lec-10 Recommender System-4

- Matrix factorization for Clustering
- Hyperparameter tuning for MF algo
- RS with multiple data matrices

Recap

n users , m items

$$X = \begin{array}{c|ccccc|} & I_1 & I_2 & \cdots & I_j & \cdots & I_m \\ u_1 & & & & \vdots & & \\ u_2 & & & & \vdots & & \\ \vdots & & & & \vdots & & \\ u_i & \cdots & & & \boxed{A_{ij}} & & \\ \vdots & & & & & & \\ u_n & & & & & & \end{array} \longrightarrow \text{sparse} \longrightarrow \boxed{\text{Mf}} \longrightarrow \hat{X}$$

$\downarrow$

user item interaction

data completion / data approximation

U = user embedding matrix

V = Item " "

$$X = \underset{n \times d}{U} \; \underset{d \times m}{V^T}$$

$$\underset{n \times d}{U} = \begin{vmatrix} \cdots & \bar{u}_1^T & \cdots \\ \cdots & \bar{u}_2^T & \cdots \\ \cdots & u_3^T & \cdots \\ & \vdots & \\ \cdots & \bar{u}_n^T & \cdots \end{vmatrix} \qquad \underset{m \times d}{V} = \begin{vmatrix} \cdots & \bar{v}_1^T & \cdots \\ \cdots & \bar{v}_2^T & \cdots \\ & \vdots & \\ & \bar{v}_m^T & \end{vmatrix}$$

$$\bar{u}_i, \bar{v}_j \in \mathbb{R}^d$$

$$U V^T = \begin{bmatrix} \cdots & \bar{u}_1^T & \cdots \\ - & \bar{u}_2^T & \cdots \\ & \vdots & \\ - & \bar{u}_N^T & \cdots \end{bmatrix}_{N \times d} \begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ \bar{v}_1 & \bar{v}_2 & \bar{v}_3 & \cdots & \bar{v}_M \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix}_{d \times M}$$

A_{ij}' = predicted interaction = $\bar{u}_i^T \bar{v}_j$

$$\text{loss} = \sum_{i=1}^{N} \sum_{j=1}^{M} \left(A_{ij} - \bar{u}_i^T \bar{v}_j \right)^2$$

$\forall$ i, j $\in$ observed

→ SGD

→ ALS / coordinate descent

SVD → special case of MF

$$X = \underset{n\times n}{U}\;\underset{n\times m}{\Sigma}\;\underset{m\times m}{V^T}$$

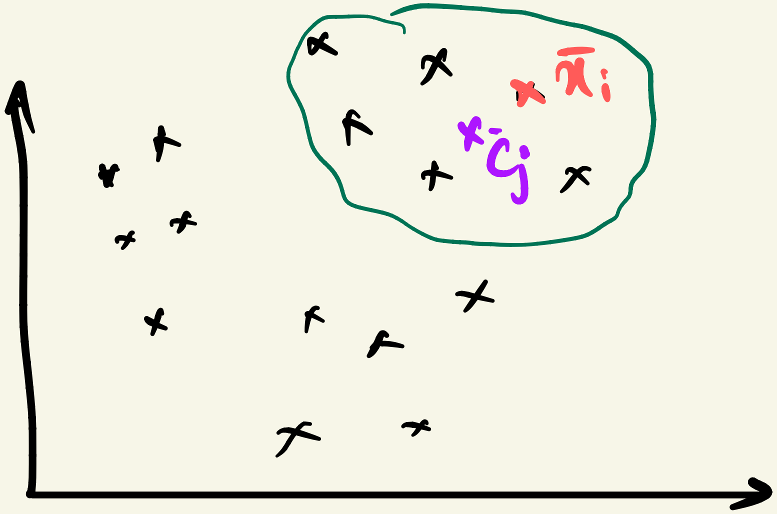
U = user embeddings

V = Item embeddings

Σ = singular value matrix

feature importance

Recap - kMeans



k - clusters $\rightarrow$ k centroids

n - points

$$WCSS = \sum_{j=1}^{k} \sum_{i \in S_j} (\bar{x}_i - \bar{c}_j)^2$$

Goal: min WCSS

all $\bar{x}_i \in R^d$

all $\bar{c}_j \in R^d$

eg $\bar{x}_1 \to c_1$

$x_2 \to c_3$

cluster assignment matrix $= Z =$

$$
\begin{array}{c|cccccc|}
 & c_1 & c_2 & c_3 & \cdots & & c_K \\
\bar{x}_1 & 1 & 0 & 0 & \cdots 0 & 0 \cdots & 0 \\
\bar{x}_2 & 0 & 0 & 1 & \cdots & 0 \cdots & 0 \\
\bar{x}_3 & & & & & & \\
\vdots & & & & & & \\
\bar{x}_i & & & & & & \\
\vdots & & & & & & \\
\bar{x}_N & & & & & &
\end{array}_{n \times K}
$$

$$
C = \begin{vmatrix} \cdots & \bar{c}_1^T & \cdots \\ \cdots & \bar{c}_2 & \cdots \\ \cdots & \bar{c}_3 & \cdots \\ & \vdots & \\ \cdots & \bar{c}_K & \cdots \end{vmatrix}_{K \times d}
$$

Centroid Matrix

Eg.

$$Z = \begin{array}{c} x_1 \\ x_2 \\ x_3 \end{array} \begin{vmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{vmatrix}_{3\times 3} \qquad C = \begin{vmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{vmatrix}_{3\times 3} \begin{array}{c} \bar{c}_1^T \\ \\ \\ \end{array}$$

$$ZC = \begin{array}{c} x_1 \\ x_2 \\ \bar{x}_3 \end{array} \begin{vmatrix} 1 & 2 & 3 \\ -1 & 0 & 1 \\ 1 & 2 & 3 \end{vmatrix} \begin{array}{c} \bar{c}_1^T \\ \bar{c}_3^T \\ \bar{c}_1^T \end{array}$$

$$X = \begin{bmatrix} - & \bar{x}_1^T & - \\ - & \bar{x}_2^T & - \\ - & \bar{x}_3^T & - \\ & \vdots & \\ & \bar{x}_i & \\ & \vdots & \\ & \bar{x}_N^T & \end{bmatrix} \qquad ZC = \begin{bmatrix} c_1^T \\ c_3^T \\ \vdots \\ \bar{c}_j^T \\ \vdots \end{bmatrix}$$

$$\Rightarrow \quad X - ZC = \begin{bmatrix} \bar{x}_1^T - \bar{c}_1^T \\ \bar{x}_2^T - \bar{c}_3^T \\ \vdots \\ \bar{x}_i - \bar{c}_j^T \\ \vdots \end{bmatrix}$$

$$\bar{v} = [1, 2, -4] \Rightarrow \|\bar{v}\| = \sqrt{1^2 + 2^2 + (-4)^2}$$

$$A = \begin{vmatrix} 1 & 2 \\ 3 & -1 \end{vmatrix} \Rightarrow \|A\|_F = \underbrace{\sqrt{1^2 + 2^2 + 3^2 + (-1)^2}}_{\text{frobenius norm of a matrix}}$$

$$\|X - ZC\|_F^2 = \sum_{j=1}^{k} \sum_{i \in S_j} (\bar{x}_i^T - \bar{c}_j)^2 = WCSS$$

KMeans optimization problem $\Rightarrow$ min WCSS

$\rightarrow$ $\min \|X - ZC\|_F^2$ $\quad X, ZC$

$\hookrightarrow$ theoretically min $= 0$

$\Rightarrow X - ZC = 0$

$\Rightarrow X = ZC$ $\Rightarrow$ Matrix factorization

Break X into two factor Z and C

and $Z_{ij} \in \{0, 1\}$

→ Similarly we can formulate a solution to GMM using MF ⇒ $z_{ij} \in [0, 1]$

MF → Data approximation/completion → Rec Sys

MF → PCA, SVD

MF → KMeans (hard assignment)

MF → GMM (soft assignment)

Hyperparameter 'd'

$$X = \begin{array}{c} f_1 \;\; f_2 \;\; f_3 \cdots\cdots f_d \\ \begin{bmatrix} - & \bar{u}_1^T & - \\ - & \bar{u}_2^T & - \\ & \vdots & \\ - & \bar{u}_N^T & - \end{bmatrix}_{N \times d} \\ U \end{array} \begin{array}{c} \begin{bmatrix} | & | & & | \\ \bar{v}_1 & \bar{v}_2 & \cdots & \bar{v}_m \\ | & | & & | \end{bmatrix} \\ V^T \end{array}$$

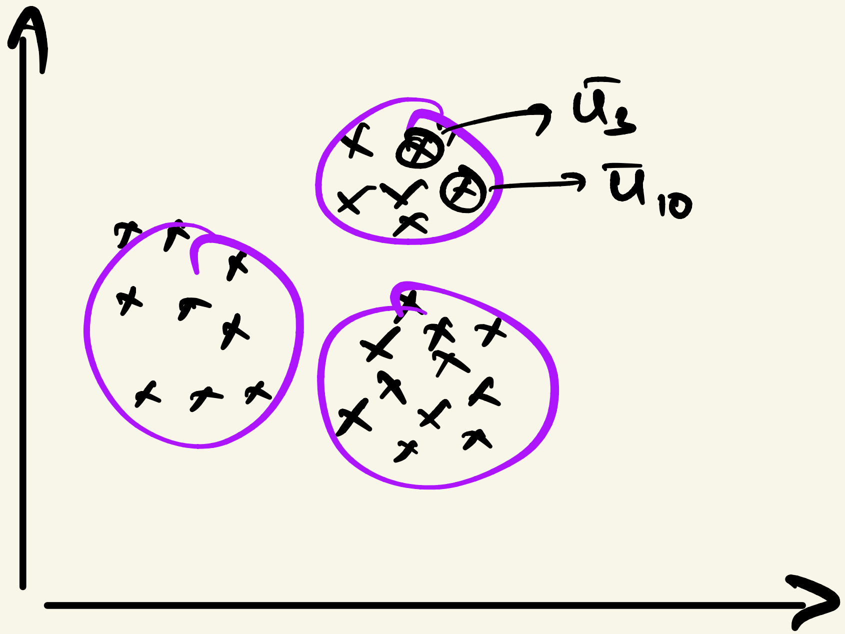
⇒ In the user embedding → 'd' no./features representing each user.

Latent features

↳ no interpretability

Eg. Clustering

↳ TSNE/UMAP



Eg. Similarity scores b/w items and users and recommend items using:

a) Item item similarity

b) User user similarity

$$X = \underset{n \times d}{U} \; \underset{d \times m}{V^T}$$

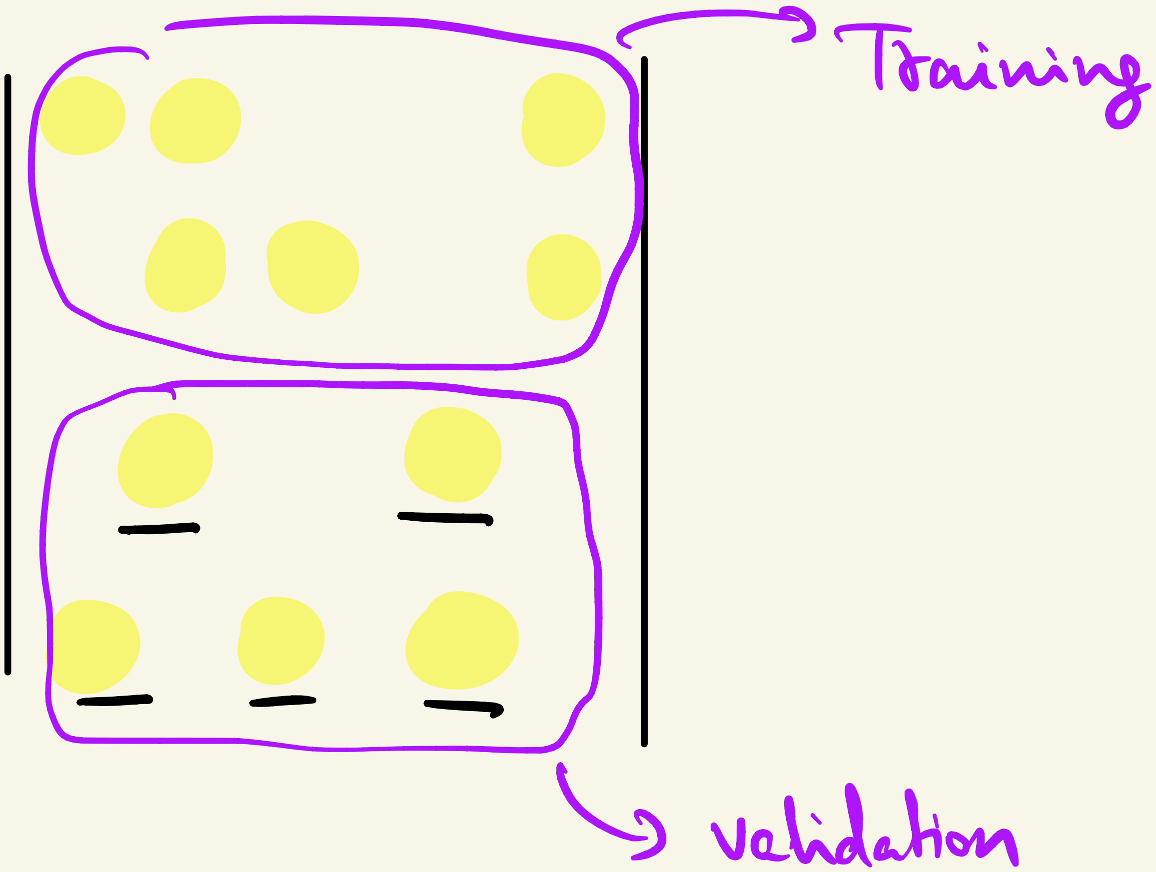
hyperparameters

Eg. $d = 1 \Rightarrow$ underfit

Eg. $d = 100 \Rightarrow$ overfit

Method 1

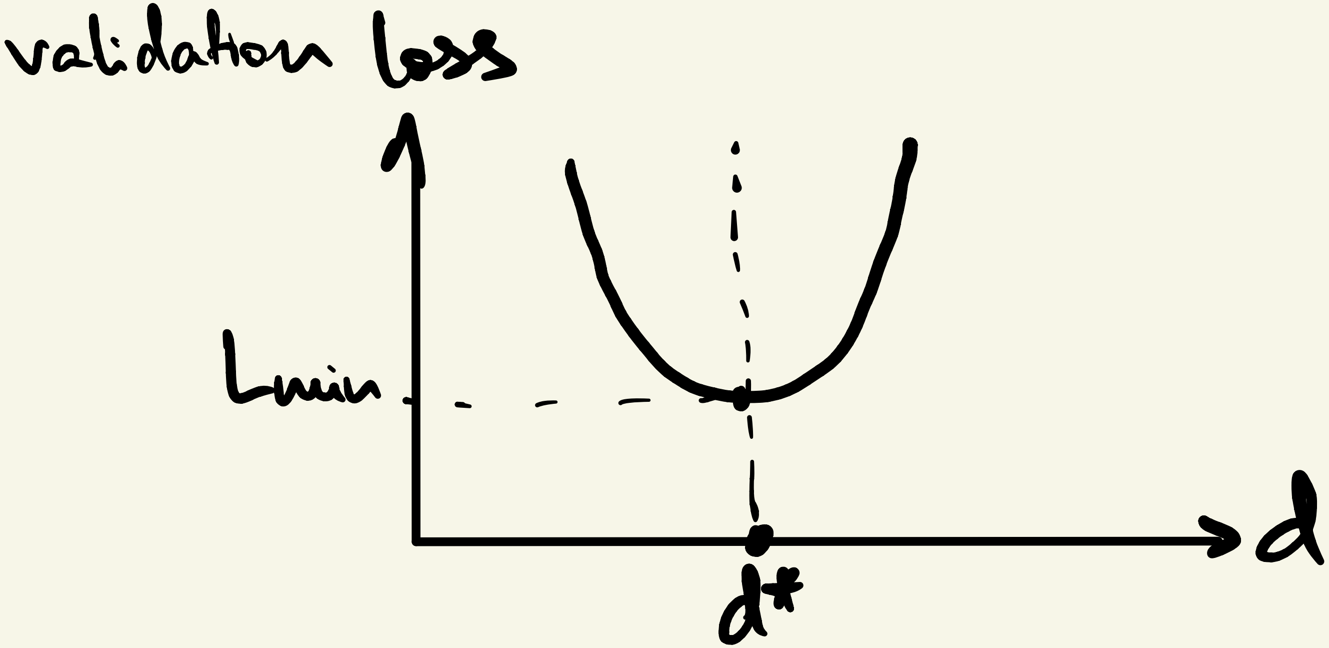
$X =$



in training $\quad \text{loss} = \sum_{i=1}^{n}\sum_{j=1}^{m} \left(A_{ij} - u_i^T \bar{v}_j \right)^2$

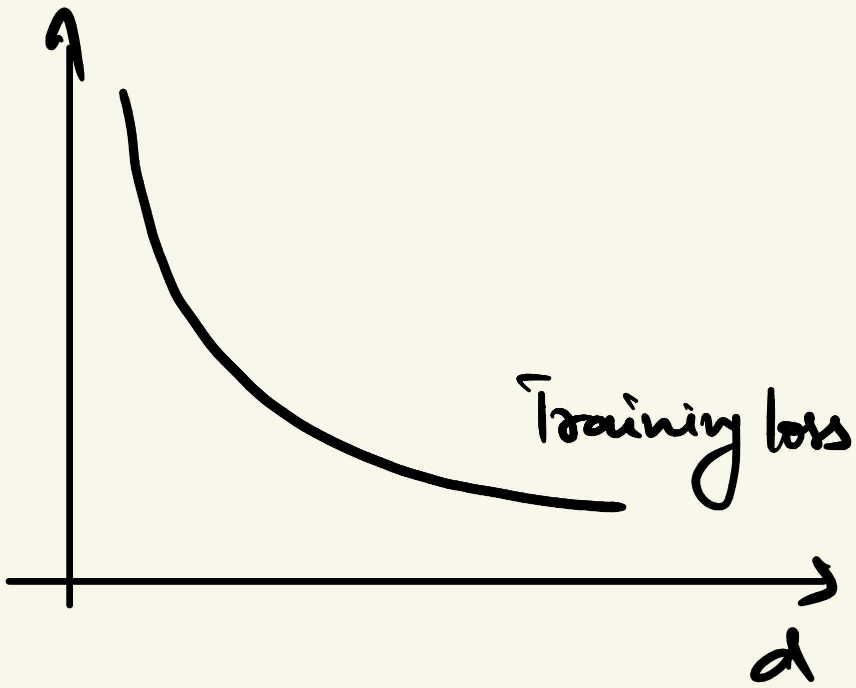
$\forall\, i,j \in$ training

⇒ for different values of d measure the loss on validation dataset



Method 2

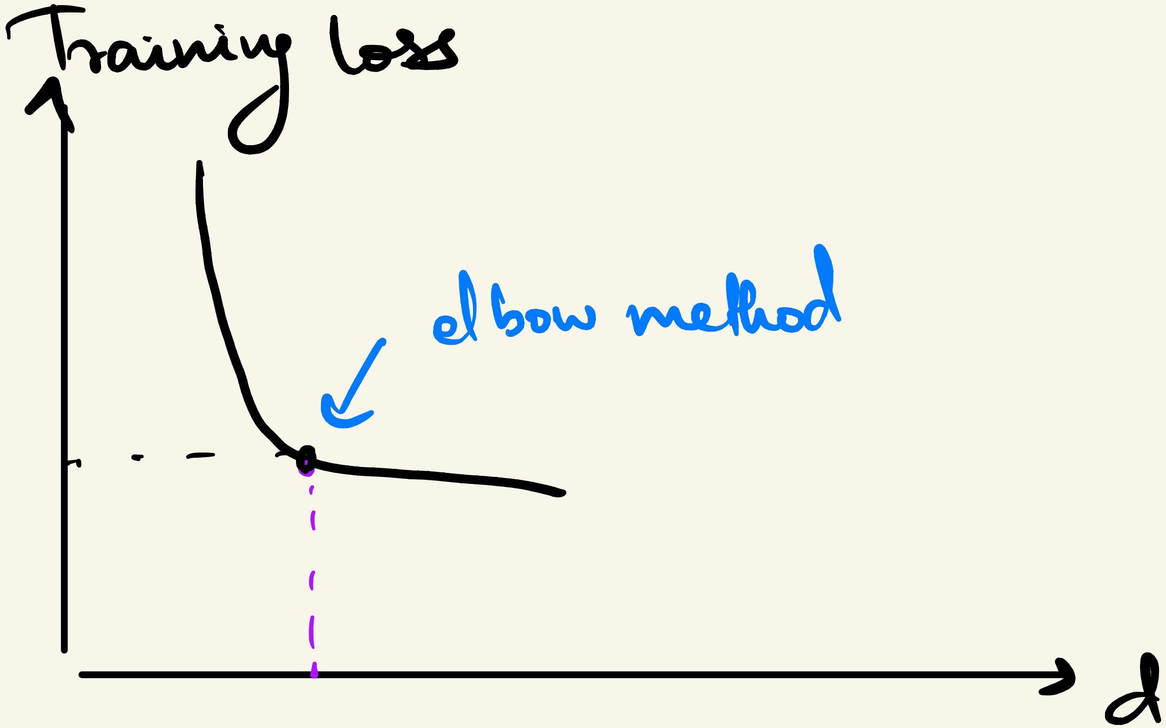
⇒ As d increases, the training loss decreases

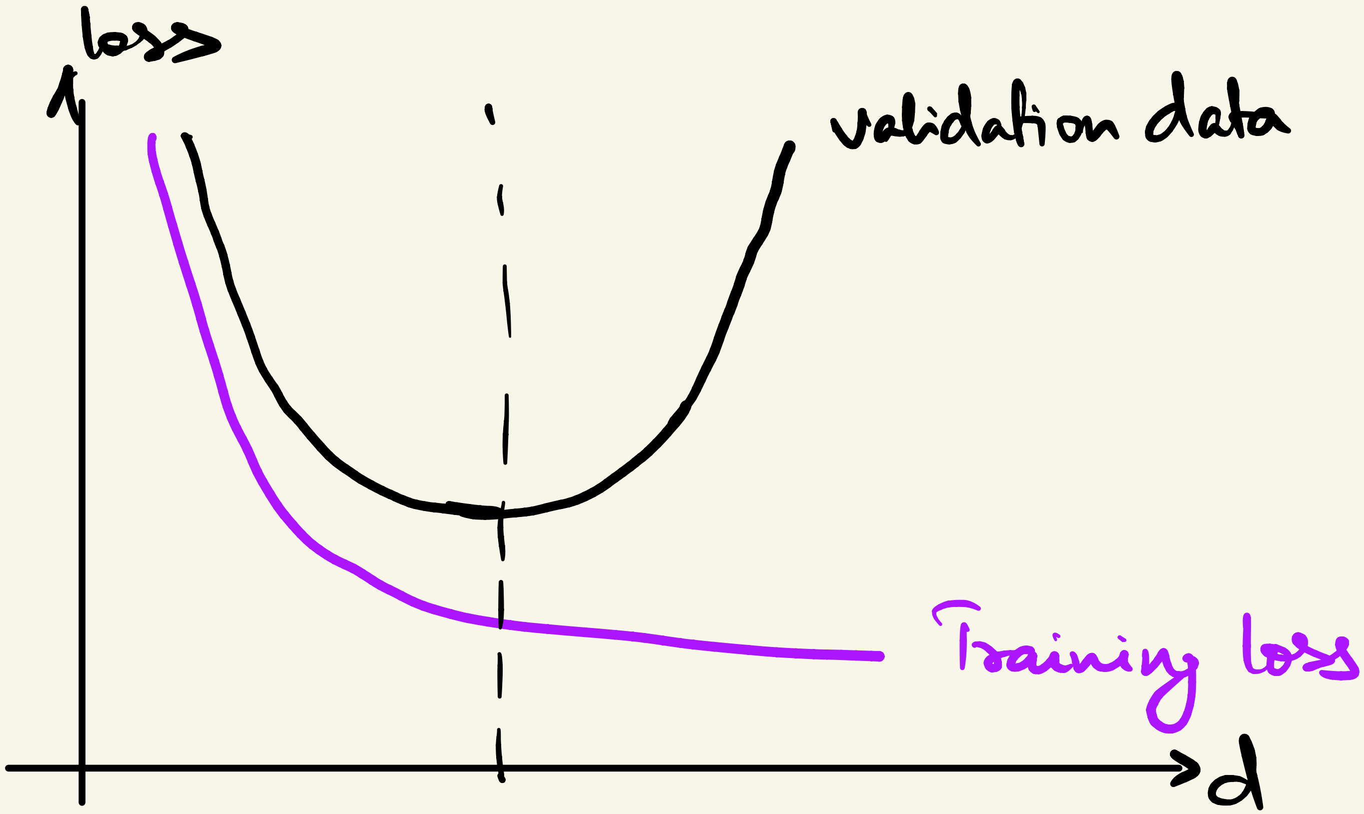


⇒ we can use elbow method to find optimum 'd'

↳ might lead to overfit

Eg.




loss
validation data
Training loss
d

Non-Negative MF (NMF)

Eg. A_{ij} in the user-item interaction matrix

$\rightarrow$ rating given to the item

$\Rightarrow A_{ij} \geq 0$

$X = UV^T \Rightarrow U, V \geq 0$

$\Rightarrow \bar{u}_i, \bar{v}_j \geq 0$

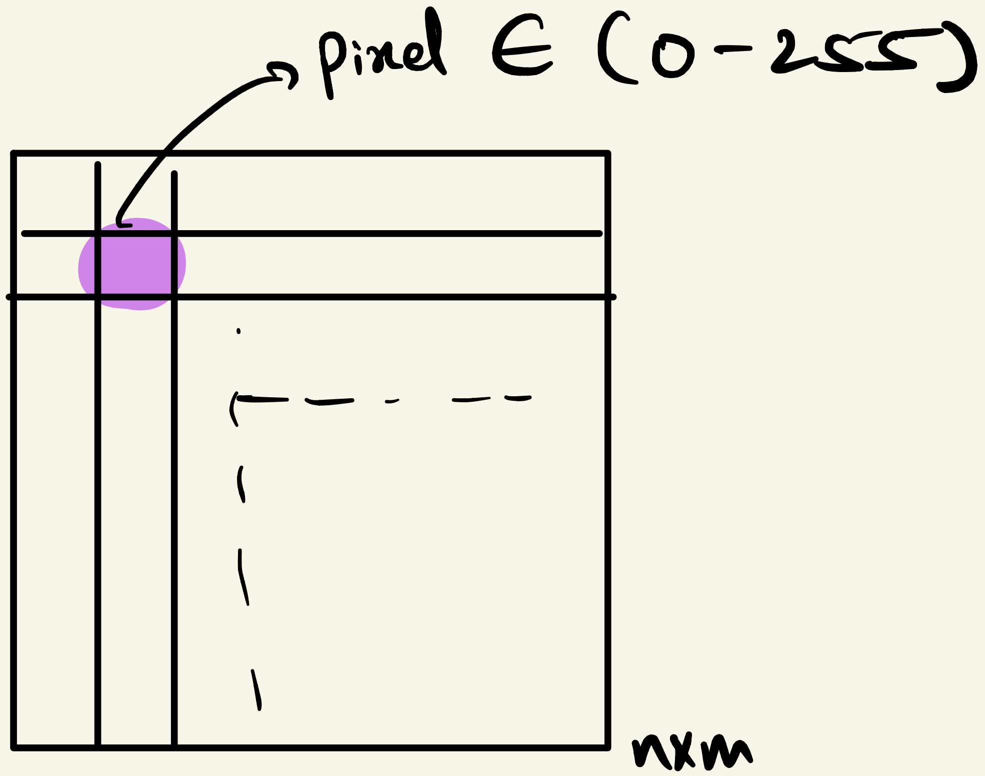
we want all the embeddings to be non-negative

$$\Rightarrow \quad \text{loss} = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(A_{ij} - \bar{u}_i^T \bar{v}_j \right)^2$$

$$\Rightarrow \quad \underbrace{\min \text{ loss} \quad \text{s.t. } \bar{u}_i \geq 0 \text{ and } \bar{v}_j \geq 0}_{\text{non-negative matrix factorization.}}$$

→ <u>Images</u>

Image 1 =
(grayscale)



2K

Image → Resolution = $1920 \times 1080 \Rightarrow \simeq 2 \times 10^6$ pixels

B C

Image 1 = $[\;]_{1920 \times 10} [\;]_{10 \times 1080}$

2×10^6 million

30K

⇒ Here also we can use NMF → image embeddings

HW) google - Eigenfaces

Rec Sys with Multiple datasets

L → interaction matrix → whether user liked the item or not

W → " " → watch time of the user

(alternative → whether the user has watched more than 50% of video or not)

R → Ratings

D → whether the person disliked the item or not

1 → dislike

0 → not disliked

idea 1 : Add all interaction matrices and then do MF

$\downarrow$

Completion

$\downarrow$

Collaborative filtering

$\Rightarrow$ Not a good idea $\rightarrow$ different scales

idea 2) : Adding with weights

$$X = \overset{+1}{\alpha_1} L + \overset{0.1}{\alpha_2} W + \alpha_3 R + \overset{-1}{\alpha_4} D$$

idea 3)

$$
\left.\begin{aligned}
L &= \bar{U}_i^L, \bar{V}_j^L \\
W &= \bar{U}_i^W, \bar{V}_j^L \\
R &= \bar{U}_i^R, \bar{V}_j^R \\
D &= \bar{U}_i^D, \bar{V}_j^D
\end{aligned}\right\} \text{later combine}
$$