Data Science
Principal Component Analysis

Linear latent decomposition

Stéphane Marchand-Maillet

Department of Computer Science

UNIVERSITÉ DE GENÈVE
FACULTÉ DES SCIENCES

VIPER

Master en Sciences Informatiques - Autumn semester

Table of contents

What is this lecture about?

- ⋆ Base representations may not be optimal (to be defined)
- ⋆ Latent models promise to exhibit the underlying (latent) factors that drive the process in question

- ⋆ This initial (but fundamental) definition of latent factors uses statistical correlation

$\Rightarrow$ It exhibits linear latent factors

$\Rightarrow$ Enables "simplification" of the data by sound decimation

- ⋆ Also: we will study several interpretations

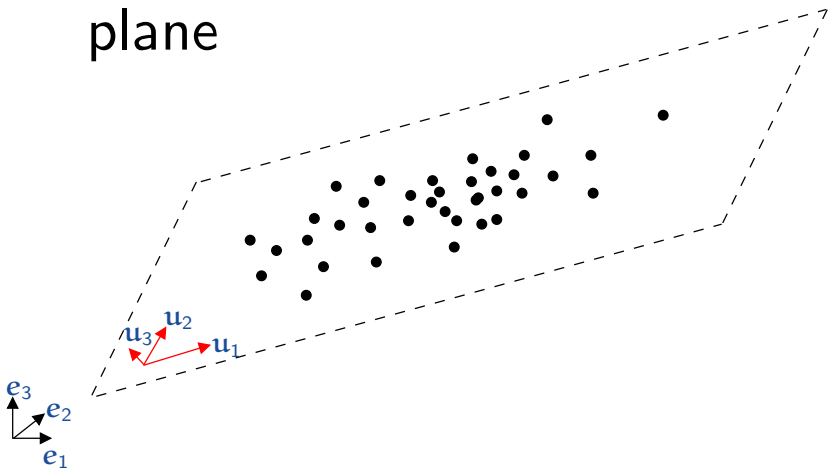
Reading: [1] (chap 12) and [3] (chap 3 and 10.3)

Intuition

Given $\mathcal{X} \subset \Omega$, we wish to decompose Ω into subspaces such that the projection of $\mathcal{X}$ onto these subspaces retains the most "information".

Q: What information should we consider?

- ⋆ Say $\mathcal{X}$ is almost "contained" into a 2D plane in a 3D space
- ⋆ A relevant choice for our subspace is to chose a basis $\{\mathbf{u}_1, \mathbf{u}_2\}$ for the plane



Q: What characterizes $\{\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3\}$?

$\Rightarrow$ the fact that the data varies most along these directions

$\Rightarrow$ the fact that the data varies least orthogonally to these directions ($\mathbf{u}_3$)

Formalization

Given $\mathcal{X}$, $\{\mathbf{u}_i\}_{i \in [\![D]\!]}$ is a new orthonormal basis of $\mathbb{R}^D$. The Principal Component $\mathbf{u}_1$ is chosen such that the variance of the data projected over $\mathbf{u}_1$ is maximum. $\mathbf{u}_2$ is chosen using $\text{Proj}_{\mathbf{u}_1^\perp}(\mathcal{X})$.

Model

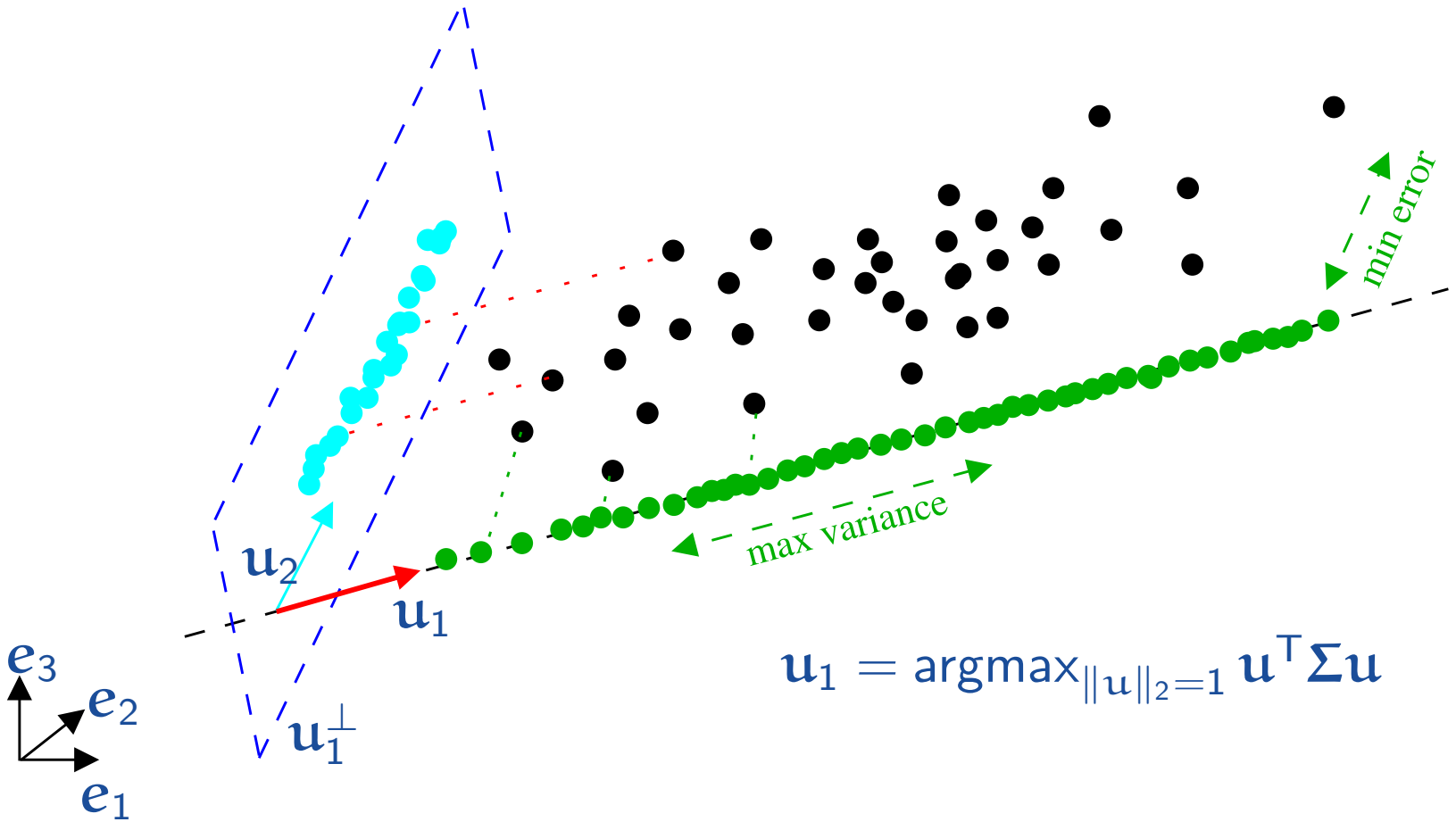
Given $\mathcal{X}$, the sample mean ($\overline{\mathbf{x}}$) and the variance of the data projected over $\mathbf{u}_1$ are

$$\overline{\mathbf{x}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \quad \text{and} \quad \nu_{\mathbf{u}_1} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{u}_1^T \mathbf{x}_i - \mathbf{u}_1^T \overline{\mathbf{x}})^2 = \mathbf{u}_1^T \mathbf{\Sigma} \mathbf{u}_1$$

where $\mathbf{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} (\mathbf{x}_i - \overline{\mathbf{x}})(\mathbf{x}_i - \overline{\mathbf{x}})^T$ is the data covariance matrix. Therefore

$$\mathbf{u}_1 = \underset{\mathbf{u}^T\mathbf{u}=1}{\text{argmax}}\, \mathbf{u}^T \mathbf{\Sigma} \mathbf{u}$$

Intuition



$$\mathbf{u}_1 = \text{argmax}_{\|\mathbf{u}\|_2=1} \mathbf{u}^\top \mathbf{\Sigma} \mathbf{u}$$

Formalization

$$\mathbf{u}_1 = \underset{\mathbf{u}^T\mathbf{u}=1}{\operatorname{argmax}}\, \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} \quad \Rightarrow \quad J(\mathbf{u}) = \mathbf{u}^T\boldsymbol{\Sigma}\mathbf{u} + \lambda(1 - \mathbf{u}^T\mathbf{u})$$

So that

$$\frac{\partial J(\mathbf{u})}{\partial \mathbf{u}}|_{\mathbf{u}=\mathbf{u}_1} = 0 \qquad \text{and} \qquad \frac{\partial J(\mathbf{u})}{\partial \lambda}|_{\lambda=\lambda_1} = 0$$

Hence

$$\boldsymbol{\Sigma}\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \quad \Rightarrow \quad \nu_{\mathbf{u}_1} = \mathbf{u}_1^T\boldsymbol{\Sigma}\mathbf{u}_1 = \lambda_1$$

$\Rightarrow (\mathbf{u}_1, \lambda_1)$ is an eigenpair of the covariance matrix $\boldsymbol{\Sigma}$
$\Rightarrow$ continuing with the decimation process, we obtain the set of Principal Components as the eigenpairs $\{(\mathbf{u}_i, \lambda_i)\}_{i\in[\![D]\!]}$ of the covariance matrix $\boldsymbol{\Sigma}$ of data $\mathcal{X}$

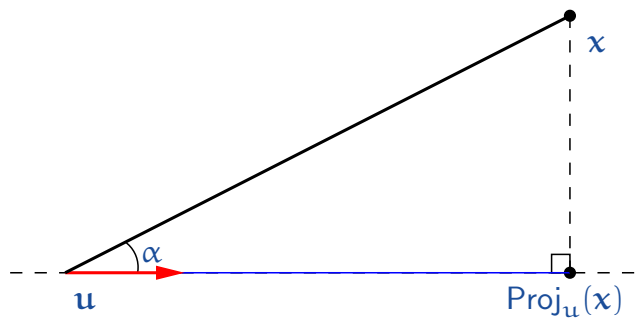
$$\boldsymbol{\Lambda} = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_D \end{pmatrix} = \begin{pmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_D \\ | & & | \end{pmatrix}^T \boldsymbol{\Sigma} \begin{pmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_D \\ | & & | \end{pmatrix} = \mathbf{U}^T\boldsymbol{\Sigma}\mathbf{U}$$

Formalization

- ⋆ The variance $\nu_{\mathbf{u}_1}$ is expressed as a sum of squares
  $\nu_{\mathbf{u}_1} = \frac{1}{N}\sum_{i=1}^{N}(\mathbf{u}_1^T(\mathbf{x}_i - \bar{\mathbf{x}}))^2$
- ⇒ To maximize $\nu_{\mathbf{u}_1}$, terms $\mathbf{u}_1^T(\mathbf{x}_i - \bar{\mathbf{x}})$ should be collectively maximized
- ⇒ Since $(\mathbf{x}_i - \bar{\mathbf{x}})$ is fixed, Pythagoras tells us it is equivalent to minimize the distance to the axis of projection (approximation error)
- ⇒ A Principal Component is a quadratic regression over the data

$$\mathbf{u}_1 = \underset{\mathbf{u}^T\mathbf{u}=1}{\operatorname{argmin}} \sum_{i=1}^{N} \|(\mathbf{x}_i - \bar{\mathbf{x}}) - [\mathbf{u}^T(\mathbf{x}_i - \bar{\mathbf{x}})]\mathbf{u}\|_2^2 \quad \Rightarrow \quad \boldsymbol{\Sigma}\mathbf{u}_1 = \lambda_1\mathbf{u}_1 \qquad (\varphi)$$

Maximize Projection Variance
⇔ Minimize Approximation Error



Physical interpretation

- ⋆ Consider a physical system $\mathcal{X}$ with masses $m_i = 1$ at positions $\mathbf{x_i}$
- ⋆ The inertia of the system w.r.t $\mathbf{a} \in \Omega$ is $I_{\mathbf{a}}(\mathcal{X}) = \sum_{i=1}^{N} d^2(a, x_i)$
- ⋆ Huygens theorem tells us that if $\mathbf{g} = \frac{1}{N} \sum_{i=1} \mathbf{x_i}$ then

$$I_{\mathbf{a}}(\mathcal{X}) = d^2(\mathbf{a}, \mathbf{g}) + I_{\mathbf{g}}(\mathcal{X}) \qquad (\mathrm{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}X)^2)$$

- ⋆ and if $\mathfrak{F}$ is a subspace of Ω going thru $\mathbf{g}$ then

$$I_{\mathfrak{F}}(\mathcal{X}) = \sum_{i=1}^{N} d^2(\mathfrak{F}, \mathbf{x_i}) \quad \text{where} \quad d(\mathfrak{F}, \mathbf{x_i}) = \|x_i - \mathrm{Proj}_{\mathfrak{F}}(\mathbf{x_i})\|$$

$\Rightarrow$ A Principal Component is a subspace of least inertia w.r.t $\mathcal{X}$

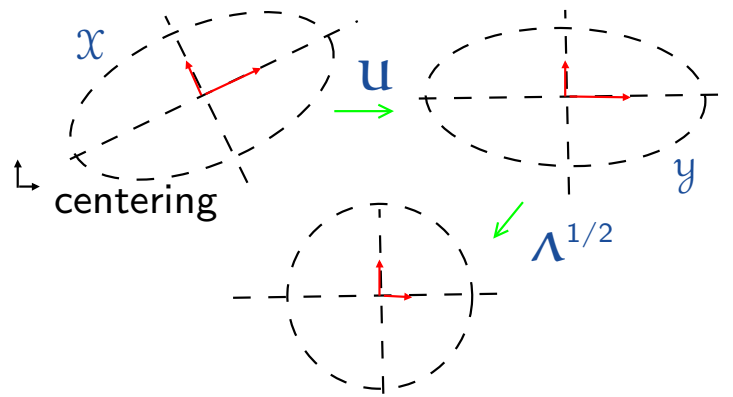
Structure of the latent space

The latent space with basis $\{\mathbf{u}_i, \cdots, \mathbf{u}_D\}$ has the following properties:

- ⋆ By construction $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_D$ and $\lambda \overset{\text{def}}{=} \sum_{d=1}^{D} \lambda_d = \text{Tr}(\mathbf{\Sigma})$
- ⋆ $\nu_{\mathbf{u}_d} = \lambda_d \Rightarrow \lambda$ represents the total variance $\Rightarrow$ latent features are decorrelated $\mathbf{u}_d^T \mathbf{u}_{d'} = 0$ ($\mathbf{\Lambda}$ is the diagonal latent covariance matrix)
- ⋆ The basis $\{\mathbf{u}_i, \cdots, \mathbf{u}_D\}$ induces latent coordinates $\mathbf{y}_i$ for the data:

$$\mathbf{y}_i(d) = \langle \mathbf{u}_d, \mathbf{u}_i - \overline{\mathbf{x}} \rangle = \mathbf{u}_d^T (\mathbf{x}_i - \overline{\mathbf{x}}) \qquad \text{so that} \qquad \mathbf{y}_i = \mathbf{U}^T (\mathbf{x}_i - \overline{\mathbf{x}})$$

- ⋆ The transform is linear and composed of:
  - ◦ Centering on $\overline{\mathbf{x}}$
  - ◦ Rotation using $\mathbf{U}$
  - ◦ Scaling using $\mathbf{\Lambda}^{1/2}$ (whitening)



Structure of the model

- ⋆ The latent space preserves the variance (of the centered data)

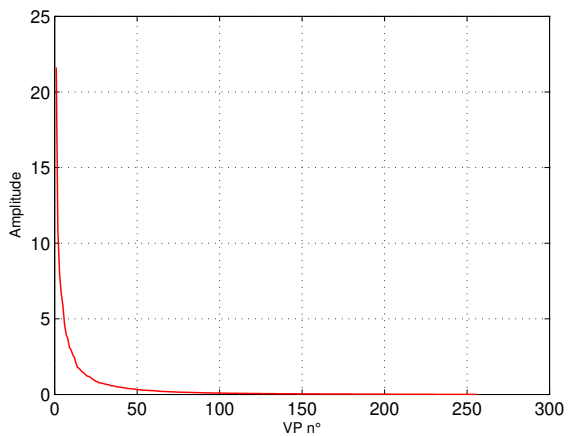
$\Rightarrow$ the underlying data model is $X_i \sim f_X = \mathcal{N}(\bar{x}, \Sigma)$

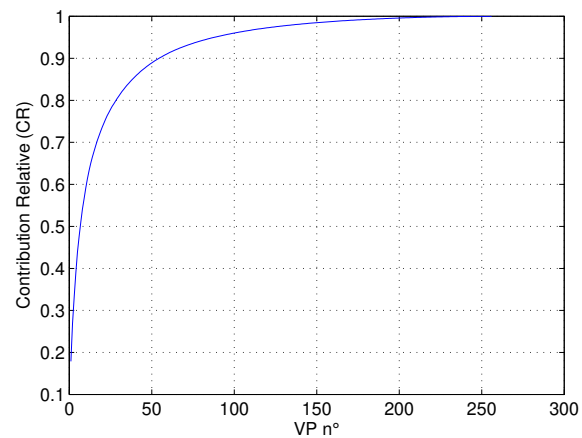
$\Rightarrow$ PCA will not be relevant for non-Gaussian data (e.g clustered)

⚠ Important: So far PCA is an exact transform

$$y_i = U^T(x_i - \bar{x}) \qquad \text{so that} \qquad Uy_i = UU^T(x_i - \bar{x}) = x_i - \bar{x}$$

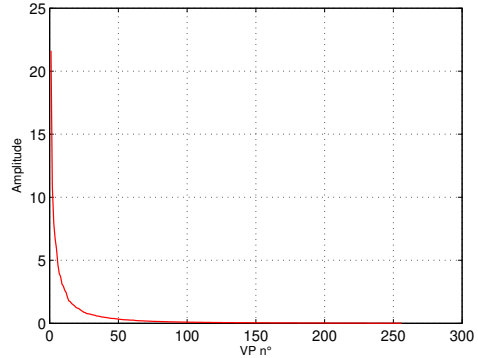
$\Rightarrow$ at this stage, one purpose is to study the spectrum $\{\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_D\}$ of the data



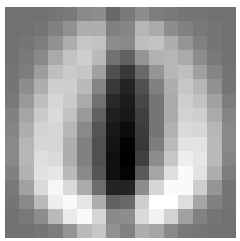


Decomposition via PCA

φ MNIST partial dataset: $N = 7291$ images 16×16 (8bits) $\Rightarrow D = 256$

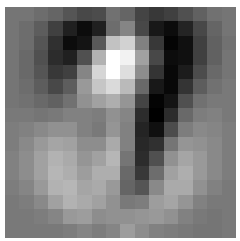


CP n°1



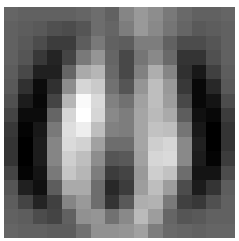
$cr(\Delta_1)$=18%

CP n°2



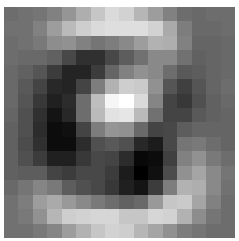
$cr(\Delta_2)$=27%

CP n°3



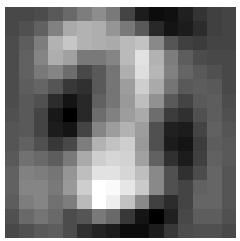
$cr(\Delta_3)$=33%

CP n°4



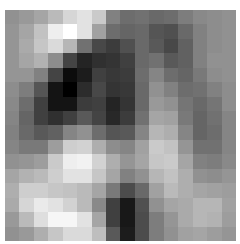
$cr(\Delta_4)$=39%

CP n°5



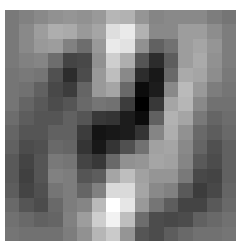
$cr(\Delta_5)$=44%

CP n°6



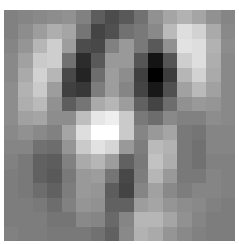
$cr(\Delta_6)$=48%

CP n°7



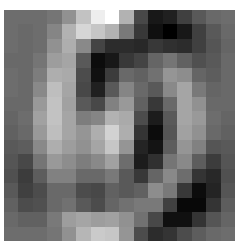
$cr(\Delta_7)$=51%

CP n°8



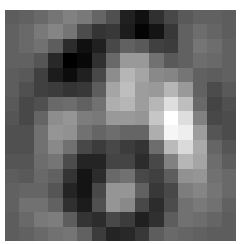
$cr(\Delta_8)$=54%

CP n°9



$cr(\Delta_9)$=57%

CP n°10



$cr(\Delta_{10})$=59%

Approximation via PCA

Low-rank approximation from the Eckart and Young theorem:

If $\boldsymbol{\Sigma} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\mathsf{T}$ and for $K < D$ define $\boldsymbol{\Sigma}_K \overset{\text{def}}{=} \sum_{d=1}^{K} \lambda_d \mathbf{u}_d \mathbf{u}_d^\mathsf{T}$ then

$$\underset{\text{rank}(\mathbf{S})=K}{\text{argmin}} \|\boldsymbol{\Sigma} - \mathbf{S}\|_\mathsf{F}^2 = \boldsymbol{\Sigma}_K \qquad \text{and} \qquad \|\boldsymbol{\Sigma} - \boldsymbol{\Sigma}_K\|_\mathsf{F}^2 = \sum_{d=K+1}^{D} \lambda_d$$

$\Rightarrow$ $\boldsymbol{\Sigma}_K$ is the closest K-rank matrix to $\boldsymbol{\Sigma}$

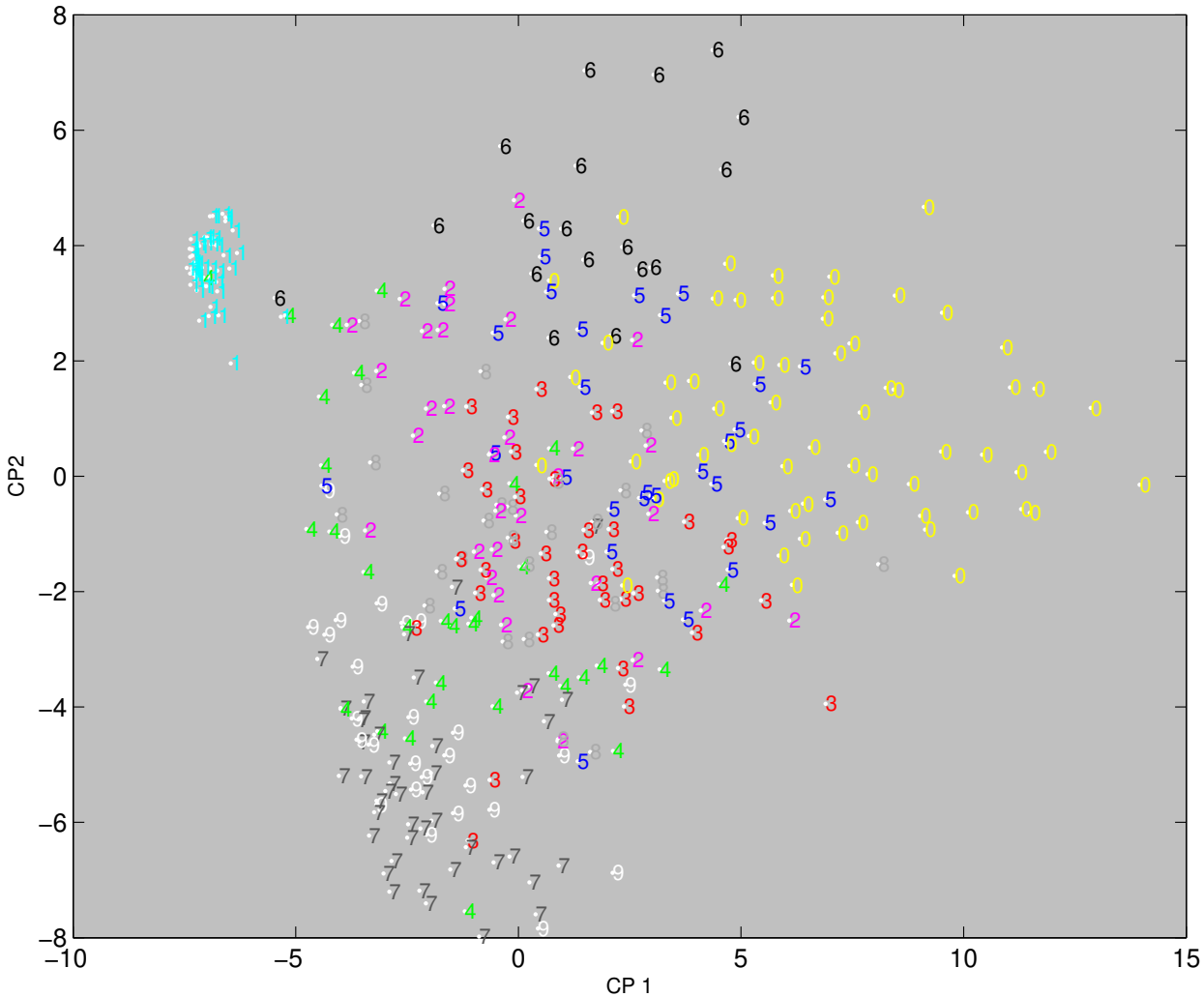
Truncation (of $\mathbf{U}$ and $\boldsymbol{\Lambda}$)

$$\boldsymbol{\Sigma}_K = \begin{pmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_K \\ | & & | \end{pmatrix} \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_K \end{pmatrix} \begin{pmatrix} | & & | \\ \mathbf{u}_1 & \cdots & \mathbf{u}_K \\ | & & | \end{pmatrix}^\mathsf{T} = \mathbf{U}_K \boldsymbol{\Lambda}_K \mathbf{U}_K^\mathsf{T}$$

$$\text{⚠} \quad \Rightarrow \tilde{\mathbf{y}}_i = \mathbf{U}_K^\mathsf{T} \mathbf{x}_i \in \mathbb{R}^K \text{ and } \tilde{\mathbf{x}}_i = \mathbf{U}_K \mathbf{U}_K^\mathsf{T} \mathbf{x}_i + \bar{\mathbf{x}}$$

Visualization via PCA

- MNIST partial dataset: $K = 2 \Rightarrow \tilde{\mathbf{y}}_i \in \mathbb{R}^2$



Geometry of PCA

The quality of reconstruction can be measured by

- ⋆ the relative contribution of each dimension to the variance
  $c_d = \frac{\lambda_d}{\sum_k \lambda_k}$

⇒ depends on the distribution of the spectrum

- ⋆ the projection ratio of each data $\mathbf{x}_i$ over a latent factor
  $\rho_d(\mathbf{x}_i) = \frac{\langle \mathbf{u}_d, \mathbf{x}_i \rangle^2}{\|\mathbf{x}\|^2} = \cos^2(\angle(\mathbf{u}_d, \mathbf{x}_i))$

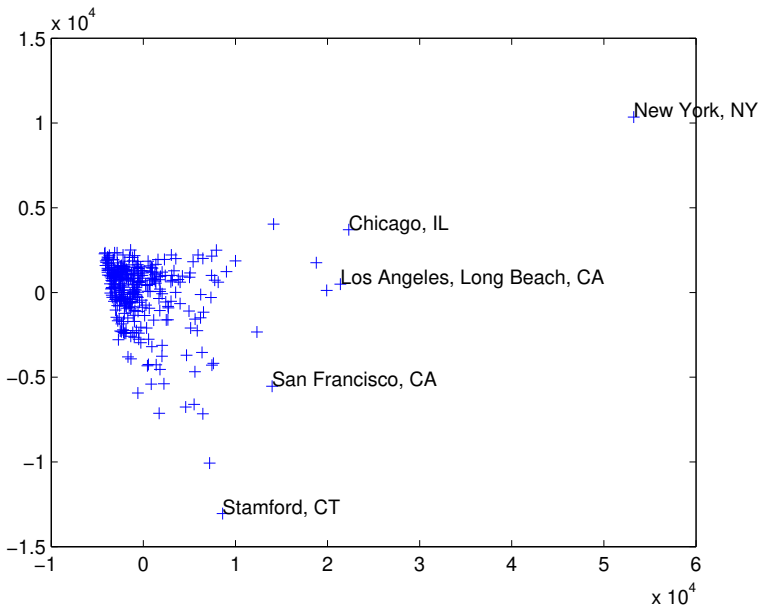
⇒ the closer $\rho_d(\mathbf{x}_i)$ is to 1, the more $\mathbf{x}_i$ lies on $\mathbf{u}_d$

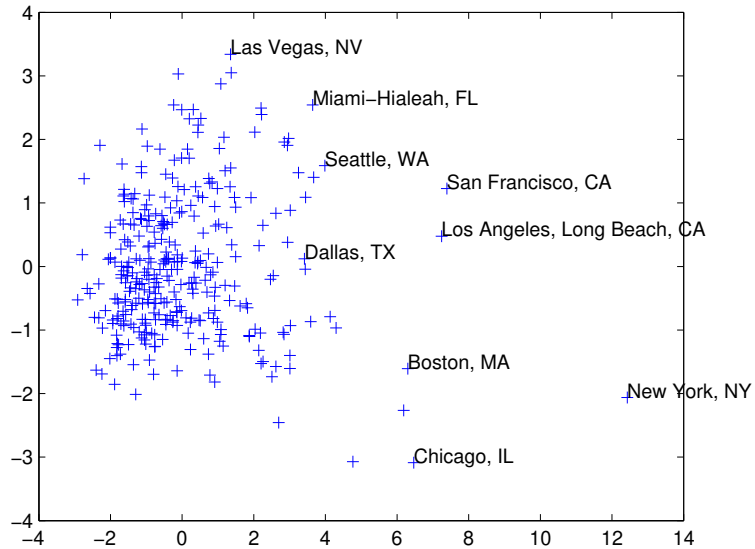
⇒ The above can be grouped (summed) to evaluate wrt a subspace $\{\mathbf{u}_1, \mathbf{u}_2, \cdots\}$

Geometry of PCA

Every original data comes with its unit (scale), that we can estimate via σ_d^2 the sample variance along original dimension d. PCA is more effective is all scales are similar.

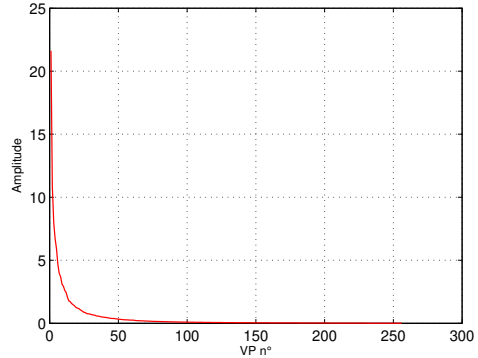
$\Rightarrow$ we create the scaling matrix $S = \text{diag}[\sigma_1^2, \cdots, \sigma_d^2]$ and we define the metric $d_S^2(x, y) = (x - y)^T S^{-1}(x - y) \Rightarrow$ in that metric space, the covariance matrix Σ_S is also rescaled (into the correlation matrix) and used as a base for PCA.

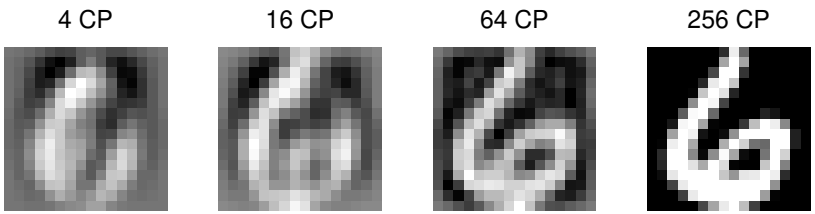


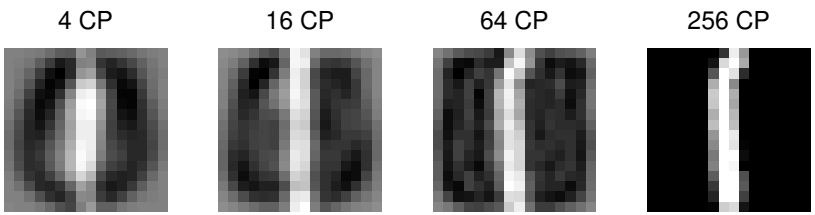


Approximation via PCA

φ MNIST partial dataset: $N = 7291$ images 16×16 (8bits) $\Rightarrow D = 256$



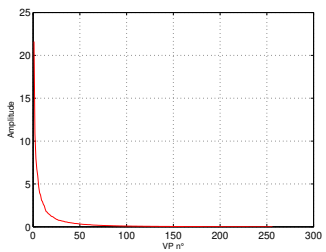


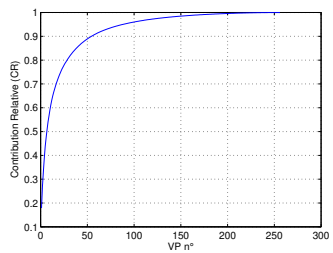


Practical PCA

Given $\mathcal{X} \in \Omega$

- ⋆ Compute the sample mean $\bar{\mathbf{x}}$
- ⋆ Center the data $\mathbf{x}_i \leftarrow (\mathbf{x}_i - \bar{\mathbf{x}})$ and form centered data matrix $\mathbf{X}$
- ⋆ $\mathbf{\Sigma} = \frac{1}{N}\mathbf{X}\mathbf{X}^T$ and $\mathbf{\Sigma} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T$ ← ⚠ Exact transform so far
- ⋆ Select the number of components K
- ⋆ Define $\mathbf{U}_K$, $\mathbf{\Lambda}_K$ and compute $\{\tilde{\mathbf{y}}_i\}_{i\in[\![N]\!]}$ and/or $\{\tilde{\mathbf{x}}_i\}_{i\in[\![N]\!]}$





Choice of K

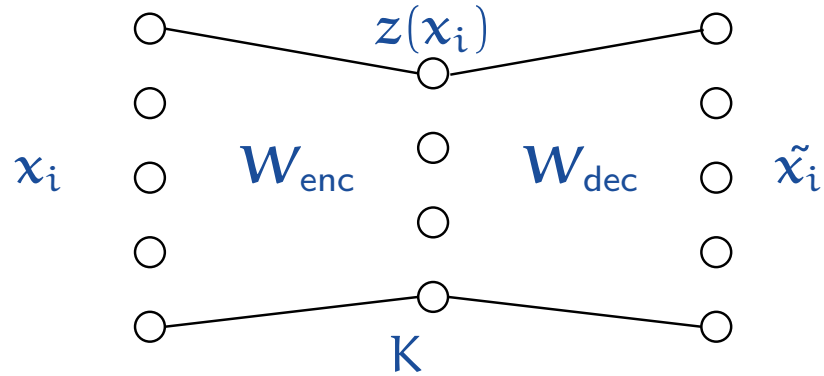
1. $K = d$, the target dimension $\Rightarrow \tilde{\mathbf{y}}_i \in \mathbb{R}^d$
2. Require $\text{Var}(\mathcal{Y}) = \tau.\text{Var}(\mathcal{X})$ $\Rightarrow$ K such that $\frac{\sum_{d=1}^{K}\lambda_d}{\sum_{d=1}^{D}\lambda_d} \geqslant \tau$
3. Train K such that $\mathcal{L}(\mathcal{X}) \leqslant \varepsilon$ (e.g $\mathcal{L}(\mathcal{X}) = \sum_i \|\mathbf{x}_i - \tilde{\mathbf{x}}_i\|^2$)

PCA and linear AutoEncoders

A linear AE is a simple structure

- ⋆ $\boldsymbol{W}_{\text{enc}}$ and $\boldsymbol{W}_{\text{dec}}$: encoder and decoder weights

$$\boldsymbol{z}(\boldsymbol{x}_i) = \boldsymbol{W}_{\text{enc}}\boldsymbol{x}_i \quad \tilde{\boldsymbol{x}}_i = \boldsymbol{W}_{\text{dec}}\boldsymbol{z}(\boldsymbol{x}_i)$$



We optimize the weight matrices

$$\theta^* = \underset{\boldsymbol{W}_1,\boldsymbol{W}_2}{\text{argmin}} \sum_{i=1}^{N} \frac{1}{2} \sum_{d=1}^{D} (\boldsymbol{x}_{i[d]} - \tilde{\boldsymbol{x}}_{i[d]})^2 = \underset{\text{rank}(\boldsymbol{W})=K}{\text{argmin}} \|\boldsymbol{X} - \boldsymbol{W}\boldsymbol{Z}\|_F^2$$

Using again the Eckart and Young theorem with $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{V}^T$ then the solution is $\boldsymbol{W}_{\text{dec}}\boldsymbol{Z} = \boldsymbol{U}_K\boldsymbol{\Psi}_K\boldsymbol{V}_K^T$.

Setting $\boldsymbol{W}_{\text{dec}} = \boldsymbol{U}_K\boldsymbol{\Psi}_K$, then clearly $\boldsymbol{W}_{\text{enc}} = \boldsymbol{\Psi}_K^{-1}\boldsymbol{U}_K^T$ ℗

$\boldsymbol{W}_{\text{enc}}\boldsymbol{X} = \boldsymbol{Z} = \boldsymbol{V}^T = \boldsymbol{V}^T(\boldsymbol{X}^T\boldsymbol{X})^{-1}(\boldsymbol{X}^T\boldsymbol{X}) = \boldsymbol{V}^T(\boldsymbol{V}\boldsymbol{\Psi}\boldsymbol{\Psi}\boldsymbol{V}^T)^{-1}(\boldsymbol{V}\boldsymbol{\Psi}\boldsymbol{U}^T)\boldsymbol{X} = \boldsymbol{\Psi}_K^{-1}\boldsymbol{U}_K^T\boldsymbol{X}$

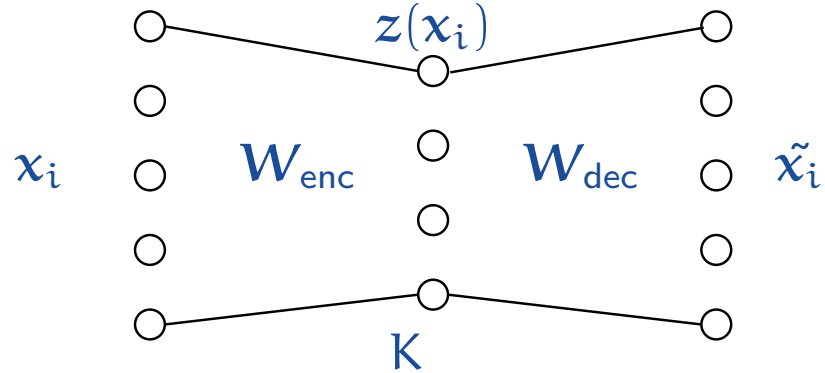
Note: $\boldsymbol{W}_{\text{dec}} = \boldsymbol{W}_{\text{dec}} = \text{Id}_D$ cannot be a solution if $K < D$ and $\boldsymbol{W}_{\text{enc}}$ is not the inverse of $\boldsymbol{W}_{\text{dec}}$

PCA and linear AutoEncoders

A linear AE is a simple structure

- $\boldsymbol{W}_{\text{enc}}$ and $\boldsymbol{W}_{\text{dec}}$: encoder and decoder weights

$$\boldsymbol{z}(\boldsymbol{x}_i) = \boldsymbol{W}_{\text{enc}}\boldsymbol{x}_i \quad \tilde{\boldsymbol{x}}_i = \boldsymbol{W}_{\text{dec}}\boldsymbol{z}(\boldsymbol{x}_i)$$



We optimize the weight matrices

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{V}^T \text{ and } \boldsymbol{W}_{\text{dec}} = \boldsymbol{U}\boldsymbol{\Psi} \text{ and } \boldsymbol{W}_{\text{enc}} = \boldsymbol{\Psi}^{-1}\boldsymbol{U}^T \Rightarrow \tilde{\boldsymbol{x}}_i = \boldsymbol{U}_K\boldsymbol{U}_K^T\boldsymbol{x}_i$$

Relation to PCA (centered data)

- PCA: $\boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ so that $\tilde{\boldsymbol{x}}_i = \boldsymbol{U}_K\boldsymbol{U}_K^T\boldsymbol{x}_i$
- AE: $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Psi}\boldsymbol{V}^T \Rightarrow \boldsymbol{\Sigma} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T = \frac{1}{N}\boldsymbol{U}\boldsymbol{\Psi}^2\boldsymbol{U}^T$ and $\tilde{\boldsymbol{x}}_i = \boldsymbol{U}_K\boldsymbol{U}_K^T\boldsymbol{x}_i$

$\Rightarrow \boldsymbol{\Lambda} = \frac{1}{N}\boldsymbol{\Psi}^2 = (\frac{1}{\sqrt{N}}\boldsymbol{\Psi})(\frac{1}{\sqrt{N}}\boldsymbol{\Psi})$ so that $\text{PCA}(\boldsymbol{X}) \leftrightarrow \text{AE}(\frac{1}{\sqrt{N}}\boldsymbol{X})$

$\Rightarrow$ A linear AE performs a PCA if the data is centered and scaled

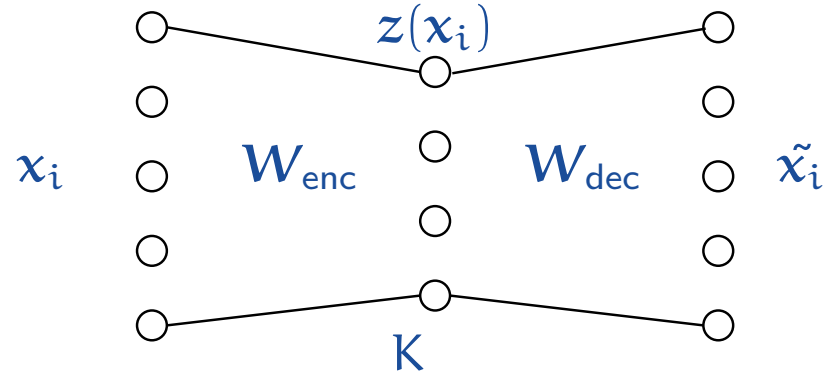
Note: $\boldsymbol{W}_{\text{dec}} = \boldsymbol{W}_{\text{dec}} = \text{Id}_D$ cannot be a solution if $K < D$ and $\boldsymbol{W}_{\text{enc}}$ is not the inverse of $\boldsymbol{W}_{\text{dec}}$

PCA and linear AutoEncoders

A linear AE is a simple structure

- ⋆ $\boldsymbol{W}_{\text{enc}}$ and $\boldsymbol{W}_{\text{dec}}$: encoder and decoder weights

$$\boldsymbol{z}(\boldsymbol{x}_i) = \boldsymbol{W}_{\text{enc}}\boldsymbol{x}_i \quad \tilde{\boldsymbol{x}}_i = \boldsymbol{W}_{\text{dec}}\boldsymbol{z}(\boldsymbol{x}_i)$$



We optimize the weight matrices

$\Rightarrow$ A linear AE performs a PCA if the data is centered and scaled

- ⋆ If adding hidden layers, the latent space becomes non-linear

$\Rightarrow$ non-linear AutoEncoders

- ⋆ Changing the loss function (ELBO) enables sparsity in the latent space

$\Rightarrow$ Variational AutoEncoders (VAE)

Note: $\boldsymbol{W}_{\text{dec}} = \boldsymbol{W}_{\text{dec}} = \text{Id}_{\text{D}}$ cannot be a solution if $\text{K} < \text{D}$ and $\boldsymbol{W}_{\text{enc}}$ is not the inverse of $\boldsymbol{W}_{\text{dec}}$

Alternative formulations

- ⋆ Probabilistic PCA reformulates PCA with an explicit latent distribution $\mathbb{P}(z) = \mathcal{N}(0, \mathrm{Id}_K)$ and the conditional model is $\mathbb{P}(x|z) = \mathcal{N}(Wz + \mu + \sigma^2 \mathrm{Id}_D)$ so that
  - the model accomodates "measurement noise" (with variance σ^2)
  - the model can run in a generative mode
  - parameters can be estimated via Maximum Likelihood
  - an EM algorithm (see later) can be derived for saving computations
  - Bayesian PCA reverts the conditional so as to find K by training
- ⋆ Kernel PCA embarks a nonlinear mapping $\phi(x_i)$ via a kernel function $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ to perform PCA within a more favorable space
- ⋆ Local PCA perform PCA on data neighborhoods to consider the local intrinsic dimensionality only

⚠ The limitation of PCA is often the decomposition of Σ in $O(D^3)$

Summary

- ⋆ PCA is part of the linear latent models
- ⋆ PCA applies on centered data and uses variance as a criterai for decomposition
- ⋆ PCA is an exact complete decomposition into decorrelated components
- ⋆ PCA assumes a Normal distribution of the data
- ⋆ PCA can be equivalently formulated as a regression
- ⋆ PCA offers a sound decimation strategy based on a low rank approximation
- ⋆ PCA can be used for denoising via a Gaussian noise model
- ⋆ PCA is equivalent to a linear AutoEncoder
- ⋆ PCA may be given a stochastic formulation
- ⋆ PCA may be generalized to the non-linear case via kernels

Example questions [mostly require formal – mathematical – answers]

- ⋆ Explain how PCA uses variance as a criterion
- ⋆ Show that variance maximization is equivalent to error minimization
- ⋆ Show how PCA uses the Eckart-Young theorem
- ⋆ Given some data, how do you apply PCA?
- ⋆ What information does it provide you with?
- ⋆ How do you reconstruct data with $K < D$ components?
- ⋆ How do you select the components to keep?
- ⋆ Can you apply PCA on any data?
- ⋆ Is it relevant to apply PCA on any data?
- ⋆ How can I apply PCA over clustered data?
- ⋆ Show that PCA is equivalent to a linear AE

Note: Make sure you can explain in detail what is: linear transform, orthogonal matrix, coordinate, rank, mean, variance, projection, eigen decomposition, trace, Frobenius norm, Lagrange Multiplier

References I

[1] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. (available online).

[2] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020. (available online).

[3] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley, New York, 2 edition, 2001.

License