \gamma$. Suppose the pair $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma) \in \mathbb{R}^{n \times \gamma} \times \mathbb{R}^{m \times \gamma}$ can be biconjugated into $(\mathbf{U}_\gamma, \mathbf{V}_\gamma)$ such that $\mathbf{V}_\gamma^\top \mathbf{A} \mathbf{U}_\gamma = \Omega_\gamma = \text{diag}(w_1, w_2, \dots, w_\gamma)$ is a nonsingular diagonal matrix. Then it follows that

$$\det(\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma) = \prod_{i=1}^{\gamma} w_i.$$

Proof [of Corollary 14.11] By Theorem 14.10, since $(\mathbf{X}_\gamma, \mathbf{Y}_\gamma)$ are biconjugatable, then there exist unit upper triangular matrices $\mathbf{R}_\gamma^{(x)}$ and $\mathbf{R}_\gamma^{(y)}$ such that $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma = \mathbf{R}_\gamma^{(y)\top} \Omega_\gamma \mathbf{R}_\gamma^{(x)}$. The

determinant is simply the product of the diagonal elements. \blacksquare

Lemma 14.12: (Biconjugatable in principal minors) Let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be any matrix of rank $r \geq \gamma$. In the Wedderburn sequence, we choose \mathbf{x}_i as the i -th standard 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 standard 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, $\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma$ corresponds to 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 into $(\mathbf{U}_\gamma, \mathbf{V}_\gamma)$ such that $\mathbf{V}_\gamma^\top \mathbf{A} \mathbf{U}_\gamma = \mathbf{\Omega}_\gamma = \text{diag}(w_1, w_2, \dots, w_\gamma)$ is a nonsingular diagonal matrix if and only if the γ -th leading principal minor of \mathbf{A} is nonzero, i.e., $\det(\mathbf{A}[1 : \gamma, 1 : \gamma]) \neq 0$. In this case, the γ -th leading principal minor of \mathbf{A} is given by $\prod_{i=1}^\gamma w_i$.

Proof [of Lemma 14.12] The proof is straightforward that the γ -th leading principal minor of \mathbf{A} being nonzero will imply that $w_i \neq 0$ for all $i \leq \gamma$. Thus, the Wedderburn sequence can be successfully obtained. The converse holds because Corollary 14.11 implies that $\det(\mathbf{Y}_\gamma^\top \mathbf{A} \mathbf{X}_\gamma)$ is nonzero. \blacksquare

We have now arrived at the LDU decomposition for square matrices.

Theorem 14.13: (LDU: Biconjugate decomposition for square matrices) For any matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, the pair $(\mathbf{I}_n, \mathbf{I}_n)$ 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

$$\mathbf{A} = \mathbf{V}_n^{-\top} \mathbf{\Omega}_n \mathbf{U}_n^{-1} = \mathbf{L} \mathbf{D} \mathbf{U},$$

where $\mathbf{\Omega}_n = \mathbf{D}$ is a diagonal matrix with nonzero values along its 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 14.13] As per Lemma 14.12, it is evident that the pair $(\mathbf{I}_n, \mathbf{I}_n)$ is bi-conjugatable. Based on Corollary 14.9, 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. This completes the proof. \blacksquare

14.3.2 Cholesky Decomposition

For symmetric and positive definite matrices, all leading principal minors are always positive. The proof for this statement can be found in Section 2.3. The following theorem shows how the Cholesky decomposition arises naturally from biconjugate decomposition in the case of positive definite matrices.

Theorem 14.14: (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 derived from the Wedderburn sequence by setting $(\mathbf{X}_n, \mathbf{Y}_n)$ as the pair $(\mathbf{I}_n, \mathbf{I}_n)$. In this case, \mathbf{A} can be factored as

$$\mathbf{A} = \mathbf{U}_n^{-\top} \mathbf{\Omega}_n \mathbf{U}_n^{-1} = (\mathbf{U}_n^{-\top} \mathbf{\Omega}_n^{1/2})(\mathbf{\Omega}_n^{1/2} \mathbf{U}_n^{-1}) = \mathbf{R}^\top \mathbf{R},$$

where Ω_n is a diagonal matrix with positive values along the diagonal, and \mathbf{U}_n^{-1} is a unit upper triangular matrix.

Proof [of Theorem 14.14] Given that the leading principal minors of positive definite matrices are positive, $w_i > 0$ for all $i \in \{1, 2, \dots, n\}$. It follows from the LDU factorization via biconjugation and the symmetry of \mathbf{A} that $\mathbf{A} = \mathbf{U}_n^{-\top} \Omega_n \mathbf{U}_n^{-1}$. Since w_i 's are positive, Ω_n is positive definite and can be factored as $\Omega_n = \Omega_n^{1/2} \Omega_n^{1/2}$. This implies that $\Omega_n^{1/2} \mathbf{U}_n^{-1}$ is the Cholesky factor. ■

14.3.3 QR Decomposition

Without loss of generality, we assume that $\mathbf{A} \in \mathbb{R}^{n \times n}$ has full rank, which allows for the QR decomposition: $\mathbf{A} = \mathbf{Q}\mathbf{R}$, where $\mathbf{Q} \in \mathbb{R}^{n \times n}$ is orthogonal, and $\mathbf{R} \in \mathbb{R}^{n \times n}$ is upper triangular with full rank and positive diagonal values. We now show how this decomposition can be obtained through biconjugate decomposition.

Theorem 14.15: (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 by setting $(\mathbf{X}_n, \mathbf{Y}_n)$ as $(\mathbf{I}_n, \mathbf{A})$. Thus, \mathbf{A} can be factored as

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

where $\mathbf{Q} = \mathbf{V}_n \Omega_n^{-1/2}$ is an orthogonal matrix, and $\mathbf{R} = \Omega_n^{1/2} \mathbf{R}_n^{(x)}$ is an upper triangular matrix, according to the **Form 4** in Theorem 14.10, with $\gamma = n$:

$$\mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n = \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \Omega_n \mathbf{R}_n^{(x)},$$

where we set $\gamma = n$ because γ can be any value such that $\gamma \leq r$, and the rank $r = n$.

Proof [of Theorem 14.15] Since $(\mathbf{X}_n, \mathbf{Y}_n) = (\mathbf{I}_n, \mathbf{A})$, applying Theorem 14.10, we have the decomposition $\mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n = \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \Omega_n \mathbf{R}_n^{(x)}$. Substituting $(\mathbf{X}_n, \mathbf{Y}_n) = (\mathbf{I}_n, \mathbf{A})$ into the decomposition above, we obtain:

$$\begin{aligned} \mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n &= \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \Omega_n \mathbf{R}_n^{(x)}; \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}_n^{(y)\top} \Omega_n \mathbf{R}_n^{(x)}; \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}_1^\top \Omega_n \mathbf{R}_1; \quad (\mathbf{A}^\top \mathbf{A} \text{ is symmetric and let } \mathbf{R}_1 = \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)}) \quad (14.18) \\ \mathbf{A}^\top \mathbf{A} &= (\mathbf{R}_1^\top \Omega_n^{1/2\top})(\Omega_n^{1/2} \mathbf{R}_1); \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}^\top \mathbf{R}. \quad (\text{let } \mathbf{R} = \Omega_n^{1/2} \mathbf{R}_1) \end{aligned}$$

To see why Ω_n can be factored as $\Omega_n = \Omega_n^{1/2\top} \Omega_n^{1/2}$, we consider the following steps. Suppose $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_n]$ is the column partition of \mathbf{A} . We obtain $w_i = \mathbf{y}_i^\top \mathbf{A} \mathbf{x}_i = \mathbf{a}_i^\top \mathbf{a}_i > 0$, since \mathbf{A} is nonsingular. Therefore, $\Omega_n = \text{diag}(w_1, w_2, \dots, w_n)$ is positive definite and it can be factored as

$$\Omega_n = \Omega_n^{1/2} \Omega_n^{1/2} = \Omega_n^{1/2\top} \Omega_n^{1/2}. \quad (14.19)$$

By $\mathbf{X}_\gamma = \mathbf{U}_\gamma \mathbf{R}_\gamma^{(x)}$ in Theorem 14.10 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 14.10 for all $\gamma \in \{1, 2, \dots, n\}$, we have

$$\begin{aligned} \mathbf{Y}_n &= \mathbf{V}_n \mathbf{R}_1; \\ \mathbf{A} &= \mathbf{V}_n \mathbf{R}_1; \\ \mathbf{A}^\top \mathbf{A} &= \mathbf{R}_1^\top \mathbf{V}_n^\top \mathbf{V}_n \mathbf{R}_1; \\ \mathbf{R}_1^\top \boldsymbol{\Omega}_n \mathbf{R}_1 &= \mathbf{R}_1^\top \mathbf{V}_n^\top \mathbf{V}_n \mathbf{R}_1; \\ (\mathbf{R}_1^\top \boldsymbol{\Omega}_n^{1/2\top}) (\boldsymbol{\Omega}_n^{1/2} \mathbf{R}_1) &= (\mathbf{R}_1^\top \boldsymbol{\Omega}_n^{1/2\top} \boldsymbol{\Omega}_n^{-1/2\top}) \mathbf{V}_n^\top \mathbf{V}_n (\boldsymbol{\Omega}_n^{-1/2} \boldsymbol{\Omega}_n^{1/2} \mathbf{R}_1); \\ \mathbf{R}^\top \mathbf{R} &= \mathbf{R}^\top (\boldsymbol{\Omega}_n^{-1/2\top} \mathbf{V}_n^\top) (\mathbf{V}_n \boldsymbol{\Omega}_n^{-1/2}) \mathbf{R}. \end{aligned} \quad \begin{array}{l} (\mathbf{A} = \mathbf{Y}_n) \\ \text{(Equation (14.18))} \\ \text{(Equation (14.19))} \end{array}$$

Thus, $\mathbf{Q} = \mathbf{V}_n \boldsymbol{\Omega}_n^{-1/2}$ is an orthogonal matrix. ■

14.3.4 SVD

To explore the SVD of a square matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ within the biconjugation decomposition, we introduce the following notation: let $\mathbf{A} = \mathbf{U}^{\text{svd}} \boldsymbol{\Sigma}^{\text{svd}} \mathbf{V}^{\text{svd}\top}$ be the SVD of \mathbf{A} , where $\mathbf{U}^{\text{svd}} = [\mathbf{u}_1^{\text{svd}}, \mathbf{u}_2^{\text{svd}}, \dots, \mathbf{u}_n^{\text{svd}}]$ is orthogonal, $\mathbf{V}^{\text{svd}} = [\mathbf{v}_1^{\text{svd}}, \mathbf{v}_2^{\text{svd}}, \dots, \mathbf{v}_n^{\text{svd}}]$ is orthogonal, and $\boldsymbol{\Sigma}^{\text{svd}} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$ is diagonal. Without loss of generality, we assume $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\text{rank}(\mathbf{A}) = n$. Readers can verify the equivalence for a general matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$.

If the pair $(\mathbf{X}_n = \mathbf{V}^{\text{svd}}, \mathbf{Y}_n = \mathbf{U}^{\text{svd}})$ effects a rank-reducing process for \mathbf{A} . From the definitions of \mathbf{u}_k and \mathbf{v}_k in Equation (14.11) or Equation (14.16), we have

$$\mathbf{u}_k = \mathbf{v}_k^{\text{svd}} \quad \text{and} \quad \mathbf{v}_k = \mathbf{u}_k^{\text{svd}} \quad \text{and} \quad w_k = \mathbf{y}_k^\top \mathbf{A} \mathbf{x}_k = \sigma_k.$$

This implies $\mathbf{V}_n = \mathbf{U}^{\text{svd}}$, $\mathbf{U}_n = \mathbf{V}^{\text{svd}}$, and $\boldsymbol{\Omega}_n = \boldsymbol{\Sigma}^{\text{svd}}$, where we set $\gamma = n$ because γ can be any value such that $\gamma \leq r$, and the rank $r = n$.

By $\mathbf{X}_n = \mathbf{U}_n \mathbf{R}_n^{(x)}$ in Theorem 14.10, we have

$$\mathbf{X}_n = \mathbf{U}_n \mathbf{R}_n^{(x)} \implies \mathbf{V}^{\text{svd}} = \mathbf{V}^{\text{svd}} \mathbf{R}_n^{(x)} \implies \mathbf{I}_n = \mathbf{R}_n^{(x)}.$$

By $\mathbf{Y}_n = \mathbf{V}_n \mathbf{R}_n^{(y)}$ in Theorem 14.10, we have

$$\mathbf{Y}_n = \mathbf{V}_n \mathbf{R}_n^{(y)} \implies \mathbf{U}^{\text{svd}} = \mathbf{U}^{\text{svd}} \mathbf{R}_n^{(y)} \implies \mathbf{I}_n = \mathbf{R}_n^{(y)}.$$

Applying Theorem 14.10 again and setting $\gamma = n$, we have

$$\mathbf{Y}_n^\top \mathbf{A} \mathbf{X}_n = \mathbf{R}_n^{(y)\top} \mathbf{V}_n^\top \mathbf{A} \mathbf{U}_n \mathbf{R}_n^{(x)} = \mathbf{R}_n^{(y)\top} \boldsymbol{\Omega}_n \mathbf{R}_n^{(x)}.$$

This simplifies to $\mathbf{U}^{\text{svd}\top} \mathbf{A} \mathbf{V}^{\text{svd}} = \boldsymbol{\Sigma}^{\text{svd}}$, which corresponds precisely to the form of a SVD. This demonstrates the equivalence between the SVD and the biconjugate decomposition when the Wedderburn sequence is applied with $(\mathbf{V}^{\text{svd}}, \mathbf{U}^{\text{svd}})$ as $(\mathbf{X}_n, \mathbf{Y}_n)$.

14.4. Proof: General Term Formula of Wedderburn Sequence

In Lemma 14.6, we present the general term formula for the Wedderburn sequence. Given any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, the Wedderburn sequence of \mathbf{A} is defined recursively by $\mathbf{A}_{k+1} = \mathbf{A}_k - w_k^{-1} \mathbf{A}_k \mathbf{x}_k \mathbf{y}_k^\top \mathbf{A}_k$ with $\mathbf{A}_1 = \mathbf{A}$. The proof of the general term formula for this sequence is as follows:

Proof [of Lemma 14.6] For \mathbf{A}_2 , let $\mathbf{u}_1 = \mathbf{x}_1$ and $\mathbf{v}_1 = \mathbf{y}_1$. We have:

$$\mathbf{A}_2 = \mathbf{A}_1 - w_1^{-1} \mathbf{A}_1 \mathbf{x}_1 \mathbf{y}_1^\top \mathbf{A}_1 = \mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}.$$

For \mathbf{A}_3 , we can write out the equation as:

$$\begin{aligned}
 \mathbf{A}_3 &= \mathbf{A}_2 - w_2^{-1} \mathbf{A}_2 \mathbf{x}_2 \mathbf{y}_2^\top \mathbf{A}_2 \\
 &= (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) - w_2^{-1} (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) \mathbf{x}_2 \mathbf{y}_2^\top (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) \quad (\text{substitute } \mathbf{A}_2) \\
 &= (\mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A}) - w_2^{-1} \mathbf{A} (\mathbf{x}_2 - w_1^{-1} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A} \mathbf{x}_2) (\mathbf{y}_2^\top - w_1^{-1} \mathbf{y}_2^\top \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top) \mathbf{A} \quad (\text{factor out } \mathbf{A}) \\
 &= \mathbf{A} - w_1^{-1} \mathbf{A} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A} - w_2^{-1} \mathbf{A} \mathbf{u}_2 \mathbf{v}_2^\top \mathbf{A} = \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A},
 \end{aligned}$$

where $\mathbf{u}_2 = \mathbf{x}_2 - w_1^{-1} \mathbf{u}_1 \mathbf{v}_1^\top \mathbf{A} \mathbf{x}_2 = \mathbf{x}_2 - \frac{\mathbf{v}_1^\top \mathbf{A} \mathbf{x}_2}{w_1} \mathbf{u}_1$, and $\mathbf{v}_2 = \mathbf{y}_2 - w_1^{-1} \mathbf{y}_2^\top \mathbf{A} \mathbf{u}_1 \mathbf{v}_1 = \mathbf{y}_2 - \frac{\mathbf{y}_2^\top \mathbf{A} \mathbf{u}_1}{w_1} \mathbf{v}_1$. Similarly, we can find the expression of \mathbf{A}_4 by \mathbf{A} :

$$\begin{aligned}
 \mathbf{A}_4 &= \mathbf{A}_3 - w_3^{-1} \mathbf{A}_3 \mathbf{x}_3 \mathbf{y}_3^\top \mathbf{A}_3 \\
 &= \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} - w_3^{-1} (\mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}) \mathbf{x}_3 \mathbf{y}_3^\top (\mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A}) \quad (\text{substitute } \mathbf{A}_3) \\
 &= \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} - w_3^{-1} \mathbf{A} (\mathbf{x}_3 - \sum_{i=1}^2 w_i^{-1} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} \mathbf{x}_3) (\mathbf{y}_3^\top - \sum_{i=1}^2 w_i^{-1} \mathbf{y}_3^\top \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top) \mathbf{A} \quad (\text{factor out } \mathbf{A}) \\
 &= \mathbf{A} - \sum_{i=1}^2 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A} - w_3^{-1} \mathbf{A} \mathbf{u}_3 \mathbf{v}_3^\top \mathbf{A} = \mathbf{A} - \sum_{i=1}^3 w_i^{-1} \mathbf{A} \mathbf{u}_i \mathbf{v}_i^\top \mathbf{A},
 \end{aligned}$$

where $\mathbf{u}_3 = \mathbf{x}_3 - \sum_{i=1}^2 \frac{\mathbf{v}_i^\top \mathbf{A} \mathbf{x}_3}{w_i} \mathbf{u}_i$, and $\mathbf{v}_3 = \mathbf{y}_3 - \sum_{i=1}^2 \frac{\mathbf{y}_3^\top \mathbf{A} \mathbf{u}_i}{w_i} \mathbf{v}_i$. Continuing 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 the general term of the Wedderburn sequence can be proved by induction. ■

~~~~~ Chapter 14 Problems ~~~~

1. Following the proof of Theorem 14.1, prove Corollaries 14.2 and 14.3.
2. Discuss the Wedderburn sequence of $(\mathbf{X}_n, \mathbf{Y}_n) = (\mathbf{I}_n, \mathbf{A})$ in Theorem 14.15 if \mathbf{A} is singular.
3. Following Section 14.3.4, verify the equivalence between the SVD and the biconjugate decomposition for a general matrix \mathbf{A} of size $m \times n$.
4. **Rank reduction theorem.** Let $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{X} \in \mathbb{R}^{n \times k}$, and $\mathbf{Y} \in \mathbb{R}^{m \times k}$ such that $\mathbf{W} = \mathbf{Y}^\top \mathbf{A} \mathbf{X}$ is nonsingular. Show that

$$\text{rank}(\mathbf{A} - \mathbf{A} \mathbf{X} \mathbf{W}^{-1} \mathbf{Y}^\top \mathbf{A}) = \text{rank}(\mathbf{A}) - \text{rank}(\mathbf{A} \mathbf{X} \mathbf{W}^{-1} \mathbf{Y}^\top \mathbf{A}).$$

When $k = 1$, this is the rank-one reduction (Theorem 14.1). Discuss how this general result relates to Corollary 14.3.

5. Show that if \mathbf{A} is symmetric and $\mathbf{X} = \mathbf{Y}$, then $\mathbf{U} = \mathbf{V}$ in Lemma 14.6.

Bibliography

- Charu C Aggarwal. *Linear algebra and optimization for machine learning*, volume 156. Springer, 2020.
- Shun-ichi Amari and Hiroshi Nagaoka. *Methods of information geometry*, volume 191. American Mathematical Soc., 2000.
- İsmail An, Umut Şimşekli, Ali Taylan Cemgil, and Laie Akarun. Large scale polyphonic music transcription using randomized matrix decompositions. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 2020–2024. IEEE, 2012.
- Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Convex optimization with sparsity-inducing norms. 2011.
- Sudipto Banerjee and Anindya Roy. *Linear algebra and matrix analysis for statistics*, volume 181. CRC Press Boca Raton, FL, USA, 2014.
- Amir Beck. *Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB*. SIAM, 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.
- Dennis S. Bernstein. Matrix mathematics: Theory, facts, and formulas. 2008.
- Dennis S Bernstein. *Matrix mathematics: theory, facts, and formulas*. Princeton university press, 2009.
- Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.
- Christopher M Bishop. Pattern recognition. *Machine learning*, 128(9), 2006.
- Åke Björck. The calculation of linear least squares problems. *Acta Numer.*, 13:1–51, 2004.
- Åke Björck. *Numerical methods for least squares problems*. SIAM, 2024.
- Jonathan Borwein and Adrian Lewis. *Convex Analysis*. Springer, 2006.
- Christos Boutsidis and Efstratios Gallopoulos. SVD based initialization: A head start for nonnegative matrix factorization. *Pattern recognition*, 41(4):1350–1362, 2008.

- 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.
- Thomas Brouwer, Jes Frellsen, and Pietro Lió. Comparative study of inference methods for Bayesian nonnegative matrix factorisation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 513–529. Springer, 2017.
- Jean-Philippe Brunet, Pablo Tamayo, Todd R Golub, and Jill P Mesirov. Metagenes and molecular pattern discovery using matrix factorization. *Proceedings of the national academy of sciences*, 101(12):4164–4169, 2004.
- Zvonimir Bujanovic, Lars Karlsson, and Daniel Kressner. A householder-based algorithm for hessenberg-triangular reduction. *SIAM Journal on Matrix Analysis and Applications*, 39(3):1270–1294, 2018.
- James R Bunch and Linda Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Mathematics of computation*, pages 163–179, 1977.
- 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.
- Eric C Chi and Tamara G Kolda. On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications*, 33(4):1272–1299, 2012.
- 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.
- Martino Ciaperoni, Aristides Gionis, and Heikki Mannila. The Hadamard decomposition problem. *Data Mining and Knowledge Discovery*, pages 1–42, 2024.
- Randall E Cline and Robert E Funderlic. The rank of a difference of matrices and associated generalized inverses. *Linear Algebra and its Applications*, 24:185–215, 1979.
- Joel E Cohen and Uriel G Rothblum. Nonnegative ranks, decompositions, and factorizations of nonnegative matrices. *Linear Algebra and its Applications*, 190:149–168, 1993.
- 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.
- Robyn M Dawes and Bernard Corrigan. Linear models in decision making. *Psychological bulletin*, 81(2):95, 1974.
- 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, Ravi Kannan, and Michael W Mahoney. Fast Monte Carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on computing*, 36(1):158–183, 2006.
- Jean-Guillaume Dumas and Clément Pernet. Symmetric indefinite triangular factorization revealing the rank profile matrix. In *Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation*, pages 151–158, 2018.

- Eugen Egervary. On rank-diminishing operations and their applications to the solution of linear equations. *Zeitschrift fur angewandte Mathematik und Physik ZAMP*, 11:376–386, 1960.
- Lars Elden. *Matrix methods in data mining and pattern recognition*. SIAM, 2007.
- Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences*, 35(11):652–655, 1949.
- C edric F evotte and J er ome Idier. Algorithms for nonnegative matrix factorization with the β -divergence. *Neural computation*, 23(9):2421–2456, 2011.
- C edric F evotte, Nancy Bertin, and Jean-Louis Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.
- Ricardo D Fierro and Per Christian Hansen. Low-rank revealing UTV decompositions. *Numerical Algorithms*, 15(1):37–55, 1997.
- Derry FitzGerald, Matt Cranitch, and Eugene Coyle. On the use of the beta divergence for musical source separation. 2009.
- 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.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.
- Shmuel Friedland and Anatoli Torokhti. Generalized rank-constrained matrix approximations. *SIAM Journal on Matrix Analysis and Applications*, 29(2):656–659, 2007.
- Jean Gallier and Jocelyn Quaintance. Linear algebra for computer vision, robotics, and machine learning, 2019.
- Yuan Gao and George Church. Improving molecular cancer class discovery through sparse non-negative matrix factorization. *Bioinformatics*, 21(21):3970–3975, 2005.
- 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, Walter Murray, and Margaret H Wright. *Practical optimization*. SIAM, 2019.
- 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.
- Nicolas Gillis. *Nonnegative matrix factorization*. SIAM, 2020.

- Nicolas Gillis and François Glineur. Accelerated multiplicative updates and hierarchical als algorithms for nonnegative matrix factorization. *Neural computation*, 24(4):1085–1105, 2012.
- 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, Nikolai Leonidovich Zamarashkin, and Evgenii Evgen'evich Tyrtyshnikov. Pseudo-skeleton approximations by matrices of maximal volume. *Mathematical Notes*, 62(4):515–519, 1997.
- Jørgen Pedersen Gram. Ueber die entwicklung reeller funktionen in reihen mittelst der methode der kleinsten quadrate. 1883.
- David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.
- AW Hales and IBS Passi. Jordan decomposition. In *Algebra*, pages 75–87. Springer, 1999.
- Richard J Hanson and Charles L Lawson. Extensions and applications of the householder algorithm for solving linear least squares problems. *Mathematics of Computation*, 23(108):787–812, 1969.
- Moritz Hardt, Raghu Meka, Prasad Raghavendra, and Benjamin Weitz. Computational limits for matrix completion. In *Conference on Learning Theory*, pages 703–725. PMLR, 2014.
- 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.
- Trevor Hastie, Robert Tibshirani, and Martin Wainwright. Statistical learning with sparsity. *Monographs on statistics and applied probability*, 143(143):8, 2015.
- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.
- Nicholas J Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002a.
- Nicholas J Higham. Computing the nearest correlation matrix—a problem from finance. *IMA journal of Numerical Analysis*, 22(3):329–343, 2002b.
- Nicholas J Higham. Cholesky factorization. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(2):251–254, 2009.
- Nicholas J Higham and Natasa Strabic. Bounds for the distance to the nearest correlation matrix. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1088–1102, 2016.
- Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge University Press, 2012.
- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

- Alston S Householder. *The theory of matrices in numerical analysis*. Blaisdell, 1964.
- Kejun Huang, Nicholas D Sidiropoulos, and Athanasios P Liavas. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization. *IEEE Transactions on Signal Processing*, 64(19):5052–5065, 2016.
- Tsung-Min Hwang, Wen-Wei Lin, and Eugene K Yang. Rank revealing LU factorizations. *Linear algebra and its applications*, 175:115–141, 1992.
- Prateek Jain, Purushottam Kar, et al. Non-convex optimization for machine learning. *Foundations and Trends® in Machine Learning*, 10(3-4):142–336, 2017.
- Camille Jordan. *Traité des substitutions et des équations algébriques*. Gauthier-Villars, 1870.
- Bo Kagström, Daniel Kressner, Enrique S Quintana-Ortí, and Gregorio Quintana-Ortí. Blocked algorithms for the reduction to hessenberg-triangular form revisited lapack working note 198. 2008.
- Jingu Kim and Haesun Park. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing*, 33(6):3261–3281, 2011.
- Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.
- 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.
- 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 H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *nature*, 401(6755):788–791, 1999.
- 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.
- Hyekyoung Lee and Seungjin Choi. CUR+NMF for learning spectral features from large data matrix. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1592–1597. IEEE, 2008.
- Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of applied mathematics*, 2(2):164–168, 1944.
- Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.
- Patricio López-Serrano, Christian Dittmar, Yigitcan Özer, and Meinard Müller. Nmf toolbox: Music processing applications of nonnegative matrix factorization. In *Proceedings of the International Conference on Digital Audio Effects DAFx*, volume 19, pages 2–6, 2019.
- 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. Numerical matrix decomposition. *arXiv preprint arXiv:2107.02579*, 2021c.
- Jun Lu. A rigorous introduction to linear models. *arXiv preprint arXiv:2105.04240*, Eliva Press, 2021d.
- Jun Lu. Bayesian low-rank interpolative decomposition for complex datasets. *arXiv preprint arXiv:2205.14825*, *Studies in Engineering and Technology*, 9(1):1–12, 2022a.
- Jun Lu. Comparative study of inference methods for interpolative decomposition. *arXiv preprint arXiv:2206.14542*, 2022b.
- Jun Lu. Gradient descent, stochastic optimization, and other tales. *arXiv preprint arXiv:2205.00832*, Eliva Press, 2022c.
- Jun Lu. Bayesian matrix decomposition and applications. *arXiv preprint arXiv:2302.11337*, 2023.
- Jun Lu. Practical topics in optimization. *arXiv preprint arXiv:2503.05882*, 2025.
- Jun Lu and Joerg Osterrieder. Feature selection via the intervened interpolative decomposition and its application in diversifying quantitative strategies. 2022.
- Jun Lu and Xuanyu Ye. Flexible and hierarchical prior for Bayesian nonnegative matrix factorization. *arXiv preprint arXiv:2205.11025*, 2022.
- Jun Lu and Shao Yi. Autoencoding conditional GAN for portfolio allocation diversification. *arXiv preprint arXiv:2207.05701*, 2022.
- Jun Lu, Tianyi Xu, Bill Ding, David Li, and Yu Kang. Large language model compression via the nested activation-aware decomposition. *arXiv preprint arXiv:2503.17101*, 2025.
- Kaj Madsen, Hans Bruun Nielsen, and Ole Tingleff. Methods for non-linear least squares problems. 2004.
- Michael W Mahoney. Lecture notes on randomized linear algebra. *arXiv preprint arXiv:1608.04481*, 2016.
- 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.
- Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
- Per-Gunnar Martinsson. Randomized methods for matrix computations. *The Mathematics of Data*, 25(4):187–231, 2019.
- Roy Mathias and L Hogben. Singular values and singular value inequalities, 2014.
- Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11: 2287–2322, 2010.
- Thomas McSweeney. *Modified Cholesky decomposition and applications*. PhD thesis, University of Manchester, 2017.
- Jose Menchero, D Orr, and Jun Wang. The Barra US equity model (USE4), methodology notes. *English, MSCI (May*, 2011.
- L Miranian and Ming Gu. Strong rank revealing LU factorizations. *Linear algebra and its applications*, 367:1–16, 2003.

- Cleve B Moler and Gilbert W Stewart. An algorithm for generalized matrix eigenvalue problems. *SIAM Journal on Numerical Analysis*, 10(2):241–256, 1973.
- Raphael A Mrode. *Linear models for the prediction of animal breeding values*. Cabi, 2014.
- Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*, volume 5. Springer, 2015.
- Silvia Noschese, Lionello Pasquini, and Lothar Reichel. Tridiagonal toeplitz matrices: properties and novel applications. *Numerical linear algebra with applications*, 20(2):302–326, 2013.
- Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.
- C-T Pan. On the existence and computation of rank-revealing LU factorizations. *Linear Algebra and its Applications*, 316(1-3):199–222, 2000.
- 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.
- Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719, 2005.
- 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.
- Erhard Schmidt. Zur theorie der linearen und nichtlinearen integralgleichungen. *Mathematische Annalen*, 63(4):433–476, 1907.
- Matthias Seeger. Low rank updates for the Cholesky decomposition. Technical report, 2004.
- Farial Shahnaz, Michael W Berry, V Paul Pauca, and Robert J Plemmons. Document clustering using nonnegative matrix factorization. *Information Processing & Management*, 42(2):373–386, 2006.
- Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.
- Gilbert W Stewart. Conjugate direction methods for solving systems of linear equations. *Numerische Mathematik*, 21(4):285–297, 1973.
- Gilbert W Stewart. An updating algorithm for subspace tracking. *IEEE Transactions on Signal Processing*, 40(6):1535–1541, 2002.
- GW Stewart. The decompositional approach to matrix computation. *Computing in Science & Engineering*, 2(1):50–59, 2000.
- 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. *SIAM Review*, 64(1):181–190, 2022.
- Kuduvally Swamy. On Sylvester’s criterion for positive-semidefinite matrices. *IEEE Transactions on Automatic Control*, 18(3):306–306, 1973.
- 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.
- Saul A Teukolsky, Brian P Flannery, WH Press, and W Vetterling. Numerical recipes in c. *SMR*, 693(1):59–70, 1992.
- Andrei N Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Sov Dok*, 4:1035–1038, 1963.
- Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. SIAM, 1997.
- Saygun Turkay, Eduardo Epperlein, and Nicos Christofides. Correlation stress testing for value-at-risk. *Journal of Risk*, 5:75–90, 2003.
- Robert van de Geijn and Margaret Myers. Advanced linear algebra: Foundations to frontiers. *Creative Commons NonCommercial (CC BY-NC)*, 2020.
- 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.
- Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM journal on optimization*, 20(3):1364–1377, 2010.
- Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.
- SJ Wright and John Norman Holt. An inexact Levenberg-Marquardt method for large sparse nonlinear least squares. *The ANZIAM Journal*, 26(4):387–403, 1985.
- Zhirong Yang and Erkki Oja. Linear and nonlinear projective nonnegative matrix factorization. *IEEE Transactions on Neural Networks*, 21(5):734–749, 2010.
- Zhijian Yuan and Erkki Oja. Projective nonnegative matrix factorization for image compression and feature extraction. In *Image Analysis: 14th Scandinavian Conference, SCIA 2005, Joensuu, Finland, June 19-22, 2005. Proceedings 14*, pages 333–342. Springer, 2005.
- 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.

Alphabetical Index

L -strongly smoothness, 260

ℓ_1 regularization, 222

ℓ_2 regularization, 205

Adjugate, 9, 81, 114

Algebraic multiplicity, 128, 140, 158

ALS, 210

Alternating update, 245

Anisotropic scaling, 183

ANLS, *see* Nonnegative least squares

Approximate least squares, 192

Backward substitution, 15

Basis, 4, 85, 99, 180

Bayesian inference, 240

Bayesian matrix decomposition, 240

Bayesian optimization, 240

Bidiagonal matrix, 130

Cancellation, 59

Cayley–Hamilton theorem, 148

CGS, 56

Characteristic polynomial, 148, 158

Classical Gram–Schmidt process, 56

Clustering, 172

Cochran’s theorem, 161

Collaborative filtering, 208

Column pivoting, 61, 63

Column space, 4, 52, 85

Column-pivoted QR (CPQR), 61

Complete pivoting, 23, 169

Condition number, 38, 205

Consistent estimator, 193

Contour plot, 221

Convex function, 204, 221

Convexity, 210

Coordinate descent algorithm, 210

Coordinate transformation, 183

CPQR, 61, 86, 109, 169

Cramer’s rule, 107, 114

Cross-validation, 209, 215

Data least squares, 206

Data storage, 94, 102, 111

Data whitening, 195

Decomposition: ALS, 207

Decomposition: Biconjugate, 264

Decomposition: Bidiagonal, 130

Decomposition: Cholesky, 28

Decomposition: Complex Shur, 145

Decomposition: CR, 94

Decomposition: EVD, 140

Decomposition: Hessenberg, 119

Decomposition: ID, 105

Decomposition: LDU, 19, 269

Decomposition: LU, 14, 218

Decomposition: NMF, 238

Decomposition: QR, 52

Decomposition: RR UTV, 87

Decomposition: Schur, 143

Decomposition: Skeleton, 102

Decomposition: Smith, 99

Decomposition: Spectral, 152

Decomposition: SVD, 176

Decomposition: Tridiagonal, 126

Decomposition: UTV, 84

Determinant, 9, 143, 269

- Dimension, 4
- Eckart–Young–Mirsky theorem, 196
- Eigenvalue, 3
- Eigenvector, 3
- Elementary transformation, 15
- Fan's inequality, 174
- Fermat's theorem, 204, 235
- Fibonacci number, 146
- First-order optimality condition, 235
- Floating point operations (flops), 22
- Frobenius norm, 11, 197
- Fundamental spaces, 7, 73
- Fundamental theorem, 57, 99
- Fundamental theorem of linear algebra, 7
- Gauss–Newton method, 78
- Gaussian elimination, 15
- Gaussian kernel, 171
- Gaussian mixture models, 44
- Gaussian process, 28
- General formula of a sequence, 146
- Generative process, 28
- Geometric interpretation, 183, 221
- Geometric multiplicity, 128, 140, 158
- Givens geometric decomposition, 83
- Givens rotation, 67
- Global minimum, 210
- Golub–Kahan process, 133, 135
- Gradient descent, 218
- Gram–Schmidt, 54, 113, 265, 266
- Greedy search, 218
- Hadamard product, 215
- Hardy–Littlewood–Pólya inequality, 174
- Hessenberg matrix, 119
- Hidden features, 216
- Hierarchical ANLS, 241
- Householder geometric decomposition, 83
- Householder reflector, 64
- Householder transformation, 64
- Idempotent, 99, 167
- Implicit hierarchy, 257
- Implicit Q theorem, 123, 127
- Independence check, 60
- Induction, 36
- Inner product, 216
- Inverse of a matrix, 25
- Jordan block, 142
- Kernel clustering, 172
- Kernel function, 171
- Kernel outlier detection, 173
- KKT condition, 79, 240
- Krylov matrix, 124, 127
- Kullback–Leibler divergence, 245
- Latent semantic analysis, 197
- Leading principal minor, 14, 270
- Least squares, 42, 187, 189, 190, 203
- Level curves, 219
- Level surfaces, 219
- Levenberg–Marquardt method, 79
- LHC bidiagonalization, 134, 135
- Linear approximation, 218
- Linear kernel, 171
- Linear models, 203
- Linear update, 218
- Linearly independent, 4, 52, 94, 102, 105
- Logistic matrix decomposition, 234
- Low-rank approximation, 47, 80, 111, 196
- Machine precision, 248
- Matlab-style notation, 3, 106
- Matrix bandwidth, 21, 128
- Matrix completion, 208
- Matrix inverse, 148, 218
- Matrix multiplication, 183
- Matrix norm, 11, 197
- Matrix polynomial, 146
- MGS, 56
- Mirsky's theorem, 174
- Missing entries, 215
- Modified Gram–Schmidt process, 56
- Modified Newton's method, 45
- Multiplicative update, 245
- Multiplicity, 158
- Nearest correlation matrix problem, 46
- Netflix recommender, 207, 215
- Newton's method, 45, 78
- NMF, 238
- NNLS, *see* Nonnegative least squares
- Non-pivot columns, 98
- Nonlinear least squares, 78, 83
- Nonnegative least squares, 240
- Nonnegativity constraint, 238
- Nonsingular matrix, 10, 25, 103, 146

- Normal equation, 53, 187, 204, 241
 Null space (nullspace, kernel), 5
 Numerical stability, 56
- Orthogonal, 55, 84, 119, 126, 152, 176
 Orthogonal complement, 5
 Orthogonal matrix, 8, 55
 Orthogonal projection, 57, 167, 178
 Orthogonally equivalence, 181
 Orthonormal, 55
 Orthonormal basis, 73, 85, 180
 Outlier detection, 173, 232
 Overfitting, 221, 240
- Partial pivoting, 22
 Permutation matrix, 8, 61, 164
 Pivot, 17, 34, 95
 Pivot columns, 98
 Pivoting, 24, 169
 Polynomial kernel, 171
 Positive definite, 30, 32, 168
 Positive semidefinite, 30, 40, 168
 Principal component analysis, 193
 Principal minor, 14, 269
 Projection gradient descent, 221
 Projection matrix (projector), 57, 167
 Pseudo-inverse, 94, 99, 111, 190
- QR algorithm, 124
- Range, 4
 Rank, 5, 6, 94, 176
 Rank decomposition, 98, 99, 106, 175
 Rank-deficient, 189, 205
 Rank-one change, 73
 Rank-one reduction, 262
 Rank-one update, 43, 73
 Rank-revealing, 24, 40, 63, 169
 Rank-revealing LU, 24
 Rank-revealing QR, 64
 Rank-two update, 45
 Recursive algorithm, 31
 Reduced row echelon form, 97
 Regression analysis, 203
 Regularization, 213, 221, 248
 Revealing rank r deficiency, 64
 Revealing rank-one deficiency, 63
- Rook pivoting, 24
 Row equivalent, 26
 Row space, 72, 85, 103, 178
 Row-pivoted, 72
 RPLQ, 72
- Saddle point, 205
 Scatter matrix, 171
 Schur complement, 19, 30, 115
 Semidefinite rank-revealing, 40, 169
 Shared matrix decomposition, 235
 Similar matrices, 120, 158
 Similarity transformation, 120, 127, 142, 143
- Singular matrix, 10
 Skeleton, 102
 Skew-symmetric matrix, 163
 Smith decomposition, 99
 Span, 4, 52
 Sparsity, 222, 238
 Spectral radius, 4
 Spectrum, 4, 120
 Stochastic coordinate descent, 222
 Stochastic gradient descent, 218, 222
 Subspace, 4, 52, 178
 Sylvester's criterion, 33
 Sylvester's theorem, 148
- Tikhonov regularization, 205
 Toeplitz matrix, 129
 Total least squares, 206
 Trace, 100, 120
 Tridiagonal matrix, 126
 Truncated SVD, 194
 Trust region method, 79
 Two-block coordinate descent, 209
- Unbiased estimator, 193
 Uniqueness, 37, 71, 97, 157
 Upper bidiagonal matrix, 130
 Upper triangular, 35, 84, 143
- Variable separation, 166
 Variational autoencoder, 28
 Vector norm, 11
- Wedderburn sequence, 264
 Weighted matrix decomposition, 234