Unsupervised Learning: Clustering and Matrix Factorization

EE219: Large Scale Data Mining

Professor Roychowdhury

K-means clustering algorithm

Algorithm

0. Randomly initialize K cluster centers (centroids)
1. Iterate until convergence
   - 1.1 For each data point, find closest cluster center (partitioning step)
   - 1.2 Replace each centroid by average of data points in its partition

Objective function

Write $x_i = (x_{i1}, ...x_{ip})$:
Let the centroids be denoted by $m_1, m_2, ...m_k$ and the clusters by $c_1, c_2, ...c_k$, then the objective function of K-means is to minimize Euclidean distance of the points with the centroids of corresponding clusters (within cluster sum of squares):

$$\sum_{k=1}^{K} \sum_{i \in c_k} ||x_i - m_k||^2$$

$$m_k = \frac{1}{N_k} \sum_{i \in K_k} x_i$$

- Consider the assignment function $C(i)$:

$$C : 1, 2, ...N \rightarrow (1, 2, ...K)$$

- K-means minimizes $W(C)$

$$\begin{aligned} W(C) &= \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} ||x_i - x_j||^2 \\ &= \sum_{k=1}^{K} N_k \sum_{C(i)=k} ||x_i - \bar{x}_k||^2 \end{aligned}$$

  A proof for this equivalence is given in the following slide

- K-means solves the following problem to find assignment function C:

$$\min_{C, m_1 ... m_k} \sum_{k} \sum_{C(i)=k} ||x_i - m_k||^2$$

The outer summation ($k = 1$ through K) is over different clusters. The summands for each k are data within the cluster k. So we just prove the equivalence for each cluster k. In other words, we show that if the number of data points in cluster k is N_k, then:

$$\sum_{C(i)=k} \sum_{C(j)=k} \|x_i - x_j\|^2 = \sum_{i=1}^{N_k} \sum_{j=1}^{N_k} \|x_i - x_j\| = N_k \sum_{i=1}^{N_k} \|x_i - \bar{x}\|^2$$

$$\begin{aligned}
&\sum_{i=1}^{N_k} \left(\sum_{j=1}^{N_k} \|x_i - x_j\|^2 \right) \\
&= \sum_{i=1}^{N_k} \left(\sum_{j=1}^{N_k} \|(x_i - \bar{x}) - (x_j - \bar{x})\|^2 \right) \\
&= \sum_{i=1}^{N_k} \left(\sum_{j=1}^{N_k} \left[(x_i - \bar{x}) - (x_j - \bar{x})\right]^T \left[(x_i - \bar{x}) - (x_j - \bar{x})\right] \right) \\
&= \sum_{i=1}^{N_k} \left(\sum_{j=1}^{N_k} \left(\|x_i - \bar{x}\|^2 + \|x_j - \bar{x}\|^2 - 2(x_i - \bar{x})^T (x_j - \bar{x}) \right) \right)
\end{aligned}$$

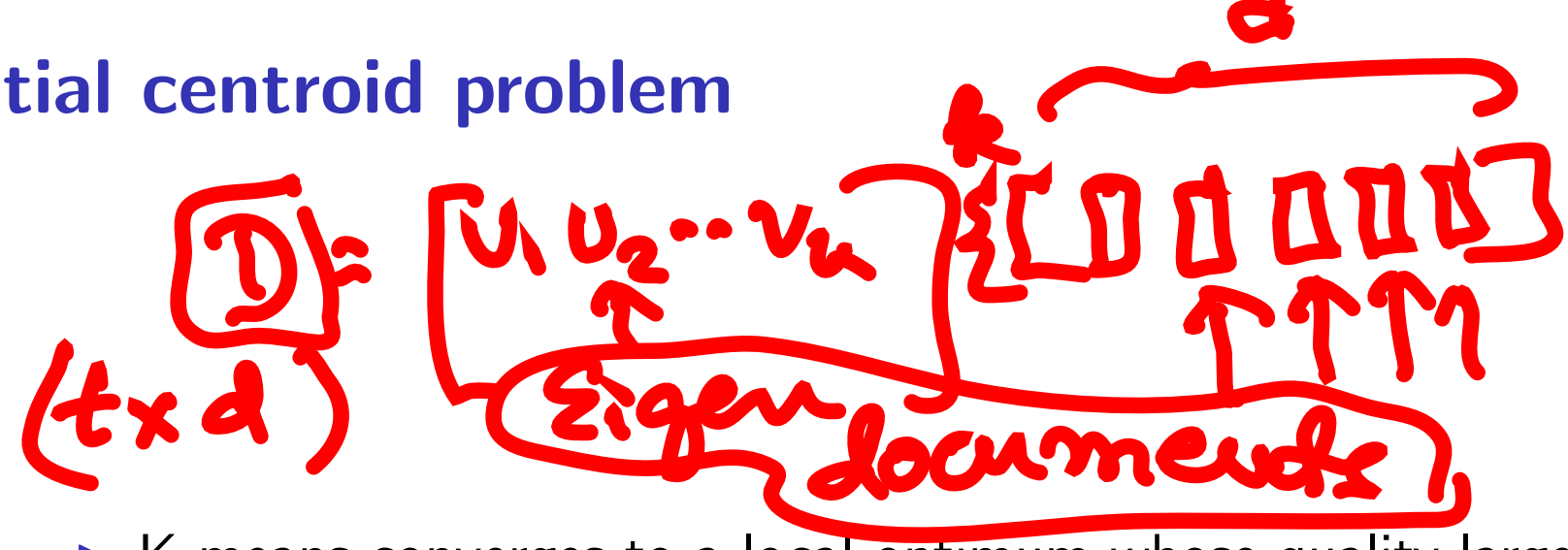
$$
\begin{aligned}
&= \sum_{i=1}^{N_k} \left(\sum_{j=1}^{N_k} \left(\|x_i - \bar{x}\|^2 + \|x_j - \bar{x}\|^2 - 2(x_i - \bar{x})^T (x_j - \bar{x}) \right) \right) \\
&= \sum_{i=1}^{N_k} \left(N_k \|x_i - \bar{x}\|^2 + \sum_{j=1}^{n} \|x_j - \bar{x}\|^2 - 2 \sum_{j=1}^{n} (x_i - \bar{x})^T (x_j - \bar{x}) \right) \\
&= \sum_{i=1}^{N_k} \left(2N_k \|x_i - \bar{x}\|^2 - 2(x_i - \bar{x})^T \sum_{j=1}^{n} (x_j - \bar{x}) \right) \\
&= 2N_k \sum_{i=1}^{N_k} \left(\|x_i - \bar{x}\|^2 \right)
\end{aligned}
$$

Thus:

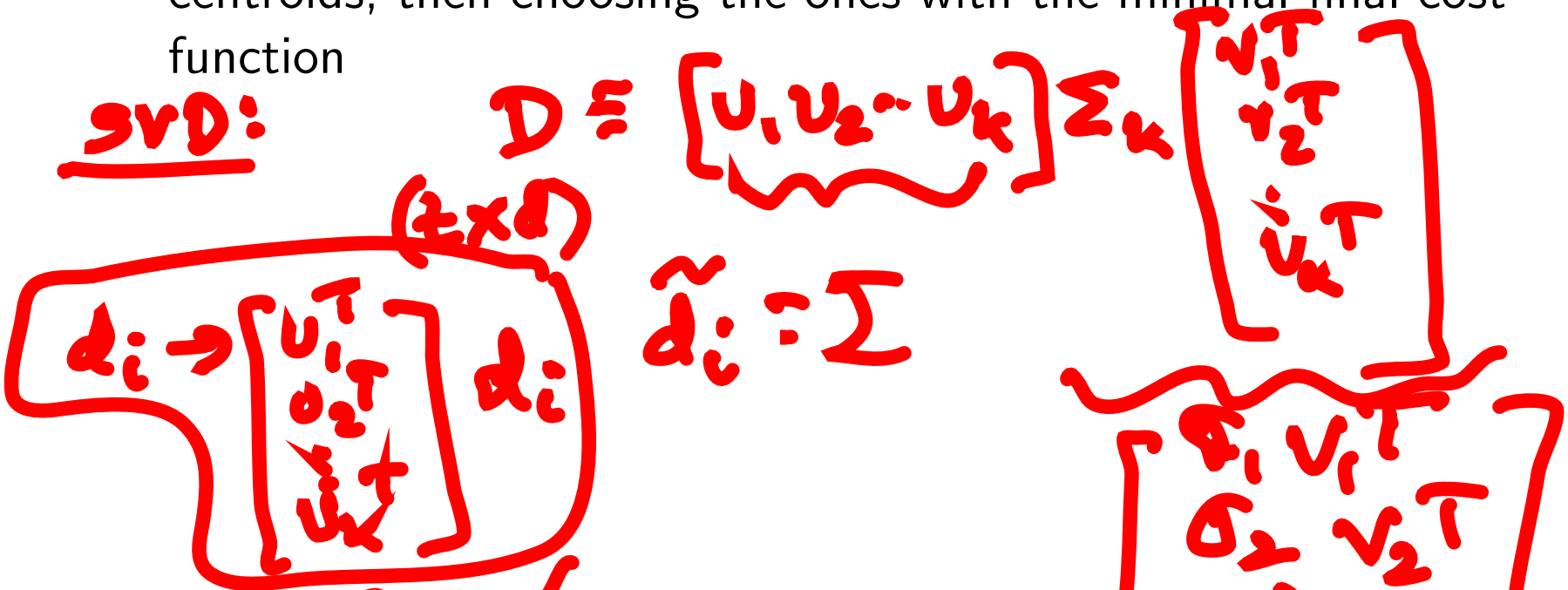
$$
\sum_{i=1}^{N_k} \left(\sum_{j=1}^{N_k} \|x_i - x_j\|^2 \right) = 2N_k \sum_{i=1}^{N_k} \|x_i - \bar{x}\|^2 \Rightarrow
$$

$$
\sum_{\substack{i,j=1 \\ i<j}}^{N_k} \|x_i - x_j\|^2 = N_k \sum_{i=1}^{N_k} \|x_i - \bar{x}\|^2
$$

Initial centroid problem

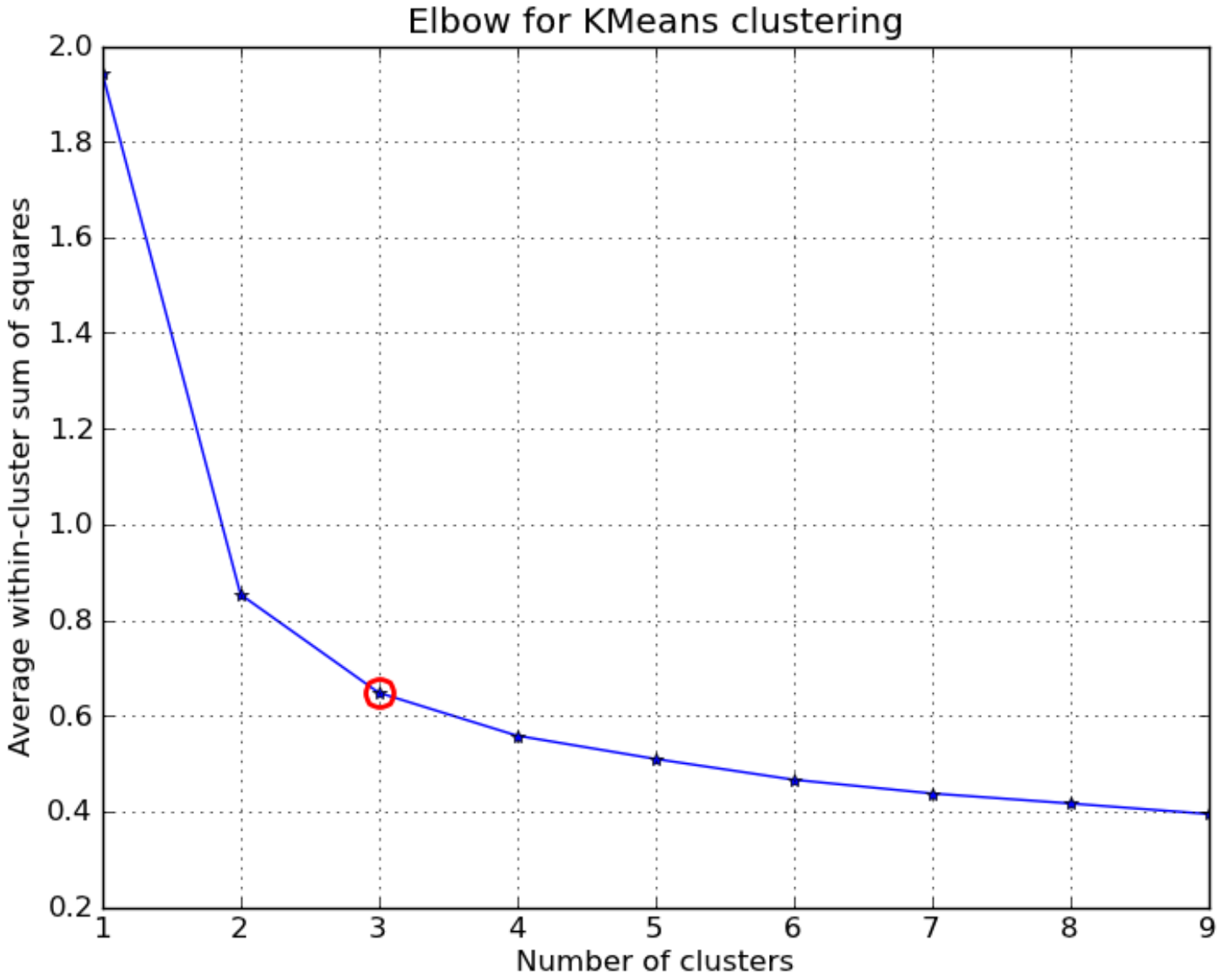


- K-means converges to a local optimum whose quality largely depends on the initial choice of centroids
- Solution: multiple (e.g. 100) runs with random initial cluster centroids, then choosing the ones with the minimal final cost function



How to choose K?

- Elbow method



$$X = \underset{m \times n}{\begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ x_1 & x_2 & \cdots & x_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}}$$

$$XX^T = \sum_{i=1}^{n} x_i x_i^T$$

$$(m \times m)$$

$$= \begin{bmatrix} z_1 & \cdots & z_m \end{bmatrix} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_m \end{bmatrix} \begin{bmatrix} z_1^T \\ \vdots \\ z_m^T \end{bmatrix}$$

$$\lambda_1 z_1 z_1^T + \lambda_2 z_2 z_2^T + \cdots + \lambda_m z_m z_m^T$$

$$(XX^T) z_i = \lambda_i z_i$$

$X X^T \approx$

$$\begin{bmatrix} z_1 & z_2 & \cdots & z_k \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_k \end{bmatrix} \begin{bmatrix} z_1^T \\ \vdots \\ z_k^T \end{bmatrix}$$

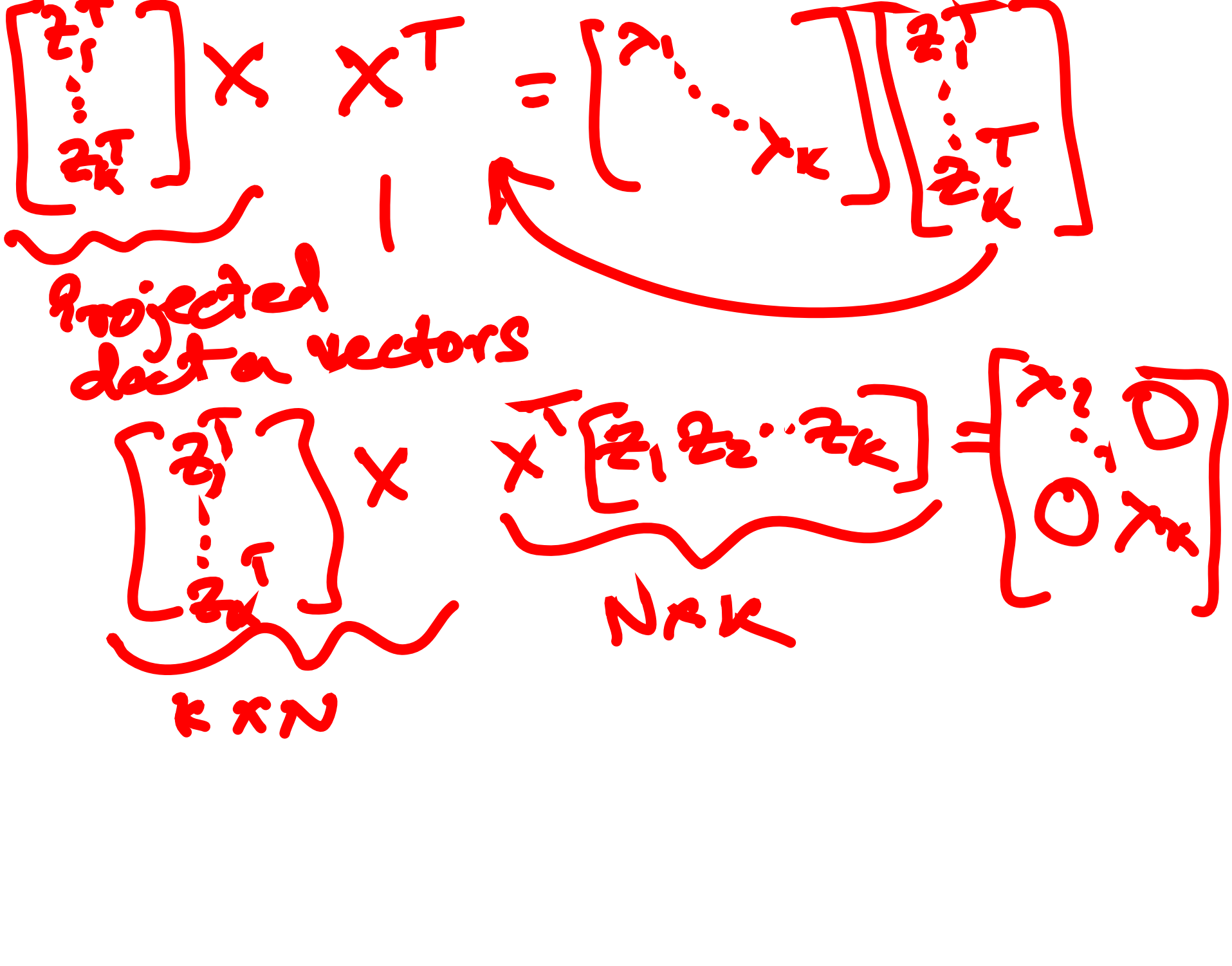
$$\approx \sum_{i=1}^{k} \lambda_i z_i z_i^T$$

$x_i \longrightarrow$ $\underset{m \times 1}{\quad}$ $\underset{k \times 1}{\quad}$

$$\begin{bmatrix} z_1^T x_i \\ \vdots \\ z_k^T x_i \end{bmatrix}$$

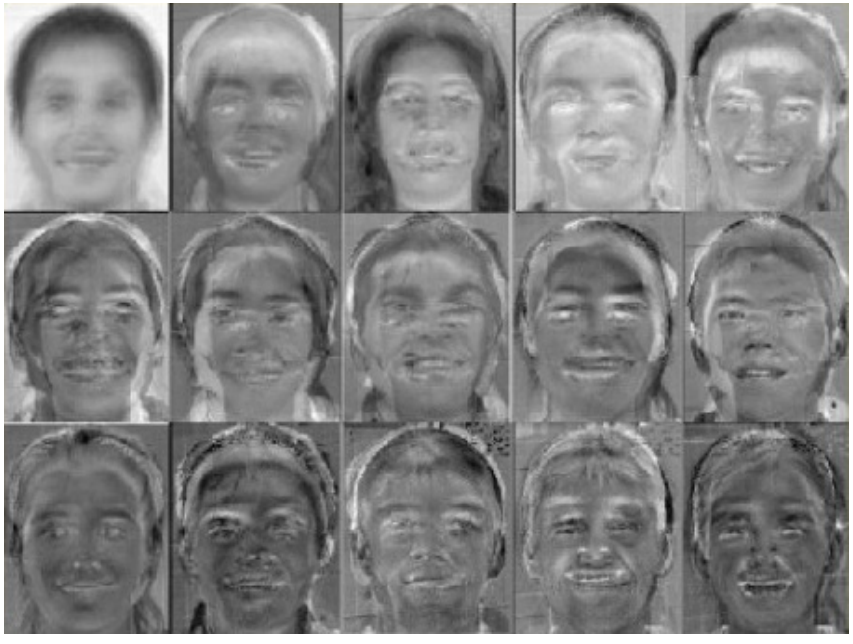
$$\begin{bmatrix} z_1^T \\ \vdots \\ z_k^T \end{bmatrix} X X^T \approx \begin{bmatrix} z_1^T \\ z_2^T \\ \vdots \\ z_k^T \end{bmatrix} \begin{bmatrix} z_1 & z_2 & \cdots & z_k \end{bmatrix}$$

$\cdots$


projected data vectors
k x N
N x k

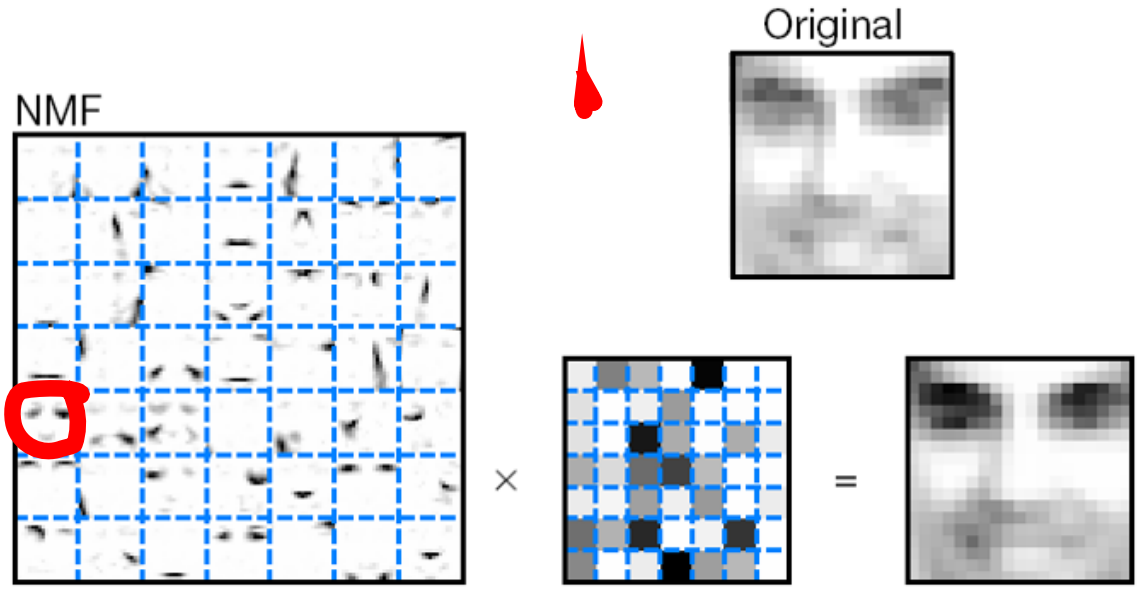
Non-negative matrix factorization (NMF)
Motivation

- PCA involves some basis adding some images and then subtracting others Eigen images (e.g. faces) are not intuitive



- Similar arguments in the context of document classification: subtracting doesn't make intuitive sense

- NMF is like PCA, except the coefficients in the linear combination cannot be negative.
- In other words, we only allowing adding of basis images to make intuitive sense
- In the context of images, non-negative basis images represent *"parts"*



- We don't produce optimal (in norm-2 sense) basis images like PCA, however enforcing the non-negativity constraint gives better properties for classification tasks

Objective function

- Norm distance

$$\|A - B\|_F^2$$

where $\|\cdot\|_F$ denotes the Frobenius norm:

$$\|A\|_F = \left(\sum_{i,j=1}^{n} |a_{ij}|^2 \right)^{1/2}$$

- KL-divergence

$$D\left(A||B\right) = \sum_{ij} \left(A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij} \right)$$

Objective function

- Matrix V is the original $m \times n$ matrix, and matrices $W_{m \times r}$ and $H_{r \times n}$ give a lower rank approximation for V
- We can choose either of the following problems to solve
  1. minimize $\|V - WH\|_F^2$ s.t. $W, H \geq 0$
  2. minimize $D(V||WH)$ s.t. $W, H \geq 0$
- Using gradient descent to find a local minimum, the gradient descent update rule for the norm distance cost function becomes [Lee, Seung '01]

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu} \left[(W^T V)_{a\mu} - (W^T W H)_{a\mu} \right]$$

A nice property of the update rule

- The update rule is

$$H_{a\mu} \leftarrow H_{a\mu} + \eta_{a\mu}\left[(W^T V)_{a\mu} - (W^T W H)_{a\mu}\right]$$

- We set $\eta_{a\mu} = \frac{H_{a\mu}}{(W^T W H)_{a\mu}}$
- Then the update rule becomes

$$H_{a\mu} \leftarrow H_{a\mu} \frac{(W^T V)_{a\mu}}{(W^T W H)_{a\mu}}$$

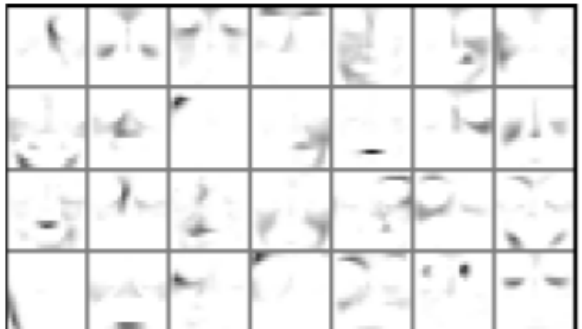
- Similarly for W we get

$$W_{ia} \leftarrow W_{ia} \frac{(V H^T)_{ia}}{(W H H^T)_{ia}}$$

- This is a multiplicative update instead of an additive update. If the initial values of W and H are both non-negative, then W and H will always remain non-negative.
- This guarantees a non-negative factorization.

Sparsity

- NMF doesn't always give parts based result [Hoyer '04]
- In figure below, on the left, a dataset of $m = 2429$ facial images, each consisting of $n = 19 \times 19$ pixels, is factorized over $r = 49$ basis images; and the basis images are displayed. On the right, another dataset of $m = 400$ images having $n = 112 \times 92$ pixels is factorized over $r = 25$ basis images.
- The factors on the right are different in that they are qualitatively global rather than local.
- The difference in results is attributed to the fact that in the dataset used for the experiment on the left, facial images are well aligned, whereas in the experiment on the right the faces have significant variations in face angles, illumination angle, facial expressions, etc.



Sparsity

- We put a constraint on a sparsity measure defined as follows:

$$sparseness(x) = \frac{\sqrt{n} - (\sum |x_i|)/\sqrt{\sum x_i^2}}{\sqrt{n} - 1}$$

- This function evaluates to unity if and only if x contains only a single non-zero component, and takes a value of zero if and only if all components are equal (up to signs), interpolating smoothly between the two extremes.

For $X \in \mathbb{R}^{n \times n}$, by definition, $Tr(X) = \sum_{i=1}^{n} X_{ii}$. Let us define the inner product of two matrices $A, B \in \mathbb{R}^{n \times n}$ as:

$$\langle A, B \rangle = Tr(AB^T)$$

$\|A\|_F^2$
$= Tr(AA^T)$
$= Tr(A^TA)$

i.e. it has the following properties:

1. Symmetry:
$$\langle A, B \rangle = \langle B, A \rangle$$

2. Linearity in the first argument:
$$\langle aA, B \rangle = a \langle A, B \rangle$$
$$\langle A, B + C \rangle = \langle A, B \rangle + \langle A, C \rangle$$

3. Positive-definiteness:
$$\langle A, A \rangle \geq 0$$
$$\langle A, A \rangle = 0 \Leftrightarrow A = 0$$

Given the aforementioned definitions we have:

$$\begin{aligned} \|V - WH\|_F^2 &= \langle V - WH, V - WH \rangle \\ &= \langle V, V \rangle + \langle WH, WH \rangle - 2\langle V, WH \rangle \\ &= Tr\left(V^T V\right) + Tr\left(H^T W^T WH\right) - 2Tr\left(V^T WH\right) \end{aligned}$$

Taking the gradient of the objective function we get the following:

$$\begin{aligned} \nabla_H \|V - WH\|_F^2 &= \nabla_H \left[Tr\left(H^T W^T WH\right) - 2Tr\left(V^T WH\right) \right] \\ &= 2\left(W^T WH - V^T W\right) \end{aligned}$$

Where we used:

$$\nabla_X Tr\,(AX) = A$$
$$\nabla_X Tr(X^T AX) = (A + A^T)X$$

- The proof of $Tr(.)$ properties are straightforward from definition. Below we prove the second property as an example:

$$\begin{aligned} y &= Tr(X^T A X) \\ &= \sum_i \left(\sum_j \sum_k (X^T)_{ij} A_{jk} X_{ki} \right) \\ &= \sum_{i,j,k} X_{ji} X_{ki} A_{jk} \end{aligned}$$

$$\begin{aligned} \frac{\partial y}{\partial X_{mn}} &= \sum_{i,j,k} \left(X_{ki} A_{jk} |_{j=m,i=n} + X_{ji} A_{jk} |_{k=m,i=n} \right) \\ &= \sum_{i,j,k} (X_{kn} A_{mk}) + \sum_{i,j,k} (X_{jn} A_{jm}) \\ &= (AX)_{mn} + (A^T X)_{mn} \end{aligned}$$