

# 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

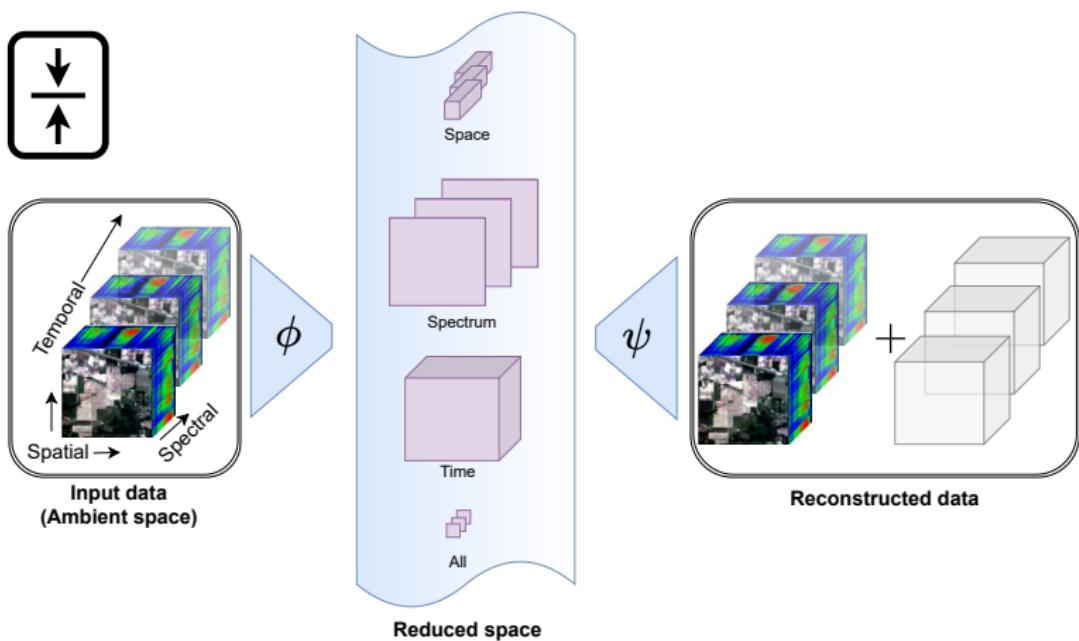
Input a data cube with spectral, spatial, and temporal dimensions.

# Dimensionality Reduction in Remote Sensing

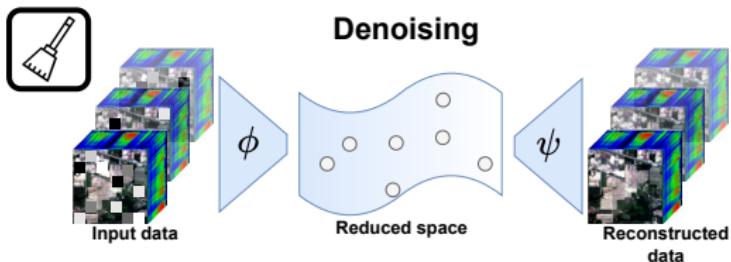
Input a data cube with spectral, spatial, and temporal dimensions.

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

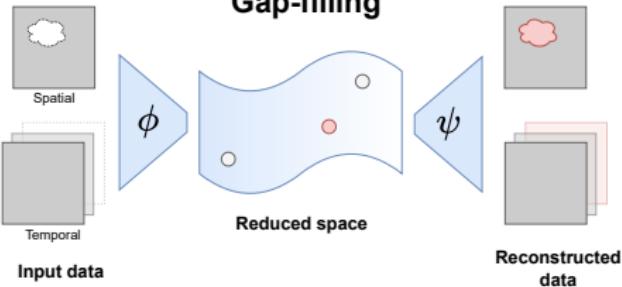
# Compression



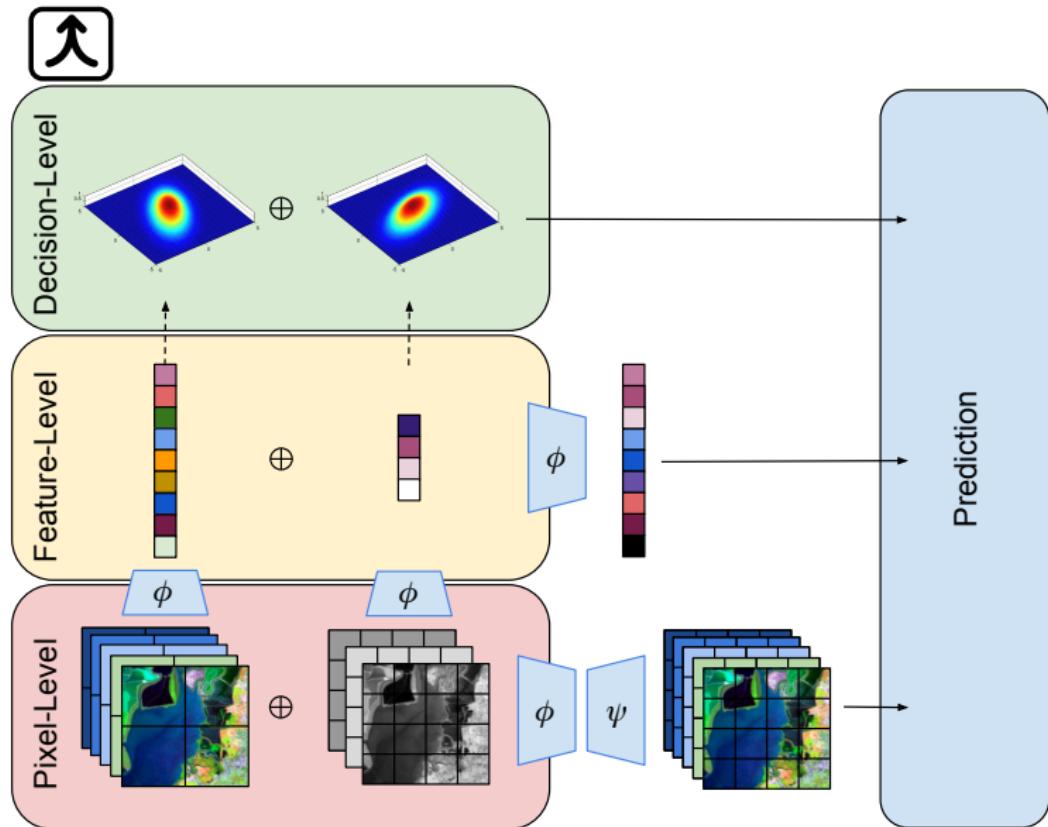
# Denoising



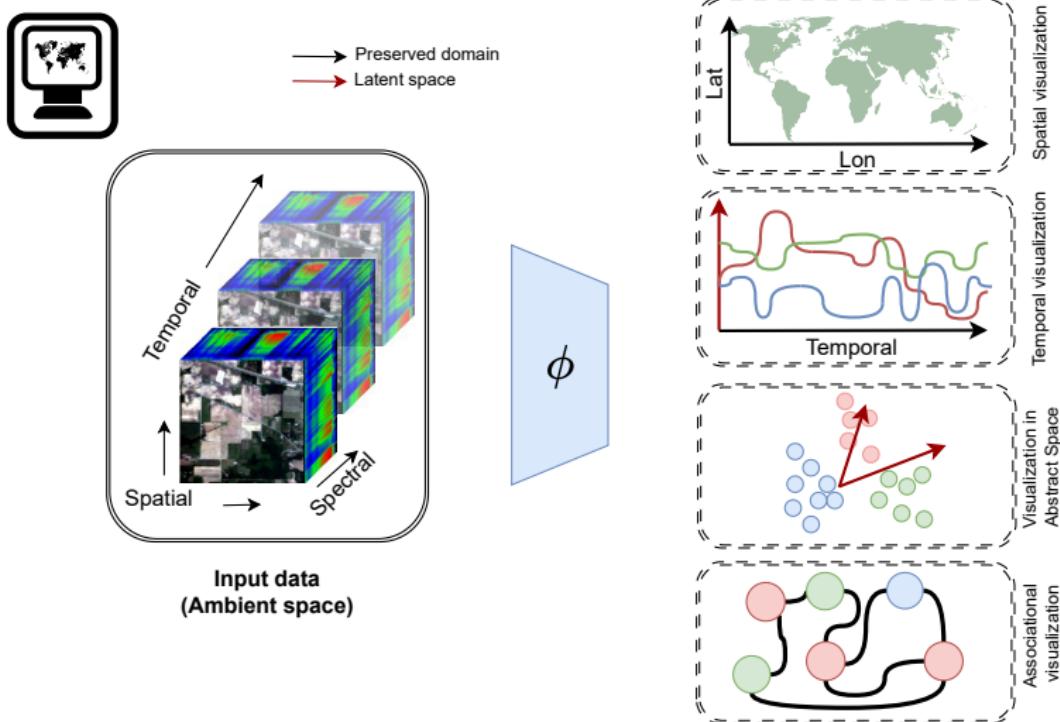
# Gap-filling



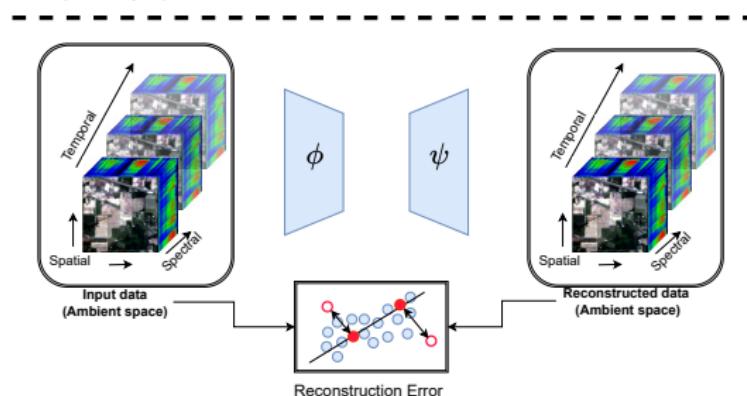
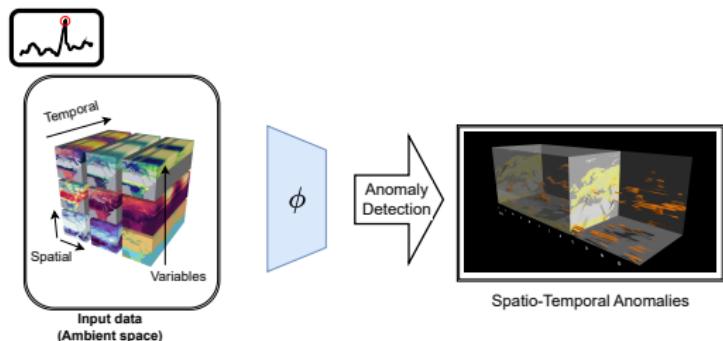
# Fusion



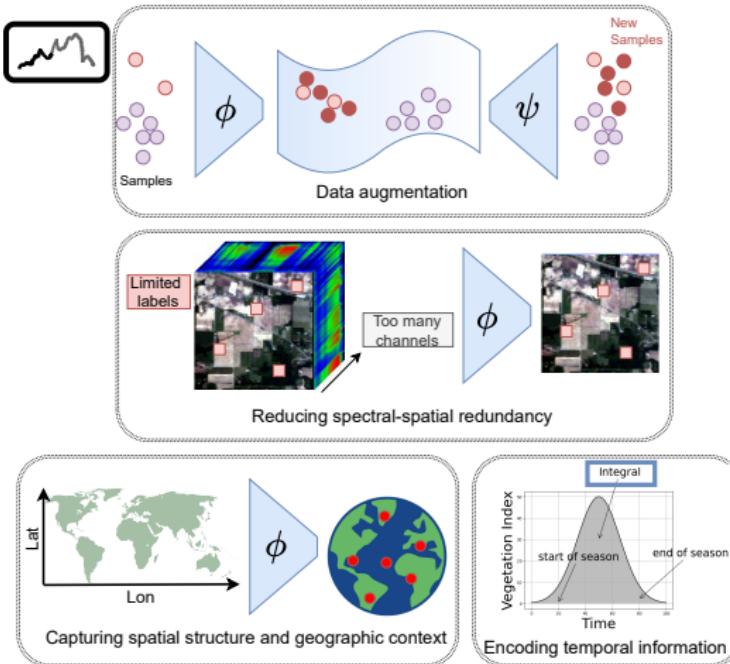
## Visualization



# Anomaly Detection



# Improved Predictions

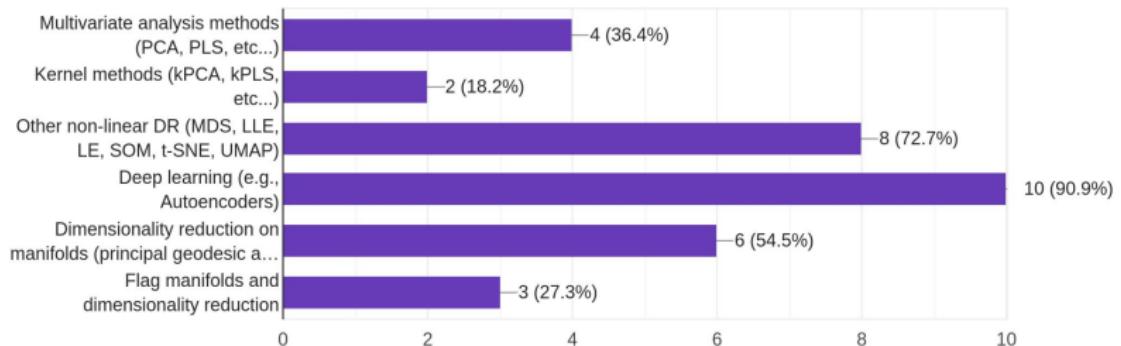


# Course Survey Results

# Survey Results

## Topics covered (choose up to 3)

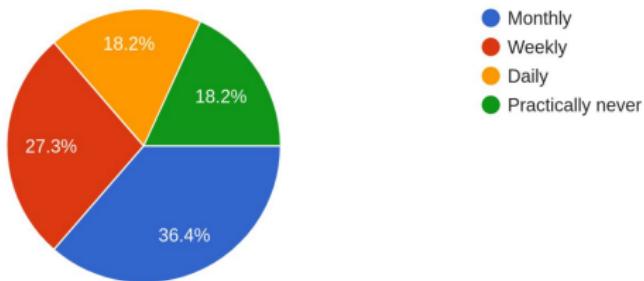
11 responses



# Survey Results

## How often do you use linear algebra

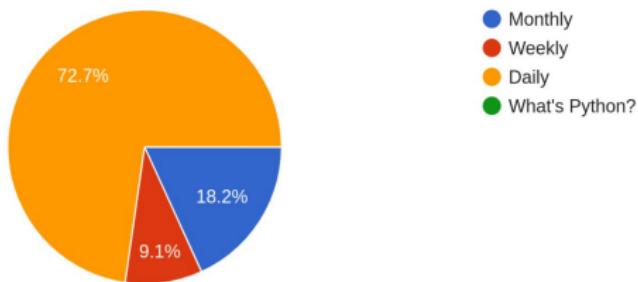
11 responses



# 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. Matrix Decompositions
4. Gradient Descent

# Vectors & Inner Products

## 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  $\mathbb{R}$ .

$$\mathbb{R}^2 = \left\{ \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} : a_1, a_2 \in \mathbb{R} \right\}$$

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  $V$ .

# Subspaces, Orthogonality, Projections

# Matrices

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

# Matrices

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

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

# Matrices

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

- ▶ 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

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

- ▶ 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

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

- ▶ 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

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

- ▶ 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

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

# Matrix multiplication

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

# 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  $\mathbf{A}\mathbf{A}^T$ .

# Lab 1: Vectors & Matrices

*Go to Lab1\_VectorsMatrices.ipynb*

# Matrix Decompositions

# 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  $\Sigma$

# 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  $\Sigma$

*Exercises:*

- ▶ Show the eigenvalue decomposition of  $\mathbf{A}\mathbf{A}^\top$  is  $\mathbf{U}\Sigma\Sigma^\top\mathbf{U}^\top$
- ▶ 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:

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

Condition number:

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

Big  $\kappa$  bad, Small  $\kappa$  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:

- ▶  $\theta$  are the model parameters

# Update Rule

# Update Rule

At each iteration, the parameters are updated as:

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

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

# 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."*

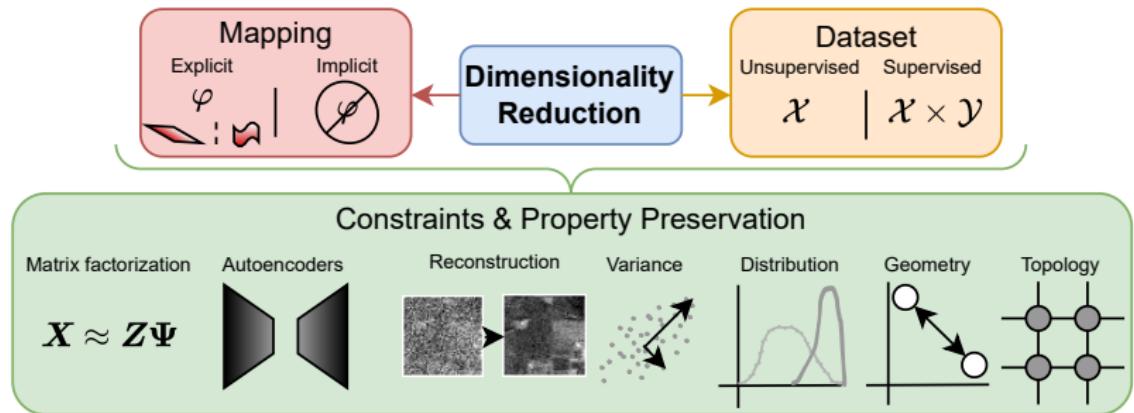
— **Dr. Max E. Stress**, Professor of Reduced Expectations

---

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  $\phi$  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:**  $\phi, \psi$  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

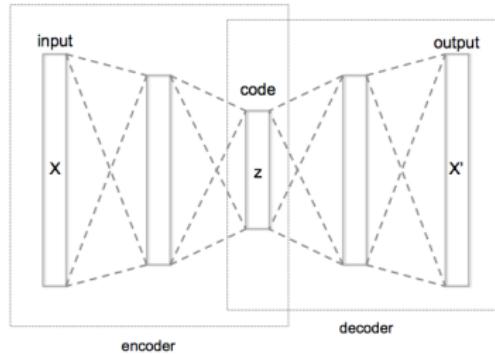


Figure: Image from  
[https://commons.wikimedia.org/wiki/File:Autoencoder\\_structure.png](https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png)

# Autoencoders

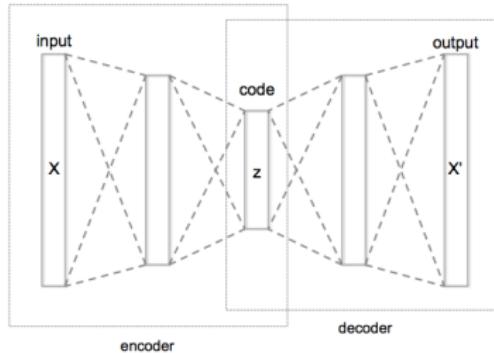


Figure: Image from  
[https://commons.wikimedia.org/wiki/File:Autoencoder\\_structure.png](https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png)

- Neural networks trained to reconstruct input

# Autoencoders

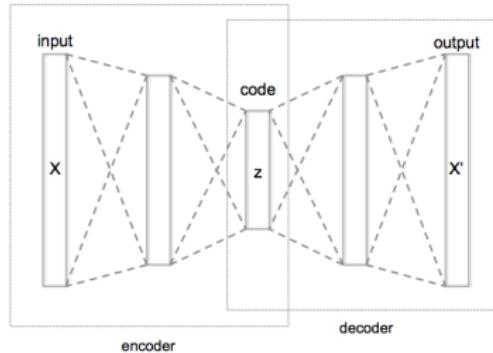


Figure: Image from  
[https://commons.wikimedia.org/wiki/File:Autoencoder\\_structure.png](https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png)

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

# Autoencoders

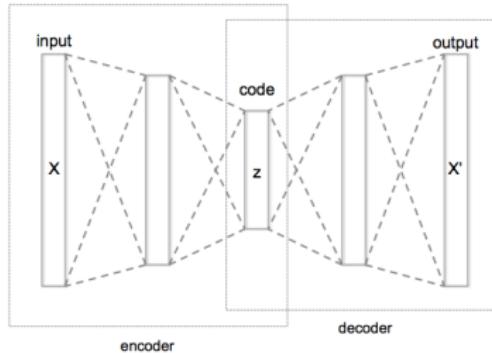


Figure: Image from

[https://commons.wikimedia.org/wiki/File:Autoencoder\\_structure.png](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

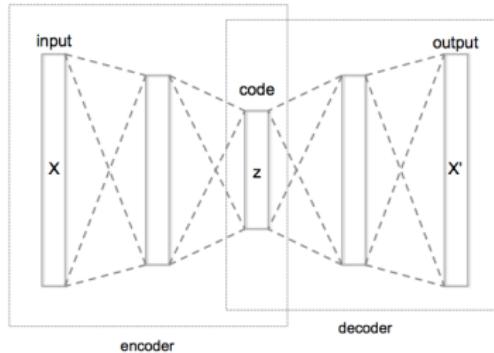


Figure: Image from  
[https://commons.wikimedia.org/wiki/File:Autoencoder\\_structure.png](https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png)

- ▶ 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

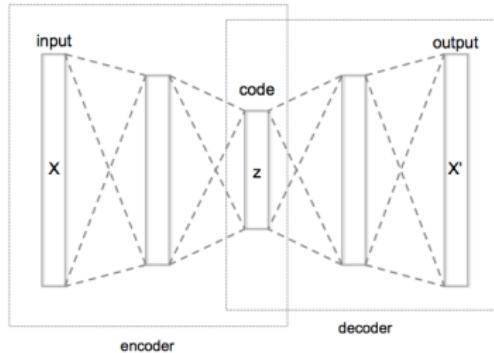


Figure: Image from  
[https://commons.wikimedia.org/wiki/File:Autoencoder\\_structure.png](https://commons.wikimedia.org/wiki/File:Autoencoder_structure.png)

- ▶ 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  $\mathbf{x}$  and DR mapping  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^k$

## Variance-preserving

Data  $\mathbf{x}$  and DR mapping  $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^k$

$$\phi = \arg \max \mathbb{E}[\phi(\mathbf{x})^\top \phi(\mathbf{x})] \quad (5)$$

# 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

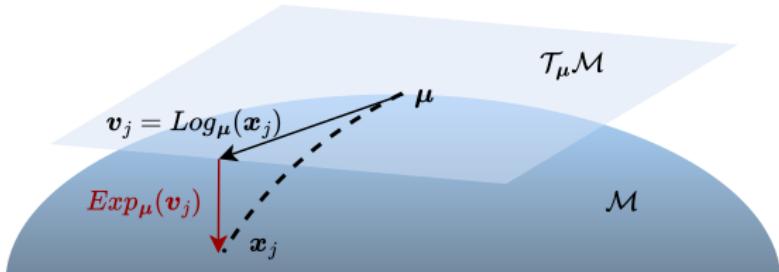
$$H(p, q) = - \sum_{x \in \mathcal{X}} p(x) \log q(x) = - \sum_{x \in \mathcal{X}} p(x) \log p(x) + \text{KL}(p \parallel q)$$

## Kullback–Leibler (KL) Divergence

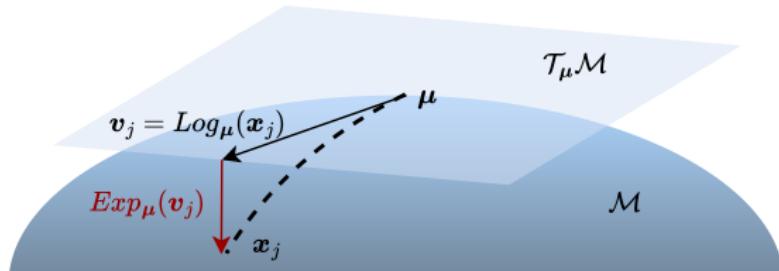
$$\text{KL}(p \parallel q) = H(p, q) - H(p) = \sum_{x \in \mathcal{X}} p(x) \log \frac{p(x)}{q(x)} \quad (6)$$

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

# Geometry-preserving

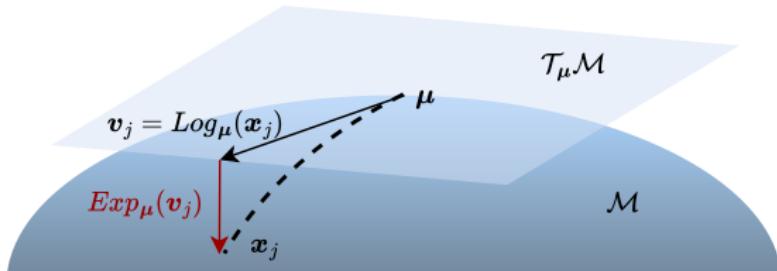


# Geometry-preserving



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

# Geometry-preserving



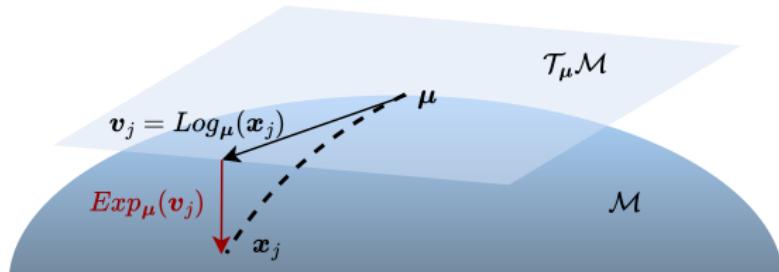
**Manifold Hypothesis:** High-dimensional data lies (approximately) on a low-dimensional manifold  $\mathcal{M} \subset \mathbb{R}^d$ .

**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.

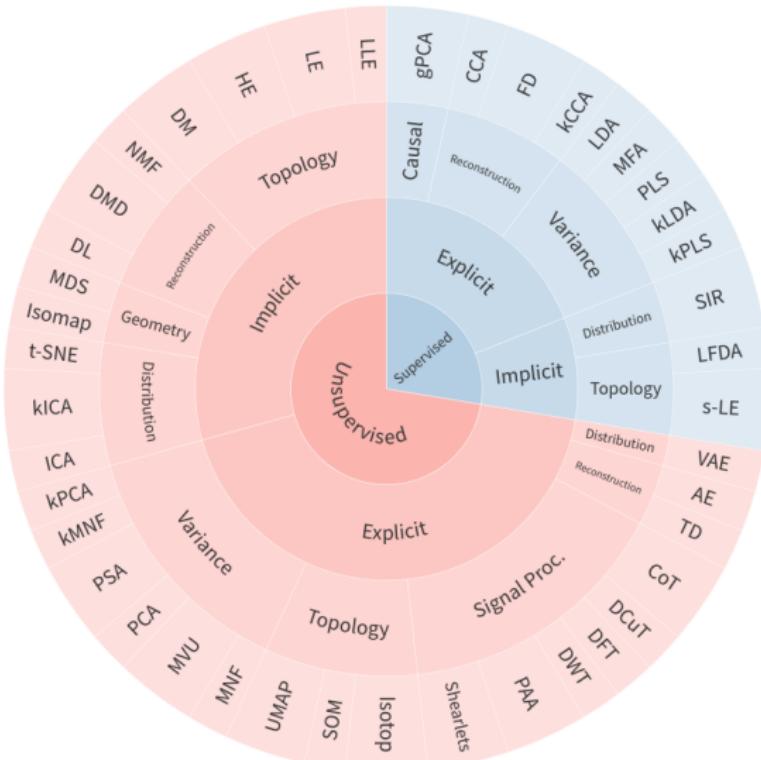
## Data-driven topology (learned):

- ▶ 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..



# Linear Dimensionality Reduction

# Outline

1. Principal Component Analysis Hotelling [1933]; Shlens [2014]
2. Linear Discriminant Analysis Tharwat et al. [2017]
3. Dynamic Mode Decomposition Schmid [2010]

# Principal Component Analysis

# 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

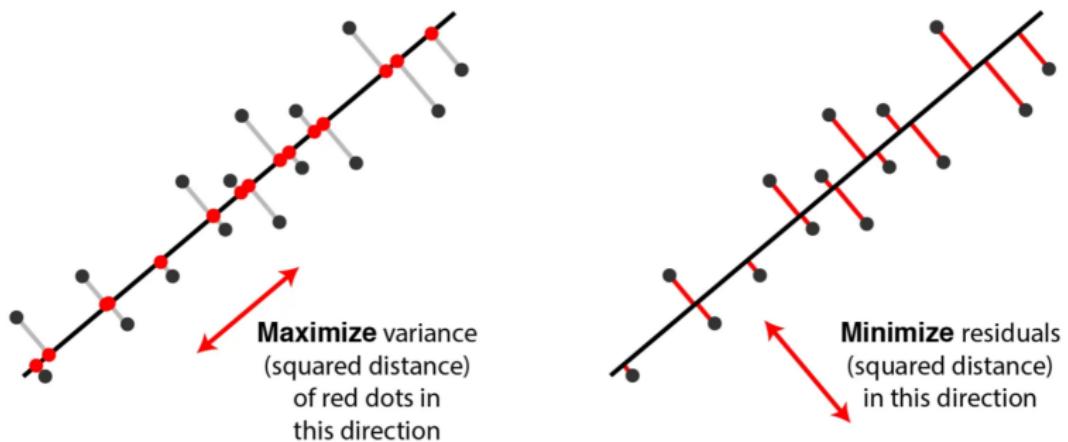


Figure: Image borrowed from <https://medium.com/@fraidoonmarzai99/principal-component-analysis-pca-in-depth-93c871f25dfa>.

## Transformation

We call **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  $\mathbf{X}$ :

# 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  $\mathbf{X}$ :

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

# 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  $\mathbf{X}$ :

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

PCA map of an individual sample:

# 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  $\mathbf{X}$ :

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

PCA map of an individual sample:

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

# 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  $\mathbf{X}$ :

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

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:

# 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  $\mathbf{X}$ :

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

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:

$$\hat{\mathbf{x}} = \psi \circ \phi(\mathbf{x}) = \mathbf{V}\mathbf{V}^\top \mathbf{x} = \Pi_{\mathbf{V}}(\mathbf{x}) \in \mathbb{R}^d$$

## Explained Variance

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

## Explained Variance

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

Variance of component  $i$ :

# 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

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  $i$ :

# 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  $n$ :

$$\frac{\lambda_n}{\sum_{j=1}^d \lambda_j} = \frac{\lambda_n}{\text{tr}(\mathbf{X}^\top \mathbf{X})}$$

# 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  $n$ :

$$\frac{\lambda_n}{\sum_{j=1}^d \lambda_j} = \frac{\lambda_n}{\text{tr}(\mathbf{X}^\top \mathbf{X})}$$

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

# 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  $n$ :

$$\frac{\lambda_n}{\sum_{j=1}^d \lambda_j} = \frac{\lambda_n}{\text{tr}(\mathbf{X}^\top \mathbf{X})}$$

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

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

# 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  $n$ :

$$\frac{\lambda_n}{\sum_{j=1}^d \lambda_j} = \frac{\lambda_n}{\text{tr}(\mathbf{X}^\top \mathbf{X})}$$

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

*Rule of thumb is select components that explain > 90% of variance. (but there's other methods Gavish and Donoho [2014])*

# When to Use PCA

# When to Use PCA

- ▶ Have unlabeled data

## When to Use PCA

- ▶ Have unlabeled data
- ▶ Want to reduce dimensionality while preserving variance

# When to Use PCA

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

# When to Use PCA

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

# 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]

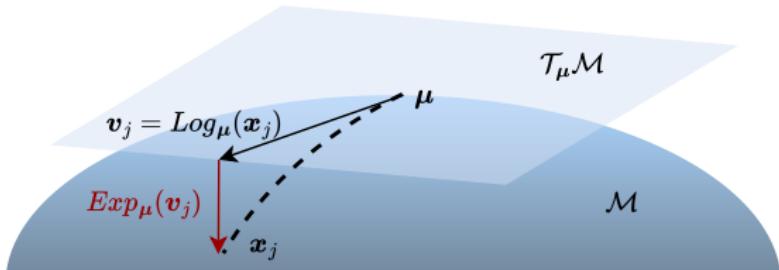
## Aside- PCA on Manifolds (Tangent-PCA)

## Aside- PCA on Manifolds (Tangent-PCA)

Assume data samples from known, Riemannian manifold...

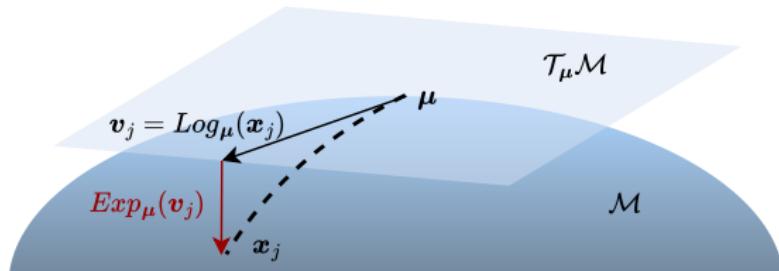
# Aside- PCA on Manifolds (Tangent-PCA)

Assume data samples from known, Riemannian manifold...



## Aside- PCA on Manifolds (Tangent-PCA)

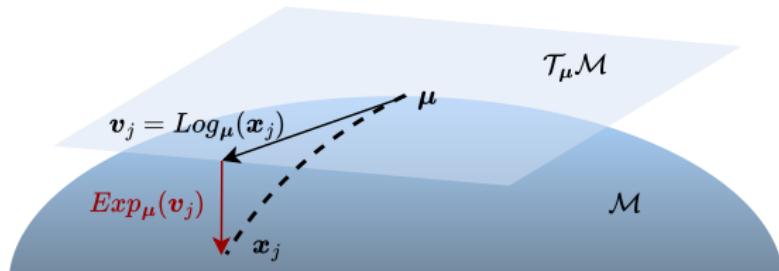
Assume data samples from known, Riemannian manifold...



- ▶ Map data to tangent space

## Aside- PCA on Manifolds (Tangent-PCA)

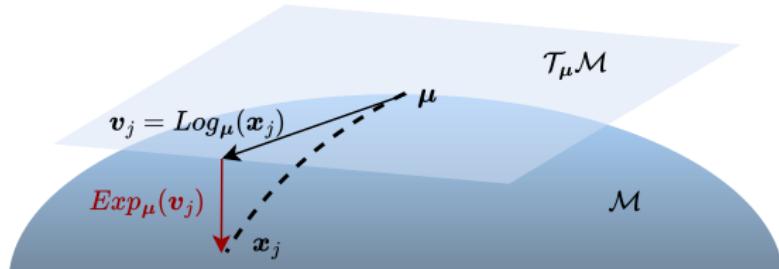
Assume data samples from known, Riemannian manifold...



- ▶ Map data to tangent space
- ▶ Run PCA in tangent space

# Aside- PCA on Manifolds (Tangent-PCA)

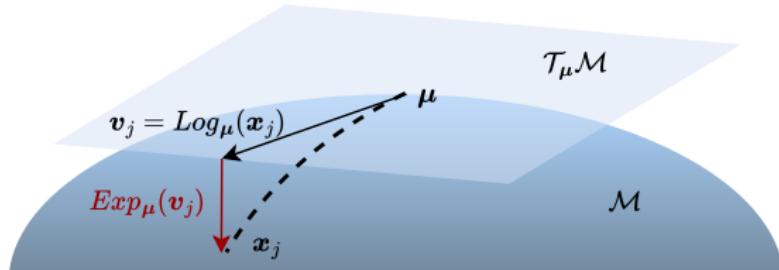
Assume data samples from known, Riemannian manifold...



- ▶ Map data to tangent space
- ▶ Run PCA in tangent space
- ▶ Map back to manifold

# Aside- PCA on Manifolds (Tangent-PCA)

Assume data samples from known, Riemannian manifold...



- ▶ Map data to tangent space
- ▶ Run PCA in tangent space
- ▶ Map back to manifold

Read more here: Fletcher et al. [2004]; Pennec [2018]

# Lab 3: PCA

*Go to Lab3\_PCA.ipynb*

# Linear Discriminant Analysis

# What is LDA?

# What is LDA?

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

# 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.

# 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**

# 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**

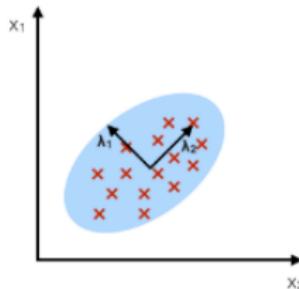
*"PCA with class information!"*

# LDA Intuition

# LDA Intuition

## PCA:

component axes that maximize the variance



## LDA:

maximizing the component axes for class-separation

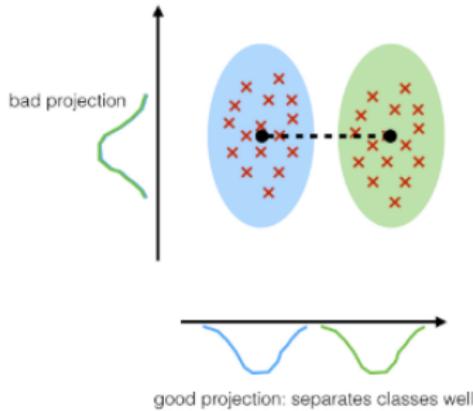


Figure: Image borrowed from

[https://sebastianraschka.com/Articles/2014\\_python\\_lda.html](https://sebastianraschka.com/Articles/2014_python_lda.html)

# Mathematical Formulation

## Mathematical Formulation

Given labeled dataset:

# 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:  $C$ .

# 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:**

$$\mathbf{S}_W = \sum_{c=1}^C \sum_{\mathbf{x}_i \in c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^\top$$

# 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:**

$$\mathbf{S}_W = \sum_{c=1}^C \sum_{\mathbf{x}_i \in c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^\top$$

**Between-class scatter matrix:**

$$\mathbf{S}_B = \sum_{c=1}^C (\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^\top$$

# Optimization Problem

# Optimization Problem

We solve the following generalized eigenvalue problem:

# Optimization Problem

We solve the following generalized eigenvalue problem:

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

# 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:**

# 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

# 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

# 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

# When to Use LDA

- ▶ Labeled data

## When to Use LDA

- ▶ Labeled data
- ▶ Want low-dimensional features that separate classes well

## When to Use LDA

- ▶ Labeled data
- ▶ Want low-dimensional features that separate classes well
- ▶ Want a fast, interpretable linear projection

## When to Use LDA

- ▶ Labeled data
- ▶ Want low-dimensional features that separate classes well
- ▶ Want a fast, interpretable linear projection
- ▶ Linearly separable classes

# 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

# LDA vs PCA

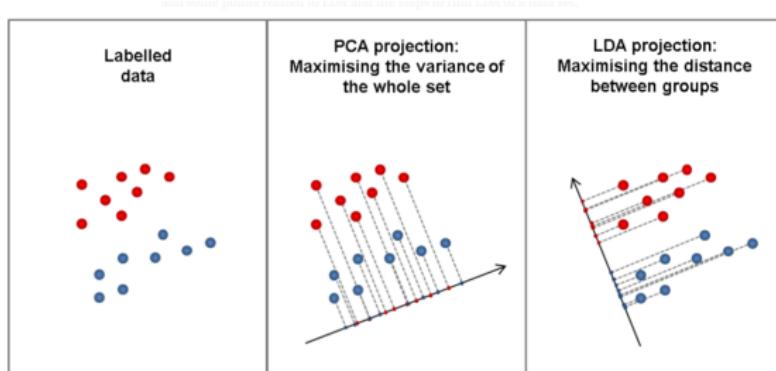


Figure: Image borrowed from <https://vivekmuraleedharan73.medium.com/what-is-linear-discriminant-analysis-lda-7e33ff59020a>.

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*

# Dynamic Mode Decomposition

# Dynamical Systems

# Dynamical Systems

A **dynamical system** describes how a state evolves over time according to a fixed rule.

# Dynamical Systems

A **dynamical system** describes how a state evolves over time according to a fixed rule.

Discrete-time:  $\mathbf{x}(t + \tau) = f(\mathbf{x}(t))$

Continuous-time:  $\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t))$

# Dynamical Systems

A **dynamical system** describes how a state evolves over time according to a fixed rule.

Discrete-time:  $\mathbf{x}(t + \tau) = f(\mathbf{x}(t))$

Continuous-time:  $\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t))$

- ▶  $\mathbf{x}(t) \in \mathbb{R}^n$ : the state at time  $t$

# Dynamical Systems

A **dynamical system** describes how a state evolves over time according to a fixed rule.

Discrete-time:  $\mathbf{x}(t + \tau) = f(\mathbf{x}(t))$

Continuous-time:  $\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t))$

- ▶  $\mathbf{x}(t) \in \mathbb{R}^n$ : the state at time  $t$
- ▶  $f, g$ : a (possibly nonlinear) update rule

# Dynamical Systems

A **dynamical system** describes how a state evolves over time according to a fixed rule.

Discrete-time:  $\mathbf{x}(t + \tau) = f(\mathbf{x}(t))$

Continuous-time:  $\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t))$

- ▶  $\mathbf{x}(t) \in \mathbb{R}^n$ : the state at time  $t$
- ▶  $f, g$ : a (possibly nonlinear) update rule
- ▶ Goal: Understand, predict, or control the evolution of  $\mathbf{x}(t)$

# Dynamical Systems

A **dynamical system** describes how a state evolves over time according to a fixed rule.

Discrete-time:  $\mathbf{x}(t + \tau) = f(\mathbf{x}(t))$       Continuous-time:  $\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t))$

- ▶  $\mathbf{x}(t) \in \mathbb{R}^n$ : the state at time  $t$
- ▶  $f, g$ : a (possibly nonlinear) update rule
- ▶ Goal: Understand, predict, or control the evolution of  $\mathbf{x}(t)$

## Linear Systems

If  $f(\mathbf{x}) = A\mathbf{x}$ , the system is linear:

$$\mathbf{x}(t + \tau) = A\mathbf{x}(t)$$

# Dynamical Systems

A **dynamical system** describes how a state evolves over time according to a fixed rule.

Discrete-time:  $\mathbf{x}(t + \tau) = f(\mathbf{x}(t))$       Continuous-time:  $\frac{d\mathbf{x}}{dt} = g(\mathbf{x}(t))$

- ▶  $\mathbf{x}(t) \in \mathbb{R}^n$ : the state at time  $t$
- ▶  $f, g$ : a (possibly nonlinear) update rule
- ▶ Goal: Understand, predict, or control the evolution of  $\mathbf{x}(t)$

## Linear Systems

If  $f(\mathbf{x}) = A\mathbf{x}$ , the system is linear:

$$\mathbf{x}(t + \tau) = A\mathbf{x}(t)$$

⇒ Solutions evolve through powers of  $A$ : eigenvalues/eigenvectors govern behavior.

# Dynamic Mode Decomposition

# Dynamic Mode Decomposition

Assume that the data is sampled from the timeseries

$$\mathbf{x}(t + \tau) \approx \mathbf{A} \mathbf{x}(t).$$

# Dynamic Mode Decomposition

Assume that the data is sampled from the timeseries

$$\mathbf{x}(t + \tau) \approx \mathbf{A} \mathbf{x}(t).$$

**Decompose the system** into spatial and temporal patterns.

# Dynamic Mode Decomposition

Assume that the data is sampled from the timeseries

$$\mathbf{x}(t + \tau) \approx \mathbf{A}\mathbf{x}(t).$$

**Decompose the system** into spatial and temporal patterns.

Analyzing  $\mathbf{A}$  results in the DMD Schmid [2010]

$$\mathbf{x}(t) = \sum_{j=1}^k \phi_j e^{\omega_j t} b_j.$$

# Dynamic Mode Decomposition

Assume that the data is sampled from the timeseries

$$\mathbf{x}(t + \tau) \approx \mathbf{A}\mathbf{x}(t).$$

**Decompose the system** into spatial and temporal patterns.

Analyzing  $\mathbf{A}$  results in the DMD Schmid [2010]

$$\mathbf{x}(t) = \sum_{j=1}^k \phi_j e^{\omega_j t} b_j.$$

- ▶  $\phi_j \in \mathbb{C}^n$  **feature patterns** (dynamic modes)
- ▶  $\omega_j \in \mathbb{C}$  **temporal characteristics** (continuous time eigenvalues)
- ▶  $b_j \in \mathbb{R}$  scalar loadings (a.k.a. amplitudes)

# Eigenvalue Interpretation

# Eigenvalue Interpretation

$$\lambda_j = e^{\omega_j \tau}$$

- ▶ Discrete time  $\lambda_j$
- ▶ Continuous time  $\omega_j$

# Eigenvalue Interpretation

$$\lambda_j = e^{\omega_j \tau}$$

- ▶ Discrete time  $\lambda_j$
- ▶ Continuous time  $\omega_j$

Write in polar form

$$\lambda_j = r e^{i\theta}$$

# Eigenvalue Interpretation

$$\lambda_j = e^{\omega_j \tau}$$

- ▶ Discrete time  $\lambda_j$
- ▶ Continuous time  $\omega_j$

Write in polar form

$$\lambda_j = r e^{i\theta}$$

- ▶ Trend:  $r = |\lambda_j|$

# Eigenvalue Interpretation

$$\lambda_j = e^{\omega_j \tau}$$

- Discrete time  $\lambda_j$
- Continuous time  $\omega_j$

Write in polar form

$$\lambda_j = r e^{i\theta}$$

- Trend:  $r = |\lambda_j|$
- Oscillation frequency (per  $\tau$ ):  $\theta = -i \log(\lambda_j/r)$

# Eigenvalue Interpretation

$$\lambda_j = e^{\omega_j \tau}$$

- Discrete time  $\lambda_j$
- Continuous time  $\omega_j$

Write in polar form

$$\lambda_j = r e^{i\theta}$$

- Trend:  $r = |\lambda_j|$
- Oscillation frequency (per  $\tau$ ):  $\theta = -i \log(\lambda_j/r)$

# Exact Dynamic Mode Decomposition Dawson et al. [2016]

# Exact Dynamic Mode Decomposition Dawson et al. [2016]

## 1. Stack the data

$$\mathbf{X} = [\mathbf{x}(1) | \cdots | \mathbf{x}(T - \tau)] \in \mathbb{R}^{n \times p}, \quad \mathbf{X}' = [\mathbf{x}(1 + \tau) | \cdots | \mathbf{x}(T)] \in \mathbb{R}^{n \times p},$$

# Exact Dynamic Mode Decomposition Dawson et al. [2016]

1. Stack the data

$$\mathbf{X} = [\mathbf{x}(1) | \cdots | \mathbf{x}(T - \tau)] \in \mathbb{R}^{n \times p}, \quad \mathbf{X}' = [\mathbf{x}(1 + \tau) | \cdots | \mathbf{x}(T)] \in \mathbb{R}^{n \times p},$$

2. Want to solve

$$\min_{\mathbf{A}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F$$

# Exact Dynamic Mode Decomposition Dawson et al. [2016]

1. Stack the data

$$\mathbf{X} = [\mathbf{x}(1) | \cdots | \mathbf{x}(T-\tau)] \in \mathbb{R}^{n \times p}, \quad \mathbf{X}' = [\mathbf{x}(1+\tau) | \cdots | \mathbf{x}(T)] \in \mathbb{R}^{n \times p},$$

2. Want to solve

$$\min_{\mathbf{A}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F$$

3. Rank- $r$  truncated SVD  $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$  leads to

$$\tilde{\mathbf{A}} = \mathbf{U}^\top \mathbf{X}' \mathbf{V} \Sigma^{-1} \in \mathbb{R}^{k \times k}, \quad \tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \Lambda, \quad (\text{eigendecomposition})$$

# Exact Dynamic Mode Decomposition Dawson et al. [2016]

1. Stack the data

$$\mathbf{X} = [\mathbf{x}(1) | \cdots | \mathbf{x}(T-\tau)] \in \mathbb{R}^{n \times p}, \quad \mathbf{X}' = [\mathbf{x}(1+\tau) | \cdots | \mathbf{x}(T)] \in \mathbb{R}^{n \times p},$$

2. Want to solve

$$\min_{\mathbf{A}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F$$

3. Rank- $r$  truncated SVD  $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$  leads to

$$\tilde{\mathbf{A}} = \mathbf{U}^\top \mathbf{X}' \mathbf{V} \Sigma^{-1} \in \mathbb{R}^{k \times k}, \quad \tilde{\mathbf{A}} \mathbf{W} = \mathbf{W} \Lambda, \quad (\text{eigendecomposition})$$

4. Dynamic modes

$$\phi_j = \frac{1}{\lambda_j} \mathbf{X}' \mathbf{V} \Sigma^{-1} \mathbf{w}_j$$

# Exact Dynamic Mode Decomposition Dawson et al. [2016]

1. Stack the data

$$\mathbf{X} = [\mathbf{x}(1) | \cdots | \mathbf{x}(T-\tau)] \in \mathbb{R}^{n \times p}, \quad \mathbf{X}' = [\mathbf{x}(1+\tau) | \cdots | \mathbf{x}(T)] \in \mathbb{R}^{n \times p},$$

2. Want to solve

$$\min_{\mathbf{A}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F$$

3. Rank- $r$  truncated SVD  $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$  leads to

$$\tilde{\mathbf{A}} = \mathbf{U}^\top \mathbf{X}' \mathbf{V}\Sigma^{-1} \in \mathbb{R}^{k \times k}, \quad \tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda, \quad (\text{eigendecomposition})$$

4. Dynamic modes

$$\phi_j = \frac{1}{\lambda_j} \mathbf{X}' \mathbf{V}\Sigma^{-1} \mathbf{w}_j$$

5. Eigenvalues (discrete time)

$$\lambda_j = e^{\omega_j \tau}$$

# Exact Dynamic Mode Decomposition Dawson et al. [2016]

1. Stack the data

$$\mathbf{X} = [\mathbf{x}(1) | \cdots | \mathbf{x}(T-\tau)] \in \mathbb{R}^{n \times p}, \quad \mathbf{X}' = [\mathbf{x}(1+\tau) | \cdots | \mathbf{x}(T)] \in \mathbb{R}^{n \times p},$$

2. Want to solve

$$\min_{\mathbf{A}} \|\mathbf{X}' - \mathbf{A}\mathbf{X}\|_F$$

3. Rank- $r$  truncated SVD  $\mathbf{X} = \mathbf{U}\Sigma\mathbf{V}^\top$  leads to

$$\tilde{\mathbf{A}} = \mathbf{U}^\top \mathbf{X}' \mathbf{V}\Sigma^{-1} \in \mathbb{R}^{k \times k}, \quad \tilde{\mathbf{A}}\mathbf{W} = \mathbf{W}\Lambda, \quad (\text{eigendecomposition})$$

4. Dynamic modes

$$\phi_j = \frac{1}{\lambda_j} \mathbf{X}' \mathbf{V}\Sigma^{-1} \mathbf{w}_j$$

5. Eigenvalues (discrete time)

$$\lambda_j = e^{\omega_j \tau}$$

6. Loadings found by solving

$$\Phi \mathbf{b} = \mathbf{x}(1).$$

# My Interpretation

## My Interpretation

The system evolves in the low-dimensional reduced space mapped to by  
 $\mathbf{U}^T \in \mathbb{R}^{k \times n}$

## My Interpretation

The system evolves in the low-dimensional reduced space mapped to by  $\mathbf{U}^T \in \mathbb{R}^{k \times n}$

$$\mathbf{U}^T \mathbf{x}(t + \tau) = \mathbf{z}(t + \tau) = \tilde{\mathbf{A}}\mathbf{z}(t) = \tilde{\mathbf{A}}\mathbf{U}^T \mathbf{x}(t). \quad (10)$$

## My Interpretation

The system evolves in the low-dimensional reduced space mapped to by  $\mathbf{U}^T \in \mathbb{R}^{k \times n}$

$$\mathbf{U}^T \mathbf{x}(t + \tau) = \mathbf{z}(t + \tau) = \tilde{\mathbf{A}}\mathbf{z}(t) = \tilde{\mathbf{A}}\mathbf{U}^T \mathbf{x}(t). \quad (10)$$

Projected dynamic modes:

$$\phi_j = \mathbf{U}\mathbf{w}_j$$

## My Interpretation

The system evolves in the low-dimensional reduced space mapped to by  $\mathbf{U}^T \in \mathbb{R}^{k \times n}$

$$\mathbf{U}^T \mathbf{x}(t + \tau) = \mathbf{z}(t + \tau) = \tilde{\mathbf{A}}\mathbf{z}(t) = \tilde{\mathbf{A}}\mathbf{U}^T \mathbf{x}(t). \quad (10)$$

Projected dynamic modes:

$$\phi_j = \mathbf{U}\mathbf{w}_j$$

DMD analyzes the stability of this system and generates spatial patterns in  $\mathbb{R}^n$  (ambient space)

# Using Dynamic Mode Decomposition

# Using Dynamic Mode Decomposition

- ▶ Spatial & temporal patterns of timeseries

# Using Dynamic Mode Decomposition

- ▶ Spatial & temporal patterns of timeseries
- ▶ Ill-conditioning of  $\mathbf{X}$  and  $\mathbf{A}$

# Using Dynamic Mode Decomposition

- ▶ Spatial & temporal patterns of timeseries
- ▶ Ill-conditioning of  $\mathbf{X}$  and  $\mathbf{A}$
- ▶ Not robust to noise

## References I

- Scott TM Dawson, Maziar S Hemati, Matthew O Williams, and Clarence W Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57:1–19, 2016.
- P Thomas Fletcher, Conglin Lu, Stephen M Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE transactions on medical imaging*, 23(8):995–1005, 2004.
- Matan Gavish and David L Donoho. The optimal hard threshold for singular values is  $4/\sqrt{3}$ . *IEEE Transactions on Information Theory*, 60(8):5040–5053, 2014.
- 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.

## References II

- 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.
- Xavier Pennec. Barycentric subspace analysis on manifolds. *The Annals of Statistics*, 2018.
- Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.
- Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.
- Gilbert Strang. *Linear algebra and its applications*. 2000.
- 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.

## References III

Alaa Tharwat, Tarek Gaber, Abdelhameed Ibrahim, and Aboul Ella Hassanien. Linear discriminant analysis: A detailed tutorial. *AI communications*, 30(2):169–190, 2017.