NMF: Non-negative Matrix Factorization

$\downarrow$

Application $\Rightarrow$ Dimensionality Reduction

---

1. Dimension Reduction

Formulation: n data $\{v_1, \dots, v_n\}$ $v_i \in \mathbb{R}^m$

$\downarrow$

find r basis vector $\{w_1, w_2, \dots, w_r\}$

Then
$$\begin{cases} v_j \approx \sum_{i=1}^{r} w_i \underline{H_{ij}} \rightarrow \underline{\{H_{1j}, \dots, H_{rj}\}} \\ \quad = [w_1, \dots, w_r] \begin{bmatrix} H_{1j} \\ \vdots \\ H_{rj} \end{bmatrix} \end{cases}$$

$\Downarrow$

"representation vector" of v_j

matrix form $V = [v_1, \dots, v_n]$

coordinates

$$= [w_1, \dots, w_r] \cdot \begin{bmatrix} \boxed{\begin{matrix} H_{11} \\ \vdots \\ H_{r1} \end{matrix}} & \cdots & \begin{matrix} H_{1n} \\ \vdots \\ H_{rn} \end{matrix} \end{bmatrix}$$

$$:= W \cdot H$$

Here $\begin{cases} V \in \mathbb{R}^{m\times n} \\ W \in \mathbb{R}^{m\times r} \quad H \in \mathbb{R}^{r\times n} \end{cases}$

$\rightarrow$ Error Measure
$$\begin{cases} 1. \ \|V - WH\|_F^2 \\ 2. \ \|V - WH\|_1 \end{cases}$$
etc...

$\rightarrow$ Constraints
$$\begin{cases} 1. \ \text{Non-negative} \quad W \geq 0 \quad H \geq 0 \\ 2. \ \text{orthogonal} \quad HH^T = I_r \\ 3. \ \text{symmetry} \quad \underline{H = W^T} \quad (m = n\,!) \end{cases}$$

$\boxed{\text{NMF}}$

2. NMF Task

→ given $V \in \mathbb{R}_+^{m \times n}$ and a rank r (of V)

→ output $W \in \mathbb{R}_+^{m \times r}$ $H \in \mathbb{R}_+^{r \times n}$

This is achieved by :
$$\begin{cases} \min_{W,H} & \frac{1}{2} \| V - WH \|_F^2 \\ \text{s.t} & \boxed{W \geq 0} \quad \boxed{H \geq 0} \end{cases}$$

basis

coordinates

Rmk : 1. object $f(W, H)$ is non-convex
but bi-convex (fix one)

$\boxed{\text{interpretation}}$

2. each column of V : represents one image !

(application)

$\boxed{\text{flatten}}$

3. Data

↓

in each image dataset, images are well-aligned

↓

$\boxed{\text{noses are roughly in the same location}}$

4. Application

① $\begin{cases} \text{Face Data } V \\ \boxed{\text{manually set } r} \end{cases}$ → how many items can re-construct one face !

⇒ NMF ⇒ $\boxed{W \in \mathbb{R}^{m \times r} \quad V \in \mathbb{R}^{r \times n}}$ → Low-rank approximation

Interpretation : (BASIS)

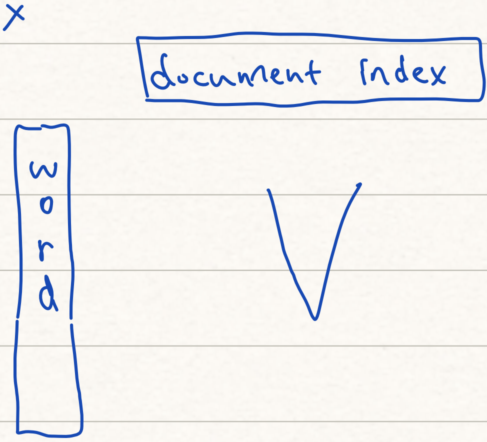
1. each column of W $\Rightarrow$ one type of feature in one image (e.g. mouse, nose e.t.c.)

2. each column of H $\Rightarrow$ the ingredient (intensity) of each feature in matrix W

② Text mining

a) term-document matrix

several sentences $\longrightarrow$



V_{ij} := (# of word i appear in doc j)

b) $V \approx W \cdot H$

interpretation :

- column of W : topic
- column of H : intensity of each topic in one doc

5. Exact NMF

→ given $V \in \mathbb{R}_+^{m \times n}$, if there exists $\begin{cases} W \in \mathbb{R}_+^{m \times r} \\ H \in \mathbb{R}_+^{r \times n} \end{cases}$

s.t $V = W \cdot H$ (perfect recovery)

then [WH is an exact NMF of V]

Rmk :

1. exact NMF may not exist for some r
2. exact NMF may not be unique
3. [non-negative rank] $\text{rank}_+(V)$ ↙ smallest r s.t $V = W_r \cdot H_r$
4. $\boxed{\text{rank}(V) \le \text{rank}_+(V)} \le \min(m, n)$

   (SVD)

   (trivial)

   ⇓ $\boxed{\text{non-negative constraint}}$

5. $V \ge 0$, $\text{rank}(V) \le 2$.

   then $\text{rank}(V) = \text{rank}_+(V)$

6. [conter-example] of 5.)

   when $\text{rank}(V) = 3$, $\text{rank}_+(V)$ can be larger than 3

   ↓

$$V = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \Rightarrow \text{rank}_+(V) = 4$$

6. Variants of NMF $\Rightarrow$ adding prior knowledge

1. Orthogonal NMF $\to$ $\boxed{\text{for } H}$? why
2. Symmetric NMF
3. Sparse NMF ($\|\cdot\|_1$ penalty)

1. Orthogonal NMF

Lemma : $H \geq 0$ + $HH^T = I_r$

$\Downarrow$

each column of H has at most 1 positive member

Pf : $H = \begin{bmatrix} H_{1\cdot} \\ \vdots \\ H_{r\cdot} \end{bmatrix} \Rightarrow (HH^T)_{ij} = H_{i\cdot} H_{j\cdot}^T = 0 \quad i \neq j$

$\Rightarrow$ each row has disjoint support

$\Rightarrow$ each column of H at most has 1 positive entry

Rmk : Lemma says that , Orthogonal NMF $\Rightarrow$ clustering

Then, column of W $\Rightarrow$ $\boxed{\text{cluster centroid}}$

$\boxed{\text{column of } H}$ $\Rightarrow$ $\boxed{\text{cluster membership}}$

membership matrix $H \in \mathbb{R}^{r \times n}$

To solve orthogonal NMF, we add penalty term

Orthogonal NMF Formulation :

$$\begin{cases} \min_{W,H} \ \frac{1}{2} \| V - WH \|_F^2 + \frac{\lambda}{2} \| HH^T - I_r \|_F^2 \\ \text{s.t} \quad W \geq 0 \quad H \geq 0 \end{cases}$$

② Sparse NMF → Difficult to optimize

$$\begin{cases} \min_{W,H} \ \frac{1}{2} \| V - WH \|_F^2 + \lambda_W \| W \|_1 + \lambda_H \| H \|_1 \\ \text{s.t} \quad W \geq 0, \ H \geq 0 \end{cases}$$

③ Adjacency Matrix V (undirected graph G)

$$\boxed{V_{ij} = \begin{cases} 0, & i \not\leftrightarrow j \\ 1, & i \leftrightarrow j \end{cases}} \rightarrow \text{symmety}$$

$\Downarrow$

Symmetric NMF → when V is symmetric

$$\begin{cases} \min_{W} \ \frac{1}{2} \| V - WW^T \|_F^2 \\ \text{s.t} \quad W \geq 0 \end{cases}$$

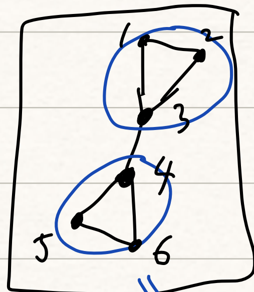
$\boxed{\text{Application}} \rightarrow \begin{cases} \text{Adjacency Matrix of Undirected Graph} \\ \text{Community detection} \end{cases}$

Community detection ↳ $\boxed{\text{dense connected nodes}}$

$\downarrow$

$r = (\#\text{ of communities})$

[E.g.] Graph



$\Downarrow$

$\boxed{r=2}$

$\longrightarrow$ Adjacency Matrix V

$$V = \begin{pmatrix} 0 & 1 & 1 & & & \\ 1 & 0 & 1 & & 0 & \\ 1 & 1 & 0 & 1 & & \\ & & 1 & 0 & 1 & 1 \\ & 0 & & 1 & 0 & 1 \\ & & & 1 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{6\times 6}$$

obviously, we have 2 communities

$$\boxed{W \in \mathbb{R}^{6\times 2}} = \begin{array}{c} \text{node} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \text{community} \\ \begin{array}{cc} 1 & 2 \end{array} \\ \begin{pmatrix} a & 0 \\ b & 0 \\ c & d \\ d & c \\ 0 & b \\ 0 & a \end{pmatrix} \end{array}$$

$d \approx 0.24$

$a \approx b \approx c \approx 0.8$

---

7. Algorithms for NMF

① PG & APG

② MU

③ ALS

④ HALS

③, ④ → block coordinate descent

① PG

$$\min_{W,H} \underbrace{\tfrac{1}{2}\|V - WH\|_F^2}_{f(W,H)} + \delta_{\mathbb{R}_+^{m\times r}}(W) + \delta_{\mathbb{R}_+^{r\times n}}(H)$$

for simplicity, define $\Pi_+(\cdot) := \Pi_{\mathbb{R}_+^{m\times r}}(\cdot) := \Pi_{\mathbb{R}_+^{r\times n}}(\cdot)$

$$\begin{cases} W^{(k+1)} = \Pi_+\left(W^{(k)} - \alpha_k \nabla_W f(W^{(k)}, H^{(k)})\right) \\ H^{(k+1)} = \Pi_+\left(H^{(k)} - \alpha_k \nabla_H f(W^{(k)}, H^{(k)})\right) \end{cases}$$

② MU Algorithm (Variant of PG)

↓

※ $\boxed{\text{w.r.t step length}}$

$$\begin{cases} W^{(k+1)} = \Pi_+\left(W^{(k)} - \underline{S_{W^{(k)}} \odot \nabla_W f(W^{(k)}, H^{(k)})}\right) \\ H^{(k+1)} = \Pi_+\left(H^{(k)} - \underline{S_{H^{(k)}} \odot \nabla_H f(W^{(k)}, H^{(k)})}\right) \end{cases}$$

element-wise multiplication

Here : $\begin{cases} S_{W^{(k)}} := W^{(k)} \oslash [W^{(k)} H^{(k)} H^{(k)T}] \\ S_{H^{(k)}} := H^{(k)} \oslash [W^{(k)T} W^{(k)} H^{(k)}] \end{cases}$

↓

the reason to choose this step length is:

can simplify update equation

⇓

Reason : $W - S_W \odot \nabla_W f$

$= W + S_W \odot RH^T$

$= W + \frac{W}{WHH^T} \odot RH^T$

$$= W + \frac{W}{WHH^T} \odot (VH^T - WHH^T)$$

$$= W \odot VH^T \oslash WHH^T \geq 0$$

③ ALS ($\boxed{\text{Alternating}}$ Least Square) → $\boxed{\text{alternative minimization}}$ (EM-like) (交替下降法)

$\boxed{\text{efficient solver}}$

$$\begin{cases} H \leftarrow \underset{H}{\arg\min} \|V - WH\|_F^2 & \text{(given } W\text{)} \\ H \leftarrow \Pi_+(H) & \\ W \leftarrow \underset{W}{\arg\min} \|V - WH\|_F^2 & \text{(given } H\text{)} \\ W \leftarrow \Pi_+(W) & \end{cases}$$

---

8. Choice of r

Choice of r ↓ $\boxed{\text{Cross-Validation}}$ → $\boxed{\text{k-fold}}$ ↓ $\boxed{\text{Select r}}$ → $\boxed{\text{test}}$

$\boxed{\text{train}}$ ↔ $\boxed{\text{test}}$; $\boxed{\text{train}}$ ↓ $\boxed{\text{k-fold}}$

estimate performance