LINEAR ALGEBRA: LOW RANK AND COMPRESSED SENSING

A DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE AWARD OF THE DEGREE

OF

MASTER OF SCIENCE

IN

MATHEMATICS

by

SANJYOTI CHAURASIYA

(2K24/MSCMAT/38)

Under the Supervision of

Mr. Jamkhongam Touthang

Department of Applied Mathematics



Department of Applied Mathematics

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Shahbad Daulatpur, Main Bawana Road, Delhi-110042, India

CANDIDATE'S DECLARATION

I, **Sanjyoti Chaurasiya**, Roll No. **24/MSCMAT/38**, student of M.Sc. Applied Mathematics, hereby declare that the dissertation entitled **"Low Rank and Compressed Sensing"** submitted to the Department of Applied Mathematics, Delhi Technological University, in partial fulfillment of the requirements for the award of the Degree of Master of Science, is a record of original work carried out by me under the guidance and supervision of **Mr. Jamkhongam Touthang**.

I further declare that this dissertation has not been submitted, either in part or in full, to any other University or Institution for the award of any degree, diploma, or any other academic qualification.

Date: _____

Place: Delhi

Sanjyoti Chaurasiya
Roll No. 24/MSCMAT/38

CERTIFICATE BY THE SUPERVISOR

This is to certify that **Sanjyoti Chaurasiya**, Roll No. **24/MSCMAT/38**, has successfully carried out the research work presented in the dissertation entitled **"Low Rank and Compressed Sensing"** submitted to the Department of Applied Mathematics, Delhi Technological University, in partial fulfillment of the requirements for the award of the Degree of Master of Science under my supervision.

The dissertation embodies the results of original work carried out by the candidate. To the best of my knowledge, the contents of this dissertation do not form the basis for the award of any other degree, diploma, or similar recognition to the candidate by this or any other University/Institution.

Date: _____

Place: Delhi

Mr. Jamkhongam Touthang
Assistant Professor
Department of Applied Mathematics
Delhi Technological University

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, **Mr. Jamkhongam Touthang**, for his invaluable guidance, constant encouragement, and continuous support throughout the completion of this dissertation work. His insightful suggestions and motivation greatly helped me in understanding the subject and completing this research successfully.

I am also thankful to the faculty members and staff of the Department of Applied Mathematics, Delhi Technological University, for providing a healthy academic environment and necessary facilities during the course of this work.

I extend my heartfelt thanks to my family and friends for their encouragement, patience, and unwavering support throughout this journey.

Finally, I express my gratitude to all those who directly or indirectly contributed to the successful completion of this dissertation entitled **"Low Rank and Compressed Sensing"**.

Sanjyoti Chaurasiya

ABSTRACT

This dissertation presents a detailed study of low rank matrices and compressed sensing with emphasis on their mathematical foundations, optimization techniques, and practical applications in scientific computing, machine learning, and signal processing. The study begins with the theoretical background of low rank approximation, matrix perturbation, singular value decomposition, and sparse representations.

The dissertation discusses important topics such as the Sherman–Morrison–Woodbury formula, recursive least squares, Kalman filtering, quasi-Newton methods, interlacing eigenvalues, Krylov matrices, Vandermonde matrices, and low-displacement rank structures. Special focus is given to rapidly decaying singular values and their role in efficient low rank approximations.

The compressed sensing part explores sparse optimization techniques including Basis Pursuit, LASSO, augmented Lagrangian methods, ADMM (Alternating Direction Method of Multipliers), and soft thresholding algorithms. These methods are important for recovering sparse signals from incomplete measurements while reducing computational complexity.

Finally, the dissertation highlights recent developments and applications of low rank matrix theory and compressed sensing in image processing, machine learning, optimization, and high-dimensional data analysis.

Contents

1 Historical Background **3**
1.1 Introduction . . . 3
1.2 Historical Background of Low Rank Matrix Theory . . . 3
1.3 Historical Background of Compressed Sensing . . . 4
1.4 Importance of Low Rank Matrix Theory and Compressed Sensing . . . 5
1.5 Contributors of Low Rank Matrix Theory and Compressed Sensing . . . 5
1.5.1 Foundational Contributors in Low Rank Matrix Theory . . . 6
1.5.2 Pioneers of Compressed Sensing . . . 6
1.5.3 Contributors to Matrix Completion and Optimization . . . 6
1.5.4 Modern Contributions in Compressed Sensing . . . 6
1.6 Key Developments in Low Rank Matrix Theory and Compressed Sensing 6
1.7 Purpose of the Dissertation . . . 8

2 Preliminaries **10**

3 Matrix Perturbation and Low-Rank Approximation **12**
3.1 Changes in A^{-1} from Changes in A . . . 13
3.1.1 Rank-One Inverse (3×3 Case) . . . 13
3.1.2 Higher Rank Update: Inverting $M = I - UV^T$ . . . 14
3.1.3 Rank-2 Update . . . 15
3.2 Sherman–Morrison–Woodbury Formula . . . 16
3.3 Updating Least Squares (Recursive Least Squares) . . . 18
3.4 The Kalman Filter . . . 19
3.5 Quasi-Newton Update Methods . . . 19
3.6 Interlacing Eigenvalues and Low-Rank Signals . . . 20
3.7 Secular Equation and Graphical Interlacing . . . 25
3.8 Saddle Points and the Courant–Fischer Theorem . . . 26
3.9 Rapidly Decaying Singular Values . . . 28
3.9.1 NUDFT and the Vandermonde Matrix . . . 28
3.9.2 Chebyshev Expansions and Near-Low-Rank Approximation . . . 29
3.9.3 The Vandermonde Matrix . . . 29
3.9.4 The Vandermonde Matrix . . . 30
3.9.5 The Sylvester Equation and Low-Displacement Rank . . . 31
3.10 The Krylov Matrix and ADI Method . . . 32

4 Compressed Sensing **34**
4.1 Split Algorithms for $\ell^2 + \ell^1$ . . . 34
4.1.1 Basis Pursuit and the ℓ^1 Norm . . . 34

4.1.2 LASSO (Least Absolute Shrinkage and Selection Operator) 35
4.2 ADMM: Alternating Direction Method of Multipliers 38
4.3 ADMM:(Alternating Direction Method of Multipliers) 41
4.4 Proximal Algorithms and the Prox Operator 43
4.5 Split Bregman and Sparse Kaczmarz 45
4.6 Bounded Variation: ℓ^1 Norm for Gradients 46

5 Literature Review: Comparative Scholarly Analysis 47
5.1 Overview 47
5.2 Paper 1 — Güttel, Nakatsukasa, Webb & Bloor Riley (2024) 47
5.3 Paper 2 — Clouâtre, Klippenstein & Slevinsky (2023/2025) 48
5.4 Paper 3 — Zhang et al. (2024) 49
5.5 Paper 4 — Abbe, Fan, Yang & Zhong (2023, JMLR) 50
5.6 Paper 5 — Chen, Huang, Jia & Wang (2023) 51
5.7 Comparative Synthesis and Critical Analysis 52
5.8 Conclusion 53

Chapter 1

Historical Background

1.1 Introduction

Low rank matrix theory and compressed sensing are fundamental areas of modern applied mathematics that address the problem of efficient representation and recovery of high-dimensional data. In many practical situations, data appears to be large and complex; however, it often possesses an inherent structure that can be exploited to simplify its representation. Low rank matrix theory is concerned with approximating a given matrix by another matrix of lower rank, thereby reducing dimensionality while preserving essential information. This idea is particularly useful in applications such as data compression, image processing, and recommendation systems, where large datasets can be efficiently represented using low rank structures. Compressed sensing, on the other hand, is a relatively recent development in signal processing that focuses on the recovery of sparse signals from a limited number of linear measurements. Unlike traditional sampling methods, which require a large number of observations, compressed sensing utilizes the sparsity of signals to reconstruct them accurately from significantly fewer samples. A key aspect of both theories is the concept of recovering structured data from incomplete or reduced information. While compressed sensing deals with sparse vectors, low rank matrix theory extends similar principles to matrices by exploiting low-dimensional structures. The study of these fields has led to significant advancements in areas such as signal processing, machine learning, and optimization. Their ability to reduce complexity while maintaining accuracy makes them essential tools in handling large-scale data.

1.2 Historical Background of Low Rank Matrix Theory

The concept of low rank matrix theory originates from classical linear algebra, where the notion of matrix rank was introduced to measure the intrinsic dimensionality of a linear transformation. Matrix rank provides information about the number of linearly independent rows or columns in a matrix and plays a fundamental role in understanding the structure of matrices. In the early 20th century, mathematicians began investigating methods to approximate complex matrices by simpler ones with lower rank. A significant milestone in this direction was achieved by Eckart and Young, who established

the Eckart–Young theorem. This theorem states that the best approximation of a given matrix by another matrix of lower rank can be obtained through its singular value decomposition (SVD). This result provided a rigorous mathematical foundation for low rank approximation and became a cornerstone of modern numerical linear algebra. With the advancement of computational methods in the mid to late 20th century, low rank matrix approximation gained importance in various scientific and engineering applications. Researchers observed that many real-world datasets, despite being high-dimensional, often lie close to low-dimensional subspaces. This observation led to the development of efficient techniques for data compression and dimensionality reduction. In later years, the focus shifted towards more complex problems such as matrix completion, where only a subset of the entries of a matrix is known, and the goal is to recover the entire matrix under the assumption that it has low rank. This problem gained widespread attention due to its applications in recommendation systems, image processing, and machine learning. Furthermore, the development of optimization techniques, particularly convex optimization, played a crucial role in advancing low rank matrix theory. Methods such as nuclear norm minimization were introduced as tractable alternatives to rank minimization, which is generally a non-convex and computationally difficult problem. Thus, the historical development of low rank matrix theory reflects a transition from classical theoretical concepts to modern computational techniques aimed at efficiently handling large-scale and high-dimensional data.

1.3 Historical Background of Compressed Sensing

Compressed sensing is a relatively recent development in signal processing and applied mathematics that emerged in the early 21st century. It was developed as a response to the limitations of traditional sampling methods based on the Nyquist–Shannon sampling theorem, which requires a large number of samples to accurately reconstruct a signal. In many practical applications, this leads to inefficiency, especially when the underlying signal contains redundancy. The fundamental idea behind compressed sensing is that many natural signals are sparse or compressible in some basis, meaning that they can be represented using only a small number of significant coefficients. This observation motivated researchers to explore whether it is possible to reconstruct such signals from fewer measurements than traditionally required. A major breakthrough in this area was achieved through the pioneering work of Donoho, Candès, and Tao in the early 2000s. Their research demonstrated that sparse signals can be recovered exactly from a limited number of linear measurements by solving suitable optimization problems. This marked a significant shift in signal processing, introducing the idea that sampling and compression can be performed simultaneously. The development of compressed sensing relies on key mathematical concepts such as sparsity, incoherence, and optimization. One of the central approaches involves replacing the non-convex problem of minimizing the number of non-zero components with a convex optimization problem, typically based on L_1 minimization. This made the recovery process computationally feasible and theoretically well-founded. Another important development in the theory of compressed sensing is the introduction of conditions that guarantee accurate signal recovery. In particular, the Restricted Isometry Property (RIP) was introduced to describe how well a measurement matrix preserves the structure of sparse signals. Although verifying such conditions in practice can be challenging, they provide strong theoretical support for the effectiveness

of compressed sensing. Over time, compressed sensing has found numerous applications in areas such as medical imaging, communication systems, and data acquisition. For example, in magnetic resonance imaging (MRI), compressed sensing techniques allow faster data acquisition while maintaining high-quality image reconstruction. Thus, the historical development of compressed sensing represents a major advancement in the efficient acquisition and reconstruction of signals, transforming traditional approaches and enabling new possibilities in modern data processing.

1.4 Importance of Low Rank Matrix Theory and Compressed Sensing

Low rank matrix theory and compressed sensing have become essential tools in modern applied mathematics due to their ability to efficiently represent, process, and recover high-dimensional data. Their importance lies in both theoretical advancements and wide-ranging practical applications. Low rank matrix theory plays a crucial role in simplifying complex datasets by exploiting their inherent low-dimensional structure. Many real-world data matrices, although large in size, can be well approximated by matrices of much lower rank. This property allows significant reduction in storage requirements and computational complexity. As a result, low rank methods are widely used in data compression, image processing, recommendation systems, and machine learning. They also form the basis for dimensionality reduction techniques, which are important in handling large-scale data. Compressed sensing, on the other hand, has transformed the traditional approach to signal acquisition and reconstruction. By utilizing the sparsity of signals, compressed sensing enables accurate recovery from a limited number of measurements. This reduces the need for large data samples, leading to faster data acquisition and lower storage costs. The ability to reconstruct signals from incomplete data has made compressed sensing highly valuable in applications such as medical imaging, wireless communication, and sensor networks. Another important aspect of both theories is their reliance on optimization techniques. Problems in low rank matrix approximation and sparse signal recovery are often formulated as optimization problems, which has led to the development of efficient algorithms and computational methods. This connection has strengthened the relationship between applied mathematics, optimization theory, and data science. Furthermore, both low rank matrix theory and compressed sensing contribute significantly to modern technological advancements. Their applications in artificial intelligence, pattern recognition, and big data analytics highlight their practical relevance in solving real-world problems. In summary, the importance of low rank matrix theory and compressed sensing lies in their ability to reduce complexity, efficiently utilize data, and provide accurate reconstruction from limited information. These properties make them indispensable tools in contemporary scientific and engineering research.

1.5 Contributors of Low Rank Matrix Theory and Compressed Sensing

The development of low rank matrix theory and compressed sensing has been driven by several key researchers whose contributions have shaped both the theoretical and applied aspects of these fields.

1.5.1 Foundational Contributors in Low Rank Matrix Theory

- **Eckart and Young** Introduced the Eckart–Young theorem, which provides the best low rank approximation of a matrix using singular value decomposition.
- **Gene H. Golub** Made significant contributions to numerical linear algebra, particularly in the computation and application of singular value decomposition.
- **William Kahan** Contributed to the development of stable numerical methods for matrix computations, which are essential for low rank approximations.

1.5.2 Pioneers of Compressed Sensing

- **David Donoho** Established the theoretical foundation of compressed sensing and demonstrated that sparse signals can be recovered from limited measurements.
- **Emmanuel Candès** Developed mathematical guarantees for sparse recovery and introduced L_1 minimization techniques for signal reconstruction.
- **Terence Tao** Contributed to the theoretical framework of compressed sensing, including the development of the Restricted Isometry Property.

1.5.3 Contributors to Matrix Completion and Optimization

- **Maryam Fazel** Introduced nuclear norm minimization as a convex relaxation for rank minimization problems.
- **Benjamin Recht** Worked on matrix completion problems, showing that low rank matrices can be recovered from incomplete data using convex optimization.
- **Pablo Parrilo** Contributed to optimization techniques that are widely used in low rank matrix recovery problems.

1.5.4 Modern Contributions in Compressed Sensing

- **Richard Baraniuk** Contributed to practical implementations and applications of compressed sensing in signal processing.
- **Justin Romberg** Worked on reconstruction algorithms and theoretical guarantees in compressed sensing.
- **Jelani Nelson** Known for contributions to randomized algorithms and dimensionality reduction related to sparse recovery.

1.6 Key Developments in Low Rank Matrix Theory and Compressed Sensing

The fields of low rank matrix theory and compressed sensing have evolved significantly over time through several important theoretical and computational developments. Some of the key developments are summarized below:

- **Singular Value Decomposition (SVD)** The introduction and development of singular value decomposition provided a fundamental tool for analyzing matrices and obtaining their optimal low rank approximations. It forms the backbone of modern low rank matrix theory.
- **Low Rank Approximation Theory** The formulation of methods to approximate high-dimensional matrices by lower rank matrices led to efficient techniques for data compression and noise reduction.
- **Matrix Completion Problem** The study of recovering a low rank matrix from incomplete observations became a major area of research, particularly due to its applications in recommendation systems and machine learning.
- **Convex Relaxation Techniques** Since rank minimization is a non-convex problem, convex relaxation methods such as nuclear norm minimization were developed to make the problem computationally tractable.
- **Emergence of Compressed Sensing** The introduction of compressed sensing in the early 2000s marked a major breakthrough, showing that sparse signals can be recovered from a small number of measurements.
- L_1 **Minimization Methods** The replacement of L_0 minimization with L_1 minimization provided an efficient and practical approach for sparse signal recovery.
- **Restricted Isometry Property (RIP)** The development of RIP provided theoretical guarantees for accurate reconstruction of sparse signals, becoming a central concept in compressed sensing theory.
- **Development of Efficient Algorithms** Various algorithms such as Basis Pursuit, Orthogonal Matching Pursuit (OMP), and Singular Value Thresholding (SVT) were developed for practical implementation of low rank and sparse recovery problems.
- **Applications in Modern Technology** The application of these theories in areas such as medical imaging, communication systems, data science, and machine learning has significantly expanded their importance and impact.

Problems in Low Rank Matrix Theory and Compressed Sensing

Despite their wide applicability and theoretical significance, low rank matrix theory and compressed sensing involve several challenges and limitations. These problems arise due to computational complexity, practical constraints, and theoretical difficulties.

- **Non-Convexity of Rank Minimization** In low rank matrix theory, minimizing the rank of a matrix is a non-convex problem. Such problems are generally difficult to solve and do not guarantee a unique or globally optimal solution.
- **Computational Complexity** Many problems related to low rank approximation and sparse recovery are computationally expensive, especially for large-scale datasets. Efficient implementation remains a major challenge.

- **NP-Hard Nature of Rank Minimization** The exact rank minimization problem is known to be NP-hard, making it impractical to solve directly for real-world applications. This necessitates the use of approximation techniques.
- **Dependence on Convex Relaxation** Techniques such as nuclear norm minimization and L_1 minimization are used as approximations. However, these may not always produce exact or optimal solutions.
- **Design of Measurement Matrices** In compressed sensing, the choice of measurement matrix plays a crucial role in accurate signal recovery. Designing matrices that satisfy theoretical conditions is often difficult in practice.
- **Restricted Isometry Property (RIP) Verification** Although RIP provides theoretical guarantees for signal recovery, verifying whether a given matrix satisfies this property is computationally challenging.
- **Sensitivity to Noise** In practical applications, measurements are often corrupted by noise. This affects the accuracy of both low rank matrix recovery and sparse signal reconstruction.
- **Incomplete or Missing Data** Recovering data from incomplete observations is inherently difficult and may lead to instability or incorrect reconstruction if assumptions are not satisfied.
- **Trade-off Between Accuracy and Efficiency** Many algorithms provide high accuracy but require significant computational resources, leading to a trade-off between performance and efficiency.

1.7 Purpose of the Dissertation

The purpose of this dissertation is to study and analyze the fundamental concepts, theoretical developments, and applications of low rank matrix theory and compressed sensing. These areas play a significant role in modern applied mathematics, particularly in the efficient representation and recovery of high-dimensional data. This dissertation aims to provide a clear understanding of low rank matrix structures and sparse signal representations, along with the mathematical techniques used for their analysis. Special emphasis is given to the study of matrix approximation, matrix completion, and sparse recovery methods. Another objective of this work is to explore the connection between low rank matrix theory and compressed sensing, highlighting how both fields address the problem of recovering structured data from incomplete information. Furthermore, the dissertation seeks to examine important algorithms and optimization techniques used in these areas, as well as their practical applications in fields such as signal processing, machine learning, and data science. Overall, this study aims to present a systematic and comprehensive overview of low rank matrix theory and compressed sensing, emphasizing both their theoretical importance and real-world relevance.

Motivation and Objectives of the Research

Motivation: In recent years, the rapid growth of data in various fields such as science, engineering, and technology has created a need for efficient methods to store, process, and analyze high-dimensional information. Many real-world datasets are large in size but possess an underlying structure, such as sparsity or low dimensionality, which can be exploited to simplify their representation. Low rank matrix theory and compressed sensing provide powerful mathematical tools to address these challenges. The motivation for this research arises from the need to understand how complex data can be efficiently represented using low rank structures and how sparse signals can be accurately recovered from limited measurements. These concepts are not only theoretically interesting but also highly relevant in practical applications such as image processing, signal reconstruction, and machine learning.

Objectives:

The main objectives of this research are as follows:

- To study the fundamental concepts of low rank matrix theory and compressed sensing.
- To understand the mathematical principles behind matrix approximation, sparse representation, and data recovery.
- To analyze the relationship between low rank structures and sparse signal recovery.
- To explore important methods such as matrix completion and L_1 minimization techniques.
- To examine key algorithms used in low rank and compressed sensing problems.
- To investigate the applications of these theories in areas such as signal processing, data science, and machine learning.

Chapter 2

Preliminaries

In this chapter, we present the fundamental mathematical concepts that are essential for the study of low rank matrix theory and compressed sensing. These concepts are mainly drawn from linear algebra and provide the necessary foundation for the subsequent chapters.

Vector Spaces and Linear Independence

A vector space V over a field R is a set equipped with two operations, namely vector addition and scalar multiplication, satisfying the standard axioms of associativity, commutativity, distributivity, and existence of identity elements. A finite set of vectors $\{v_1, v_2, \ldots, v_n\}$ in V is said to be linearly independent if

$$a_1v_1 + a_2v_2 + \cdots + a_nv_n = 0$$

implies that

$$a_1 = a_2 = \cdots = a_n = 0.$$

Explanation: Linear independence ensures that no vector in the set can be expressed as a combination of the others. This concept is fundamental in determining the dimension and basis of a vector space. **Example:** The vectors $(1, 0)$ and $(0, 1)$ in R^2 are linearly independent, whereas the vectors $(1, 2)$ and $(2, 4)$ are linearly dependent.

Matrix Rank

The rank of a matrix $A \in R^{m\times n}$ is defined as the maximum number of linearly independent rows or columns of A. It is denoted by $\text{rank}(A)$. **Explanation:** The rank provides a measure of the intrinsic dimensionality of the matrix. A low rank matrix contains redundant information and can be represented in a lower-dimensional form. **Example:** Consider the matrix

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

Since the second row is a multiple of the first, $\text{rank}(A) = 1$.

Norms

A norm on a vector space is a function that assigns a non-negative real number to each vector, representing its magnitude. For a vector $x = (x_1, x_2, \ldots, x_n)$, common norms are defined as follows:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|x\|_2 = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}.$$

For a matrix $A = (a_{ij})$, the Frobenius norm is given by

$$\|A\|_F = \left(\sum_{i,j} |a_{ij}|^2 \right)^{1/2}.$$

Explanation: Norms are used to measure the size of vectors and matrices and play a crucial role in optimization problems.

Singular Value Decomposition (SVD)

Let $A \in R^{m \times n}$. Then A can be decomposed as

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

where U and V are orthogonal matrices and Σ is a diagonal matrix containing the singular values of A. **Explanation:** SVD reveals the internal structure of a matrix and is widely used to obtain low rank approximations by retaining only the largest singular values. **Example:** If a matrix has only one non-zero singular value, then it is a rank one matrix.

Sparsity

A vector $x \in R^n$ is said to be sparse if most of its components are zero. The sparsity level is defined as the number of non-zero elements in the vector. **Example:** The vector $x = (1, 0, 0, 5, 0)$ is sparse since it contains only two non-zero entries. **Explanation:** Sparsity is a key concept in compressed sensing, as it allows signals to be represented efficiently using only a few significant components.

Linear Measurement Model

In compressed sensing, the observed data is obtained through linear measurements of the form

$$y = Ax,$$

where $x \in R^n$ is the original signal, $A \in R^{m \times n}$ is the measurement matrix, and $y \in R^m$ is the measurement vector. **Explanation:** When $m < n$, the system is underdetermined, and recovering x becomes challenging. However, if x is sparse, it can still be recovered using appropriate methods.

Optimization Perspective

Many problems in low rank matrix theory and compressed sensing can be formulated as optimization problems. For instance, one seeks to minimize rank or sparsity subject to certain constraints. Since direct minimization of rank or sparsity is difficult, alternative approaches such as convex optimization are used.

Conclusion

The concepts discussed in this chapter provide the mathematical foundation required to study low rank matrix theory and compressed sensing. A clear understanding of these preliminaries is essential for the advanced topics discussed in the subsequent chapters.

Chapter 3

Matrix Perturbation and Low-Rank Approximation

Overview

This chapter investigates three fundamental categories of low-rank matrices that arise naturally in mathematics, engineering, data science, and signal processing. Understanding the structure of such matrices—and how to exploit it—is one of the most powerful ideas in modern linear algebra. The three categories are:

1. **Truly low-rank matrices:** Matrices of the form

   $$uv^T$$

   (rank = 1) or

   $$UV^T$$

   (rank = k), where all the information is compressed into a small number of directions.

2. **Matrices with rapidly (exponentially) decaying singular values:** These matrices are technically full rank but are extremely well approximated by a low-rank matrix. They have what is called "low effective rank."

3. **Incomplete matrices:** Matrices with many missing entries that can be completed via a low-rank assumption. This is the foundation of recommendation systems like Netflix.

The first type of matrix is not invertible (since rank $< n$). The second type is invertible in theory but extremely ill-conditioned in practice—small perturbations to the input produce enormous changes in the output. A famous example is the Hilbert matrix, whose (i, j) entry is

$$\frac{1}{i+j-1}$$

The third type is approached through matrix completion, formulated as a minimization problem over all possible choices of the missing entries:

$$\text{Minimize } \|A\|_N$$

over all possible choices of the missing entries. Here $\|A\|_N$ denotes the nuclear norm of A, which is the sum of all singular values of A. Minimizing the nuclear norm gives a

well-posed convex problem that effectively replaces the nonconvex problem of minimizing rank directly. The nuclear norm plays the same role for matrices that the ℓ^1 norm plays for vectors: it promotes low-rank (respectively, sparse) solutions. The rank of a matrix corresponds, in a deep way, to the number of nonzero entries in a vector. Just as a sparse vector has very few nonzeros, a low-rank matrix captures all its information in a small number of directions. This analogy—from vectors to matrices, from sparsity to low-rankness—drives everything in this chapter.

3.1 Changes in A^{-1} from Changes in A

Suppose we subtract a low-rank matrix from an invertible matrix A. This section derives an exact formula for the resulting change in

$$A^{-1}$$

This formula, called the matrix inversion lemma or the Sherman–Morrison–Woodbury formula, is one of the most useful and elegant results in all of numerical linear algebra.

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}$$

The formula shows that a low-rank update to a matrix produces only a low-rank correction to the inverse. This is enormously important computationally because solving systems involving low-rank perturbations can be done efficiently without recomputing the full inverse.

3.1.1 Rank-One Inverse (3×3 Case)

Let

$$u = v = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

so that

$$v^T u = 3.$$

The matrix

$$M = I - uv^T$$

is given by

$$M = I - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [1\ 1\ 1] = I - J_3,$$

where J_3 denotes the all-ones 3×3 matrix. Since

$$v^T u = 3 \neq 1,$$

the matrix M is invertible. Using the rank-one inverse formula,

$$M^{-1} = I + \frac{uv^T}{1 - v^T u}.$$

Substituting the values,

$$M^{-1} = I + \frac{\begin{bmatrix}1\\1\\1\end{bmatrix}[1\ 1\ 1]}{1-3} = I - \frac{1}{2}J_3.$$

To verify,

$$MM^{-1} = (I - J_3)\left(I - \frac{1}{2}J_3\right).$$

Expanding,

$$= I - \frac{1}{2}J_3 - J_3 + \frac{1}{2}J_3^2.$$

Since

$$J_3^2 = 3J_3,$$

we obtain

$$I - \frac{1}{2}J_3 - J_3 + \frac{3}{2}J_3 = I.$$

Hence the inverse formula is verified. Geometrically, the matrix M collapses the all-ones direction, while M^{-1} restores it scaled by $-1/2$.

3.1.2 Higher Rank Update: Inverting $M = I - UV^T$

We now generalize the rank-one perturbation to a rank-k perturbation. Let U and V be matrices of size $n \times k$. Then UV^T is an $n \times n$ matrix having rank at most k. The matrix is defined as

$$M = I_n - UV^T.$$

Its inverse is

$$M^{-1} = I_n + U(I_k - V^TU)^{-1}V^T.$$

This formula replaces the scalar quantity

$$1 - v^Tu$$

from the rank-one case by the $k \times k$ matrix

$$I_k - V^TU.$$

A major computational advantage arises when $k \ll n$. Instead of computing the inverse of a large $n \times n$ matrix, we only need to invert a much smaller $k \times k$ matrix. To verify the formula, compute

$$MM^{-1} = (I_n - UV^T)\left(I_n + U(I_k - V^TU)^{-1}V^T\right).$$

Expanding,

$$= I_n - UV^T + U(I_k - V^TU)^{-1}V^T - UV^TU(I_k - V^TU)^{-1}V^T.$$

Factorizing,

$$= I_n - UV^T + U\left[I_k - V^TU\right](I_k - V^TU)^{-1}V^T.$$

Since

$$(I_k - V^T U)(I_k - V^T U)^{-1} = I_k,$$

we obtain

$$MM^{-1} = I_n.$$

Thus the inverse formula is proved. An important related identity is

$$U(I_k - V^T U) = (I_n - UV^T)U.$$

This identity allows movement between the large and small matrix inverses and is often called a push-through identity. A related determinant identity is

$$\det(I_n - UV^T) = \det(I_k - V^T U).$$

The two matrices, although different in size ($n \times n$ versus $k \times k$), possess the same determinant. This follows from the fact that both matrices share the same nonzero eigenvalues.

3.1.3 Rank-2 Update

Let

$$n = 3, \qquad k = 2.$$

Choose

$$U = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \qquad V = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ 1 & 0 \end{bmatrix}.$$

Then

$$UV^T = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and

$$V^T U = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Now compute

$$I - V^T U = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}.$$

Since

$$\det(I - V^T U) = 1,$$

the matrix is invertible. Its inverse is

$$(I - V^T U)^{-1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

Using the higher-rank update formula,

$$M^{-1} = I_3 + U(I - V^T U)^{-1} V^T.$$

Thus M^{-1} becomes a rank-2 correction to the identity matrix I_3. Instead of inverting a 3×3 matrix directly, we only invert a 2×2 matrix, leading to substantial computational savings.

3.2 Sherman–Morrison–Woodbury Formula

The most general version begins with an arbitrary invertible matrix A and perturbs it using a rank-k matrix UV^T. Define

$$M = A - UV^T.$$

If M is invertible, then the Sherman–Morrison–Woodbury formula states that

$$M^{-1} = (A - UV^T)^{-1} = A^{-1} + A^{-1}U(I - V^T A^{-1}U)^{-1}V^T A^{-1}.$$

For the special case $k = 1$, this reduces to the rank-one perturbation formula derived earlier. To prove the formula, consider the extended block matrix

$$E = \begin{bmatrix} A & U \\ V^T & I_k \end{bmatrix}.$$

The matrix E is invertible whenever

$$M = A - UV^T$$

is invertible. Using block elimination, two different expressions for E^{-1} can be obtained. Comparing the $(1,1)$-blocks of these expressions yields the Sherman–Morrison–Woodbury identity. Specifically, one side produces

$$M^{-1},$$

while the other gives

$$A^{-1} + A^{-1}U(I - V^T A^{-1}U)^{-1}V^T A^{-1}.$$

Hence the formula follows. The Sherman–Morrison–Woodbury formula plays a major role in numerical linear algebra and appears in applications such as the Kalman filter, recursive least squares, quasi-Newton optimization methods, engineering computations, and statistics. There are also several important matrix identities closely related to the

Sherman–Morrison–Woodbury formula.

The Derivative of A^{-1}

The Sherman–Morrison–Woodbury formula has a beautiful infinitesimal version. Suppose $A = A(t)$ is a matrix that changes smoothly with time. Its inverse $A^{-1}(t)$ also changes with t. We wish to determine the derivative:

$$\frac{dA^{-1}}{dt}$$

Start from the fundamental identity:

$$B^{-1} - A^{-1} = B^{-1}(A - B)A^{-1}$$

This identity is easy to verify: multiply both sides on the left by B and on the right by A. If $A - B$ has rank 1 (or rank k), then $B^{-1} - A^{-1}$ also has rank 1 (or rank k). This is a finite version of the derivative formula. Now let

$$B = A + \Delta A$$

where ΔA is a small perturbation. Then

$$B^{-1} - A^{-1} = (A + \Delta A)^{-1} - A^{-1}$$

and the right-hand side of equation (11) becomes

$$-(A + \Delta A)^{-1} \Delta A A^{-1}$$

Dividing by Δt and letting $\Delta t \to 0$, we obtain the following theorem.

Derivative of the Matrix Inverse

If $A(t)$ is a smooth matrix-valued function of t, and $A(t)$ is invertible for all t, then

$$\frac{dA^{-1}}{dt} = -A^{-1} \left(\frac{dA}{dt} \right) A^{-1}$$

This is the matrix analogue of the scalar derivative:

$$\frac{d}{dt} \left(\frac{1}{t} \right) = -\frac{1}{t^2}$$

The formula

$$\frac{dA^{-1}}{dt} = -A^{-1} \left(\frac{dA}{dt} \right) A^{-1}$$

is fundamentally important. Unlike the scalar case, the order of multiplication matters. The matrix A^{-1} appears on both sides of $\frac{dA}{dt}$ because matrices generally do not commute.

Example– Verifying the Derivative Formula

Let

$$A(t) = tI$$

where I is the identity matrix. Then

$$A^{-1}(t) = \frac{1}{t} I$$

We have

$$\frac{dA}{dt} = I$$

and

$$\frac{dA^{-1}}{dt} = -\frac{1}{t^2} I$$

Using formula (12):

$$-A^{-1}\left(\frac{dA}{dt}\right)A^{-1} = -\left(\frac{1}{t}I\right)(I)\left(\frac{1}{t}I\right) = -\frac{1}{t^2}I$$

which verifies the result. For the general case:

$$\frac{d(A^2)}{dt} = A\left(\frac{dA}{dt}\right) + \left(\frac{dA}{dt}\right)A \neq 2A\left(\frac{dA}{dt}\right)$$

unless A commutes with $\frac{dA}{dt}$.

3.3 Updating Least Squares (Recursive Least Squares)

A major application of the Sherman–Morrison–Woodbury formula appears in recursive least squares. The standard least squares problem minimizes

$$\|b - Ax\|^2$$

and yields the normal equations

$$A^T A\hat{x} = A^T b$$

Suppose A has m rows and a new measurement arrives, giving a new row r^T. The updated least squares problem involves

$$A^T A + rr^T$$

which is a rank-one update. We do not want to solve the full normal equations from scratch. Instead, we use the Woodbury formula:

$$\left(A^T A + rr^T\right)^{-1} = (A^T A)^{-1} - c(A^T A)^{-1} rr^T (A^T A)^{-1}$$

where

$$c = \frac{1}{1 + r^T (A^T A)^{-1} r}$$

The scalar c is computed by first solving

$$(A^T A)y = r^T$$

and then setting

$$c = \frac{1}{1 + r^T y}$$

This is recursive least squares: each new data point updates the solution $\hat{x}$ without re-solving the entire system. The same idea extends to M new rows, forming the basis of many online learning algorithms.

3.4 The Kalman Filter

The Kalman filter extends recursive least squares to dynamic problems. The state vector x is not static; it changes with time according to a state equation. A GPS satellite's position $x(t)$ moves by

$$\Delta x = v\Delta t$$

At each step there are two updates:

1. **Prediction:** Use the state equation to predict the next state.
2. **Correction:** Use the new measurement b_{m+1} to correct the prediction through the Kalman gain matrix K.

The update equation is

$$x_{\text{new}} = x_{\text{state}} + K(b_{m+1} - rx_{\text{state}})$$

where K is the Kalman gain matrix. The gain matrix K is constructed from A, r, and the covariance matrices V_{state} and V_b. When the new measurement b_{m+1} exactly agrees with the prediction rx_{state}, there is zero correction. The Kalman filter is optimal in the sense that it minimizes the variance of the estimation error at each step.

3.5 Quasi-Newton Update Methods

The Sherman–Morrison formula appears in another important context: quasi-Newton methods for solving

$$f(x) = 0$$

or minimizing $F(x)$. Newton's method requires the Jacobian

$$J(x) = \left[\frac{\partial f_i}{\partial x_j}\right]$$

and performs the update

$$x_{\text{new}} = x_{\text{old}} - J(x_{\text{old}})^{-1} f(x_{\text{old}})$$

Recomputing J at every step is expensive for large n. Quasi-Newton methods instead maintain an approximate Jacobian

$$J_{\text{new}} = J(x_{\text{old}}) + \Delta J$$

updated through a rank-one or rank-two correction.
The key condition is

$$J_{\text{new}}(x_{\text{new}} - x_{\text{old}}) = f_{\text{new}} - f_{\text{old}}$$

This ensures that J remains consistent with the observed change in f. The Sherman–Morrison formula updates J^{-1} directly without recomputing it. The celebrated BFGS correction is a rank-two quasi-Newton update discovered independently by four authors. It remains one of the most widely used optimization methods in modern numerical software.

3.6 Interlacing Eigenvalues and Low-Rank Signals

When the matrix A changes by a low-rank perturbation, how do its eigenvalues and singular values change? The previous section addressed the change in A^{-1}. Now we ask the analogous question about eigenvalues. The answer is remarkably structured: the eigenvalues of the perturbed matrix interlace the eigenvalues of the original matrix.

The Interlacing Property

Interlacing Theorem (Rank-One Symmetric Update) Let S be a symmetric $n \times n$ matrix with eigenvalues

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$$

Let

$$S + \theta u u^T$$

be a rank-one positive perturbation with $\theta > 0$, having new eigenvalues

$$z_1 \geq z_2 \geq \cdots \geq z_n$$

Then the new eigenvalues interlace the old eigenvalues:

$$z_1 \geq \lambda_1 \geq z_2 \geq \lambda_2 \geq \cdots \geq z_n \geq \lambda_n$$

Equivalently, each new eigenvalue z_i lies between λ_{i-1} and λ_i. The word *interlacing* describes the alternating structure: old and new eigenvalues interleave along the real line. This result is intuitive because adding a positive semidefinite matrix

$$\theta u u^T$$

should increase eigenvalues, but no eigenvalue can jump past the next one above it. An important subtlety is that the increase in the largest eigenvalue can be dramatic. Suppose u is the second eigenvector q_2 of S. Then

$$S q_2 = \lambda_2 q_2$$

and

$$(S + C q_2 q_2^T) q_2 = \lambda_2 q_2 + C q_2 = (\lambda_2 + C) q_2$$

Thus the second eigenvalue jumps by C. The interlacing relation becomes

$$z_2 \leq z_1 = \lambda_2 + C$$

The new largest eigenvalue z_1 absorbs the large increase, while z_2 satisfies

$$z_2 \leq \lambda_1$$

Example- Interlacing with a 4×4 Matrix

Let

$$S = \text{diag}(5, 3, 1, -1)$$

with eigenvalues

$$\lambda_1 = 5, \quad \lambda_2 = 3, \quad \lambda_3 = 1, \quad \lambda_4 = -1$$

Add the rank-one perturbation

$$\theta u u^T$$

with

$$\theta = 2$$

and

$$u = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Then

$$S + \theta u u^T = \text{diag}(7, 3, 1, -1)$$

so the new eigenvalues are

$$z_1 = 7, \quad z_2 = 3, \quad z_3 = 1, \quad z_4 = -1$$

Check the interlacing condition:

$$z_1 = 7 \geq \lambda_1 = 5 \geq z_2 = 3 \geq \lambda_2 = 3 \geq z_3 = 1 \geq \lambda_3 = 1 \geq z_4 = -1 \geq \lambda_4 = -1$$

Now try

$$u = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Then the new matrix has eigenvalues

$$5, 5, 1, -1$$

so λ_2 and λ_1 effectively swap roles. If θ is very large, then

$$z_1 \gg \lambda_1$$

while

$$z_2 \leq \lambda_1 = 5$$

Thus the top eigenvalue grows dramatically, while the remaining eigenvalues stay bounded.

Derivative of an Eigenvalue

If $A(t)$ changes smoothly with t, its eigenvalues $\lambda(t)$ also change. We want to compute

$$\frac{d\lambda}{dt}.$$

The key ingredients are the three facts:

$$A(t)x(t) = \lambda(t)x(t)$$

$$y^T(t)A(t) = \lambda(t)y^T(t)$$

$$y^T(t)x(t) = 1$$

Here $x(t)$ is the right eigenvector of $A(t)$ and $y(t)$ is the left eigenvector (row of $(X^{-1})^T$ for A^T). The normalization

$$y^T x = 1$$

is essential. From these facts: **Formula Derivative of an Eigenvalue**

$$\lambda(t) = y^T(t)A(t)x(t)$$

and

$$\frac{d\lambda}{dt} = y^T(t)\left(\frac{dA}{dt}\right)x(t)$$

The derivative of an eigenvalue involves only the left and right eigenvectors and the derivative of A — not the derivatives of the eigenvectors themselves. To derive (4), take the derivative of

$$\lambda = y^T Ax$$

using the product rule. This gives three terms. The first and third terms involve

$$\frac{dy^T}{dt} \quad \text{and} \quad \frac{dx}{dt}$$

respectively, but these terms sum to zero because of the eigenvalue equations and the normalization

$$y^T x = 1.$$

$$\frac{d\lambda}{dt} = \left(\frac{dy^T}{dt}\right) Ax + y^T \left(\frac{dA}{dt}\right) x + y^T A \left(\frac{dx}{dt}\right)$$

$$= \lambda \frac{d}{dt}(y^T x) + y^T \left(\frac{dA}{dt}\right) x = y^T \left(\frac{dA}{dt}\right) x$$

Example- Computing $\frac{d\lambda}{dt}$ for a 2×2 Matrix Let

$$A(t) = \begin{bmatrix} 2t & 1 \\ 2t & 2 \end{bmatrix}$$

$$A(0) = \begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix}$$

Eigenvalue equation:

$$\lambda^2 - 2(1+t)\lambda + 2t = 0$$

so

$$\lambda = 1 + t \pm \sqrt{1+t^2}.$$

At $t = 0$:

$$\lambda_1 = 2, \qquad \lambda_2 = 0.$$

Derivatives:

$$\left.\frac{d\lambda_1}{dt}\right|_{t=0} = 1 + \frac{1}{\sqrt{1}} = 2.$$

Wait, let us use formula (4). Eigenvectors at $t = 0$:

$$y_1^T = [0 \quad 1],$$

$$x_1 = \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}$$

(normalized so that $y_1^T x_1 = 1$).

Also,

$$\frac{dA}{dt} = \begin{bmatrix} 2 & 0 \\ 2 & 0 \end{bmatrix}.$$

Therefore,

$$\frac{d\lambda_1}{dt} = y_1^T \left(\frac{dA}{dt} \right) x_1$$

$$= [0 \quad 1] \begin{bmatrix} 2 & 0 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} 1/2 \\ 1 \end{bmatrix}$$

$$= [0 \quad 1] \begin{bmatrix} 1 \\ 1 \end{bmatrix} = 1.$$

This confirms the formula gives the correct derivative without computing

$$\frac{d(\text{eigenvector})}{dt}.$$

Derivative of a Singular Value

For a non-square matrix A, the analogous formula involves singular values and singular vectors. If

$$Av = \sigma u$$

with

$$u^T u = v^T v = 1,$$

then:

$$u^T(t) \left(\frac{dA}{dt} \right) v(t) = \frac{d\sigma}{dt}$$

This beautiful formula says: the rate of change of a singular value equals the bilinear form of dA/dt with the left and right singular vectors. It is the direct generalization of the eigenvalue formula (4) to the non-symmetric case. **Note:** When $A(t)$ is symmetric positive definite,

$$\sigma(t) = \lambda(t)$$

and

$$u = v = y = x$$

Formulas (4) and (9) then coincide.

3.7 Secular Equation and Graphical Interlacing

The interlacing theorem has a beautiful graphical interpretation. For a symmetric matrix S with eigenvalues

$$\lambda_1, \ldots, \lambda_n$$

and a rank-one perturbation

$$\theta u u^T$$

the new eigenvalues z are determined by the secular equation:

$$\frac{1}{\theta} = u^T (zI - S)^{-1} u = \sum_k \frac{c_k^2}{z - \lambda_k}$$

Here

$$c_k = q_k^T u$$

the component of u in the direction of the k-th eigenvector q_k of S. The right-hand side is a sum of simple fractions (partial fractions), each with a pole at

$$z = \lambda_k$$

The graph of the right-hand side consists of n curves, each going from $-\infty$ to $+\infty$ between consecutive poles. The horizontal line

$$y = \frac{1}{\theta}$$

intersects each of these curves exactly once, giving the n solutions

$$z_1, \ldots, z_n$$

These intersection points visually demonstrate interlacing: each z_i is between λ_i and λ_{i-1}.

The Largest Eigenvalue of $S + T$

Subadditivity of Largest Eigenvalue

For symmetric matrices S and T,

$$\lambda_{\max}(S + T) \leq \lambda_{\max}(S) + \lambda_{\max}(T)$$

Proof:

$$\lambda_{\max}(S + T) = \max_x \frac{x^T (S + T) x}{x^T x}$$

$$\leq \max_x \frac{x^T S x}{x^T x} + \max_x \frac{x^T T x}{x^T x}$$

$$= \lambda_{\max}(S) + \lambda_{\max}(T)$$

Similarly,

$$\lambda_{\min}(S+T) \geq \lambda_{\min}(S) + \lambda_{\min}(T)$$

The more precise bounds are given by the Weyl inequalities:

$$\lambda_{i+j-1}(S+T) \leq \lambda_i(S) + \lambda_j(T)$$

Also,

$$\lambda_k(S) + \lambda_n(T) \leq \lambda_k(S+T) \leq \lambda_k(S) + \lambda_1(T)$$

These inequalities give tight control over all eigenvalues of $S+T$ in terms of individual eigenvalues of S and T. They are heavily used in random matrix theory, quantum information, and numerical analysis.

3.8 Saddle Points and the Courant–Fischer Theorem

The middle eigenvalues

$$\lambda_2, \ldots, \lambda_{n-1}$$

are saddle points of the Rayleigh quotient

$$R(x) = \frac{x^T S x}{x^T x}$$

The maximum and minimum are:

$$\text{Maximum: } \lambda_1 = R(q_1) = \frac{q_1^T S q_1}{q_1^T q_1}$$

$$\text{Minimum: } \lambda_n = R(q_n) = \frac{q_n^T S q_n}{q_n^T q_n}$$

The saddle points occur at

$$q_2, \ldots, q_{n-1}$$

where all partial derivatives of R are zero but R is neither a local maximum nor a local minimum. The rigorous characterization is the Courant–Fischer max-min principle.

Courant–Fischer Max-Min Theorem

For a symmetric matrix S and a matrix A,

$$\lambda_i(S) = \max_{\dim V = i} \min_{x \in V} \frac{x^T S x}{x^T x}$$

$$\sigma_i(A) = \max_{\dim W = i} \min_{x \in W} \frac{\|Ax\|}{\|x\|}$$

The i-th eigenvalue is the best possible minimum of $R(x)$ over all i-dimensional subspaces V. This elegant characterization explains why the interlacing theorem is true: removing the last row and column of S (giving S_{n-1}) forces x to be perpendicular to the last standard basis vector. This is equivalent to restricting the $(n-1)$-dimensional optimization in Courant–Fischer, and the result is that the $n-1$ eigenvalues

$$\alpha_i$$

of S_{n-1} interlace the n eigenvalues of S.

Example- Interlacing After Removing a Row/Column

Let

$$S = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

The eigenvalues are

$$\lambda = 3, 3, 0$$

Remove the last row and column:

$$S_{n-1} = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

The eigenvalues are

$$\alpha = 3, 1$$

Check interlacing:

$$\lambda_1 = 3 \geq \alpha_1 = 3 \geq \lambda_2 = 3 \geq \alpha_2 = 1 \geq \lambda_3 = 0$$

Interpretation: The 2×2 submatrix eigenvalues sit between those of the 3×3 matrix.

Interlacing of Singular Values

For rectangular matrices, the analogous result holds for singular values. If B is obtained from A by adding a matrix whose singular values are

$$\sigma_1(B - A) \leq \cdots \leq \sigma_r(B - A)$$

the Weyl inequalities give:

$$\sigma_{i+j-1}(A+B) \leq \sigma_i(A) + \sigma_j(B)$$

$$|\sigma_i(A+B) - \sigma_i(A)| \leq \|B\|$$

This Inequality is especially important: each singular value can change by at most the spectral norm

$$\|B\| = \sigma_1(B)$$

when B is added to A. This is the Weyl perturbation theorem for singular values.

The saddle points occur at

$$q_2, \ldots, q_{n-1}$$

where all partial derivatives of R are zero but R is neither a local maximum nor a local minimum. The rigorous characterization is the Courant–Fischer max-min principle.

3.9 Rapidly Decaying Singular Values

This section investigates the second major class of low-rank matrices: matrices with exponentially decaying singular values. Such matrices are full-rank in the strict sense, but their singular values decay so rapidly that only a few of them are numerically significant. These matrices have low effective rank.

The two canonical examples studied in detail are the Nonuniform Discrete Fourier Transform (NUDFT) matrix and the Vandermonde matrix. Both belong to a large family of structured matrices that satisfy a Sylvester equation, whose solution theory explains why their singular values decay exponentially.

3.9.1 NUDFT and the Vandermonde Matrix

The standard (uniform) DFT matrix F has entries

$$F_{jk} = e^{-2\pi ijk/N}, \qquad j,k = 0, \ldots, N-1.$$

This is a Vandermonde matrix. The interpolation points are

$$x_j = e^{-2\pi ij/N},$$

equally spaced on the unit circle. F is unitary:

$$F^*F = NI.$$

Its singular values are all equal to $\sqrt{N}$, so there is no decay. The nonuniform DFT (NUDFT) matrix U has entries

$$U_{jk} = e^{-2\pi i x_j k/N},$$

where x_j are not equally spaced. For the uniform case $U = F$, the matrix is perfectly orthogonal and all singular values are equal. For nonuniform spacing, U differs from F by a correction matrix A, defined entry-wise by

$$U_{jk} = A_{jk}F_{jk},$$

where

$$A_{jk} = e^{-2\pi ik(x_j - j/N)}.$$

The matrix A has entries of the form $e^{i\theta}$, all close to 1 if the x_j are close to j/N. Ruiz-Antolin and Townsend showed that A is nearly low rank:

$$A \approx y_1 z_1^T + \cdots + y_r z_r^T,$$

for a small number r. This implies

$$Uc \approx Y_1 F Z_1 c + \cdots + Y_r F Z_r c,$$

where Y_i and Z_i are diagonal $N \times N$ matrices. The fast NUDFT uses this decomposition. Instead of a direct $O(N^2)$ matrix-vector product, we use r fast $O(N \log N)$ DFTs.

3.9.2 Chebyshev Expansions and Near-Low-Rank Approximation

The mathematical mechanism behind the low-rank approximation of A is the Chebyshev polynomial expansion. This reflects a fundamental principle of numerical analysis.

Series Type	**Best Suited For**
Fourier series	Periodic functions on $\lvert\theta\rvert \leq \pi$
Chebyshev series	Nonperiodic functions on $\lvert x\rvert \leq 1$
Taylor series	Analytic functions near a point

3.9.3 The Vandermonde Matrix

The Chebyshev polynomials $T_n(x) = \cos(n \arccos x)$ form an orthogonal basis on $[-1, 1]$. They satisfy

$$T_n(\cos\theta) = \cos(n\theta).$$

The key property is that Chebyshev series converge exponentially fast for analytic functions on $[-1, 1]$. The number of terms needed to achieve 10^{-15} accuracy is much smaller than for Taylor series. For the NUDFT matrix, the function being approximated is

$$e^{-iyz},$$

for

$$z \in [z_{\min}, z_{\max}]$$

and

$$y \in [y_{\min}, y_{\max}].$$

This function is analytic, so its Chebyshev approximation truncates after

$$r \approx \log(1/\varepsilon)$$

terms. Hence the matrix A has rank

$$r \approx \log(1/\varepsilon),$$

which gives remarkable compression.

3.9.4 The Vandermonde Matrix

Given n distinct real numbers

$$x_1, \ldots, x_n,$$

the Vandermonde matrix is

$$V_{ij} = x_i^{j-1},$$

where rows are indexed by

$$i = 1, \ldots, n$$

and columns by

$$j = 0, \ldots, n-1.$$

The matrix V is invertible whenever the points

$$x_1, \ldots, x_n$$

are distinct. The equation

$$Vc = f$$

means finding coefficients

$$c_0, \ldots, c_{n-1}$$

of the polynomial

$$p(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

that interpolates f at the points

$$x_1, \ldots, x_n.$$

The Vandermonde matrix with complex points

$$x_j = e^{-2\pi i k/N},$$

equally spaced on the unit circle, becomes the DFT matrix F, which is perfectly orthogonal and whose singular values are all equal to

$$\sqrt{N}.$$

For real interpolation points

$$x_1, \ldots, x_n,$$

the situation is very different. The singular values of V decay exponentially:

$$\sigma_k \leq Ce^{-ak},$$

for constants $C, a > 0$. Thus V is highly ill-conditioned. The condition number

$$\kappa(V) = \frac{\sigma_1}{\sigma_n}$$

can become extremely large. This explains why polynomial interpolation at equally spaced points is numerically unstable for large n (the Runge phenomenon).

Example- Vandermonde Singular Value Decay For a Vandermonde matrix with

$$n = 50$$

equally spaced points on $[-1, 1]$,

$$\sigma_1 \approx 10^0,$$

$$\sigma_{10} \approx 10^{-5},$$

$$\sigma_{20} \approx 10^{-10},$$

$$\sigma_{50} \approx 10^{-15}.$$

Hence the condition number satisfies

$$\kappa(V) \approx 10^{15},$$

which means that V is numerically almost singular. Using Chebyshev points

$$x_k = \cos\left(\frac{\pi(2k-1)}{2n}\right)$$

greatly improves stability, and the interpolation remains well-conditioned.

3.9.5 The Sylvester Equation and Low-Displacement Rank

The unifying framework for all these matrices is provided by the Sylvester matrix equation. This is the "A, B, C test" developed by Beckermann and Townsend. **Sylvester Equation and Rapid Singular Value Decay** If

$$AX - XB = C$$

has rank r for normal matrices A, B with no shared eigenvalues, then the singular values of X decay at an exponential rate determined by A, B, and C.

$$\sigma_k(X) \leq C' e^{-ak}$$

for some constants C', $a > 0$. A matrix is called *normal* if

$$A^H A = A A^H$$

equivalently, A has orthogonal eigenvectors. Symmetric, skew-symmetric, unitary, and orthogonal matrices are all normal. The Sylvester test applies to all of them. For the Vandermonde matrix V with real points

$$x_1, x_2, \ldots, x_n,$$

the matrices A, B, and C in the Sylvester equation are

$$A = \text{diag}(x_1, \ldots, x_n)$$

$$B = \text{companion matrix of monic polynomial}$$

$$C = \text{rank-1 matrix}$$

A and B are normal matrices. Their eigenvalues are the x_i (real) and the roots of the characteristic polynomial, which lie on the unit circle. Since real numbers are generally different from unit-circle numbers, A and B have no shared eigenvalues. The Sylvester test passes, confirming exponential decay of singular values of V.

3.10 The Krylov Matrix and ADI Method

Definition- Krylov Matrix Given a matrix A $(n \times n)$ and a vector b, the Krylov matrix of order N is

$$K = [\, b \quad Ab \quad A^2b \quad \cdots \quad A^{N-1}b \,]$$

K is the basis of Krylov subspace methods such as the conjugate gradient method and GMRES. It also satisfies the Sylvester equation

$$AK - KB = C$$

with B equal to the companion matrix and C rank-1. Both the Vandermonde matrix V and the Krylov matrix K satisfy a Sylvester equation. Their singular value plots exhibit characteristic exponential decay, confirming extremely low effective rank. The Alternating Direction Implicit (ADI) method is an iterative method for solving the Sylvester equation

$$AX - XB = C.$$

At each step,

$$X_{j+\frac{1}{2}}(B - p_j I) = C - (A - p_j I)X_j$$

$$(A - q_j I)X_{j+1} = C - X_{j+\frac{1}{2}}(B - q_j I)$$

The shift parameters p_j and q_j are chosen by solving a Zolotarev rational approximation problem. Find the rational function

$$r(z) = \frac{p(z)}{q(z)}$$

that best approximates the function connecting the spectra of A and B.

Log-Rank and Applications

Definition- Log-Rank Family A family of matrices C_n has log-rank if

$$\|(C_n - E_n)\| < \varepsilon$$

for nearby matrices E_n that satisfy

$$\text{rank}(E_n) < c(\log n)^q$$

In other words, C_n can be approximated to precision ε by a matrix of rank at most

$$c(\log n)^q$$

Townsend showed that "log-rank" for C in the Sylvester test leads to "log-rank" for X. Udell and Townsend showed that log-rank is a very widespread property: many matrices arising in practice, including kernel matrices from machine learning, are log-rank.

Example 3.3.2 — RBF Kernel Matrix (Log-Rank) The Radial Basis Function (RBF) kernel is

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

For n feature vectors

$$x_1, x_2, \ldots, x_n,$$

the kernel matrix K has entries satisfying

$$0 < K_{ij} < 1$$

Direct computation of K is

$$O(n^2)$$

which becomes infeasible for

$$n = 10^6$$

However, K has log-rank. It can be approximated by a matrix of rank

$$O((\log n)^q)$$

for small q. This enables the "kernel trick" in support vector machines and Gaussian processes. Applications include:

- Face recognition
- Text classification
- Molecular dynamics simulations

Chapter 4

Compressed Sensing

4.1 Split Algorithms for $\ell^2 + \ell^1$

This section addresses the computational side: how do we actually solve the optimization problems that arise in compressed sensing and low-rank approximation? The key idea is splitting: separate the ℓ^1 part of the objective from the ℓ^2 part, and alternate between solving each sub-problem. This "splitting" approach underlies several major algorithms: basis pursuit, LASSO, ADMM, split Bregman, and sparse Kaczmarz.

4.1.1 Basis Pursuit and the ℓ^1 Norm

Consider a linear system $Ax = b$ where A is $m \times n$ with $m \ll n$ (many more unknowns than equations). There are infinitely many solutions. The minimum ℓ^2 norm solution is

$$x^+ = A^T(AA^T)^{-1}b$$

(the pseudoinverse solution). But this solution is generally dense — all n components are nonzero. In many applications (signal processing, medical imaging, compressed sensing), we know the true solution is sparse (most components are zero). How do we find it? The sparsest solution minimizes

$$\|x\|_0$$

(number of nonzero components). But minimizing $\|x\|_0$ is NP-hard (we would have to check all subsets of columns of A). The key insight of compressed sensing is that we can often find the sparsest solution by minimizing the ℓ^1 norm instead.

Definition- Basis Pursuit Given a matrix A ($m \times n$, $m \ll n$) and vector b, basis pursuit solves:

$$\min \|x\|_1 = |x_1| + |x_2| + \cdots + |x_n|$$

subject to

$$Ax = b$$

This is a convex optimization problem (ℓ^1 norm is convex and piecewise linear). Under appropriate conditions (the Restricted Isometry Property), basis pursuit finds the sparsest solution. The geometry of why ℓ^1 promotes sparsity: the unit ℓ^1 ball

$$\{x : \|x\|_1 \leq 1\}$$

is a diamond (or cross-polytope) with sharp corners along the coordinate axes. When the constraint set

$$\{Ax = b\}$$

meets the ℓ^1 ball, it almost always meets at a corner, where one or more coordinates are exactly zero — giving a sparse solution. The unit ℓ^2 ball is a sphere, which has no corners; it touches the constraint set at a generic point with all coordinates nonzero.
Example- Basis Pursuit in 2D

Let

$$A = [1 \;\; 1]$$

so

$$Ax = b$$

becomes

$$x_1 + x_2 = 1$$

There are infinitely many solutions. The ℓ^2 minimum norm solution is

$$x^+ = \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

(both components nonzero).
The ℓ^1 minimum solutions are

$$x^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \qquad \text{or} \qquad x^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

(one component zero = sparse). Geometrically, the line

$$x_1 + x_2 = 1$$

is tangent to the ℓ^1 ball at a vertex, but cuts through the interior of the ℓ^2 ball. This shows why ℓ^1 minimization promotes sparsity while ℓ^2 does not.

4.1.2 LASSO (Least Absolute Shrinkage and Selection Operator)

Basis pursuit requires an exact solution to

$$Ax = b$$

In the presence of noise (the realistic case in science and engineering), we cannot expect

$$Ax = b$$

exactly. The LASSO (invented by Tibshirani) adds an ℓ^1 penalty to the least squares objective:

$$\min \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1$$

or equivalently,

$$\min \frac{1}{2}\|Ax - b\|_2^2$$

subject to

$$\|x\|_1 \leq t$$

The parameter λ (or t) controls the trade-off between fitting the data (making Ax close to b) and promoting sparsity (keeping $\|x\|_1$ small). Large λ gives a very sparse solution; small λ gives a dense solution that fits the data more closely. LASSO was invented to improve on ridge regression, which minimizes

$$\frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_2^2$$

Ridge regression uses the ℓ^2 penalty, which shrinks all coefficients toward zero but never sets any exactly to zero — it does not select variables. LASSO uses the ℓ^1 penalty, which produces exact zeros — it performs automatic variable selection. **Example- LASSO vs**

Ridge Regression
Suppose

$$x^* = [3, 0, 0, \ldots, 0]^T$$

is the true sparse signal (only one nonzero). With noisy measurements,

$$b = Ax^* + \text{noise}$$

Ridge regression gives

$$x_{\text{ridge}} = [2.1, 0.3, 0.2, \ldots, 0.1]^T$$

(all nonzero, all shrunk). LASSO gives

$$x_{\text{LASSO}} = [2.8, 0, 0, \ldots, 0]^T$$

(sparse; zeros exactly set). LASSO correctly identifies that only the first component matters. The key geometric reason: the ℓ^1 ball has flat faces and sharp edges; solutions often land on edges (sparse).

The Primal Problem and Lagrangian

The general framework for $\ell^2 + \ell^1$ optimization begins with the primal problem:

$$\min f(x)$$

subject to

$$Ax = b \qquad (A \text{ is } m \times n)$$

We introduce Lagrange multipliers

$$y = (y_1, \ldots, y_m)^T$$

(one for each constraint) and form the Lagrangian:

$$L(x, y) = f(x) + y^T(Ax - b)$$

or equivalently,

$$L(x, y) = f(x) + y^T Ax - y^T b$$

The combined solution

$$(x^*, y^*)$$

is a saddle point of L:

$$\min_x \max_y L = \max_y \min_x L$$

The optimality conditions are:

$$\frac{\partial L}{\partial x} = 0$$

(minimizing over x) and

$$\frac{\partial L}{\partial y} = 0$$

(which gives $Ax = b$ back). The function

$$m(y) = \min_x L(x, y)$$

is called the dual function. The dual problem is to maximize

$$m(y)$$

When m is maximized,

$$\nabla m = Ax^* - b = 0$$

The gradient tells us the steepest direction to increase $m(y)$:

$$y_{k+1} = y_k + s_k \nabla m = y_k + s_k(Ax_{k+1} - b)$$

Dual Decomposition and Augmented Lagrangian

When

$$f(x) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n)$$

is separable, the Lagrangian

$$L(x, y)$$

splits into N independent sub-problems that can be solved in parallel:

$$f(x) + y^T(Ax - b) = \sum_i \left[f_i(x_i) + y^T A_i x_i - \frac{1}{N} y^T b \right]$$

Each processor i solves for x_i independently, then the dual variable y is updated globally. This is dual decomposition, the first step toward ADMM. To improve convergence when f is not strictly convex, we augment $f(x)$ by a penalty term

$$\frac{\rho}{2} \|Ax - b\|_2^2$$

giving the Augmented Lagrangian:

$$L_\rho(x, y) = f(x) + y^T(Ax - b) + \frac{\rho}{2} \|Ax - b\|_2^2$$

The penalty parameter

$$\rho > 0$$

acts like a step size. This augmentation improves convergence but breaks separability: if $f(x)$ is separable,

$$L_\rho$$

is not (because the coupling term

$$\|Ax - b\|_2^2$$

mixes all components of x).

4.2 ADMM: Alternating Direction Method of Multipliers

ADMM resolves the conflict between convergence and separability by splitting

$$f(x)$$

into two parts:

$$f(x)$$

(the ℓ^1 part) and

$$g(z)$$

(the ℓ^2 part). A new variable z is introduced and the constraint

$$x = z$$

is added:

$$\min\ f(x) + g(z)$$

subject to

$$Ax + Bz = c \qquad \text{and} \qquad x = z$$

The ADMM augmented Lagrangian for this split problem is:

$$L_\rho(x, z, y) = f(x) + g(z) + y^T(Ax + Bz - c) + \frac{\rho}{2}\|Ax + Bz - c\|_2^2$$

Algorithm- ADMM Iterations Starting from

$$x_0,\ z_0,\ y_0$$

iterate until convergence:

$$x_{k+1} = \arg\min_x L_\rho(x, z_k, y_k)$$

(ℓ^2 step: least squares or proxy)

$$z_{k+1} = \arg\min_z L_\rho(x_{k+1}, z, y_k)$$

(ℓ^1 step: soft thresholding)

$$y_{k+1} = y_k + \rho(Ax_{k+1} + Bz_{k+1} - c)$$

(dual variable update) Key advantage: x and z are updated alternately (not jointly), preserving separability. ADMM is surprisingly fast in practice. It may be slow to converge to very high accuracy, but it quickly reaches moderate accuracy — often sufficient for machine learning applications where overfitting makes high accuracy counterproductive. The consensus-form ADMM (with scaled dual variable

$$u = \frac{y}{\rho}$$

) is more numerically stable:

$$x_{k+1} = \arg\min_x \left(f(x) + \frac{\rho}{2} \|Ax + Bz_k - c + u_k\|_2^2 \right)$$

$$z_{k+1} = \arg\min_z \left(g(z) + \frac{\rho}{2} \|Ax_{k+1} + Bz - c + u_k\|_2^2 \right)$$

$$u_{k+1} = u_k + Ax_{k+1} + Bz_{k+1} - c$$

Soft Thresholding — Solving the ℓ^1 Step

The ℓ^1 sub-problem in ADMM (step z) has an exact closed-form solution called soft thresholding. For

$$f(x) = \lambda \|x\|_1$$

and the ADMM coupling term, the solution is:
Definition- Soft Thresholding Operator The soft thresholding operator

$$S_{\lambda/\rho}$$

applied to each component v_i gives:

$$x_i^* = \text{sign}(v_i) \max(|v_i| - \lambda/\rho, 0) = v_i \left(1 - \frac{\lambda}{\rho |v_i|}\right)_+$$

Interpretation: If

$$|v_i| \leq \lambda/\rho$$

set

$$x_i^* = 0$$

Otherwise, shrink

$$|v_i|$$

by

$$\lambda/\rho$$

toward zero. Soft thresholding is the proximal operator of the ℓ^1 norm:

$$\text{Prox}_f(v) = S_{\lambda/\rho}(v)$$

Soft thresholding is a shrinkage operator: every component of v moves toward zero. Components smaller than

$$\lambda/\rho$$

are zeroed out entirely; larger components are shrunk by

$$\lambda/\rho$$

This is why LASSO produces sparse solutions: soft thresholding at each iteration gradually drives small coefficients to exactly zero.

Example- Soft Thresholding in Action Let

$$v = [3.5, -1.2, 0.4, -2.8]$$

and threshold

$$\lambda/\rho = 1.5$$

For

$$v_1 = 3.5 > 1.5$$

$$x_1^* = 3.5 - 1.5 = 2.0$$

For

$$v_2 = -1.2$$

$$|v_2| < 1.5$$

$$x_2^* = 0$$

(zeroed out) For

$$v_3 = 0.4 < 1.5$$

$$x_3^* = 0$$

(zeroed out) For

$$v_4 = -2.8$$

$$|v_4| > 1.5$$

$$x_4^* = -2.8 + 1.5 = -1.3$$

Result:

$$x^* = [2.0, 0, 0, -1.3]$$

exactly sparse, with two zeros. This shows how ADMM + soft thresholding solves LASSO efficiently.

4.3 ADMM:(Alternating Direction Method of Multipliers)

Example 1: Constrained Minimization (Projection onto Convex Set) Minimize

$$f(x)$$

for

$$x \in K$$

in a closed convex set K. ADMM rewrites "x in K" as an indicator function

$$g(z)$$

defined by

$$g(z) = 0 \quad \text{if } z \in K$$

$$g(z) = +\infty \quad \text{if } z \notin K$$

ADMM iteration:

$$x_{k+1} = \arg\min \left[f(x) + \frac{\rho}{2} \|x - z_k + u_k\|^2 \right]$$

$$z_{k+1} = \text{projection of } (x_{k+1} + u_k) \text{ onto } K$$

The z-step is just a projection onto K — computable in closed form for many convex sets (balls, boxes, cones, affine subspaces, etc.).

Example 2: Soft Thresholding (ℓ^1 Minimization) Minimize

$$\lambda \|x\|_1$$

subject to

$$Ax = b$$

(basis pursuit). ADMM splits

$$f(x) = \lambda \|x\|_1$$

and

$$g(z) = 0$$

with

$$x = z$$

The update becomes

$$x_{k+1} = \text{soft-threshold} \left[(I - A^T A)(z_k - u_k) + A^T b,\ \lambda/\rho \right]$$

(approximate form).

In practice, the x-step is done by conjugate gradient or a direct solve when

$$A^T A$$

is sparse.

Example 3: Nonnegative Matrix Factorization (NMF) Approximate

$$A \approx CR$$

with all entries

$$C_{ij} \geq 0$$

and

$$R_{ij} \geq 0$$

This is NMF- fundamental in image processing, topic modeling, and bioinformatics. Optimization problem:

$$\min \|A - X\|_F^2 + I_+(C) + I_+(R)$$

subject to

$$X = CR$$

ADMM alternates between updating X (and C, R) and projecting onto nonnegative matrices. The rows of

$$X_{k+1}$$

and

$$C_{k+1}$$

can be found separately, enabling parallel computation.
Example 4: LASSO via Scaled ADMM LASSO problem:

$$\min \frac{1}{2}\|Ax - b\|_2^2 + \lambda\|x\|_1$$

Scaled ADMM iteration:

$$x_{k+1} = (A^T A + \rho I)^{-1}(A^T b + \rho(z_k - u_k))$$

$$z_{k+1} = S_{\lambda/\rho}(x_{k+1} + u_k)$$

(soft thresholding)

$$u_{k+1} = u_k + x_{k+1} - z_{k+1}$$

By storing the LU factors of

$$A^T A + \rho I$$

the first step reduces to a back substitution - very fast for sparse A. Replacing

$$\|x\|_1$$

by

$$\|Fx\|_1$$

(total variation) gives denoising algorithms for image processing.

4.4 Proximal Algorithms and the Prox Operator

Definition- Proximal Operator The proximal operator of a convex function F is defined as follows:

$$\text{prox}_F(v) = \arg\min_x \left(F(x) + \frac{1}{2}\|x - v\|_2^2 \right)$$

Interpretation:

$$\text{Prox}_F(v)$$

finds the point x that minimizes F but does not move too far from v.

$$\text{Prox}_F$$

is a generalization of gradient descent: [x-$\alpha\nabla F(x)$ corresponds to

$$\text{Prox}_{\alpha F}(x)$$

Proximal algorithms are to nonsmooth optimization what gradient descent is to smooth optimization. They handle functions that are not differentiable (like)

$$\|x\|_1$$

by using the proximal operator instead of the gradient. Key facts:

- ℓ^1 norm:
$$\text{Prox}_F(v) = \text{soft thresholding } S_\lambda(v)$$
- Indicator function of K:
$$\text{Prox}_F(v) = \text{projection onto } K$$
- ℓ^2 norm:
$$\text{Prox}_F(v) = \frac{v}{1+\lambda}$$
(shrinkage toward zero)

The two-step ADMM algorithm can be written as two proximal operators alternating between the ℓ^2 part (F_2) and the ℓ^1 part (F_1):

$$x_{k+1} = (\partial F_2 + \alpha I)^{-1}(\alpha z_k - \alpha u_k)$$

(ℓ^2 proximal step)

$$z_{k+1} = (\partial F_1 + \alpha I)^{-1}(\alpha x_{k+1} + \alpha u_k)$$

(ℓ^1 proximal step)

$$u_{k+1} = u_k + x_{k+1} - z_{k+1}$$

(dual update)

Bregman Distance

Definition- Bregman Distance Given a convex function f with gradient

$$\nabla f$$

(or subgradient)

$$\partial f,$$

the Bregman distance from u to v is:

$$D_f(u, v) = f(u) - f(v) - \langle \nabla f(v), u - v \rangle$$

The Bregman distance is not a true distance:

$$D_f(u, v) \neq D_f(v, u)$$

in general. But

$$D_f(u, v) \geq 0$$

and

$$D_f(u, v) = 0$$

if

$$u = v$$

(for strictly convex f). The Bregman distance generalizes the squared Euclidean distance. If

$$f(x) = \frac{1}{2}\|x\|_2^2$$

then

$$D_f(u, v) = \frac{1}{2}\|u - v\|_2^2$$

For

$$f(x) = \|x\|_1$$

the Bregman distance uses subgradients — any tangent line lying below the graph of f. The subgradient of

$$|x|$$

at

$$x = 0$$

is any value in

$$[-1, 1]$$

4.5 Split Bregman and Sparse Kaczmarz

The linearized Bregman iteration is a simple two-step algorithm for minimizing

$$\lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$$

subject to

$$Ax = b$$

The iteration step is:

$$y_{k+1} = y_k - s_k A^T(Ax_k - b)$$

(gradient descent step on)

$$\|Ax - b\|^2$$

Split Bregman and Sparse Kaczmarz The iteration step is:

$$x_{k+1} = S(y_{k+1}) = \text{sign}(y_{k+1}) \max(|y_{k+1}| - \lambda, 0)$$

(soft thresholding) This iteration converges to

$$x^*$$

that minimizes

$$\lambda\|x\|_1 + \frac{1}{2}\|x\|_2^2$$

subject to

$$Ax = b$$

provided

$$s\lambda_{\max}(A^T A) < 1$$

The sparse Kaczmarz algorithm is the per-equation version of linearized Bregman, effective for big sparse systems where equations arrive one at a time. The sparse Kaczmarz iteration is:

$$y_{k+1} = x_k - s_k a_i$$

(Kaczmarz step: projection onto the i-th equation)

$$x_{k+1} = S(y_{k+1})$$

(soft thresholding) Lin and Zhou have shown that sparse Kaczmarz is an online learning algorithm: the step

$$k \to k + 1$$

is taken as soon as the new observation

$$b_i$$

and equation

$$a_i^T x = b_i$$

arrive, enabling real-time processing of streaming data.

4.6 Bounded Variation: ℓ^1 Norm for Gradients

Natural images have edges: across these edges, the defining function

$$u(x, y)$$

can have a jump. The gradient

$$\nabla u = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y} \right)$$

can be smooth along the edge, but it has a delta function in the perpendicular direction. The standard energy

$$\|\nabla u\|_2^2$$

is infinite for such functions. The bounded variation (BV) norm is the correct substitute:

$$\|\nabla u\|_2^2 = \iint (u_x^2 + u_y^2)\, dx\, dy = \infty$$

(infinite for step functions)

$$\|u\|_{BV} = \|\nabla u\|_1 = \iint \sqrt{u_x^2 + u_y^2}\, dx\, dy < \infty$$

(finite for step functions) The BV norm is the ℓ^1 norm of the gradient. Minimizing the BV norm is called total variation regularization, introduced by Rudin, Osher, and Fatemi. In image denoising, we fit the data while minimizing the BV norm. This allows edges (which the ℓ^2 norm would smooth away) while removing random "speckled" noise. Total variation denoising using ADMM is one of the most successful image processing algorithms ever developed.

Chapter 5

Literature Review: Comparative Scholarly Analysis

5.1 Overview

This chapter provides a systematic literature review of the theoretical and computational themes developed in Chapters 3 and 4, with primary focus on four interconnected research areas: (i) low-rank matrix perturbation and the Sherman–Morrison–Woodbury (SMW) formula, (ii) eigenvalue interlacing and matrix perturbation theory, (iii) structured matrices with exponentially decaying singular values (Vandermonde and NUDFT families), and (iv) compressed sensing, LASSO, and ADMM-based optimization. The review surveys five landmark or recent scholarly contributions published between 2021 and 2024, comparing their findings with the treatment given in Chapters 3 and 4, and situating each within the broader mathematical and engineering literature. The field of low-rank matrix analysis has expanded dramatically over the past decade, driven by large-scale data problems in machine learning, signal processing, and scientific computing. The chapters above synthesize foundational ideas — the SMW formula, interlacing theorems, the Sylvester equation, nuclear norm minimization, and splitting algorithms — that serve as unifying threads across this diverse scholarly landscape. The following sections examine how recent scholarship extends, refines, or challenges those ideas.

5.2 Paper 1 — Güttel, Nakatsukasa, Webb & Bloor Riley (2024)

"A Sherman–Morrison–Woodbury Approach to Solving Least Squares Problems with Low-Rank Updates" Reference: S. Güttel, Y. Nakatsukasa, M. Webb, and A. B. Riley. *arXiv:2406.15120*, June 2024.

Summary

This 2024 paper extends the classical SMW formula — which efficiently updates the inverse of A when it undergoes a low-rank modification — to the non-square rectangular setting of least squares problems. The authors derive a simple closed-form update for the pseudoinverse of a full-rank rectangular matrix A when subjected to a low-rank perturbation UV^T. The resulting algorithm, *WoodburyLS*, is shown to be dramatically faster than solving the modified least squares problem from scratch whenever the rank of the update r is much smaller than the number of rows or columns. The paper demonstrates that the speedup mirrors the classical square-matrix case: instead of recomputing a full QR

factorization of the modified system, only a small $r \times r$ linear system needs to be solved in addition to the original pseudoinverse computation. The authors provide rigorous bounds on the numerical stability of the update and validate performance on deflation problems arising from multiple-local-minima searches in nonlinear least squares.

Comparison with Chapter 3

Chapter 3 (Section 1.2) develops the SMW formula for square invertible matrices and then applies it in Section 1.3 to recursive least squares. The Güttel et al. (2024) paper directly generalizes this framework. Where Chapter 3 derives the update for the inverse $(A^T A)^{-1}$ via the Woodbury identity, this paper works directly with the pseudoinverse A^+ of the rectangular matrix, obtaining a formula valid even when A is non-square or rank-deficient. The "push-through identity" discussed in Chapter 3 (Section 1.1.2) — which allows movement between large and small matrix inverses — has a direct analogue in the rectangular pseudoinverse update. A key divergence is that Chapter 3 focuses primarily on the computational advantage for rank-one updates in the context of streaming data (recursive least squares), while Güttel et al. address the more general rank-r case and provide speedup guarantees measured against QR factorization costs. The paper also treats numerical stability rigorously, a topic that Chapter 3 discusses qualitatively in the context of ill-conditioned Hilbert matrices but does not formalize.

Aspect	**Chapter 3 Treatment**	**Güttel et al. (2024)**
Setting	Square invertible matrices; RLS	Rectangular full-rank; least squares
Update type	Rank-1 and rank-k	Rank-r (general)
Key formula	SMW for $(A^T A)^{-1}$	Pseudoinverse update
Stability analysis	Qualitative (Hilbert matrix)	Rigorous numerical bounds
Application	Kalman filter, RLS	Nonlinear least squares deflation

5.3 Paper 2 — Clouâtre, Klippenstein & Slevinsky (2023/2025)

"Lifting Sylvester Equations: Singular Value Decay for Non-Normal Coefficients" Reference: R. Clouâtre, B. Klippenstein, and R. M. Slevinsky. *Integral Equations and Operator Theory* (Springer), published online January 2025. *arXiv:2308.11533.*

Summary

This paper addresses a fundamental theoretical question underlying Chapter 3, Section 1.9: under what conditions does a low-displacement-rank Sylvester equation

$$AX - XB = C$$

force the solution X to have exponentially decaying singular values? The foundational result, due to Beckermann and Townsend, asserts this when A and B are normal with well-separated spectra. Clouâtre, Klippenstein, and Slevinsky relax the normality condition through the technique of operator dilations, replacing normal matrices A, B with dilations that are normal on a larger space. Their central result – proved using commutant lifting

theory — establishes that the singular value decay rate for X depends on the Zolotarev numbers connecting the spectra of A and B, even when A is non-normal. As a practical corollary, they show that the factored alternating direction implicit (ADI) method for solving

$$AX - XB = C$$

converges quickly without requiring A to be normal, removing a long-standing theoretical restriction on the method.

Comparison with Chapter 3

Chapter 3, Section 1.9.5 introduces the Sylvester equation

$$AX - XB = C$$

as the unifying framework behind rapidly decaying singular values, and explicitly states the Beckermann–Townsend theorem: if A, B are normal with no shared eigenvalues, then singular values of X decay exponentially. The ADI method of Section 1.10 then relies directly on this structure. The Clouâtre et al. paper therefore extends a core theorem of Chapter 3 to a broader class of coefficient matrices. The practical implication is significant: the Vandermonde matrix example in Section 1.9.4, where

$$A = \mathrm{diag}(x_1, \ldots, x_n)$$

is normal (diagonal) and B is the companion matrix (which is generally non-normal), now falls under the relaxed framework. Chapter 3 glosses over the non-normality of the companion matrix — stating merely that A and B "have no shared eigenvalues." Clouâtre et al. formalize exactly why this suffices (via the dilation argument) and provide quantitative decay bounds that Chapter 3 leaves implicit. Furthermore, the Krylov matrix example in Section 1.10 — where B equals the companion matrix — also benefits from this theoretical advance, confirming the exponential decay illustrated in the chapter's singular value plots even for non-normal B.

5.4 Paper 3 — Zhang et al. (2024)

"Image Compressive Sensing Reconstruction via Nonlocal Low-Rank Residual-Based ADMM Framework" Reference: Z. Zhang et al. *Computer Vision and Image Understanding*, Vol. 248, November 2024, Article 104114.

Summary

This paper proposes a novel Nonlocal Low-Rank Residual (NLRR) framework for image compressive sensing reconstruction. The key insight is that existing methods apply low-rank approximation directly to degraded nonlocal image groups (patches of similar texture), which can yield suboptimal results when the degradation is severe. Instead, Zhang et al. progressively approximate the underlying low-rank matrix by minimizing the low-rank residual — the difference between the current approximation and the true signal — using an ADMM-based iterative scheme. The ADMM implementation in this paper exploits two complementary priors simultaneously:

1. the nonlocal self-similarity (NSS) prior, which clusters similar patches into groups for low-rank modeling, and

2. a weighted nuclear norm penalty that places different weights on different singular values, allowing the method to avoid excessive shrinkage of large (signal-carrying) singular values.

The paper demonstrates state-of-the-art performance on benchmark datasets at low sampling ratios.

Comparison with Chapters 3 and 4

The chapters establish the theoretical foundations on which this applied paper builds. Chapter 3, Section 1 motivates the low-rank assumption through the matrix completion framework, where minimizing the nuclear norm — the sum of all singular values — is the convex relaxation of minimizing rank. Chapter 4 then develops the ADMM computational machinery, including the augmented Lagrangian, splitting, and soft thresholding. Zhang et al. (2024) make two direct extensions. First, they replace the standard nuclear norm penalty (which equally shrinks all singular values) with a weighted version, implementing a more sophisticated proximal operator than the soft-thresholding operator of Chapter 4, Section 2.2. This is theoretically motivated by the observation — implicit in Chapter 3's treatment of rapidly decaying singular values — that large singular values carry more information and should be preserved. Second, their residual-based formulation represents a form of iterative Bregman refinement, closely related to the linearized Bregman iteration discussed in Chapter 4, Section 2.5. A notable contrast is that Chapter 4's ADMM convergence analysis assumes standard convexity conditions, while Zhang et al. operate in a setting where the weighted nuclear norm penalty introduces non-convexity. Their paper provides convergence guarantees for the specific NLRR-ADMM scheme but does not offer the general framework Chapter 4 presents.

5.5 Paper 4 — Abbe, Fan, Yang & Zhong (2023, JMLR)

"Entrywise Estimation of Singular Vectors of Low-Rank Matrices with Heteroskedastic Noise and the Matrix Completion Problem" Reference: E. Abbe, J. Fan, K. Yang, and Y. Zhong. *Journal of Machine Learning Research*, 24(167), 1–38, 2023.

Summary

This JMLR paper establishes minimax-optimal statistical guarantees for nuclear-norm-based matrix completion when the observed matrix has nonlinear low-dimensional structure, going beyond the classical linear low-rank assumption. The authors prove that the standard nuclear-norm-minimization estimator is consistent for matrices satisfying a smooth manifold embedding hypothesis, and that the convergence rate depends on the smoothness degree L and the manifold dimension K through the exponent

$$\frac{2L}{2L+K}.$$

A matching minimax lower bound is derived, confirming that nuclear norm penalization is rate-optimal up to logarithmic factors. The paper also addresses entrywise estimation of singular vectors under heteroskedastic (non-uniform) noise, providing ℓ_∞-norm bounds on the error in individual singular vector entries. These results underpin applications such as recommendation systems — the Netflix-style completion problem described in Chapter 3 — where noise may vary significantly across users and items.

Comparison with Chapter 3

Chapter 3 introduces matrix completion through the nuclear norm minimization framework:

$$\min \|A\|_N$$

over missing entries as a convex proxy for rank minimization. The analogy drawn in Chapter 3 — that the nuclear norm is to matrices what the ℓ_1 norm is to vectors — is precisely the theoretical motivation Abbe et al. formalize in the JMLR paper through statistical optimality theory. However, Chapter 3 does not address statistical noise or sample complexity. The Abbe et al. paper fills this gap, answering the question: how many observed entries are needed for nuclear-norm completion to provably succeed? Their answer depends on the underlying structure (linear vs. nonlinear low-rank), and the proof techniques rely on the Courant–Fischer max-min theorem and Weyl-type eigenvalue perturbation bounds as key tools for controlling singular vector perturbations.

Theme	**Chapter 3**	**Abbe et al. (2023)**
Completion model	Nuclear norm minimization	Statistical analysis with noise
Sample complexity	Not addressed	Minimax rate $\frac{2L}{2L+K}$
Noise model	None (exact observations)	Heteroskedastic noise
Key tools used	SMW, nuclear norm	Courant–Fischer, Weyl inequalities
Structure assumption	Linear low-rank	Nonlinear smooth manifold

5.6 Paper 5 — Chen, Huang, Jia & Wang (2023)

"Enhanced ADMM for Solving the LASSO Problem: Integrating the Nesterov Accelerated Gradient Method" Reference: J. Chen, J. Huang, X. Jia, and R. Wang. *Control Theory and Technology*, Springer, 2023.

Summary

This paper proposes an accelerated variant of the standard ADMM for solving the LASSO problem, integrating Nesterov's momentum technique into the dual variable update step. The standard ADMM iteration updates the dual variable y via a simple gradient ascent step on the dual function. By incorporating the Nesterov acceleration — adding an inertial momentum term that extrapolates from previous iterates — the modified algorithm achieves

$$O\left(\frac{1}{k^2}\right)$$

convergence of the primal objective residual compared to the

$$O\left(\frac{1}{k}\right)$$

rate of standard ADMM.

The paper targets the standard LASSO formulation

$$\min \frac{1}{2}\|Ax - b\|^2 + \lambda\|x\|_1$$

and its solution via the scaled consensus-form ADMM.

Comparison with Chapter 4

Chapter 4 develops the ADMM algorithm from first principles through the augmented Lagrangian, splitting, and soft thresholding, noting that ADMM is fast in practice but may converge slowly to very high accuracy. The Chen et al. paper directly addresses this limitation, providing a concrete mechanism — Nesterov momentum — to accelerate convergence without sacrificing the computational structure that makes ADMM attractive. The x-update and z-update (soft thresholding) steps of Chapter 4 remain unchanged in the N-ADMM framework; only the dual variable update is modified. Thus, all practical LASSO examples in Chapter 4 carry over directly to N-ADMM with essentially the same computational cost per iteration.

5.7 Comparative Synthesis and Critical Analysis

Taken together, the five reviewed papers illuminate both the strengths and the gaps in the treatment given across Chapters 3 and 4.

Theoretical Completeness

Chapters 3 and 4 provide a self-contained and mathematically rigorous introduction to their topics, but necessarily stop short of the frontier. The Beckermann–Townsend normality requirement for exponential singular value decay is relaxed by Clouâtre et al. (2023), while the nuclear norm minimization framework is shown by Abbe et al. (2023) to possess stronger statistical optimality properties than originally suggested.

Computational Extensions

The SMW formula and recursive least squares treatment of Chapter 3 are extended in two directions by the literature. Güttel et al. (2024) broaden the formula to the rectangular pseudoinverse setting, while related Kalman filter literature demonstrates superior numerical stability through iterative implementations.

Applied Methodology

Chapter 4's ADMM treatment remains highly relevant to modern compressed sensing and low-rank optimization. The reviewed literature extends ADMM through:

1. weighted and non-convex regularizers beyond the uniform nuclear norm and ℓ_1 norm, and

2. acceleration using Nesterov momentum.

Cross-Cutting Themes

A recurring theme across all five papers is the central role of the singular value decomposition (SVD) and eigenvalue structure in connecting theory to computation. Weyl inequalities, interlacing theorems, Zolotarev numbers, and the Courant–Fischer theorem collectively unify the theoretical developments reviewed throughout this chapter.

Paper	**Year**	**Core Topic**	**Extends**
Güttel et al.	2024	SMW for pseudoinverse	Ch. 3
Clouâtre et al.	2023/25	Sylvester decay theory	Ch. 3
Zhang et al.	2024	NLRR-ADMM image CS	Ch. 3 and 4
Abbe et al.	2023	Matrix completion statistics	Ch. 3
Chen et al.	2023	Accelerated ADMM-LASSO	Ch. 4

5.8 Conclusion

Chapters 3 and 4 establish a coherent and pedagogically complete treatment of low-rank matrix perturbation, singular value structure, and compressed sensing algorithms. The scholarly literature reviewed here confirms that the foundational ideas of those chapters — the SMW formula, nuclear norm minimization, exponential singular value decay via Sylvester equations, and ADMM-based splitting — remain central to active research. The comparative analysis reveals three principal directions in which recent literature extends the material:

1. theoretical relaxation of normality and linearity assumptions,

2. computational extension to rectangular and large-scale settings, and

3. applied methodology for image reconstruction under non-convex penalties.

Together, these developments demonstrate that the interplay between convexity, singular value structure, sparsity, and efficient computation forms a unified mathematical framework underlying modern data science, optimization, and numerical linear algebra.

Bibliography

The following references are cited in this literature review chapter. They are listed in the order in which they first appear, corresponding to the five core scholarly papers reviewed (§5.2–§5.6) followed by foundational references.

1. S. Güttel, Y. Nakatsukasa, M. Webb, and A. Bloor Riley. "A Sherman–Morrison–Woodbury Approach to Solving Least Squares Problems with Low-Rank Updates." arXiv preprint arXiv:2406.15120, June 2024. Presented at the University of Manchester Mathematical Colloquium, 2024.

2. R. Cloüte, B. Klippenstein, and R. M. Slevinsky. "Lifting Sylvester Equations: Singular Value Decay for Non-Normal Coefficients." *Integral Equations and Operator Theory*, Springer Nature Link, published online January 13, 2025. arXiv:2308.11533 (2023).

3. Z. Zhang, S. Zhang, L. Zha, and X. Zhang. "Image Compressive Sensing Reconstruction via Nonlocal Low-Rank Residual-Based ADMM Framework." *Computer Vision and Image Understanding*, Vol. 248, Article 104114, October 28, 2024.
Foundational References

   (a) B. Beckermann and A. Townsend. "On the Singular Values of Matrices with Displacement Structure." *SIAM Journal on Matrix Analysis and Applications*, 38(4), pp. 1227–1248, 2017.

   (b) S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers." *Foundations and Trends in Machine Learning*, 3(1), pp. 1–122, 2011.

   (c) E. J. Candès and B. Recht. "Exact Matrix Completion via Convex Optimization." *Foundations of Computational Mathematics*, 9(6), pp. 717–772, 2009.

— End of Chapter 5 —