PCA Summary (Family)

① Probabilistic PCA (P-PCA) → Latent Variable Model (EM Application)

Recap → 2 Framework { EM Algorithm ; alt-opt }

Aim: $\hat{\theta}_{MLE} = \arg\max_{\theta} \log P(X|\theta) = \arg\max_{\theta} \log \sum_{z} P(z|\theta) P(X|z,\theta)$

$\log P(X|\theta)$: ILL: Incomplete Log-likelihood

→ ALT-OPT

1. initialize $\theta = \hat{\theta}$
2. estimate $\hat{z} = \arg\max_{z} \log P(z|X,\hat{\theta})$
3. estimate $\hat{\theta} = \arg\max_{\theta} \log P(X,\hat{z}|\theta)$

$P(X,\hat{z}|\theta)$: CLL: Complete Log-likelihood

slight modified

→ EM:

1. initialize $\theta = \hat{\theta}$
2. compute posterior distribution $P(z|X,\hat{\theta})$
3. estimate θ by Expected CLL

$$\hat{\theta} = \arg\max_{\theta} \mathbb{E}_{z\sim P(z|X,\hat{\theta})}[\log P(X,z|\theta)]$$

---

1. Probabilistic PCA → Generative Model

↓

Latent Variable Model for Dimensionality Reduction

Motivation consider $x \in \mathbb{R}^D$

imagine that x is generated by

scalar form ← $x \approx \sum_{k=1}^{K} z_k w_k$ $\quad w_k \in \mathbb{R}^D$ $\quad k = 1, 2, \ldots, K$

vector form ← $x \approx Wz$ → Notation $\begin{cases} W = (w_1, \ldots, w_K) \in \mathbb{R}^{D \times K} \\ z = \begin{pmatrix} z_1 \\ \vdots \\ z_K \end{pmatrix} \in \mathbb{R}^K \end{cases}$

Rmk Here, z is the low-dimension representation

Question: How to model "$\approx$" ?

- sol 1: minimize Re-construct Error → PCA
- sol 2: $x|z \sim \mathcal{N}(Wz, \sigma^2 I_D)$ → P-PCA (PCA included)

Note: Reconstruct Error $= \|X - ZW^T\|_F^2$ ~> directly from $x \approx Wz$

2. P-PCA Model

① $x|z \sim \mathcal{N}(Wz, \sigma^2 I_D)$

② $z \sim \mathcal{N}(0, I_K)$

Q: what is $x \sim$?

Answer: $p(x) \propto \int_z \exp\left\{-\frac{1}{2\sigma^2}(x - Wz)^T(x - Wz)\right\} \exp\left\{-\frac{1}{2} z^T z\right\} dz$

$$= \exp\left\{-\frac{1}{2\sigma^2} x^T x\right\} \int_z \exp\left\{-\frac{1}{2\sigma^2} z^T W^T W z - \frac{1}{2} z^T z + \frac{1}{\sigma^2} x^T W z\right\}$$

$$= \exp\left\{-\frac{1}{2\sigma^2}x^T x\right\} \int_z \exp\left\{-\frac{1}{2\sigma^2} z^T (W^T W + \sigma^2 I_K) z + \frac{1}{\sigma^2} z^T W^T x\right\} dz$$

$$\propto \exp\left\{-\frac{1}{2} x^T (WW^T + \sigma^2 I_D)^{-1} x\right\}$$

trick 1: $-\frac{1}{2}(z-\mu)^T \Sigma^{-1}(z-\mu) = -\frac{1}{2} z^T \Sigma^{-1} z + z^T \Sigma^{-1}\mu - \frac{1}{2}\mu^T \Sigma^{-1}\mu$ ← important term

trick 2:
$$W(W^T W + \sigma^2 I_K)^{-1} W^T$$
$$= WW^T (WW^T + \sigma^2 I_D)^{-1}$$
$$= (WW^T + \sigma^2 I_D - \sigma^2 I_D)(WW^T + \sigma^2 I_D)^{-1}$$
$$= I_D - \sigma^2 (WW^T + \sigma^2 I_D)^{-1}$$

That is: $x \sim N(0, WW^T + \sigma^2 I_D)$ → Model Necessary Condition → from assumption

Suppose $X \sim N(0, \Sigma)$

then $WW^T + \sigma^2 I_D \approx \Sigma$

- few parameters: $(D \cdot K + 1)$
- Many parameters: $\frac{D(D+1)}{2}$

$K \ll D$

★ Rmk: fewer parameters, less likely to over-fitting!!!

To conclude: in this model

① $x|z \sim N(Wz, \sigma^2 I_D)$

② $z \sim N(0, I_K)$

③ $\Rightarrow x \sim N(0, WW^T + \sigma^2 I_D)$

3. P-PCA ⟶ Learning parameters

$\|X - ZW^T\|_F^2$ ↗ interpretation for matrix factorization

a) Connection with Re-construction Error (ALT-OPT)

→ only focus $\log P(X|Z, W, \sigma^2)$

$$\propto -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\|x_i - Wz_i\|_2^2 - \frac{ND}{2}\log(2\pi\sigma^2)$$

trick is that: $z \sim N(0, I_k)$

$$= -\frac{1}{2\sigma^2}\|X - ZW^T\|_F^2 - \frac{ND}{2}\log(\sigma^2)$$

→ Suppose σ^2 is fixed! (for simplicity)

ALT-OPT Framework is exactly close to minimize Re-construct Error!

⇒ ① initialize $W = \hat{W}$

② estimate $\hat{Z} = \underset{Z}{\text{argmax}}\ P(Z|X, \hat{W}, \sigma^2)$

$$= \underset{Z}{\text{argmax}}\ \log \frac{P(X|Z, \hat{W}, \sigma^2)\,P(Z)}{P(X|\hat{W}, \sigma^2)}$$

$$= \underset{Z}{\text{argmax}}\ \log P(X|Z, \hat{W}, \sigma^2)$$

$$= \underset{Z}{\text{argmin}}\ \|X - Z\hat{W}^T\|_F^2$$

③ estimate $\hat{W} = \underset{W}{\text{argmax}}\ P(X, \hat{Z}|W, \sigma^2)$

Recap: $Z \sim N(0, I_k)$

$$= \underset{W}{\text{argmax}}\ \log P(X|\hat{Z}, W, \sigma^2)$$

$$= \underset{W}{\operatorname{argmin}} \; \| X - \hat{Z} W^T \|_F^2$$

Remark: step ② & ③ are exactly Multi-output Regression

input $z \in \mathbb{R}^K \xrightarrow{W}$ output $\tilde{x} = Wz \in \mathbb{R}^D$ $\quad W \in \mathbb{R}^{D \times K}$

Loss function: $\frac{1}{2} \| \tilde{x} - x \|_2^2$ (ℓ_2 - loss function)

$X = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$ $\quad Z = \begin{pmatrix} z_1 \\ \vdots \\ z_N \end{pmatrix}$ $\Rightarrow$ N-observation!

→ All the above discussion is not related to MLE!

b) MLE & EM Framework to learn P-PCA parameters

① (Brute-Force MLE)

Recap: $x \sim N(0, WW^T + \sigma^2 I_D)$

it is possible to find the log-likelihood

$$\Rightarrow \ell(W, \sigma) = \log P(X \mid W, \sigma)$$

$$= \sum_{i=1}^{N} \log P(x_i \mid W, \sigma)$$

$x_i \sim N(0, WW^T + \sigma^2 I_D)$

$\Rightarrow$ exact MLE estimator:

$$\begin{cases} \hat{W}_{MLE} = U_K (L_K - \hat{\sigma}^2_{MLE} I)^{\frac{1}{2}} R \\ \hat{\sigma}^2_{MLE} = \frac{1}{D-K} \sum_{k=K+1}^{D} \lambda_k \end{cases}$$

where

$$\begin{cases} U_K = (u_1, \dots, u_K) \quad u_i \in \mathbb{R}^D \longrightarrow \text{the eigenvector of } S = X^T X \\ L_K = \text{Diag}(\lambda_1, \dots, \lambda_K) \\ R \longrightarrow \text{Rotate Matrix} \end{cases}$$

Rmk: ① This solution $\hat{W}_{MLE} \longrightarrow$ PCA when $R = I$ & $\hat{\sigma}^2_{MLE} \to 0$

② It is computationally expensive to achieve the eigenvalue decomposition for $S = X^T X \in \mathbb{R}^{D \times D}$

② EM Framework

Recap: EM algo

E-step: estimate $p(z_n | x_n, \hat{W}, \hat{\sigma}^2)$

$$M = \hat{W}^T \hat{W} + \hat{\sigma}^2 I_K$$

Conclusion: $z_n | x_n, \hat{W}, \hat{\sigma}^2 \sim N(M^{-1} \hat{W}^T x_n, \hat{\sigma}^2 M^{-1})$

Recap: (from the calculate of $P(x)$)

$$P(z|x) \propto \exp\left\{ -\frac{1}{2\sigma^2} z^T (W^T W + \sigma^2 I_K) z + \frac{1}{\sigma^2} z^T W^T x \right\}$$

$$= \exp\left\{ -\frac{1}{2} z^T \underbrace{\left[\sigma^{-2} (W^T W + \sigma^2 I_K) \right]}_{\Sigma^{-1}} z + z^T \underbrace{\sigma^{-2} W^T x}_{\Sigma^{-1} \mu} \right\}$$

(easy calculation)

M-step: $(\hat{W}, \hat{\sigma}^2) = \underset{W,\sigma}{\operatorname{argmax}}\ \mathbb{E}_{Z\sim p(\cdot|X,\hat{W},\hat{\sigma}^2)}[\log p(X,Z|W,\sigma^2)]$

Calculation: $\log p(X,Z|W,\sigma^2)$

$$= \sum_{i=1}^{N}\{\log p(x_n|z_n,W,\sigma^2) + \log p(z_n)\}$$

$$\propto -\sum_{n=1}^{N}\left[\frac{D}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\|x_n\|_2^2 - \frac{1}{\sigma^2}z_n^T W^T x_n\right.$$

omit $\mathbb{E}_{Z\sim p(\cdot|X,\hat{\theta})}$ for simplicity

$$\left. + \frac{1}{2\sigma^2}\operatorname{tr}(z_n z_n^T W^T W) + \frac{1}{2}\operatorname{tr}(z_n z_n^T)\right]$$

$$\Rightarrow \mathbb{E}[\log p(X,Z|W,\sigma^2)]$$

$$= -\sum_{n=1}^{N}\left[\frac{D}{2}\log\sigma^2 + \frac{1}{2\sigma^2}\|x_n\|_2^2 - \frac{1}{\sigma^2}\mathbb{E}[z_n]^T W^T x_n\right.$$

$$\left. + \frac{1}{2\sigma^2}\operatorname{tr}(\mathbb{E}[z_n z_n^T] W^T W) + \frac{1}{2}\operatorname{tr}(\mathbb{E}[z_n z_n^T])\right]$$

then, since $(\hat{W},\hat{\sigma}^2) = \underset{W,\sigma^2}{\operatorname{argmax}}\ \mathbb{E}[\log p(X,Z|W,\sigma^2)]$

$\Rightarrow$ (closed-form solution) ☆ (Matrix Calculus)

then $\begin{cases} \hat{W} = \left[\sum_{n=1}^{N} x_n \mathbb{E}[z_n]^T\right]\left[\sum_{n=1}^{N}\mathbb{E}[z_n z_n^T]\right]^{-1} \\ \hat{\sigma}^2 = \frac{1}{ND}\sum_{n=1}^{N}\{\|x_n\|_2^2 - 2\mathbb{E}[z_n]^T\hat{W}^T x_n + \operatorname{tr}(\mathbb{E}[z_n z_n^T]\hat{W}^T\hat{W})\} \end{cases}$

$$M = W^T W + \sigma^2 I_K$$

obviously, since $z_n|x_n, W, \sigma^2 \sim \mathcal{N}(M^{-1}W^T x_n, \sigma^2 M^{-1})$

then $\begin{cases} \mathbb{E}[z_n] = M^{-1}W^T x_n \\ \mathbb{E}[z_n z_n^T] = \sigma^2 M^{-1} + \mathbb{E}[z_n]\mathbb{E}[z_n]^T \end{cases}$

Rmk: since when $\sigma^2 \to 0$, MLE $\longrightarrow$ PCA

therefore, when we set $\sigma^2 = 0$, the EM framework can be viewed as the efficient algorithm to solve PCA

PCA $\longrightarrow$ eigenvector decomposition

P-PCA $\longrightarrow$ iterative algorithm to solve eigenvector decomp. issue

---

② Mixture P-PCA Model

1. Motivation

Recap: P-PCA Model

$$\Downarrow$$

$$\begin{cases} x_n \mid z_n, W, \sigma^2 \sim \mathcal{N}(Wz_n, \sigma^2 I_D) \\ z_n \sim \mathcal{N}(0, I_k) \end{cases}$$

$\bigstar$ Assumption !!! center at origin

$$\Downarrow$$

$$x_n \sim \mathcal{N}(0, WW^T + \sigma^2 I_D)$$

Guideline

$\Rightarrow$ Before doing P-PCA, we should center the data!

$\longrightarrow$ Another interpretation for centering data $\begin{cases} \text{P-PCA with } \mu \\ \text{P-PCA without } \mu \end{cases}$

PCA without μ

P-PCA Model (Previous Equivalent form)

$$x_n = Wz_n + \varepsilon_n \Leftarrow \begin{cases} \varepsilon_n \sim \mathcal{N}(0, \sigma^2 I_D) \\ z_n \sim \mathcal{N}(0, I_k) \end{cases}$$

$$\Downarrow$$

$\Rightarrow$ center at origin

Modification: $x_n = \mu + W z_n + \varepsilon_n$

↓ PCA with μ

$\Rightarrow \boxed{x_n \sim N(\mu, WW^T + \sigma^2 I_D)}$

⇓ assumption: data has one cluster

$$\hat{\mu}_{MLE} = \frac{1}{N}\sum_{n=1}^{N} x_n$$

Interpretation:

Therefore, if our data has one cluster, but not center at the origin, then we can first center the data and apply P-PCA without μ. This is actually equivalent to apply P-PCA with μ.

2. Mixture of P-PCA Model

$$\begin{cases} ① \ c_n \sim \text{multi-noulli}(\pi_1, \dots, \pi_M) \\ ② \ z_n \sim N(0, I_K) \\ ③ \ x_n | c_n, z_n, \theta \sim N(\mu_{c_n} + W_{c_n} z_n, \sigma_{c_n}^2 I_D) \end{cases}$$

Some induction:

$$\begin{cases} 1. \text{ Parameters: } \{\pi_m, \mu_m, W_m, \sigma_m^2\}_{m=1}^{M} \\ 2. \ x_n | c_n \sim N(\mu_{c_n}, W_{c_n} W_{c_n}^T + \sigma_{c_n}^2 I_D) \\ 3. \ x_n \sim \text{GMM (Low-rank GMM)} \end{cases}$$

3. Advantages & Remarks

Limitation of P-PCA

1. If the data has several clusters, then we cannot use P-PCA since the assumption is violated!

But Mixture of P-PCA still works!

2. Non-linear Dimensionality Reduction

3. Special Case of Gaussian Mixture Model [GMM] → Density Estimation
   (Low-rank form)

4. Combining { learn the clusters
               learn the dimensionality reduction

---

(3) Standard PCA

{ 1. Maximize Variance
  2. Minimize Re-construction Error
  (3). Changing Basis to de-correlated features

1. Maximize Variance (Projection)

$z_i = u_1^T x_i \Longrightarrow$ project towards u_1 direction

and z_i is the corresponding coordinate

Maximize variance

$$\Leftrightarrow \max_{u_1} \frac{1}{N} \sum_{n=1}^{N} (z_i - \bar{z})^2$$

$$\Leftrightarrow \max_{u_1} \frac{1}{N} \sum_{n=1}^{N} u_1^T \boxed{(x_n - \bar{x})(x_n - \bar{x})^T} u_1$$

* center the data (pre-step)

$\Leftrightarrow \max_{u_1} u_1^T S u_1 \qquad S := \frac{1}{N}\sum_{n=1}^{N} \tilde{x}_n \tilde{x}_n^T$

$= \frac{1}{N} X^T X \qquad X = \begin{pmatrix} x_1^T \\ \vdots \\ x_N^T \end{pmatrix}$

→ first-component $\hat{u}_1$

→ second-component $\hat{u}_2 = \begin{cases} \arg\max_{u_2} & u_2^T S u_2 \\ \text{s.t} & \|u_2\| = 1 \\ & u_2^T \hat{u}_1 = 0 \end{cases}$

↓

→ etc...

then $Z = X U_K \qquad \begin{cases} U_K = (u_1, \ldots, u_K) \in \mathbb{R}^{D \times K} \\ Z \in \mathbb{R}^{N \times K} \Rightarrow \text{embedding matrix} \end{cases}$

principal component score

(terminology)

2. Minimize Re-construction Error (DSA5105)

Trick : ① $\tilde{x}_n = \sum_{d=1}^{K} (x_n^T u_d) \cdot u_d$

$= \sum_{d=1}^{K} u_d \cdot (u_d^T x_n)$

$= \sum_{d=1}^{K} (u_d \cdot u_d^T) x_n$

② $\sum_{n=1}^{N} \| x_n - \tilde{x}_n \|_2^2$

$= \sum_{n=1}^{N} \| x_n - U_K \cdot z_n \|_2^2$

$= \| X - Z \cdot U_K^T \|_2^2$

$= \| X - X U_K U_K^T \|_2^2$

③ $\sum_{n=1}^{N} \| x_n - \tilde{x}_n \|_2^2$

$= \sum_{n=1}^{N} \| x_n - \sum_{d=1}^{K} (u_d u_d^T) x_n \|_2^2$

$= \sum_{n=1}^{N} \|x_n\|_2^2 - 2 \sum_{n=1}^{N} x_n^T U x_n + \sum_{n=1}^{N} x_n^T U^T U x_n$

★

$U^T U = \sum_{i=1}^{K} \sum_{j=1}^{K} (u_i u_i^T)(u_j u_j^T)$

$= \sum_{i=1}^{K} \sum_{j=1}^{K} u_i (u_i^T u_j) u_j^T$

$= \sum_{i=1}^{K} u_i (u_i^T u_i) u_i^T$

$= \sum_{i=1}^{K} u_i u_i^T = U$

$= \sum_{n=1}^{N} \|x_n\|_2^2 - \sum_{n=1}^{N} x_n^T U x_n$

$= C - \sum_{n=1}^{N} \sum_{d=1}^{K} x_n^T u_d u_d^T x_n$

$= C - \sum_{d=1}^{K} \sum_{n=1}^{N} u_d^T x_n x_n^T u_d$

$= C - \sum_{d=1}^{K} u_d^T S u_d \qquad S = X^T X$