Dimensionality Reduction

Machine Learning

Nguyễn Bảo Long, MSc

August 3, 2025

Content

Section 1

Introduction

Curse of Dimensionality

- Increasing dimension leads to increasing data space
  $\rightarrow$ Distance measures perform poorly

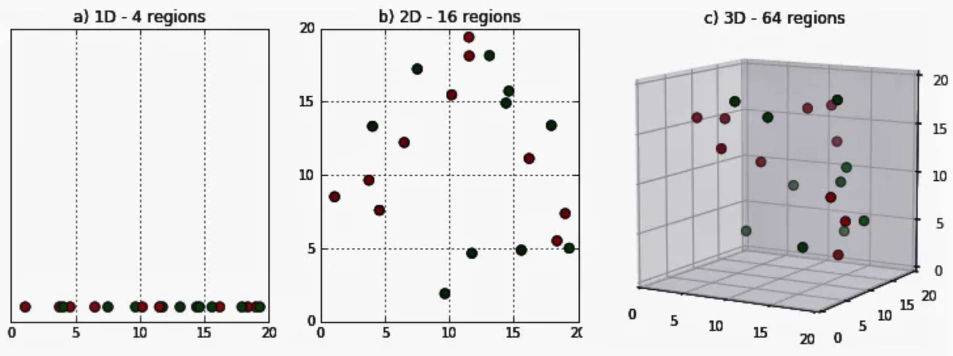

Figure 1: Increase dimension $\rightarrow$ Sparse data $\rightarrow$ Poor measurement

Curse of Dimensionality

- Increasing dimension leads to increasing data space
  $\rightarrow$ Predictive models become less effective at exploring patterns

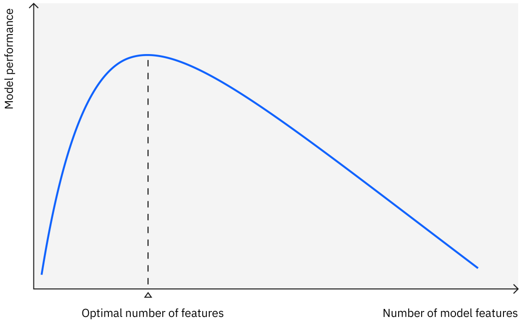

Figure 2: Model performance vs. #feature

Dimensionality reduction

- **Transform** n-dim data to k-dim data ($k < n$) while **preserving** as much information as possible
- In the example, we reduce datapoint from 2-D to 1-D (on x-axis)

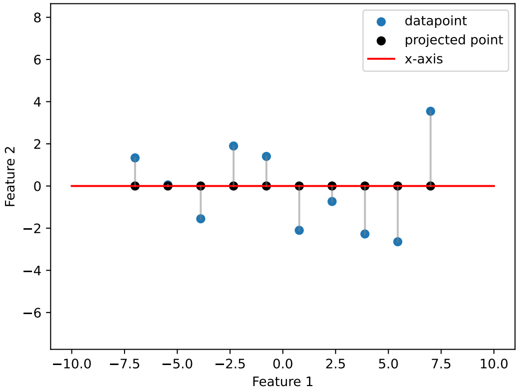

Figure 3: Project data onto x-axis

Dimensionality reduction

- What if x-axis and y-axis are not enough to preserve information?
  $\rightarrow$ Project data onto a new axis
- Preserving information $\equiv$ **Maximizing standard deviation**
  $\rightarrow$ Idea of PCA

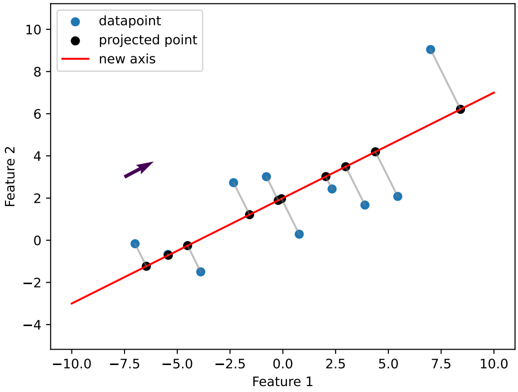

Figure 4: Project data onto a new axis

Section 2

Principal Component Analysis (PCA)

Problem statement

- Given $X = (\vec{X}_1, \vec{X}_2, \dots, \vec{X}_n) \in \mathbb{R}^{m \times n}$: Data matrix
- Find an $f(X)$, that maps every $\vec{x} \in \mathbb{R}^n$ to $\mathbb{R}$

$$\vec{P} = f(X) = X\vec{w} \quad (\vec{w} \in \mathbb{R}^n)$$

$$= \begin{bmatrix} — & \vec{x_1}^T & — \\ — & \vec{x_2}^T & — \\ & \vdots & \\ — & \vec{x_m}^T & — \end{bmatrix} \vec{w} = \begin{bmatrix} \vec{x_1}^T \vec{w} \\ \vec{x_2}^T \vec{w} \\ \vdots \\ \vec{x_m}^T \vec{w} \end{bmatrix}$$

- Such that σ_P^2 is maximum. $\vec{P}$ is called a **principal component**

Table 1: Dataset X

	$\vec{X}_1$	$\vec{X}_2$	$\dots$	$\vec{X}_n$
$\vec{x_1}^T$	x_{11}	x_{12}	$\dots$	x_{1n}
$\vec{x_2}^T$	x_{21}	x_{22}	$\dots$	x_{2n}
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\vec{x_m}^T$	x_{m1}	x_{m2}	$\dots$	x_{mn}

Problem statement

- Given $X = (\vec{X}_1, \vec{X}_2, \dots, \vec{X}_n) \in \mathbb{R}^{m \times n}$: Data matrix
- Find an $f(X)$, that maps every $\vec{x} \in \mathbb{R}^n$ to $\mathbb{R}$

$$\vec{P} = f(X) = X\vec{w} \quad (\vec{w} \in \mathbb{R}^n)$$
$$= \begin{bmatrix} \text{—} \; \vec{x_1}^T \; \text{—} \\ \text{—} \; \vec{x_2}^T \; \text{—} \\ \vdots \\ \text{—} \; \vec{x_m}^T \; \text{—} \end{bmatrix} \vec{w} = \begin{bmatrix} \vec{x_1}^T \vec{w} \\ \vec{x_2}^T \vec{w} \\ \vdots \\ \vec{x_m}^T \vec{w} \end{bmatrix}$$

- Such that σ_P^2 is maximum. $\vec{P}$ is called a **principal component**

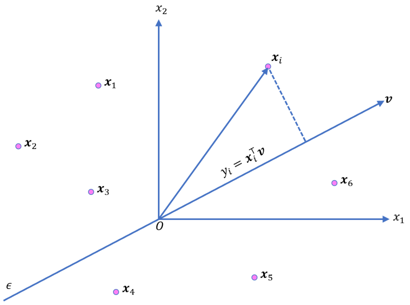

Figure 5: Project a vector onto a direction

Problem statement

- Given $X = (\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_n) \in \mathbb{R}^{m \times n}$: Data matrix
- $\vec{w}$ is a unit vector

PCA problem

$$\max_{\vec{w}} \left\{ \sigma_P^2 \right\}$$

$$s.t. \; \|\vec{w}\|_2 = 1$$

Find $\vec{w}$ such that $\vec{P} = X\vec{w}$, $\|\vec{w}\|_2 = 1$ and σ_P^2 is maximum

- Consider σ_P^2:

$$
\begin{aligned}
\sigma_P^2 &= \mathbb{E}\left[\left(\vec{x_i}^T\vec{w} - \mu_P\right)^2\right] \qquad (\mu_P = \mathbb{E}[\vec{x_i}^T\vec{w}] = \vec{\mu_X}^T\vec{w}) \\
&= \mathbb{E}\left[(X\vec{w} - \bar{X}\vec{w})^T(X\vec{w} - \bar{X}\vec{w})\right] \\
&= \mathbb{E}\left[\vec{w}^T(X - \bar{X})^T(X - \bar{X})\vec{w}\right] \\
&= \vec{w}^T\Sigma[X]\vec{w}
\end{aligned}
$$

$(\Sigma[X] = \frac{1}{m-1}(X - \bar{X})^T(X - \bar{X})$: **Unbiased covariance matrix**)

Find $\vec{w}$ such that $\vec{P} = X\vec{w}$, $\|\vec{w}\|_2 = 1$ and σ_P^2 is maximum

- Combine with constrain $\|w\|_2 = 1$ via **Lagrange multiplier**:

$$\begin{aligned} &\max_{\vec{w}} \left\{ \vec{w}^T \Sigma[X] \vec{w} - \lambda(\vec{w}^T\vec{w} - 1) \right\} \\ &= \max_{\vec{w}} J \end{aligned}$$

- Solve the above problem by considering $\frac{dJ}{d\vec{w}} = 0$

$$\begin{aligned} \frac{dJ}{d\vec{w}} &= \frac{d}{d\vec{w}} \left\{ \vec{w}^T \Sigma[X] \vec{w} - \lambda(\vec{w}^T\vec{w} - 1) \right\} \\ &= I\Sigma[X]\vec{w} + \Sigma[X]^T\vec{w} - 2\lambda\vec{w} \\ &= 2\Sigma[X]\vec{w} - 2\lambda\vec{w} = 0 \quad (\text{since } \Sigma[X] = \Sigma[X]^T) \end{aligned}$$

$\Leftrightarrow \Sigma[X]\vec{w} = \lambda\vec{w}$ this is Eigenvalue problem for Covariance matrix

in which, λ is variance of projected data (on direction $\vec{w}$)

Find $\vec{w}$ such that $\vec{P} = X\vec{w}$, $\|\vec{w}\|_2 = 1$ and σ_P^2 is maximum

- By solving $\Sigma[X]\vec{w} = \lambda\vec{w}$ with constrain $\|\vec{w}\|_2 = 1$, we obtain n pairs (n axes) $(\lambda, \vec{w})$ (since $\Sigma[X] \in \mathbb{R}^{n\times n}$)
- The eigenvector corresponding to the largest eigenvalue is the new axis that preserves the most information, and so on...
- Project data onto $\vec{w}_1$, we obtain the **first principal component**

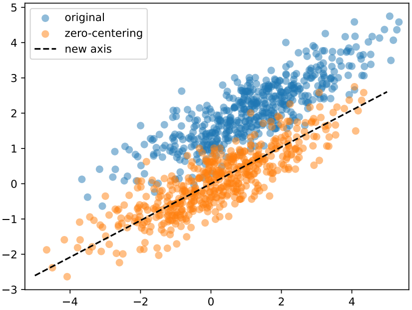

Figure 6: Which data to project, X or $(X - \bar{X})$?

PCA - Step by step

1. Compute unbiased covariance matrix of zero-centering data: $\tilde{X} = X - \bar{X}$

$$\Sigma[X] = \frac{1}{m-1}\tilde{X}^T\tilde{X}$$

2. Solve the eigenvalue problem to obtain n pairs $(\lambda, \vec{w})$, which are n directions of our data

$$\Sigma[X]\vec{w} = \lambda\vec{w}$$

3. Sort eigenvalues in descending order and select k eigenvectors corresponding to k largest eigenvalues to form W. Our new dataset is $P = \tilde{X}W \in \mathbb{R}^{m\times k}$

$$P = \tilde{X}W = \tilde{X}\begin{bmatrix} | & | & & | \\ \vec{w}_1 & \vec{w}_2 & \dots & \vec{w}_k \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \tilde{X}\vec{w}_1 & \tilde{X}\vec{w}_2 & \dots & \tilde{X}\vec{w}_k \\ | & | & & | \end{bmatrix}$$

PCA on a tabular dataset

- Problem: Given a 2D dataset (denote $X \in \mathbb{R}^{3\times 2}$). Obtain the first and second component of X

Table 2: Dataset X

	Feature 1	Feature 2
$\vec{x_1}^T$	2	1
$\vec{x_2}^T$	1	2
$\vec{x_3}^T$	0	0

PCA on a tabular dataset

- Problem: Given a 2D dataset (denote $X \in \mathbb{R}^{3\times 2}$). Obtain the first and second component of X
- Step 1: Compute **unbiased covariance matrix**

$$\tilde{X} = X - \bar{X} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{bmatrix}$$

$$\Sigma[X] = \frac{1}{2}\tilde{X}^T\tilde{X}$$

$$= \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

Table 3: Dataset X with means

	Feature 1	Feature 2
$\vec{x_1}^T$	2	1
$\vec{x_2}^T$	1	2
$\vec{x_3}^T$	0	0
	$\mu_1 = 1$	$\mu_2 = 1$

PCA on a tabular dataset

- Problem: Given a 2D dataset (denote $X \in \mathbb{R}^{3\times 2}$). Obtain the first and second component of X
- Step 2: Compute **eigenvalues, eigenvectors**

$$\Sigma[X]\vec{w} = \lambda\vec{w}$$

$$\Leftrightarrow \quad (\Sigma[X] - \lambda I)\vec{w} = 0$$

- Since $\vec{w} \neq 0 \Rightarrow \det(\Sigma[X] - \lambda I) = 0$

$$\det\left(\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) = 0$$

$$\Leftrightarrow \quad \lambda_1 = 1.5 \text{ or } \lambda_2 = 0.5$$

- Substitute λ into $\Sigma[X]\vec{w} = \lambda\vec{w}$ to obtain **eigenvectors**

$$\lambda_1 = 1.5 \rightarrow \vec{w_1} = [t, t]^T \quad (t \in \mathbb{R})$$

$$\lambda_2 = 0.5 \rightarrow \vec{w_2} = [t, -t]^T \quad (t \in \mathbb{R})$$

- Since $\|w\|_2 = 1$, we obtain the normalized $\vec{w}$

$$\vec{w_1} = \left[\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right]^T$$

$$\vec{w_2} = \left[\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right]^T$$

PCA on a tabular dataset

- Problem: Given a 2D dataset (denote $X \in \mathbb{R}^{3\times 2}$). Obtain the first and second component of X
- Step 3: Form W and compute new dataset

$$P = \tilde{X}W = \begin{bmatrix} | & | \\ \tilde{X}\vec{w_1} & \tilde{X}\vec{w_2} \\ | & | \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ -\sqrt{2} & 0 \end{bmatrix}$$

Table 4: New dataset

	1^{st} comp.	2^{nd} comp.
$\vec{x_1}^T$	$\frac{1}{\sqrt{2}}$	$\frac{1}{\sqrt{2}}$
$\vec{x_2}^T$	$\frac{1}{\sqrt{2}}$	$-\frac{1}{\sqrt{2}}$
$\vec{x_3}^T$	$-\sqrt{2}$	0

How much information does 1^{st} comp. preserve?

$$R_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2} = 75\%$$

PCA on image

- Problem: Given a 2D image, size (512×512). Compress the image and compare to the original one
- Solution:
  - Divide the image into patches, size $(16 \times 16) \rightarrow 1024$ patches
  - In each patch, every pixel is a feature $\rightarrow 256$ features
  - Obtain a tabular data of size $(1024 \times 256) \rightarrow$ Perform PCA

Figure 7: Lena

PCA on image

- Reconstruct image from P : $\hat{X} = PW^T = \tilde{X}WW^T$
- The quality of reconstructive images are lower than the original one

Figure 8: Original image vs. Reconstructive images

PCA on image

- Measure the loss between original one and reconstructive ones by the distance between them

$$L(X, \hat{X}) = d(X, \hat{X})$$

- In this case, we choose $l_2 - norm$

$$L(X, \hat{X}) = \|X - \hat{X}\|_2$$

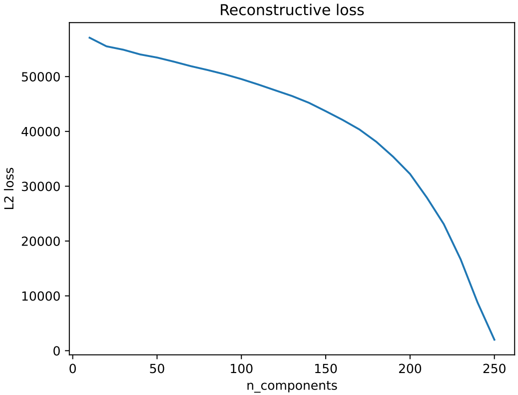

Figure 9: Reconstructive loss of PCA

Section 3

Conclusion

Conclusion

Dimensionality reduction

- Curse of Dimensionality: High dimension $\rightarrow$ Poor distance measurement $+$ Model efficiency
- Reduce dimension by projecting onto new space

PCA

- Find directions, in which projected data have large variance
- Optimize σ^2 via Lagrange multiplier

Other approaches

- Kernel PCA: Transform data to another space via a kernel function (which introduces more corelation between variables), then perform PCA.
- Multidimensional Scaling: Focus on preserving distance between datapoint instead of std.
- Non-Negative Matrix Factorization: Similar to PCA but the return values are non-negative. Use for non-negative data (movie rating, human-related features, frequency, intensity)

References

Christopher M. Bishop.
Pattern recognition and machine learning.

Tiep Vu Huu.
Machine learning cơ bản.

Marc Peter Deisenroth; A. Aldo Faisal; Cheng Soon Ong.
Mathematic for machine learning.