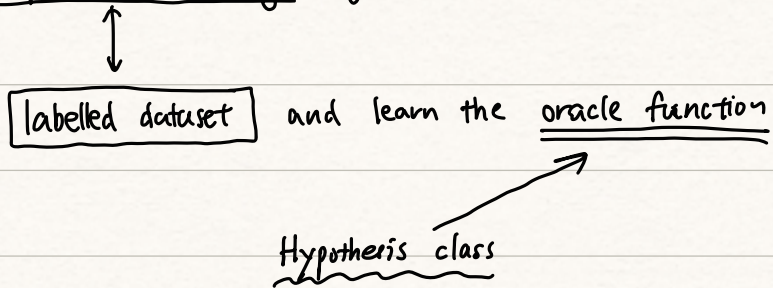


Recap:

up to now, there are supervised learning algos.



Unsupervised Learning

★ understand the data (Aim)

① find some structure of data

② dimension reduction → reduce cost

will lose part of information

(depends on the situation)

→ cluster users

③ clustering

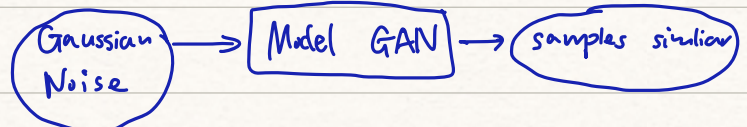
④ density estimation (non-parametric stats)

$$\underline{p(x) dx = \mathbb{P}(X \in dx)}$$

Application: generate new samples

⑤ Generative Model (GANs)

↓
Stable diffusion model (SOTA)



why?

- ① labelled data expensive
- ② labelled data impossible to attain
- ③ different application scenarios

PCA

① Recall

a) $\underline{Au = \lambda u} \rightarrow \begin{cases} \text{eigenvalue: } \lambda \\ \text{eigenvector: } u \end{cases}$

b) Diagonalization of $A \Leftrightarrow A = P \Lambda P^{-1}$ P is invertible
 $\Leftrightarrow AP = P \Lambda \Leftrightarrow \boxed{A p_i = \lambda_i p_i}$
 \Rightarrow column of P is eigenvector

c) symmetric $A \Rightarrow \boxed{A = U \Lambda U^T}$
 where $\begin{cases} U U^T = I \text{ (} U \text{ is orthonormal)} \\ \Lambda \text{ diagonal} \end{cases}$

d) PD $A \Leftrightarrow x^T A x > 0$ for all $x \in \mathbb{R}^n$ and $x \neq 0$

PSD $A \Leftrightarrow x^T A x \geq 0$ for all $x \in \mathbb{R}^n$

e) symmetric A PD $\Leftrightarrow A = U \Lambda U^T$
 and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n > 0$
 \downarrow
eigenvalue

② PCA $\begin{cases} \text{min error} \\ \text{max variance} \end{cases}$ 2 interpretation (formulation)

a) max var derivation

$\mathcal{D} = \{ (x_i) \}_{i=1}^N$

Goal: Find a direction $u \in \mathbb{R}^d$ s.t. "Variance" of the projection is maximized

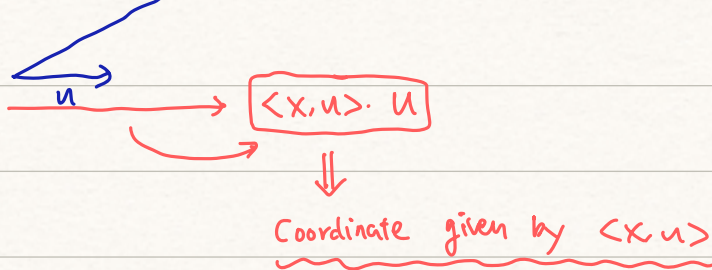
Assume $\|u\| = 1$.

1. How to compute projection

某一
 沿着 某个方向 的 projection

$A \cdot x$

$\nearrow x$



$$2. \text{ Variance} = \frac{1}{N} \sum_{i=1}^N (a_i - \bar{a})^2$$

Then we can compute:

(Assume $\bar{x} = \frac{1}{N} \sum x_i = 0$) \rightarrow we can normalize $\tilde{x}_i = x_i - \bar{x}$

then let $z_i = x_i^T u \Rightarrow z = Xu \quad X = \begin{pmatrix} x_1^T \\ \vdots \\ x_N^T \end{pmatrix}$

$$\Rightarrow \bar{z} = \frac{1}{N} \sum_{i=1}^N z_i = \frac{1}{N} \sum_{i=1}^N x_i^T u = 0$$

$$\Rightarrow \text{Variance} = \frac{1}{N} \sum_{i=1}^N z_i^2$$

$$= \frac{1}{N} z^T z$$

$$= \frac{1}{N} u^T X^T X u$$

symmetric, PSD

$$= u^T \left(\frac{1}{N} X^T X \right) u$$

$$\frac{1}{N} \sum_{i=1}^N x_i x_i^T$$

$$\text{Variance} := \underline{u^T S u} \quad \underline{S \text{ is symmetric \& PSD}}$$

\rightarrow max variance formulation

$$\Rightarrow \begin{cases} \max_{u \in \mathbb{R}^d} u^T S u \\ \text{s.t. } \|u\| = 1 \end{cases}$$

convex prog ($S \succeq 0$)

$$\rightarrow L(u; \lambda) = u^T S u + \lambda (1 - \|u\|_2^2)$$

$$\nabla L(u; \lambda) = 2S u - 2\lambda u$$

$$\nabla_u L(\hat{u}; \hat{\lambda}) = 0$$

$$\Leftrightarrow S \hat{u} = \hat{\lambda} \hat{u}$$

Strong duality

Stationarity

\hat{u} is the eigenvector of some
eigenvalue $\hat{\lambda}$

$$\max_{u \in \mathbb{R}^d} \lambda u^T u$$

$$\text{s.t. } \|u\| = 1$$

u is eigenvector of S for some λ

$(\hat{\lambda}, \hat{u})$ should be the largest eigenvalue ($\hat{\lambda}$)

and corresponding eigenvector (\hat{u})

$\Rightarrow \begin{cases} \hat{u} : \text{the first principal component} \end{cases}$

$$Z_i = x_i^T \hat{u} : \text{the scores}$$

x_i 's score for one direction

Generalization to several directions: (u_1, u_2)

Modification

$$\begin{cases} \max_{u \in \mathbb{R}^d} u^T S u \\ \text{s.t. } \|u\| = 1 \end{cases}$$

$$u^T u_1 = 0$$

$$\longrightarrow \mathcal{L}(u; \lambda, \beta) = u^T S u + \lambda (1 - u^T u) + \beta u^T u_1$$

strong duality

$$\Rightarrow (\hat{u}; \hat{\lambda}, \hat{\beta}) \text{ s.t. } \begin{cases} \nabla_u \mathcal{L}(\hat{u}; \hat{\lambda}, \hat{\beta}) = 0 \\ \hat{u}^T u_1 = 0 \end{cases}$$

$$\Rightarrow 2u_1^T S \hat{u} + \hat{\beta} = 0$$

$$\Rightarrow 2\lambda_1 \underline{u_1^T \hat{u}} + \hat{\beta} = 0$$

$$\Rightarrow \hat{\beta} = 0 \xrightarrow{\text{set } \lambda} \nabla_u \mathcal{L}$$

$$\Rightarrow S \hat{u} = \hat{\lambda} \cdot \hat{u} \Rightarrow \underline{\text{eigenvalue \& eigenvector}}$$

$$\begin{aligned} & \max_{\lambda} \lambda \\ & \text{s.t. } \|u\|=1 \quad \Leftrightarrow \quad \text{s.t. } \lambda \text{ is eigenvalue} \\ & \quad \quad \quad \lambda \neq \lambda_1 \\ & \quad \quad \quad u^T u_1 = 0 \\ & (u, \lambda) \text{ is the (eigenvector, eigenvalue)} \end{aligned}$$

\Downarrow
 $(\hat{\lambda}, \hat{u})$ should be the second largest eigenvalue
 and corresponding eigenvector

✱ explained variance \rightarrow ① $U_m = (u_1, \dots, u_m)$
 $Z_m = X U_m$ is the score matrix
 ② $\sum_{i=1}^m \lambda_i$ is the explained variance

b) min error (second interpretation)
 (projection error)

defn \downarrow
 $\|x - (x^T u) \cdot u\|_2^2$

1) if $x = \alpha u$, then projection error = 0

the overall projection error: $= \frac{1}{N} \sum_{i=1}^N \|x_i - (x_i^T u) \cdot u\|_2^2$

our aim: $\min_u \frac{1}{N} \sum_{i=1}^N \|x_i - (x_i^T u) \cdot u\|_2^2$ $\|X - X \cdot u \cdot u^T\|_F^2$
 s.t. $\|u\|=1$ $X u = (x_1^T u, \dots, x_N^T u)$

calculation: $\sum_{i=1}^N \|x_i - (x_i^T u) u\|_2^2$

$$= \sum_{i=1}^N \|x_i\|_2^2 - 2 (x_i^T u)^2 + \|(x_i^T u) \cdot u\|_2^2$$

$$= \sum_{i=1}^N \|x_i\|_2^2 - (x_i^T u)^2$$

$$\min \sum_{i=1}^N (\|x_i\|_2^2 - (x_i^T u)^2)$$

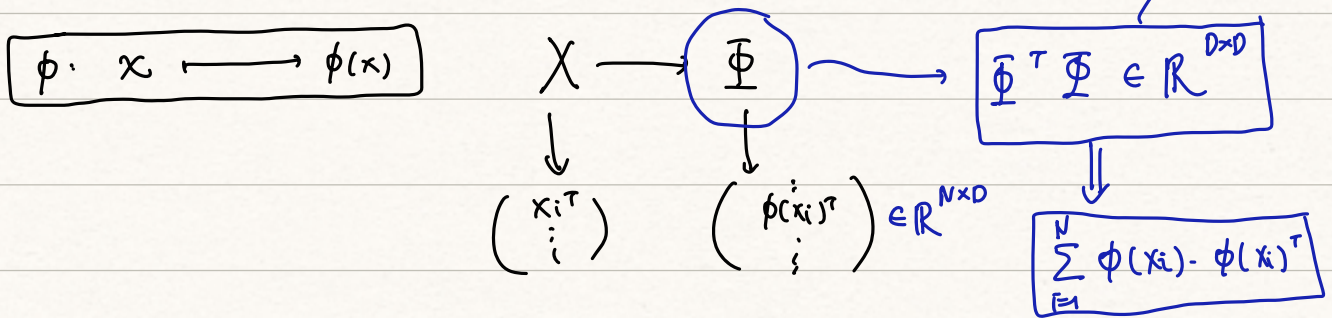
$$\Leftrightarrow \max \sum_{i=1}^N (x_i^T u)^2$$

$$\Leftrightarrow \max u^T X^T X u \rightarrow \text{max variance interpretation !!!}$$

Application \rightarrow don't use diagonalization to compute $\begin{cases} \text{Eigenvalue} \\ \text{Eigenvectors} \end{cases}$ for $X^T X$

SVD instead

Kernel PCA \rightarrow focus on Non-linear data (circle ...)



Recall Principal component scores $Z = X \cdot U \in \mathbb{R}^{N \times m}$ $U \in \mathbb{R}^{d \times m}$

Defn transformation $X' = X U \Lambda^{-\frac{1}{2}} = \left(\frac{1}{\sqrt{\lambda_j}} x_i^T u_j \right)$ $X^T X = U \Lambda U^{-1}$

whitening \rightarrow 去相关性

normalize the score

对每个 sample 的 feature 做线性组合并消除相关性

give them the same importance

$\begin{cases} E[X'] = 0 \\ \text{Cov}(X') = I \end{cases}$

PCA \rightarrow Compression Algorithm

$$Z_m = X \cdot U_m \rightarrow \text{project on } \underline{\text{first-}m\text{-component}}$$

Recover $\rightarrow X^* = Z_m U_m^T$ Note $U_m \cdot U_m^T = \begin{pmatrix} I & \\ & 0 \end{pmatrix}$

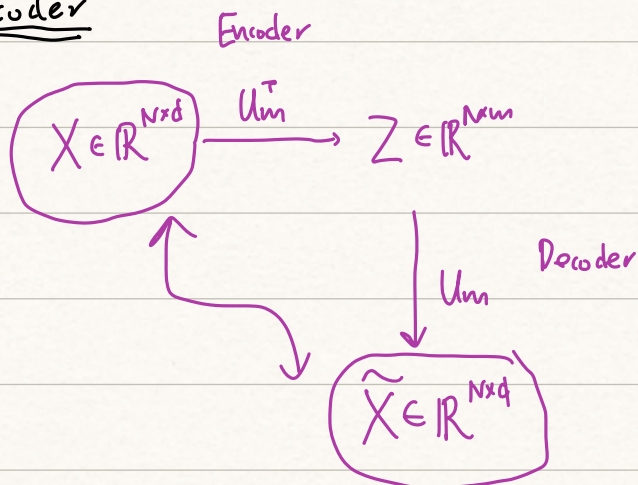
Generalization
(AE) \rightarrow

Auto-Encoder

$$\begin{cases} \underline{\text{Encoder}} : T_{\text{enc}}(X; \theta) = Z_m \\ \underline{\text{Decoder}} : T_{\text{dec}}(Z_m; \theta) = X^* \end{cases}$$

Generalization

RNN Encoder-Decoder Architecture



What have done in this class?

① $\mathcal{D} = \{(x_i)\}_{i=1}^N \Rightarrow X = (x_i^T) \in \mathbb{R}^{N \times d}$

$$\bar{X} := \begin{pmatrix} \bar{x}^T \\ \vdots \\ \bar{x}^T \end{pmatrix}$$

projection in direction $u \in \mathbb{R}^d : \begin{cases} Z = Xu \in \mathbb{R}^N \\ \bar{Z} = \bar{X}u \in \mathbb{R}^N \end{cases}$

$$\begin{aligned} \text{then } \text{var}(Z) &= \frac{1}{N} (Z - \bar{Z})^T (Z - \bar{Z}) \\ &= \frac{1}{N} u^T (X - \bar{X})^T (X - \bar{X}) u \end{aligned}$$

$$= \frac{1}{N} u^T \tilde{X}^T \tilde{X} u \rightarrow \underline{\text{this means:}}$$

we can normalize the data first,

and then using the optimal

$$\begin{aligned} \max_u \quad & \text{var}(Z) \\ \text{s.t.} \quad & \|u\|_2^2 = 1 \end{aligned}$$

$$\Leftrightarrow \hat{u} = \text{largest eigenvector}$$

projection direction

for the original data

$$\text{score } \hat{z} = \tilde{X} \hat{u} \quad \rightarrow \text{under centralization}$$

② Generally speaking, given $X \xrightarrow{\text{centralization}} \tilde{X}$

$$\textcircled{2} \quad S = \tilde{X}^T \tilde{X} \rightarrow \text{covariance mat}$$

③ achieve first m eigenvalue & vector

$$(\lambda_1, u_1), \dots, (\lambda_m, u_m)$$

④

$$U_m = (u_1, \dots, u_m) \rightarrow \text{orthogonal basis}$$

$$Z = X U_m \in \mathbb{R}^{N \times m}$$

score matrix

$$\text{explained variance} = \sum_{i=1}^m \lambda_i$$

$$Z = \begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix} \rightarrow \begin{array}{l} \text{the coordinates} \\ \text{for sample 1} \end{array}$$

Encoder

$$\begin{cases} x \xrightarrow{u_m^T} z = u_m^T x \\ X = \begin{pmatrix} x_1^T \\ \vdots \\ x_N^T \end{pmatrix} \rightarrow Z = X \cdot U_m \end{cases}$$

Decoder

$$\begin{cases} z \xrightarrow{u_m} \hat{x} = u_m z \\ Z = \begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix} \rightarrow X^* = Z \cdot U_m^T \end{cases}$$

Consider the 2nd formulation

$$\{u_1, \dots, u_d\} \rightarrow \text{orthogonal basis}$$

$$x = \sum_{i=1}^d (u_i^T x) \cdot u_i \quad \tilde{x} = \sum_{i=1}^m \beta_i u_i \approx x$$

formulation of problem

linear projection

$$\begin{aligned} \textcircled{1} x &\rightarrow Px \quad \text{坐标变换} \\ \textcircled{2} X &\rightarrow X \cdot P^T \quad \text{坐标降} \end{aligned}$$

optimization

$$X = (x_i^T)_{i \in I}$$

equivalent

$$\min_{\substack{u \\ \beta_i \\ \text{s.t. } \{u_1, \dots, u_d\} \text{ orthogonal basis}}} \frac{1}{N} \sum_{i=1}^N \|x_i - \tilde{x}_i\|_2^2$$

$$\text{s.t. } \{u_1, \dots, u_d\} \text{ orthogonal basis}$$

$$x_i - \tilde{x}_i = \sum_{j=1}^d [(u_j^T x_i) - \beta_{ij}] \cdot u_j + \sum_{j=m+1}^d u_j^T x_i u_j$$

$$\min_{\substack{u \\ \beta_i \\ \text{s.t. } U \text{ orthogonal}}} \frac{1}{N} \sum_{i=1}^N \left(\sum_{j=1}^m (\beta_{ij} - u_j^T x_i)^2 + \sum_{j=m+1}^d (u_j^T x_i)^2 \right)$$

↑

for $i=1, \dots, N, j=1, \dots, m$.

$$\beta_{ij} = u_j^T x_i$$

$$\min_u \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d (u_j^T x_i)^2$$

s.t. $\{u_1, \dots, u_d\}$ orthogonal

$$\beta_{ij} = u_j^T x_i \text{ (score)}$$

\Downarrow

one choice of $\hat{u}_1, \dots, \hat{u}_d$ is the eigenvector of

$$S = \frac{1}{N} X^T X \in \mathbb{R}^{d \times d}$$

$$Z_m = X \cdot u_m \rightarrow Z_m = \begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix}$$

$$u_m = (u_1, \dots, u_m)$$

Score (coordinates) given basis

$$\hat{u}_1, \dots, \hat{u}_m$$

$$x \xrightarrow{u_m^T} z \xrightarrow{u_m} \tilde{x}$$

then, $\frac{1}{N} \sum_{i=1}^N \|x_i - \tilde{x}_i\|_2^2$

$$= \sum_{j=1}^d \lambda_j$$

$$\tilde{X} = \begin{pmatrix} \tilde{x}_1^T \\ \vdots \\ \tilde{x}_N^T \end{pmatrix} = X u_m u_m^T = \begin{pmatrix} z_1^T \\ \vdots \\ z_N^T \end{pmatrix} \cdot u_m^T$$

un explained variance

= average compression loss