Linear Algebra & Convex Optimization – Lecture 10

SVD Applications

References: Online

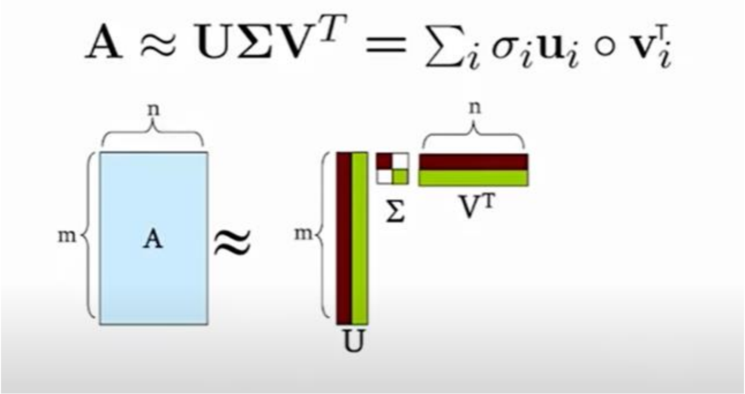
SVD : Matrix Approximation

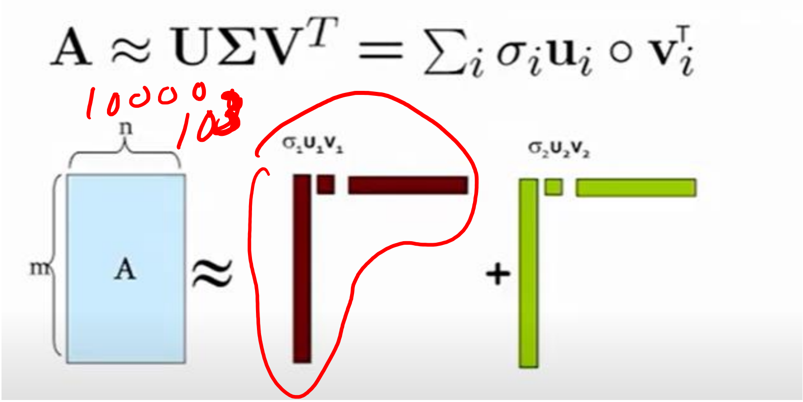
$\underline{\sigma_1} >> \sigma_2 > \sigma_3 \ldots$

$u_1 \Rightarrow 10^4 \times 1$

$v_1 \Rightarrow 10^3 \times 1$

$\sigma_1 \Rightarrow 1 \times 1$





10000

10^3

10000

10^4

Linear Transformation of x via Ax can be considered as :

1. Linear transformation using individual matrices $\sigma_i u_i v_i^T$.

2. Summation of linear transformation using rank-1 matrices $\sigma_i u_i v_i^T$.

$A_{m \times n} = U \Sigma V^T$

rank-r

$\underbrace{\begin{bmatrix} u_1 & u_2 & \cdots & u_r \end{bmatrix}}_{r}$ $m \times r$

$\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}$

Singular value

SVD – Perspective ($m = n = 2, r = 2$)

$$
\begin{array}{ccccccc}
 & U & S & V^T & x & & \\
\underbrace{A}_{2\times 2}\underbrace{x}_{2\times 1} = & \underbrace{\begin{pmatrix} | & | \\ \hat{u}_1 & \hat{u}_2 \\ | & | \end{pmatrix}}_{2\times 2} & \begin{pmatrix} s_1 & \\ & s_2 \end{pmatrix} & \begin{pmatrix} -\hat{v}_1^T- \\ -\hat{v}_2^T- \end{pmatrix} & \begin{pmatrix} \; \end{pmatrix} & = & \underbrace{y}_{2\times 1}
\end{array}
$$

$$
s_1(\hat{u}_1\hat{v}_1^T)\vec{x} + s_2(\hat{u}_2\hat{v}_2^T)\vec{x} \qquad \underbrace{\begin{bmatrix} s_1(\hat{v}_1^T \bullet \vec{x}) \\ s_2(\hat{v}_2^T \bullet \vec{x}) \end{bmatrix}}_{2\times 1} \qquad \underbrace{\begin{bmatrix} \hat{v}_1^T \bullet \vec{x} \\ \hat{v}_2^T \bullet \vec{x} \end{bmatrix}}_{2\times 1}
$$

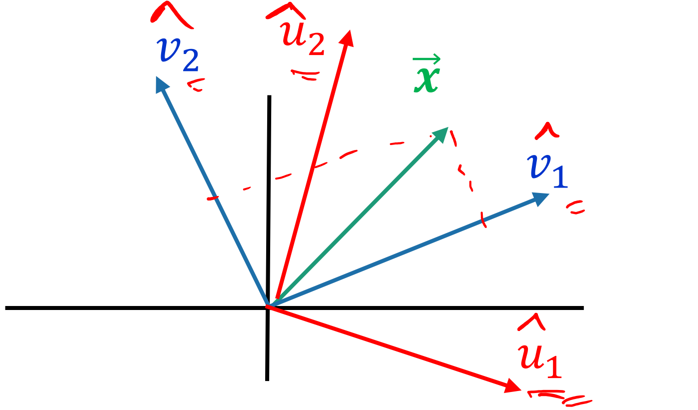
Linear Transformation of x via Ax is in 3 stages:

$$\left(\sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T\right) x$$

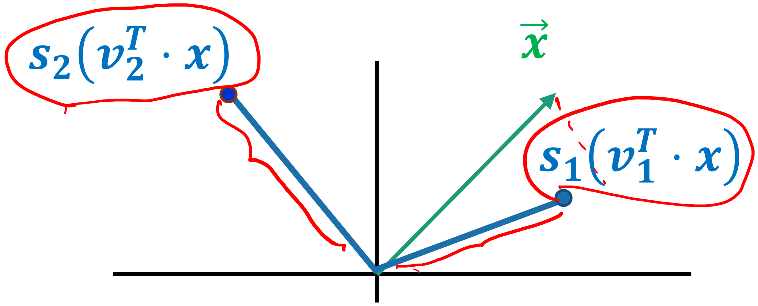
1. Input transformation or vector x with Matrix V^T
2. Scaling with Matrix S

$$= \sigma_1 u_1 v_1^T x + \sigma_2 u_2 v_2^T x$$

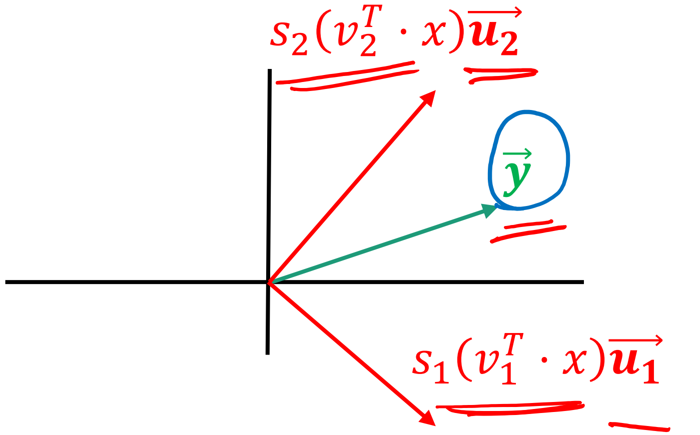
3. Output vector formation with columns of Matrix U



i) Input with Left & Right Singular Vectors



ii) Input Transformation & Scaling

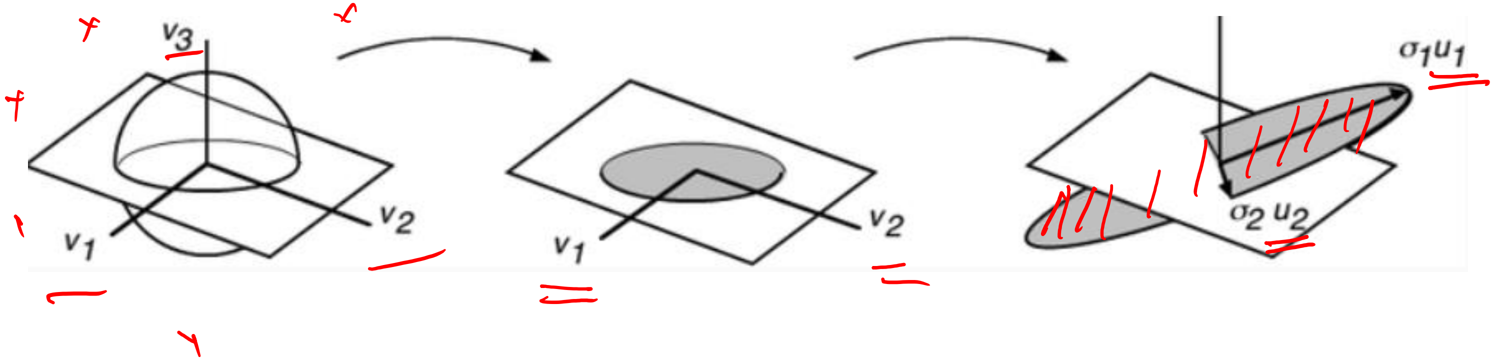


iii) Output Formation

SVD – Perspective ($m = n = 3, r = 2$)

$$Ax = \begin{array}{ccc} U & \Sigma & V^T \\ \begin{pmatrix} | & | & | \\ \boldsymbol{u_1} & \boldsymbol{u_2} & \boldsymbol{u_3} \\ | & | & | \end{pmatrix} & \begin{pmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & 0 \end{pmatrix} & \underbrace{\begin{pmatrix} — & v_1^T & — \\ — & v_2^T & — \\ — & v_3^T & — \end{pmatrix}} \end{array} x = y$$

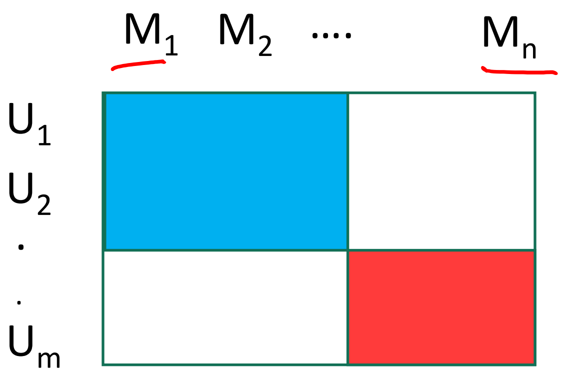
Graphical Representation of basis transformation:



SVD- Applications : Latent Concept Discovery

User- Movie Matrix : Toy Example

$$A_{m\times n} =$$



Matrix Entries: Star Rating from the users

Objective: Discover the latent "concepts" in Movies from the User Ratings

$$\begin{array}{c|c}
 & \text{Matrix} \quad \text{Alien} \quad \text{Serenity} \quad \text{Casablanca} \quad \text{Amelie} \\
\begin{matrix} U_1 \\ U_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ U_m \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}
\end{array}$$

10×5 A

'latent' concepts

SciFi

Romance

Assumption: There exists two type of Hidden Concepts in Movies

$$A = \overbrace{\begin{bmatrix} \uparrow & \uparrow \\ u_1 & u_2 \\ \downarrow & \downarrow \end{bmatrix}}^{U}_{10\times 2} \underbrace{\begin{bmatrix} & \\ & \end{bmatrix}}_{2\times 2} \underbrace{\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}}_{(2\times 5)}$$

SVD- Applications : Latent Concept Discovery

$\boldsymbol{U}$: "user to concept" similarity matrix

$\boldsymbol{V^T}$: "concept to movie" similarity matrix

$\boldsymbol{\Sigma}$: Concept Strength

SciFi-concept

Romance-concept

$$
\begin{array}{c|c}
 & \text{Matrix} \quad \text{Alien} \quad \text{Serenity} \quad \text{Casablanca} \quad \text{Amelie} \\
\begin{array}{c} U_1 \\ U_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ U_m \end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & 0 & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & 0 & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{array}{c}
 \\
\begin{array}{c} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{array}
\end{array}
\overset{U}{\underset{\quad u_1 \qquad u_2.}{
\begin{bmatrix}
\mathbf{0.14} & 0.00 \\
\mathbf{0.42} & 0.00 \\
\mathbf{0.56} & 0.00 \\
\mathbf{0.70} & 0.00 \\
0.00 & \mathbf{0.60} \\
0.00 & \mathbf{0.75} \\
0.00 & \mathbf{0.30}
\end{bmatrix}}}
\times
\overset{\Sigma}{\underset{2 \times 2}{
\begin{bmatrix}
\overset{\sigma_1}{12.4} & 0 \\
0 & \overset{\sigma_2}{9.5}
\end{bmatrix}}}
\times
\begin{array}{c} v_1^T \\ v_2^T \end{array}
\overset{V^T}{\underset{2 \times 5}{
\begin{bmatrix}
\mathbf{0.58} & \mathbf{0.58} & \mathbf{0.58} & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & \mathbf{0.71} & \mathbf{0.71}
\end{bmatrix}}}
$$

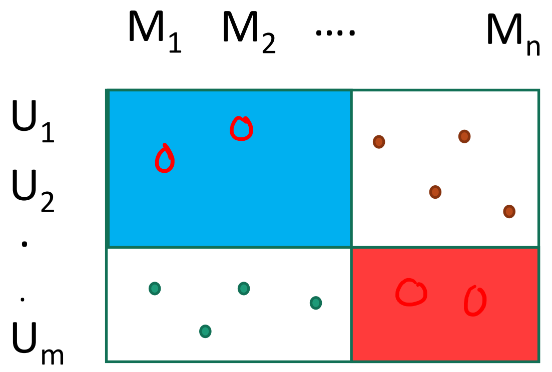
$m = 7$

7×2

Matrix Reconstruction with Noisy Data

User –Movie Matrix with Noisy Data:

$$A_{m \times n} =$$



$$
\begin{array}{cc}
 & \begin{matrix}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{matrix} \\
\begin{matrix} U_1 \\ U_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ U_m \end{matrix} &
\underset{7 \times 5}{\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}}
\end{array}
=
\begin{array}{c}
\begin{bmatrix}
\mathbf{0.13} & -0.02 & -0.01 \\
\mathbf{0.41} & -0.07 & -0.03 \\
\mathbf{0.55} & -0.09 & -0.04 \\
\mathbf{0.68} & -0.11 & -0.05 \\
0.15 & \mathbf{0.59} & \mathbf{0.65} \\
0.07 & \mathbf{0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{0.29} & \mathbf{0.32}
\end{bmatrix} \\
\begin{matrix} u_1 & u_2 & u_3 \end{matrix}
\end{array}
\times
\begin{bmatrix}
\overset{\sigma_1}{\mathbf{12.4}} & 0 & 0 \\
0 & \overset{\sigma_2}{\mathbf{9.5}} & 0 \\
0 & 0 & \overset{\sigma_3}{\mathbf{1.3}}
\end{bmatrix}
\times
\begin{array}{cc}
\begin{matrix} v_1^T \\ v_2^T \\ v_3^T \end{matrix} &
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
-0.12 & 0.02 & -0.12 & \mathbf{0.69} & \mathbf{0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
\end{array}
$$

Low singular value implies the 'concept' is not important

Rank – k Approximation

$$\boldsymbol{A}= \quad \sigma_1 . u_1\, v_1^T + \sigma_2 . u_2\, v_2^T + \sigma_3 . u_3\, v_3^T \implies \text{Rank – 3 Matrix}$$

$$\hat{A} = \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} = \begin{bmatrix} \mathbf{0.13} & -0.02 & \cancel{-0.01} \\ \mathbf{0.41} & -0.07 & \cancel{-0.03} \\ \mathbf{0.55} & -0.09 & \cancel{-0.04} \\ \mathbf{0.68} & -0.11 & \cancel{-0.05} \\ 0.15 & \mathbf{0.59} & \cancel{\mathbf{0.65}} \\ 0.07 & \mathbf{0.73} & \cancel{\mathbf{-0.67}} \\ 0.07 & \mathbf{0.29} & \cancel{\mathbf{0.32}} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \cancel{\mathbf{1.3}} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ -0.12 & 0.02 & -0.12 & \mathbf{0.69} & \mathbf{0.69} \\ \cancel{0.40} & \cancel{\mathbf{-0.80}} & \cancel{0.40} & \cancel{0.09} & \cancel{0.09} \end{bmatrix}$$

Rank – k Approximation

$\sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T$

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} \approx \begin{array}{c} \\ \\ \\ \\ u_5 \\ \\ u_7 \end{array} \overset{A_k}{\begin{bmatrix} 0.92 & 0.95 & 0.92 & 0.01 & 0.01 \\ 2.91 & 3.01 & 2.91 & -0.01 & -0.01 \\ 3.90 & 4.04 & 3.90 & 0.01 & 0.01 \\ 4.82 & 5.00 & 4.82 & 0.03 & 0.03 \\ 0.70 & 0.53 & 0.70 & 4.11 & 4.11 \\ -0.69 & 1.34 & -0.69 & 4.78 & 4.78 \\ 0.32 & 0.23 & 0.32 & 2.01 & 2.01 \end{bmatrix}}$$

Reconstructed Low –Rank Matrix, A_k :

- $\sigma_1 . u_1 \, v_1^T + \sigma_2 . u_2 \, v_2^T$ ==> Rank – 2 Matrix
- Reduces Noise components
- Fills in zero entries

Applications : Movie Recommender Systems

Data: Users rating movies represented as ‘user-movie’ Matrix
Sparse and often noisy

Assumptions:

- There are k basic ~~user~~ movie profiles, and each user is a linear combination of these profiles
  E.g., action, comedy, drama, romance, actor specific
- Each user is a weighted combination of these profiles. The “true” matrix has rank k

If we had the matrix A with all ratings of all users for all movies, the matrix A_k would tell us the true preferences of the users for the movies.

Observed Matrix $\tilde{A} = A_k +$ Noise

What we observe is a noisy, and incomplete version , $\tilde{A}$

Problem Statement:

Given the incomplete matrix $\tilde{A}$, get the missing ratings that A_k would produce

Applications : Movie Recommender Systems

Algorithm:

- Compute the rank-k approximation $\tilde{A}_k$ of and matrix $\tilde{A}$
- Predict for user u and movie m, the value $\tilde{A}_k[m, u]$.

$$
\tilde{A} = \begin{array}{c} \\ u_2 \\ u_3 \\ \\ u_5 \\ \\ u_7 \end{array}
\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{0} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{0} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix}
=
\begin{array}{c}
\begin{bmatrix} \mathbf{0.14} & -0.06 & -0.04 \\ \mathbf{0.30} & -0.11 & \mathbf{-0.61} \\ \mathbf{0.43} & -0.16 & \mathbf{0.76} \\ \mathbf{0.74} & -0.31 & -0.18 \\ 0.15 & \mathbf{0.53} & 0.02 \\ 0.07 & \mathbf{0.70} & -0.03 \\ 0.07 & \mathbf{0.27} & 0.01 \end{bmatrix} \\
\begin{array}{ccc} u_1 & u_2 & u_3 \end{array}
\end{array}
\times
\begin{array}{ccc} \sigma_1 & \sigma_2 & \sigma_3 \end{array}
\begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \mathbf{1.3} \end{bmatrix}
\times
\begin{array}{c} v_1^T \\ v_2^T \\ v_3^T \end{array}
\begin{bmatrix} \mathbf{0.51} & \mathbf{0.66} & \mathbf{0.44} & 0.23 & 0.23 \\ \mathbf{-0.24} & -0.13 & -0.21 & \mathbf{0.66} & \mathbf{0.66} \\ 0.59 & 0.08 & \mathbf{-0.80} & 0.01 & 0.01 \end{bmatrix}
$$

Applications : Movie Recommender Systems

Collaborative Filtering

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{0} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{0} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} = \begin{bmatrix} \mathbf{0.14} & -0.06 & \cancel{-0.04} \\ \mathbf{0.30} & -0.11 & \cancel{\mathbf{-0.61}} \\ \mathbf{0.43} & -0.16 & \cancel{\mathbf{0.76}} \\ \mathbf{0.74} & -0.31 & \cancel{-0.18} \\ 0.15 & \mathbf{0.53} & \cancel{0.02} \\ 0.07 & \mathbf{0.70} & \cancel{-0.03} \\ 0.07 & \mathbf{0.27} & \cancel{0.01} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 & 0 \\ 0 & \mathbf{9.5} & 0 \\ 0 & 0 & \cancel{\mathbf{1.3}} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.51} & \mathbf{0.66} & \mathbf{0.44} & 0.23 & 0.23 \\ \mathbf{-0.24} & -0.13 & -0.21 & \mathbf{0.66} & \mathbf{0.66} \\ \cancel{0.59} & \cancel{0.08} & \cancel{\mathbf{-0.80}} & \cancel{0.01} & \cancel{0.01} \end{bmatrix}$$

$\tilde{A}$ $\qquad$ U $\qquad$ Σ $\qquad$ V^T

$\tilde{A} = A_k + \text{noise}.$

$$\tilde{A}_k = \begin{array}{c} u_1 \\ u_2 \\ \\ \\ \\ \\ \\ \end{array} \begin{bmatrix} \mathbf{0.96} & \mathbf{1.14} & \mathbf{0.82} & -0.01 & -0.01 \\ \mathbf{1.94} & \mathbf{2.32} & \mathbf{1.66} & 0.07 & 0.07 \\ \mathbf{2.77} & \mathbf{3.32} & \mathbf{2.37} & 0.08 & 0.08 \\ \mathbf{4.84} & \mathbf{5.74} & \mathbf{4.14} & -0.08 & 0.08 \\ 0.40 & \mathbf{1.42} & 0.33 & \mathbf{4.06} & \mathbf{4.06} \\ -0.42 & 0.63 & -0.38 & \mathbf{4.92} & \mathbf{4.92} \\ 0.20 & \mathbf{0.71} & 0.16 & \mathbf{2.03} & \mathbf{2.03} \end{bmatrix}$$

- Filled in non-entries with approximate values
- Reduced the value of noisy entries