Low-Rank Matrix Approximations, SVD and Connections to PCA

Archer Yang

McGill University

October 4, 2024

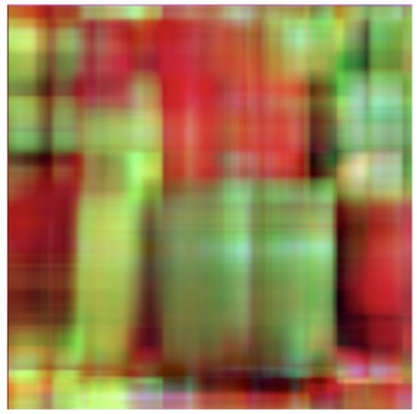
- **Reading assignment:** read chapter 4.5-4.6 of Mathematics for Machine Learning book.

Low-Rank Matrix

Applications

- Data compression
- Feature learning

Data compression

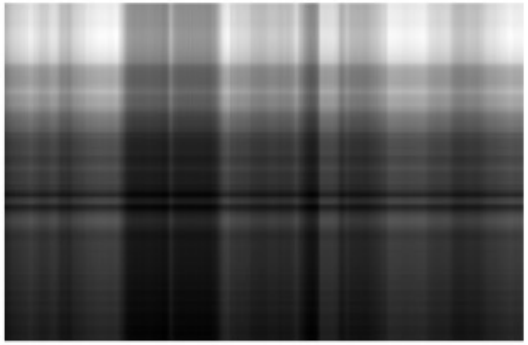



The size of the four compressed images is 45Kb, 345Kb, 1.1Mb and 3.4Mb[1].

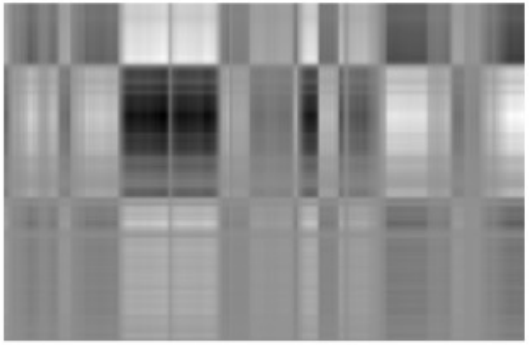
[1] *Multivariate Statistics*, Richard Wilkinson

Feature representation learning

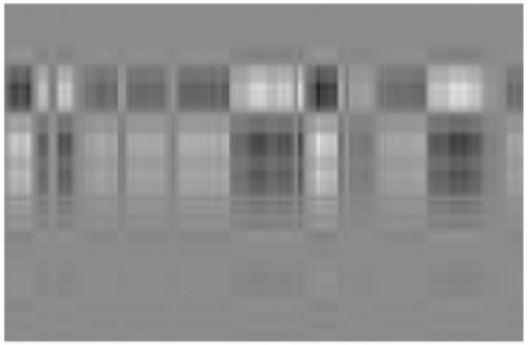
(a) Original image $\boldsymbol{A}$.



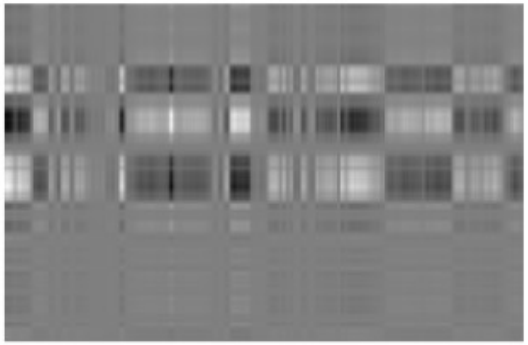
(b) $\boldsymbol{A}_1,\ \sigma_1 \approx 228,052.$



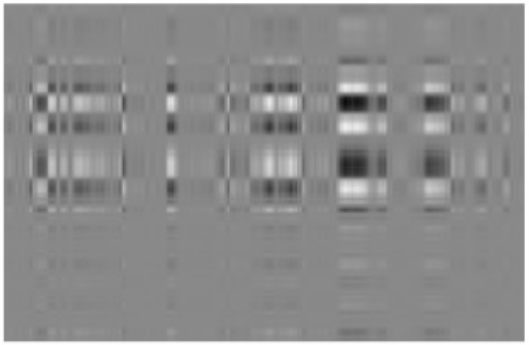
(c) $\boldsymbol{A}_2,\ \sigma_2 \approx 40,647.$



(d) $\boldsymbol{A}_3,\ \sigma_3 \approx 26,125.$



(e) $\boldsymbol{A}_4,\ \sigma_4 \approx 20,232.$



(f) $\boldsymbol{A}_5,\ \sigma_5 \approx 15,436.$

- Let's review the concept of matrix rank.

Rank-0 matrix There is only one rank-zero matrix of a given size, namely the all-zero matrix.

Rank-1 matrix all rows are multiples of each other. In the example in (4), all columns are also multiples of each other; this is not an accident.

An equivalent definition of a rank-1 $m \times n$ matrix is as the outer product $\boldsymbol{u}\boldsymbol{v}^\top$ of an m-vector $\boldsymbol{u}$ and an n-vector $\boldsymbol{v}$.

$$\boldsymbol{A} = \boldsymbol{u}\boldsymbol{v}^\top = \begin{pmatrix} u_1\boldsymbol{v}^\top \\ u_2\boldsymbol{v}^\top \\ \vdots \\ u_m\boldsymbol{v}^\top \end{pmatrix} = \begin{pmatrix} v_1\boldsymbol{u} & v_2\boldsymbol{u} & \cdots & v_n\boldsymbol{u} \end{pmatrix}$$

Note that each row is a multiple of $\boldsymbol{v}^\top$, and each column is multiple of $\boldsymbol{u}$.

Rank-2 matrix A rank-two matrix is just a sum of two rank-1 matrices

$$\boldsymbol{A} = \boldsymbol{u}_1\boldsymbol{v}_1^\top + \boldsymbol{u}_2\boldsymbol{v}_2^\top = \begin{pmatrix} u_{11}\boldsymbol{v}_1^\top + u_{21}\boldsymbol{v}_2^\top \\ u_{12}\boldsymbol{v}_1^\top + u_{22}\boldsymbol{v}_2^\top \\ \vdots \\ u_{1m}\boldsymbol{v}_1^\top + u_{2m}\boldsymbol{v}_2^\top \end{pmatrix}$$

$$= \begin{pmatrix} | & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 \\ | & | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_1^\top & - \\ - & \boldsymbol{v}_2^\top & - \end{pmatrix} = \begin{pmatrix} | & | \\ \boldsymbol{u}_1 & \boldsymbol{u}_2 \\ | & | \end{pmatrix} \begin{pmatrix} | & | \\ \boldsymbol{v}_1 & \boldsymbol{v}_2 \\ | & | \end{pmatrix}^\top$$

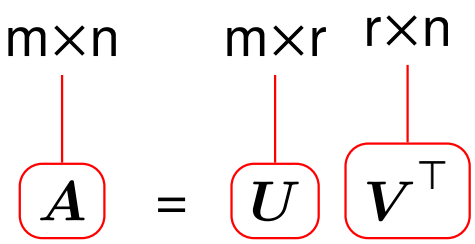
Note: a rank-2 matrix is one that can be written as the sum of two rank-1 matrices and is not itself a rank-0 or rank-1 matrix.

Rank-r matrix A rank-r matrix can be written as the sum of r rank-1 matrices, and **cannot** be written as the sum of $r-1$ or fewer rank-1 matrices.

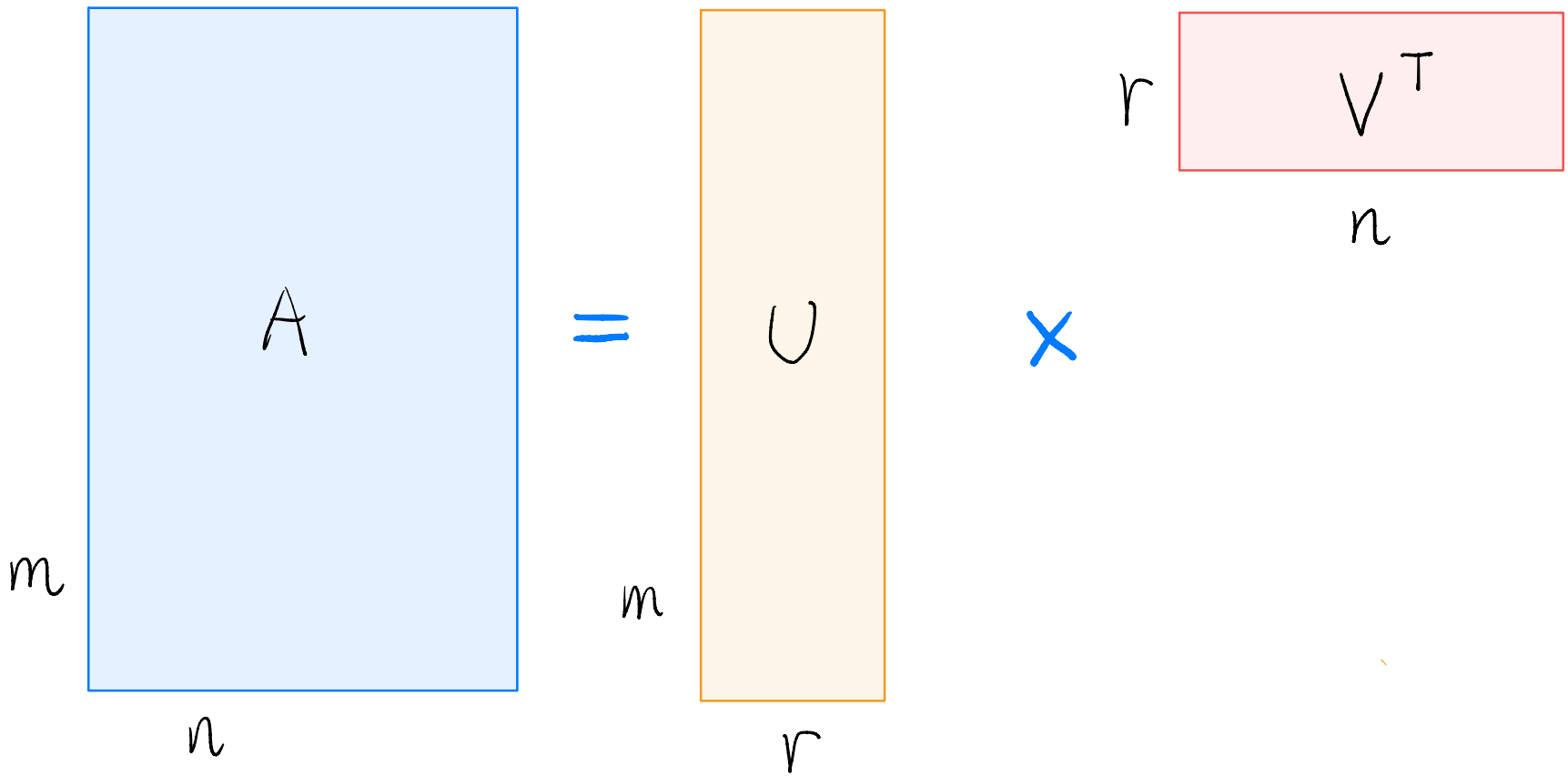
$$\boldsymbol{A} = \boldsymbol{u}_1\boldsymbol{v}_1^\top + \cdots + \boldsymbol{u}_r\boldsymbol{v}_r^\top = \begin{pmatrix} u_{11}\boldsymbol{v}_1^\top + \cdots + u_{r1}\boldsymbol{v}_r^\top \\ u_{12}\boldsymbol{v}_1^\top + \cdots + u_{r2}\boldsymbol{v}_r^\top \\ \vdots \\ u_{1m}\boldsymbol{v}_1^\top + \cdots + u_{rm}\boldsymbol{v}_r^\top \end{pmatrix}$$

$$= \begin{pmatrix} | & & | \\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_r \\ | & & | \end{pmatrix} \begin{pmatrix} - & \boldsymbol{v}_1^\top & - \\ & \vdots & \\ - & \boldsymbol{v}_r^\top & - \end{pmatrix} = \underbrace{\begin{pmatrix} | & & | \\ \boldsymbol{u}_1 & \cdots & \boldsymbol{u}_r \\ | & & | \end{pmatrix}}_{\boldsymbol{U}} \underbrace{\begin{pmatrix} | & & | \\ \boldsymbol{v}_1 & \cdots & \boldsymbol{v}_r \\ | & & | \end{pmatrix}^\top}_{\boldsymbol{V}^\top}$$

- Rank-r matrix $\boldsymbol{A}$ can be factored into the product of a "tall and skinny" matrix $\boldsymbol{U}$ and a "short and fat" matrix $\boldsymbol{V}^\top$



- Equivalent definitions:
  - The largest linearly independent subset of columns of $\boldsymbol{A}$ has size r. i.e. , all n columns of $\boldsymbol{A}$ arise as linear combinations of only r of them.
  - The largest linearly independent subset of rows of $\boldsymbol{A}$ has size r.


A
=
U
X
r
V^T
n
m
m
n
r

Low-Rank Matrix Approximations: Motivation

- **Low-rank approximation:** To approximate a rank-r matrix $\boldsymbol{A}$ with a rank-k matrix $\boldsymbol{A}(k)$ ($k \leq r$)

$$\boldsymbol{A} \approx \boldsymbol{A}(k)$$

or

$$\min_{\boldsymbol{A}(k)} ||\boldsymbol{A} - \boldsymbol{A}(k)||$$

- **Applications:**
  - **Compression**: $m \times n$ numbers in $\boldsymbol{A}$; $k(m + n)$ numbers in $\boldsymbol{A}(k)$
  - **De-noising**

$$\boldsymbol{A} = \boldsymbol{A}(k) + \text{error}$$

Singular Value Decomposition

Singular Value Decomposition (SVD)

- To solve low-rank matrix approximation problem, we apply **the singular value decomposition** (SVD), a central matrix decomposition method in linear algebra.
- It applied to **all matrices**, not only to square matrices.

Singular Value Decomposition (SVD)

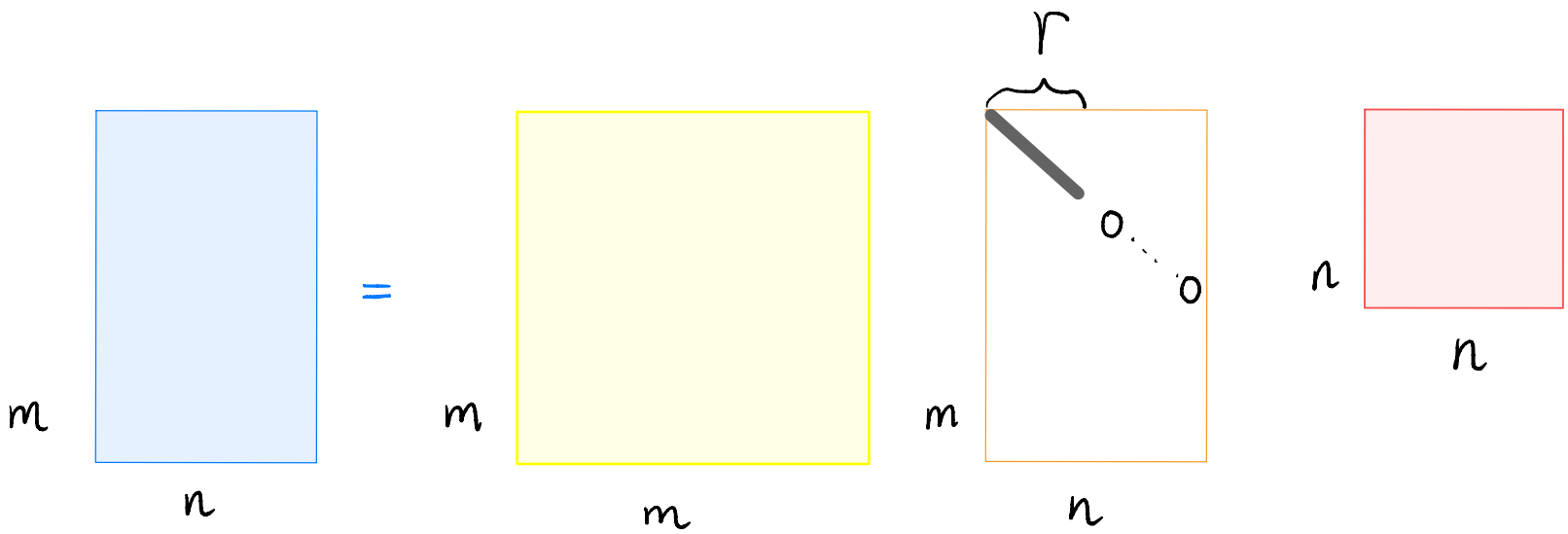
Theorem 4.22 MML (SVD Theorem)

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be a rectangular matrix. The SVD of $\boldsymbol{A}$ is a decomposition of the form

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$$

- $\boldsymbol{U} = (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_m)$ is an $m \times m$ orthogonal matrix, $\boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_{m \times m}$.
  Columns of $\boldsymbol{U}$ are **left singular vectors**.
- $\boldsymbol{V} = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n)$ is an $n \times n$ orthogonal matrix, $\boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}_{n \times n}$.
  Columns of $\boldsymbol{V}$ are **right singular vectors**.
- $\boldsymbol{\Sigma}$ is an $m \times n$ diagonal matrix with (ordered) nonnegative entries.
  Diagonal entries $\sigma_1 \geq \cdots \geq \sigma_n \geq 0$ are **singular values**.

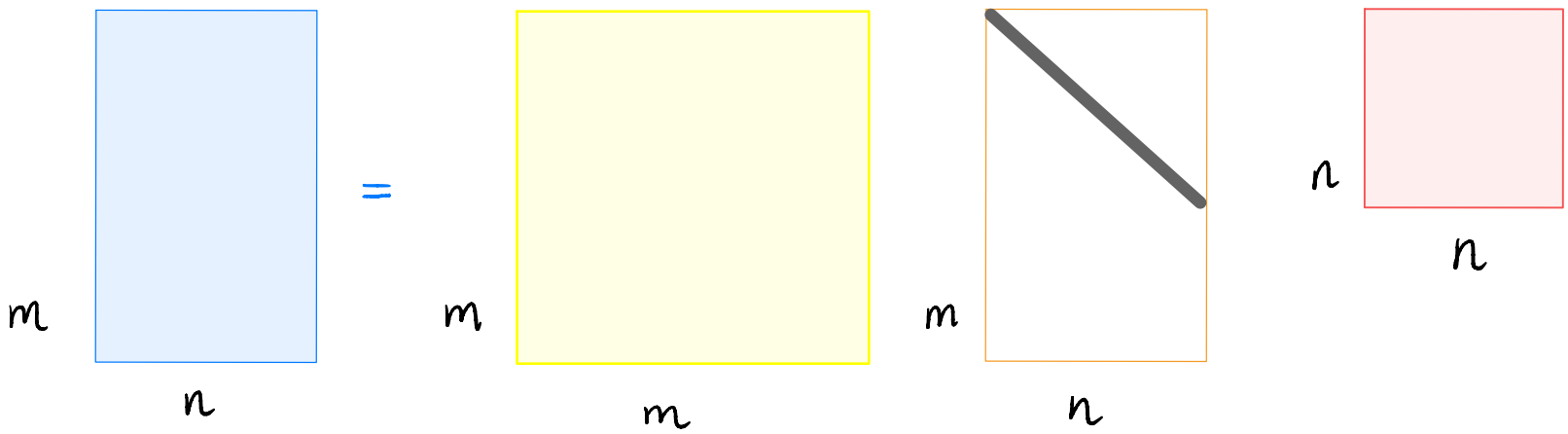
SVD (rank-r)



$$A = U \Sigma V^T$$

$m \times n$ $\quad$ $m \times m$ $\quad$ $m \times n$ $\quad$ $n \times n$

SVD (full column rank rank$(A) = n$)



$$A = U \Sigma V^T$$

$m \times n$ $\quad$ $m \times m$ $\quad$ $m \times n$ $\quad$ $n \times n$

SVD vs Eigen-decomposition

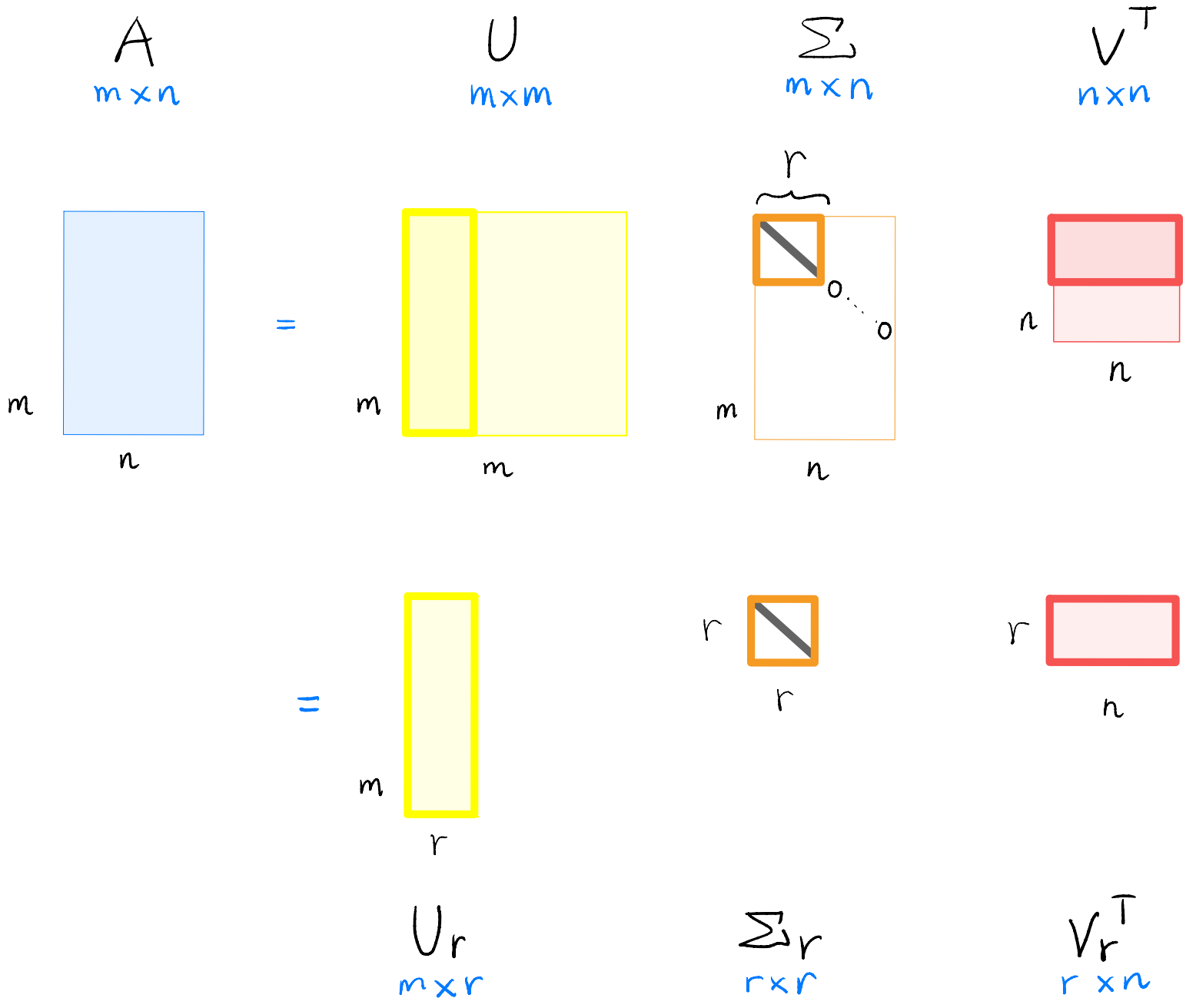
In contrast to the eigen-decomposition

$$\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$$

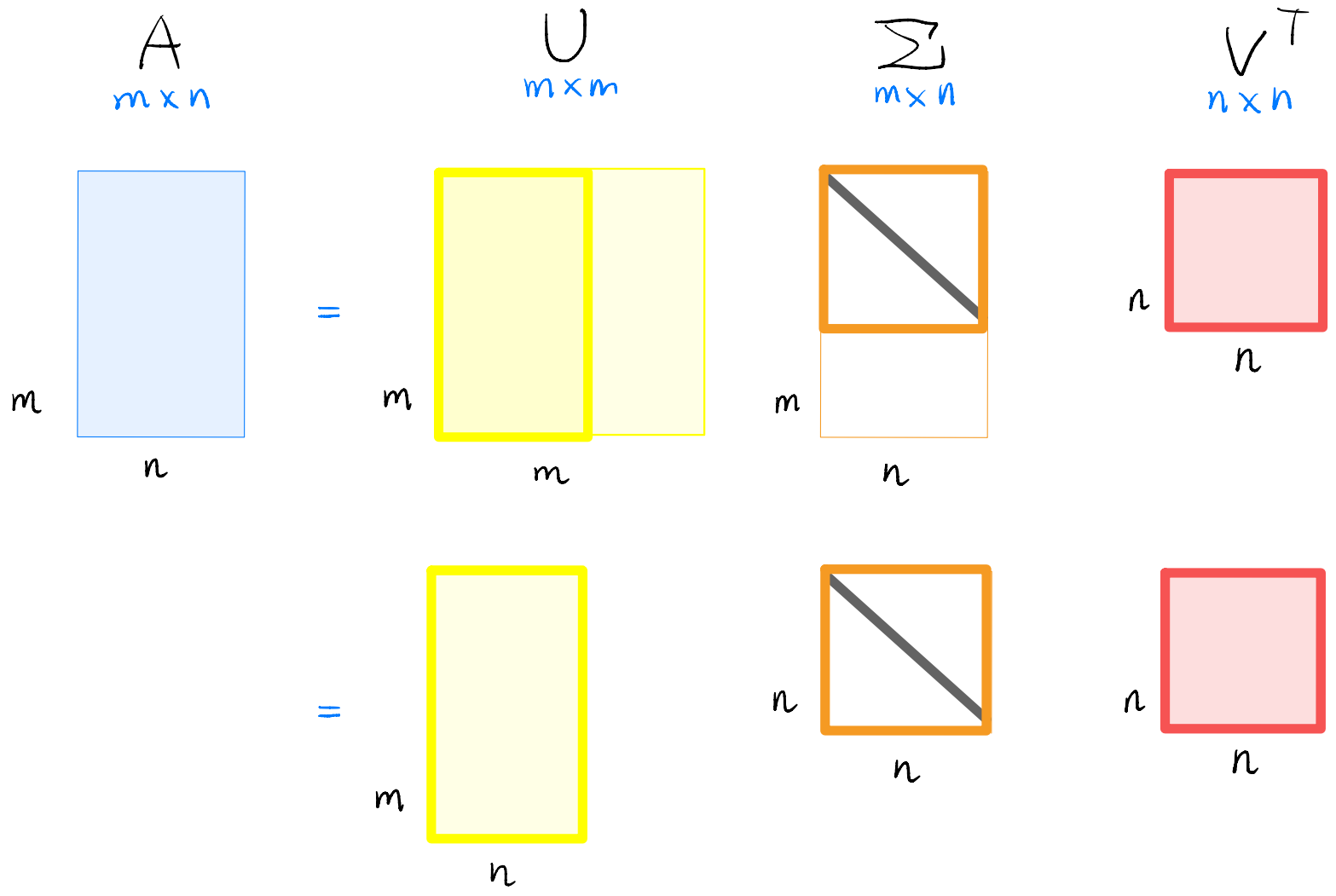
in SVD:

- Orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ are not the same, $\boldsymbol{U}$ and $\boldsymbol{V}$ need not even have the same dimension.
- $\boldsymbol{A}$ need not be square.

Reduced SVD (rank$(A) = r \leq n$)



Reduced SVD (full column rank rank$(A) = n$)



Reduced SVD for Rank-r matrix

If rank($\boldsymbol{A}$) = r, the SVD expresses $\boldsymbol{A}$ as a nonnegative linear combination of r rank-1 matrices,

$$\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \sum_{i=1}^{m}\sum_{j=1}^{n} \sigma_{ij}\boldsymbol{u}_i\boldsymbol{v}_j^\top \tag{1}$$

$$= \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i\boldsymbol{v}_i^\top = \sum_{i=1}^{r} \sigma_i \boldsymbol{A}_i$$

where σ_{ij} is the i-th row j-th column of $\boldsymbol{\Sigma}$. $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the corresponding left and right singular vectors.

Reduced SVD for Rank-r matrix

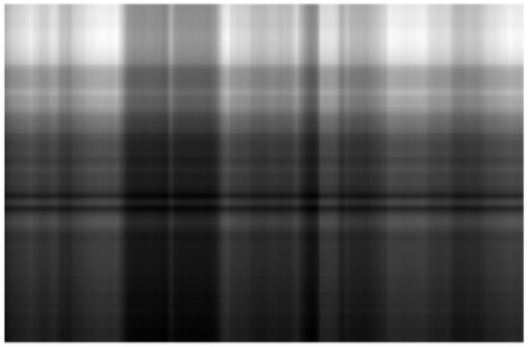
Eq. (1) can be further simplified

$$
\begin{aligned}
\boldsymbol{A} &= \sum_{i=j} \sigma_{ij} \boldsymbol{u}_i \boldsymbol{v}_j^\top + \sum_{i \neq j} \sigma_{ij} \boldsymbol{u}_i \boldsymbol{v}_j^\top \\
&= \sum_{i=j \leq r} \sigma_{ij} \boldsymbol{u}_i \boldsymbol{v}_j^\top + \underbrace{\sum_{i=j>r} \sigma_{ij} \boldsymbol{u}_i \boldsymbol{v}_j^\top}_{=0} + \underbrace{\sum_{i \neq j} \sigma_{ij} \boldsymbol{u}_i \boldsymbol{v}_j^\top}_{=0} \\
&= \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \\
&\equiv \sum_{i=1}^{r} \sigma_i \boldsymbol{A}_i
\end{aligned}
$$

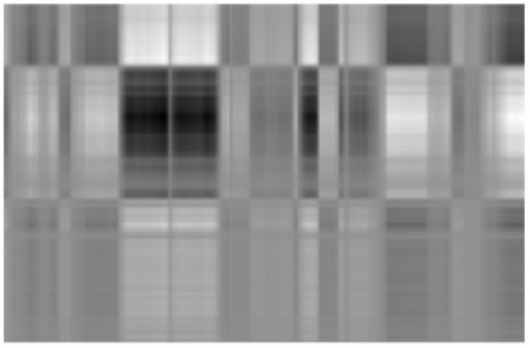
where $\sigma_i \equiv \sigma_{ii}$ and $\boldsymbol{A}_i = \boldsymbol{u}_i \boldsymbol{v}_i^\top$ is a rank-1 matrix.

Feature representation learning

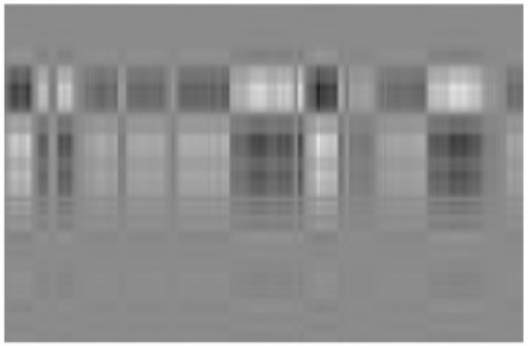
(a) Original image $\boldsymbol{A}$.



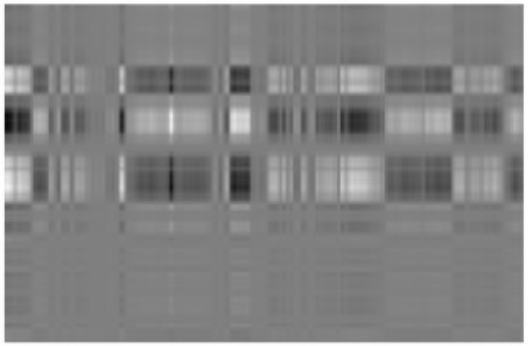
(b) $\boldsymbol{A}_1,\ \sigma_1 \approx 228,052.$



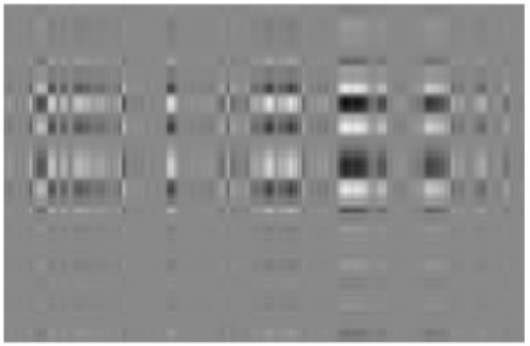
(c) $\boldsymbol{A}_2,\ \sigma_2 \approx 40,647.$



(d) $\boldsymbol{A}_3,\ \sigma_3 \approx 26,125.$



(e) $\boldsymbol{A}_4,\ \sigma_4 \approx 20,232.$



(f) $\boldsymbol{A}_5,\ \sigma_5 \approx 15,436.$

Question: why we observe vertical and horizontal lines in $\boldsymbol{A}_1, \cdots, \boldsymbol{A}_5$?

Reduced SVD for rank-r matrix

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ be a rank-r matrix, where $1 \le r \le \min(n, p)$. The reduced SVD of $\boldsymbol{A}$ is

$$\boldsymbol{A} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

- $\boldsymbol{U}_r = (\boldsymbol{u}_1, \cdots, \boldsymbol{u}_r)$ is an $m \times r$ orthogonal matrix, $\boldsymbol{U}_r^\top \boldsymbol{U}_r = \boldsymbol{I}_{r \times r}$.
- $\boldsymbol{V}_r = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_r)$ is an $n \times r$ orthogonal matrix, $\boldsymbol{V}_r^\top \boldsymbol{V}_r = \boldsymbol{I}_{r \times r}$.
- $\boldsymbol{\Sigma}_r$ is an $r \times r$ diagonal matrix with singular values $\sigma_1 \ge \cdots \ge \sigma_r > 0$ (all positive).

Low-rank Matrix Approximation

Rank-k approximation

- How do we approximate a rank-r matrix $\boldsymbol{A}$ by a rank-k matrix ($k \leq r$)?
- Recall that in reduced SVD, a rank-r matrix $\boldsymbol{A}$ can be expressed as

$$\boldsymbol{A} = \boldsymbol{U}_r \boldsymbol{\Sigma}_r \boldsymbol{V}_r^\top = \sum_{i=1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top \tag{2}$$

  where $\sigma_1 \geq \cdots \geq \sigma_r > 0$

- **Rank-k approximation of $\boldsymbol{A}$**: keep only the first k terms in (2)

$$\widehat{\boldsymbol{A}}(k) = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

  we see that $\text{rank}(\widehat{\boldsymbol{A}}(k)) = k$ and

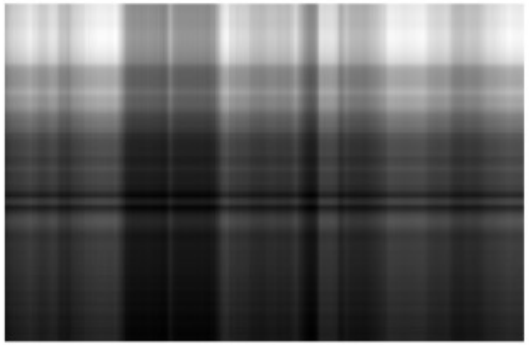
$$\boldsymbol{A} - \widehat{\boldsymbol{A}}(k) = \sum_{i=k+1}^{r} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

Algorithm

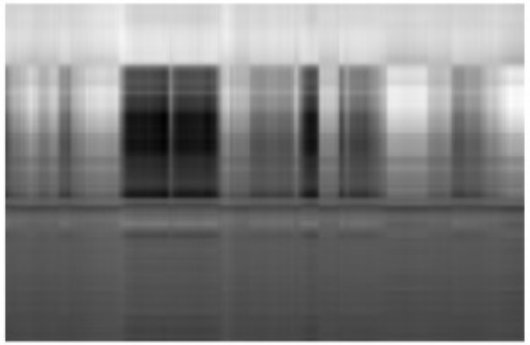
1. Compute the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\boldsymbol{\Sigma}$ is a nonnegative $m \times n$ diagonal matrix with diagonal entries sorted from high to low.
2. Keep only the **top k right singular vectors**: Set $\boldsymbol{V}_k^\top$ equal to the first k rows of $\boldsymbol{V}^\top$ (a $k \times n$ matrix)
3. Keep only the **top k left singular vectors**: Set $\boldsymbol{U}_k$ equal to the first k rows of $\boldsymbol{U}$ (a $m \times k$ matrix)
4. Keep only the **top k singular values**: Set $\boldsymbol{\Sigma}_k$ equal to the first k rows and columns of $\boldsymbol{\Sigma}$ (a $k \times k$ matrix)
5. The **rank-k approximation** is then

$$\widehat{\boldsymbol{A}}(k) = \boldsymbol{U}_k\boldsymbol{\Sigma}_k\boldsymbol{V}_k^\top = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

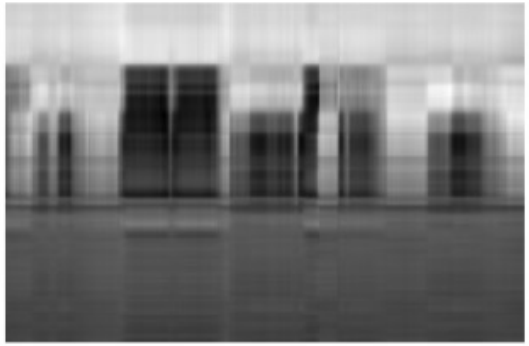
(a) Original image $\boldsymbol{A}$.



(b) Rank-1 approximation $\widehat{\boldsymbol{A}}(1)$.



(c) Rank-2 approximation $\widehat{\boldsymbol{A}}(2)$.



(d) Rank-3 approximation $\widehat{\boldsymbol{A}}(3)$.

(e) Rank-4 approximation $\widehat{\boldsymbol{A}}(4)$.

(f) Rank-5 approximation $\widehat{\boldsymbol{A}}(5)$.

The size of the four compressed images is 45Kb, 345Kb, 1.1Mb and 3.4Mb[2]. More demos are here. [3]

[2] *Multivariate Statistics*, Richard Wilkinson

[3] https://timbaumann.info/svd-image-compression-demo/

Quantifying approximation error

- How to quantify the approximation error $||\boldsymbol{A} - \widehat{\boldsymbol{A}}(k)||$?
- Need the notion of a matrix norm $||\cdot||$.
- e.g. **the Frobenious norm** or **the spectral norm**.

Frobenius norm

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. The Frobenius norm of $\boldsymbol{A}$ is

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\text{tr}(\boldsymbol{A}^\top \boldsymbol{A})}$$

- (Homework) Show that for any matrix $\boldsymbol{A}$ and orthogonal matrices $\boldsymbol{U}$ and $\boldsymbol{V}$:

$$\|\boldsymbol{U}^\top \boldsymbol{A}\|_F^2 = \|\boldsymbol{A}\|_F^2 \qquad \|\boldsymbol{A}\boldsymbol{V}\|_F^2 = \|\boldsymbol{A}\|_F^2 \tag{3}$$

- (Homework) Combining the result in (3) and using the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$ to show that

$$\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i^2}$$

where $r \equiv \text{rank}(\boldsymbol{A})$ and σ_i are the singular values of $\boldsymbol{A}$.

Spectral norm

Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. The spectral norm of $\boldsymbol{A}$ is defined as

$$\|\boldsymbol{A}\|_2 = \max_{\boldsymbol{x}} \frac{\|\boldsymbol{A}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2} = \max_{\boldsymbol{x}:\|\boldsymbol{x}\|=1} \|\boldsymbol{A}\boldsymbol{x}\|_2$$

Spectral norm

(Homework) Let $\boldsymbol{A} \in \mathbb{R}^{m \times n}$. Show that

$$\|\boldsymbol{A}\|_2 = \sigma_1$$

where σ_1 is the first (largest) singular value of $\boldsymbol{A}$.

Eckart-Young-Mirsky Theorem

For either the spectral norm $\|\cdot\|_2$ or the Frobenious norm $\|\cdot\|_F$, the rank-k approximation

$$\widehat{\boldsymbol{A}}(k) = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top = \sum_{i=1}^{k} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^\top$$

minimizes the approximation error

$$\|\boldsymbol{A} - \widehat{\boldsymbol{A}}(k)\| \leq \|\boldsymbol{A} - \boldsymbol{C}\| \qquad \text{for all rank-}k \text{ matrices } \boldsymbol{C}$$

In other words

$$\widehat{\boldsymbol{A}}(k) = \underset{\text{rank}(\boldsymbol{C})=k}{\arg\min} \|\boldsymbol{A} - \boldsymbol{C}\|$$

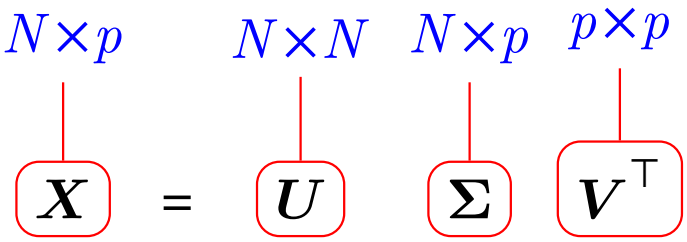
Moreover,

$$\|\boldsymbol{A} - \widehat{\boldsymbol{A}}(k)\| = \begin{cases} \sigma_{k+1} & \text{for the } \|\cdot\|_2 \text{ norm} \\ \sqrt{\sum_{i=k+1}^{r} \sigma_i^2} & \text{for the } \|\cdot\|_F \text{ norm} \end{cases}$$

Relationship with eigen-decomposition

Relation to eigenvalue decomposition

- Given $\boldsymbol{X}$ as a $N \times p$ matrix. If $\boldsymbol{X}$ has SVD



- Then it follows

$$\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top \boldsymbol{X} = \frac{1}{N}(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$$
$$= \frac{1}{N}\boldsymbol{V}\boldsymbol{\Sigma}^\top \boldsymbol{U}^\top \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top = \boldsymbol{V}\frac{\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}}{N}\boldsymbol{V}^\top$$

  where $\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \cdots, \sigma_r^2, 0, \cdots, 0) \in \mathbb{R}^{p\times p}$ with $r = \text{rank}(\boldsymbol{X})$.

- Meanwhile, by the eigen-decomposition

$$\boldsymbol{S} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$$

Relation to eigenvalue decomposition

- Therefore

$$\boldsymbol{V}\left(\frac{\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}}{N}\right)\boldsymbol{V}^\top = \boldsymbol{S} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^\top$$

- Right-singular vectors of $\boldsymbol{X}$ (the columns of $\boldsymbol{V}$) are eigenvectors of $\boldsymbol{S} = \frac{1}{N}\boldsymbol{X}^\top\boldsymbol{X}$.
- The non-zero eigenvalues of $\boldsymbol{S}$ are related to the non-zero singular values of $\boldsymbol{X}$ via

$$\boldsymbol{\Lambda} = \frac{\boldsymbol{\Sigma}^\top\boldsymbol{\Sigma}}{N} \qquad \text{i.e.} \qquad \lambda_d = \frac{\sigma_d^2}{N} \qquad \text{for } d = 1, \cdots, p$$

Relation to eigenvalue decomposition

- We also have

$$\begin{aligned}\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top &= \frac{1}{N}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top)^\top \\ &= \frac{1}{N}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top = \frac{1}{N}\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top\end{aligned}$$

- Therefore, left-singular vectors of $\boldsymbol{X}$ (the columns of $\boldsymbol{U}$) are eigenvectors of $\frac{1}{N}\boldsymbol{X}\boldsymbol{X}^\top$.

PCA as Low-rank Matrix Approximation

PCA as Low-rank Matrix Approximation

- PCA solves

$$\min_{\boldsymbol{B}} \frac{1}{N} \|\boldsymbol{X}^\top - \boldsymbol{B}\boldsymbol{B}^\top \boldsymbol{X}^\top\|_F^2 \equiv \frac{1}{N} \left\|\boldsymbol{X} - \boldsymbol{X}\boldsymbol{B}\boldsymbol{B}^\top\right\|_F^2$$

$$\text{subject to } \boldsymbol{B}^\top \boldsymbol{B} = \boldsymbol{I}_{k\times k}$$

- On the other hand, rank-k matrix approximation of $\boldsymbol{X}$ solves

$$\widehat{\boldsymbol{X}}(k) = \underset{\text{rank}(\boldsymbol{C})=k}{\arg\min} \|\boldsymbol{X} - \boldsymbol{C}\|_F \quad \text{where } \widehat{\boldsymbol{X}}(k) = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top$$

- One can show that $\boldsymbol{V}_k = \boldsymbol{B}$, $\boldsymbol{U}_k \boldsymbol{\Sigma}_k = \boldsymbol{X}\boldsymbol{B}$ and

$$\boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top = \boldsymbol{X}\boldsymbol{B}\boldsymbol{B}^\top$$

PCA as Low-rank Matrix Approximation

This relation suggest that SVD can also be used to produce PCA for $\boldsymbol{X}$ (k-dim. subspace), as follows

1. Perform rank-k matrix approximation of $\boldsymbol{X} \approx \widehat{\boldsymbol{X}}(k) = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top$.
2. Let $\boldsymbol{B}$ denote the top k **principal components** of $\boldsymbol{S} = \frac{1}{N} \boldsymbol{X}^\top \boldsymbol{X}$, set $\boldsymbol{B} = \boldsymbol{V}_k = (\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k)$.
3. (Homework) The **PC scores** can be obtained by $\boldsymbol{X}\boldsymbol{B} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k$.
   - e.g. for the n-th observation, the projected coordinates on the subspace spanned by $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_k$ are

$$(z_{n1}, \cdots, z_{nk}) = (\boldsymbol{v}_1^\top \boldsymbol{x}_n, \cdots, \boldsymbol{v}_k^\top \boldsymbol{x}_n) = \underbrace{\boldsymbol{x}_n^\top \boldsymbol{B}}_{\text{n-th row of } \boldsymbol{X}\boldsymbol{B}} = \underbrace{[\boldsymbol{U}_k \boldsymbol{\Sigma}_k]_{n\bullet}}_{\text{n-th row of } \boldsymbol{U}_k \boldsymbol{\Sigma}_k}$$

Computation complexity

Computing PCA for $\boldsymbol{X} \in \mathbb{R}^{N \times p}$ using SVD has less computation complexity then using eigen-decomposition

- **Eigen-decomposition approach:**
  - $\boldsymbol{S}$ covariance matrix computation $O(Np\min(N,p))$
  - Eigen-decomposition $O(p^3)$
  - Total: $O(Np\min(N,p))$ + $O(p^3)$
- **SVD approach:**
  - SVD $O(Np\min(N,p))$

Matrix completion via low rank approximation

What Are The Missing Entries?

- Consider the following matrix,

$$\boldsymbol{X} = \begin{bmatrix} 7 & ? & ? \\ ? & 8 & ? \\ ? & 12 & 6 \\ ? & ? & 2 \\ 21 & 6 & ? \end{bmatrix}$$

- **Matrix completion problem**: What are the missing entries?

- Consider an extreme assumption: that all rows are multiples of each other.

$$\boldsymbol{X} = \begin{bmatrix} 7 & ? & ? \\ ? & 8 & ? \\ ? & 12 & 6 \\ ? & ? & 2 \\ 21 & 6 & ? \end{bmatrix}$$

- Under such assumption, here is the completed matrix

$$\widehat{\boldsymbol{X}} = \begin{bmatrix} 7 & 2 & 1 \\ 28 & 8 & 4 \\ 42 & 12 & 6 \\ 14 & 4 & 2 \\ 21 & 6 & 3 \end{bmatrix} \tag{4}$$

- When you know something about the **"structure"** of a partially known matrix, then sometimes it's possible to recover all of the "**lost**" information.

Matrix completion using matrix approximation

- **Applications:**
  - **Matrix completion:**

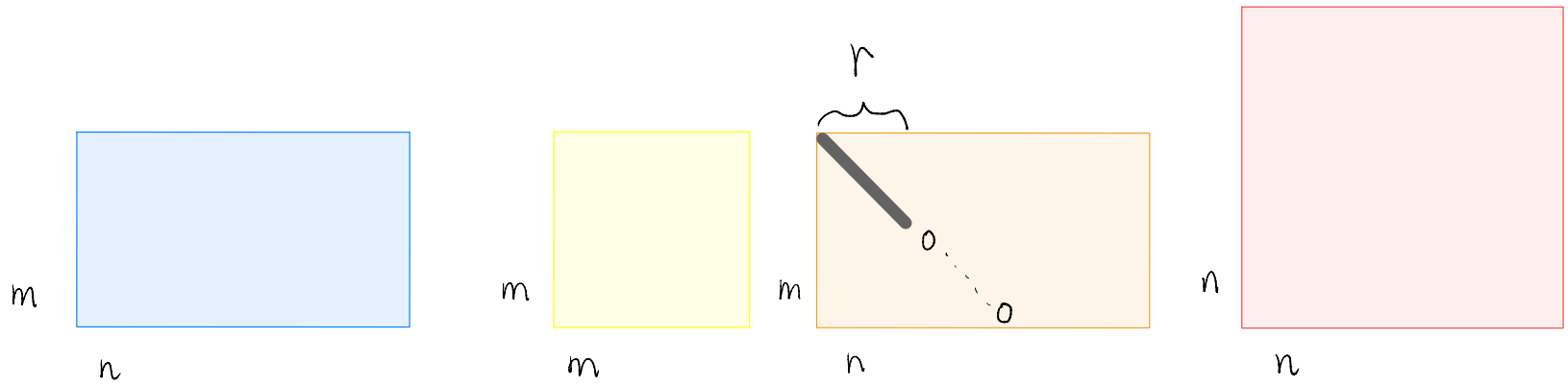
1. First impute $\boldsymbol{X}$ to obtain $\boldsymbol{X}'$. Fill the missing entries of $\boldsymbol{X}$ using one of the following methods
   - 0;
   - the average of the known entries in the same column;
   - the average of the known entries in the same row;
   - the average of the known entries of the matrix.
2. Compute the best rank-k approximation to $\boldsymbol{X}'$

$$\widehat{\boldsymbol{X}}(k) = \underset{\text{rank}(\boldsymbol{C})=k}{\arg\min} \|\boldsymbol{X}' - \boldsymbol{C}\|$$

- Why not we just do the first step to use $\boldsymbol{X}'$?

SVD for High-dimensional Data

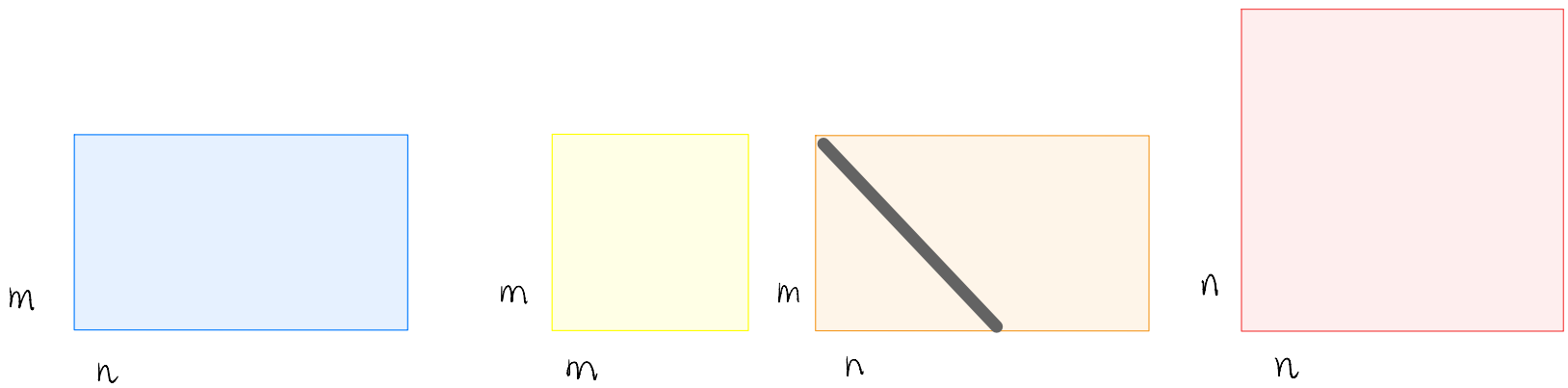
SVD ($m < n$, rank-r)



$$A = U \Sigma V^T$$

$m \times n \qquad m \times m \qquad m \times n \qquad n \times n$

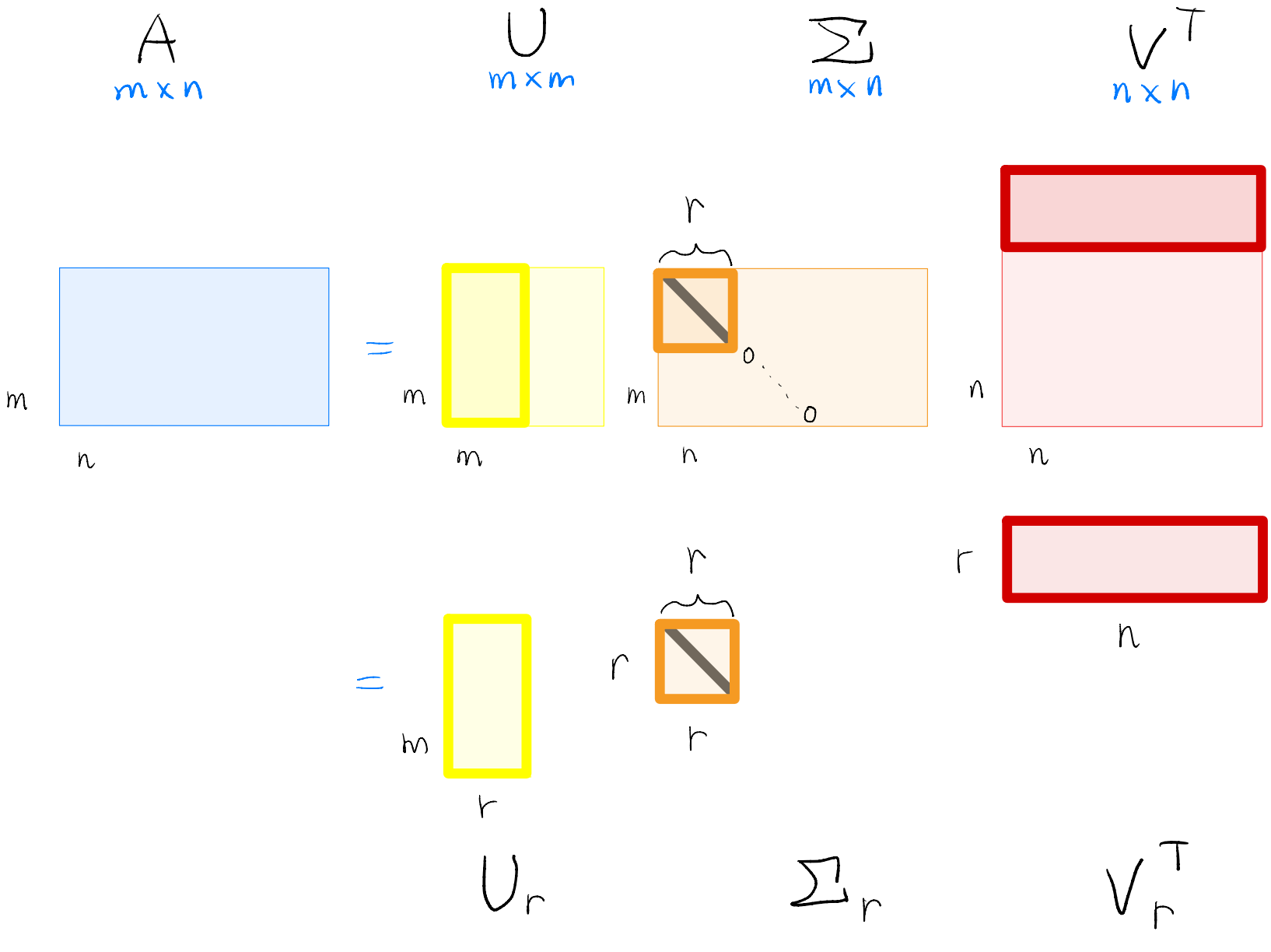
SVD ($m < n$, rank$(A) = m$)



$$A = U \Sigma V^T$$

$m \times n \qquad m \times m \qquad m \times n \qquad n \times n$

Reduced SVD ($m < n$, rank-r)



Reduced SVD ($m < n$, rank$(A) = m$)