

Advances in Dimensionality Reduction

Nathan Mankovich



ISP · Image & Signal Processing
Universitat de València

DR Reduces the Feature Space

DR Reduces the Feature Space

- ▶ p samples

DR Reduces the Feature Space

- ▶ p samples
- ▶ d dimensional ambient space

DR Reduces the Feature Space

- ▶ p samples
- ▶ d dimensional ambient space
- ▶ Input dataset \mathcal{D} (d features)

DR Reduces the Feature Space

- ▶ p samples
- ▶ d dimensional ambient space
- ▶ Input dataset \mathcal{D} (d features)
- ▶ Output: reduced dataset \mathcal{R} (k features)

DR Reduces the Feature Space

- ▶ p samples
- ▶ d dimensional ambient space
- ▶ Input dataset \mathcal{D} (d features)
- ▶ Output: reduced dataset \mathcal{R} (k features)

Dimensionality Reduction (DR) maps d -dimensional features to k -dimensional features ($k < d$).

DR Reduces the Feature Space

- ▶ p samples
- ▶ d dimensional ambient space
- ▶ Input dataset \mathcal{D} (d features)
- ▶ Output: reduced dataset \mathcal{R} (k features)

Dimensionality Reduction (DR) maps d -dimensional features to k -dimensional features ($k < d$).

Question: Why do we want to use dimensionality reduction?

Dimensionality Reduction in Remote Sensing

Dimensionality Reduction in Remote Sensing

Input a data cube with spectral, spatial, and temporal dimensions.

Dimensionality Reduction in Remote Sensing

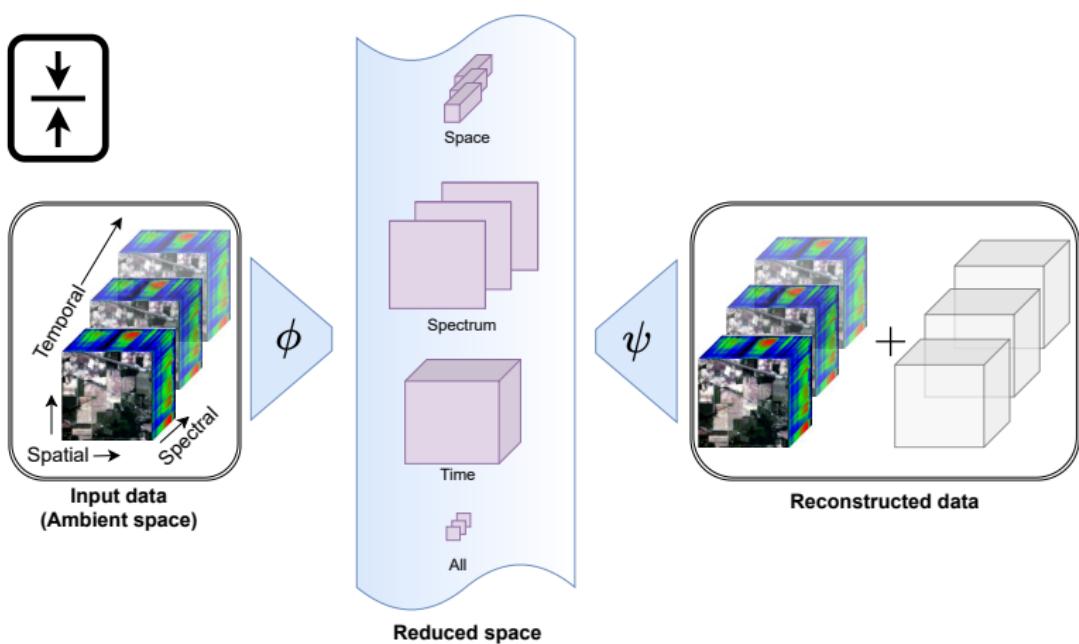
Input a data cube with spectral, spatial, and temporal dimensions.

Dimensionality Reduction in Remote Sensing

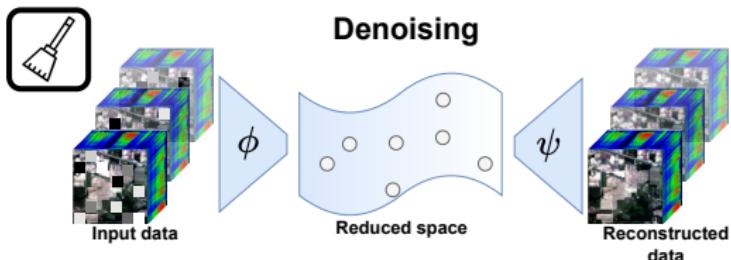
Input a data cube with spectral, spatial, and temporal dimensions.

- ▶ Compression
- ▶ Denoising
- ▶ Fusion
- ▶ Visualization
- ▶ Anomaly detection
- ▶ Improving predictions

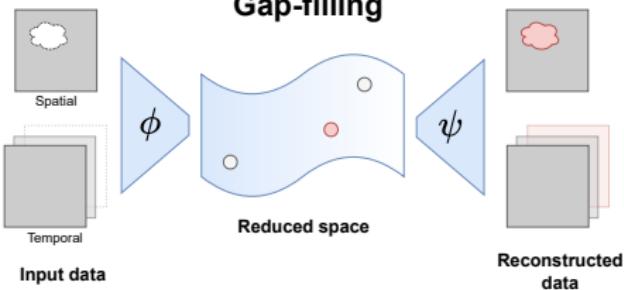
Compression



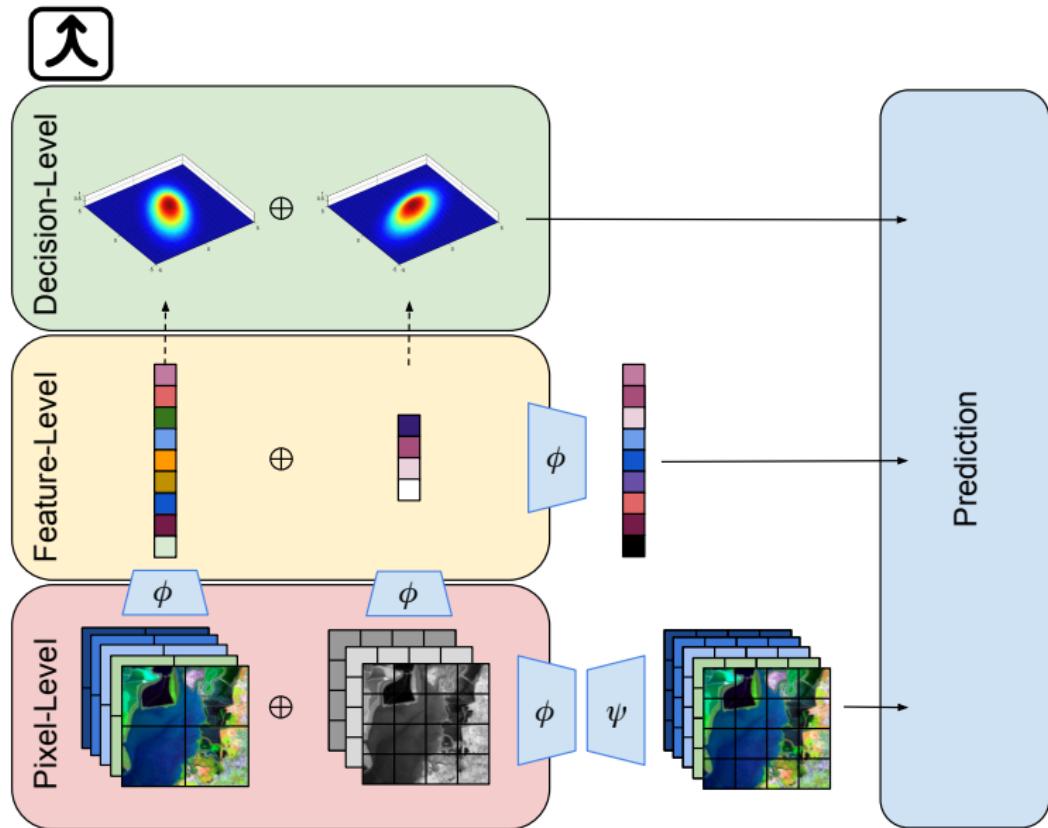
Denoising



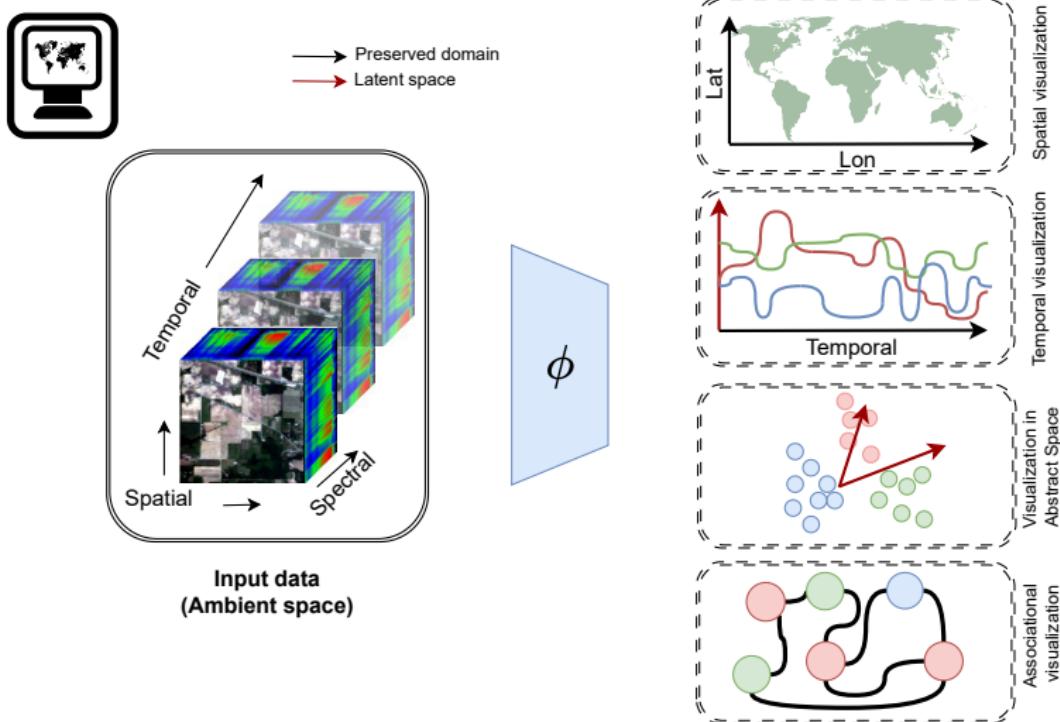
Gap-filling



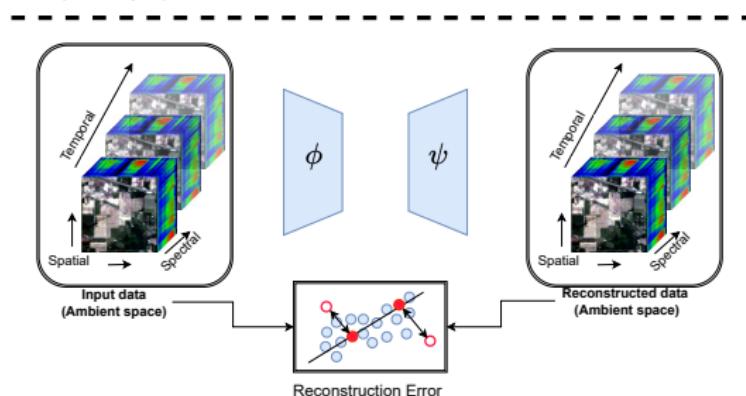
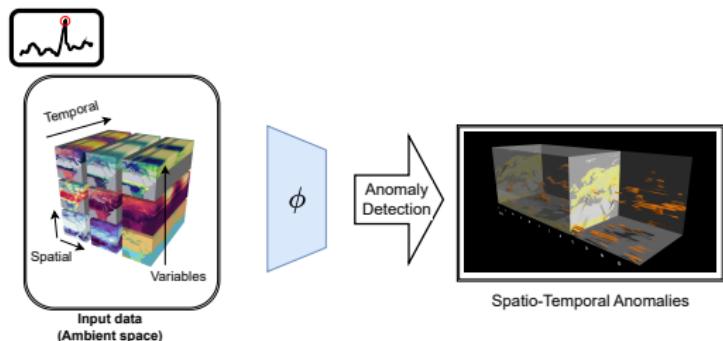
Fusion



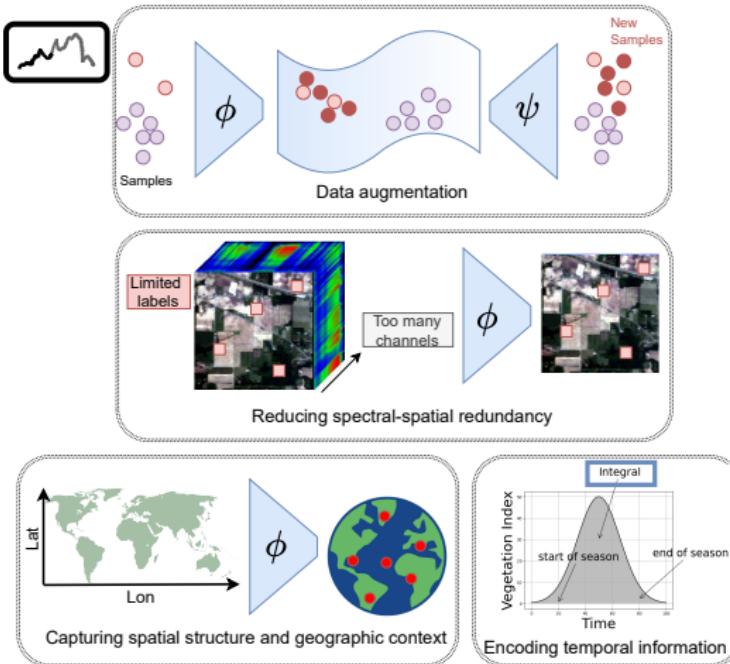
Visualization



Anomaly Detection



Improved Predictions

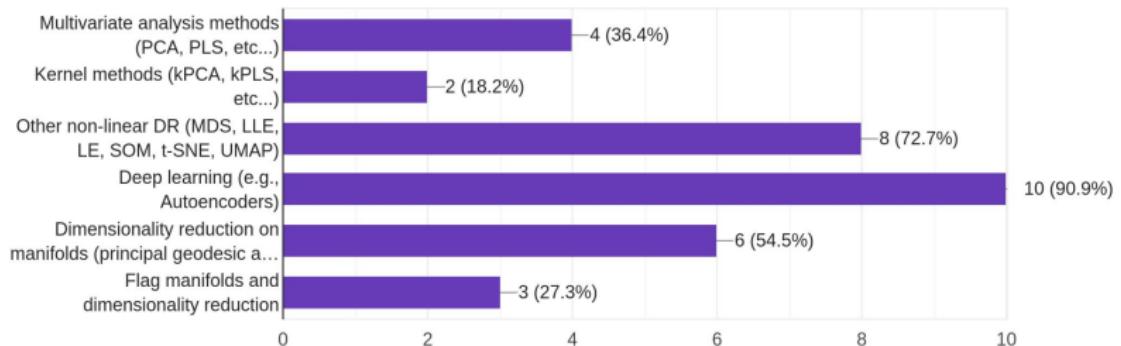


Course Survey Results

Survey Results

Topics covered (choose up to 3)

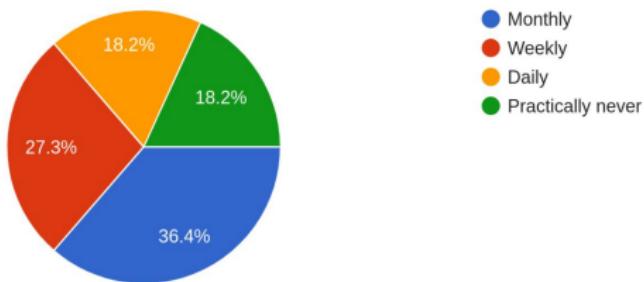
11 responses



Survey Results

How often do you use linear algebra

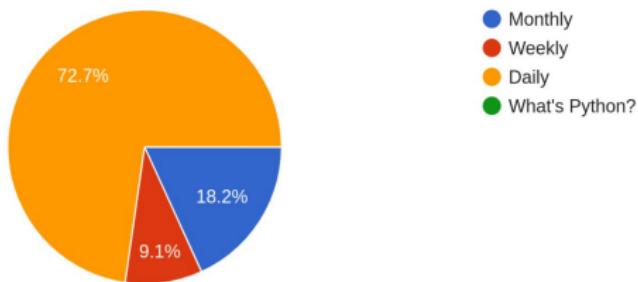
11 responses



Survey Results

How often do you code in python?

11 responses



Course Outline

1. Linear Algebra Review
2. Intro to DR
3. Linear Dimensionality Reduction
4. Nonlinear Dimensionality Reduction
5. Neural Network-Based Methods

Linear Algebra Review

Further reading [Strang, 2000]

Outline

1. Vectors & Inner Products
2. Subspaces, Orthogonality, Projections
3. Singular Value Decomposition (SVD)
4. Eigenvalue Decomposition

Field Axioms

A field F is a set equipped with two operations (addition and multiplication) satisfying the following axioms for all $a, b, c \in F$:

Field Axioms

A field F is a set equipped with two operations (addition and multiplication) satisfying the following axioms for all $a, b, c \in F$:

► **Associativity:**

- Addition: $a + (b + c) = (a + b) + c$
- Multiplication: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

Field Axioms

A field F is a set equipped with two operations (addition and multiplication) satisfying the following axioms for all $a, b, c \in F$:

► **Associativity:**

- Addition: $a + (b + c) = (a + b) + c$
- Multiplication: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

► **Commutativity:**

- Addition: $a + b = b + a$
- Multiplication: $a \cdot b = b \cdot a$

Field Axioms

A field F is a set equipped with two operations (addition and multiplication) satisfying the following axioms for all $a, b, c \in F$:

► **Associativity:**

- Addition: $a + (b + c) = (a + b) + c$
- Multiplication: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

► **Commutativity:**

- Addition: $a + b = b + a$
- Multiplication: $a \cdot b = b \cdot a$

► **Identities:**

- Additive identity: there exists $0 \in F$ such that $a + 0 = a$
- Multiplicative identity: there exists $1 \in F$, $1 \neq 0$, such that $a \cdot 1 = a$

Field Axioms

A field F is a set equipped with two operations (addition and multiplication) satisfying the following axioms for all $a, b, c \in F$:

► **Associativity:**

- Addition: $a + (b + c) = (a + b) + c$
- Multiplication: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

► **Commutativity:**

- Addition: $a + b = b + a$
- Multiplication: $a \cdot b = b \cdot a$

► **Identities:**

- Additive identity: there exists $0 \in F$ such that $a + 0 = a$
- Multiplicative identity: there exists $1 \in F$, $1 \neq 0$, such that $a \cdot 1 = a$

► **Inverses:**

- Additive inverse: for every $a \in F$, there exists $-a \in F$ such that $a + (-a) = 0$
- Multiplicative inverse: for every $a \neq 0$ in F , there exists $a^{-1} \in F$ such that $a \cdot a^{-1} = 1$

Field Axioms

A field F is a set equipped with two operations (addition and multiplication) satisfying the following axioms for all $a, b, c \in F$:

► **Associativity:**

- Addition: $a + (b + c) = (a + b) + c$
- Multiplication: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

► **Commutativity:**

- Addition: $a + b = b + a$
- Multiplication: $a \cdot b = b \cdot a$

► **Identities:**

- Additive identity: there exists $0 \in F$ such that $a + 0 = a$
- Multiplicative identity: there exists $1 \in F$, $1 \neq 0$, such that $a \cdot 1 = a$

► **Inverses:**

- Additive inverse: for every $a \in F$, there exists $-a \in F$ such that $a + (-a) = 0$
- Multiplicative inverse: for every $a \neq 0$ in F , there exists $a^{-1} \in F$ such that $a \cdot a^{-1} = 1$

► **Distributivity:** $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

The set of all real numbers (denoted \mathbb{R}) is a Field

- ▶ Addition and multiplication operations
- ▶ Additive identity is 0
- ▶ Multiplicative identity is 1

Exercise: Show that \mathbb{R} satisfies the axioms of a field, whereas \mathbb{Z} (the set of all integers) does not.

Vector Space Axioms

A vector space over a field F satisfies the following axioms for all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and all scalars $a, b \in F$:

Vector Space Axioms

A vector space over a field F satisfies the following axioms for all $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$ and all scalars $a, b \in F$:

- ▶ **Associativity of addition:** $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
- ▶ **Commutativity of addition:** $\mathbf{u} + \mathbf{v} = \mathbf{v} + \mathbf{u}$
- ▶ **Additive identity:** There exists $\mathbf{0} \in V$ such that $\mathbf{v} + \mathbf{0} = \mathbf{v}$
- ▶ **Additive inverse:** For each $\mathbf{v} \in V$, there exists $-\mathbf{v} \in V$ such that $\mathbf{v} + (-\mathbf{v}) = \mathbf{0}$
- ▶ **Compatibility with scalar multiplication:** $a(b\mathbf{v}) = (ab)\mathbf{v}$
- ▶ **Identity element of scalar multiplication:** $1\mathbf{v} = \mathbf{v}$, where 1 is the multiplicative identity in F
- ▶ **Distributivity over vector addition:** $a(\mathbf{u} + \mathbf{v}) = a\mathbf{u} + a\mathbf{v}$
- ▶ **Distributivity over field addition:** $(a + b)\mathbf{v} = a\mathbf{v} + b\mathbf{v}$

\mathbb{R}^2 is a Vector Space

\mathbb{R}^2 is a Vector Space over \mathbb{R} .

$$\mathbb{R}^2 = \{\mathbf{a} = [a_1 \ a_2] : a_1, a_2 \in \mathbb{R}\}$$

Exercise: Show that \mathbb{R}^2 satisfies the axioms of a vector space.

Dot Product & Friends

Dot Product & Friends

► Transpose

$$\mathbf{a}^\top = [a_1 \quad a_2]$$

Dot Product & Friends

► Transpose

$$\mathbf{a}^\top = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$$

► Dot product

$$\mathbf{a}^\top \mathbf{b} = a_1 b_1 + a_2 b_2$$

Dot Product & Friends

► Transpose

$$\mathbf{a}^\top = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$$

► Dot product

$$\mathbf{a}^\top \mathbf{b} = a_1 b_1 + a_2 b_2$$

Two vectors are called “orthogonal” if their dot product is zero.

Dot Product & Friends

- ▶ **Transpose**

$$\mathbf{a}^\top = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$$

- ▶ **Dot product**

$$\mathbf{a}^\top \mathbf{b} = a_1 b_1 + a_2 b_2$$

Two vectors are called “orthogonal” if their dot product is zero.

- ▶ **2-Norm**

$$\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^\top \mathbf{a}}$$

Dot Product & Friends

- ▶ **Transpose**

$$\mathbf{a}^\top = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$$

- ▶ **Dot product**

$$\mathbf{a}^\top \mathbf{b} = a_1 b_1 + a_2 b_2$$

Two vectors are called “orthogonal” if their dot product is zero.

- ▶ **2-Norm**

$$\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^\top \mathbf{a}}$$

- ▶ **Distance**

$$\|\mathbf{a} - \mathbf{b}\|_2$$

Dot Product & Friends

- ▶ **Transpose**

$$\mathbf{a}^\top = \begin{bmatrix} a_1 & a_2 \end{bmatrix}$$

- ▶ **Dot product**

$$\mathbf{a}^\top \mathbf{b} = a_1 b_1 + a_2 b_2$$

Two vectors are called “orthogonal” if their dot product is zero.

- ▶ **2-Norm**

$$\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^\top \mathbf{a}}$$

- ▶ **Distance**

$$\|\mathbf{a} - \mathbf{b}\|_2$$

- ▶ **Angle between vectors**

$$\cos(\theta) = \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\|_2 \|\mathbf{b}\|_2}$$

Span

Span

The *span* of two vectors \mathbf{a}, \mathbf{b} is the set of all vectors that can be written as a linear combination of \mathbf{a} and \mathbf{b}

Span

The *span* of two vectors \mathbf{a}, \mathbf{b} is the set of all vectors that can be written as a linear combination of \mathbf{a} and \mathbf{b}

$$\text{span}(\mathbf{a}, \mathbf{b}) = \alpha\mathbf{a} + \beta\mathbf{b} : \alpha, \beta \in \mathbb{R} \quad (1)$$

Linear Independence

A set of vectors $\{v_1, v_2, \dots, v_k\}$ in a vector space V is **linearly independent** if:

$$c_1v_1 + c_2v_2 + \dots + c_kv_k = 0 \quad \Rightarrow \quad c_1 = c_2 = \dots = c_k = 0$$

Linear Independence

A set of vectors $\{v_1, v_2, \dots, v_k\}$ in a vector space V is **linearly independent** if:

$$c_1v_1 + c_2v_2 + \dots + c_kv_k = 0 \quad \Rightarrow \quad c_1 = c_2 = \dots = c_k = 0$$

In other words, the only solution to the linear combination equaling zero is the **trivial solution**.

Linear Independence

A set of vectors $\{v_1, v_2, \dots, v_k\}$ in a vector space V is **linearly independent** if:

$$c_1v_1 + c_2v_2 + \dots + c_kv_k = 0 \quad \Rightarrow \quad c_1 = c_2 = \dots = c_k = 0$$

In other words, the only solution to the linear combination equaling zero is the **trivial solution**.

Linear Independence

A set of vectors $\{v_1, v_2, \dots, v_k\}$ in a vector space V is **linearly independent** if:

$$c_1v_1 + c_2v_2 + \dots + c_kv_k = 0 \quad \Rightarrow \quad c_1 = c_2 = \dots = c_k = 0$$

In other words, the only solution to the linear combination equaling zero is the **trivial solution**.

Example:

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

are linearly independent in \mathbb{R}^2 , since neither can be written as a multiple of the other.

What is a Basis?

What is a Basis?

A **basis** of a vector space V is a set of vectors $\{v_1, v_2, \dots, v_n\} \subset V$ such that:

1. The vectors are **linearly independent**.

What is a Basis?

A **basis** of a vector space V is a set of vectors $\{v_1, v_2, \dots, v_n\} \subset V$ such that:

1. The vectors are **linearly independent**.
2. The vectors **span** V , i.e., every vector in V can be written as a linear combination of v_1, \dots, v_n .

What is a Basis?

A **basis** of a vector space V is a set of vectors $\{v_1, v_2, \dots, v_n\} \subset V$ such that:

1. The vectors are **linearly independent**.
2. The vectors **span** V , i.e., every vector in V can be written as a linear combination of v_1, \dots, v_n .

What is a Basis?

A **basis** of a vector space V is a set of vectors $\{v_1, v_2, \dots, v_n\} \subset V$ such that:

1. The vectors are **linearly independent**.
2. The vectors **span** V , i.e., every vector in V can be written as a linear combination of v_1, \dots, v_n .

Example: The standard basis for \mathbb{R}^3 is:

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

What is a Basis?

A **basis** of a vector space V is a set of vectors $\{v_1, v_2, \dots, v_n\} \subset V$ such that:

1. The vectors are **linearly independent**.
2. The vectors **span** V , i.e., every vector in V can be written as a linear combination of v_1, \dots, v_n .

Example: The standard basis for \mathbb{R}^3 is:

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\} \quad (2)$$

What is a Basis?

A **basis** of a vector space V is a set of vectors $\{v_1, v_2, \dots, v_n\} \subset V$ such that:

1. The vectors are **linearly independent**.
2. The vectors **span** V , i.e., every vector in V can be written as a linear combination of v_1, \dots, v_n .

Example: The standard basis for \mathbb{R}^3 is:

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right\} \quad (2)$$

The **dimension** of a vector space V is the number of elements in a basis for V .

Matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

Matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

- ▶ The **row space** of \mathbf{A} (denoted $\text{row}(\mathbf{A})$) is the set of linear combinations of rows.

Matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

- ▶ The **row space** of \mathbf{A} (denoted $\text{row}(\mathbf{A})$) is the set of linear combinations of rows.
- ▶ The **column space** of \mathbf{A} (denoted $\text{col}(\mathbf{A})$) is the set of linear combinations of columns.

Matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

- ▶ The **row space** of \mathbf{A} (denoted $\text{row}(\mathbf{A})$) is the set of linear combinations of rows.
- ▶ The **column space** of \mathbf{A} (denoted $\text{col}(\mathbf{A})$) is the set of linear combinations of columns.
- ▶ The **rank** of \mathbf{A} is the dimension of its column space.

Matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

- ▶ The **row space** of \mathbf{A} (denoted $\text{row}(\mathbf{A})$) is the set of linear combinations of rows.
- ▶ The **column space** of \mathbf{A} (denoted $\text{col}(\mathbf{A})$) is the set of linear combinations of columns.
- ▶ The **rank** of \mathbf{A} is the dimension of its column space.
- ▶ The **trace** of \mathbf{A} is the sum of the diagonal entries

$$\text{tr}(\mathbf{A}) = a_{11} + a_{22}$$

Matrices

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix}$$

- ▶ The **row space** of \mathbf{A} (denoted $\text{row}(\mathbf{A})$) is the set of linear combinations of rows.
- ▶ The **column space** of \mathbf{A} (denoted $\text{col}(\mathbf{A})$) is the set of linear combinations of columns.
- ▶ The **rank** of \mathbf{A} is the dimension of its column space.
- ▶ The **trace** of \mathbf{A} is the sum of the diagonal entries

$$\text{tr}(\mathbf{A}) = a_{11} + a_{22}$$

- ▶ The Frobenius norm of \mathbf{A} is

$$\|\mathbf{A}\|_F = \sqrt{\text{tr}(\mathbf{A}^\top \mathbf{A})}$$

Matrix multiplication

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

Matrix multiplication

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} [a_{11} & a_{12}] \\ [a_{21} & a_{22}] \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \\ b_{11} \\ b_{21} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{12} \\ b_{22} \\ b_{12} \\ b_{22} \end{bmatrix}$$

Lab 1: Vectors & Matrices

Go to Lab1_VectorsMatrices.ipynb

Projections

Projections

Projection of \mathbf{b} onto $\text{col}(\mathbf{A})$:

$$\Pi_{\mathbf{A}}(\mathbf{b}) = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

Projections

Projection of \mathbf{b} onto $\text{col}(\mathbf{A})$:

$$\Pi_{\mathbf{A}}(\mathbf{b}) = \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b}$$

Exercise: If the columns of \mathbf{A} are orthogonal, show that the projection onto the column space of \mathbf{A} is $\mathbf{A}\mathbf{A}^T$.

Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD)

SVD of matrix $\mathbf{A} \in \mathbb{R}^{p \times d}$ of rank r :

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$$

Singular Value Decomposition (SVD)

SVD of matrix $\mathbf{A} \in \mathbb{R}^{p \times d}$ of rank r :

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$$

- ▶ $\mathbf{U} \in \mathbb{R}^{p \times r}$: left singular vectors, $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$
- ▶ $\Sigma \in \mathbb{R}^{r \times r}$: diagonal matrix of singular values
- ▶ $\mathbf{V}^\top \in \mathbb{R}^{r \times d}$: right singular vectors, $\mathbf{V}^\top\mathbf{V} = \mathbf{I}$

Singular Value Decomposition (SVD)

SVD of matrix $\mathbf{A} \in \mathbb{R}^{p \times d}$ of rank r :

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$$

- ▶ $\mathbf{U} \in \mathbb{R}^{p \times r}$: left singular vectors, $\mathbf{U}^\top \mathbf{U} = \mathbf{I}$
- ▶ $\Sigma \in \mathbb{R}^{r \times r}$: diagonal matrix of singular values
- ▶ $\mathbf{V}^\top \in \mathbb{R}^{r \times d}$: right singular vectors, $\mathbf{V}^\top \mathbf{V} = \mathbf{I}$

Applications: dimensionality reduction, image compression, linear systems.

Eigenvalue Decomposition from SVD

Given the SVD

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$$

Eigenvalue Decomposition from SVD

Given the SVD

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$$

The Eigenvalue Decomposition of $\mathbf{A}^\top \mathbf{A}$ is

$$\mathbf{C} = \mathbf{A}^\top \mathbf{A} = \mathbf{V}\Sigma^\top \Sigma\mathbf{V}^\top = \mathbf{V}\Lambda\mathbf{V}^\top$$

Eigenvalue Decomposition from SVD

Given the SVD

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$$

The Eigenvalue Decomposition of $\mathbf{A}^\top \mathbf{A}$ is

$$\mathbf{C} = \mathbf{A}^\top \mathbf{A} = \mathbf{V}\Sigma^\top \Sigma\mathbf{V}^\top = \mathbf{V}\Lambda\mathbf{V}^\top$$

- ▶ **Eigenvectors**: columns of \mathbf{V} , $\mathbf{V}^\top \mathbf{V} = \mathbf{I}$
- ▶ **Eigenvalues**: diagonal of Σ

Eigenvalue Decomposition from SVD

Given the SVD

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

The Eigenvalue Decomposition of $\mathbf{A}^T\mathbf{A}$ is

$$\mathbf{C} = \mathbf{A}^T\mathbf{A} = \mathbf{V}\Sigma^T\Sigma\mathbf{V}^T = \mathbf{V}\Lambda\mathbf{V}^T$$

- ▶ **Eigenvectors:** columns of \mathbf{V} , $\mathbf{V}^T\mathbf{V} = \mathbf{I}$
- ▶ **Eigenvalues:** diagonal of Σ

Exercises:

- ▶ Show the eigenvalue decomposition of $\mathbf{A}^T\mathbf{A}$ is $\mathbf{U}\Sigma\Sigma^T\mathbf{U}^T$
- ▶ Show the trace of \mathbf{C} is the sum of its eigenvalues

Eigenvalue Optimization Formulation

Eigenvalue Optimization Formulation

1. Raleigh quotient:

$$\lambda_i = \max_{\mathbf{v}^\top \mathbf{v}_j = 0 \ \forall j < i} \frac{\mathbf{v}^\top \mathbf{C} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}$$

Eigenvalue Optimization Formulation

1. Raleigh quotient:

$$\lambda_i = \max_{\mathbf{v}^\top \mathbf{v} = 0 \ \forall j < i} \frac{\mathbf{v}^\top \mathbf{C} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}$$

2. Via trace:

$$\mathbf{V} = \arg \max_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \text{tr}(\mathbf{W}^\top \mathbf{C} \mathbf{W})$$

Eigenvalue Optimization Formulation

1. Raleigh quotient:

$$\lambda_i = \max_{\mathbf{v}^\top \mathbf{v}_j = 0 \ \forall j < i} \frac{\mathbf{v}^\top \mathbf{C} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}$$

2. Via trace:

$$\mathbf{V} = \arg \max_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \text{tr}(\mathbf{W}^\top \mathbf{C} \mathbf{W})$$

3. Reconstruction error:

$$\mathbf{V} = \arg \min_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \|\mathbf{A} - \mathbf{A} \mathbf{W} \mathbf{W}^\top\|_F^2$$

Eigenvalue Optimization Formulation

1. Raleigh quotient:

$$\lambda_i = \max_{\mathbf{v}^\top \mathbf{v} = 0 \ \forall j < i} \frac{\mathbf{v}^\top \mathbf{C} \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}$$

2. Via trace:

$$\mathbf{V} = \arg \max_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \text{tr}(\mathbf{W}^\top \mathbf{C} \mathbf{W})$$

3. Reconstruction error:

$$\mathbf{V} = \arg \min_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \|\mathbf{A} - \mathbf{A} \mathbf{W} \mathbf{W}^\top\|_F^2$$

Exercise: Show that these optimizations are equivalent.

Generalized Eigenvalue Decomposition

Solve:

$$\mathbf{C}_A \mathbf{w} = \lambda \mathbf{C}_B \mathbf{w}$$

Optimization form:

$$\lambda_i = \max_{\mathbf{w}^\top \mathbf{w}_j = 0 \ \forall j < i} \frac{\mathbf{w}^\top \mathbf{C}_A \mathbf{w}}{\mathbf{w}^\top \mathbf{C}_B \mathbf{w}}$$

Approximate with:

$$\mathbf{C}_B^{-1} \mathbf{C}_A$$

Use SVD for pseudo-inverse:

- ▶ $\mathbf{C}_B = \mathbf{U} \Sigma \mathbf{V}^\top$
- ▶ $\mathbf{C}_B^\dagger = \mathbf{V} \Sigma^{-1} \mathbf{U}^\top$

Condition number:

$$\kappa(\mathbf{C}_B) = \frac{\sigma_{\max}}{\sigma_{\min}}$$

Big κ bad, Small κ good

Lab 2: Matrix Decompositions

Go to `Lab2_MatrixDecompositions.ipynb`

Gradient Descent

Gradient Descent: Overview

- ▶ Gradient Descent is an optimization algorithm to minimize a function
- ▶ It iteratively moves in the direction of steepest descent (negative gradient)

Objective

Minimize a loss function $f(\theta)$, where:

- ▶ θ are the model parameters

Update Rule

Update Rule

At each iteration, the parameters are updated as:

$$\theta_{t+1} = \theta_t - \eta \nabla f(\theta_t)$$

- ▶ θ_t : Current parameters at iteration t
- ▶ η : Learning rate (step size)
- ▶ $\nabla f(\theta_t)$: Gradient of the loss function

Intuition

Intuition

- ▶ $\nabla f(\theta_t)$ points to the direction of steepest ascent.

Intuition

- ▶ $\nabla f(\theta_t)$ points to the direction of steepest ascent.
- ▶ Moving in the opposite direction reduces the function value.

Intuition

- ▶ $\nabla f(\theta_t)$ points to the direction of steepest ascent.
- ▶ Moving in the opposite direction reduces the function value.
- ▶ Learning rate η controls the step size to ensure convergence.

Convergence

Gradient Descent converges to a local minimum (or global minimum for convex functions) when:

$$\lim_{t \rightarrow \infty} \nabla f(\theta_t) = 0$$

The Landscape of Dimensionality Reduction

Inspired by Lee and Verleysen [2007]

DR Reduces the Feature Space

DR Reduces the Feature Space

Data

DR Reduces the Feature Space

Data

- ▶ p samples

DR Reduces the Feature Space

Data

- ▶ p samples
- ▶ d dimensional ambient space

DR Reduces the Feature Space

Data

- ▶ p samples
- ▶ d dimensional ambient space
- ▶ Dataset \mathcal{D} (d features)

DR Reduces the Feature Space

Data

- ▶ p samples
- ▶ d dimensional ambient space
- ▶ Dataset \mathcal{D} (d features)
- ▶ Reduced dataset \mathcal{R} (k features)

Dimensionality Reduction (DR) maps d -dimensional features to k -dimensional features ($k < d$).

Why DR?

Why DR?

give me 5 fake and funny quotes from researchers about how DR changed their life



Sure! Here are 5 fake and funny quotes from fictional researchers on how dimensionality reduction (DR) changed their lives:

1. *"Before PCA, my life was just noise. Now it's all principal components — including my dating life!"*

— **Dr. Ada Scatter**, Romantic Data Scientist

2. *"I applied t-SNE to my thoughts and finally found out why I kept forgetting my keys."*

— **Prof. Moe Mertz**, Cognitive Chaos Lab

3. *"After discovering manifold learning, I realized my career path wasn't linear either — just locally Euclidean."*

— **Dr. Lin E. Ar**, Nonlinear Trajectory Researcher

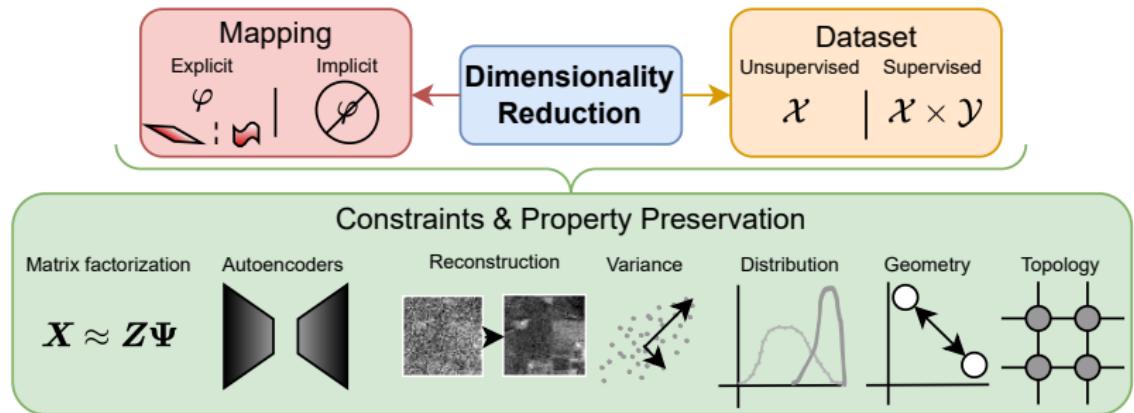
4. *"With UMAP, I reduced my midlife crisis to just two dimensions: wine and existential dread."*

— **Dr. Max E. Stress**, Professor of Reduced Expectations

5. *"Autoencoders helped me compress my emotions. Now I only cry in low resolution."*

— **Dr. Dee Pression**, Deep Learning Enthusiast

DR Summary



Dataset

Unsupervised

$$\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\} \subset \mathbb{R}^d$$

e.g., Visualize data in 2D to see if there are any patterns

Dataset

Unsupervised

$$\mathcal{D} = \mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\} \subset \mathbb{R}^d$$

e.g., Visualize data in 2D to see if there are any patterns

Supervised

$$\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_p\}, \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_p\}$$

e.g., Find a low-dimensional, discriminatory feature space w.r.t. \mathcal{Y}

DR Mapping

DR Mapping

Explicit DR mapping outputs ϕ where

$$\phi(\mathbf{x}_i) = \mathbf{z}_i$$

- ▶ Approximate inverse $\psi \approx \phi^{-1}$
- ▶ Reconstructions: $\mathbf{x}_i \approx \hat{\mathbf{x}}_i = \psi(\phi(\mathbf{x}_i))$

e.g., Want to fit a model on some data, then apply it to “unseen” data.

Implicit: ϕ, ψ are not explicitly defined but inferred

e.g., Model fit on all the data to be reduced

Matrix Factorization

Matrix Factorization

Data matrix: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]^\top \in \mathbb{R}^{p \times d}$

Matrix Factorization

Data matrix: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]^\top \in \mathbb{R}^{p \times d}$

Reduced data matrix: $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_p]^\top \in \mathbb{R}^{p \times k}$

Matrix Factorization

Data matrix: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]^\top \in \mathbb{R}^{p \times d}$

Reduced data matrix: $\mathbf{Z} = [\mathbf{z}_1 \cdots \mathbf{z}_p]^\top \in \mathbb{R}^{p \times k}$

The model:

$$\mathbf{X} \approx \mathbf{Z}\Psi$$

Autoencoders

Autoencoders

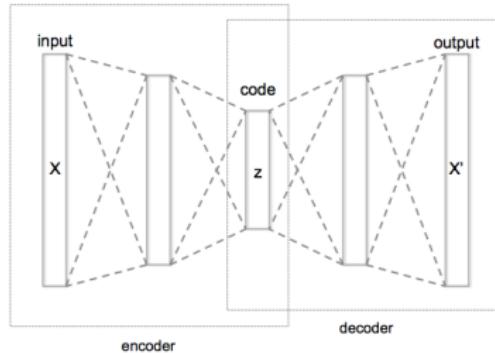


Figure: Image from
https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png

Autoencoders

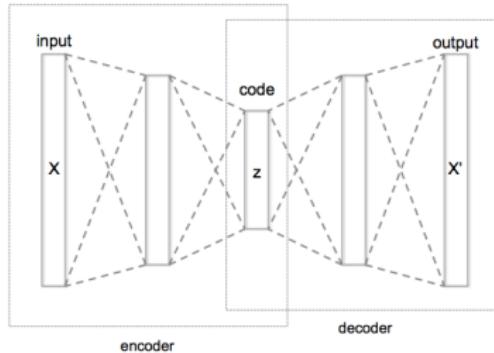


Figure: Image from
https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png

- Neural networks trained to reconstruct input

Autoencoders

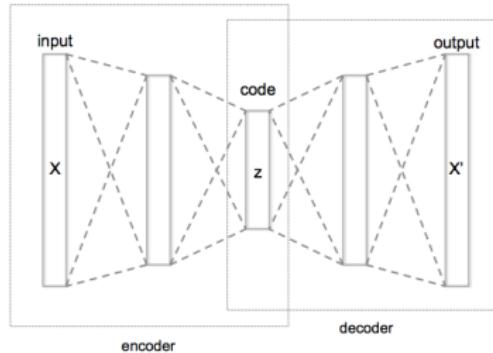


Figure: Image from
https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png

- ▶ Neural networks trained to reconstruct input
- ▶ Bottleneck structure forces compression

Autoencoders

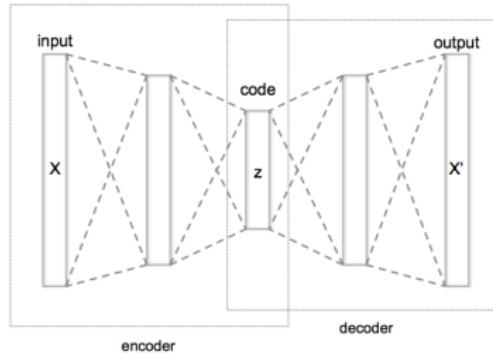


Figure: Image from

https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png

- ▶ Neural networks trained to reconstruct input
- ▶ Bottleneck structure forces compression
- ▶ Encoder: $\phi(x) = z$, Decoder: $\psi(z) = \hat{x}$

Autoencoders

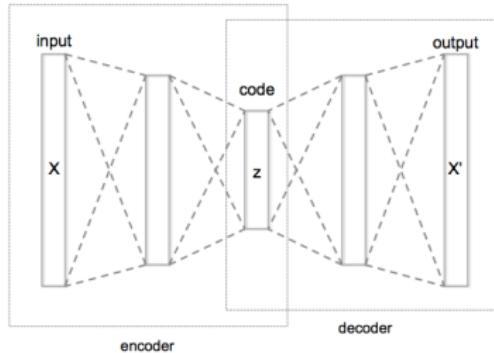


Figure: Image from
https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png

- ▶ Neural networks trained to reconstruct input
- ▶ Bottleneck structure forces compression
- ▶ Encoder: $\phi(x) = z$, Decoder: $\psi(z) = \hat{x}$
- ▶ Loss: e.g., MSE $\sum_i \|x_i - \psi \circ \phi(x_i)\|^2$

Autoencoders

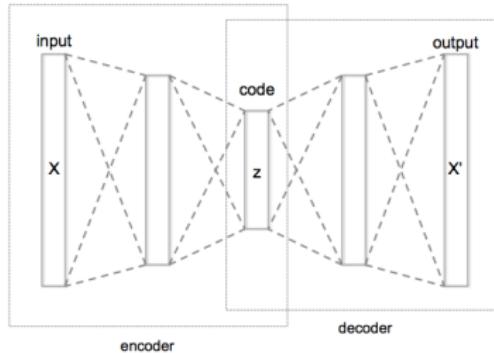


Figure: Image from

https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png

- ▶ Neural networks trained to reconstruct input
- ▶ Bottleneck structure forces compression
- ▶ Encoder: $\phi(x) = z$, Decoder: $\psi(z) = \hat{x}$
- ▶ Loss: e.g., $\text{MSE} \sum_i \|x_i - \psi \circ \phi(x_i)\|^2$
- ▶ Variants: Denoising, Variational (VAE), Convolutional

DR as Optimization

$$\arg \min_{\mathcal{Z}} L(\mathcal{X}, \mathcal{Z}) \quad (3)$$

DR as Optimization

$$\arg \min_{\mathcal{Z}} L(\mathcal{X}, \mathcal{Z}) \quad (3)$$

This general form preserves

- ▶ Reconstructions
- ▶ Variance
- ▶ Probability distributions
- ▶ Geometry
- ▶ Graph structures

Reconstruction-preserving

Reconstruction-preserving

$$\begin{aligned} \min_{\psi, \phi} \quad & \sum_{i=1}^p \|\mathbf{x}_i - \psi \circ \phi(\mathbf{x}_i)\|^2 \\ \text{s.t.} \quad & \phi: \mathbb{R}^d \rightarrow \mathbb{R}^k, \quad \psi: \mathbb{R}^k \rightarrow \mathbb{R}^d \end{aligned} \tag{4}$$

Reconstruction-preserving

$$\begin{aligned} \min_{\psi, \phi} \quad & \sum_{i=1}^p \|\mathbf{x}_i - \psi \circ \phi(\mathbf{x}_i)\|^2 \\ \text{s.t. } \quad & \phi : \mathbb{R}^d \rightarrow \mathbb{R}^k, \quad \psi : \mathbb{R}^k \rightarrow \mathbb{R}^d \end{aligned} \tag{4}$$

Useful when interpretability or invertibility is important.

Variance-preserving

Data matrix: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]^\top$

Variance-preserving

Data matrix: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]^\top$

$$\begin{aligned}\Phi = \arg \max \text{Tr}(\mathbf{W}^\top \mathbf{X}^\top \mathbf{X} \mathbf{W}) \\ \text{s.t. } \mathbf{W}^\top \mathbf{W} = I\end{aligned}\tag{5}$$

Variance-preserving

Data matrix: $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_p]^\top$

$$\begin{aligned}\Phi = \arg \max \text{Tr}(\mathbf{W}^\top \mathbf{X}^\top \mathbf{X} \mathbf{W}) \\ \text{s.t. } \mathbf{W}^\top \mathbf{W} = I\end{aligned}\tag{5}$$

Here, we assume the DR mapping is linear: $\phi(\mathbf{x}_i) = \mathbf{W}^\top \mathbf{x}_i$

Distribution-preserving

Distribution-preserving

- ▶ P “true” (target) distribution with distribution function p
- ▶ Q “predicted” (modeled) distribution with distribution function q

Distribution-preserving

- ▶ P “true” (target) distribution with distribution function p
- ▶ Q “predicted” (modeled) distribution with distribution function q

Entropy

$$H(p) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

Distribution-preserving

- ▶ P “true” (target) distribution with distribution function p
- ▶ Q “predicted” (modeled) distribution with distribution function q

Entropy

$$H(p) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

Cross Entropy

$$H(p, q) = - \sum_{x \in \mathcal{X}} p(x) \log q(x) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) + \text{KL}(p \parallel q)$$

Distribution-preserving

- ▶ P “true” (target) distribution with distribution function p
- ▶ Q “predicted” (modeled) distribution with distribution function q

Entropy

$$H(p) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

Cross Entropy

$$H(p, q) = - \sum_{x \in \mathcal{X}} p(x) \log q(x) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) + \text{KL}(p \parallel q)$$

Kullback–Leibler (KL) Divergence

$$\text{KL}(p \parallel q) = H(p, q) - H(p) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \quad (6)$$

Distribution-preserving

- ▶ P “true” (target) distribution with distribution function p
- ▶ Q “predicted” (modeled) distribution with distribution function q

Entropy

$$H(p) = - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

Cross Entropy

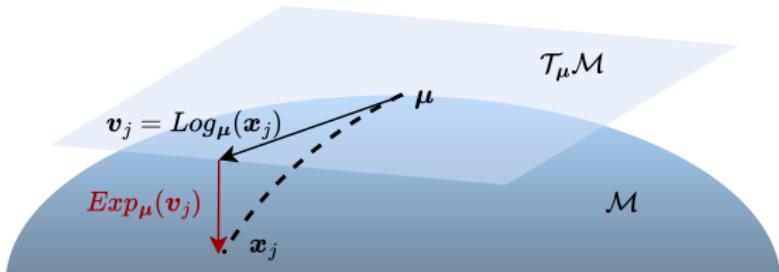
$$H(p, q) = - \sum_{x \in \mathcal{X}} p(x) \log q(x) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) + \text{KL}(p \parallel q)$$

Kullback–Leibler (KL) Divergence

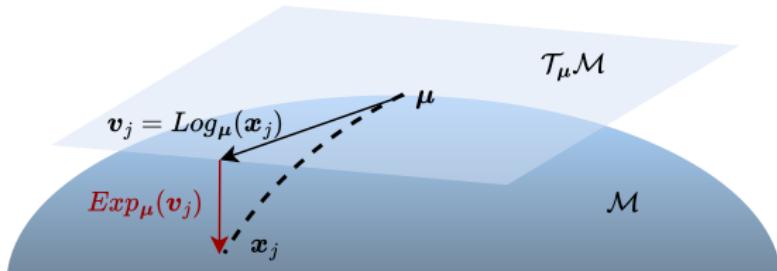
$$\text{KL}(p \parallel q) = H(p, q) - H(p) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \quad (6)$$

1. Determine two distributions to compare
2. Minimize KL divergence directly
3. With conditional distributions, max Evidence Lower BOund (ELBO)

Geometry-preserving

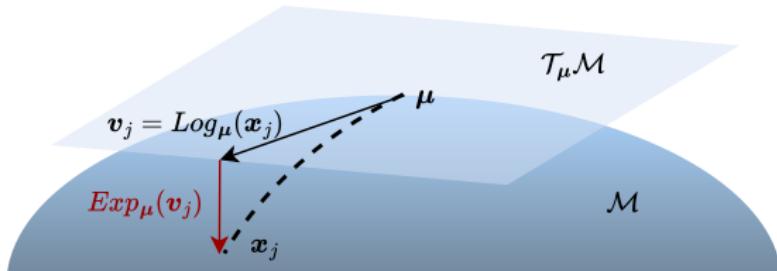


Geometry-preserving



Manifold Hypothesis: High-dimensional data lies (approximately) on a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$.

Geometry-preserving



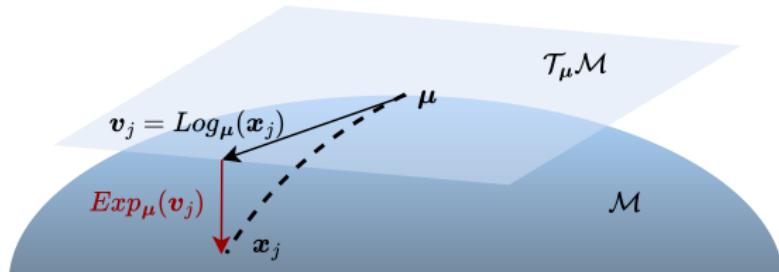
Manifold Hypothesis: High-dimensional data lies (approximately) on a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$.

Local Linearity: Around each $\mathbf{x}_i \in \mathcal{M}$, there exists a neighborhood U such that:

$$\mathcal{M} \cap U \approx T_{\mathbf{x}_i} \mathcal{M}$$

where $T_{\mathbf{x}_i} \mathcal{M}$ is the tangent space at \mathbf{x}_i .

Geometry-preserving



Manifold Hypothesis: High-dimensional data lies (approximately) on a low-dimensional manifold $\mathcal{M} \subset \mathbb{R}^d$.

Local Linearity: Around each $\mathbf{x}_i \in \mathcal{M}$, there exists a neighborhood U such that:

$$\mathcal{M} \cap U \approx T_{\mathbf{x}_i} \mathcal{M}$$

where $T_{\mathbf{x}_i} \mathcal{M}$ is the tangent space at \mathbf{x}_i .

Geodesic Distances: For $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{M}$, define

$$d_{\mathcal{M}}(\mathbf{x}_i, \mathbf{x}_j) = \text{length of shortest path on } \mathcal{M}$$

Improves global geometry preservation.

Topology-preserving

Goal: Dimensionality reduction often assumes a topology that determines local neighborhoods and connectivity.

Topology-preserving

Goal: Dimensionality reduction often assumes a topology that determines local neighborhoods and connectivity.

Data-driven topology (learned):

- ▶ Construct graph $G = (\mathcal{X}, E)$ from data.
- ▶ Edge weights: $w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right)$ or k -nearest neighbors.
- ▶ Preserve graph in low-dimensional representation

Topology-preserving

Goal: Dimensionality reduction often assumes a topology that determines local neighborhoods and connectivity.

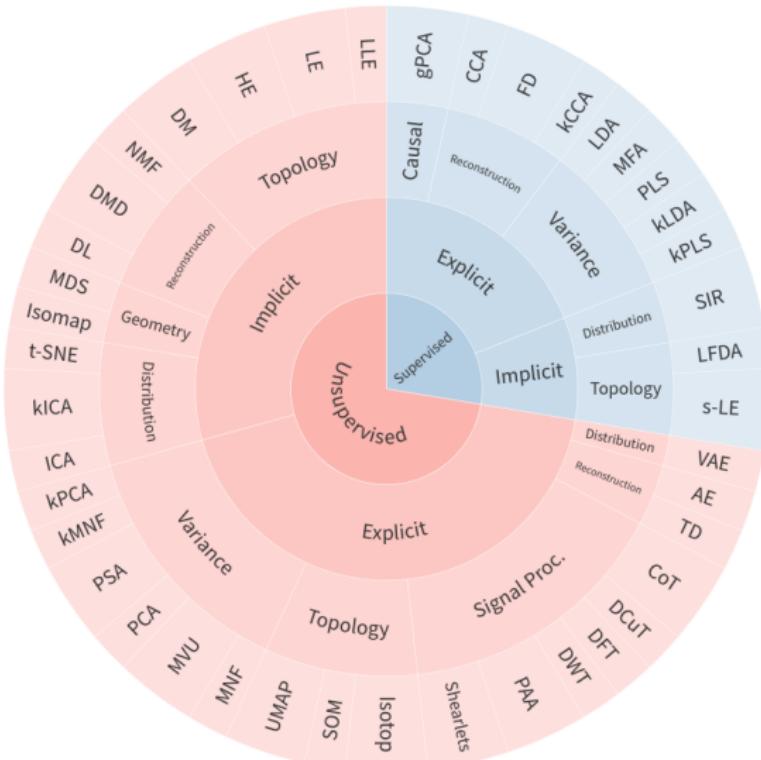
Data-driven topology (learned):

- ▶ Construct graph $G = (\mathcal{X}, E)$ from data.
- ▶ Edge weights: $w_{ij} = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right)$ or k -nearest neighbors.
- ▶ Preserve graph in low-dimensional representation

Predefined topology (fixed):

- ▶ Impose grid or lattice structure: \mathcal{G} = 1D or 2D lattice.
- ▶ Each reduce-space representation \mathbf{z}_i corresponds to a fixed node in \mathcal{G} .

Too Many DR Methods..



DR Methods to be Covered

DR Methods to be Covered

Method	Map		Data		Constr. & Prop. Pres.						Code	
	Exp.	Lin.	Sup.	M.F.	A.E.	Rec.	Var.	Dist.	Geo.	Top		
PCA	✓	✓	✗	✓	✗	✓	✓	✗	✗	✗	✗	sklearn
LDA	✓	✓	✓	✓	✗	✓	✓	✓	✗	✗	✗	sklearn
VAE	✓	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	github
MDS	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	sklearn
Isomap	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	sklearn
t-SNE	✗	✗	✗	✗	✗	✗	✓	✓	✗	✓	✓	sklearn
UMAP	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✓	sklearn

Linear Dimensionality Reduction

Outline

1. Principal Component Analysis Hotelling [1933]; Shlens [2014]
2. Linear Discriminant Analysis Tharwat et al. [2017]

Dataset

We consider a dataset of p samples with d features:

$$\{\mathbf{x}_i\}_{i=1}^p \subset \mathbb{R}^d$$

We collect the dataset into a matrix:

$$\mathbf{X} \in \mathbb{R}^{p \times d}$$

Important: The data must be **mean-centered**:

$$\frac{1}{p} \sum_{i=1}^p \mathbf{x}_i = \mathbf{0}$$

Optimization Goal

We aim to extract $k < p$ features that describe the directions of maximum variance.

We find the j th direction of maximum variance as

$$\mathbf{v}_j = \arg \max_{\substack{\mathbf{v}^\top \mathbf{v} = 1 \\ \mathbf{v}^\top \mathbf{v}_\ell = 0 \forall \ell < i}} \sum_i (\mathbf{v} \mathbf{x}_i)^\top (\mathbf{v} \mathbf{x}_i) \quad (7)$$

$$= \arg \max_{\substack{\mathbf{v}^\top \mathbf{v} = 1 \\ \mathbf{v}^\top \mathbf{v}_\ell = 0 \forall \ell < i}} \sum_i \|\mathbf{x}_i\|_2^2 \cos \theta(\mathbf{v}, \mathbf{x}_i) \quad (8)$$

$$= \arg \max_{\substack{\mathbf{v}^\top \mathbf{v} = 1 \\ \mathbf{v}^\top \mathbf{v}_\ell = 0 \forall \ell < i}} \mathbf{v}^\top \mathbf{X}^\top \mathbf{X} \mathbf{v} \quad (9)$$

Alternative Formulation: Reconstruction Error

This is equivalent to finding the rank- k projection:

$$\Pi_{\mathbf{W}}(\mathbf{x}) := \mathbf{W}\mathbf{W}^\top \mathbf{x}$$

We solve:

$$\mathbf{V} = \arg \min_{\mathbf{W}^\top \mathbf{W} = \mathbf{I}} \mathbb{E} [\|\mathbf{x}_i - \mathbf{W}\mathbf{W}^\top \mathbf{x}_i\|_2^2]$$

Exercise Show that this is an eigenvalue problem.

PCA Objective Summary

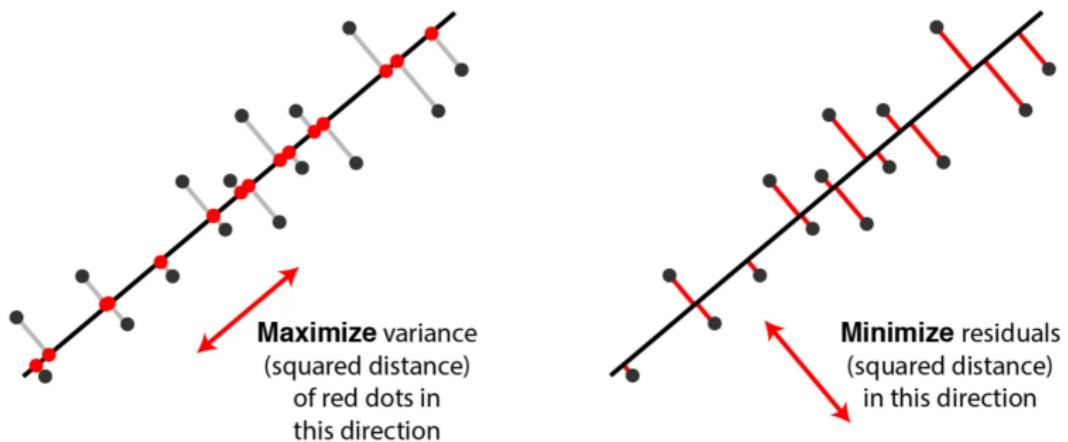


Figure: Image borrowed from <https://medium.com/@fraidoonmarzai99/principal-component-analysis-pca-in-depth-93c871f25dfa>.

Transformation

We call \mathbf{V} the matrix of **PCA weights** or **Empirical Orthogonal Functions (EOFs)**.

Transformation

We call \mathbf{V} the matrix of **PCA weights** or **Empirical Orthogonal Functions (EOFs)**.

1st EOF is direction of maximum variance, 2nd EOF in direction of maximum variance that is orthogonal to 1st EOF, and so on ...

Transformation

We call \mathbf{V} the matrix of **PCA weights** or **Empirical Orthogonal Functions (EOFs)**.

1st EOF is direction of maximum variance, 2nd EOF in direction of maximum variance that is orthogonal to 1st EOF, and so on ...

The first k principal components of \mathbf{X} :

$$\hat{\mathbf{X}} = \mathbf{X}\mathbf{V} \in \mathbb{R}^{p \times k}$$

PCA map of an individual sample:

$$\mathbf{z} = \phi(\mathbf{x}) = \mathbf{V}^\top \mathbf{x} \in \mathbb{R}^k$$

PCA reconstruction of an individual sample:

$$\hat{\mathbf{x}} = \psi \circ \phi(\mathbf{x}) = \mathbf{V}\mathbf{V}^\top \mathbf{x} = \mathbf{\Pi}_\mathbf{V}(\mathbf{x}) \in \mathbb{R}^n$$

Explained Variance

The **explained variance** is how much variance each principal component captures:

Variance of component i :

$$\lambda_i = \mathbf{v}_i^\top \mathbf{X}^\top \mathbf{X} \mathbf{v}_i$$

Explained variance ratio of component n :

$$\frac{\lambda_n}{\sum_{j=1}^n \lambda_j}$$

This helps select the number of components to keep...

Rule of thumb is select components that explain > 90% of variance.

When to Use PCA

- ▶ Have unlabeled data
- ▶ Want to reduce dimensionality while preserving variance
- ▶ Want a fast, interpretable linear projection
- ▶ Want to decorrelate features
- ▶ Few outliers

When to Use PCA

- ▶ Have unlabeled data
- ▶ Want to reduce dimensionality while preserving variance
- ▶ Want a fast, interpretable linear projection
- ▶ Want to decorrelate features
- ▶ Few outliers

Beyond PCA... robust subspace recovery Lerman and Maunu [2018] & flag manifolds Mankovich et al. [2024]; Szwagier and Pennec [2024, 2025]

Lab 3: PCA

Go to Lab3_PCA.ipynb

What is LDA?

Linear Discriminant Analysis (LDA) is a supervised dimensionality reduction technique.

Goal: Project high-dimensional data onto a lower-dimensional space that best separates multiple classes.

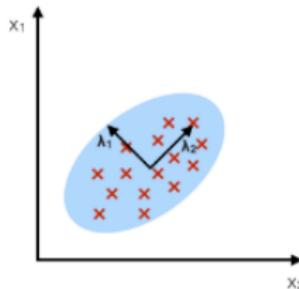
- ▶ Maximizes **between-class variance**
- ▶ Minimizes **within-class variance**

"PCA with class information!"

LDA Intuition

PCA:

component axes that maximize the variance



LDA:

maximizing the component axes for class-separation

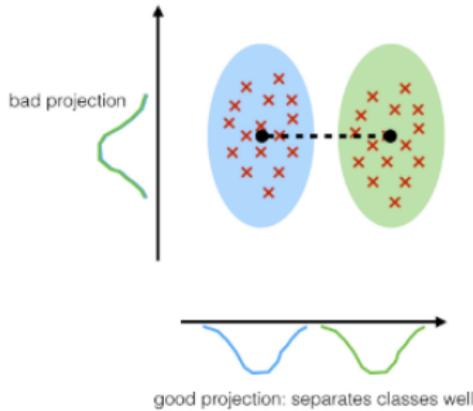


Figure: Image borrowed from

https://sebastianraschka.com/Articles/2014_python_lda.html

Mathematical Formulation

Given labeled dataset:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^p, \quad \mathbf{x}_i \in \mathbb{R}^d$$

Number of classes: C .

Within-class scatter matrix:

$$S_W = \sum_{c=1}^C \sum_{\mathbf{x}_i \in c} (\mathbf{x}_i - \mu_c)(\mathbf{x}_i - \mu_c)^\top$$

Between-class scatter matrix:

$$S_B = \sum_{c=1}^C (\mu_c - \mu)(\mu_c - \mu)^\top$$

Optimization Problem

We solve the following generalized eigenvalue problem:

$$\mathbf{W} = \arg \max_{\mathbf{W}} \frac{\text{tr}(\mathbf{W}^\top S_B \mathbf{W})}{\text{tr}(\mathbf{W}^\top S_W \mathbf{W})}$$

Output:

- ▶ $\mathbf{W} \in \mathbb{R}^{d \times k}$: DR mapping matrix
- ▶ $k \leq C - 1$: max number of discriminative components

Reduced data:

$$\mathbf{z}_i = \mathbf{W}^\top \mathbf{x}_i$$

When to Use LDA

- ▶ Labeled data
- ▶ Want low-dimensional features that separate classes well
- ▶ Want a fast, interpretable linear projection
- ▶ Linearly separable classes
- ▶ Each class follows a multivariate normal distribution with equal covariances

LDA vs PCA

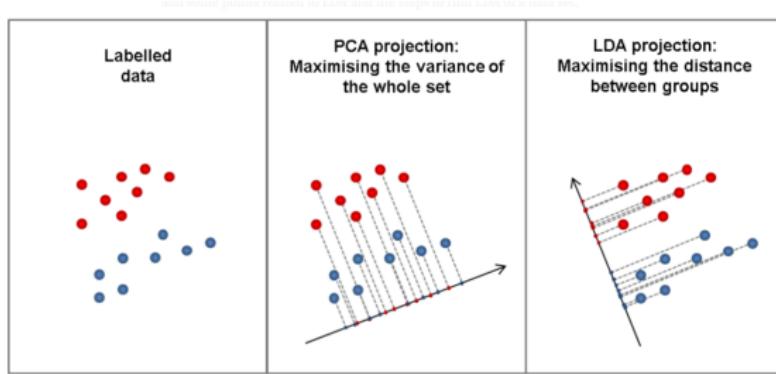


Figure: Image borrowed from <https://vivekmuraleedharan73.medium.com/what-is-linear-discriminant-analysis-lda-7e33ff59020a>.

Aspect	PCA Shlens [2014]	LDA Tharwat et al. [2017]
Supervised?	No	Yes
Objective	Maximize variance	Maximize class separation
Axes chosen	max variance	best separation
Max dimensions	\leq input dimension	\leq number of classes - 1

Lab 3: LDA

Go to Lab4_LDA.ipynb

References I

- Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.
- John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.
- Gilad Lerman and Tyler Maunu. An overview of robust subspace recovery. *Proceedings of the IEEE*, 106(8):1380–1410, 2018.
- Nathan Mankovich, Gustau Camps-Valls, and Tolga Birdal. Fun with flags: Robust principal directions via flag manifolds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 330–340, 2024.
- Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.
- Gilbert Strang. *Linear algebra and its applications*. 2000.

References II

- Tom Szwagier and Xavier Pennec. The curse of isotropy: From principal components to principal subspaces. 2024. URL <https://arxiv.org/abs/2307.15348>.
- Tom Szwagier and Xavier Pennec. Nested subspace learning with flags. *arXiv preprint arXiv:2502.06022*, 2025.
- Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI communications*, 30(2):169–190, 2017.