CZ4041/SC4000: Machine Learning

Lesson 12: Dimensionality Reduction

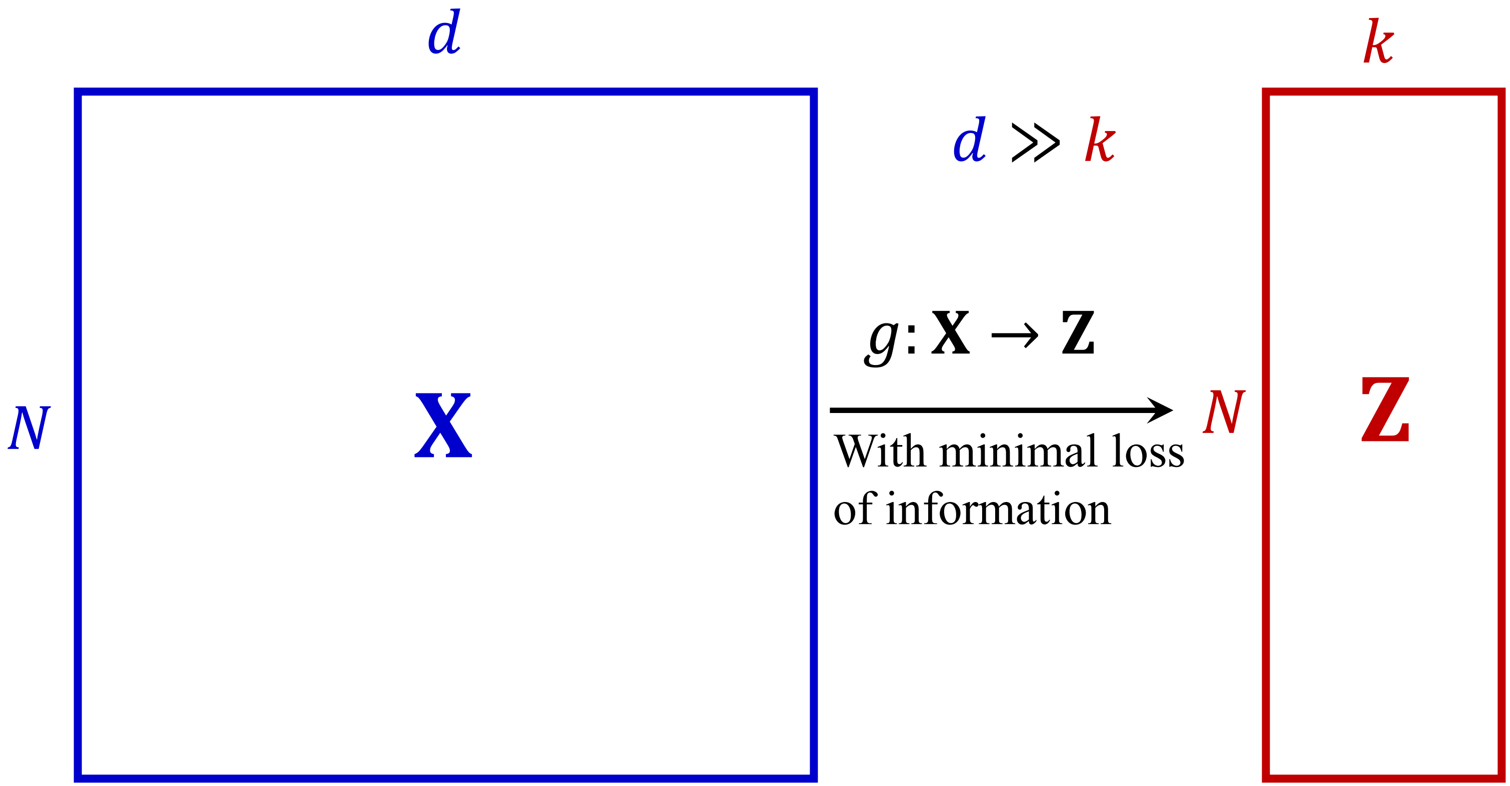
Li Boyang, Albert

School of Computer Science and Engineering, NTU, Singapore

Acknowledgements: some content is adapted from the lecture notes of Xiaojin Zhu @University of Wisconsin–Madison. Slides are modified from the version prepared by Dr. Sinno Pan.

High-level Idea

- To summarize observed high-dimensional data points with low-dimensional vectors



Why Dimensionality Reduction

- To avoid curse of dimensionality
  - Decision boundary in SVM: $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$
  - Linear Regression: $f(x) = \boldsymbol{w} \cdot \boldsymbol{x}$
  - Perceptron: $f(x) = \text{sign}(\boldsymbol{w} \cdot \boldsymbol{x})$
  - One parameter to learn for every input dimension
  - Difficult to accurately estimate the best parameters when the number of data points is small

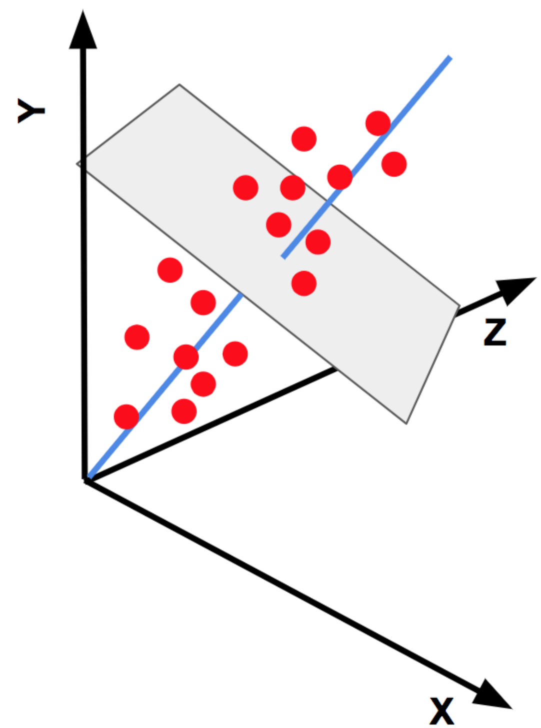
Why Dimensionality Reduction

- To identify the features, or the transformations of features that capture the most important data characteristics
- All measurements contain error or noise. Mathematically we may write

$$\boldsymbol{x} = \widetilde{\boldsymbol{x}} + \boldsymbol{e}, \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma} I)$$

Why Dimensionality Reduction

- All measurements contain error or noise
  $$\boldsymbol{x} = \widetilde{\boldsymbol{x}} + \boldsymbol{e}, \boldsymbol{e} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\sigma} I)$$
- Valid hypothesis: Data reside on a straight line, but the noise is isotropic (equal amount of variation) in all three dimensions.

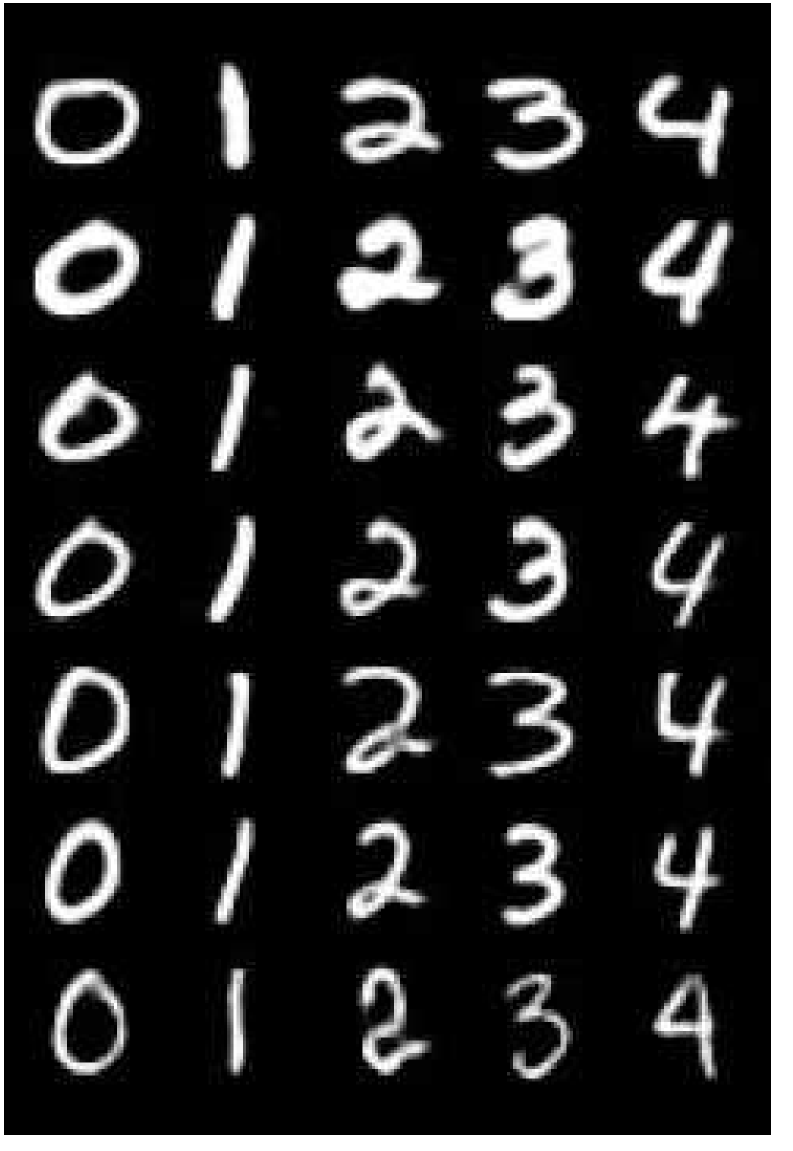


What is noise?

- These photos depict the same person
- Different lighting, make-up, facial hair, expressions, etc., create visual differences
- The data variation that we care about is far smaller than the pixel-level differences.

What is noise?

- The MNIST dataset
- Each column shows the same number.
- Some noise, which we do not have names for, change how they look.
- The data variation that we care about is far smaller than the pixel-level differences

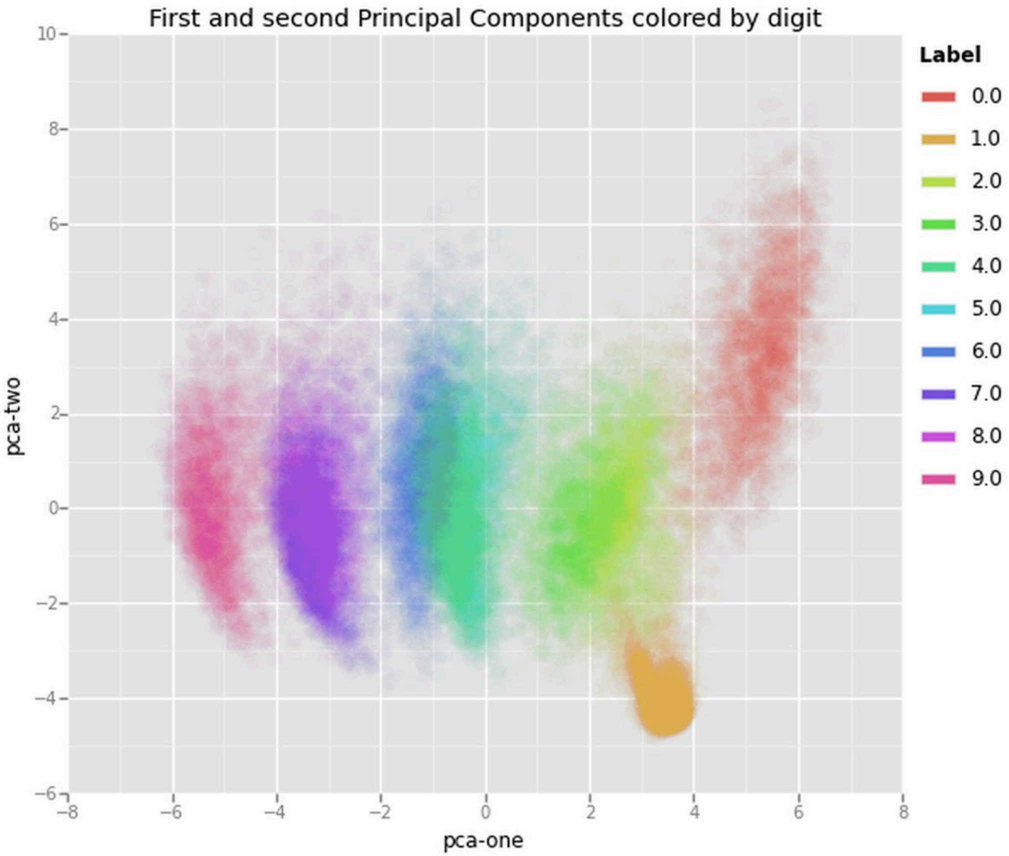


Why Dimensionality Reduction

- Thus, we should identify important variations in the data and discard the noise.
- Benefits
  - Reduces storage requirements
  - Allows visualization in 2D or 3D
  - Reduces noise and improve the performance of machine learning

A Case Study: MNIST

- We use PCA to reduce the number of dimensions to 2 and visualize the results.
- Some clustering structure



https://luckylwk.github.io/2015/09/13/visualising-mnist-pca-tsne/

Dimensionality Reduction Approaches

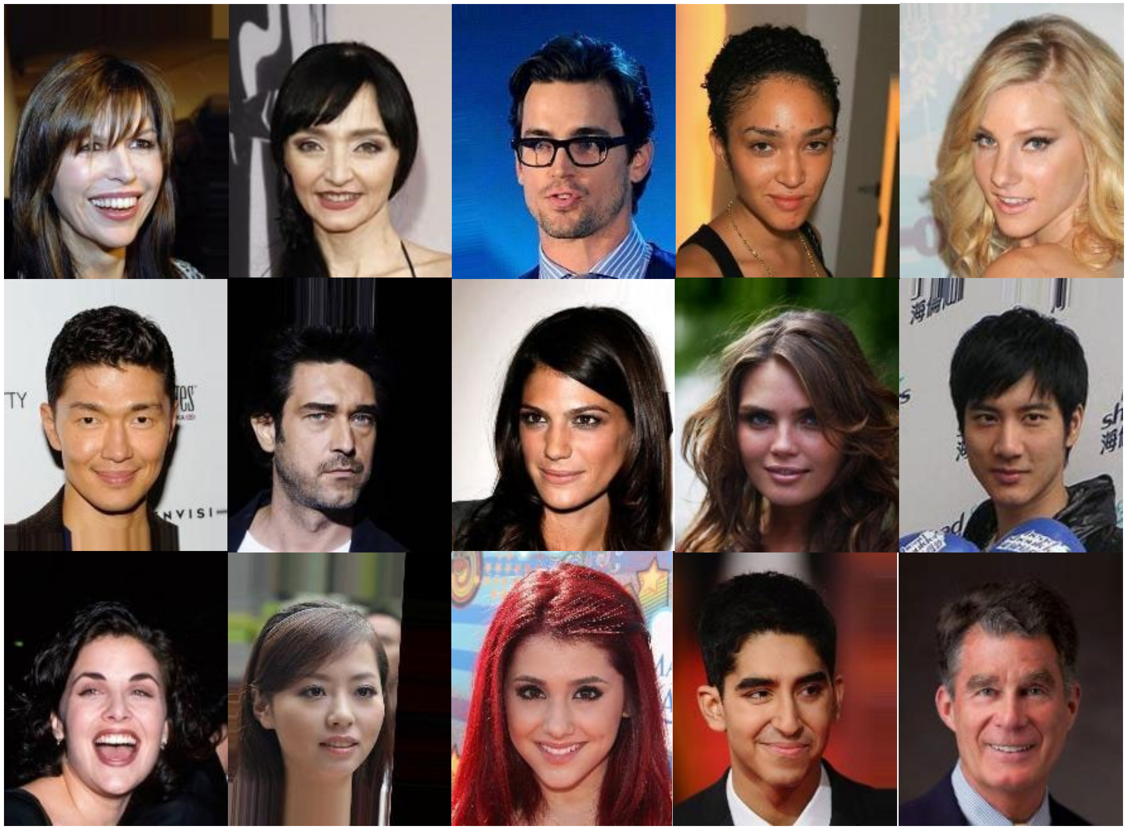
- Feature Selection
  - To select a subset of k features from the original d features to represent each data instance
    - Brute-force approach
    - Greedy search
- Feature Extraction
  - To learn k new features from the original d features to represent each data instance
    - Linear combination of original features
      - Principal component analysis
    - Nonlinear combination of original features

Principal Component Analysis

- One of the most widely-used (unsupervised) dimensionality reduction methods
- Takes a data matrix of N data points by d features, and summarizes it by principal components that are linear combinations of the original d variables
- The first k components display as much as possible of the variation among data instances

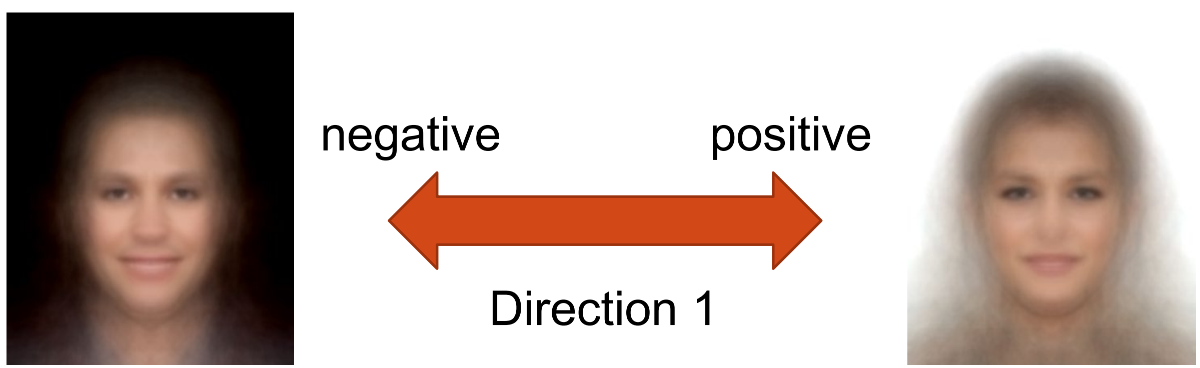
A Case Study: Eigenface

- PCA on images of human faces
- A technique initially created for face identification.



A Case Study: Eigenface

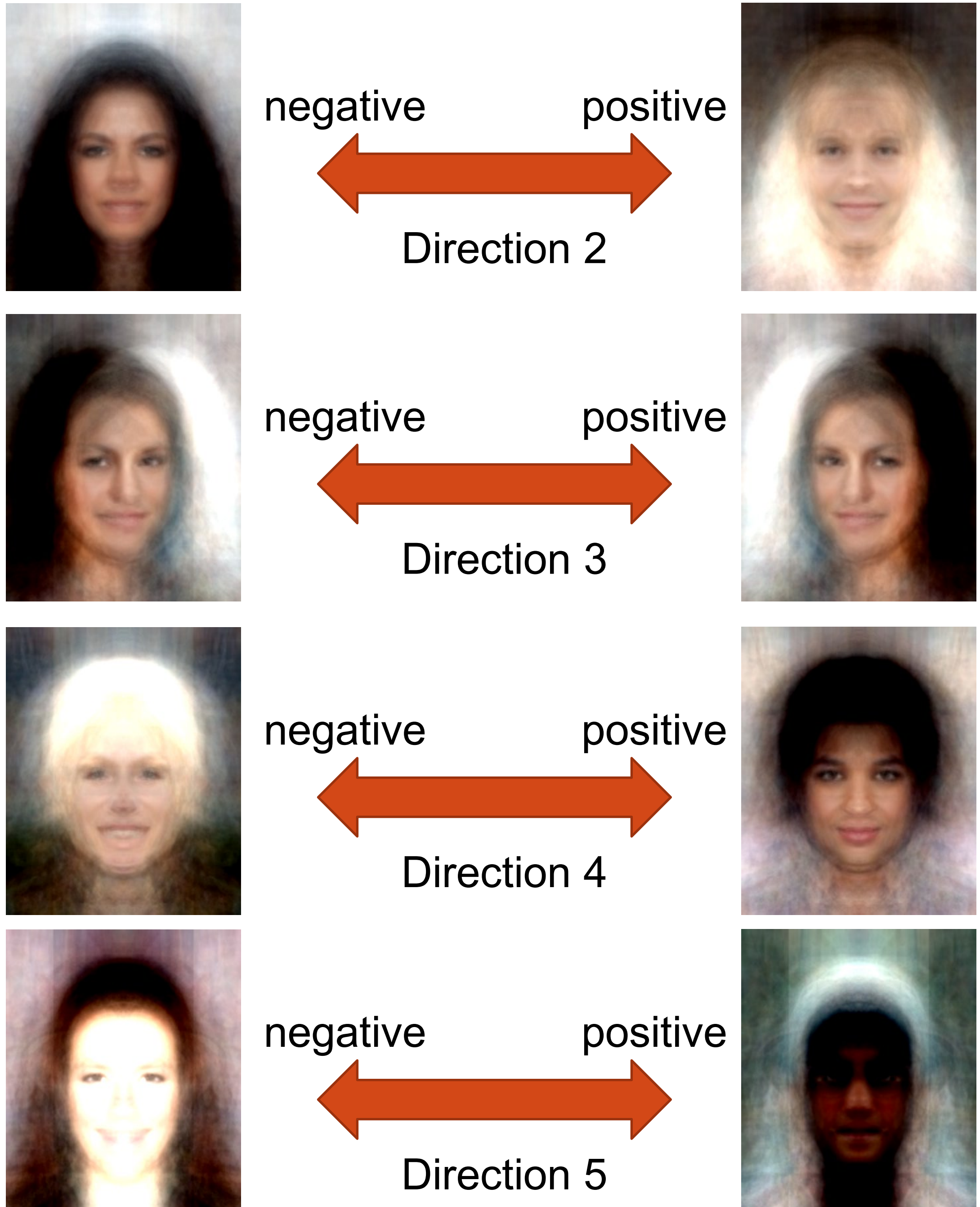
- We use PCA to identify 10 directions that capture most variations in the data.
- We visualize them by adding each direction to the average face, without which the faces are hard to interpret and look scary.
- The first direction seems to be mostly about lighting and background.

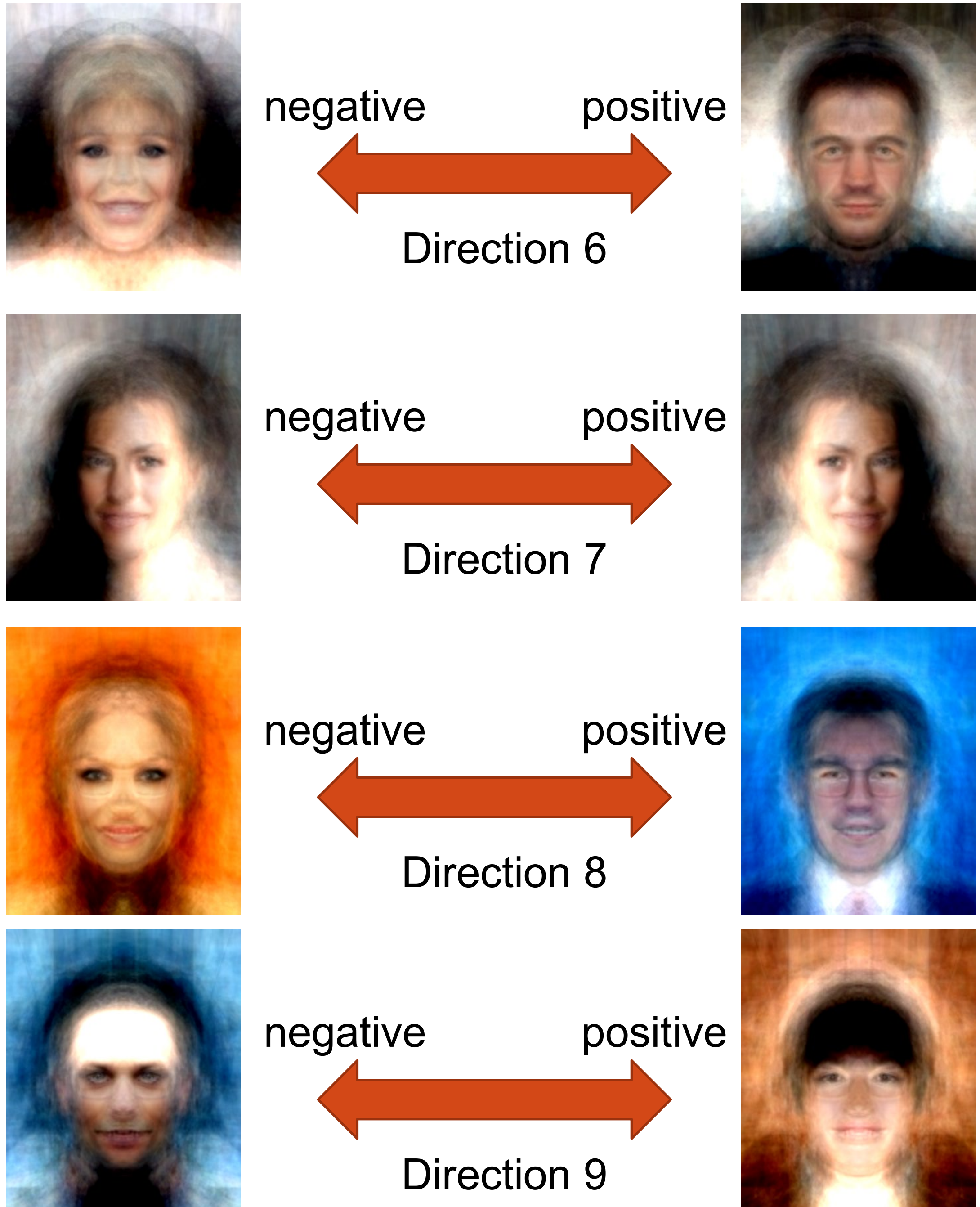


Long dark hair

Right facing

Light-colored
hair or bald



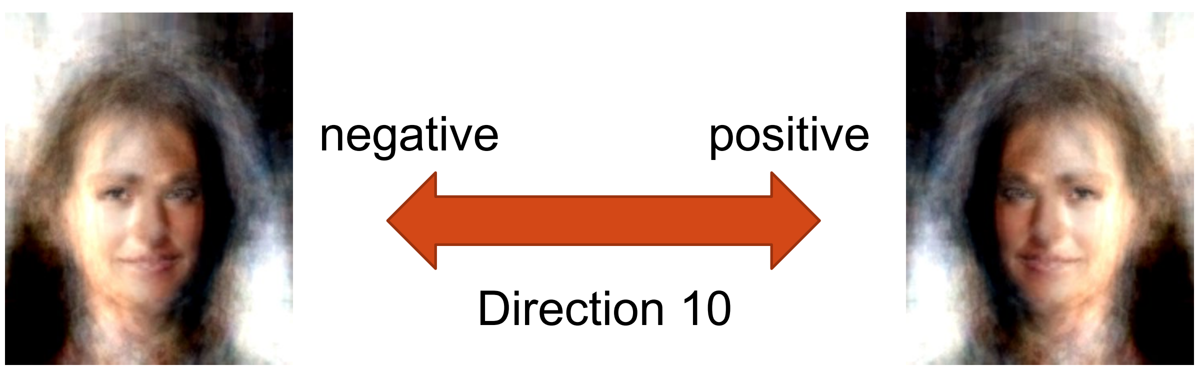


Background + suit?

Light above eyebrows or below chin

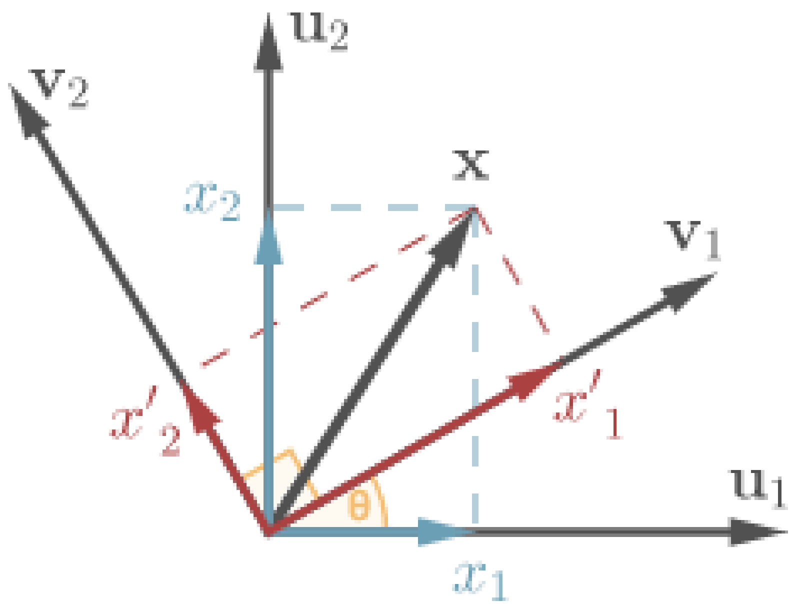
A Case Study: Eigenface

- PCA directions capture variations in data, which correspond to some semantic categories.
- But they are not always interpretable.
- PCA is linear. It would struggle to capture semantic categories that require non-linear variations, such as formal vs. casual dress.



Linear Algebra: Change of Basis

- Vector $\boldsymbol{x} = (x_1, x_2) = x_1\boldsymbol{u}_1 + x_2\boldsymbol{u}_2$ where $\boldsymbol{u}_1 = (1, 0)$ and $\boldsymbol{u}_2 = (0, 1)$ are the basis vectors.
- Can we change the basis vectors to $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$?
- $\boldsymbol{x} = x'_1\boldsymbol{v}_1 + x'_2\boldsymbol{v}_2$. We just need to find x'_1 and x'_2 by projecting $\boldsymbol{x}$ onto $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

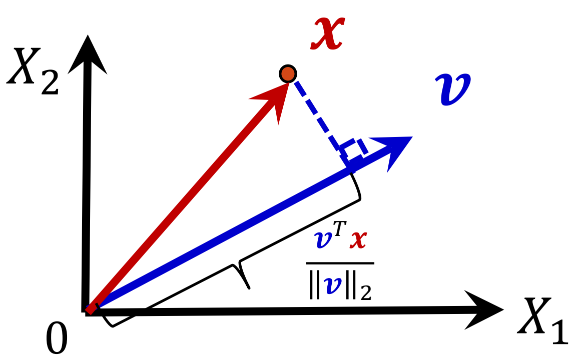


Linear Algebra: Change of Basis

- Consider a projection of a data point $\boldsymbol{x}$ onto a vector $\boldsymbol{v}$
- The projection of $\boldsymbol{x}$ onto $\boldsymbol{v}$ is

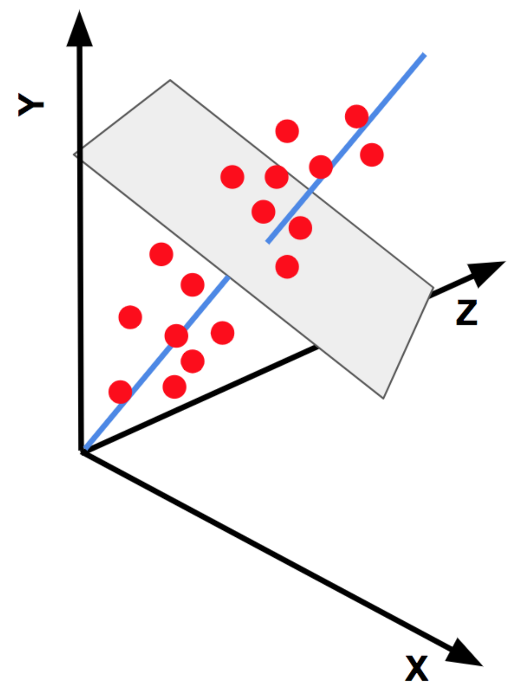
$$\frac{\boldsymbol{v}^T\boldsymbol{x}}{\|\boldsymbol{v}\|_2} = \frac{\|\boldsymbol{v}\|_2\|\boldsymbol{x}\|_2\cos(\theta)}{\|\boldsymbol{v}\|_2} = \|\boldsymbol{x}\|_2\cos(\theta)$$

- For simplicity, consider $\boldsymbol{u}$ with unit length, i.e., $\|\boldsymbol{v}\|_2 = 1$



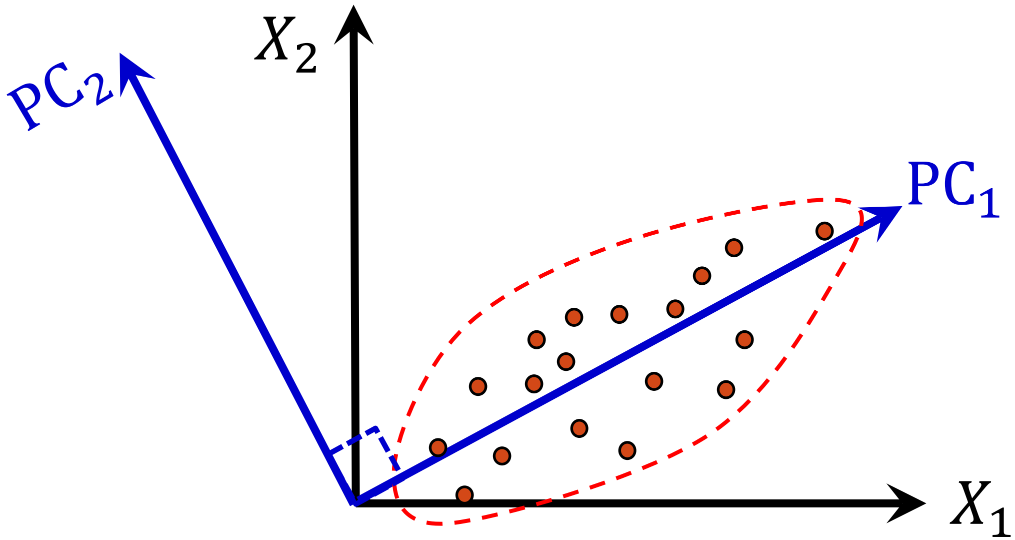
PCA: Geometric Rationale

- Data may only occupy a small subspace of the high-dimensional $\mathbb{R}^d$ space.
- We want to pick a few ($< d$) basis vectors that data are projected onto.
- Which basis vectors would you pick?



PCA: Geometric Rationale

- Goal: to find a projection or rotation of the original d-dimensional coordinate system to capture the largest amount of variation in data
  - Ordered s.t. the 1st principal component has the highest variance, the 2nd component has the next highest variance, …, the d-th component has the lowest variance
  - Principal components are orthogonal to each other



PCA: Algorithm

Input: $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ a set of observed data

1. Centering the data points s.t. the mean is **0**

$$\widehat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \longrightarrow \boldsymbol{x}_i = \boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}$$

2. Compute sample covariance matrix

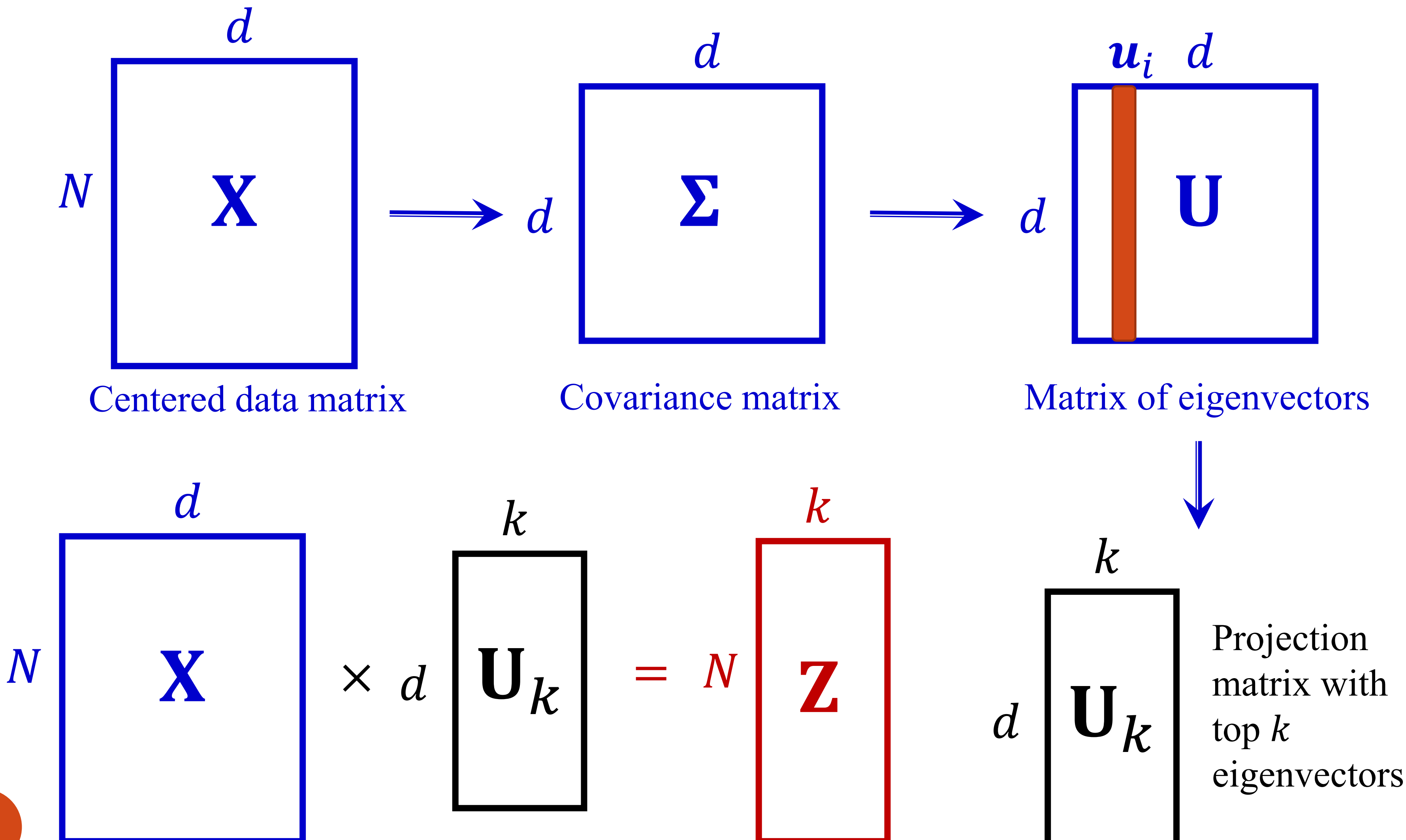
$$\widetilde{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N} \boldsymbol{x}_i {\boldsymbol{x}_i}^T$$

Each $\boldsymbol{u}_i$ is of d dimensions

3. Compute eigenvectors of $\widetilde{\boldsymbol{\Sigma}}$, $\{\boxed{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_d}\}$, which are sorted based on their eigenvalues in non-increasing order, i.e., $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$

4. Select the first k eigenvectors to construct principal components

PCA: Algorithm (Illustration)



Derivation of PCA

- The variance preservation view
  - The first k components display as much as possible of the variation among data instances
- The minimum reconstruction view
  - The first k components convey maximum useful information of original data instances

Appendix (optional)

Eigenvalues & Eigenvectors

- Given a d-by-d square matrix $\mathbf{A}$, if there exists a non-zero d-dimensional vector $\boldsymbol{u}$, s.t.

$$\mathbf{A}\boldsymbol{u} = \boxed{\lambda}\boldsymbol{u}$$

scalar

then $\boldsymbol{u}$ is an eigenvector of $\mathbf{A}$, and λ is called the corresponding eigenvalue

- Notes:
  - There are d eigenvectors and eigenvalues
  - An eigenvalue can be positive, negative or zero
  - An eigenvector cannot be a zero vector
  - <u>For symmetric matrices</u>, eigenvectors are orthogonal to each other

Properties of Eigenvalues

Given a square matrix $\mathbf{A}$ (d-by-d)

- $\mathbf{A}$ is invertible ($\mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$ or $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$) if all the eigenvalues of $\mathbf{A}$ are non-zero (positive or negative)
- If all the eigenvalues of $\mathbf{A}$ are non-negative, then $\mathbf{A}$ is a positive semi-definite matrix:

  For any non−zero vector $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$, we have $\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} \geq 0$
- If all the eigenvalues of $\mathbf{A}$ are positive, then $\mathbf{A}$ is a positive definite matrix:

  For any non−zero vector $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$, we have $\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} > 0$

Properties of Eigenvalues (cont.)

- Recall: when inducing a closed form solution of regularized linear regression model, we mentioned that if a matrix **A** can be written as

$$\mathbf{A} = \mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}, \text{where } \mathbf{X} \in \mathbb{R}^{N\times d}, \mathbf{I} \in \mathbb{R}^{d\times d} \text{and } \lambda > 0$$

then **A** is always invertible:

$$\exists\ \mathbf{A}^{-1}, \text{s.t.}, \mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \text{ and } \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

Properties of Eigenvalues (cont.)

- We first prove **A** is positive definite
  - For any non-zero vector $\boldsymbol{x} \in \mathbb{R}^{d\times 1}$

$$\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} = \boldsymbol{x}^T(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})\boldsymbol{x}$$

$$= \boldsymbol{x}^T(\mathbf{X}^T\mathbf{X})\boldsymbol{x} + \boldsymbol{x}^T(\lambda\mathbf{I})\boldsymbol{x}$$

Denote $\boldsymbol{z} = \mathbf{X}\boldsymbol{x}$

$$= \boldsymbol{z}^T\boldsymbol{z} + \lambda\boldsymbol{x}^T\boldsymbol{x}$$

$$= \|\boldsymbol{z}\|_2^2 + \lambda\|\boldsymbol{x}\|_2^2$$

$\|\boldsymbol{z}\|_2^2 \geq 0$ and $\|\boldsymbol{z}\|_2^2 = 0$ if and only if $\boldsymbol{z} = \mathbf{0}$

$\|\boldsymbol{x}\|_2^2 > 0$ because $\boldsymbol{x} \neq \mathbf{0} \Rightarrow \lambda\|\boldsymbol{x}\|_2^2 > 0$ as long as $\lambda > 0$

$$\boldsymbol{x}^T\mathbf{A}\boldsymbol{x} > 0$$

Properties of Eigenvalues (cont.)

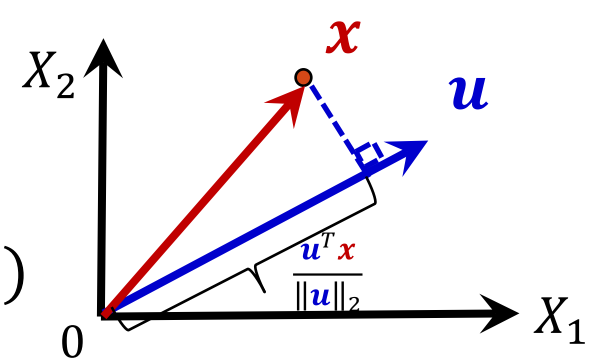
- As **A** is positive definite, all of its eigenvalues are positive, i.e., non-zero
- Recall: **A** is invertible if all the eigenvalues of **A** are non-zero (either positive or negative)
- Therefore, if a matrix **A** can be written as

$$\mathbf{A} = \mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}, \text{ where } \mathbf{X} \in \mathbb{R}^{N\times d}, \mathbf{I} \in \mathbb{R}^{d\times d} \text{ and } \lambda > 0$$

then **A** is invertible!

PCA: Variance Preservation

- The first k components display as much as possible of the variation among data instances
- Consider a projection of a data point $\boldsymbol{x}$ onto a vector going through the origin, represented by $\boldsymbol{u}$
- The projection of $\boldsymbol{x}$ onto $\boldsymbol{u}$ is



$$\frac{\boldsymbol{u}^T\boldsymbol{x}}{\|\boldsymbol{u}\|_2} = \frac{\|\boldsymbol{u}\|_2\|\boldsymbol{x}\|_2\cos(\theta)}{\|\boldsymbol{u}\|_2} = \|\boldsymbol{x}\|_2\cos(\theta)$$

- For simplicity, consider $\boldsymbol{u}$ with unit length, i.e., $\|\boldsymbol{u}\|_2 = 1$
- The projected instances $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$ onto $\boldsymbol{u}$ are

$$\{\boldsymbol{u}^T\boldsymbol{x}_1, \boldsymbol{u}^T\boldsymbol{x}_2, \dots, \boldsymbol{u}^T\boldsymbol{x}_N\}$$

Variance Preservation (cont.)

- In PCA, data points are centered at the beginning

$$\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = 0$$

- After projection onto $\boldsymbol{u}$, the mean of data points is still 0

$$\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{u}^T\boldsymbol{x}_i = \boldsymbol{u}^T\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = 0$$

- The variance of the data points projected onto $\boldsymbol{u}$ is

$$\frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{u}^T\boldsymbol{x}_i - 0)^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{u}^T\boldsymbol{x}_i)^2$$

$$= \boldsymbol{u}^T\boxed{\widetilde{\boldsymbol{\Sigma}}}\boldsymbol{u} \qquad \widetilde{\boldsymbol{\Sigma}} = \frac{1}{N-1}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^T = \frac{1}{N-1}\mathbf{X}^T\boxed{\mathbf{X}}$$

Each row is a data instance

Variance Preservation (cont.)

- The goal of PCA (for simplicity, projected on 1 principal component only) is to find $\boldsymbol{u}$ that maximizes the variance, expecting to maximally preserve distinction among data
- The resultant optimization problem is

$$\max_{\boldsymbol{u}} \quad \boldsymbol{u}^T \widetilde{\boldsymbol{\Sigma}}\, \boldsymbol{u}$$
$$\text{s.t.} \quad \|\boldsymbol{u}\|_2^2 = 1$$

- It can be solved by forming the Lagrangian

$$\boldsymbol{u}^T \widetilde{\boldsymbol{\Sigma}} \boldsymbol{u} + \lambda(1 - \boldsymbol{u}^T \boldsymbol{u})$$

- By setting the gradient w.r.t. $\boldsymbol{u}$ to zero, we have

$$2\widetilde{\boldsymbol{\Sigma}}\boldsymbol{u} - 2\lambda\boldsymbol{u} = \boldsymbol{0} \longrightarrow \boxed{\widetilde{\boldsymbol{\Sigma}}\boldsymbol{u} = \lambda\boldsymbol{u}}$$

The desired direction $\boldsymbol{u}$ is an eigenvector of $\widetilde{\boldsymbol{\Sigma}}$

$\widetilde{\boldsymbol{\Sigma}}$ has d eigenvectors, which one?

Variance Preservation (cont.)

- Recall that the variance of the projected dataset $\mathcal{D} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ is $\boldsymbol{u}^T \widetilde{\boldsymbol{\Sigma}}\, \boldsymbol{u}$
- By substituting $\widetilde{\boldsymbol{\Sigma}}\boldsymbol{u} = \lambda \boldsymbol{u}$ into the above formula, the projected variance becomes

$\|\boldsymbol{u}\|_2^2 \quad (\|\boldsymbol{u}\|_2^2 = 1)$

$$\boldsymbol{u}^T \widetilde{\boldsymbol{\Sigma}}\, \boldsymbol{u} = \boldsymbol{u}^T \lambda \boldsymbol{u} = \lambda \boxed{\boldsymbol{u}^T \boldsymbol{u}} = \lambda$$

- To find a direction that maximizes the projected variance is to find the eigenvector $\boldsymbol{u}$ of $\widetilde{\boldsymbol{\Sigma}}$ with the largest eigenvalue
- Generalized to multiple components case: let $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$ be the eigenvalues of $\widetilde{\boldsymbol{\Sigma}}$, and $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_d$ be the corresponding eigenvectors, and choose the top k eigenvectors as the principal components

Determine Value of *k*

- Wrapper approaches
  - Dimensionality reduction is usually an intermediate step for some downstream tasks, such as classification, regression, clustering
  - Use cross-validation based on the performance of the final task to tune the value of k

Determine Value of *k* (cont.)

- Based on the percentage of variance preserved

$$p_{\text{var}} = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \times 100$$

  - All the λ_i's are nonnegative
  - Predefine a value for the percentage of variance to determine the value of *k*

Compute Eigenvalues and Eigenvectors

- How to compute eigenvalues and eigenvectors of $\widetilde{\boldsymbol{\Sigma}} = \frac{1}{N-1}\mathbf{X}^T\mathbf{X}$?
- In a general case, if a d-by-d square matrix $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{X}^T\mathbf{X}, \text{ where } \mathbf{X} \in \mathbb{R}^{N\times d}$$

then eigenvectors and eigenvalues of $\mathbf{A} = \mathbf{X}^T\mathbf{X}$ can be computed by performing Singular Value Decomposition (SVD) on $\mathbf{X}$

Orthogonal Vectors

- Two vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are said to be orthogonal if they are perpendicular to each other, i.e., the inner or dot product of two vectors is 0
  - $\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = 0$
- A set of vectors $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_d\}$ are mutually orthogonal if every pair of vectors are orthogonal
  - $\boldsymbol{v}_i \cdot \boldsymbol{v}_j = 0$, for any $i \neq j$

$$\boldsymbol{v}_1 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \quad \boldsymbol{v}_2 = \begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix} \quad \boldsymbol{v}_3 = \begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix}$$

$$\boldsymbol{v}_1 \cdot \boldsymbol{v}_2 = \boldsymbol{v}_1 \cdot \boldsymbol{v}_3 = \boldsymbol{v}_2 \cdot \boldsymbol{v}_3 = 0$$

Orthonormal Vectors

- A set of vectors $\{v_1, \dots, v_d\}$ are mutually orthonormal if every pair of vectors are orthogonal, and the L_2 norm of each vector is 1
  - $\boldsymbol{v}_i \cdot \boldsymbol{v}_j = 0$, for any $i \neq j$
  - $\|\boldsymbol{v}_i\|_2 = \sqrt{\boldsymbol{v}_i \cdot \boldsymbol{v}_i} = 1$
- A set of orthogonal vectors $\{\boldsymbol{v}_1, \dots, \boldsymbol{v}_d\}$ can be normalized to orthonormal via $\left\{\frac{\boldsymbol{v}_1}{\|\boldsymbol{v}_1\|_2}, \dots, \frac{\boldsymbol{v}_d}{\|\boldsymbol{v}_d\|_2}\right\}$

$$\boldsymbol{v}_1 = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \quad \|\boldsymbol{v}_1\|_2 = \sqrt{2} \qquad \boldsymbol{v}_2 = \begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix} \quad \|\boldsymbol{v}_2\|_2 = 2 \qquad \boldsymbol{v}_3 = \begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix} \quad \|\boldsymbol{v}_3\|_2 = 2$$

$$\boldsymbol{v}_1' = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \quad \|\boldsymbol{v}_1'\|_2 = 1 \qquad \boldsymbol{v}_2' = \frac{1}{2}\begin{pmatrix} 1 \\ \sqrt{2} \\ 1 \end{pmatrix} \quad \|\boldsymbol{v}_2'\|_2 = 1 \qquad \boldsymbol{v}_3' = \frac{1}{2}\begin{pmatrix} 1 \\ -\sqrt{2} \\ 1 \end{pmatrix} \quad \|\boldsymbol{v}_3'\|_2 = 1$$

Orthonormal Vectors (cont.)

- Given a matrix $\mathbf{V} = (\boldsymbol{v}_1, \dots, \boldsymbol{v}_d)$, where $\boldsymbol{v}_i$ is an N-dimensional column vector, and $N \geq d$
- If the columns of $\mathbf{V}$ are mutually orthonormal, then we have

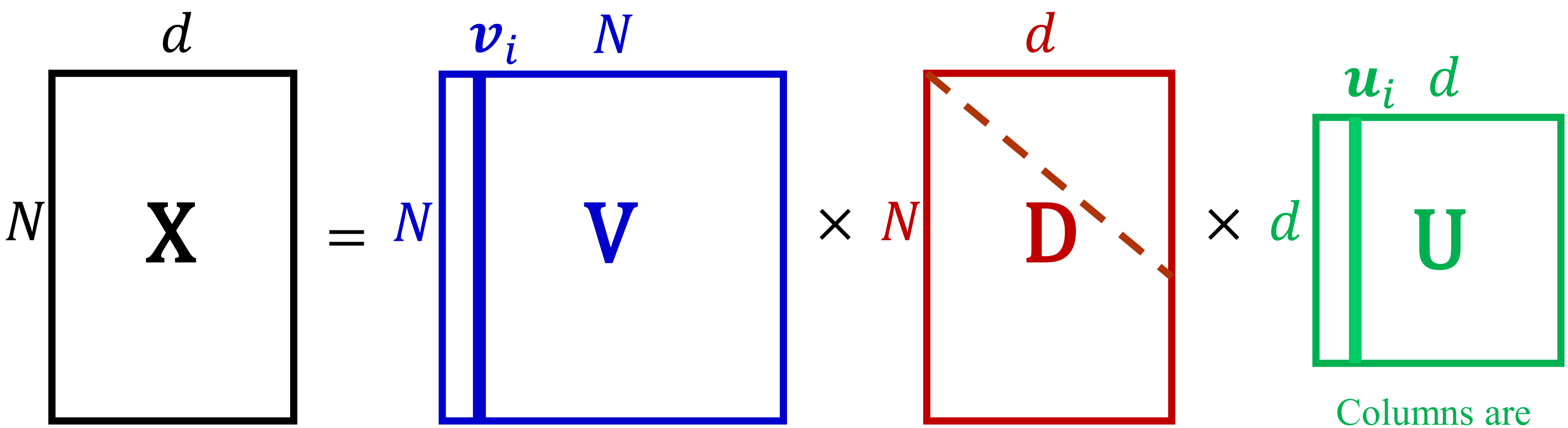
$$\mathbf{V}^T\mathbf{V} = \mathbf{I}_d$$

$$\mathbf{I}_d = \overbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}}^{d} \left.\vphantom{\begin{pmatrix} 1 \\ 0 \\ \cdots \\ 0 \end{pmatrix}}\right\} d$$

Singular Value Decomposition (SVD)

- The SVD of $\mathbf{X}$ (N-by-d) has the following form

$$\mathbf{X} = \mathbf{V}\mathbf{D}\mathbf{U}^T$$



Columns are orthonormal to each other

$$\mathrm{V}^T\mathrm{V} = \mathrm{I}_N$$

Supporse $d < N$. All elements are zero expect for the first d-by-d submatrix diagonal being $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \geq 0$

Columns are orthonormal to each other

$$\mathrm{U}^T\mathrm{U} = \mathrm{I}_d$$

Obtain Eigenvectors via SVD

- Perform SVD on $\mathbf{X}$ to get $\mathbf{X} = \mathbf{VDU}^T$
- Then $\mathbf{A}$ can be rewritten as

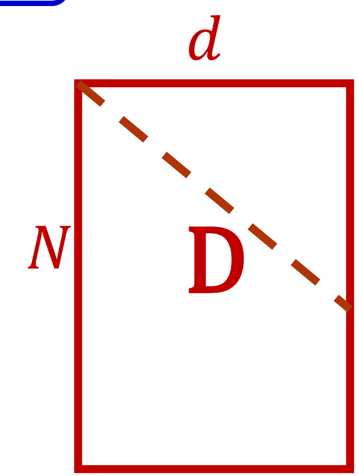
$\mathbf{V}^T\mathbf{V} = \mathbf{I}_N$

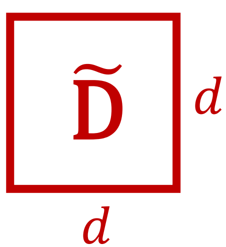
$$\mathbf{A} = \mathbf{X}^T\mathbf{X} = (\mathbf{VDU}^T)^T\mathbf{VDU}^T = \mathbf{UD}^T\boxed{\mathbf{V}^T\mathbf{V}}\mathbf{DU}^T$$

$$\mathbf{A} = \mathbf{UD}^T\mathbf{DU}^T$$

Denote $\widetilde{\mathbf{D}} = \mathbf{D}^T\mathbf{D}$

$$= \mathbf{U}\widetilde{\mathbf{D}}\mathbf{U}^T$$





d-by-d diagonal matrix with diagonal elements $\lambda_1^2 \geq \lambda_2^2 \geq \cdots \geq \lambda_d^2 \geq 0$

Suppose $d < N$. All elements are zero expect for the first d-by-d submatrix diagonal being $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$

Eigen Components via SVD (cont.)

$$\mathbf{A} = \mathbf{U}\widetilde{\mathbf{D}}\mathbf{U}^T$$

$\widetilde{\mathbf{D}}$ ($d \times d$)

Diagonal matrix with diagonal elements $\lambda_1^2 \geq \lambda_2^2 \geq \cdots \geq \lambda_d^2 \geq 0$

$$\mathbf{U}^T\mathbf{U} = \mathbf{I}_d$$

$$\boxed{\mathbf{AU}} = \mathbf{U}\widetilde{\mathbf{D}}\cancel{\mathbf{U}^T\mathbf{U}} = \boxed{\mathbf{U}\widetilde{\mathbf{D}}}$$

$$(\mathbf{A} \times \boldsymbol{u}_1, \mathbf{A} \times \boldsymbol{u}_2, \dots, \mathbf{A} \times \boldsymbol{u}_d) = [\lambda_1^2 \times \boldsymbol{u}_1, \lambda_2^2 \times \boldsymbol{u}_2, \dots, \lambda_d^2 \times \boldsymbol{u}_d]$$

$$\mathbf{A}\boldsymbol{u}_i = \lambda_i^2 \boldsymbol{u}_i, i = 1, \dots, d$$

Each column $\boldsymbol{u}_i$ of $\mathbf{U}$ is an eigenvector of $\mathbf{A}$ with the eigenvalue λ_i^2

Reference (Optional)

- For feature subset selection:
  - An Introduction to Variable and Feature Selection, Isabelle Guyon, Andre Elisseeff, in JMLR 2003
- For dimensionality reduction:
  - Dimensionality Reduction: A Comparative Review, L.J.P. van der Maaten and E. O. Postma and H. J. van den Herik, Technical Report, 2008
  - https://lvdmaaten.github.io/drtoolbox/

Thank you!

Appendix (optional)

Derivation of PCA

- The variance preservation view
  - The first k components display as much as possible of the variation among data instances
- The minimum reconstruction view
  - The first k components convey maximum useful information of original data instances

Appendix (optional)

Minimum Reconstruction Error

- Given any <u>orthonormal</u> basis $\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_d$, a data point $\boldsymbol{x}_i$ (has been centered) can be written as

$$\boldsymbol{x}_i = \sum_{j=1}^{d} \boxed{\alpha_{ij}} \boldsymbol{v}_j \qquad \alpha_{ij} = \boldsymbol{v}_j^T \boldsymbol{x}_i$$

$$\boxed{\sum_{j=1}^{d} \boldsymbol{v}_j^T \boldsymbol{x}_i \boldsymbol{v}_j = \boldsymbol{x}_i \sum_{j=1}^{d} \boldsymbol{v}_j^T \boldsymbol{v}_j = \boldsymbol{x}_i}$$

- Consider the k-term approximation of $\boldsymbol{x}_i$:

$$\hat{\boldsymbol{x}}_i \approx \sum_{j=1}^{k} \alpha_{ij} \boldsymbol{v}_j$$

- The error of the approximate over all data points is

$$E = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2^2 = \frac{1}{N} \sum_{i=1}^{N} \left\| \sum_{j=k+1}^{d} \alpha_{ij} \boldsymbol{v}_j \right\|_2^2 = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=k+1}^{d} \alpha_{ij}^2$$

Appendix (optional)

Minimum Reconstruction Error (cont.)

- The error of the approximate over all data points

$$E = \frac{1}{N}\sum_{i=1}^{N}\|\widehat{\boldsymbol{x}}_i - \boldsymbol{x}_i\|_2^2 = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=k+1}^{d}\alpha_{ij}^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=k+1}^{d}\boldsymbol{v}_j^T\boldsymbol{x}_i\boldsymbol{x}_i{}^T\boldsymbol{v}_j \approx \sum_{j=k+1}^{d}\boldsymbol{v}_j^T\widetilde{\boldsymbol{\Sigma}}\boldsymbol{v}_j$$

- Suppose $k = d - 1$, i.e., we aim to remove a single dimension, then resultant optimization problem is

$$\min_{\boldsymbol{v}_d} \quad \boldsymbol{v}_d^T\widetilde{\boldsymbol{\Sigma}}\,\boldsymbol{v}_d$$
$$\text{s.t.} \quad \|\boldsymbol{v}_d\|_2^2 = 1$$

Appendix (optional)

Minimum Reconstruction Error (cont.)

- By setting the gradient of the Lagrangian w.r.t. $\boldsymbol{v}$ to zero, we have

$$2\widetilde{\boldsymbol{\Sigma}}\boldsymbol{v}_d - 2\lambda\boldsymbol{v}_d = \boldsymbol{0} \longrightarrow \boxed{\widetilde{\boldsymbol{\Sigma}}\boldsymbol{v}_d = \lambda\boldsymbol{v}_d}$$

The desired direction $\boldsymbol{v}_d$ is an eigenvector of $\widetilde{\boldsymbol{\Sigma}}$

$\widetilde{\boldsymbol{\Sigma}}$ has d eigenvectors, which one?

- Our goal is to minimize the reconstruction error $\boldsymbol{v}_d^T\widetilde{\boldsymbol{\Sigma}}\,\boldsymbol{v}_d$

$$\boldsymbol{v}_d^T\widetilde{\boldsymbol{\Sigma}}\,\boldsymbol{v}_d = \boldsymbol{v}_d^T\lambda\boldsymbol{v}_d = \lambda\boldsymbol{v}_d^T\boldsymbol{v}_d = \lambda$$

- Therefore, $\boldsymbol{v}_d$ should be the eigenvector $\boldsymbol{u}_d$ of $\widetilde{\boldsymbol{\Sigma}}$ with the smallest eigenvalue because $\boldsymbol{u}_d^T\widetilde{\boldsymbol{\Sigma}}\boldsymbol{u}_d = \lambda_d$
- Similarly, the other dimensions to remove are subsequently the eigenvectors corresponding to the least eigenvalues