L18: Multidimensional Scaling ($\boxed{\text{MDS}}$),

Linear Discriminent Analysis ($\boxed{\text{LDA}}$),

and &

Distance Metric Learning ($\boxed{\text{DML}}$)

Principal Component Analysis

Input $A \in \mathbb{R}^{n \times d}$

map $u : \mathbb{R}^d \to \mathbb{R}^k$

goal $\boxed{B \in \mathbb{R}^{n \times k}}$ $b_i = u(a_i)$

$a_i \in$ row in A

$n =$ data points

in low dimensional $\mathbb{R}^k$

so $d(i,j) = \| b_i - b_j \|$

Multidimensional Scaling (MDS)

Input: distance matrix $D \in \mathbb{R}^{n \times n}$

$$D_{ij} = d(i, j)$$

Exemples

[illegible]

$d(i,j)$ = cost of airline flight between i, j

- more abstractly just be given D

Classical MDS

1. Convert D into $D^{(2)}$: $D^{(2)}_{ij} = (D_{ij})^2$

2. Double Centering

centering matrix $C_n = I_n - \frac{1}{n} \mathbb{1}\mathbb{1}^T$

$$M = -\frac{1}{2} C_n D^{(2)} C_n$$ (turns into $n \times n$ inner products)

3. Eigendecomposition $[L, V] = \text{eigs}(M)$

$M = V L V^T = (V L^{1/2})(V L^{1/2})^T$

4. Project onto top k eigenvectors

return $B = V_k L_k^{1/2} \in \mathbb{R}^{n \times k}$

embedded data points $b_i = V_k L_k^{1/2}$

$\in \mathbb{R}^{k \times n}$ $\mathbb{R}^{n \times n}$

Why does MDS work?

Like instead of dist matrix D

unknown $A \in \mathbb{R}^{n \times d}$

rows

↳ similarity matrix S : $S_{ij} = \langle a_i, a_j \rangle$

well $S = AA^T \in \mathbb{R}^{n \times n}$

$S_{ij} = \langle a_i, a_j \rangle$

best embedding of A, from $S = AA^T$

top k eigenvectors S, or top k left singular vectors A

$$\overbrace{\|a_i - a_j\|}^{D_{ij}}{}^2 = \|a_i\|^2 + \|a_j\|^2 - 2\boxed{\langle a_i, a_j \rangle} = S_{ij}$$

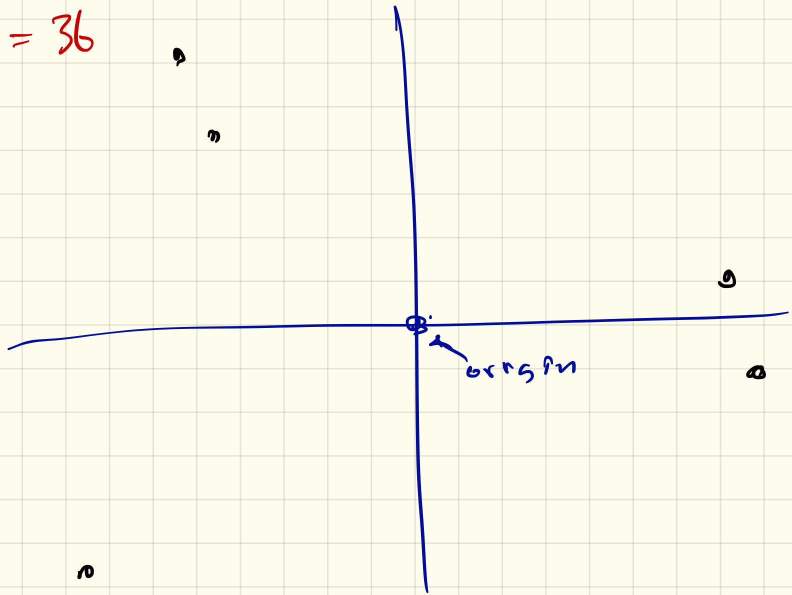
\ trick set $a_1 = (0, 0, \dots 0) \Rightarrow \|a_i\|^2 = \|a_i - a_1\|^2 = D_{i1}^2$

$$S_{ij} = \langle a_i, a_j \rangle = \boxed{-\frac{1}{2}(D_{ij}^2} - D_{i1}^2 - D_{j1}^2)$$

average over all $a_i = 0$

$$D = \begin{bmatrix} 0 & 4 & 3 & 7 & 8 \\ 4 & 0 & 1 & 6 & 7 \\ 3 & 1 & 0 & 5 & 7 \\ 7 & \boxed{6} & 5 & 0 & 1 \\ 8 & 7 & 7 & 1 & 0 \end{bmatrix} \in \mathbb{R}^{5 \times 5}$$

$D_{4,2}$ $\quad$ $D^2_{4,2} = 36$



Linear Discriment Analysis (LDA)

Input $A \in \mathbb{R}^{n \times d}$, also clusters

$S_1, S_2, \dots S_k$

$\bigcup S_j = A \qquad S_i \cap S_j = \phi$
$i \neq j$

Goal: Find the best linear embedding to preserve clusters

(Aside:
t-SNE: find best embedding (non-linear) that preserves cluster structure)

$$\mu_i = \frac{1}{|S_i|} \sum_{x \in S_i} x \quad \text{mean} \quad \in \mathbb{R}^d$$

$$\Sigma_i = \frac{1}{|S_i|} \sum_{x \in S_i} (x-\mu_i)(x-\mu_i)^T \in \mathbb{R}^{d \times d}$$

covariance

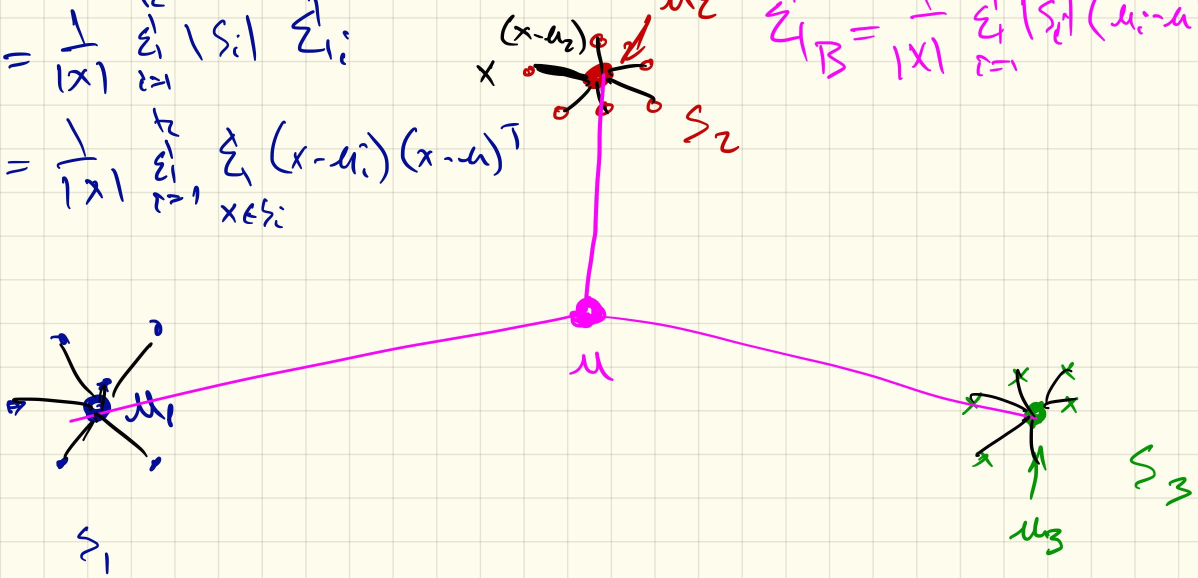
$$\mu = \frac{1}{|X|} \sum_{x \in X} x$$

within class covariance

$$\Sigma_W = \frac{1}{|X|} \sum_{i=1}^{k} |S_i| \, \Sigma_i$$

$$= \frac{1}{|X|} \sum_{i=1}^{k} \sum_{x \in S_i} (x-\mu_i)(x-\mu_i)^T$$

$$\Sigma_B = \frac{1}{|X|} \sum_{i=1}^{k} |S_i| (\mu_i - \mu)(\mu_i - \mu)^T$$



LDA: 1. top k' eigenvectors of

$$\boxed{{\Sigma'_W}^{-1} \Sigma'_B} \in \mathbb{R}^{d \times d}$$

$\hookrightarrow V_{k'}$

2. Project $\hat{X} \leftarrow V_{k'}^T X$

$$\hat{x} = V_{k'}^T x = (\langle x, v_1 \rangle, \langle x, v_2 \rangle, \ldots, \langle x, v_{k'} \rangle)$$

$\in \mathbb{R}^{k'}$

orig data point $\in \mathbb{R}^d$

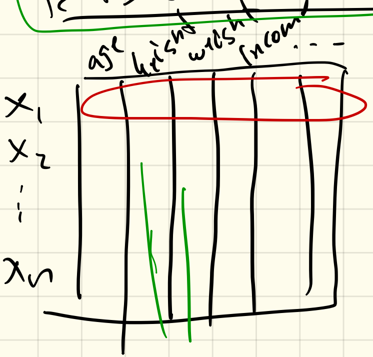
top eigen vector

$$v_1 = \underset{\|u\|=1}{\arg\max} \frac{u^T \Sigma'_B u}{u^T \Sigma'_W u}$$

for top eig Σ'_B

bottom eig Σ'_W

Distance Metric Learning



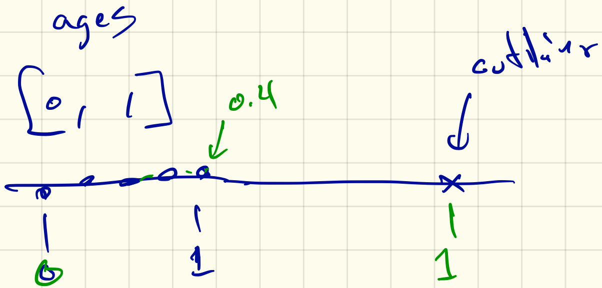
$a_i \in \mathbb{R}^d$

$$\|a_i - a_j\| = \sqrt{(a_i - a_j)^T (a_i - a_j)}$$

$$= \sqrt{\sum_{l=1}^{d} (a_{il} - a_{jl})^2}$$

fine

mixing units



Learn Mahalanobis dist

$$d_M(a_i, a_j) = \|a_i - a_j\|_M = \sqrt{(a_i - a_j)^T M (a_i - a_j)}$$

$M \in \mathbb{R}^{d \times d}$

positive defi

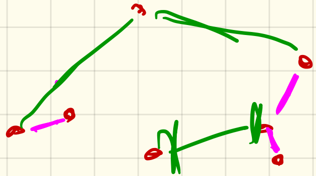
$M \in \mathbb{P}$

Learn $M \in \mathbb{P}^{d\times d}$ d_M

Input $X \in \mathbb{R}^{n\times d}$ far pairs $F \subset X \times X$

close pairs $C \subset X \times X$

$$M^* = \max_{M \in \mathbb{P}} \min_{\{x_i, x_j\} \in F} d_M(x_i, x_j)^2$$



$$\text{s.t.} \sum_{\{x_i, x_j\} \in C} d_M(x_i, x_j)^2 \le K$$

restrict $Tr(M) = d$

$$H = \sum_{\{x_i, x_j\} \in C} (x_i - x_j)(x_i - x_j)^T \in \mathbb{R}^{d\times d}$$

$H = H + \delta I$ makes H full rank

$\Delta = \{\alpha \in \mathbb{R}^{|F|} \mid \sum_{i=1}^{|F|} \alpha_i = 1, \alpha_i \ge 0\}$

probability dist on F.

$T_{ij} \in F$

$X_{T_{ij}} = (x_i - x_j)(x_i - x_j)^T \in \mathbb{R}^{d\times d}$

$$\tilde{X}_T = H^{-1/2} X_T H^{-1/2}$$

$$\underset{M}{\operatorname{argmax}} \min_{\alpha \in \Delta} \sum_{T \in F} \alpha_T \langle \hat{X}_T, M \rangle$$

$$\underset{M \in \mathbb{P}}{\underline{\operatorname{argmax}}} \quad \min_{\alpha \in \Delta} \quad \sum_{\tau \in F}{}' \alpha_\tau \underline{\langle \tilde{X}_\tau, M \rangle}$$

$$\langle X, M \rangle = \sum_{s,t}{}' X_{s,t} M_{s,t} \quad (\text{think of } d_M(X_\tau))$$

<u>optimize</u> (Frank-Wolfe)

smoothin para $\sigma = \text{ob. } 10^{-5}$

gradient

$$g_\sigma(M) = \frac{\sum_{\tau \in F}{}' \exp\left(-\langle \tilde{X}_\tau, M \rangle / \sigma\right) \boxed{\tilde{X}_\tau} \in \mathbb{R}^{d \times d}}{\sum_{\tau \in F}{}' \exp\left(-\langle \hat{X}_\tau, M \rangle / \sigma\right)}$$

w_τ

$$v_{\sigma,M} = \text{top eig}\left(g_\sigma(M)\right)$$

1. Init $M \in \mathbb{P}$ (arbitrarily $M = I$)

2. $v_t = v_{\sigma,M}$ $\leftarrow$ gradient

3. Update $M_t = \frac{t-1}{t} M_{t-1} + \frac{1}{t} \underline{v_t v_t^T}$ $\in \mathbb{R}^{d \times d}$

t steps

Return M