Math 405 : Learning From Data

Week 3 : Eigendecomposition

Dr. Maher Boudabra

Table of Contents

Lecture 1: Fundamentals of Eigenvalues and Eigenvectors

Lecture 2: Eigendecomposition and Diagonalization

Lecture 3: Applications in Data Science

Lecture 1 Outline

Lecture 1: Fundamentals of Eigenvalues and Eigenvectors

Lecture 2: Eigendecomposition and Diagonalization

Lecture 3: Applications in Data Science

What are Eigenvalues and Eigenvectors?

Definition

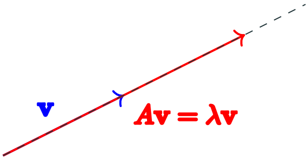
For a square matrix $A \in \mathbb{R}^{n \times n}$, a nonzero vector $\mathbf{v}$ is an **eigenvector** if:

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

where λ is a scalar called the **eigenvalue** corresponding to $\mathbf{v}$.

- Eigenvectors remain in the same direction after transformation by A
- They are only scaled by the eigenvalue λ
- Eigenvalues can be real or complex numbers

Geometric Interpretation



- Eigenvectors point in directions that are invariant under the transformation
- The matrix A only stretches or compresses eigenvectors
- The amount of stretching/compression is given by the eigenvalue

Finding Eigenvalues: The Characteristic Equation

To find eigenvalues, we solve:

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

This is called the **characteristic equation** of matrix A.

Example

For $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$:

$$\det\left(\begin{bmatrix} 2-\lambda & 1 \\ 1 & 2-\lambda \end{bmatrix}\right) = (2-\lambda)^2 - 1 = \lambda^2 - 4\lambda + 3 = 0$$

Eigenvalues: $\lambda_1 = 3$, $\lambda_2 = 1$

Finding Eigenvectors

For each eigenvalue λ_i, solve:

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

Example

Continuing with $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$:

For $\lambda_1 = 3$:

$$(A - 3I)\mathbf{v} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \end{bmatrix}\mathbf{v} = \mathbf{0} \Rightarrow \mathbf{v}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

For $\lambda_2 = 1$:

$$(A - I)\mathbf{v} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}\mathbf{v} = \mathbf{0} \Rightarrow \mathbf{v}_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Properties of Eigenvalues and Eigenvectors

- The sum of eigenvalues equals the trace of the matrix: $\sum \lambda_i = \text{tr}(A)$
- The product of eigenvalues equals the determinant: $\prod \lambda_i = \det(A)$
- Eigenvectors corresponding to distinct eigenvalues are linearly independent
- If A is symmetric ($A = A^\top$), then:
  - All eigenvalues are real
  - Eigenvectors corresponding to distinct eigenvalues are orthogonal

Eigenspace and Multiplicity

Definitions

- The **eigenspace** of λ is the set of all eigenvectors for that eigenvalue plus the zero vector
- **Geometric multiplicity**: dimension of the eigenspace
- **Algebraic multiplicity**: how many times λ appears as a root of the characteristic equation

Theorem

For any eigenvalue, geometric multiplicity $\leq$ algebraic multiplicity.

Example

$A = \begin{bmatrix} 2 & 1 \\ 0 & 2 \end{bmatrix}$ has eigenvalue $\lambda = 2$ with:

- Algebraic multiplicity: 2
- Geometric multiplicity: 1 (only one eigenvector $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$)

Table of Contents

Lecture 1: Fundamentals of Eigenvalues and Eigenvectors

Lecture 2: Eigendecomposition and Diagonalization

Lecture 3: Applications in Data Science

Lecture 2 Outline

Lecture 1: Fundamentals of Eigenvalues and Eigenvectors

Lecture 2: Eigendecomposition and Diagonalization

Lecture 3: Applications in Data Science

Diagonalization of Matrices

Theorem

*An $n \times n$ matrix A is **diagonalizable** if it has n linearly independent eigenvectors.*

If A has eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ with eigenvalues $\lambda_1, \ldots, \lambda_n$, then:

$$A = PDP^{-1}$$

where:

- $P = \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_n \end{bmatrix}$ (matrix of eigenvectors)
- $D = \text{diag}(\lambda_1, \ldots, \lambda_n)$ (diagonal matrix of eigenvalues)

Why Diagonalization Matters

Diagonalization simplifies many matrix operations:

- Matrix powers: $A^k = PD^kP^{-1}$ (much easier than computing A^k directly)
- Matrix exponentials: $e^A = Pe^DP^{-1}$
- Solving systems of differential equations
- Understanding long-term behavior of dynamical systems

Example

If $A = PDP^{-1}$, then:

$$A^{100} = PD^{100}P^{-1} = P \begin{bmatrix} \lambda_1^{100} & & \\ & \ddots & \\ & & \lambda_n^{100} \end{bmatrix} P^{-1}$$

Spectral Theorem

Theorem (Spectral Theorem for Real Symmetric Matrices)

If A is a real symmetric matrix ($A = A^\top$), then:

1. *All eigenvalues are real*
2. *Eigenvectors corresponding to distinct eigenvalues are orthogonal*
3. *A is orthogonally diagonalizable: $A = QDQ^\top$*

where Q is an orthogonal matrix ($Q^\top Q = I$) containing the eigenvectors.

This is a fundamental result with profound implications for data analysis and machine learning.

Eigendecomposition of Symmetric Matrices

For symmetric matrices, the eigendecomposition becomes:

$$A = Q\Lambda Q^\top = \sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^\top$$

where:

- $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$
- $Q = \begin{bmatrix} \mathbf{q}_1 & \cdots & \mathbf{q}_n \end{bmatrix}$ is orthogonal
- $\mathbf{q}_i \mathbf{q}_i^\top$ are rank-1 matrices (outer products)

This is called the **spectral decomposition**.

Positive Definite Matrices

Definition

A symmetric matrix A is **positive definite** if:

$$\mathbf{x}^\top A\mathbf{x} > 0 \quad \text{for all } \mathbf{x} \neq \mathbf{0}$$

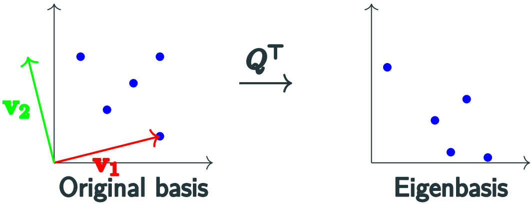
Theorem

A symmetric matrix is positive definite if and only if all its eigenvalues are positive.

Importance in Data Science

- Covariance matrices are positive semi-definite
- Kernel matrices in machine learning are positive definite
- Positive definiteness ensures optimization problems are convex

Visualizing Eigendecomposition



Eigendecomposition finds the natural coordinate system for the data.

Table of Contents

Lecture 1: Fundamentals of Eigenvalues and Eigenvectors

Lecture 2: Eigendecomposition and Diagonalization

Lecture 3: Applications in Data Science

Lecture 3 Outline

Lecture 1: Fundamentals of Eigenvalues and Eigenvectors

Lecture 2: Eigendecomposition and Diagonalization

Lecture 3: Applications in Data Science

Principal Component Analysis (PCA)

PCA is arguably the most important application of eigendecomposition in data science.

The PCA Problem

Given data points $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$, find:

1. Directions of maximum variance in the data
2. Lower-dimensional representation that preserves most information

Theorem

The principal components are the eigenvectors of the covariance matrix:

$$\Sigma = \frac{1}{m} \sum_{i=1}^{m} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top$$

ordered by decreasing eigenvalues.

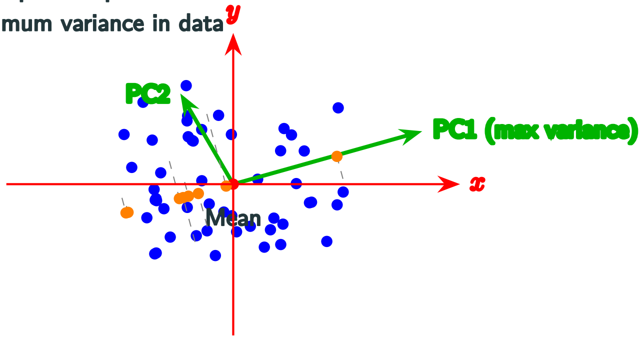
PCA Step by Step

1. Center the data: $\mathbf{x}_i \leftarrow \mathbf{x}_i - \boldsymbol{\mu}$
2. Compute covariance matrix: $\Sigma = \frac{1}{m} X X^\top$
3. Compute eigendecomposition: $\Sigma = Q \Lambda Q^\top$
4. Sort eigenvectors by decreasing eigenvalues
5. Project data onto top k eigenvectors: $Y = Q_k^\top X$

PCA Step by Step

PCA Visualization

- Blue dots: Original data points
- Orange dots: Projections onto PC1
- Green arrows: Principal Components
- PC1 captures maximum variance in data



Dimensionality Reduction with PCA

Variance Explained

The proportion of variance explained by the first k principal components is:

$$\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i}$$

Example

If eigenvalues are: $\lambda_1 = 8$, $\lambda_2 = 1.5$, $\lambda_3 = 0.5$

- First PC explains: $8/(8 + 1.5 + 0.5) = 80\%$ of variance
- First two PCs explain: $(8 + 1.5)/10 = 95\%$ of variance

We might keep only 2 dimensions while preserving 95% of the information!

Google's PageRank Algorithm

PageRank uses eigendecomposition to rank web pages by importance.

The PageRank Idea

- Model the web as a graph with pages as nodes and links as edges
- Create a transition matrix P where $P_{ij} = 1/\text{outdegree}(j)$ if page j links to page i
  item The importance vector $\mathbf{v}$ satisfies: $P\mathbf{v} = \mathbf{v}$
- This is an eigenvector problem with eigenvalue 1!

The PageRank vector is the principal eigenvector of the modified transition matrix.

Spectral Clustering

Spectral clustering uses eigenvectors of similarity matrices for clustering.

1. Construct similarity graph from data
2. Compute graph Laplacian matrix L
3. Compute first k eigenvectors of L
4. Cluster points in this eigenvector space using k-means

Theorem

The multiplicity of the eigenvalue 0 of the Laplacian equals the number of connected components in the graph.

Spectral clustering can find complex cluster structures that traditional methods miss.

Eigendecomposition in Machine Learning

- **Linear Discriminant Analysis (LDA)**: Finds directions that maximize class separation using generalized eigenvalue problems
- **Recommendation Systems**: Singular Value Decomposition (SVD, related to eigendecomposition) factorizes user-item matrices
- **Neural Networks**: Eigendecomposition helps analyze learning dynamics and understand vanishing/exploding gradients
- **Kernel Methods**: Many kernel-based algorithms rely on eigendecomposition of kernel matrices
- **Graph Analytics**: Eigenvectors of adjacency and Laplacian matrices reveal community structure

Numerical Considerations

Important Practical Notes

- For large matrices, we don't compute full eigendecomposition (too expensive)
- We use iterative methods (power method, Lanczos algorithm) to find dominant eigenvectors
- In practice, we often use Singular Value Decomposition (SVD) which is more numerically stable
- For symmetric matrices, specialized algorithms exploit the structure

Summary: Why Eigendecomposition Matters

- **Fundamental understanding**: Reveals the intrinsic structure of linear transformations
- **Computational efficiency**: Simplifies matrix operations and analyses
- **Data insight**: Finds natural coordinate systems and patterns in data
- **Dimensionality reduction**: Enables working with high-dimensional data
- **Algorithm foundation**: Underlies many core data science algorithms

Eigendecomposition is truly one of the most powerful tools
in the data scientist's toolkit!

References and Further Reading

- Strang, G. (2016). *Introduction to Linear Algebra*
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*
- Golub, G. H., & Van Loan, C. F. (2013). *Matrix Computations*
- Jolliffe, I. T. (2002). *Principal Component Analysis*
- Von Luxburg, U. (2007). A tutorial on spectral clustering