Eigenvectors and Diagonalizable Matrices

Chapter 3 - Linear Algebra and Optimization for Machine Learning

Pratham Gupta

Indian Institute of Science

January 27, 2025

Acknowledgements

These slides are made for Winter Reading Project in **Mathematics for Machine Learning** 2024-2025 under the guidance of **Prof. Chandrasekaran Pandurangan** at **Indian Institute of Science, Bengaluru**.

I have utilized the following resources for making these slides:

- **Textbook Followed:** *Linear Algebra and Optimization for Machine Learning* by Charu C. Aggarwal.
- **Images:** I have utilized images from the textbook and other online resources.
- **Other References:**
  - *Linear Algebra by Hoffman and Kunze* for Linear Algebra
  - *Matrix computations by Golub and Van Loan* for Numerical Methods
  - Other Online resources

Overview

1. Preliminaries
    - Linear Algebra
    - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
    - Determinants
    - Eigenvectors and Eigenvalues
    - Diagonalization
    - Spectral Theorem
3. Machine Learning and Optimization
    - Diagonalizable Matrices in Machine Learning
    - PCA
    - Quadratic Optimization
    - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Goals of this Presentation

- **Build Foundation for Further Presentations:** Understanding Linear Algebra concepts is crucial for Machine Learning.
- **Understanding How Linear Algebra Plays Role in Machine Learning**
- **Learn Some Numerical Methods**

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Scalars

- **Scalar**
  - A scalar is a single numerical value, often represented as $a \in \mathbb{R}$, where $\mathbb{R}$ denotes the set of real numbers.
  - In general, scalars belong to a field $\mathbb{F}$, for example, $\mathbb{R}$ or $\mathbb{C}$ (real or complex numbers).
  - We will work with real numbers for the most part and sometimes with complex numbers.
- **Vector**
  - Vectors are arrays of numerical values. A d-dimensional vector is represented as:

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix}, \quad \text{where } v_i \in \mathbb{R}.$$

Vector Operations

- **Addition:** $(\mathbf{v} + \mathbf{w})_i = v_i + w_i$
- **Scalar Multiplication:** $(c\mathbf{v})_i = cv_i$, where $c \in \mathbb{R}$.
- **Dot Product:** $\mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^{d} v_i w_i$
- **Norm:** $\|\mathbf{v}\|_2 = \sqrt{\sum_{i=1}^{d} v_i^2}$

Matrices

A matrix is a rectangular array of numerical values represented as:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Matrix Operations

- **Addition:** $(\mathbf{A}+\mathbf{B})_{ij} = a_{ij} + b_{ij}$
- **Scalar Multiplication:** $(c\mathbf{A})_{ij} = ca_{ij}$
- **Matrix Multiplication:** $(\mathbf{AB})_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$
- **Transpose:** $(\mathbf{A}^T)_{ij} = a_{ji}$
- **Inverse (if it exists):** $\mathbf{AA}^{-1} = \mathbf{I}$

Linear Transformations

A **linear transformation** is a mapping $T : \mathbb{R}^d \to \mathbb{R}^m$ defined by:

$$T(\mathbf{v}) = \mathbf{A}\mathbf{v},$$

where $\mathbf{A}$ is an $m \times d$ matrix.

Properties of Linear Transformations:

- **Additivity:** $T(\mathbf{v} + \mathbf{w}) = T(\mathbf{v}) + T(\mathbf{w})$
- **Homogeneity:** $T(c\mathbf{v}) = cT(\mathbf{v})$

Special Classes of Matrices

- **Square Matrix:** Number of rows equals the number of columns.
- **Diagonal Matrix:** Non-zero entries only on the diagonal.

$$\begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

- **Symmetric Matrix:** $\mathbf{A} = \mathbf{A}^T$.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{12} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nn} \end{bmatrix}$$

- **Orthogonal Matrix:** $\mathbf{A}\mathbf{A}^T = \mathbf{I}$.
- **Sparse Matrix:** Most entries are zero.

Special Classes of Matrices

- **Permutation Matrix:** Obtained by permuting rows of the identity matrix.

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- **Positive Definite Matrix:** $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$ for all $\mathbf{x} \neq \mathbf{0}$.
- **Positive Semi-Definite Matrix:** $\mathbf{x}^T \mathbf{A} \mathbf{x} \geq 0$ for all $\mathbf{x}$.

Some Common Operations on Matrices

- **Transpose:** Rows become columns and vice versa.

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \implies \mathbf{A}^T = \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

- **Conjugate Transpose:** Complex conjugate of the transpose.

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \implies \mathbf{A}^H = \mathbf{A}^\dagger = \begin{bmatrix} \bar{a} & \bar{c} \\ \bar{b} & \bar{d} \end{bmatrix}$$

- **Trace:** Sum of diagonal elements of a square matrix.

$$\mathrm{tr}(\mathbf{A}) = \sum_{i=1}^{n} a_{ii}$$

- **Determinant:** Product of eigenvalues of a square matrix.

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

- **Rank:** Dimension of the column space of a matrix.

Vector Spaces in General

A **vector space** over a field $\mathbb{F}$ is a non-empty set $\mathbf{V}$ together with a binary operation $+$ and a binary function $\cdot$ that satisfy the eight axioms listed below. In this context, the elements of $\mathbf{V}$ are commonly called vectors, and the elements of $\mathbb{F}$ are called scalars.

- **Associativity of Vector Addition:** $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$.
- **Commutativity of vector addition:** $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$.
- **Identity element of vector addition:** There exists a zero vector $\mathbf{0}$ such that $\mathbf{v} + \mathbf{0} = \mathbf{v}$.
- **Inverse elements of vector addition:** For every $\mathbf{v}$, there exists an element $-\mathbf{v}$ such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$.
- **Compatibility of scalar multiplication with field multiplication:** $a(b\mathbf{v}) = (ab)\mathbf{v}$.
- **Identity element of scalar multiplication:** $1\mathbf{v} = \mathbf{v}$.
- **Distributivity of scalar multiplication with respect to vector addition:** $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$.
- **Distributivity of scalar multiplication with respect to field addition:** $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$.

Bases, Vector Coordinates and Subspaces

- **Span:** The set of all possible linear combinations of a set of vectors.
- **Linear Independence:** A set of vectors is linearly independent if no vector can be expressed as a linear combination of the others.
- **Subspace:** A subset of a vector space that is itself a vector space.
- **Basis:** A set of linearly independent vectors that span a vector space.
- **Dimension:** The number of vectors in a basis.
- **Vector Coordinates:** The coefficients of a vector with respect to a basis.

If we have a basis $\mathbf{B}$ for a vector space $\mathbf{V}$, then any vector $\mathbf{v} \in \mathbf{V}$ can be expressed as a linear combination of the basis vectors:

$$\mathbf{v} = \sum_{i=1}^{n} c_i \mathbf{b}_i$$

then the coefficients c_i are the coordinates of $\mathbf{v}$ with respect to the basis $\mathbf{B}$.

$$v = (c_1, c_2, \ldots, c_n)$$

What are matrices in this context

Definition

Let $\mathbf{V}$ and $\mathbf{W}$ be vector spaces over a field $\mathbb{F}$. A linear transformation $T : \mathbf{V} \to \mathbf{W}$ is a function that satisfies the following properties:

- $T(\mathbf{v} + \mathbf{w}) = T(\mathbf{v}) + T(\mathbf{w})$ for all $\mathbf{v}, \mathbf{w} \in \mathbf{V}$
- $T(c\mathbf{v}) = cT(\mathbf{v})$ for all $c \in \mathbb{F}$ and $\mathbf{v} \in \mathbf{V}$

If $\mathbf{V}$ and $\mathbf{W}$ are finite-dimensional vector spaces with bases $\mathbf{B} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ and $\mathbf{C} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_m\}$ respectively, then the linear transformation T can be represented as a matrix $\mathbf{A}$ such that

$$T(\mathbf{v}_i) = \sum_{j=1}^{m} a_{ji}\mathbf{w}_j$$

where a_{ji} are the elements of the matrix $\mathbf{A}_{\mathbf{B},\mathbf{C}}$.
Where $\mathbf{A}_{\mathbf{B},\mathbf{C}}$ is the matrix representation of the linear transformation T with respect to the bases $\mathbf{B}$ and $\mathbf{C}$.

Rank and Nullity

- **Rank:** The dimension of the column space of a matrix.
- **Nullity:** The dimension of the null space of a matrix.
- **Rank-Nullity Theorem:** For a matrix $\mathbf{A}$, the rank and nullity are related by

$$\text{rank}(\mathbf{A}) + \text{nullity}(\mathbf{A}) = \text{number of columns of } \mathbf{A}$$

Dot Products and Inner Products

- **Dot Product:** A special case of the inner product for real vector spaces.

$$\mathbf{v} \cdot \mathbf{w} = \sum_{i=1}^{n} v_i w_i$$

- **Complex Dot Product:** A generalization of the dot product to complex vector spaces.

$$\langle \mathbf{v}, \mathbf{w} \rangle = \sum_{i=1}^{n} v_i \bar{w}_i$$

- **Norm:** The square root of the inner product of a vector with itself.

$$\|\mathbf{v}\| = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}$$

Understanding Matrices

Matrices as a array of numbers does not give us much insight into the structure of the underlying transformation.
So we generally try to decompose the matrix into simpler forms which we understand.
Some matrices we understand are:

- Diagonal Matrices
- Rotational Matrices
- Reflections
- Elementary Matrices

What are Diagonal Matrices

Definition

A matrix is said to be diagonal if all the elements outside the main diagonal are zero.

$$\begin{bmatrix} a_{11} & 0 & 0 & \cdots & 0 \\ 0 & a_{22} & 0 & \cdots & 0 \\ 0 & 0 & a_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{nn} \end{bmatrix}$$

Geometrically, a diagonal matrix scales the input vector along the coordinate axes.

What are Rotational Matrices

Definition

A matrix is said to be a rotational matrix if it rotates the input vector by a certain angle.

example in 2 dimensions:

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

In higher dimensions, we have Orthogonal matrices which are generalization of rotational matrices.
We classify these matrices into a group called Special Orthogonal Group $\mathbf{SO(n)}$ which are matrices with determinant 1.

What are Reflection Matrices

Definition

A matrix is said to be a reflection matrix if it reflects the input vector along a certain axis.

example in 2 dimensions:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}$$

In higher dimensions, we have Householder matrices which are generalization of reflection matrices.

$$\mathbf{P} = \mathbf{I} - 2\mathbf{v}\mathbf{v}^{\dagger}$$

where $\mathbf{v}$ is the vector we wish to reflect along.
Also $\mathbf{u}\mathbf{v}^{T}$ is outer product of the vector u and v.

What are Elementary Matrices

Definition

A matrix is said to be an elementary matrix if it is obtained by performing a single elementary row operation on the identity matrix.

Elementary row operations are:

- Interchanging two rows.
- Multiplying a row by a non-zero scalar.
- Adding a multiple of one row to another row.

Elementary matrices are used in Gaussian Elimination and LU Decomposition.
Example:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$$

LU Decomposition

A matrix **A** can be decomposed as:

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

where:

- **L** is a lower triangular matrix.
- **U** is an upper triangular matrix.

Some Applications of LU Decomposition are:

- Solving systems of linear equations.
- Inverting matrices.
- Finding basis Sets.

Properties of LU Decomposition

- If $\mathbf{A}$ is non-singular, the LU decomposition is unique.
- If $\mathbf{A}$ is non-singular and $a_{11} = 0$, then there are no there are no $\mathbf{L}$ & $\mathbf{U}$ such that $\mathbf{A} = \mathbf{LU}$.
- To obtain the LU decomposition in such situtation, we can use the $\mathbf{PA = LU}$ decomposition where $\mathbf{P}$ is a permutation matrix.
  - All Square matrices have a $\mathrm{PA} = \mathrm{LU}$ decomposition.
  - One can perform following decompositions also

$$\mathbf{PAQ} = \mathbf{LU}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are permutation matrices for rows and columns respectively.

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

where $\mathbf{L}$ is a lower triangular matrix with 1s on the diagonal, $\mathbf{D}$ is a diagonal matrix and $\mathbf{U}$ is an upper triangular matrix with 1s on the diagonal.

Algorithm for LU Decomposition

1. Start with **A**.
2. Perform Gaussian elimination to obtain **U**.

$$\begin{bmatrix}1 & 2 & 3\\4 & 5 & 6\\7 & 8 & 9\end{bmatrix} \xrightarrow{R_2-4R_1, R_3-7R_1} \begin{bmatrix}1 & 2 & 3\\0 & -3 & -6\\0 & -6 & -12\end{bmatrix} \xrightarrow{R_3-2R_2} \begin{bmatrix}1 & 2 & 3\\0 & -3 & -6\\0 & 0 & 0\end{bmatrix}$$

3. Record the operations in a lower triangular matrix **L**.

$$\mathbf{L_1} = \begin{bmatrix}1 & 0 & 0\\4 & 1 & 0\\7 & 0 & 1\end{bmatrix} \mathbf{L_2} = \begin{bmatrix}1 & 0 & 0\\0 & 1 & 0\\0 & 2 & 1\end{bmatrix} \mathbf{L} = \mathbf{L_1}\mathbf{L_2} = \begin{bmatrix}1 & 0 & 0\\4 & 1 & 0\\7 & 2 & 1\end{bmatrix}$$

Gram-Schmidt Orthogonalization

- Given a set of linearly independent vectors $\{v_1, v_2, \dots, v_n\}$, the goal of Gram-Schmidt is to construct an orthogonal basis set $\{u_1, u_2, \dots, u_n\}$ such that:

$$\langle u_i, u_j \rangle = 0 \quad \text{for} \quad i \neq j$$

Procedure:

- Start with the first vector v_1. Set the first orthogonal vector as $u_1 = v_1$.
- For each subsequent vector v_k $(k = 2, 3, \dots, n)$, subtract the projection of v_k onto each of the previously computed orthogonal vectors:

$$u_k = v_k - \sum_{i=1}^{k-1} \frac{\langle v_k, u_i \rangle}{\langle u_i, u_i \rangle} u_i$$

- Normalize each u_k if an orthonormal set is required:

$$e_k = \frac{u_k}{\|u_k\|}$$

where $\|u_k\|$ is the norm of u_k.

QR Decomposition

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

where:

- $\mathbf{Q}$ is an orthogonal matrix.
- $\mathbf{R}$ is an upper triangular matrix.

Algorithm:

1. Start with $n \times d$ matrix $\mathbf{A}$ with linearly independent columns.
2. Perform Gram-Schmidt orthogonalization to obtain $n \times d$ matrix $\mathbf{Q}$ with orthonormal columns.
3. Compute $d \times d$ matrix $\mathbf{R}$ as $\mathbf{Q}^T\mathbf{A}$.

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Basic Problems in Machine Learning

Machine learning involves building models that capture patterns in data to perform tasks such as prediction and classification. Key problems include:

- **Matrix Factorization:** Used for dimensionality reduction and recommender systems.
- **Clustering:** Grouping data points based on similarity.
- **Classification and Regression:** Predicting labels or continuous values from input features.
- **Outlier Detection:** Identifying anomalous data points.

Our method of choice to solve these problems it obtain a lot of data and then use Techniques to extract patterns and stastical relationships from the data.

Notation: We assume to have a dataset of n samples and d features. The dataset is represented as a $n \times d$ matrix $\mathbf{D}$. Each row of $\mathbf{D}$ corresponds to a sample and each column corresponds to a feature. The i-th row of $\mathbf{D}$ is denoted by $\mathbf{X}_i$.

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning

2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem

3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization

4. Numerical Methods for finding Eigenvalues

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Determinants

Determinant is a map from a square matrix to underlying scalar field. The determinant of a matrix $\mathbf{A}$ is denoted by $\det(\mathbf{A})$ or $|\mathbf{A}|$.

Definition

Recursive Definition of Determinant: The determinant of a 1×1 matrix is the single entry of the matrix. For a $d \times d$ matrix $\mathbf{A}$, the determinant is defined as:

$$\det(\mathbf{A}) = \sum_{i=1}^{n} (-1)^{i+j} a_{ij} \det(\mathbf{A}_{ij})$$

where $\mathbf{A}_{ij}$ is the $(n-1) \times (n-1)$ matrix obtained by deleting the i-th row and j-th column of $\mathbf{A}$.

Determinants

Definition

Explicit Formula for Determinant: For a $d \times d$ matrix $\mathbf{A} = [a_{ij}]$ and let Σ be set of all d! permutations of $[1, 2, \ldots, d]$
Let $\text{sgn}(\sigma)$ be the sign of the permutation σ defined as :

$$\text{sgn}(\sigma) = \begin{cases} +1 & \text{if } \sigma \text{ is reached by an even number of interchange} \\ -1 & \text{ otherwise} \end{cases}$$

Then the determinant of $\mathbf{A}$ is given by:

$$\det(\mathbf{A}) = \sum_{\sigma \in \Sigma} (\text{sgn}(\sigma) \prod_{i=1}^{d} a_{i\sigma(i)})$$

where $\sigma = \sigma_1 \sigma_2 \ldots \sigma_d \in \Sigma$ here σ_i is permutate value.

Determinants

Alternate Definition of Determinant: The determinant function of a $d \times d$ matrix $\mathbf{A}$ is a unique function that satisfies the following properties:

- **Normalization:** $\det(\mathbf{I}) = 1$, where $\mathbf{I}$ is the identity matrix.
- **Degeneracy:** $\det(\mathbf{A}) = 0$ if two rows are equal

Determinants

- **Homogeneity:** If one of the rows of $\mathbf{A}$ is multiplied by a scalar c, the determinant of $\mathbf{A}$ is scaled by c. Specifically, if

$$\mathbf{A} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_i \\ \vdots \\ \mathbf{r}_d \end{pmatrix},$$

where $\mathbf{r}_i$ denotes the i-th row of $\mathbf{A}$, and the i-th row is multiplied by c, then:

$$\det \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ c \cdot \mathbf{r}_i \\ \vdots \\ \mathbf{r}_d \end{pmatrix} = c \cdot \det \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_i \\ \vdots \\ \mathbf{r}_d \end{pmatrix}.$$

Determinants

- **Additivity:** For

$$\mathbf{A} = \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_j + \mathbf{r}_j' \\ \vdots \\ \mathbf{r}_d \end{pmatrix},$$

then:

$$\det \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_j + \mathbf{r}_j' \\ \vdots \\ \mathbf{r}_d \end{pmatrix} = \det \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_j \\ \vdots \\ \mathbf{r}_d \end{pmatrix} + \det \begin{pmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \vdots \\ \mathbf{r}_j' \\ \vdots \\ \mathbf{r}_d \end{pmatrix}.$$

Determinants

Geometric Interpretation of Determinants: The determinant of a matrix **A** can be interpreted as the volume of the parallelepiped spanned by the column vectors of **A**. The absolute value of the determinant gives the volume of the parallelepiped, and the sign of the determinant indicates the orientation of the parallelepiped.

It can also be interpreted as the scaling factor for generalized volume of the transformation represented by the matrix.

Properties of Determinants:

- The determinant of a matrix is zero if and only if the matrix is singular.
- The determinant of a product of matrices is the product of the determinants of the matrices.
- The determinant of the transpose of a matrix is equal to the determinant of the matrix.
- The determinant of the inverse of a matrix is the reciprocal of the determinant of the matrix.
- The determinant of a diagonal matrix is the product of the diagonal elements.

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Eigenvectors and Eigen Values

Definition

An **eigenvector** of a $d \times d$ matrix $\mathbf{A}$ is a non-zero vector $\mathbf{v}$ if it follows the relation

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}$$

where λ is a scalar called the eigenvalue of $\mathbf{v}$.

One can view that $\mathbf{v}$ is a eigen vector of $\mathbf{A}$ with eigen value λ if it belongs to the null space of the matrix $(\mathbf{A} - \lambda\mathbf{I})$.

Definition

The **characteristic polynomial** of a $d \times d$ matrix $\mathbf{A}$ is the degree d polynomial in λ obtained by expanding $\det(\mathbf{A} - \lambda\mathbf{I})$

Characteristic Polynomial

Observation

Eigen Values of a $d \times d$ matrix $\mathbf{A}$ are roots of the characteristic polynomial $f(\lambda)$ of the $\mathbf{A}$.
Reason: If λ_i is an eigen value of $\mathbf{A}$, then $\exists \mathbf{v} \neq 0$ such that $(\mathbf{A} - \lambda_i \mathbf{I})\mathbf{v} = 0$ since $(\mathbf{A} - \lambda_i \mathbf{I})$ is not full rank, so its determinant is zero.

let $\lambda_1, \lambda_2, \ldots, \lambda_d$ be the eigenvalues of $d \times d$ matrix $\mathbf{A}$. Then the characteristic polynomial of $\mathbf{A}$ is given by

$$f(\lambda) = (\lambda - \lambda_1)(\lambda - \lambda_2) \ldots (\lambda - \lambda_d)$$

In general, if $\lambda_1, \lambda_2, \ldots, \lambda_k$ are the eigenvalues of $d \times d$ matrix A, then the characteristic polynomial of $\mathbf{A}$ is given by

$$f(\lambda) = \prod_{i=1}^{k} (\lambda_i - \lambda)^{r_i}$$

where r_i is the algebraic multiplicity of the eigenvalue λ_i.

Algorithm

- To find eigenvalues of a matrix $\mathbf{A}$, we find the roots of the characteristic polynomial $f(\lambda)$ of $\mathbf{A}$.
- To find the eigenvectors of a matrix $\mathbf{A}$, we solve the equation $(\mathbf{A} - \lambda_i \mathbf{I})\mathbf{v} = 0$ for each eigenvalue λ_i.
    - This is done by row reducing the matrix $(\mathbf{A} - \lambda_i \mathbf{I})$ to row echelon form and solving for the free variables.

Cayley Hamilton Theorem

Theorem

Let $\mathbf{A}$ be any $d \times d$ matrix with characteristic polynomial $f(\lambda) = det(A - \lambda I)$. Then $f(\mathbf{A})$ evaluates to zero matrix.

Consider the matrix

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

The characteristic polynomial of $\mathbf{A}$ is given by

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

According to the Cayley-Hamilton theorem, $f(\mathbf{A}) = 0$. Therefore,

$$\mathbf{A}^2 - 4\mathbf{A} + 3\mathbf{I} = 0$$

Simplifying, we get

$$\mathbf{A}^2 = 4\mathbf{A} - 3\mathbf{I}$$

Proof of Cayley Hamilton

Definition

We define the adjoint of a matrix $\mathbf{A}$ as the matrix $\text{adj}(\mathbf{A})$ such that

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

A explicit construction of the adjoint is given by transpose of a $d \times d$ matrix $\mathbf{B}$ such that the ijth element of $\mathbf{B}$ is the determinant of the matrix obtained by deleting the ith row and jth column of $\mathbf{A}$ multiplied by $(-1)^{i+j}$.

Proof:

Let $\mathbf{B} = \text{adj}(t\mathbf{I} - A) = \sum_{k=0}^{n-1} t^k \mathbf{B}_k$ According to the definition of adjoint, we have

$$(t\mathbf{I} - \mathbf{A})\mathbf{B} = (t\mathbf{I} - A)\text{adj}(t\mathbf{I} - \mathbf{A}) = \det(t\mathbf{I} - A)\mathbf{I} = f(t)\mathbf{I}$$

Proof of Cayley Hamilton

Now expanding Left hand side we get

$$
\begin{aligned}
f(t)\mathbf{I} = (t\mathbf{I}-\mathbf{A})\mathbf{B} &= (t\mathbf{I}-\mathbf{A})\sum_{k=0}^{n-1} t^k\mathbf{B}_k \\
&= \sum_{k=0}^{n-1} t^k(t\mathbf{I}-\mathbf{A})\mathbf{B}_k \\
&= \sum_{k=0}^{n-1} t^k(t\mathbf{I}-\mathbf{A})\mathbf{B}_k \\
&= \sum_{k=0}^{n-1} t^k(t\mathbf{I}\mathbf{B}_k-\mathbf{A}\mathbf{B}_k) \\
&= \sum_{k=0}^{n-1} t^{k+1}\mathbf{B}_k - \sum_{k=0}^{n-1} t^k\mathbf{A}\mathbf{B}_k \\
&= t^n\mathbf{B}_{n-1} + \sum_{k=1}^{n-1} t^k(\mathbf{B}_{k-1}-\mathbf{A}\mathbf{B}_i) - \mathbf{A}\mathbf{B}_0
\end{aligned}
$$

Proof of Cayley Hamilton

let $f(t)\mathbf{I} = t^n\mathbf{I} + t^{n-1}c_{n-1}\mathbf{I} + \cdots + tc_1\mathbf{I} + c_0\mathbf{I}$,
Comparing the coefficients of t^i on both sides, we get

$$\begin{aligned}
\mathbf{B}_{n-1} &= \mathbf{I}\\
\mathbf{B}_{n-2} - \mathbf{A}\mathbf{B}_{n-1} &= c_{n-1}\mathbf{I}\\
\mathbf{B}_{n-3} - \mathbf{A}\mathbf{B}_{n-2} &= c_{n-2}\mathbf{I}\\
&\vdots\\
\mathbf{B}_{0} - \mathbf{A}\mathbf{B}_{1} &= c_{1}\mathbf{I}\\
-\mathbf{A}\mathbf{B}_{0} &= c_{0}\mathbf{I}
\end{aligned}$$

Now we left multiply equation of t^i coeff by $\mathbf{A}^i$ and sum over i to get

$$\mathbf{A}^n + c_{n-1}\mathbf{A}^{n-1} + \cdots + c_1\mathbf{A} + c_0\mathbf{I} = 0$$

Hence proved.

Applications of Cayley Hamilton Theorem

We can use Cayley Hamilton theorem to find the inverse of a matrix.

Theorem

Let $\mathbf{A}$ be any $d \times d$ invertible matrix with characteristic polynomial.

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

Then $f(\mathbf{A}) = \mathbf{A}[g(\mathbf{A})] + c\mathbf{I}$ where g($\mathbf{A}$) is a polynomial of degree $d-1$ and c is a constant.

Then inverse of $\mathbf{A}$ is given by

$$\mathbf{A}^{-1} = -\frac{1}{c} g(\mathbf{A})$$

Complex Eigenvalues

Definition

A field $\mathbb{F}$ is algebraically closed if every non-constant polynomial in $\mathbb{F}$[x] (the univariate polynomial ring with coefficients in $\mathbb{F}$) has a root in $\mathbb{F}$

- Uptil now we have only considered Linear transformations with real entries.
- However, real field is not algebraically closed. ex. $x^2 + 1 = 0$ has no real roots.
- Hence we may not always have real eigenvalues for a matrix.
- We can extend the real field to complex field to find complex eigenvalues.
- We can also find eigenvectors in complex field corresponding to complex eigenvalues.

Geometrically complex eigenvalues correspond to rotation and scaling of the vector space.

Complex Eigenvalues

Consider the matrix

$$B = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

The characteristic polynomial of B is given by

$$f(\lambda) = \det(B - \lambda I) = \det \begin{pmatrix} -\lambda & -1 \\ 1 & -\lambda \end{pmatrix} = \lambda^2 + 1$$

The roots of the characteristic polynomial are $\lambda = i$ and $\lambda = -i$, which are complex eigenvalues.
To find the eigenvectors, we solve the equation $(B - \lambda I)\mathbf{v} = 0$ for each eigenvalue.
For $\lambda = i$:

$$\begin{pmatrix} -i & -1 \\ 1 & -i \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0$$

Solving, we get the eigenvector $\mathbf{v} = \begin{pmatrix} 1 \\ i \end{pmatrix}$.

Complex Eigenvalues

Similarly, for $\lambda = -i$:

$$\begin{pmatrix} i & -1 \\ 1 & i \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = 0$$

Solving, we get the eigenvector $\mathbf{v} = \begin{pmatrix} 1 \\ -i \end{pmatrix}$.

we can see

$$B = \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix} \begin{pmatrix} i & 0 \\ 0 & -i \end{pmatrix} \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}^{-1}$$

Complex Eigenvalues

- We can see that the matrix B is have eigen values in complex field while it doesn't have any real eigen values.
- We will find ourselves extending the field to complex field to find eigen values of a matrix for such purposes.
- When extending to complex field our terms would transform to there complex forms:
  - Transpose $\iff$ Conjugate Transpose.
  - Dot product $\iff$ Inner product defined in complex field.
  - Symmetric matrices $\iff$ Hermitian matrices.
  - Orthogonal matrices $\iff$ Unitary matrices.

Left and Right Eigenvectors

Definition

Let $\mathbf{A}$ be a $d \times d$ matrix. A left eigenvector of $\mathbf{A}$ is a non-zero row vector $\mathbf{w}$ such that

$$\mathbf{wA} = \lambda \mathbf{w}$$

where λ is the eigenvalue of $\mathbf{w}$.

Definition

Let $\mathbf{A}$ be a $d \times d$ matrix. A right eigenvector of $\mathbf{A}$ is a non-zero column vector $\mathbf{v}$ such that

$$\mathbf{Av} = \lambda \mathbf{v}$$

where λ is the eigenvalue of $\mathbf{v}$.

Left and Right Eigenvectors

Observation

For a Symmetric matrix $\mathbf{A}$, the left eigenvector is transpose of some right eigenvector.

Proof:

Let $\mathbf{v}$ be a right eigenvector of $\mathbf{A}$ with eigenvalue λ. Then

$$\mathbf{A}\mathbf{v} = \lambda\mathbf{v}$$

Taking transpose on both sides, we get

$$\mathbf{v}^T\mathbf{A}^T = \mathbf{v}^T\mathbf{A} = \lambda\mathbf{v}^T$$

Hence $\mathbf{v}^T$ is a left eigenvector of $\mathbf{A}$ with eigenvalue λ.

Left and Right Eigenvectors

Observation

For a general matrix $\mathbf{A}$, left and right eigenvalues are same

Proof:
Characteristic polynomial of $\mathbf{A}$ is given by $f(\lambda) = \det(\mathbf{A} - \lambda\mathbf{I})$.
But determinant of a matrix is same as determinant of its transpose. Hence $f(\lambda) = \det(\mathbf{A}^T - \lambda\mathbf{I})$.
Since the characteristic polynomial is same for $\mathbf{A}$ and $\mathbf{A}^T$, the eigenvalues are same.

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Diagonalization

Definition

A matrix $\mathbf{A}$ is said to be diagonalizable if it can be expressed as

$$\mathbf{A} = \mathbf{V}\Delta\mathbf{V}^{-1}$$

where $\mathbf{V}$ is a matrix with columns as eigenvectors of $\mathbf{A}$ and Δ is a diagonal matrix with eigenvalues of $\mathbf{A}$ on the diagonal.

In $\mathbf{A} = \mathbf{V}\Delta\mathbf{V}^{-1}$, the columns of $\mathbf{V}$ are right eigenvectors of $\mathbf{A}$ and the rows of $\mathbf{V}^{-}1$ are left eigenvectors of $\mathbf{A}$.
We can use algorithm to find eigenvalues and eigenvectors to diagonalize a matrix. (As done in the previous slides)

Diagonalization

Observation: Eigenvectors belonging to distinct eigenvalues are linearly independent

Proof:
let $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ be eigenvectors of a matrix $\mathbf{A}$ belonging to distinct eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$.
Suppose there exists scalars $c_1, c_2, \ldots, c_k$ such that $\sum_{i=1}^{k} \alpha_i \mathbf{v_i} = 0$ such the not all α_i are zero. Then we can multiply by $(\mathbf{A} - \lambda_2 I)(\mathbf{A} - \lambda_3 I) \ldots (\mathbf{A} - \lambda_k I)$ to get

$$\alpha_i [\prod_{i=2}^{k} (\lambda_1 - \lambda_i)] \mathbf{v_1} = 0$$

Since eigenvalues are distinct $\prod_{i=2}^{k} (\lambda_1 - \lambda_i) \neq 0$ and hence $\alpha_1 = 0$. Similarly, we can show that all α_i are zero. Hence proof by contradiction.

Diagonalization

If roots of the characteristic polynomial of a matrix **A** are distinct, then we can find linearly independent basis of vector space. Hence we can diagonalize the matrix.

Let characteristic polynomial be $f(\lambda) = \prod_{i=1}^{k} (\lambda_i - \lambda)^{r_i}$

- if $r_i = 1$ for all i, then **A** is diagonalizable.
- if $r_i > 1$ for some i, then
  - If rank of eigenspace of λ_i is less than r_i, then **A** is not diagonalizable.
  - If rank of eigenspace of λ_i is equal to r_i $\forall i$, then **A** is diagonalizable.

Uniqueness of Diagonalization:

- If all eigenvalues of a matrix are distinct, then then we can induce a unique diagonalization by ordering the eigenvalues.
- If some eigenvalues are repeated but the matrix is diagonalizable, then the diagonalization cannot be unique due to repeated eigenvalues.

Triangulization

If rank of eigenspace of λ_i is less than r_i, then $\mathbf{A}$ is not diagonalizable. Where does rest of the eigenvectors go?

Characteristic Polynomial of $\mathbf{A}$ tells us that $(\mathbf{A} - \lambda_i \mathbf{I})^{r_i}\mathbf{v} = 0$ has r_i linearly independent solutions. But it does not tell us $(\mathbf{A} - \lambda_i \mathbf{I})$ has r_i linearly independent solutions.

Generalized Eigenvectors

A vector $\mathbf{v}$ is called a generalized eigenvector of rank m of $\mathbf{A}$ if

$$(\mathbf{A} - \lambda \mathbf{I})^m \mathbf{v} = 0$$

but

$$(\mathbf{A} - \lambda \mathbf{I})^{m-1} \mathbf{v} \neq 0$$

.

Triangulization

Jordan Chain

Def: Let $\mathbf{v_m}$ be a generalized eigenvector of rank m of matrix $\mathbf{A}$ for eigenvalue λ. Then the set of vectors $\{\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_m}\}$ given by

$$\mathbf{v_j} = (\mathbf{A} - \lambda \mathbf{I})^{m-j}\mathbf{v_m} = (\mathbf{A} - \lambda \mathbf{I})\mathbf{v_{j+1}} \text{ for } j \in \{1, 2, \ldots, m\}.$$

is called a Jordan Chain. Here $\mathbf{v_1}$ is the ordinary eigenvector of $\mathbf{A}$ for eigenvalue λ.

Assume $\mathbf{v}_1$ is a normal eigenvector of $\mathbf{A}$ for eigenvalue λ. Then

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{v}_1 = 0$$

Now assume $\mathbf{v}_2$ is a generalized eigenvector of rank 2 of $\mathbf{A}$ for eigenvalue λ. Then if we let $(\mathbf{A} - \lambda \mathbf{I})\mathbf{v}_2 = \mathbf{v}_1$, then

$$(\mathbf{A} - \lambda \mathbf{I})^2\mathbf{v}_2 = (\mathbf{A} - \lambda \mathbf{I})\mathbf{v}_1 = 0$$

. This kind of construction can be extended to higher ranks.

Triangulization

Jordan Normal Form

A matrix $\mathbf{A}$ is said to be in Jordan Normal Form if it is a block diagonal matrix with Jordan Blocks on the diagonal.

$$\mathbf{A} = \mathbf{V}\mathbf{U}\mathbf{V}^{-1}$$

where V is a matrix with columns as eigenvectors and generalized eigenvectors of $\mathbf{A}$ and U is a Upper Triangular matrix.

Triangulization

Jordan Normal Form

Suppose A has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$ with algebraic multiplicities $r_1, r_2, \ldots, r_k$ and geometric multiplicities $g_1, g_2, \ldots, g_k$ respectively. Then in Jordan Normal Form, the matrix U is of form:

$$U = \begin{pmatrix} J_1 & 0 & 0 & \ldots & 0 \\ 0 & J_2 & 0 & \ldots & 0 \\ 0 & 0 & J_3 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & J_k \end{pmatrix}$$

Triangulization

Jordan Normal Form

The Jordan Block J_i corresponding to eigenvalue λ_i is of form:

$$J_i = \begin{pmatrix} G & 1 & 0 & \dots & 0 \\ 0 & \lambda_i & 1 & \dots & 0 \\ 0 & 0 & \lambda_i & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_i \end{pmatrix}$$

where G is a $g_i \times g_i$ diagonal matrix with all elements as λ_i on diagonal.

Triangulization

Suppose $r_i = 3$ for some i. Then the Jordan Block J_i is of form:

Case 1:

$$J_i = \begin{pmatrix} \lambda_i & 0 & 0 \\ 0 & \lambda_i & 0 \\ 0 & 0 & \lambda_i \end{pmatrix}$$

Case 2:

$$J_i = \begin{pmatrix} \lambda_i & 0 & 0 \\ 0 & \lambda_i & 1 \\ 0 & 0 & \lambda_i \end{pmatrix}$$

Case 3:

$$J_i = \begin{pmatrix} \lambda_i & 1 & 0 \\ 0 & \lambda_i & 1 \\ 0 & 0 & \lambda_i \end{pmatrix}$$

- Case 1: 3 linearly independent eigenvectors
- Case 2: 2 linearly independent eigenvectors and 1 generalized eigenvector
- Case 3: 1 linearly independent eigenvector and 2 generalized eigenvectors

Schur Decomposition

Triangulization are not unique, for Jordan normal form the structure of **U** is special. An Alternative is Schur's Decomposition where U is a upper triangular matrix and we impose the condition that V is an orthogonal matrix.

Schur Decomposition

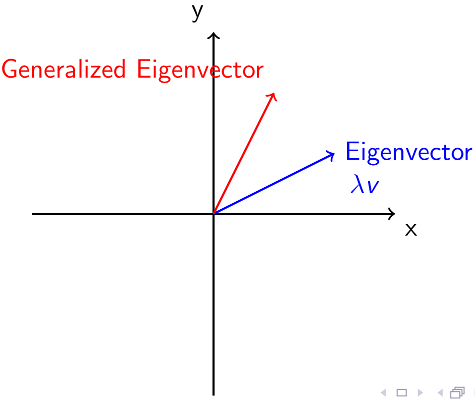
Let A be a $d \times d$ matrix. Then there exists a Orthogonal(Unitary in case of complex values) matrix P such that

$$A = PUP^H$$

where U is an upper triangular matrix.

Geometric Interpretation of Jordan Normal Form

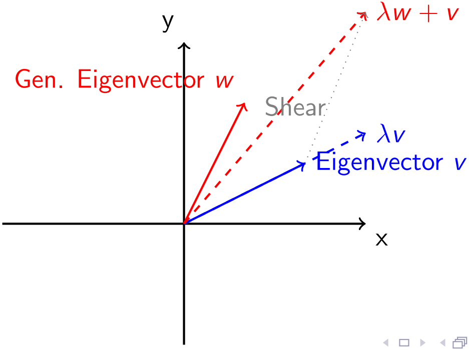
- The Jordan Normal Form (JNF) provides a nearly diagonal representation of a matrix, simplifying its action on vector spaces.
- Geometric insights:
  - Diagonalizable matrices scale vectors along independent eigenvector directions.
  - Non-diagonalizable matrices "shear" or couple subspaces through generalized eigenvectors.



Geometric Interpretation of Jordan Normal Form

Shearing Effect:

- A Jordan block with eigenvalue λ combines scaling and shifting effects.
- The eigenvector is stretched by λ, while the generalized eigenvector introduces a shear (sideways displacement).
- This is evident in the action of A on a vector space.



Geometric Interpretation of Triangulization

- Non Diagonalizable Matrices contains residual rotational components.
- For non-diagonalizable matrices, only scaling alone is not sufficient to describe the transformation.
- Hence we need to consider the residual rotational components.
- If we allow some rotation after scaling, then we can represent the transformation as a combination of scaling and rotation.
- This leads to **Polar Decomposition** of a matrix.

Similar Matrices Families

Definition

Two matrices A and B are said to be similar if there exists an invertible matrix V such that

$$B = VAV^{-1}$$

Interpretation:

- P represents a change of basis in the vector space $V = \mathbb{F}^n$ (generally $\mathbb{F} = \mathbb{R}$)
- Similarity relates A and B as different representations of the same linear transformation $T : V \to V$ with respect to two bases:

$$[T]_\beta = A, \quad [T]_\gamma = B,$$

  where β, γ are bases of V and P maps β to γ.

Properties of Similar Matrices

Lemma: If $A, B \in \mathbb{F}^{n \times n}$ are similar, then:

1. A and B have the same eigenvalues (including algebraic multiplicities).
2. A and B have the same characteristic polynomial:
$$\det(\lambda I - A) = \det(\lambda I - B).$$
3. The rank of A equals the rank of B.
4. A and B share the same Jordan canonical form.
5. The trace of A equals the trace of B.

Proof Outline:

- For any $x \in \mathbb{F}^n$, $Ax = \lambda x$ implies:
$$P^{-1}AP(P^{-1}x) = \lambda(P^{-1}x),$$
so eigenvalues are preserved.
- The determinant is invariant under similarity:
$$\det(\lambda I - B) = \det(P^{-1}(\lambda I - A)P) = \det(\lambda I - A).$$
- Rank and Jordan form invariance follow from the same basis transformation argument.
- The trace is preserved since it is the sum of the eigenvalues.

detour: Trace of a Matrix

Definition

The trace of a square matrix A is the sum of its diagonal elements:

$$\operatorname{tr}(A) = \sum_{i=1}^{n} A_{ii}.$$

Properties

The trace of a matrix has the following properties:

- $\operatorname{tr}(AB) = \operatorname{tr}(BA)$ provided the product AB & BA are defined.
- The trace is a linear operator:

$$\operatorname{tr}(A+B) = \operatorname{tr}(A) + \operatorname{tr}(B), \quad \operatorname{tr}(cA) = c\operatorname{tr}(A).$$

Trace and Eigenvalues

Lemma

The trace of a matrix is equal to the sum of its eigenvalues.

Proof:

- Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of A.
- The characteristic polynomial of A is:

$$f(\lambda) = \prod_{i=1}^{n}(\lambda - \lambda_i) = det(A - \lambda I).$$

- Expanding the characteristic polynomial, we find:

$$f(\lambda) = \lambda^n - (\lambda_1 + \lambda_2 + \ldots + \lambda_n)\lambda^{n-1} + \ldots + (-1)^n \lambda_1 \lambda_2 \ldots \lambda_n$$

$$= \lambda^n - \mathrm{tr}(A)\lambda^{n-1} + \ldots + (-1)^n \mathrm{det}(A).$$

- Comparing coefficients, we see that the trace of A is the sum of its eigenvalues.

Geometric Interpretation of Trace

Geometric interpretation of the trace of a matrix is not straightforward specially when the matrix is not symmetric. But we have an Observation useful in Machine Learning.

Observation

The trace of the Gram Matrix $A^T A$ is equal to energy of base matrix A.

Definition

The energy of a matrix A is defined as the sum of squares of all the elements of the matrix.

$$\text{Energy}(A) = \sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2 = ||A||_F^2 = tr(A^T A)$$

Geometric Interpretation of Trace

A more Advance interpretation of trace is as divergence of a vector field described by the matrix.
Define a vector field $\mathbf{v}(\mathbf{x}) = A\mathbf{x}$ where $\mathbf{x}$ is a point in the vector space.
As we know the divergence of a vector field is defined as

$$\nabla \cdot \mathbf{v} = \sum_{i=1}^{n} \frac{\partial v_i}{\partial x_i}$$

For the vector field $\mathbf{v}(\mathbf{x}) = A\mathbf{x}$, the divergence is given by

$$\nabla \cdot \mathbf{v} = \sum_{i=1}^{n} \frac{\partial v_i}{\partial x_i} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial A_{ij} x_j}{\partial x_i} = \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} = \mathrm{tr}(A)$$

Simultaneous Diagonalizable Matrices

Definition

Two matrices A and B are said to be simultaneously diagonalizable if there exists an invertible $d \times d$ matrix V such that both A and B are diagonalized by V ie columns of V are eigenvectors of both A and B.

$$V^{-1}AV = \Delta_1, \quad V^{-1}BV = \Delta_2,$$

where D_A and D_B are diagonal matrices.

Geometrically, simultaneously diagonalizable matrices perform anisotropic scaling along the same set of directions. The scaling factors may differ, but the directions are preserved.

Lemma

Two matrices A and B are simultaneously diagonalizable if and only if they commute:

$$AB = BA.$$

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Spectral Theorem

Spectral Theorem

Let A be a $d \times d$ symmetric matrix with real entries. Then A is always diagonalizable and has d real eigenvalues and d orthonormal eigenvectors.

$$A = V \Delta V^T$$

where V is an orthogonal matrix with columns as eigenvectors of A and Δ is a diagonal matrix with eigenvalues of A.

A-orthogonality

Definition

Two vectors $\mathbf{v}$ and $\mathbf{w}$ are said to be A-orthogonal if

$$\mathbf{v}^T A\mathbf{w} = 0.$$

- Generalization of orthogonality
- We can naturally generalize Gram-Schmidt orthogonalization to A-orthogonalization to find A-orthonormal basis.
- This holds applications like conjugate gradient descent where we are looking for A-orthogonal direction.

Lemma

If A is a symmetric matrix, then eigenvectors corresponding to distinct eigenvalues are A-orthogonal.
Proof:

$$\mathbf{v}_i^T A\mathbf{v}_j = \lambda_j \mathbf{v}_i^T \mathbf{v}_j = \lambda_j \delta_{ij} = 0$$

Positive Semidefinite Matrices

Definition

A symmetric matrix A is said to be positive semidefinite if for all non-zero vectors $\mathbf{x}$,

$$\mathbf{x}^T A\mathbf{x} \geq 0.$$

Alternative Definition

A symmetric matrix A is said to be positive semidefinite if all its eigenvalues are non-negative.

By the Spectral Theorem, a symmetric matrix A can always be diagonalized as $A = V\Delta V^T$. Let $\lambda_1, \lambda_2, \ldots, \lambda_d$ be the eigenvalues of A. Then for any vector $\mathbf{v}$, let $\mathbf{y} = V^T\mathbf{v}$

$$\mathbf{v}^T A\mathbf{v} = \mathbf{y}^T \Delta \mathbf{y} = \sum_{i=1}^{d} \lambda_i y_i^2$$

for $\sum_{i=1}^{d} \lambda_i y_i^2 \geq 0$ for all $\mathbf{v}$ if and only if all $\lambda_i \geq 0$.

Properties of Positive Semidefinite Matrices

Lemma

The matrix A is positive semidefinite if and only if it can be expressed as $A = B^T B$ for some matrix B.

Proof:

- Let A be positive semidefinite. Then by the Spectral Theorem, $A = V\Delta V^T$. Let $B = V\Delta^{1/2}$. Then

$$A = V\Delta V^T = V\Delta^{1/2}\Delta^{1/2}V^T = B^T B.$$

- Conversely, if $A = B^T B$, then for any vector $\mathbf{x}$,

$$\mathbf{x}^T A\mathbf{x} = \mathbf{x}^T B^T B\mathbf{x} = (B\mathbf{x})^T B\mathbf{x} = ||B\mathbf{x}||^2 \geq 0.$$

Cholesky Factoriaztion

Definition

A $d \times d$ matrix A is said to be positive definite if for all non-zero vectors $\mathbf{v}$,

$$\mathbf{v}^T A\mathbf{v} > 0.$$

For a positive definite matrix use of eigendecomposition is natural choice but not the only one. Given $A = BB^T$ we can use any orthogonal $d \times d$ matrix P to get $A = B(PP^T)B^T = (BP)(BP)^T$.
One of these factorization is Cholesky Factorization where $A = LL^T$ where L is a lower triangular matrix.

Cholesky Factorization

Let A be a positive semidefinite matrix. Then there exists a lower triangular matrix L such that

$$A = LL^T.$$

This is special case of LU decomposition where $U = L^T$. But Cholesky decomposition is more efficient than Generic LU decomposition.

Algorithm for Cholesky Factorization

Let matrix L be $L = [l_{ij}]_{d\times d}$ and A be $A = [a_{ij}]_{d\times d}$. We aim to set up the following equations:

$$a_{ij} = \sum_{k=1}^{d} l_{ik} l_{jk} = \sum_{k=1}^{j} l_{ik} l_{jk} \text{ for } i \geq j$$

and $a_{ij} = a_{ji}$ as A is symmetric.
Performing these computations in order $i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, i$ we can get the Cholesky factorization.

$$l_{11} = \sqrt{a_{11}}$$

$$l_{i1} = \frac{a_{i1}}{l_{11}} \text{ for } i = 2, 3, \ldots, d$$

now for $j = 2, 3, \ldots, d$

$$l_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2} \text{ for } j = 2, 3, \ldots, d$$

Algorithm for Cholesky Factorization

Algorithm 1 Cholesky Factorization

```
Initialize $L$ as a zero matrix of size $d \times d$.
for $j = 1$ to $d$ do
    Compute the diagonal elements:
        $L_[j,j] = \sqrt{A[j,j] - \sum_{k=1}^{j-1} L[j,k]^2}$
    for $i = j + 1$ to $d$ do
        Compute the off-diagonal elements:
            $L[i,j] = \frac{1}{L[j,j]} \left( A[i,j] - \sum_{k=1}^{j-1} L[i,k] L[j,k] \right)$
    end for
end for
return $L$
```

Properties of Cholesky Factorization

- Time Complexity: $O(d^3)$
- Algorithm works for positive definite matrices only. In case of singular matrices, atleast one $l_{jj} = 0$ hence division by zero.
- Cholesky Decomposition is used to test for positive definiteness of a matrix.

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Fast Matrix Powers

- If we wish to compute A^k for some positive integer k, then the naive approach of multiplying A with itself k times is inefficient.
- Also, computation of A^∞ is not possible in general.
- A better approach is to use the following algorithm:

  $$A^k = V \Delta^k V^{-1}$$

  where $A = V \Delta V^{-1}$ is the eigendecomposition of A.
- It is often easy to compute Δ^k as we only need to raise the diagonal elements to the power k.
- Also finding limiting behavior of A^k as $k \to \infty$ is easy as we only need to find the limiting behavior of Δ^k which depends on if the entry is less than, equal to or greater than 1.
- This is useful in many applications like Adjacency Matrix of Graphs, etc.

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Some Examples of Diagonalizable Matrices in Machine Learning

Several PSD matrices arise in machine learning and optimization:

- **Gram Matrix:** The Gram matrix of a set of vectors is symmetric and positive semidefinite.
- **Similarity Matrix:** The similarity matrix in clustering is symmetric and positive semidefinite.
- **Covariance Matrix:** The covariance matrix of a dataset is always symmetric and positive semidefinite.
- **Hessian Matrix:** The Hessian matrix of a twice-differentiable function is generally symmetric and positive semidefinite at a local minimum.
- **Kernel Matrix:** The kernel matrix in kernel methods is symmetric and positive semidefinite.
- etc.

Dot Product Similarity Matrix

Definition

Let D be an $n \times d$ matrix representing n data points in d-dimensional space. Then Similarity Matrix S is a $n \times n$ matrix between data points, where the element S_{ij} is the similarity function evaluated on i^{th} and j^{th} data points.

$$S_{ij} = \langle D_i, D_j \rangle$$

Definition

Dot Product Similarity Matrix is a similarity matrix where the similarity function is the dot product of the data points.

$$S = DD^T$$

- Dot Product Similarity Matrix is always symmetric and positive semidefinite.
- It is used in many machine learning algorithms like PCA, Kernel Methods, etc.

Dot Product Similarity Matrix

We can use S matrix as an alternate representation of the data points. As D is recoverable from S upto rotation and reflection.
Factoriaztion of $S = D'D'^T$ can yield a diffrent D' which can be viewed as rotatated or reflected version of D which is usually not a concern in many applications.
Most common way of recovering D from S is to use the eigendecomposition of S.

$$S = Q\Delta Q^T$$

$$S = Q\Sigma^2 Q^T = \underbrace{(Q\Sigma)}_{D'}\underbrace{(\Sigma Q)^T}_{D'^T}$$

Here $D' = Q\Sigma$ is a $n \times n$ data set.
It may seem odd that D' is a $n \times n$ matrix but it since D had d dimesions so maximum rank of S is d.
So atleast $n - d$ eigenvalues of S are zero, hence the corresponding coordinates of D' are zero.

General Similarity Matrix

- In kernel methods, instead of the dot product, we use the Gaussian kernel to compute the similarity matrix:

  $$\text{Similarity}(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{||\mathbf{x} - \mathbf{y}||^2}{\sigma^2}\right)$$

  where σ is a hyperparameter.
- In such situations, the recovered dataset may not have dummy variables, and all n dimensions can be useful.
- Such methods are used to find non-linear relationships in the data and are used in many applications like SVM, Kernel PCA, etc.
- These method is also used when data is only available in the form of similarities, we can obtain a representation of the data in a higher-dimensional space.
  Example: In a set of n graph or time series objects, we can compute the similarity between each pair of data points and use the similarity matrix to find the representation of the data in a higher-dimensional space.

Applications of Similarity Matrix

- **Principal Component Analysis (PCA):** The eigendecomposition of the dot product similarity matrix is used to find the principal components.
- **Kernel Methods:** The kernel matrix in kernel methods is a dot product similarity matrix in a high-dimensional feature space.
- **Spectral Clustering:** The eigendecomposition of the similarity matrix is used to find the clusters in the data.
- **Graph Laplacian:** The graph Laplacian matrix is a dot product similarity matrix in the graph domain.
- **Kernel PCA:** Kernel PCA uses the eigendecomposition of the kernel matrix to find the principal components.
- **etc.**

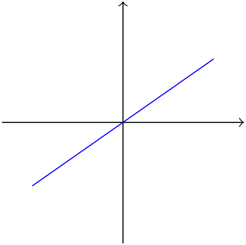
Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

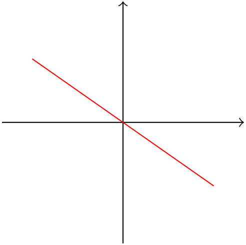
Covariance

Key concepts:

- Variance: $\sigma_x^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)^2$
- Covariance: $\sigma_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y)$



Positive



Negative

Covariance Matrix

For mean-centered data matrix D $(n \times d)$:

$$C = \frac{D^T D}{n}$$

Properties:

- Symmetric: $C_{ij} = C_{ji}$
- Diagonal elements are variances
- Off-diagonal elements are covariances
- Always positive semidefinite

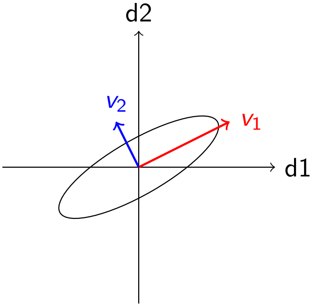
Example 2D covariance matrix:

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$

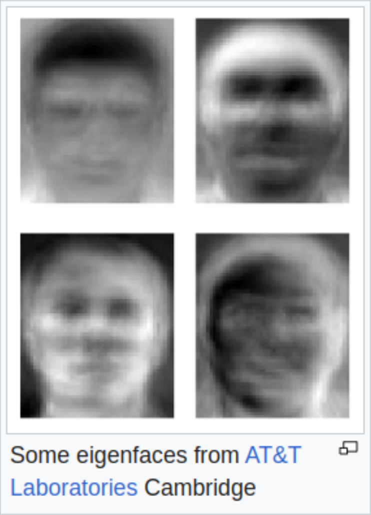
Principal Component Analysis (PCA)

Steps:

1. Compute covariance matrix C
2. Find eigenvectors and eigenvalues
3. Sort eigenvalues in descending order
4. Select top k eigenvectors
5. Project data onto new basis



Eigenfaces



Some eigenfaces from AT&T Laboratories Cambridge

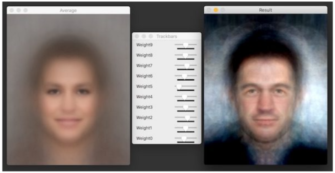


Figure 2 : On the left is the mean image. On the right is a new face produced by adding 10 Eigenfaces with different weights (shown in center).

PCA Applications

- Dimensionality reduction
  - Image compression
  - Feature selection
  - Visualization of high-dimensional data
- Data preprocessing
  - Noise reduction
  - Feature decorrelation
  - Standardization
- Applications
  - Face recognition (eigenfaces)
  - Gene expression analysis
  - Financial data analysis

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Convex and Concave Functions

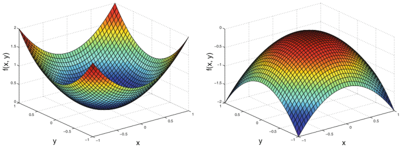
Definition

A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be convex if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\lambda \in [0, 1]$,

$$f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}).$$

A function $f : \mathbb{R}^d \to \mathbb{R}$ is said to be concave if for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\lambda \in [0, 1]$,

$$f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y}) \geq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}).$$



(a) Convex function $f(x, y) = x^2 + y^2$

(b) Concave function $f(x, y) = -(x^2 + y^2)$

Figure 3.4: Illustration of convex and concave functions

Quadratic programming

Definition

A quadratic program is an optimization problem of the form:

$$\text{Minimize } \frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \text{ subject to } A\mathbf{x} \leq \mathbf{b}$$

where Q is a symmetric positive semidefinite matrix, $\mathbf{c}$ is a vector, and A and $\mathbf{b}$ are matrices.

Quadratic programming is a type of nonlinear programming. In some special cases, this problems becomes easy. If Q is positive definite, then the problem becomes a least squares problem.

$$\text{Minimize } \frac{1}{2}||\mathbf{Rx} - \mathbf{d}||^2$$

where $Q = \mathbf{R}^T\mathbf{R}$ from Cholesky factorization and $\mathbf{c} = -\mathbf{R}^T\mathbf{d}$.
Otherwise there are other methods:

- Largrange Multipliers
- Congugate Gradient Descent

Simple Quadratic Optimization

Problem

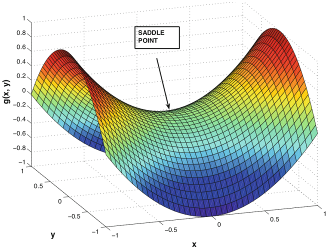
A Simple form of Quadratic Optimization is:

$$\text{Optimize } f(\mathbf{v}) = \mathbf{v}^T A\mathbf{v} + \mathbf{b}^T\mathbf{v} + c$$

where A is a symmetric matrix, $\mathbf{b}$ is a vector, and c is a scalar.

Observation

The function $f(\mathbf{v}) = \mathbf{v}^T\mathbf{A}\mathbf{v}$ is convex if and only if A is positive semidefinite.



Simple Quadratic Optimization

So Now If A is positive semidefinite, then the function $f(\mathbf{v}) = \mathbf{v}^T A\mathbf{v} + \mathbf{b}^T\mathbf{v} + c$ is convex.

Let $A = V\Delta V^T$ be the eigendecomposition of A. Then we can write:

$$f(\mathbf{v}) = \mathbf{v}^T V\Delta V^T\mathbf{v} + \mathbf{b}^T\mathbf{v} + c$$

We can now transform the variable $\mathbf{v}$ to $\mathbf{u} = V^T\mathbf{v}$. So the function becomes:

$$f(\mathbf{u}) = \mathbf{u}^T\Delta\mathbf{u} + \mathbf{b}^T V\mathbf{u} + c$$

Now we can solve this problem by solving the following problem:

$$\text{Minimize } \sum_{i=1}^{d} \lambda_i u_i^2 + \sum_{i=1}^{d} b_i' u_i + c$$

where λ_i are the eigenvalues of A.
These are d independent 1 variable optimization problems, which can be solved by setting the derivative to zero.

Simple Quadratic Optimization

- We have found the explicit solution to the problem by transforming the variable to the eigenvector basis.
- But this is not the most efficient way to solve the problem.
- If we can work with approximate solutions, then we can use iterative methods like Congugate gradient descent. Which are more efficient than eigen decomposition.

This idea is utilized to solve General Optimization problems also:

- If we can approximate the function as a quadratic function, using second order taylor expansion, around x_t
- Now we solve the quadratic optimization problem to get the next point x_{t+1}

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Norm-Constrained Optimization

Problem:

$$\text{Optimize } \mathbf{x}^T A \mathbf{x}$$
$$\text{subject to } \|\mathbf{x}\|^2 = 1$$

Let $\mathbf{v}_1, \ldots, \mathbf{v}_d$ be orthonormal eigenvectors of A:

$$\mathbf{x} = \sum_{i=1}^{d} \alpha_i \mathbf{v}_i$$

Mathematical Solution

Rewrite optimization problem:

$$\text{Optimize } \sum_{i=1}^{d} \lambda_i \alpha_i^2$$
$$\text{subject to } \sum_{i=1}^{d} \alpha_i^2 = 1$$

Solution:

- Maximum: Set $\alpha_i = 1$ for largest λ_i
- Minimum: Set $\alpha_i = 1$ for smallest λ_i
- All other $\alpha_i = 0$

k-Dimensional Extension

Problem:

$$\begin{aligned} \text{Optimize } & \sum_{i=1}^{k} \mathbf{x}_i^T A \mathbf{x}_i \\ \text{subject to } & \|\mathbf{x}_i\|_2 = 1 \text{ for } i = 1, \ldots, k \\ & \mathbf{x}_i^T \mathbf{x}_j = 0 \text{ for } i \neq j \end{aligned}$$

Solution:

- Maximum: k largest eigenvectors
- Minimum: k smallest eigenvectors

This results seems intuitive, but the proof is non-trivial and uses lagrangian relaxation and will be covered in future sessions.

Geometric Interpretation

Matrix A causes:

- Anisotropic scaling of space
- Scale factors = eigenvalues
- Scaling directions = eigenvectors

Objective function:

- Maximizes/minimizes aggregate projections
- Dot products between $\mathbf{x}_i$ and $A\mathbf{x}_i$
- Largest/smallest scaling directions give extrema

Outline

1. Preliminaries
   - Linear Algebra
   - Machine Learning
2. Diagonalizable Transformations and Eigenvectors
   - Determinants
   - Eigenvectors and Eigenvalues
   - Diagonalization
   - Spectral Theorem
3. Machine Learning and Optimization
   - Diagonalizable Matrices in Machine Learning
   - PCA
   - Quadratic Optimization
   - Norm Constrained Optimization
4. Numerical Methods for finding Eigenvalues

Overview of Eigenvalue Computation

- Eigenvalues of a matrix A are the roots of the characteristic polynomial:

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

- Direct methods:
  - Find roots of the characteristic polynomial
  - Ex: Newton's method
  - Sometimes numerically unstable and inefficient
  - Finding Root's of polynomial is a numerically harder problem then finding eigenvalues.
- Iterative methods:
  - Power method
  - Jacobi method
  - QR algorithm
- A way to find roots of large polynomial is to construct a companion matrix and find its eigenvalues.

Companion Matrix

Definition

Given a polynomial $p(\lambda) = a_0 + a_1\lambda + \ldots + a_n\lambda^n$, the companion matrix is:

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & -a_{n-1} \end{bmatrix}$$

- Eigenvalues of C are the roots of $p(\lambda)$
- Useful for finding roots of large polynomials
- Some books define the companion matrix as the transpose of the above matrix

QR Method of Eigenvalue Computation

Method:

- QR decomposition: $A_m = Q_m \cdot R_m$
- Start with $A_0 = A$
- Iteratively compute $A_{m+1} = Q_m^T \cdot A_m \cdot Q_m$
- After enough iterations, A_m converges to an upper triangular matrix

The idea of this method goes to Schur decomposition of a matrix. That is:

$$A = Q \cdot T \cdot Q^T$$

where Q is orthogonal and T is upper triangular.

- Eigenvalues are the diagonal elements
- Works for non-symmetric matrices
- QR decomposition is not unique, but the eigenvalues are
- One can use Hessenberg form of matrix to reduce the number of iterations.

Power Method for finding Eigenvector

- Computes the **largest eigenvalue** and its corresponding eigenvector of a matrix A.
- Steps:
  1. Start with an initial vector $x^{(0)}$ (random or approximate).
  2. Iteratively compute:
  $$x^{(t+1)} = \frac{Ax^{(t)}}{\|Ax^{(t)}\|}.$$
  3. As $t \to \infty$, $x^{(t)}$ converges to the eigenvector corresponding to the largest eigenvalue.
- The Rayleigh quotient: $\mu(x) = \frac{x^T Ax}{x^T x}$ will converge to eigenvalue as $t \to \infty$.
- Convergence relies on $\lambda_1 > \lambda_2$ (distinct eigenvalues).

Power Method for finding Eigenvector

Proof of Convergence:
Let $x^{(0)} = \sum_{i=1}^{d} \alpha_i v_i$ where v_i are eigenvectors of $\mathbf{A}$ and let $\mathbf{A} = \mathbf{VJV}^{-1}$ be its jordan canonical form.
Then we have:

$$
\begin{aligned}
x^{(t)} = \frac{\mathbf{A}x^{t-1}}{\|\mathbf{A}x^{t-1}\|} = \frac{\mathbf{A}^t x^{(0)}}{\|\mathbf{A}^t x^{(0)}\|} &= \frac{\mathbf{VJ}^t\mathbf{V}^{-1}x^{(0)}}{\|\mathbf{VJ}^t\mathbf{V}^{-1}x^{(0)}\|} \\
&= \frac{\mathbf{VJ}^t\mathbf{V}^{-1}\sum_{i=1}^{d}\alpha_i v_i}{\|\mathbf{VJ}^t\mathbf{V}^{-1}\sum_{i=1}^{d}\alpha_i v_i\|} \\
&= \frac{\mathbf{VJ}^t\sum_{i=1}^{d}c_i e_i}{\|\mathbf{VJ}^t\sum_{i=1}^{d}c_i e_i\|} \\
&= \left(\frac{\lambda_1}{\lambda_1}\right)^k \frac{c_1}{c_1} \frac{v_1 + \frac{1}{c_1}\mathbf{V}(\frac{1}{\lambda_1}\mathbf{J})^k(c_2 e_2 + \cdots + c_n e_n)}{\|v_1 + \frac{1}{c_1}\mathbf{V}(\frac{1}{\lambda_1}\mathbf{J})^k(c_2 e_2 + \cdots + c_n e_n)\|}
\end{aligned}
$$

Power Method for finding Eigenvector

$$(\frac{1}{\lambda_1}\mathbf{J})^k = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & (\frac{1}{\lambda_1}\mathbf{J}_2)^k & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & (\frac{1}{\lambda_1}\mathbf{J}_m)^k \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{bmatrix} \text{ as } k \rightarrow \infty$$

Hence

$$x^{(k)} \rightarrow \frac{c_1 v_1}{\|c_1 v_1\|}(\frac{\lambda_1}{\|\lambda_1\|})^k + r_k \text{ as } k \rightarrow \infty, ||r_k|| \rightarrow 0$$

Generalization to Find Top-k Eigenvectors

We can extend the power method to find the top-k eigenvectors of a matrix A:

- For symmetric matrix A we use **Deflation Technique**:
  1. Compute the largest eigenvalue λ_1 and eigenvector v_1.
  2. Deflate the matrix:
  $$A' = A - \lambda_1 v_1 v_1^T.$$
  3. Repeat the power method on A' to find the next largest eigenvalue and eigenvector.
- Repeat until the top-k eigenvectors are found.
- Assumes A is symmetric and eigenvalues are distinct.

Generalization to Find Top-k Eigenvectors

Matrix Product as Outer Product:

- We can Write a matrix product as an outer product of its columns:

$$C = AB = \sum_{i=1}^{d} \mathbf{a_i}\mathbf{b_i}^T$$

Hence if we have a matrix $A = V\Delta V^T$ then we can write:

$$A = \sum_{i=1}^{d} \lambda_i \mathbf{v_i}\mathbf{v_i}^T$$

- Hence if we find the largest eigenvalue and eigenvector, we can subtract the outer product of the eigenvector from the matrix to get a new matrix with the largest eigenvalue removed.
- This process can be repeated to find the next largest eigenvalue and eigenvector.
- We can generalize to left and right eigenvectors for Non-symmetric matrices.

Generalization to Non-Symmetric Matrices

- For non-symmetric matrices, compute both:
  - **Right eigenvectors**: $Av_i = \lambda_i v_i$.
  - **Left eigenvectors**: $w_i^H A = \lambda_i w_i^H$.
- Steps:
  1. Apply the power method to A to compute right eigenvectors.
  2. Apply the power method to A^H (conjugate transpose) to compute left eigenvectors.
  3. Use deflation to compute subsequent eigenvalues and eigenvectors:

$$A' = A - \lambda_1 v_1 w_1^H.$$

- Works for matrices with distinct eigenvalues.

Shift Method for Faster Convergence

- Improves convergence by shifting the eigenvalue spectrum.
- Define $B = A - \sigma I$ where σ is a shift parameter.
- Steps:
  1. Apply the power method to B instead of A.
  2. Largest eigenvalue of B corresponds to $\lambda_i - \sigma$ in A.
  3. Add σ back to recover the original eigenvalue.
- Choose σ close to the eigenvalue of interest for faster convergence.

Thank You!

Thank You!

Questions?