

## Lecture 10: Dimensionality Reduction with Principal Component Analysis

Yi, Yung (이웅)

Mathematics for Machine Learning  
<https://yung-web.github.io/home/courses/mathml.html>  
KAIST EE

April 6, 2021

Please watch this tutorial video by Luis Serrano on PCA.

<https://www.youtube.com/watch?v=g-Hb26agBFg>

April 6, 2021 1 / 42

April 6, 2021 2 / 42

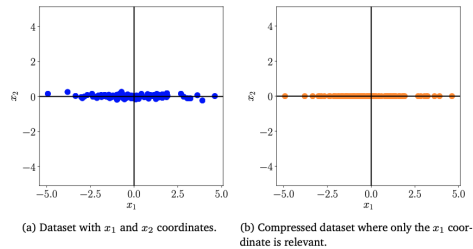
- (1) Problem Setting
- (2) Maximum Variance Perspective
- (3) Projection Perspective
- (4) Eigenvector Computation and Low-Rank Approximations
- (5) PCA in High Dimensions
- (6) Key Steps of PCA in Practice
- (7) Latent Variable Perspective

- (1) **Problem Setting**
- (2) Maximum Variance Perspective
- (3) Projection Perspective
- (4) Eigenvector Computation and Low-Rank Approximations
- (5) PCA in High Dimensions
- (6) Key Steps of PCA in Practice
- (7) Latent Variable Perspective

April 6, 2021 3 / 42

L10(1)

April 6, 2021 4 / 42



- High-dimensional data
  - hard to analyze and visualize
  - Often, overcomplete and many dimensions are redundant
- Compact data representation is always preferred just like compression.
- PCA (Principal Component Analysis) is a representative method.

L10(1)

April 6, 2021 5 / 42

- 5 dimensions
  1. Size
  2. Number of rooms
  3. Number of bathrooms
  4. Schools around
  5. Crime rate
- 2 dimensions
  - Size feature
  - Location feature

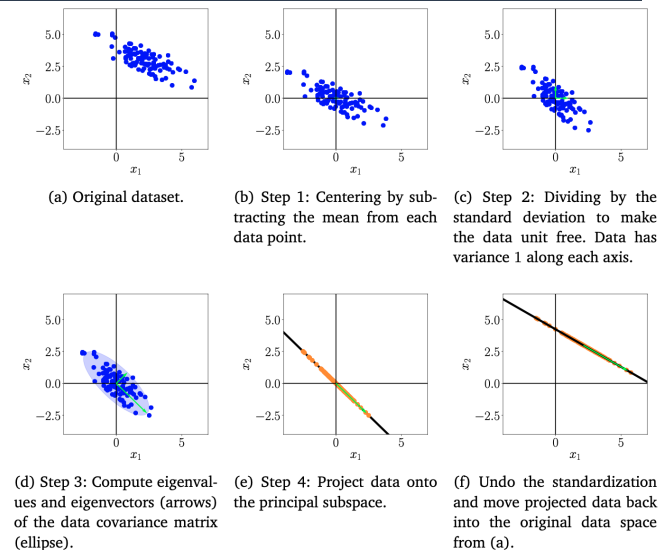
L10(1)

April 6, 2021 6 / 42

- S1. Centering.** Centering the data by subtracting mean
- S2. Standardization.** Divide the data points by the standard deviation for every dimension (original feature)  $d = 1, \dots, D$
- S3. Eigenvalue/vector.** Compute the  $M$ -largest eigenvalues and the eigenvectors of the data covariance matrix ( $M$  is the dimension that needs to be reduced)
- S4. Projection.** Project all data points onto the space defined by the eigenvectors (i.e., principal subspace).
- S5.** Undo standardization and centering.

L10(1)

April 6, 2021 7 / 42



L10(1)

April 6, 2021 8 / 42

- $N$ : number of samples,  $D$ : number of measurements (or original features)
- iid dataset  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  whose mean is 0 (well-centered), where each  $\mathbf{x}_i \in \mathbb{R}^D$ , and its corresponding data matrix

$$\mathbf{X} = (\mathbf{x}_1 \cdots \mathbf{x}_N) = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{D,1} & x_{D,2} & \cdots & x_{D,N} \end{pmatrix} \in \mathbb{R}^{D \times N}$$

- (data) covariance matrix

$$\mathbf{S} = \frac{1}{N} \mathbf{X} \mathbf{X}^T = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T \in \mathbb{R}^{D \times D}$$

L10(1)

April 6, 2021 9 / 42

- Covariance matrix for a random vector  $\mathbf{Y} = (Y_1, \dots, Y_D)^T$ ,

L6(4)

$$\Sigma_{\mathbf{Y}} = \begin{pmatrix} \text{cov}(Y_1, Y_1) & \text{cov}(Y_1, Y_2) & \cdots & \text{cov}(Y_1, Y_D) \\ \vdots & \vdots & \ddots & \vdots \\ \text{cov}(Y_D, Y_1) & \text{cov}(Y_D, Y_2) & \cdots & \text{cov}(Y_D, Y_D) \end{pmatrix}$$

- Data covariance matrix  $\mathbf{S} \in \mathbb{R}^{D \times D}$ 
  - Each  $Y_i$  has  $N$  samples  $(x_{i,1} \cdots x_{i,N})$

$$\begin{aligned} S_{ij} = \text{cov}(Y_i, Y_j) &= \frac{1}{N} \sum_{k=1}^N x_{i,k} \cdot x_{j,k} \\ &= \text{average covariance (over samples) btwn features } i \text{ and } j \end{aligned}$$

L10(1)

April 6, 2021 10 / 42

- Low-dimensional compressed representation, also called **code**:

$$\mathbf{z}_n = \mathbf{B}^T \mathbf{x}_n \in \mathbb{R}^M,$$

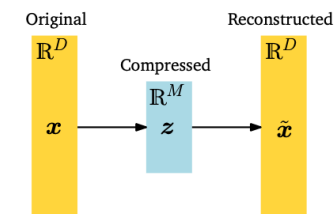
where the projection<sup>1</sup> matrix is  $\mathbf{B} := (\mathbf{b}_1, \dots, \mathbf{b}_M) \in \mathbb{R}^{D \times M}$ ,

- Assume that the columns of  $\mathbf{B}$  are orthonormal, i.e.,  $\mathbf{b}_i^T \mathbf{b}_j = 0$  if  $i \neq j$ , and  $\mathbf{b}_i^T \mathbf{b}_i = 1$  if  $i = j$ .
- Seek an  $M$ -dimensional subspace  $U \subset \mathbb{R}^D$ ,  $\dim(U) = M < D$  onto which we project data
- $\tilde{\mathbf{x}}_n \in \mathbb{R}^D$ : projected data,  $\mathbf{z}_n$ : their coordinates w.r.t. the basis vectors of  $\mathbf{B}$ .

<sup>1</sup>In L3(8), the coordinate in the projected space becomes  $\boldsymbol{\lambda} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{x}$ , which is simply  $\mathbf{B}^T \mathbf{x}$  for orthonormal bases  $\mathbf{B}$ .

L10(1)

April 6, 2021 11 / 42



- Find a suitable matrix  $\mathbf{B}$  such that  $\mathbf{z} = \mathbf{B}^T \mathbf{x}$  and  $\tilde{\mathbf{x}} = \mathbf{B} \mathbf{z}$
- $\mathbf{B}^T$ : encoder,  $\mathbf{B}$ : decoder
- **Example.** MNIST dataset
  - handwritten digits,  $N = 60,000$  data samples,  $D = 28 \times 28 = 784$  pixels

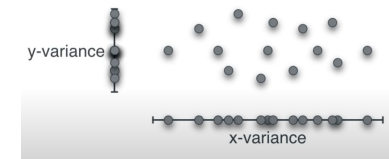


L10(1)

April 6, 2021 12 / 42

- (1) Problem Setting
- (2) **Maximum Variance Perspective**
- (3) Projection Perspective
- (4) Eigenvector Computation and Low-Rank Approximations
- (5) PCA in High Dimensions
- (6) Key Steps of PCA in Practice
- (7) Latent Variable Perspective

- Information content in the data
  - space filling
  - information in the data by looking at how much data is spread out
- PCA
  - a dimensionality reduction algorithm that maximizes the variance in the low-dimensional data representation.



source: Youtube channel by Luis Serrano

- $B = (\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_M)$ , where  $\mathbf{b}_i \in \mathbb{R}^D$  and  $B \in \mathbb{R}^{D \times M}$
- $B^T = \begin{pmatrix} \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_M^T \end{pmatrix} \in \mathbb{R}^{M \times D}$ ,  $\mathbf{b}_i^T \in \mathbb{R}^{1 \times D}$ ,  $\mathbf{x}_i \in \mathbb{R}^{D \times 1}$
- $z_n = \begin{pmatrix} z_{1n} \\ \vdots \\ z_{Mn} \end{pmatrix} = B^T \mathbf{x}_n = \begin{pmatrix} \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_M^T \end{pmatrix} \mathbf{x}_n = \begin{pmatrix} \mathbf{b}_1^T \mathbf{x}_n \\ \vdots \\ \mathbf{b}_M^T \mathbf{x}_n \end{pmatrix}$
- $z_{in}$ : new coordinate (for  $\mathbf{x}_n$ ) in the projected space by the basis  $\mathbf{b}_i$

- **Goal:** Find the orthonormal bases  $B = (\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_M)$  that maximizes the variance.
- **Result:** For the  $M$ -largest eigenvalues  $\lambda_1, \dots, \lambda_M$  of the data covariance matrix  $S$ , their corresponding  $M$  eigenvectors become  $\mathbf{b}_1, \dots, \mathbf{b}_M$
- **Question.** Why data covariance matrix? Why eigenvectors ordered by their eigenvalues?
- Strategy: Induction
  - Step 1.** We seek a single vector  $\mathbf{b}_1$  that maximizes the variance of the projected data, assuming that we project the data onto an 1D line. We show that  $\mathbf{b}_1$  is the **eigenvector of the largest eigenvalue**.
  - Step k.** Suppose that we found  $\mathbf{b}_1, \dots, \mathbf{b}_{k-1}$  for the variance maximization. Then, we seek  $\mathbf{b}_k$  that maximizes the variance of the projected data onto  $k$ -D plain with the constraint that  $\mathbf{b}_k$  is orthogonal to  $\mathbf{b}_1, \dots, \mathbf{b}_{k-1}$ . We prove that  $\mathbf{b}_k$  is the **eigenvector of the  $k$ -th largest eigenvalue**.

## Step 1: Finding $\mathbf{b}_1$ (1)

- Variance (over  $N$  sample data) of the first coordinate  $z_1$  of  $\mathbf{z} \in \mathbb{R}^M$ , so that

$$V_1 := \text{var}[z_1] = \frac{1}{N} \sum_{n=1}^N z_{1n}^2, \quad z_{1n} = \mathbf{b}_1^T \mathbf{x}_n$$

where  $z_{1n}$  ( $z_{in}$ ) is the first ( $i$ -th) coordinate of the low-dimensional representation  $\mathbf{z}_n$  of  $\mathbf{x}_n$

$$V_1 = \frac{1}{N} \sum_{n=1}^N (\mathbf{b}_1^T \mathbf{x}_n)^2 = \frac{1}{N} \sum_{n=1}^N \mathbf{b}_1^T \mathbf{x}_n \mathbf{x}_n^T \mathbf{b}_1 = \mathbf{b}_1^T \left( \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^T \right) \mathbf{b}_1 = \mathbf{b}_1^T \mathbf{S} \mathbf{b}_1$$

- Find  $\mathbf{b}_1$  that maximizes  $V_1$ .

$$\max_{\mathbf{b}_1} \mathbf{b}_1^T \mathbf{S} \mathbf{b}_1, \quad \text{subject to} \quad \|\mathbf{b}_1\|^2 = 1$$

L10(2)

April 6, 2021 17 / 42

## Step 1: Finding $\mathbf{b}_1$ (2)

- Optimization problem

$$\max_{\mathbf{b}_1} \mathbf{b}_1^T \mathbf{S} \mathbf{b}_1, \quad \text{subject to} \quad \|\mathbf{b}_1\|^2 = 1$$

- Using the Lagrange multiplier method, we get:

L7(2), L7(4)

$$\mathbf{S} \mathbf{b}_1 = \lambda_1 \mathbf{b}_1, \quad \mathbf{b}_1^T \mathbf{b}_1 = 1 \implies \lambda_1: \text{eigenvalue}, \mathbf{b}_1: \text{eigenvector of } \mathbf{S}$$

- Then,  $V_1 = \mathbf{b}_1^T \mathbf{S} \mathbf{b}_1 = \lambda_1 \mathbf{b}_1^T \mathbf{b}_1 = \lambda_1$  (the variance  $V_1$  is the eigenvalue of  $\mathbf{S}$ )
- To maximize the variance, we take the largest eigenvalue, and the corresponding eigenvector is called the (first) principal component.

L10(2)

April 6, 2021 18 / 42

## Step $k$ : Finding $\mathbf{b}_k$ (1)

- Finding  $k$ -th principal component: Solving the following optimization problem

$$\max_{\mathbf{b}} \mathbf{b}^T \mathbf{S} \mathbf{b}, \quad \text{subject to} \quad \mathbf{b}^T \mathbf{b} = 1 \text{ and } \mathbf{b}^T \mathbf{b}_i = 0, \quad i = 1, \dots, k-1$$

- Claim.** The solution of the above is the eigenvector of  $\mathbf{S}$  corresponding to its  $k$ -th largest eigenvalue.
- Proof.** By induction hypothesis,  $\mathbf{b}_1, \dots, \mathbf{b}_k$  are the orthonormal eigenvectors of  $\mathbf{S}$ . Denote the  $i$ -th largest eigenvalue of  $\mathbf{S}$  by  $\lambda_i$ , where note that  $\mathbf{S} \mathbf{b}_i = \lambda_i \mathbf{b}_i$ . The lagrangian of the objective function is:

$$\mathcal{L}(\mathbf{b}) = \mathbf{b}^T \mathbf{S} \mathbf{b} - \lambda(\mathbf{b}^T \mathbf{b} - 1) + \sum_{i=1}^k \eta_i \mathbf{b}^T \mathbf{b}_i$$

L10(2)

April 6, 2021 19 / 42

## Step $k$ : Finding $\mathbf{b}_k$ (2)

- Letting the solution be denoted by  $\mathbf{b}_{k+1}$ , the first-order necessary condition for optimality is:

$$\nabla \mathcal{L}(\mathbf{b}_{k+1}) = 2\mathbf{S} \mathbf{b}_{k+1} - 2\lambda \mathbf{b}_{k+1} + \sum_{i=1}^k \eta_i \mathbf{b}_i = 0 \quad (*)$$

- Now, for any  $j \in \{1, \dots, k\}$ ,

$$\begin{aligned} 0 &= \mathbf{b}_j^T \nabla \mathcal{L}(\mathbf{b}_{k+1}) = 2\mathbf{b}_j^T \mathbf{S} \mathbf{b}_{k+1} - 2\lambda \mathbf{b}_j^T \mathbf{b}_{k+1} + \sum_{i=1}^k \eta_i \mathbf{b}_j^T \mathbf{b}_i = 2(\mathbf{S} \mathbf{b}_j)^T \mathbf{b}_{k+1} + \eta_j \\ &= 2(\lambda \mathbf{b}_j)^T \mathbf{b}_{k+1} + \eta_j = 2\lambda \mathbf{b}_j^T \mathbf{b}_{k+1} + \eta_j = \eta_j \end{aligned}$$

- From  $\eta_j = 0$  and  $(*)$ ,  $\mathbf{S} \mathbf{b}_{k+1} = \lambda \mathbf{b}_{k+1} \implies \lambda$  is an eigenvalue and its corresponding eigenvector is  $\mathbf{b}_{k+1}$ .
- Note that the objective function is  $\lambda$ , because  $\mathbf{b}^T \mathbf{S} \mathbf{b} = \lambda \mathbf{b}^T \mathbf{b}$ .

L10(2)

April 6, 2021 20 / 42

- **Question.** How can we choose the largest  $\lambda$  with the constraint that  $\mathbf{b}_{k+1} \perp (\mathbf{b}_1, \dots, \mathbf{b}_k)$ ?
- Clearly, if  $\mathbf{b}_{k+1}$  is equal to any of these eigenvectors (up to sign), the constraint will be violated, so, to maximize  $\lambda$ ,  $\mathbf{b}_{k+1}$  should be a unit eigenvector of  $\mathbf{S}$  corresponding to  $(k+1)$ -th largest eigenvalue.
- By spectral theorem, we can choose this vector in such a way that it is orthogonal to  $\mathbf{b}_1, \dots, \mathbf{b}_k$ .

L4(4)

- (1) Problem Setting
- (2) Maximum Variance Perspective
- (3) **Projection Perspective**
- (4) Eigenvector Computation and Low-Rank Approximations
- (5) PCA in High Dimensions
- (6) Key Steps of PCA in Practice
- (7) Latent Variable Perspective

- An ordered orthonormal basis (ONB)  $\mathbf{B} = (\mathbf{b}_1, \dots, \mathbf{b}_D)$
- $\mathbf{B} = (\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_M)$ , where  $\mathbf{b}_i \in \mathbb{R}^D$  and  $\mathbf{B} \in \mathbb{R}^{D \times M}$
- Encoding:  $\mathbf{z}_n = \phi(\mathbf{x}_n)$  for some mapping  $\phi(\cdot)$
- Decoding:  $\tilde{\mathbf{x}}_n := \mathbf{B}\mathbf{z}_n = \sum_{m=1}^M z_{mn} \mathbf{b}_m$
- Goal: find the best linear projection of  $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$  onto a lower-dimensional subspace  $U$  (also, called **principal subspace**) of  $\mathbb{R}^D$  with  $\dim(U) = M$ .
- Formally, minimize the following **reconstruction error**

$$J_M := \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2,$$

where the variables are  $(\mathbf{z}_n : n = 1, \dots, N)$  and  $(\mathbf{b}_1, \dots, \mathbf{b}_M)$

**Step 1.** We optimize the coordinate  $\mathbf{z}_n$  in the space  $U$  for a given ONB  $(\mathbf{b}_1, \dots, \mathbf{b}_M)$

**Step 2.** Then, we find the optimal ONB, knowing the optimal  $\mathbf{z}_n$  in **Step 1**.

- Intuition: Orthogonal projection

L3(8)

$$\text{Result : } \tilde{\mathbf{x}}_n = \mathbf{B}(\mathbf{B}^\top \mathbf{B})^{-1} \mathbf{B}^\top \mathbf{x}_n = \mathbf{B} \mathbf{B}^\top \mathbf{x}_n = \mathbf{B} \mathbf{z}_n, \mathbf{z}_n = \mathbf{B}^\top \mathbf{x}_n$$

- Proof.** Assume an ONB  $(\mathbf{b}_1, \dots, \mathbf{b}_M)$ . Noting that  $J_M$  is a function of  $\tilde{\mathbf{x}}_n$  and  $\tilde{\mathbf{x}}_n$  is a function of  $\mathbf{z}_n$ ,

$$\frac{\partial J_M}{\partial \mathbf{z}_{in}} = \frac{\partial J_M}{\partial \tilde{\mathbf{x}}_n} \frac{\partial \tilde{\mathbf{x}}_n}{\partial \mathbf{z}_{in}}, \quad \frac{\partial J_M}{\partial \tilde{\mathbf{x}}_n} = -\frac{2}{N} (\mathbf{x}_n - \tilde{\mathbf{x}}_n)^\top, \quad \frac{\partial \tilde{\mathbf{x}}_n}{\partial \mathbf{z}_{in}} = \frac{\partial}{\partial \mathbf{z}_{in}} \left( \sum_{m=1}^M z_{mn} \mathbf{b}_m \right) = \mathbf{b}_i$$

$$\begin{aligned} \frac{\partial J_M}{\partial \mathbf{z}_{in}} &= -\frac{2}{N} (\mathbf{x}_n - \tilde{\mathbf{x}}_n)^\top \mathbf{b}_i = -\frac{2}{N} \left( \mathbf{x}_n - \sum_{m=1}^M z_{mn} \mathbf{b}_m \right)^\top \mathbf{b}_i \stackrel{\text{ONB}}{=} -\frac{2}{N} (\mathbf{x}_n^\top \mathbf{b}_i - z_{in} \mathbf{b}_i^\top \mathbf{b}_i) \\ &= -\frac{2}{N} (\mathbf{x}_n^\top \mathbf{b}_i - z_{in}) \end{aligned}$$

- $z_{in} = \mathbf{x}_n^\top \mathbf{b}_i = \mathbf{b}_i^\top \mathbf{x}_n$  for  $i = 1, \dots, M$  and  $n = 1, \dots, N$  (ortho. proj. onto 1D L3(8))

L10(3)

April 6, 2021 25 / 42

- The difference:  $\mathbf{x}_n - \tilde{\mathbf{x}}_n = \left( \sum_{j=M+1}^D \mathbf{b}_j \mathbf{b}_j^\top \right) \mathbf{x}_n = \sum_{j=M+1}^D (\mathbf{x}_n^\top \mathbf{b}_j) \mathbf{b}_j$

$$\tilde{\mathbf{x}}_n = \sum_{m=1}^M z_{mn} \mathbf{b}_m \stackrel{\text{Step 1}}{=} \sum_{m=1}^M (\mathbf{x}_n^\top \mathbf{b}_m) \mathbf{b}_m = \sum_{m=1}^M \mathbf{b}_m (\mathbf{b}_m^\top \mathbf{x}_n) = \left( \sum_{m=1}^M \mathbf{b}_m \mathbf{b}_m^\top \right) \mathbf{x}_n$$

$$\mathbf{x}_n = \sum_{d=1}^D z_{dn} \mathbf{b}_d = \left( \sum_{m=1}^M \mathbf{b}_m \mathbf{b}_m^\top \right) \mathbf{x}_n + \left( \sum_{j=M+1}^D \mathbf{b}_j \mathbf{b}_j^\top \right) \mathbf{x}_n$$

- The projection of the data point onto the orthogonal complement of the principal subspace L3(6)

L10(3)

April 6, 2021 26 / 42

$$\begin{aligned} J_M &= \frac{1}{N} \sum_{n=1}^N \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 = \frac{1}{N} \sum_{n=1}^N \left\| \sum_{j=M+1}^D (\mathbf{b}_j^\top \mathbf{x}_n) \mathbf{b}_j \right\|^2 = \frac{1}{N} \sum_{n=1}^N \sum_{j=M+1}^D (\mathbf{b}_j^\top \mathbf{x}_n)^2 \\ &= \frac{1}{N} \sum_{n=1}^N \sum_{j=M+1}^D \mathbf{b}_j^\top \mathbf{x}_n \mathbf{x}_n^\top \mathbf{b}_j = \sum_{j=M+1}^D \mathbf{b}_j^\top \left( \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n \mathbf{x}_n^\top \right) \mathbf{b}_j = \sum_{j=M+1}^D \mathbf{b}_j^\top \mathbf{S} \mathbf{b}_j \end{aligned}$$

- minimizing the squared reconstruction error = minimizing the variance when projected onto the orthogonal complement of the principal subspace = maximizing the variance of the projection in the principal subspace

- $J_M = \sum_{j=M+1}^D \lambda_j$  (because of the projection). To minimize this error, we need to choose the smallest  $D - M$  eigenvalues, which means that we need to choose the  $M$  largest eigenvalues and take their corresponding eigenvectors for projection.

L10(3)

April 6, 2021 27 / 42

- (1) Problem Setting
- (2) Maximum Variance Perspective
- (3) Projection Perspective
- (4) Eigenvector Computation and Low-Rank Approximations
- (5) PCA in High Dimensions
- (6) Key Steps of PCA in Practice
- (7) Latent Variable Perspective

L10(4)

April 6, 2021 28 / 42

- Approach 1: **EVD** L4(4)
  - Perform an eigendecomposition and compute the eigenvalues and eigenvectors of the symmetric matrix  $\mathbf{S}$  directly.
- Approach 2: **SVD** L4(5)
  - SVD of the data matrix  $\mathbf{X}$ :  $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$  ( $[D \times N] = [D \times D] \cdot [D \times N] \cdot [N \times N]$ )
  - $\mathbf{U}$  and  $\mathbf{V}^T$ : orthogonal matrices,  $\mathbf{\Sigma}$ : only nonzero entries are the singular values  $\sigma_{ii} \geq 0$ .
$$\mathbf{S} = \frac{1}{N} \mathbf{X}\mathbf{X}^T = \frac{1}{N} \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \mathbf{V}\mathbf{\Sigma}^T \mathbf{U}^T \stackrel{(\mathbf{V}^T = \mathbf{V}^{-1})}{=} \frac{1}{N} \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T \mathbf{U}^T$$
  - The columns of  $\mathbf{U}$  are the eigenvectors of  $\mathbf{X}\mathbf{X}^T$  (thus  $\mathbf{S}$ )
  - The eigenvalues  $\lambda_d$  of  $\mathbf{S}$  are related to the singular values of  $\mathbf{X}$ :  $\lambda_d = \frac{\sigma_d^2}{N}$

- In SVD,  $\mathbf{U}$  corresponds to the projection matrix  $\mathbf{B}$ , so that we maximize the variance of the projected data or minimize the average squared reconstruction error.

- Consider the best rank- $M$  approximation

$$\tilde{\mathbf{X}}_M := \arg \min_{\text{rk}(\mathbf{A})=M} \|\mathbf{X} - \mathbf{A}\|_2$$

- From Eckart-Young Theorem, by truncating the SVD at the top- $M$  singular value, we obtain the reconstructed data matrix  $\tilde{\mathbf{X}}_M$  as: L4(5), L4(6)

$$\tilde{\mathbf{X}}_M = \underbrace{\mathbf{U}_M}_{D \times M} \underbrace{\mathbf{\Sigma}_M}_{M \times M} \underbrace{\mathbf{V}_M^T}_{M \times N} \iff \tilde{\mathbf{X}}_M = \sum_{i=1}^M \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

where  $\sigma_i$  is the  $i$ -th singular value.

- (1) Problem Setting
- (2) Maximum Variance Perspective
- (3) Projection Perspective
- (4) Eigenvector Computation and Low-Rank Approximations
- (5) **PCA in High Dimensions**
- (6) **Key Steps of PCA in Practice**
- (7) Latent Variable Perspective

- In some practical cases,  $\mathbf{S} = \frac{1}{N} \mathbf{X}\mathbf{X}^T \in \mathbb{R}^{D \times D}$ , where  $D$  is pretty high.
  - **Example.** 100 × 100 pixel image:  $D = 10,000$ .
- What if  $N \ll D$ ?
  - With no duplicate data,  $\text{rk}(\mathbf{S}) = N$ , and  $D - N + 1$  eigenvalues are 0!  $\implies$  no need to maintain  $D \times D$  data covariance matrix.

- In PCA,  $\mathbf{S}\mathbf{b}_m = \lambda_m \mathbf{b}_m$ ,  $m = 1, \dots, M$ .

$$\mathbf{S}\mathbf{b}_m = \frac{1}{N} \mathbf{X}\mathbf{X}^T \mathbf{b}_m = \lambda_m \mathbf{b}_m \implies \frac{1}{N} \underbrace{\mathbf{X}^T \mathbf{X}}_{N \times N} \underbrace{\mathbf{X}^T \mathbf{b}_m}_{:= \mathbf{c}_m} = \lambda_m \mathbf{X}^T \mathbf{b}_m \iff \frac{1}{N} \mathbf{X}^T \mathbf{X} \mathbf{c}_m = \lambda_m \mathbf{c}_m$$

- $\lambda_m$  is an eigenvalue of  $\frac{1}{N} \mathbf{X}^T \mathbf{X}$  with its associated eigenvector  $\mathbf{c}_m = \mathbf{X}^T \mathbf{b}_m$
- $\frac{1}{N} \mathbf{X}^T \mathbf{X} \in \mathbb{R}^{N \times N}$ , so much easier to compute the eigenvectors
- To recover the eigenvector of  $\mathbf{S}$ , by left-multiplying  $\mathbf{X}$ , we get  $\frac{1}{N} \mathbf{X}\mathbf{X}^T \mathbf{X} \mathbf{c}_m = \lambda_m \mathbf{X} \mathbf{c}_m$



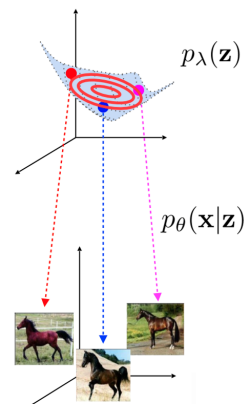
- S1. Centering.** Centering the data by subtracting mean
- S2. Standardization.** Divide the data points by the standard deviation for every dimension (original feature)  $d = 1, \dots, D$
- S3. Eigenvalue/vector.** Compute the  $M$ -largest eigenvalues and the eigenvectors of the data covariance matrix ( $M$  is the dimension that needs to be reduced)
- S4. Projection.** Project all data points onto the space defined by the eigenvectors (i.e., principal subspace).
- S5.** Undo standardization and centering.

- (1) Problem Setting
- (2) Maximum Variance Perspective
- (3) Projection Perspective
- (4) Eigenvector Computation and Low-Rank Approximations
- (5) PCA in High Dimensions
- (6) Key Steps of PCA in Practice
- (7) Latent Variable Perspective

- Generative process
  - $\mathbf{z} \sim p(\mathbf{z})$
  - $\mathbf{x} \sim p(\mathbf{x}|\mathbf{z})$

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z}) d\mathbf{z}$$

$$= \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}$$



Source: <https://dlvu.github.io/slides/dlvu.lecture06.pdf>

- $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$
- A linear relationship between  $\mathbf{z}$  and  $\mathbf{x}$ : For Gaussian observation noise  $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I})$  and affine mapping defined by  $\mathbf{B} \in \mathbb{R}^{D \times M}$  and  $\boldsymbol{\mu} \in \mathbb{R}^D$ ,
 
$$\mathbf{x} = \mathbf{B}\mathbf{z} + \boldsymbol{\mu} + \epsilon \in \mathbb{R}^D$$
- Conditional distribution for the links between latent and observed variables
 

$$p(\mathbf{x}|\mathbf{z}, \mathbf{B}, \boldsymbol{\mu}, \sigma^2) = \mathcal{N}(\mathbf{x}|\mathbf{B}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I})$$
- Data point generation: **ancestral sampling**
  - First, sample  $\mathbf{z}_n$  from  $p(\mathbf{z})$
  - Then, use  $\mathbf{z}_n$  to generate a sample  $\mathbf{x}_n \sim p(\mathbf{x}|\mathbf{z}_n, \mathbf{B}, \boldsymbol{\mu}, \sigma^2)$

- Probabilistic model: joint distribution

$$p(\mathbf{x}, \mathbf{z} | \mathbf{B}, \boldsymbol{\mu}, \sigma^2) = p(\mathbf{x} | \mathbf{z}, \mathbf{B}, \boldsymbol{\mu}, \sigma^2) p(\mathbf{z})$$

- Likelihood

$$p(\mathbf{x} | \mathbf{B}, \boldsymbol{\mu}, \sigma^2) = \int p(\mathbf{x} | \mathbf{z}, \mathbf{B}, \boldsymbol{\mu}, \sigma^2) p(\mathbf{z}) d\mathbf{z} = \int \mathcal{N}(\mathbf{x} | \mathbf{B}\mathbf{z} + \boldsymbol{\mu}, \sigma^2 \mathbf{I}) \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}) d\mathbf{z} \\ = \mathcal{N}(\boldsymbol{\mu}, \mathbf{B}\mathbf{B}^T + \sigma^2 \mathbf{I})$$

- Using the property of marginal and conditional Gaussians

L6(5)

L10(7)

April 6, 2021 37 / 42

- The joint Gaussian distribution  $p(\mathbf{x}, \mathbf{z} | \mathbf{B}, \boldsymbol{\mu}, \sigma^2)$  leads us to the posterior distribution

$$p(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z} | \mathbf{m}, \mathbf{C}), \text{ where}$$

$$\mathbf{m} = \mathbf{B}^T (\mathbf{B}\mathbf{B}^T + \sigma^2 \mathbf{I})^{-1} (\mathbf{x} - \boldsymbol{\mu}), \quad \mathbf{C} = \mathbf{I} - \mathbf{B}^T (\mathbf{B}\mathbf{B}^T + \sigma^2 \mathbf{I})^{-1} \mathbf{B}$$

L10(7)

April 6, 2021 38 / 42

## Learning Probabilistic PCA: MLE

## PCA as Linear Auto-Encoder

- For data samples  $\mathbf{X} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ , we are able to compute the likelihood as:

$$\log p(\mathbf{X} | \mathbf{B}, \boldsymbol{\mu}, \sigma^2) = \sum_{n=1}^N \log p(\mathbf{x}_n | \mathbf{B}, \boldsymbol{\mu}, \sigma^2)$$

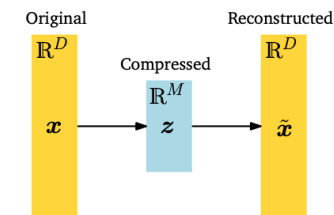
$$\boldsymbol{\mu}_{\text{ML}} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n, \quad \mathbf{B}_{\text{ML}} = \mathbf{U}(\boldsymbol{\Lambda} - \sigma^2 \mathbf{I})^{1/2} \mathbf{R}, \quad \sigma_{\text{ML}}^2 = \frac{1}{D - M} \sum_{j=M+1}^D \lambda_j, \text{ where}$$

- $\mathbf{U}$  is a  $D \times M$  matrix whose columns are eigenvectors of  $\mathbf{S}$
- $\boldsymbol{\Lambda}$  is a  $M \times M$  diagonal matrix whose elements are eigenvalues of  $\mathbf{S}$
- $\mathbf{R}$  is an arbitrary orthogonal matrix (i.e., rotation)

- In the noise-free limit where  $\sigma \rightarrow 0$ , PPCA and PCA provide the identical solution.

L10(7)

April 6, 2021 39 / 42



- **Non-linear auto-encoder**: we replace the linear mapping of PCA with a non-linear mapping. An example is a deep auto-encoder with deep neural networks.
- **(Fully) Bayesian PCA**: place a prior on the model parameters and integrate them out, rather than having a point estimate.
- **Factor analysis**: allow each observation dimension  $d$  to have a different variance  $\sigma_d^2$

L10(7)

April 6, 2021 40 / 42

1)

Questions?