Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP Signal Processing

Prof. Dr.-Ing. Timo Gerkmann

Fundamentals of Data Analytics

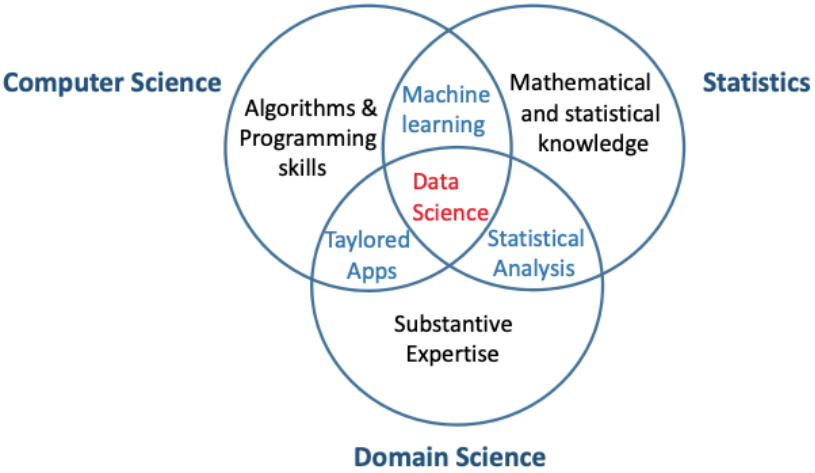
November 7, 2024

Outline

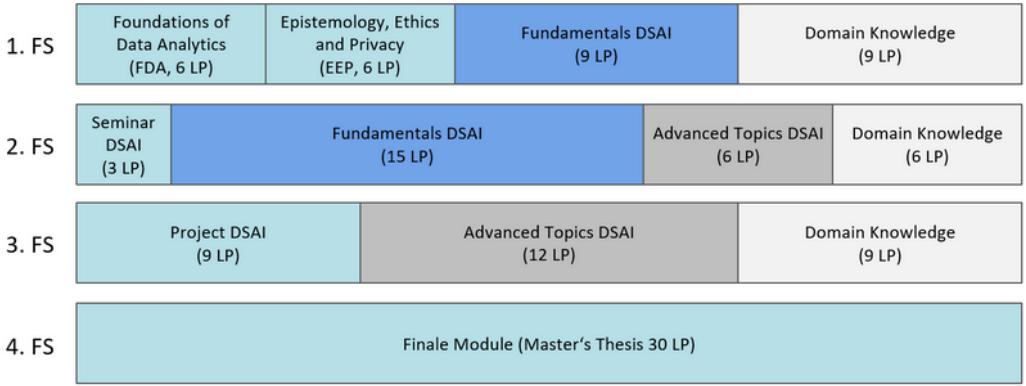
1. Introduction
2. Prerequisites from Matrix Analysis
3. Multivariate Distributions and Moments
4. Dimensionality Reduction
5. Classification and Clustering
6. Support Vector Machines
7. Machine Learning

- Data Analytics is an interdisciplinary field combining:
  - Exploratory Statistics
  - Algorithms and Information Theory
- Aim: Reveal hidden structures in large datasets.
- Key Domains:
  - Computer Science
  - Statistics
  - Domain-specific Expertise

Data Science is Interdisciplinary



Data Science and AI Study Plan



Mandatory Courses in DSAI

- Total: 54 CP
- Moduls:
  - InfM-FDA: Foundations of Data Analytics (6 CP)
  - InfM-EP: Ethics and Privacy (6 CP)
  - InfM-Proj: Project Data Science and Artificial Intelligence (9 CP)
  - InfM-Sem: Seminar Data Science (3 CP)
  - InfM-MA/DSAI: Thesis Data Science (30 CP)

- Total: 24 CP
- Moduls:
  - InfM-DE: Introduction to Data Engineering (6 CP)
  - InfM-DIS: Databases and Information Systems (9 CP)
  - InfM-ALG: Algorithmik (9 CP)
  - InfM-ML: Machine Learning (9 CP)
  - InfM-STSP: Statistical Signal Processing (9 CP)
  - InfM-SWA: Software Architecture (6 CP)
  - InfM-NN: Neural Networks (6 CP)

- Total: 18 CP
- Moduls:
  - InfM-BAI: Bioinspirierte Künstliche Intelligenz (Bio-Inspired Artificial Intelligence) (6 CP)
  - InfM-BKIM: Biostatistik und Künstliche Intelligenz in der Medizin (6 CP)
  - InfM-CV 1: Computer Vision I (6 CP)
  - InfM-CV 2: Computer Vision II (6 CP)
  - InfM-IR: Intelligente Roboter (Intelligent Robotics) (6 CP)
  - InfM-LT: Sprachtechnologie (Language Technology) (6 CP)
  - InfM-NCP: Natürliche Sprachverarbeitung und das Web (6 CP)
  - InfM-RT: Robot Technology (6 CP)
  - InfM-SSV: Sprachsignalverarbeitung (Speech Signal Processing) (6 CP)
  - InfM-WV: Wissensverarbeitung (Knowledge Processing) (6 CP)

Domain Knowledge in DSAI

- Total: 24 CP from at least 2 Domains and at least 6 CP per domain
- Domains:
  - Mathematics
  - Informatics
  - Physics
  - Chemistry
  - Biology
  - Earth System Sciences

Essentials for Successful Data Science

- Successful data science requires:
  - Machine Learning proficiency
  - Strong Statistical Analysis skills
  - Tailored applications in specific fields
- Significance of:
  - Computational advances allowing high-dimensional data analysis
  - Parallel and distributed algorithms

Applications and Impacts of Data Analytics

- Applications are manyfold:
  - Speech and Language Processing
  - Computational imaging
  - Computer vision
  - Recommender systems
  - Domain-specific applications
- Typical Techniques:
  - Classification and pattern recognition
  - Supervised and unsupervised learning

Current Trends and Demands

- Data explosion due to:
  - Digitization and Internet of Things (IoT)
  - Social media and internet-based data and data transmission
- Challenges include:
  - Handling diverse, large-scale data
  - Drawing meaningful conclusions quickly

Purpose and Structure of the Course

- Aim: To provide a comprehensive overview of data analytics fundamentals, including mathematical foundations.
- Content Highlights:
  - Matrix analysis and optimization
  - Optimization and statistical learning

1. Introduction

What is Data Analytics?

Data Analytics is the science of exploring big data and designing methods and algorithms for detecting structures and information. Key points include:

- Exploration and analysis of large (often unstructured) data.
- Creation of models to understand data behavior and drive decisions.
- Multidisciplinary approach incorporating statistics, machine learning, and more.

Understanding models in Data Analytics:

- **Statistical Models**: Assumes an underlying data distribution; focuses on estimating parameters.
  - E.g. $\mathcal{N}(\mu, \sigma^2)$, independent samples
- **Machine Learning Models**: Uses data as training sets; includes algorithms like neural networks, Bayesian inference models, support vector machines
- **Dimensionality Reduction**: Extract most prominent features, ignore the rest; includes methods like PCA or manifold learning.
- **Summarization Models**: Aggregates data into comprehensive formats; common in clustering applications.

Example: Gaussian Distribution

- Assume data are independent samples from a Gaussian distribution $N(\mu, \sigma^2)$.
- Model captures distribution via estimators for mean (μ) and variance (σ^2).

Bonferroni's Principle

In large randomn data sets, unusual features occur purely by chance.

Example 1.1

Consider finding evil-doers by screening people visiting the same hotel twice:

- 10^5 hotels.
- each individual visits a hotel once in 100 days.
- 10^9 individuals.
- People pick days and hotels independently.
- Examined over 1000 days.

Bonferroni's Principle

In large randomn data sets, unusual features occur purely by chance.

Example 1.1

Consider finding evil-doers by screening people visiting the same hotel twice:

- 10^5 hotels.
- each individual visits a hotel once in 100 days.
- 10^9 individuals.
- People pick days and hotels independently.
- Examined over 1000 days.

➡ Probability for same hotel on two days: 10^{-18}.

➡ But: Expected number of such events: 250,000.

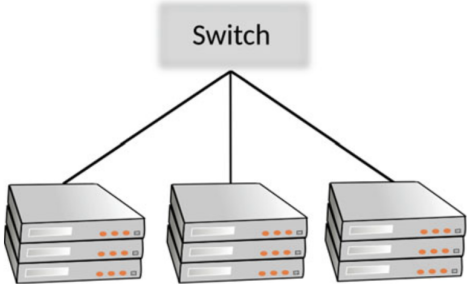
Without a solid model, large datasets can lead to misinterpretation due to randomness.

Solution:

- Probability that any two people visit a hotel on the same day is $\frac{1}{100}\frac{1}{100} = 10^{-4}$
- Probability that they pick the same hotel on the same day: $\frac{1}{10^4}\frac{1}{10^5} = 10^{-9}$
- The probability that two people visit the same hotel on two different days are $10^{-9} \cdot 10^{-9} = 10^{-18}$
- Cardinality of the event space, with $\binom{n}{2} = \frac{n!}{2!(n-2)!} \approx \frac{n^2}{2}$ for $n \gg 2$
  - pairs of people: $\binom{10^9}{2} \approx 5 \cdot 10^{17}$
  - pairs of days: $\binom{1000}{2} \approx 5 \cdot 10^5$
- Expected number of such events:
  - ➡ $5 \cdot 10^{17} \cdot 5 \cdot 10^5 \cdot 10^{-18} = 25 \cdot 10^4 = 250.000$
    pairs of people and days, probability that they pick the same hotel on two different days,
  - ➡ We need to screen 250.000 events

While we are considering an unlikely event, still the number of events is large as there are so many people.

- We briefly discuss fundamentals of implementations to deal with big data sets
- Note: For huge datasets hardware errors will occur almost surely
- Key Idea: instead of one super-computer, use many computers to process data in parallel



- Software stack
  - Distributed file system (DFS)
    - large blocks
    - redundancy by replication
  - Programming system: MapReduce
    - tolerant to hardware errors
    - able to handle large dara efficiently
    - Map: process data in parallel
    - Reduce: combine results
- Architecture
  - Compute node stored in a rack, each with its own processor and storage device
  - The racks are connected by switches (Gigabit links)

Parallel Programming and MapReduce III

- Principles
  - Files are stored redundantly to protect against failures
  - Computations are divided into independent tasks. If one fails it can be restarted without affecting others

  Distributed Files System (DFS)
  - Files are divided into large chunks (e.g. 64MB)
  - chunks are replicated to protect against hardware failures (e.g. 3 times on different racks)
  - there is a file-master node or name-node that keeps track of the location of the chunks
- MapReduce (computing paradigm)
  - System manages parallel execution, coordination of tasks
  - Two functions are written by the user: map and reduce
  - Implementations
    - MapReduce (Google, internal)
    - Hadoop (Apache, open source)

Outline

Matrix-Vector Multiplication

Multiply matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ with vector $\mathbf{v} \in \mathbb{R}^n$.

$$\mathbf{x} = \mathbf{M}\mathbf{v}, \qquad \text{i.e.} \quad x_i = \sum_{j=1}^{n} m_{ij} v_j$$

Example (Example 1.2)

Suppose matrix dimensions are large for $n = 10^7$

- Direct computation requires storing entire matrix. Inefficient for such large matrices.

MapReduce implementation:

- Store matrix $\mathbf{M}$ as $((i, j), m_{ij})$ and vector $\mathbf{v}$ as (i, v_i).
- **Map Function:** Emit key-value pairs $(i, m_{ij} v_j)$.
- **Reduce Function:** Sum values $m_{ij} v_j$ for key i to find x_i.

- While the computation of $(i, m_{ij}v_j)$ may be implemented as a vector multiplication, this would require storing the entire vector.
- If vector $\mathbf{v}$ is too large, split into blocks and process in parallel.
- The matrix $\mathbf{M}$ is divided into horizontal stripes and the vector $\mathbf{v}$ is divided into vertical stripes of same size.
- Then, the map function computes the product of the *stripe of* $\mathbf{M}$ and the stripe of $\mathbf{v}$.
- The reduce function sums the results to obtain the final product.

➡ This allows for parallel computation and is more efficient for large datasets.

Example (Example 1.3)

Given two matrices $\mathbf{M} \in \mathbb{R}^{n \times m}$ and $\mathbf{N} \in \mathbb{R}^{m \times r}$, compute their product $\mathbf{MN} \in \mathbb{R}^{n \times r}$.

Map Function

- For each m_{ij} of $\mathbf{M}$, create r key-value pairs $((i,k),(\mathbf{M},j,m_{ij}))$ for $k = 1,\ldots,r$.
- For each n_{jk} of $\mathbf{N}$, create n key-value pairs $((i,k),(\mathbf{N},j,n_{jk}))$ for $i = 1,\ldots,n$.

Reduce function computes multiplication as follows

- For each key (i,k) find the values with the same j
- Multiply m_{ij} and n_{jk}
- sum $m_{ij}n_{jk}$ over j to get $\sum_{j=1}^{m} m_{ij}n_{jk}$

- Relational algebra involves operations that search, select, or group data samples based on their relationships.
- A relation $R(A_1, \ldots, A_k)$ is defined by a sequence of attributes, known as a **schema**.
- In a directed graph, relationships like "being connected" can be modeled with attributes such as a starting vertex and target vertex.

Common Operations

Common operations in relational algebra include:

- Selection
- Projection
- Union
- Intersection
- Natural Join
- Grouping and Aggregation

This section focuses on selection and projection.

- For a relation R and a condition C, a selection operation returns tuples in R that satisfy C.
- Denoted as $\sigma_C(R)$.
- Example: Finding vertices closer than r in a geometric graph.

Map Function

- For each tuple t in R, if t satisfies C, generate a key-value pair (t, t).

Reduce Function

- For each pair (t, t), return the same key-value pair (t, t).

The selection operation is completed in the Map phase.

Projection Operation and Implementation

- Extracts a subset S of attributes from each tuple in relation R.
- Denoted as $\pi_S(R)$.
- Example: Find vertices with at least one outgoing edge in a directed graph.

Map Function

- For each tuple t in R, return only those attributes in S as t', and generate key-value pair (t', t').

Reduce Function

- For all pairs (t', t'), return a single key-value pair (t', t').

Duplicates are removed in the Reduce phase.

Projection Operation and Implementation

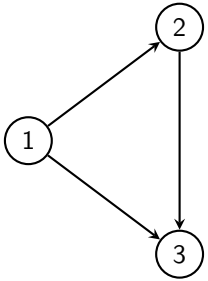
Example: Directed Graph

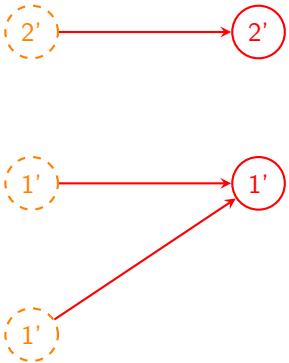
- The relation R as two attributes: the starting and target vertices of an edge.
- To find vertices with at least one outgoing edge, it suffices to select the first attribute.
- Done via projection by selectying only subset S at the attributes from each tuple in R
- The output is denoted by $\pi_S(R)$

Projection Operation and Implementation

Map Phase: Generate (1', 1'), (1', 1'), (2', 2')

Reduce Phase: Remove duplicate (1', 1')





Grouping and Aggregation

- **Grouping:** Partitions tuples in relation R based on common attributes G.
  - Example: Listing neighbors of each vertex in a directed graph.
  - ➡ **Map Function**
- **Aggregation:** Reduces groups to a single value
  - Example: counting the neighbors of each vertex.
  - ➡ **Reduce Function**

  - **Map Function:** For each tuple g with attributes in G, and for each tuple t in R, return those attributes in G^C as t', and generate key-value pair (g, t').
  - **Reduce Function:** For all (g, t') pairs, return a single key-value pair $(g, \gamma(t'))$ with the aggregation function $\gamma(\cdot)$.
- Integral in machine learning for summarizing data.

- **Objective:** Construct tuples from relations R and S that share common attributes.
- **Example:** Identify all paths of length two in a directed graph, e.g. (u, v, w), using tuples (u, v) in R and (v, w) in S.
  - Representation via natural join of $R(U, V)$ and $S(V, W)$.
- General: MapReduce for the natural join $R \bowtie S$ of $R(A, B)$ and $S(B, C)$:
  - **Map Function:**
    - R: Generate key-value pair $(b, (R, a))$ for each tuple (a, b).
    - S: Generate key-value pair $(b, (S, c))$ for each tuple (b, c).
  - **Reduce Function:** For all key-value pairs, with key b, take all a from values (R, a) and all c from values (S, c). Return all pairs (a, b, c).

Executing Set Operations

- Handles union, intersection, difference across relational data.
- **Union:** Combines all tuples from relations with shared schema.
- **Intersection/Difference:** Requires subset filtering, computed similarly with variations in conditions.
- Straightforward with MapReduce, maintains data integrity.

Facilitates efficient processing of set operations in large datasets.

Practical Example: Data Clustering

- **Scenario:** Classify data points $\{x_i\}$ closer to means μ_1 or μ_2.
- **Map:** Generate (i, x_j) if proximity to μ_i is met.
- **Reduce:** Average all x_i to update means.
- Enables scalable execution in line with clustering algorithms.

➜ critical for k-means.

Key Takeaways

- MapReduce transforms complex relational operations into manageable tasks across distributed systems.
- Facilitates execution of grouping, aggregation, and joins within machine learning workflows.
- Empowers handling of large relational datasets effectively, integral in modern data analytics.

2. Prerequisites from Matrix Analysis

Notation I

- The set of natural, integer, real, and complex numbers are denoted by $\mathbb{N}, \mathbb{Z}, \mathbb{R}, \text{and } \mathbb{C}$, respectively, while $\mathbb{R}_+$, indicates the set of nonnegative reals.
- The sets (a, b), $[a, b)$, $\text{and } [a, b]$ denote open, half-open, and closed intervals.
- Other sets are normally written by calligraphic letters.
- The union, the intersection, and the set theoretic difference of $\mathcal{A}$ and $\mathcal{B}$ are denoted by $\mathcal{A} \cup \mathcal{B}$, $\mathcal{A} \cap \mathcal{B}$, and $\mathcal{A} \setminus \mathcal{B}$, respectively.
- The optimal value of an optimization variable x is highlighted by a superscript asterisk as x^*
- We write λ^+ for the positive part of a real number λ, i.e., $\lambda^+ = \max\{0, \lambda\}$.
- Vectors are denoted by boldface lowercase letters.
- $\mathbf{0}_n$ and $\mathbf{1}_n$ are the all-zero and all-one vector of dimension n, respectively.
- The canonical basis vectors of $\mathbb{R}^m$ are written as $\mathbf{e}_1, \ldots, \mathbf{e}_m$
- The Euclidean norm of $\mathbf{x} \in \mathbb{R}^m$ is denoted by $\|\mathbf{x}\|$ or $\|\mathbf{x}\|_2$
- Boldface uppercase characters indicate matrices

Notation II

- A matrix $\mathbf{A}$ of size $m \times n$ with entries a_{ij} is written as $\mathbf{A} = \mathbf{A}_{m\times n} = (a_{ij})_{1\leq i\leq m, 1\leq j\leq n}$
- $\mathbf{A}^T$ and $\mathbf{A}^{-1}$ are the transpose and the inverse of some matrix $\mathbf{A}$.
- The determinant of $\mathbf{A}$ is denoted by $\det(\mathbf{A})$, alternatively also by $|\mathbf{A}|$.
- Some special matrices are the all-zero matrix $\mathbf{0}_{m\times n}$, the all-one matrix $\mathbf{1}_{m\times n}$, and the identity matrix $\mathbf{I}_n$.
- Diagonal matrices with all nondiagonal entries zero are denoted by $\mathbf{\Lambda} = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m)$
- A matrix $\mathbf{U} \in \mathbb{R}^{m\times m}$ is called orthogonal (or sometimes orthonormal) if $\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}_m$ holds.
- We denote the set of orthogonal matrices of size $m \times m$ by $\mathcal{O}_m$, i.e., $\mathcal{O}_m = \left\{\mathbf{U} \in \mathbb{R}^{m\times m} \,|\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}_m\right\}$
- The image or column space of some matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ is defined as $\operatorname{Img}(\mathbf{M}) = \{\mathbf{M}\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n\}$
- The kernel or null space of some matrix $\mathbf{M} \in \mathbb{R}^{m\times n}$ is defined as $\operatorname{Ker}(\mathbf{M}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{M}\mathbf{x} = \mathbf{0}_m\}$

Notation III

- The orthogonal complement of a subspace $\mathcal{V}$ of $\mathbb{R}^n$ is denoted by $\mathcal{V}^\perp = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{y}^T\mathbf{x} = 0 \text{ for all } \mathbf{y} \in \mathcal{V}\}$
  Hence, any two vectors $\mathbf{x} \in \mathcal{V}$ and $\mathbf{y} \in \mathcal{V}^\perp$ are orthogonal
- The linear span or linear hull of vectors $\mathbf{v}_1, \mathbf{v}_2, \cdot \cdot \cdot \cdot \mathbf{v}_n \in \mathbb{R}^m$ is the linear subspace formed of all linear combinations $\text{Span}(\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}) = \{\sum_{i=1}^n \alpha_i \mathbf{v}_i \mid \alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{R}\} \subseteq \mathbb{R}^m$
  - The spanning vectors may be linearly dependent, so the dimension of the linear span is at most $\min\{m, n\}$.

Outline

Motivation

- Consider
$$\mathbf{M}\mathbf{v} = \lambda\mathbf{v}$$
with $\mathbf{M} \in \mathbb{R}^{m\times m}$, $\lambda \in \mathbb{R}$ and $\mathbf{v} \in \mathbb{R}^m$, $\mathbf{v} \neq \mathbf{0}$.
- Geometric interpretation: Under transformation by $\mathbf{M}$, vector $\mathbf{v}$ experiences only a change in length or sign.
- The direction of $\mathbf{M}\mathbf{v}$ is the same as that of $\mathbf{v}$, so that $\mathbf{v}$ is stretched or shrunk or flipped.
- Vectors with this property are called eigenvectors, the scaling factor λ is called eigenvalue.
- It is clear that with any eigenvector $\mathbf{v}$ all multiples are also eigenvectors.
  - ➡ eigenvectors mostly considered to be normalized to length one.
- In this section we will deal with the problem of finding eigenvalues and eigenvectors of a given matrix $\mathbf{M}$.

Motivation

- Obviously the equation $\mathbf{M}\mathbf{v} = \lambda\mathbf{v}$ is equivalent to the so called **eigenvalue equation** $(\mathbf{M} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$.
- This is equivalent to finding some λ such that $\det(\mathbf{M} - \lambda\mathbf{I}) = 0$. (This means that the matrix $\mathbf{M} - \lambda\mathbf{I}$ considered as a linear transformation reduces the dimensionality. Only then it is possible that $(\mathbf{M} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$ for some $\mathbf{v}$)
- With the so obtained λ we can find the corresponding eigenvectors $\mathbf{v}$ by solving the equation $(\mathbf{M} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$.
- We will see that solution pairs $(\lambda, \mathbf{v})$ always exist, if the matrix $\mathbf{M}$ is symmetric.
- As a generalization, we will also consider the so-called **singular value equation** $\mathbf{M}\mathbf{w} - \sigma\mathbf{u} = 0$ with potentially rectangular $\mathbf{M}$, and find solutions $(\sigma, \mathbf{u}, \mathbf{w})$.
- The eigenvalue and singular value equations will be shown to be closely related.

Theorem (2.1: Eigenvalue Decomposition)

Let $\mathbf{M} \in \mathbb{R}^{n \times n}$ be a symmetric matrix. Then there exists an orthogonal matrix $\mathbf{V} \in \mathcal{O}_n$ and a diagonal matrix $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^{n \times n}$ such that

$$\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$$

.

- The diagonal entries of $\mathbf{\Lambda}$ are the eigenvalues of $\mathbf{M}$, and the columns $\mathbf{v}_i$ of $\mathbf{V}$ are the corresponding eigenvectors, satisfying

$$\mathbf{M}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

  for all $i = 1, \ldots, n$.
- The decomposition into $\mathbf{V}$ and $\mathbf{\Lambda}$ is called the **eigenvalue decomposition** and sometimes also calles **spectral decomposition** of $\mathbf{M}$.

REMARK 2.2

- Some number λ is an eigenvalue of the square matrix $\mathbf{M}$ if $\det(\mathbf{M} - \lambda\mathbf{I}) = \mathbf{0}$. Zero determinant means that $\mathbf{M} - \lambda\mathbf{I}$ is singular. Hence there exists some vector $\mathbf{v} \neq \mathbf{0}$ with $(\mathbf{M} - \lambda\mathbf{I})\mathbf{v} = \mathbf{0}$.

REMARK 2.3:

- From $\mathbf{M} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T$ it follows that a square matrix $\mathbf{M}$ can be written as a superposition of rank-one matrices in the form

$$\mathbf{M} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T.$$

- The set of eigenvalues defined by $\{\lambda \in \mathbb{C} \mid \det(\mathbf{M} - \lambda\mathbf{I}) = 0\}$ is called the **spectrum** of $\mathbf{M}$.
- The eigenvalues of a symmetric matrix are always real-valued.

Lemma (2.4: Positive and Nonnegative Definite)

Let $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$.

(a) $\boldsymbol{M}$ *is called* **positive definite**, *if $\lambda_i > 0$ for all $i = 1, \ldots, n$.*

(b) $\boldsymbol{M}$ *is called* **positive semidefinite** *(nonnegative definite), if $\lambda_i \geq 0$ for all $i = 1, \ldots, n$.*

(c) *If $\boldsymbol{M}$ is positive (semi-)definite, there exists a* **decomposition**

$$\boldsymbol{M} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T = \boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\left(\boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}\right)^T = \boldsymbol{C}\boldsymbol{C}^T$$

with $\boldsymbol{\Lambda}^{1/2} = \operatorname{diag}\left(\lambda_1^{1/2}, \cdots, \lambda_n^{1/2}\right)$ and $\boldsymbol{C} = \boldsymbol{V}\boldsymbol{\Lambda}^{\frac{1}{2}}$. $\mathbf{C}\mathbf{C}^T$ is called Cholesky decomposition.

➡ *If $\boldsymbol{M}$ is* **positive semidefinite** *then $\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{C}\boldsymbol{C}^T\boldsymbol{x} \geq 0$ for all $\boldsymbol{x} \in \mathbb{R}^n$.*

➡ *If $\boldsymbol{M}$ is* **positive definite** *then $\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^T\boldsymbol{x} = \boldsymbol{y}^T\boldsymbol{\Lambda}\boldsymbol{y} = \sum_{i=1}^n \lambda_i y_i^2 > 0$ for all $\boldsymbol{x} \in \mathbb{R}^n \backslash \{\mathbf{0}\}$.*

Lemma (2.4: Positive and Semipositive Definite – continued)

Let $\boldsymbol{M} \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$.

(d) *The identity matrix $\boldsymbol{I}_n$ is positive definite. The system of canonical unit vectors $\mathbf{e}_i$, $i = 1, \ldots, n$ forms a corresponding system of orthonormal eigenvectors, each with eigenvalue one. The columns of any other orthogonal matrix can also serve as a system of eigenvectors to eigenvalue 1.*

Example (2.5)

- Let $\mathbf{M} = \mathbf{A} + \mathbf{I}_n \in \mathcal{R}^{n \times n}$ be a real symmetric matrix and $\mu_1, \ldots, \mu_n$ be the eigenvalues of $\mathbf{A}$ with corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$.
- Then

$$\mathbf{M}\mathbf{v}_i = \mathbf{A}\mathbf{v}_i + \mathbf{v}_i = \mu_i \mathbf{v}_i + \mathbf{v}_i = (\mu_i + 1)\mathbf{v}_i, \ \nabla i = 1, \ \cdot\cdot, n,$$

  such that $\mathbf{M}$ and $\mathbf{A}$ have the same eigenvectors, and the eigenvalues of $\mathbf{M}$ are $\lambda_i = \mu_i + 1$, $i = 1, \ldots, n$.
- Hence, if $\mathbf{A}$ is nonnegative definite, then $\mathbf{M}$ is positive definite.

Example (2.6)

- Let $k \in \mathbb{N}$ and $\mathbf{M} \in \mathbb{R}^{n \times n}$ be a symmetric matrix with eigenvalues $\lambda_1, \ldots, \lambda_n$ and corresponding eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$.
- By iterating $\mathbf{M}^k \mathbf{v}_i = \mathbf{M}^{k-1} \mathbf{M} \mathbf{v}_i = \mathbf{M}^{k-1} \lambda_i \mathbf{v}_i$ we obtain

$$\mathbf{M}^k \mathbf{v}_i = \lambda_i^k \mathbf{v}_i, \quad i = 1, \ldots, n.$$

- The eigenvalues of $\mathbf{M}^k$ are λ_i^k, $i = 1, \ldots, n$ with the same eigenvectors, i.e.

$$\mathbf{M}^k = \mathbf{V} \operatorname{diag}(\lambda_1^k, \ldots, \lambda_n^k) \mathbf{V}^T.$$

- If k is even, then $\mathbf{M}^k$ is nonnegative definite, otherwise $\mathbf{M}^k$ and $\mathbf{M}$ have the same number of negative, positive and zero eigenvalues.

Singular value decomposition (SVD) is a generalization of the eigenvalue decomposition for rectangular matrices.

Theorem (2.7: Singular Value Decomposition)

For each $\mathbf{M} \in \mathbb{R}^{m \times n}$ there exist orthogonal matrices $\mathbf{U} \in \mathcal{O}_m$ and $\mathbf{W} \in \mathcal{O}_n$, and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ with nonnegative entries on its diagonal and zeros otherwise such that

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T.$$

- The diagonal entries of $\mathbf{\Sigma}$ are the singular values of $\mathbf{M}$, and the columns $\mathbf{u}_i$ and $\mathbf{w}_i$ of $\mathbf{U}$ and $\mathbf{W}$ are the corresponding left and right singular vectors, respectively.

REMARK 2.8

- Let $\sigma_1, \sigma_2, \ldots, \sigma_{\min\{m,n\}}$ denote the diagonal entries of $\mathbf{\Sigma}$ and $\mathbf{U} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m)$ und $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n)$. For all $i = 1, 2, ..., \min(m, n)$ it holds that

$$\mathbf{u}_i^T \mathbf{M} = \sigma_i \mathbf{w}_i^T \quad \text{and} \quad \mathbf{M}\mathbf{w}_i = \sigma_i \mathbf{u}_i \,.$$

- If $m \neq n$ then for all $\min(m, n) < i \leq \max(m, n)$ both right hand sides above are equal to 0.
- We obtain

$$\mathbf{M} = \sum_{i=1}^{\min\{m,n\}} \sigma_i \mathbf{u}_i \mathbf{w}_i^T \,.$$

Relation

Both the singular value decomposition and the spectral decomposition are closely related as follows.

- Let $\mathbf{M} \in \mathbb{R}^{m \times n}$ and $k = \min\{m, n\}$.
- Using the singular value decomposition $\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T$ with orthogonal matrices $\mathbf{U}$ and $\mathbf{W}$ we obtain

$$\mathbf{M}\mathbf{M}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{W}^T\mathbf{W}\mathbf{\Sigma}^T\mathbf{U}^T = \mathbf{U}\mathrm{diag}\big(\underbrace{\sigma_1^2, \cdots \cdot \sigma_k^2, 0, \ldots, 0}_{m \text{ entries}}\big)\mathbf{U}^T$$

and

$$\mathbf{M}^T\mathbf{M} = \mathbf{W}\mathbf{\Sigma}^T\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}\mathbf{W}^T = \mathbf{W}\mathrm{diag}\big(\underbrace{\sigma_1^2, \cdots \cdot \sigma_k^2, 0, \ldots, 0}_{n \text{ entries}}\big)\mathbf{W}^T$$

  - ➜ $\mathbf{U}$ and $\mathbf{W}$ are the eigenvectors of $\mathbf{M}\mathbf{M}^T$ and $\mathbf{M}^T\mathbf{M}$, respectively
  - ➜ The singular values of $\mathbf{M}$ are the square roots of the eigenvalues of $\mathbf{M}^T\mathbf{M}$ and $\mathbf{M}\mathbf{M}^T$.

➜ We can compute the singular value decomposition of any $\mathbf{M}$ by computing the eigenvalue decomposition of $\mathbf{M}^T\mathbf{M}$ and $\mathbf{M}\mathbf{M}^T$.

Efficient Computation of Eigenvalues

- In data science, often only a few dominant eigenvalues with corresponding eigenvectors are needed.
- There exist iterative methods, e.g.the von Mises iteration or the power iteration
  - determines solely the largest eigenvalue and the corresponding eigenvector.
  - Von Mises iteration is a very simple algorithm, which may converge slowly.

Theorem (2.9: Von Mises Iteration)

- *Let* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *be a symmetric matrix with eigenvalues* $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ *and corresponding eigenvectors* $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n$.
- *Assume that initial vector* $\mathbf{y}^{(0)}$ *is not orthogonal to the dominant eigenvector* $\mathbf{v}_1$
- *Then following iteration over* $k = 1, 2, \ldots$

$$\mathbf{y}^{(k)} = \mathbf{M}\mathbf{y}^{(k-1)}, \quad \mathbf{x}^{(k)} = \frac{\mathbf{y}^{(k)}}{||\mathbf{y}^{(k)}||}$$

and

$$\mu^{(k)} = \frac{\left(\mathbf{y}^{(k-1)}\right)^T \mathbf{y}^{(k)}}{||\mathbf{y}^{(k-1)}||^2}$$

yields

$$|\mathbf{v}_1^T \lim_{k \to \infty} \mathbf{x}^{(k)}| = 1 \quad \text{and} \quad \lim_{k \to \infty} \mu^{(k)} = \lambda_1$$

- *Hence,* $\mathbf{x}^{(k)}$ *converges to the direction of the dominant eigenvector* $\mathbf{v}_1$ *and* $\mu^{(k)}$ *converges to the dominant eigenvalue* λ_1.

Von Mises Iteration I

Proof

- With $\mathbf{M}^k = \sum_{i=1}^n \lambda_i^k \mathbf{v}_i \mathbf{v}_i^T$ it follows that

$$\mathbf{y}^{(k)} = \mathbf{M}^k \mathbf{y}^{(0)} = \sum_{i=1}^n \lambda_i^k \mathbf{v}_i \underbrace{\mathbf{v}_i^T \mathbf{y}^{(0)}}_{=\alpha_i} = \sum_{i=1}^n \lambda_i^k \alpha_i \mathbf{v}_i$$

- Since $\mathbf{y}^{(0)}$ is not orthogonal to $\mathbf{v}_1$, $\alpha_1 \neq 0$ holds and thus

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \lim_{k\to\infty} \frac{\sum_{i=1}^n \lambda_i^k \alpha_i \mathbf{v}_i}{\| \sum_{i=1}^n \lambda_i^k \alpha_i \mathbf{v}_i \|} = \lim_{k\to\infty} \frac{\lambda_1^k \alpha_1 \mathbf{v}_1 + \sum_{i=2}^n \lambda_i^k \alpha_i \mathbf{v}_i}{\| \lambda_1^k \alpha_1 \mathbf{v}_i + \sum_{i=2}^n \lambda_i^k \alpha_i \mathbf{v}_i \|} \tag{1}$$

$$= \lim_{k\to\infty} \frac{\lambda_1^k}{|\lambda_1^k|} \frac{\alpha_1 \mathbf{v}_1 + \sum_{i=2}^n (\lambda_i/\lambda_1)^k \alpha_i \mathbf{v}_i}{\| \alpha_1 \mathbf{v}_1 + \sum_{i=2}^n (\lambda_i/\lambda_1)^k \alpha_i \mathbf{v}_i \|} = \lim_{k\to\infty} \frac{\lambda_1^k \alpha_1 \mathbf{v}_1}{|\lambda_1^k| \, |\alpha_1| \, \|\mathbf{v}_1\|} \tag{2}$$

  as $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$

- Hence, $\mathbf{x}^{(k)}$ converges to the direction of the dominant eigenvector $\mathbf{v}_1$ and $\left| \mathbf{v}_1^T \lim_{k\mapsto\infty} \mathbf{x}^{(k)} \right| = 1$.

Proof

- Furthermore

$$\lim_{k\to\infty} \mu^{(k)} = \lim_{k\to\infty} \frac{(\mathbf{y}^{(0)})^T \mathbf{M}^{k-1}\mathbf{M}^k\mathbf{y}^{(0)}}{||\mathbf{M}^{k-1}\mathbf{y}^{(0)}||^2} = \lim_{k\to\infty} \frac{\sum_{i=1}^n \lambda_i^{2k-1}\alpha_i^2}{\sum_{i=1}^n \lambda_i^{2k-2}\alpha_i^2} \tag{3}$$

$$= \lambda_1 \lim_{k\mapsto\infty} \frac{\alpha_1 + \sum_{i=2}^n (\lambda_i/\lambda_1)^{2k-1}\alpha_i^2}{\alpha_1 + \sum_{i=2}^n (\lambda_i/\lambda_1)^{2k-2}\alpha_i^2} \tag{4}$$

$$= \lambda_1 \tag{5}$$

qed.

- Once the dominant eigenvalue λ_1 and the corresponding eigenvector $\mathbf{v}_1$ are determined, we can compute

$$\mathbf{M}_1 = \mathbf{M} - \hat{\boldsymbol{\lambda}}_1 \hat{\mathbf{v}}_1 \hat{\mathbf{v}}_1^T = \left(\lambda_1 \mathbf{v}_1 \mathbf{v}_1^T - \hat{\boldsymbol{\lambda}}_1 \hat{\mathbf{v}}_1 \hat{\mathbf{v}}_1^T \right) + \sum_{i=2}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^T$$

  and apply the power iteration to $\mathbf{M}_1$ for computing λ_2
- This can be iterated further as long as $|\lambda_t| > |\lambda_{t-1}|$.
- Care must be taken about numerical errors which build up iteratively because of the deviation between the numerically approximated and the true values.

Approximate Eigenvalues for Big Matrices I

- When dealing with big data, the computation of eigenvalues and eigenvectors is computationally expensive.
- In such cases, the computation of approximate eigenvalues and eigenvectors is often sufficient.
- Gershgorin's theorem provides a useful tool for bounding the eigenvalues by the sum of nondiagonal elements.

Theorem (2.10: Gershgorin's theorem)

Let $\mathbf{M} \in \mathbb{C}^{n \times n}$ with spectrum $\mathcal{S} = \{\lambda \in \mathbb{C} \mid \det(\mathbf{M} - \lambda \mathbf{I}_n) = 0\}$. For $i = 1, \dots n$ define Gershgorin circles

$$\mathcal{R}_i = \left\{ z \in \mathbb{C} \mid |z - m_{ii}| \leq \sum_{j=1, j \neq i}^{n} |m_{ij}| \right\}$$

and

$$\mathcal{C}_j = \left\{ z \in \mathbb{C} \mid |z - m_{jj}| \leq \sum_{i=1, i \neq j}^{n} |m_{ij}| \right\}$$

Then it holds that

$$\mathcal{S} \subseteq \bigcup_{i=1}^{n} (\mathcal{R}_i \cap C_i),$$

i.e., all eigenvalues of $\mathbf{M}$ are contained in at least one of the intersection of row- and column-wise Gershgorin circles.

Corollary (2.11)

If $\mathbf{M} = (m_{ij}) \in \mathbb{R}^{n \times n}$ is symmetric, then every eigenvalue of $\mathbf{M}$ lies within at least one of the intervals, $i = 1, ..., n$,

$$\Big[m_{ii} - \sum_{\substack{j=1 \\ j \neq i}} m_{ij},\ m_{ii} + \sum_{\substack{j=1 \\ j \neq i}} m_{ij}\Big].$$

Example (2.12)

$$\mathbf{M} = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 3 & -1 \\ 0 & -1 & 4 \end{pmatrix}$$

- Spectrum $\mathcal{S} = \{1.27, 3.00, 4.73\}$.
- The Gerschgorin bounds are derived from $[1, 3] \cup [1, 5] \cup [3, 5] = [1, 5]$, such that $1 \leq \lambda_i \leq 5$ for all $\lambda_i \in \mathcal{S}$ can be derived without computing the full spectrum.
- The bounds also show that M is positive definite.

Motivation

- The set of $m \times n$ matrices is obviously a linear vector space.
- Addition of matrices and multiplication by scalars is clearly defined.
- In this section we will endow this linear space by more structure.
- The trace operator will serve as a tool to introduce an inner product and a norm.
- This will allow for the concept of distance between matrices.
- Matrix computations become often much easier if the concept of partitioned matrices and corresponding calculus are available, so that we also introduce this basic tool.
- We commence by introducing two different norms, both of great importance for low-dimensional space approximations in data analytics.

Definition (2.13: matrix norms)

(a) For any $\mathbf{M} = (m_{ij}) \in \mathbb{R}^{m \times n}$

$$\|\mathbf{M}\|_{\mathrm{F}} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} m_{ij}^2 \right)^{1/2}$$

is called the **Frobenius norm** of $\mathbf{M}$.

(b) If $\mathbf{M} \in \mathcal{R}^{n \times n}$ is symmetric with eigenvalues $\lambda_1, \ldots, \lambda_n$, then

$$\|\mathbf{M}\|_S = \max_{1 \le i \le n} |\lambda_i|$$

is called the **spectral norm** of $\mathbf{M}$.

Lemma (2.14)

(a) *For any real vector* $\mathbf{x}$ *and any real matrix* $\mathbf{M}$ *of appropriate dimension*

$$\|\mathbf{M}\mathbf{x}\|_2 \leq \|\mathbf{M}\|_{F/S} \|\mathbf{x}\|_2$$

(b) *For any two matrices* $\mathbf{M}_1$ *and* $\mathbf{M}_1$ *of appropriate dimension*

$$\|\mathbf{M}_1\mathbf{M}_2\|_{F/S} \leq \|\mathbf{M}_1\|_{F/S} \|\mathbf{M}_2\|_{F/S}$$

(c) *For any real matrix* $\mathbf{M}$ *and real orthogonal matrices* $\mathbf{U}$ *and* $\mathbf{W}$ *of appropriate dimension*

$$\|\mathbf{U}\mathbf{M}\mathbf{W}\|_{F/S} = \|\mathbf{M}\|_{F/S}$$

Definition of the Trace of a Matrix

Definition (2.15)

The trace of a matrix $\mathbf{M} = (m_{ij}) \in \mathbb{R}^{n \times n}$ is defined as

$$\text{tr}(\mathbf{M}) = \sum_{i=1}^{n} m_{ii}$$

Properties of the Trace of a Matrix

Lemma (2.16)

(a) *The trace is commutative, i.e.,* $\mathrm{tr}(\mathbf{AB}) = \mathrm{tr}(\mathbf{BA})$.

(b) *The trace is linear, i.e.,* $\mathrm{tr}(\alpha\mathbf{A} + \beta\mathbf{B}) = \alpha\,\mathrm{tr}(\mathbf{A}) + \beta\,\mathrm{tr}(\mathbf{B})$.

(c) *For any symmetric matrix* $\mathbf{M}$*, the trace is the sum of the eigenvalues, i.e.,*

$$\mathrm{tr}(\mathbf{M}) = \sum_{i=1}^{n} \lambda_i.$$

and the determinant is the product of eigenvalues, i.e.,

$$\det(\mathbf{M}) = \prod_{i=1}^{n} \lambda_i.$$

(d) *For* $\mathbf{M} \in \mathbb{R}^{n\times n}$ *it holds that*

$$\mathrm{tr}(\mathbf{M}^T\mathbf{M}) = \|\mathbf{M}\|_F^2$$

Theorem (2.17: Schur Complement)

- *Consider the symmetric and invertible block matrix* $\mathbf{M} \in \mathbb{R}^{n \times n}$ *partitioned into invertible blocks* $\mathbf{A} \in \mathbb{R}^{m \times m}$, $\mathbf{B} \in \mathbb{R}^{m \times (n-m)}$,*and* $\mathbf{C} \in \mathbb{R}^{(n-m) \times (n-m)}$, $m \leq n$, *as*

$$\mathbf{M} = \begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{pmatrix}$$

- *Then*

$$\mathbf{M}^{-1} = \begin{pmatrix} \mathbf{A}^{-1} + \mathbf{F}\mathbf{E}^{-1}\mathbf{F}^T & -\mathbf{F}\mathbf{E}^{-1} \\ -\mathbf{E}^{-1}\mathbf{F}^T & \mathbf{E}^{-1} \end{pmatrix}$$

and

$$\det(\mathbf{M}) = \det(\mathbf{A}) \det(\mathbf{E}),$$

where the matrix

$$\mathbf{E} = \mathbf{C} - \mathbf{B}^T \mathbf{A}^{-1} \mathbf{B}$$

is called the **Schur complement** *of* $\mathbf{M}$, *and* $\mathbf{F} = \mathbf{A}^{-1}\mathbf{B}$.

Outline

Motivation

- Monotonicity for real functions with real arguments needs the ordering of the arguments.
- We call a real function $f : \mathbb{R} \to \mathbb{R}$ monotonically increasing if $f(a) \leq f(b)$, for all $a \leq b \in \mathbb{R}$.
- In this section we will carry over monotonicity to real functions defined on the set of matrices.
- We cannot expect that a full ordering on the set of matrices exists, in the sense that any two of them are comparable.
- However, if there is an ordering which at least applies to selected matrices, then monotonicity can be partially defined.
- Such orderings are called **semi-orderings**, saying that not necessarily all elements of a set can be compared w.r.t. the underlying ordering.
- The simple example of ordering matrices (or vectors) by comparing their real components "$\mathbf{A} \leq \mathbf{B}$, if $a_{ij} \leq b_{ij}$, for all entries" demonstrates that only selected matrices are comparable.
- More sophisticated semi-orderings are needed for data analytics. A most useful one was introduced by Karl Löwner in the class of symmetric matrices.

Definition (2.18: Loewner semi-ordering)

Let $\mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times n}$ be symmetric matrices. The Loewner semi-ordering $\geq_{\mathrm{L}}$ is defined by

$$\mathbf{V} \geq_{\mathrm{L}} \mathbf{W}$$

if $\mathbf{V} - \mathbf{W}$ is nonnegative definite (i.e. positive semidefinite)

- Corresondingly we write $\mathbf{V} \geq_{\mathrm{L}} \mathbf{0}$, if $\mathbf{V}$ is nonnegative definite.
- The reverse symbol $\leq_{\mathrm{L}}$ means $\mathbf{W} - \mathbf{V} \geq_{\mathrm{L}} \mathbf{0}$.
- The extension to positive definite matrices is straightforward: We use the notation $\mathbf{V} >_{\mathrm{L}} \mathbf{W}$, if $\mathbf{V} - \mathbf{W}$ is positive definite.

For any symmetric matrices $\mathbf{U}, \mathbf{V}, \mathbf{W} \in \mathbb{R}^{n \times n}$ and any real $\alpha \geq 0$ (i.e. $\alpha \in \mathbb{R}^{+}$) for the Loewner semi-ordering $\leq_{\mathrm{L}}$ the following properties hold:

(i) Reflexivity: $\mathbf{U} \leq_{\mathrm{L}} \mathbf{U}$.

(ii) Antisymmetry: $\mathbf{U} \leq_{\mathrm{L}} \mathbf{V}$ and $\mathbf{V} \leq_{\mathrm{L}} \mathbf{U}$ implies $\mathbf{U} = \mathbf{V}$.

(iii) Transitivity: $\mathbf{U} \leq_{\mathrm{L}} \mathbf{V}$ and $\mathbf{V} \leq_{\mathrm{L}} \mathbf{W}$ implies $\mathbf{U} \leq_{\mathrm{L}} \mathbf{W}$.

(iv) Additivity: $\mathbf{U} \leq_{\mathrm{L}} \mathbf{V}$ and $\mathbf{W} \geq_{\mathrm{L}} \mathbf{0}$ implies $\mathbf{U} + \mathbf{W} \leq_{\mathrm{L}} \mathbf{V} + \mathbf{W}$.

(v) Scalability: $\mathbf{U} \leq_{L} \mathbf{V}$ implies $\alpha \mathbf{U} \leq_{\mathrm{L}} \alpha \mathbf{V}$

Theorem (2.19)

By the Loewner semi-ordering the concept of monotonicity becomes available

Given $\mathbf{V} = (v_{ij}) \geq_{\mathrm{L}} \mathbf{0}$ *and* $\boldsymbol{W} = (w_{ij}) \geq_{\mathrm{L}} \mathbf{0}$ *with* $\boldsymbol{V} \leq_{\mathrm{L}} \mathbf{W}$, *and eigenvalues* $\lambda_1(\mathbf{V}) \geq \lambda_2(\mathbf{V}) \geq \cdots \geq \lambda_n(\mathbf{V})$ *and* $\lambda_1(\mathbf{W}) \geq \lambda_2(\mathbf{W}) \geq \cdots \cdot \geq \lambda_n(\mathbf{W})$. *Then*

(a) $\lambda_i(\mathbf{V}) \leq \lambda_i(\mathbf{W})$, *for all* $i = 1, \ldots, n$.

(b) $v_{ii} \leq w_{ii}$ *for all* $i = 1, \ldots, n$.

(c) $v_{ii} + v_{jj} - 2v_{ij} \leq w_{ii} + w_{jj} - 2w_{ij}$ *for all* $i, j = 1, \ldots, n$

(d) $\mathrm{tr}(\mathbf{V}) \leq \mathrm{tr}(\mathbf{W})$.

(e) $\det(\mathbf{V}) \leq \det(\mathbf{W})$

(f) $\mathrm{tr}(\mathbf{MV}) \leq \mathrm{tr}(\mathbf{MW})$ *for any* $\mathbf{M} \geq_L \mathbf{0}$

Once (a) is proved, the other assertions follow

Example 2.20

- For a matrix $\mathbf{M} = \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix}$ with entries $\alpha, \beta, \gamma \in \mathbb{R}$, the eigenvalues are given by:

$$\lambda_{1/2}(\mathbf{M}) = \frac{\alpha + \gamma}{2} \pm \frac{\sqrt{(\alpha - \gamma)^2 + 4\beta^2}}{2}$$

- Consider the matrices:

$$\mathbf{A} = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 5 & 2 \\ 2 & 2 \end{pmatrix}, \quad \mathbf{C} = \begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix}$$

- By the given formula, the eigenvalues are:

$$\begin{aligned} \lambda_1(\mathbf{A}) &= 5, & \lambda_2(\mathbf{A}) &= 0, \\ \lambda_1(\mathbf{B}) &= 6, & \lambda_2(\mathbf{B}) &= 1, \\ \lambda_1(\mathbf{C}) &= 6, & \lambda_2(\mathbf{C}) &= 1. \end{aligned}$$

Example 2.20

- Hence
  - Matrix $\mathbf{A}$ is non-negative definite: $\mathbf{A} \geq_L 0$.
  - Matrices $\mathbf{B}$ and $\mathbf{C}$ are positive definite: $\mathbf{B} >_L 0$, $\mathbf{C} >_L 0$.
- Differences between matrices:

$$\mathbf{B} - \mathbf{A} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \mathbf{C} - \mathbf{A} = \begin{pmatrix} -2 & 0 \\ 0 & 4 \end{pmatrix}, \quad \mathbf{C} - \mathbf{B} = \begin{pmatrix} -3 & 0 \\ 0 & 3 \end{pmatrix}$$

- $\mathbf{B} - \mathbf{A}$ is non-negative definite: $\mathbf{B} \geq_L \mathbf{A}$.
- $\mathbf{C} - \mathbf{A}$ and $\mathbf{C} - \mathbf{B}$ are indefinite, so $\mathbf{C}$ is not comparable with either $\mathbf{A}$ or $\mathbf{B}$.
- This demonstrates that $\geq_L$ is a semi-ordering.

Example 2.20

- Since $\mathbf{B} \geq_L \mathbf{A}$, all inequalities in Theorem 2.19 are satisfied by $\mathbf{A}$ and $\mathbf{B}$.
- According to the last inequality in Theorem 2.19:

$$\text{tr}(\mathbf{CA}) = \text{tr}\left(\begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix}\begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}\right) = 21$$

$$\text{tr}(\mathbf{CB}) = \text{tr}\left(\begin{pmatrix} 2 & 2 \\ 2 & 5 \end{pmatrix}\begin{pmatrix} 5 & 2 \\ 2 & 2 \end{pmatrix}\right) = 28$$

3. Multivariate Distributions and Moments

Outline

4. Dimensionality Reduction

5. Classification and Clustering

6. Support Vector Machines

7. Machine Learning

Outline