Principal Component Analysis

CoE197M/EE298M (Foundations of Machine Learning)

Rowel Atienza, Ph.D.

rowel@eee.upd.edu.ph

Reference: "Mathematics for Machine Learning". Copyright 2020 by Marc Peter Deisenroth, A. Aldo Faisal, and Cheng Soon Ong. Published by Cambridge University Press.

Motivations

Raw data representations are over-complete

Dimensionality reduction reduces the footprint of data without losing useful important information

Problem Statement

Consider a dataset $\mathcal{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\}$, $\boldsymbol{x}_n \in \mathbb{R}^D$ with mean $\mathbf{0}$ and data covariance matrix:

$$\boldsymbol{S} = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^T$$

There exists a low-dimensional compression representation (code) of $\boldsymbol{x}_n$:

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

The projection matrix:

$$\boldsymbol{B} := [\boldsymbol{b}_1, \dots, \boldsymbol{b}_M] \in \mathbb{R}^{D \times M}$$

With orthonormal basis $\boldsymbol{b}_i^T \boldsymbol{b}_j = 0$ with $i \neq j$ and $\boldsymbol{b}_i^T \boldsymbol{b}_i = 1$

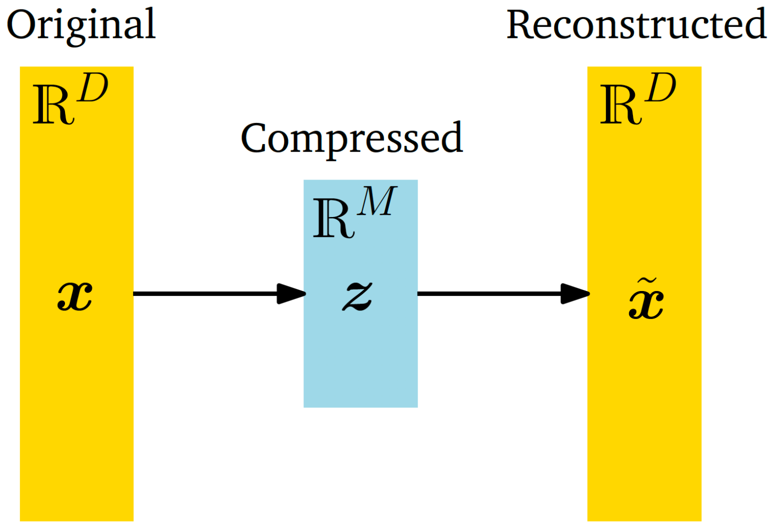
We project $\boldsymbol{x}_n$ into a low-dimensional subspace $U \subseteq \mathbb{R}^D$ with $dim(U) = M < D$

The projected data is $\widetilde{\boldsymbol{x}}_n$ with $\boldsymbol{z}_n$ as the coordinates on basis $\boldsymbol{B}$

Dimensionality Reduction

The objective is to find $\tilde{\boldsymbol{x}}_n \in \mathbb{R}^D$ or $\boldsymbol{z}_n = [z_{1n}, \dots, z_{Mn}]^T \in \mathbb{R}^{M\times 1}$ and basis $\boldsymbol{B} = [\boldsymbol{b}_1, \dots, \boldsymbol{b}_M] \in \mathbb{R}^{D\times M}$ that minimizes the loss due to compression

Example loss: squared reconstruction loss $||\boldsymbol{x}_n - \tilde{\boldsymbol{x}}_n||^2$



Encoder

Decoder

Finding Projective Coordinates

Projective Perspective

Assume ONB $B = (\boldsymbol{b}_1, \dots, \boldsymbol{b}_D) \in \mathbb{R}^D$

$$\boldsymbol{x} = \underbrace{\sum_{i=1}^{M} k_i \boldsymbol{b}_i}_{\tilde{\boldsymbol{x}}_n \in U \subseteq \mathbb{R}^D} + \sum_{j=M+1}^{D} k_j \boldsymbol{b}_j$$

where $k_i \in \mathbb{R}$

Objective Function

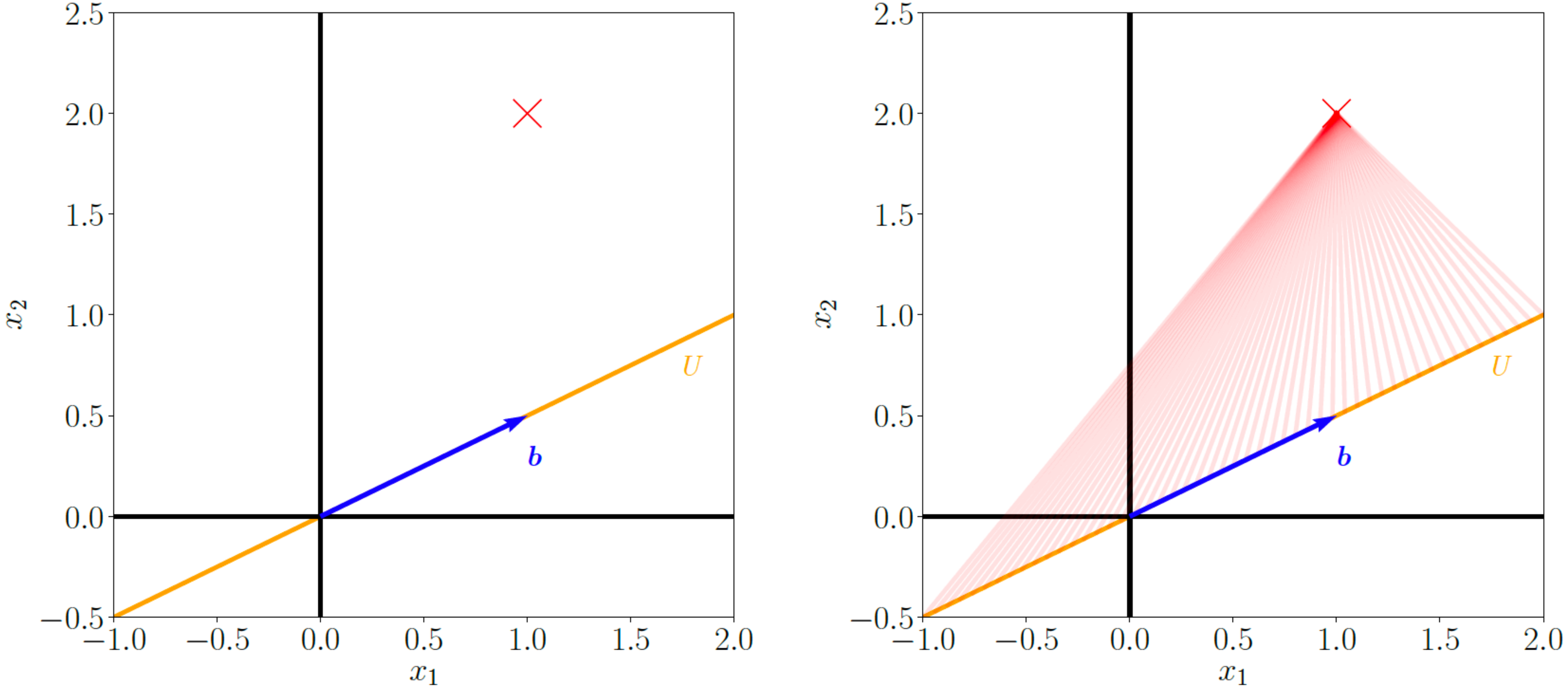
Minimize the average Euclidean reconstruction error:

$$J_M := \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n\|^2$$

Where

$$\widetilde{\boldsymbol{x}}_n = \sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m = \boldsymbol{B}\boldsymbol{z}_n \in U \subseteq \mathbb{R}^D$$

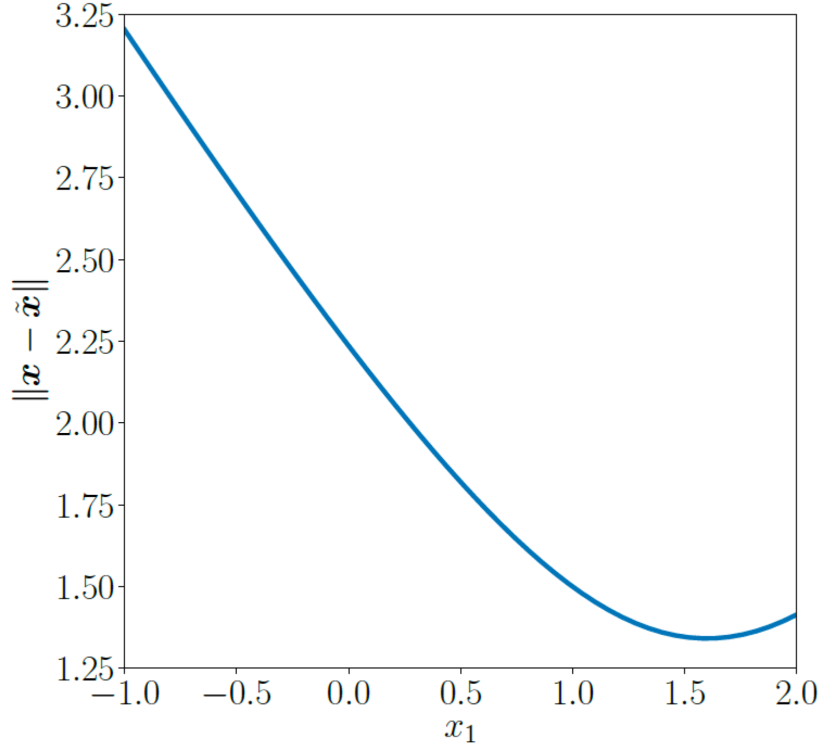
Visualizing Distance Minimization



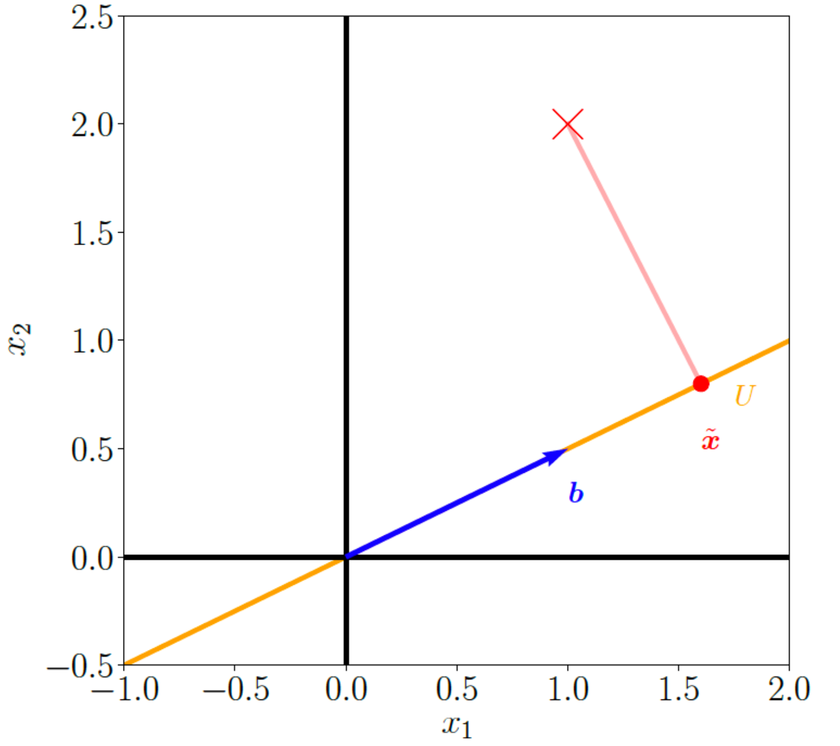
(a) Setting.

(b) Differences $\boldsymbol{x} - \tilde{\boldsymbol{x}}_i$ for 50 different $\tilde{\boldsymbol{x}}_i$ are shown by the red lines.

Visualizing Distance Minimization



(a) Distances $\|\boldsymbol{x} - \tilde{\boldsymbol{x}}\|$ for some $\tilde{\boldsymbol{x}} = z_1 \boldsymbol{b} \in U = \mathrm{span}[\boldsymbol{b}]$; see panel (b) for the setting.



(b) The vector $\tilde{\boldsymbol{x}}$ that minimizes the distance in panel (a) is its orthogonal projection onto U. The coordinate of the projection $\tilde{\boldsymbol{x}}$ with respect to the basis vector $\boldsymbol{b}$ that spans U is the factor we need to scale $\boldsymbol{b}$ in order to "reach" $\tilde{\boldsymbol{x}}$.

Optimal Coordinates

Find coordinates of $\boldsymbol{z}$ of $\widetilde{\boldsymbol{x}}_n$ for $n = 1, \ldots, N$

Assume, ONB $[\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M]$ of $U \subseteq \mathbb{R}^D$

Such that $\|\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n\|^2$ is minimized

Assume we are given dataset: $\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ where $\boldsymbol{x}_n \in \mathbb{R}^D$ and $\mathbb{E}[\mathcal{X}] = 0$ or all data are zero centered

Zero Centering

Assuming the mean $\mathbb{E}[\mathcal{X}] = \boldsymbol{\mu}$

$$\boldsymbol{x}_n = \boldsymbol{x}_n - \boldsymbol{\mu}$$

Optimal Coordinates

$$\frac{\partial J_M}{\partial z_{in}} = \frac{\partial J_M}{\partial \widetilde{\boldsymbol{x}}_n} \frac{\partial \widetilde{\boldsymbol{x}}_n}{\partial z_{in}}$$

$$\frac{\partial J_M}{\partial \widetilde{\boldsymbol{x}}_n} = -\frac{2}{N}(\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n)^T \in \mathbb{R}^{1\times D}$$

$$\frac{\partial \widetilde{\boldsymbol{x}}_n}{\partial z_{in}} = \frac{\partial}{\partial z_{in}}\left(\sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m\right) = \boldsymbol{b}_i \in \mathbb{R}^{D\times 1}$$

Optimal Projection z_{in}

$$\frac{\partial J_M}{\partial z_{in}} = -\frac{2}{N}(\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n)^T \boldsymbol{b}_i = -\frac{2}{N}\left(\boldsymbol{x}_n - \sum_{m=1}^{M} z_{mn}\boldsymbol{b}_m\right)^T \boldsymbol{b}_i$$

$$\frac{\partial J_M}{\partial z_{in}} = -\frac{2}{N}\left(\boldsymbol{x}_n^T\boldsymbol{b}_i - z_{in}\boldsymbol{b}_i^T\boldsymbol{b}_i\right)$$

$$\frac{\partial J_M}{\partial z_{in}} = -\frac{2}{N}(\boldsymbol{x}_n^T\boldsymbol{b}_i - z_{in}) = 0$$

$$z_{in} = \boldsymbol{x}_n^T\boldsymbol{b}_i = \boldsymbol{b}_i^T\boldsymbol{x}_n$$

Observations

The optimal projection $\tilde{\boldsymbol{x}}_n$ of $\boldsymbol{x}_n$ is an orthogonal projection

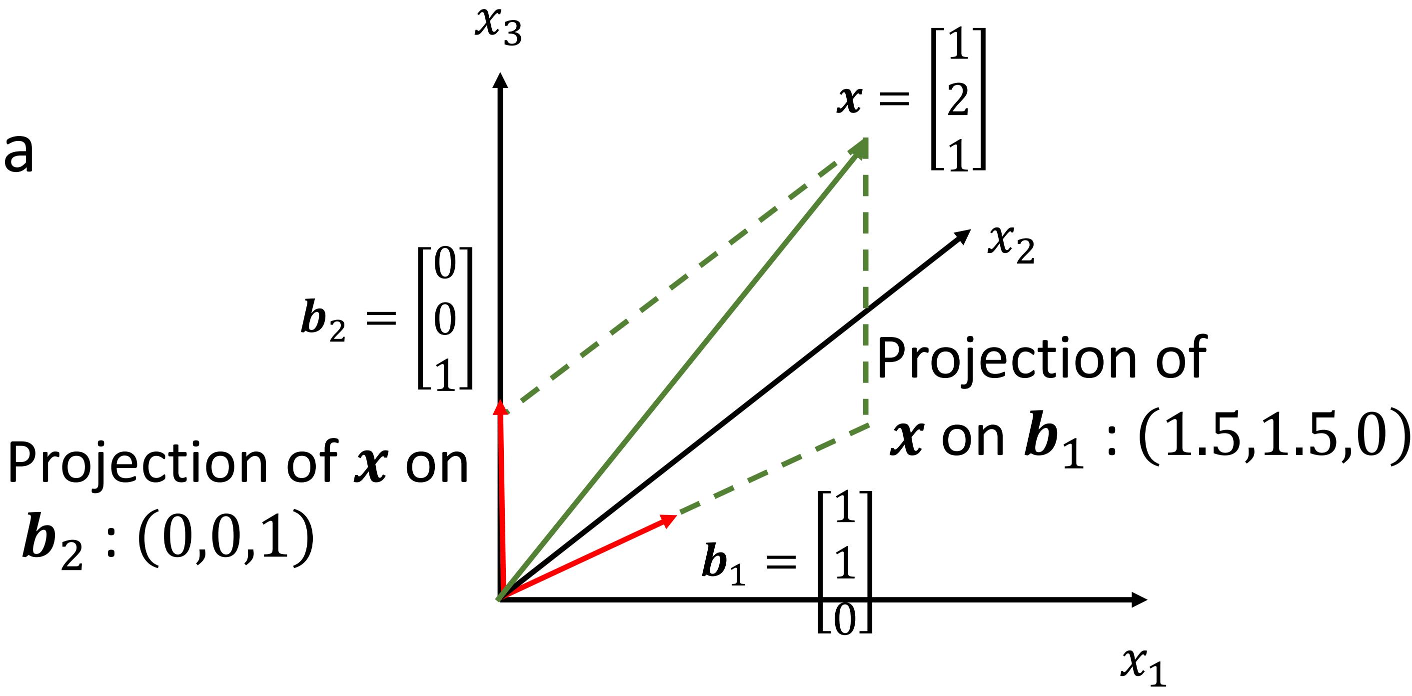
The coordinates of $\tilde{\boldsymbol{x}}_n$ with respect to $[\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M]$ are the coordinates of the orthogonal projection of $\tilde{\boldsymbol{x}}_n$ on the principal subspace

An orthogonal projection is the best linear mapping given the objective

The coordinates z_{in} for $i = 1, \ldots, m$ must be the same as k_{in} for $i = 1, \ldots, m$

PCA on 2-dim

The coordinates of $\boldsymbol{x}$ on $\boldsymbol{b}_1$ has a length $\boldsymbol{z}_1 = \sqrt{5}$.



Projection of $\boldsymbol{x}$ on $\boldsymbol{b}_1$ has length $\sqrt{1+2^2} = \sqrt{5}$.
This is split to:
$x_1 = \sqrt{5}\cos\frac{\pi}{4} = 1.5$ and $x_2 = \sqrt{5}\sin\frac{\pi}{4} = 1.5$

Orthogonal Projection with ONB

Recall projection of a vector $\boldsymbol{x} \in \mathbb{R}^n$ onto U that is closest to $\boldsymbol{x}$ is $\pi_U(\boldsymbol{x})$ with a basis vector $\boldsymbol{b} \in \mathbb{R}^n$

$$\pi_U(\boldsymbol{x}) = \lambda \boldsymbol{b} = \boldsymbol{b}\lambda = \boldsymbol{b}\frac{\boldsymbol{b}^T\boldsymbol{x}}{\|\boldsymbol{b}\|^2} = \frac{\boldsymbol{b}\boldsymbol{b}^T}{\|\boldsymbol{b}\|^2}\boldsymbol{x} = \boldsymbol{P}_\pi \boldsymbol{x}$$

Then:

$$\tilde{\boldsymbol{x}}_n = \boldsymbol{b}_i \underbrace{\boldsymbol{b}_i^T \boldsymbol{x}_n}_{z_{in} = \boldsymbol{b}_i^T \boldsymbol{x}_n}$$

General ONB

Assume, ONB $\boldsymbol{B} = [\boldsymbol{b}_1, \dots, \boldsymbol{b}_M]$ of $U \subseteq \mathbb{R}^D$

$$\widetilde{\boldsymbol{x}}_n = \boldsymbol{B}\boldsymbol{B}^T \boldsymbol{x}_n$$

$\boldsymbol{B}^T \boldsymbol{x}_n$ is the projection of $\boldsymbol{x}_n$ on ONB

Note that $\widetilde{\boldsymbol{x}}_n \in \mathbb{R}^D$ but our coordinates $[z_1, \dots, z_M]$ with respect to basis vectors $[\boldsymbol{b}_1, \dots, \boldsymbol{b}_M]$ is of dimensions $M < D$

The other coordinates $[z_{M+1}, \dots, z_D]$ with respect to basis vectors $[\boldsymbol{b}_{M+1}, \dots, \boldsymbol{b}_D]$ have zero values

Finding ONB

Find the Basis $\boldsymbol{b}_1, \ldots, \boldsymbol{b}_M$

$$\tilde{\boldsymbol{x}}_n = \sum_{m=1}^{M} z_{mn} \boldsymbol{b}_m$$

$$\tilde{\boldsymbol{x}}_n = \sum_{m=1}^{M} (\boldsymbol{x}_n^T \boldsymbol{b}_m) \boldsymbol{b}_m$$

$$\tilde{\boldsymbol{x}}_n = \left(\sum_{m=1}^{M} \boldsymbol{b}_m \boldsymbol{b}_m^T \right) \boldsymbol{x}_n$$

$$\boldsymbol{x}_n = \sum_{m=1}^{M} z_m \boldsymbol{b}_m + \sum_{j=M+1}^{D} z_j \boldsymbol{b}_j$$

$$\boldsymbol{x}_n = \left(\sum_{m=1}^{M} \boldsymbol{b}_m \boldsymbol{b}_m^T \right) \boldsymbol{x}_n + \left(\sum_{j=M+1}^{D} \boldsymbol{b}_j \boldsymbol{b}_j^T \right) \boldsymbol{x}_n$$

Therefore,

$$\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n = \left(\sum_{j=M+1}^{D} \boldsymbol{b}_j \boldsymbol{b}_j^T \right) \boldsymbol{x}_n = \sum_{j=M+1}^{D} \left(\boldsymbol{x}_n^T \boldsymbol{b}_j \right) \boldsymbol{b}_j$$

Observation on $\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n = \sum_{j=M+1}^{D} \left(\boldsymbol{x}_n^T \boldsymbol{b}_j \right) \boldsymbol{b}_j$

The difference is exactly the projection of the data point on the orthogonal complement of the principal subspace

Maximum Variance

Project to low-dimensional subspace while maximizing variance to retain as much information as possible

Finding the 1st Basis Vector $\boldsymbol{b}_1 \in \mathbb{R}^D$

Assumind i.i.d., maximize the variance of the first coordinate z_1 of $\boldsymbol{z} \in \mathbb{R}^M$:

$$V_1 = \mathbb{V}[z_1] = \frac{1}{N}\sum_{n=1}^{N} z_{1n}^2$$

Where

$$z_{1n} = \boldsymbol{b}_1^T \boldsymbol{x}_n$$

$$V_1 = \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{b}_1^T\boldsymbol{x}_n)^2 = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{b}_1^T\boldsymbol{x}_n\,\boldsymbol{x}_n^T\boldsymbol{b}_1$$

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

Where $\boldsymbol{S}$ is the data covariance matrix defined earlier.

We restrict $||\boldsymbol{b}_1||^2 = 1$ so that the variance comes from $\boldsymbol{S}$ only

Direction of Maximum Variance

$$\max_{\boldsymbol{b}_1} \boldsymbol{b}_1^T \boldsymbol{S} \boldsymbol{b}_1$$

Subject to: $\|\boldsymbol{b}_1\|^2 = 1$

The Lagrange:

$$\mathfrak{L}(\boldsymbol{b}_1, \lambda) = \boldsymbol{b}_1^T \boldsymbol{S} \boldsymbol{b}_1 + \lambda(1 - \boldsymbol{b}_1^T \boldsymbol{b}_1)$$

The partial derivative:

$$\frac{d\mathfrak{L}}{d\boldsymbol{b}_1} = 2\boldsymbol{b}_1^T \boldsymbol{S} - 2\lambda \boldsymbol{b}_1^T$$

$$\frac{d\mathfrak{L}}{d\lambda} = (1 - \boldsymbol{b}_1^T \boldsymbol{b}_1)$$

$2\boldsymbol{b}_1^T\boldsymbol{S} - 2\lambda\boldsymbol{b}_1^T = 0$ and $1 - \boldsymbol{b}_1^T\boldsymbol{b}_1 = 0$:

$$\boldsymbol{b}_1^T\boldsymbol{S} = \lambda\boldsymbol{b}_1^T \text{ or } \boldsymbol{S}\boldsymbol{b}_1 = \lambda\boldsymbol{b}_1$$
$$\boldsymbol{b}_1^T\boldsymbol{b}_1 = 1$$

Rewriting:

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

The variance is equal to the eigenvalue.

To maximize the variance, we choose the eigenvector as the basis vector with the maximum eigenvalue.

This eigenvector $\boldsymbol{b}_1$ is called first principal component

What we have so far…

We have one basis vector $\boldsymbol{b}_1$ which is the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{S}$

Problem: We need $m - 1$ more basis vectors $\boldsymbol{b}_2, \ldots, \boldsymbol{b}_m$

Finding the 2nd Basis Vector $\boldsymbol{b}_2 \in \mathbb{R}^D$

Subtract the effect of the first principal component $\boldsymbol{b}_1$ from the data:

$$\widehat{\boldsymbol{x}}_n = \boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_1 = \boldsymbol{x}_n - \boldsymbol{b}_1 \boldsymbol{b}_1^T \boldsymbol{x}_n$$

Then we can use the same argument:

$$V_2 = \boldsymbol{b}_2^T \left(\frac{1}{N} \sum_{n=1}^{N} \widehat{\boldsymbol{x}}_n \widehat{\boldsymbol{x}}_n^T \right) \boldsymbol{b}_2 = \boldsymbol{b}_2^T \widehat{\boldsymbol{S}} \boldsymbol{b}_2$$

The maximum variance is achieved at

$$\widehat{\boldsymbol{S}} \boldsymbol{b}_2 = \lambda \boldsymbol{b}_2$$

Finding the M^{th} Basis Vector $\boldsymbol{b}_M \in \mathbb{R}^D$

Subtract the effect of the first $M-1$ principal components $\boldsymbol{b}_1, \dots, \boldsymbol{b}_{M-1}$ from the data:

$$\widehat{\boldsymbol{x}}_n = \boldsymbol{x}_n - \left(\sum_{m=1}^{M-1} \boldsymbol{b}_m \boldsymbol{b}_m^T \right) \boldsymbol{x}_n$$

Then we can use the same argument:

$$V_m = \boldsymbol{b}_m^T \left(\frac{1}{N} \sum_{n=1}^{N} \widehat{\boldsymbol{x}}_n \widehat{\boldsymbol{x}}_n^T \right) \boldsymbol{b}_m = \boldsymbol{b}_m^T \widehat{\boldsymbol{S}} \boldsymbol{b}_m$$

The maximum variance is achieved at

$$\widehat{\boldsymbol{S}} \boldsymbol{b}_m = \lambda \boldsymbol{b}_m$$

Recall: Properties of Symmetric Matrix

Theorem (Spectral Theorem): If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric, there exists an **orthonormal basis** of vector space V from the eigenvectors of $\boldsymbol{A}$ and each eigenvalue is real.

Theorem: The eigenvectors $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$ of matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ with distinct eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ are linearly independent

Theorem (SPSD): For a given matrix $\boldsymbol{A} \in \mathbb{R}^{n \times n}$, we can always obtain a symmetric positive semi-definite matrix $\boldsymbol{S} \in \mathbb{R}^{n \times n}$: $\boldsymbol{S} = \boldsymbol{A}^T \boldsymbol{A}$

If $rank(\boldsymbol{A}) = n$, then $\boldsymbol{S}$ is a symmetric positive definite (SPD) matrix

Eigenvalues/Eigenvectors

$\mathcal{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}, \boldsymbol{x}_n \in \mathbb{R}^D$:

$$\boldsymbol{S} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{x}_n \boldsymbol{x}_n^T = \frac{1}{N} \boldsymbol{X} \boldsymbol{X}^T$$

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \cdots & \boldsymbol{x}_N \end{bmatrix} \in \mathbb{R}^{D \times N}$$

Typical Procedure to Obtain Eigenvalues/Eigenvectors

Perform Eigendecomposition on $\boldsymbol{S} = \boldsymbol{BDB}^{-1}$

- $\boldsymbol{D}$ are eigenvalues
- $\boldsymbol{B}$ are eigenvectors

Perform SVD on $\boldsymbol{X}$

$$\boldsymbol{X} = \boldsymbol{U\Sigma V}^T$$

$$\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}\boldsymbol{X}^T = \frac{1}{N}\boldsymbol{U\Sigma V}^T\boldsymbol{V\Sigma}^T\boldsymbol{U}^T = \frac{1}{N}\boldsymbol{U\Sigma\Sigma}^T\boldsymbol{U}^T$$

- $\boldsymbol{\Sigma}$ are eigenvalues
- $\boldsymbol{U}$ are eigenvectors

Low-Rank Approximations of $\boldsymbol{X}$

Consider the SVD of $\boldsymbol{X} = \boldsymbol{U\Sigma V}^T$

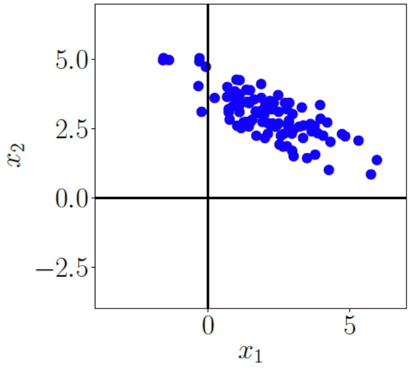
A low-rank approximation of $\boldsymbol{X}$ using the M largest eigenvalues:

$$\boldsymbol{X} = \boldsymbol{U}_M \boldsymbol{\Sigma}_M \boldsymbol{V}_M^T \in \mathbb{R}^{D \times N}$$

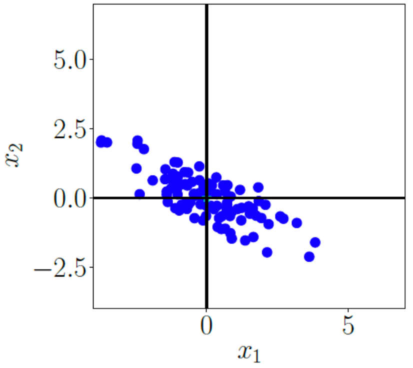
PCA Algorithm

1. Mean Subtraction : $\boldsymbol{x}_n = \boldsymbol{x}_n - \boldsymbol{\mu}$ where $\boldsymbol{\mu} = [u_1 \quad \cdots \quad u_d]^T$, d is the data dimension (eg $d = 3$ for RGB image)
2. Standardization by dividing the data by standard deviation: $\boldsymbol{x}_n = \frac{\boldsymbol{x}_n - \boldsymbol{\mu}}{\boldsymbol{\sigma}}$, where $\boldsymbol{\sigma} = [\sigma_1 \quad \cdots \quad \sigma_d]^T$
3. Eigendecomposition of covariance matrix $\boldsymbol{S} = \boldsymbol{BDB}^{-1}$
4. Projection: $\widetilde{\boldsymbol{x}}_n = \boldsymbol{B}_M \boldsymbol{B}_M^T \boldsymbol{x}_n$ where the coordinates with respect to the M principal basis vectors subspace: $\tilde{\boldsymbol{z}}_n = \boldsymbol{B}_M^T \boldsymbol{x}_n$
5. Backprojection: $\widetilde{\boldsymbol{x}}_n = \widetilde{\boldsymbol{x}}_n \boldsymbol{\sigma} + \boldsymbol{\mu}$

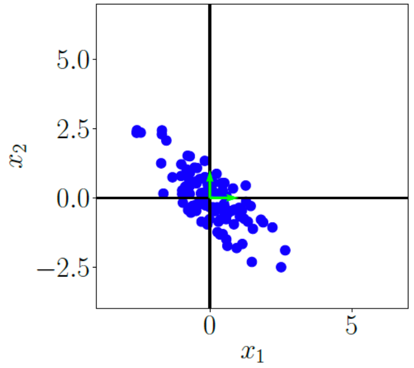
PCA Algorithm



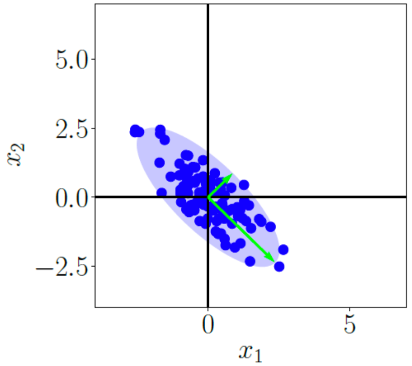
(a) Original dataset.



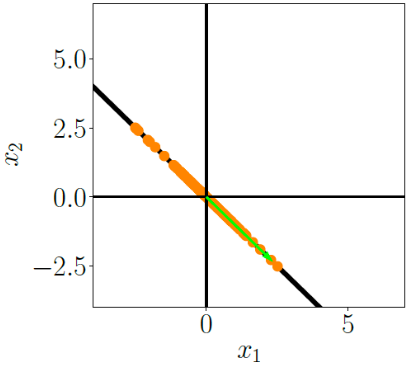
(b) Step 1: Centering by subtracting the mean from each data point.



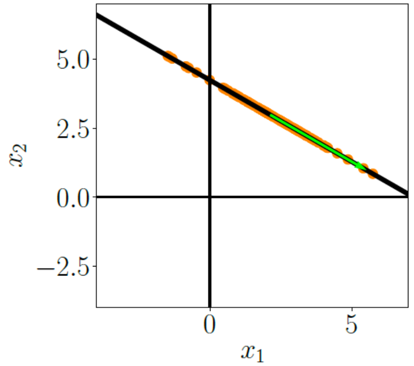
(c) Step 2: Dividing by the standard deviation to make the data unit free. Data has variance 1 along each axis.



(d) Step 3: Compute eigenvalues and eigenvectors (arrows) of the data covariance matrix (ellipse).



(e) Step 4: Project data onto the principal subspace.



(f) Undo the standardization and move projected data back into the original data space from (a).

Probabilistic Modelling

Probabilistic Model

Typical probabilistic model:

$$\underbrace{p(\boldsymbol{x}, \boldsymbol{\theta})}_{\text{Joint probability}} = \underbrace{p(\boldsymbol{x}|\boldsymbol{\theta})}_{\text{likelihood}}\underbrace{p(\boldsymbol{\theta})}_{\text{prior}}$$

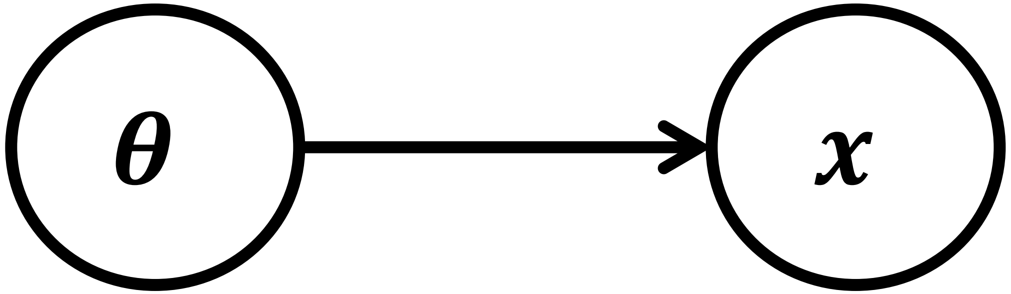
Marginal: $p(\boldsymbol{x}) = \int p(\boldsymbol{x}, \boldsymbol{\theta})\, d\boldsymbol{\theta}$

Posterior: $p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x},\boldsymbol{\theta})}{p(\boldsymbol{x})}$

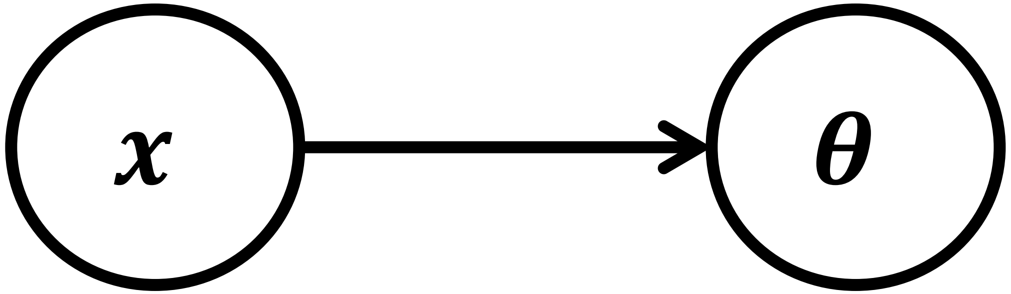
Probabilistic Model DGM

$$p(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta}) = p(\boldsymbol{\theta}|\boldsymbol{x})p(\boldsymbol{x})$$

$p(\boldsymbol{x}|\boldsymbol{\theta})$



$p(\boldsymbol{\theta}|\boldsymbol{x})$



Latent Variable Model

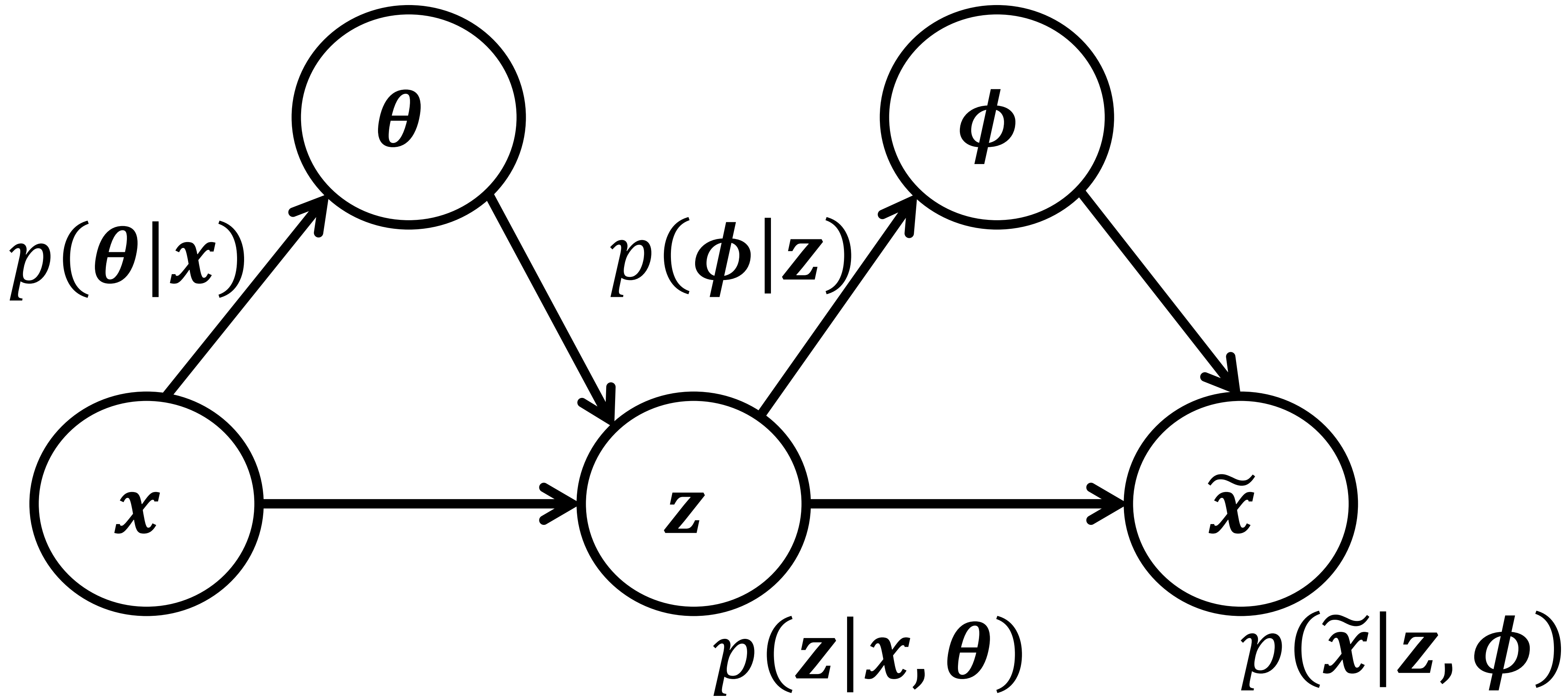
An intermediate latent variable $\boldsymbol{z}$ is introduced as part of the model

The latent variable $\boldsymbol{z}$ is not a model parameter

The latent variable $\boldsymbol{z}$ describes both the data distribution $p(\boldsymbol{x})$, thus the data generating process $p(\widetilde{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{\phi})$ where $\boldsymbol{\phi}$ represents the model parameters

Latent Variable Model DGM

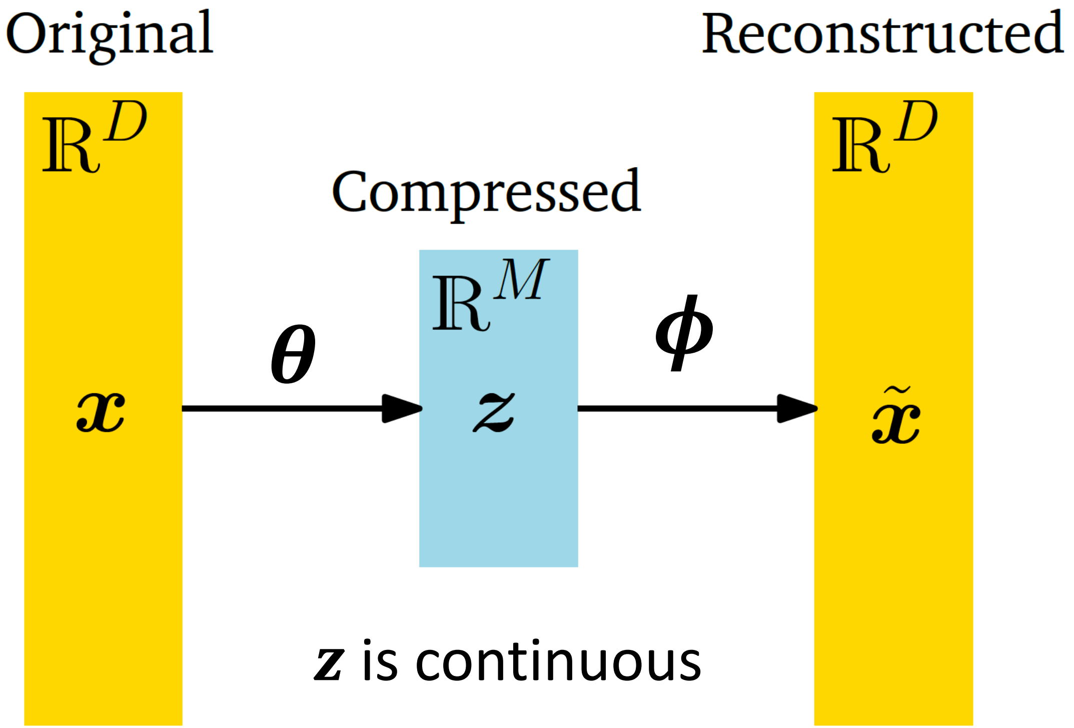
$$p(\boldsymbol{x}, \widetilde{\boldsymbol{x}}, \boldsymbol{\theta}, \boldsymbol{\phi}) = p(\boldsymbol{x})p(\boldsymbol{\theta}|\boldsymbol{x})p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{\phi}|\boldsymbol{z})p(\widetilde{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{\phi})$$



Latent Variable Model of PCA

Encoder: $p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})$

Decoder: $p(\tilde{\boldsymbol{x}}|\boldsymbol{z}, \boldsymbol{\phi})$



$\boldsymbol{z}$ is continuous

Probabilistic PCA (PPCA)

Can deal with noise

Can use Bayesian interpretation

Can use PCA decoder as a generator

Can generate new data points from the generator

Can extend to mixture of PCA

Can treat PCA as a special case

etc

PPCA

If $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and linear relationship between latent variable and observed data $\boldsymbol{x}$,

$$\boldsymbol{x} = \boldsymbol{B}\boldsymbol{z} + \boldsymbol{\mu} + \boldsymbol{\epsilon} \in \mathbb{R}^D$$

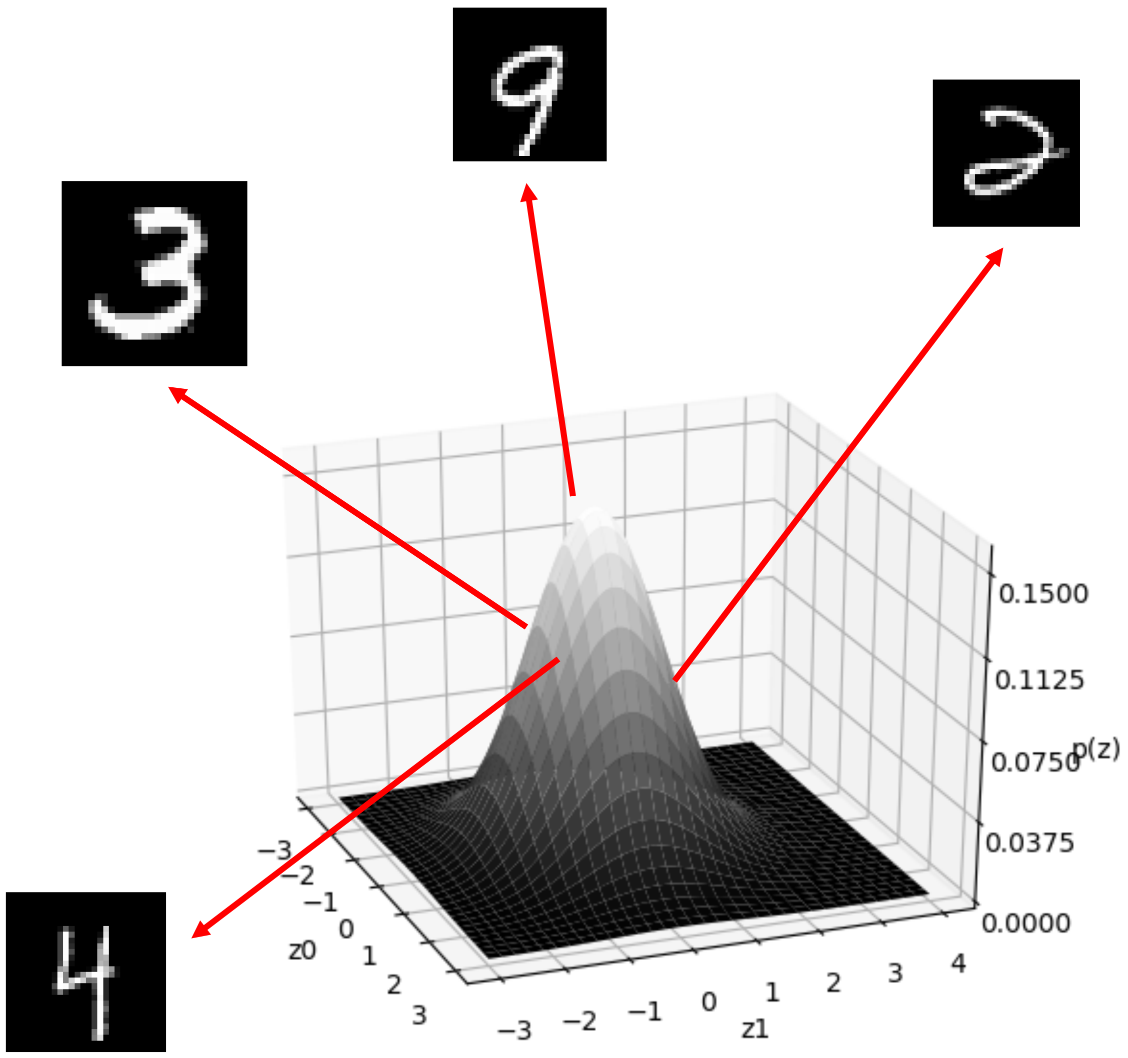
$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$, $\boldsymbol{B} \in \mathbb{R}^{D \times M}$, $\boldsymbol{\mu} \in \mathbb{R}^D$:

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

Generative Model

$$\boldsymbol{z}_n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$$

$$\boldsymbol{x} | \boldsymbol{z}_n \sim \mathcal{N}(\boldsymbol{x} | \boldsymbol{B}\boldsymbol{z}_n + \boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$$



Generative Model

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

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

$$= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu},\boldsymbol{B}\boldsymbol{B}^T+\sigma^2\boldsymbol{I})$$