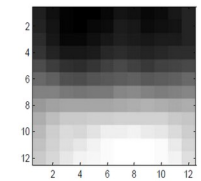
CMSC5741 Big Data Tech. & Apps.

Lecture 6: y Reduction

Prof. Michael R. Lyu
Computer Science & Engineering Dept.
The Chinese University of Hong Kong

A Compression Example



Outline

- Motivation
- SVD
- CUR
  - Application o
- PCA
  - Extension to robust PCA

Dimensionality Reduction Motivation

- **Assumption:** Data lie on or near a low *d*-dimensional subspace
- **Axes of this subspace are effective representation of the data**

Dimensionality Reduction Motivation

- **Compress / reduce dimensionality:**
  - 10^6 rows; 10^3 columns; no updates
  - Random acce **error: OK**

16 16

The above matrix is really “2-dimensional.” All rows can be reconstructed by scaling [1 1 1 0 0] or [0 0 0 1 1]

Rank of a Matrix

- **Q:** What is **rank** of a matrix **A**?
- **A:** No. of **linearly independent** rows/columns of **A**
- **For example:**

  – Matrix **A** =

    - **Why?** The first two rows are li[illegible]ndent, so the rank is at least 2, but all three rows are linearly dependent (the first is equal to the sum of the second and third) so the rank must be less than 3.

- **Why do we care about low rank?**
  – We can write **A** as two "basis" vectors: [1 2 1] [-2 -3 1]
  – And new coordinates of : [1 0] [0 1] [1 -1]

Rank is "Dimensionality"

- **Cloud of points 3D space:**
  - Think of point positions as a matrix:

A
B

1 row per point:

A

- **We can rewrite coordinates ntly!**
  - Old basis vectors: [1 0 0] [0 1 0] [0 0 1]
  - **New basis vectors: [1 2 1] [-2 -3 1]**
  - Then **A** has new coordinates: [1 0]. **B**: [0 1], **C**: [1 -1]
    - **Notice: We reduced the number of coordinates!**

Dimensionality Reduction

- **Goal of dimensionality reduction is to discover the axis of data!**

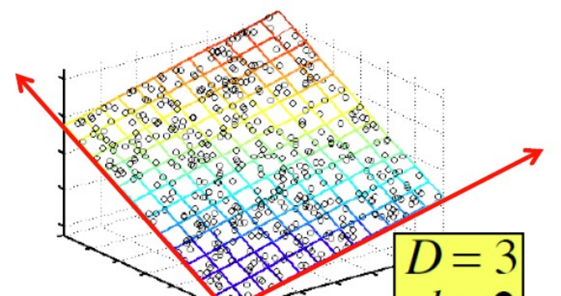
than representing
oint with 2 coordinates
ent each point with
ate (corresponding to
the position of the point on
the red line).

By doing this we incur a bit of **error** as the points do not exactly lie on the line

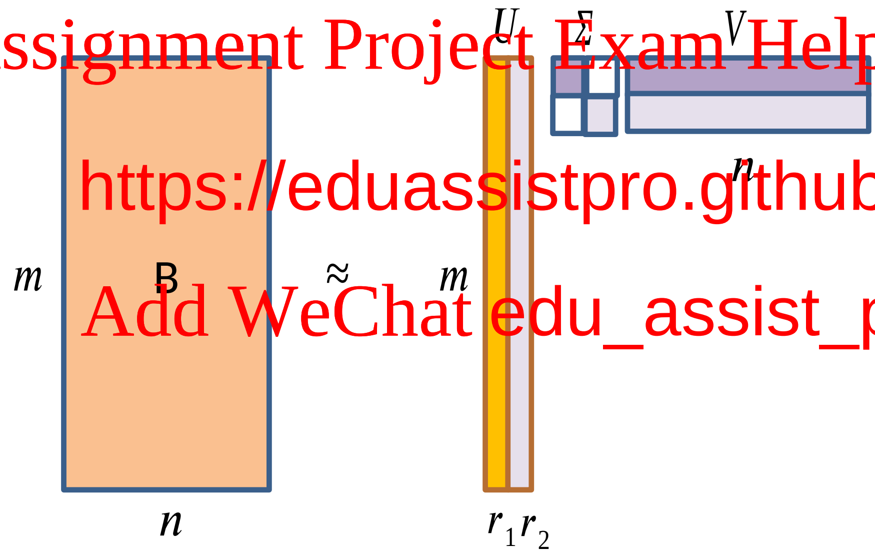
Why Reduce Dimensions?

Why reduce dimensions?

- **Discover hidden correlations/topics**
  - Words that o er
- **Remove redu tures**
  - Not all words are useful
- **Interpretation and visualization**
- **Easier storage and processing of the data**



SVD: Dimensionality Reduction

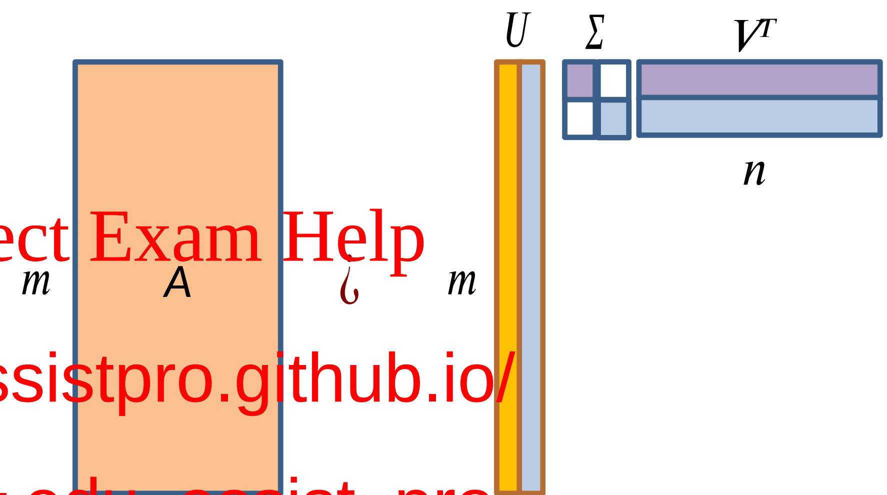


SVD: Singular Value Decomposition

- For an matrix A, we can decompose it as = ,
  where
  - *U* is an real or complex orthonormal matrix (i.e., = = I)
  - Σ is an m × n re with non-negative real n nd
  - V^T (the conjugate transpose of the transpose of *V* if *V* is real) is an real or complex orthonormal matrix.

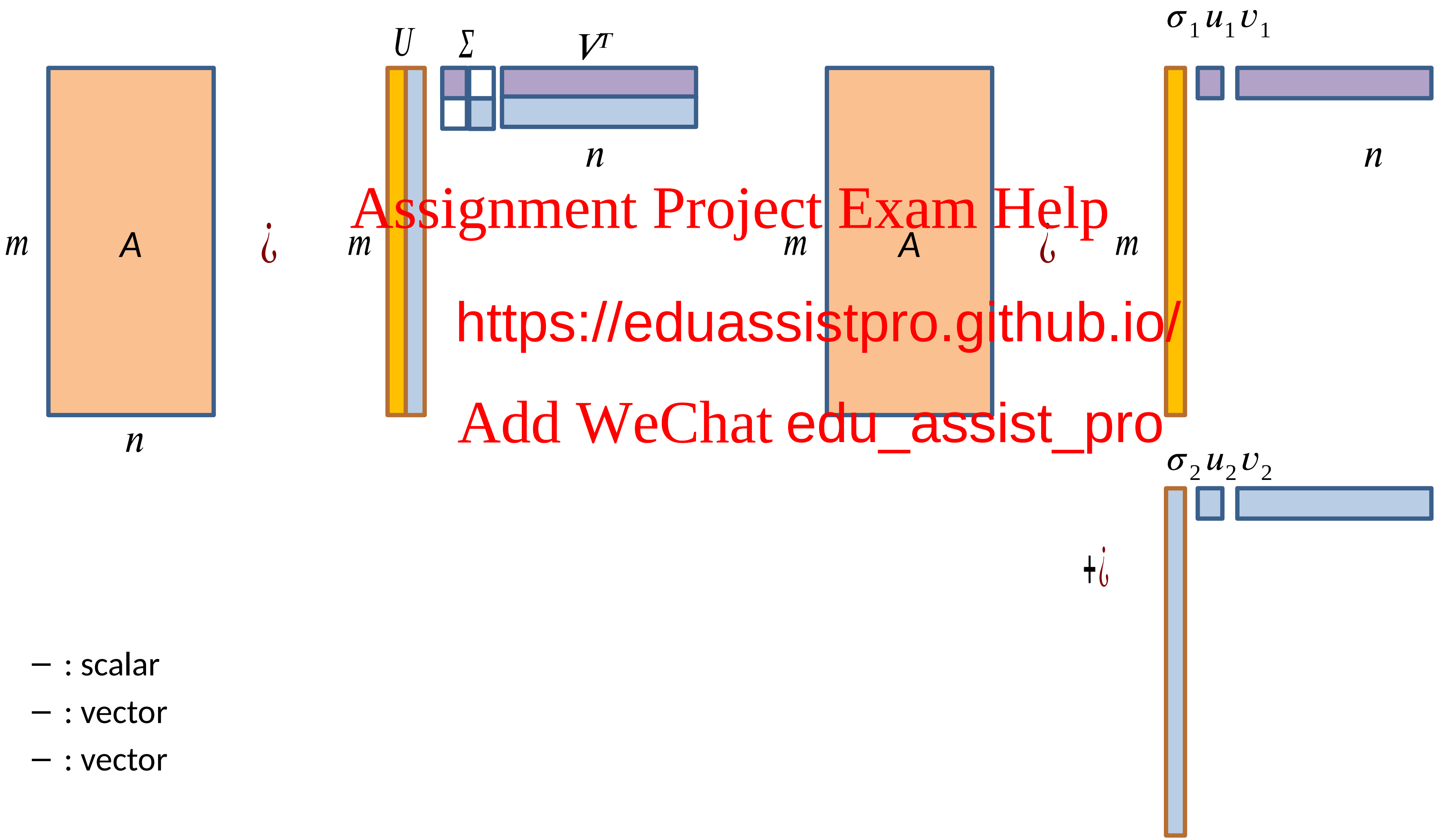
SVD: Singular Value Decomposition

- When rank(A) = r:
- : input data matrix
  - matrix (e.g., docume
- : left singular vectors
  - matrix (documents, topics)
- : singular values
  - diagonal matrix (strength of each "topic")
  - rank of matrix
- : right singular vectors
  - matrix (terms, topics)



- : scalar
- : vector
- : vector

SVD: Singular Value Decomposition



– : scalar
– : vector
– : vector

SVD Properties

- It is always possible to do SVD, i.e. decompose a matrix A into , where

- U, Σ, V: unique

- U,V: column orth
  - , (I: identity matrix)

- Σ: diagonal
  - Entries (singular values) are non-negative,
  - Sorted in decreasing order ($\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$).

SVD Example

- We give an example of a simple SVD decomposition

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 & 0 \end{bmatrix} = \qquad \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sqrt{0.2} & 0 & 0 & 0 & \sqrt{0.8} \\ 0 & 0 & 0 & 1 & 0 \\ -\sqrt{0.8} & 0 & 0 & 0 & \sqrt{0.2} \end{bmatrix}$$

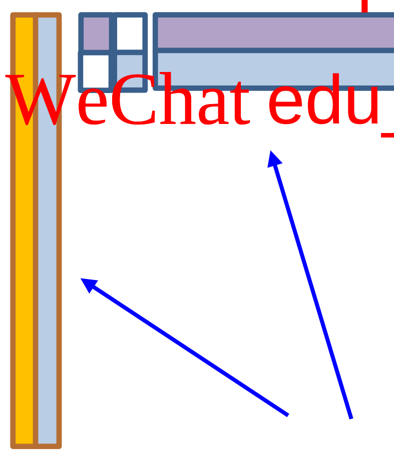
A $\qquad U$ $\qquad \Sigma$ $\qquad T$

SVD: Example – Users-to-Movies

- - example: Users to Movies

$$\begin{array}{lcc} & & \begin{array}{cccc} \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \end{array} \\ \begin{array}{l} \text{SciFi} \\ \\ \\ \text{Romance} \end{array} & m & \begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \\ & & n \end{array}$$

¿ m



“topics” or “concepts”

aka. Latent dimensions

aka. Latent factors

SVD: Example – Users-to-Movies

- - example

SciFi-concept

Romance-concept

$$
\begin{array}{cc}
 & \begin{matrix} \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \end{matrix} \\
\begin{matrix} \text{SciFi} \\ \\ \\ \\ \text{Romance} \\ \\ \\ \end{matrix} & \left[\begin{matrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{matrix}\right]
\end{array}
$$

m

n

m ¿

$$
\left[\begin{matrix} 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.5 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{matrix}\right]
\left[\begin{matrix} & 0 & 0 \\ & 7.1 & 0 \\ 0 & 0 & 2.5 \end{matrix}\right]
$$

x

$$
\left[\begin{matrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{matrix}\right]
$$

n

SVD: Example – Users-to-Movies

- - example

U is "user-to-concept" similarity matrix

SciFi-concept

Romance-concept

$$
\begin{array}{ll}
 & \begin{array}{cccc} \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \end{array} \\
\begin{array}{l} \text{SciFi} \\ \\ \\ m \\ \\ \text{Romance} \\ \\ \end{array} &
\begin{bmatrix}
1 & 3 & 1 & 0 \\
3 & 4 & 3 & 1 \\
4 & 2 & 4 & 0 \\
5 & 4 & 5 & 1 \\
0 & 1 & 0 & 4 \\
0 & 0 & 0 & 3 \\
0 & 1 & 0 & 5
\end{bmatrix} \\
 & n
\end{array}
\;\; \overset{m}{¿} \;\;
\begin{bmatrix}
[illegible] & & \\
0.48 & -0.02 & 0.43 \\
0.49 & -0.08 & -0.5 \\
0.68 & -0.07 & -0.16 \\
0.06 & 0.57 & 0.06 \\
0.01 & 0.41 & -0.20 \\
0.07 & 0.70 & -0.01
\end{bmatrix}
\begin{bmatrix}
[illegible] & 0 & 0 \\
 & 7.1 & 0 \\
0 & 0 & 2.5
\end{bmatrix}
\times
\underset{n}{\begin{bmatrix}
0.59 & 0.54 & 0.59 & 0.05 \\
-0.10 & 0.12 & -0.10 & 0.98 \\
-0.37 & 0.83 & -0.37 & -0.17
\end{bmatrix}}
$$

SVD: Example – Users-to-Movies

- - example

SciFi-concept

gth” of the SciFi-concept

$$
\begin{array}{cc|cccc}
 & & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\
\hline
\text{SciFi} & & 1 & 3 & 1 & 0 \\
 & & 3 & 4 & 3 & 1 \\
 & & 4 & 2 & 4 & 0 \\
 & m & 5 & 4 & 5 & 1 \\
\text{Romance} & & 0 & 1 & 0 & 4 \\
 & & 0 & 0 & 0 & 3 \\
 & & 0 & 1 & 0 & 5
\end{array}
$$

n

$$
m \; ¿ \; \begin{bmatrix}
 & & \\
0.48 & -0.02 & 0.43 \\
0.49 & -0.08 & -0.5 \\
0.68 & -0.07 & -0.16 \\
0.06 & 0.57 & 0.06 \\
0.01 & 0.41 & -0.20 \\
0.07 & 0.70 & -0.01
\end{bmatrix}
\begin{bmatrix}
 & 0 & 0 \\
 & 7.1 & 0 \\
0 & 0 & 2.5
\end{bmatrix}
\; \text{x}
$$

$$
\begin{bmatrix}
0.59 & 0.54 & 0.59 & 0.05 \\
-0.10 & 0.12 & -0.10 & 0.98 \\
-0.37 & 0.83 & -0.37 & -0.17
\end{bmatrix}
$$

n

SVD: Example – Users-to-Movies

- - example

V is "movie-to-concept" similarity matrix

SciFi-concept

$$
\begin{array}{ll|cccc}
 & & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\
 & \text{SciFi} & 1 & 3 & 1 & 0 \\
 & & 3 & 4 & 3 & 1 \\
 & & 4 & 2 & 4 & 0 \\
m & & 5 & 4 & 5 & 1 \\
 & \text{Romance} & 0 & 1 & 0 & 4 \\
 & & 0 & 0 & 0 & 3 \\
 & & 0 & 1 & 0 & 5 \\
\end{array}
$$

n

m ¿

$$
\begin{bmatrix}
0.48 & -0.02 & 0.43 \\
0.49 & -0.08 & -0.5 \\
0.68 & -0.07 & -0.16 \\
0.06 & 0.57 & 0.06 \\
0.01 & 0.41 & -0.20 \\
0.07 & 0.70 & -0.01
\end{bmatrix}
\begin{bmatrix}
 & 0 & 0 \\
 & 7.1 & 0 \\
0 & 0 & 2.5
\end{bmatrix}
\text{ X }
$$

$$
\begin{bmatrix}
0.59 & 0.54 & 0.59 & 0.05 \\
-0.10 & 0.12 & -0.10 & 0.98 \\
-0.37 & 0.83 & -0.37 & -0.17
\end{bmatrix}
$$

n

SciFi-concept

Q: Does the movie "Twilight" relate to concept "Romance"?

SVD: Interpretation #1

- "movies", "users" and "concepts"
  - : user-to-concept similarity matrix
  - : movie-to-co
  - : its diagonal
    - 'strength' of each concept

SVD: Interpretations #2

- SVD gives 'best' axis to project on
  - 'best' = mini
    squares of p
    errors

- In other words,

minimum reconstruction error

SVD: Interpretation #2

- - example
  - U: user-to-concept matrix
  - V: movie-to-c

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \text{¿} \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \times \begin{bmatrix} & & 0 \\ & & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \times \begin{bmatrix} \boxed{0.59 \quad 0.54 \quad 0.59 \quad 0.05} \\ -0.10 \quad 0.12 \quad -0.10 \quad 0.98 \\ -0.37 \quad 0.83 \quad -0.37 \quad -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- - example

variance ("spread")

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} ¿ \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \times \begin{bmatrix} & & 0 \\ & & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \times$$

$$\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- - example
  - : the coordinates of the points in the projection axis

Projection of users on the "Sci-Fi" axis

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \qquad \begin{bmatrix} \boxed{\begin{matrix} 2.86 \\ 5.71 \\ 5.83 \\ 8.09 \\ 0.71 \\ 0.12 \\ 0.83 \end{matrix}} & \begin{matrix} 0.24 \\ -0.24 \\ -0.95 \\ -0.83 \\ 6.78 \\ 4.88 \\ 8.33 \end{matrix} & \begin{matrix} 8.21 \\ 5.12 \\ -6.19 \\ -1.90 \\ 0.71 \\ -2.38 \\ -0.12 \end{matrix} \end{bmatrix}$$

SVD: Interpretation #2

- Q: how exactly is dimension reduction done?

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} ¿ \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \text{ x } \begin{bmatrix} & & 0 \\ & & 0 \\ 0 & 0 & 2.5 \end{bmatrix} \text{ x } \begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- Q: how exactly is dimension reduction done?
- A: Set smallest singular values to zero

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \text{¿} \begin{bmatrix} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{bmatrix} \text{ X } \begin{bmatrix} & & 0 \\ & & 0 \\ 0 & 0 & \cancel{2.5} \end{bmatrix} \text{ X }$$

$$\begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{bmatrix}$$

SVD: Interpretation #2

- Q: how exactly is dimension reduction done?
- A: Set smallest singular values to zero
  - Approximate -rank matrices

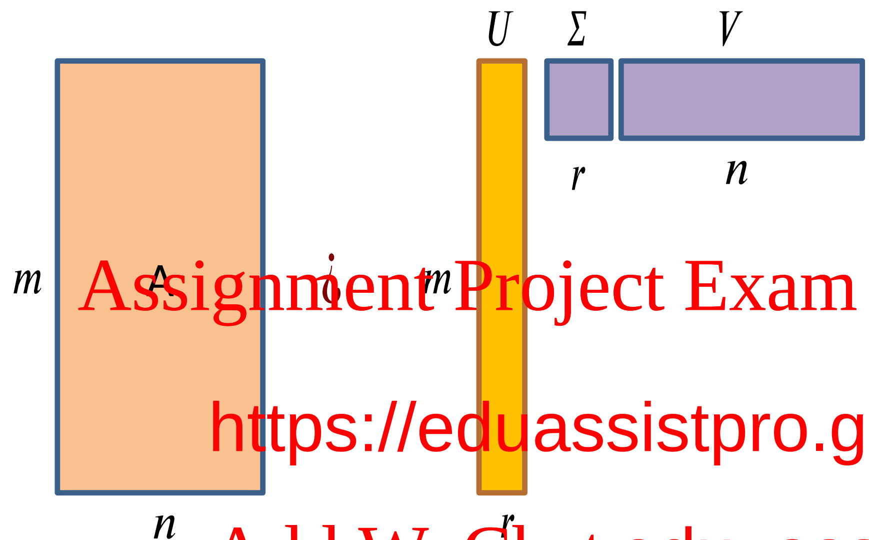
$$
\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \approx \begin{bmatrix} 0.24 & 0.02 & \cancel{0.69} \\ 0.48 & -0.02 & \cancel{0.43} \\ 0.49 & -0.08 & \cancel{-0.52} \\ 0.68 & -0.07 & \cancel{-0.16} \\ 0.06 & 0.57 & \cancel{0.06} \\ 0.01 & 0.41 & \cancel{-0.20} \\ 0.07 & 0.70 & \cancel{-0.01} \end{bmatrix} \times \begin{bmatrix} & & 0 \\ & & 0 \\ 0 & 0 & \cancel{2.5} \end{bmatrix} \times \begin{bmatrix} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ \cancel{-0.37} & \cancel{0.83} & \cancel{-0.37} & \cancel{-0.17} \end{bmatrix}
$$

SVD: Interpretation #2

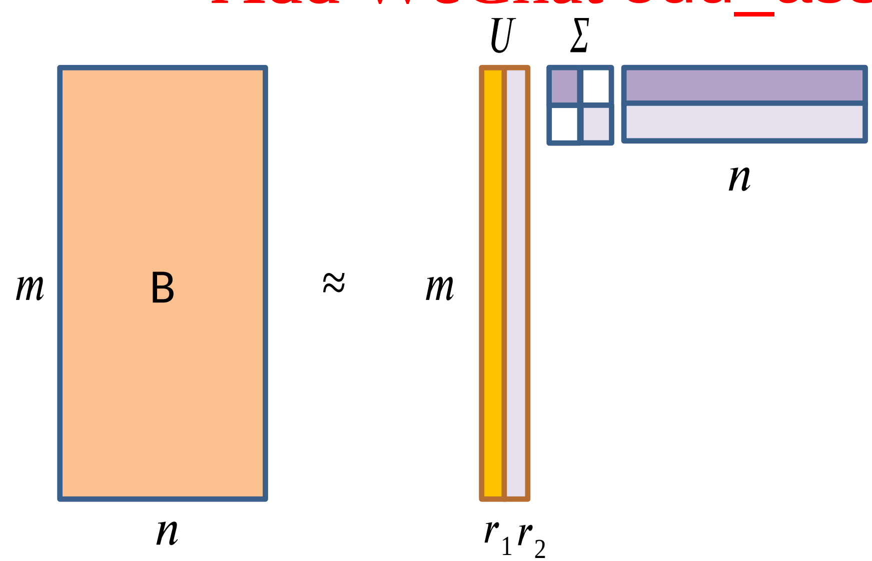
- Q: how exactly is dimension reduction done?
- A: Set smallest singular values to zero
  - Approximate -rank matrices

$$\begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \approx \quad \text{x} \quad \text{x}$$

SVD: Best Low Rank Approximation



is a rank matrix



is the **best** rank approximation of matrix

SVD: Best Low Rank Approximation

- **Theorem**: Let (), and
  - = diagonal matrix where (and ()
  - or equivalently, is the best rank- approximation to :
  - or equivalently,

- Intuition (spectra
  - Why setting small to 0 is the right thing to do?
    - Vectors and are unit length, so scales them.
    - Therefore, zeroing small introduces less error.

SVD: Interpretation #2

- Q: How many σ_i to keep?
- A: Rule-of-a thumb

Keep 80~90

$$\begin{array}{cc} m & \begin{bmatrix} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{bmatrix} \\ & n \end{array} \text{¿} \quad \sigma_1 u_1 \circ v_1^T + \sigma_2 u_2 \circ v_2^T + \cdots$$

Assume: $\sigma_1 \geq \sigma_2 \geq \cdots$

SVD: Complexity

- SVD for full matrix

- But
  - Less work, if w singular values
  - or if we only want first k sing (thin-svd).
  - or if the matrix is sparse (sparse svd).
- Stable implementations
  - LINPACK, Matlab, Splus, Mathematica...
  - Available in most common languages

SVD: Conclusions so far

- SVD: : unique
  - user-to-concept similarities
  - movie-to-co
  - : strength to
- Dimensionality reduction
  - Keep the few largest singular values (80-90% of "energy")
  - SVD: picks up linear correlations

SVD: Relationship to Eigen-decomposition

- SVD gives us

- Eigen-decomp

  - is symmetric
  - are orthonormal (
  - are diagonal

SVD: Relationship to Eigen-decomposition

- Eigen-decomposition of and

  - So, , and
  - That is, is the matrix of eigenv and is the matrix of eigenvectors of
  - This shows how to use eigen-decomposition to compute SVD
  - Ahe singular values of are the square roots of the corresponding eigenvalues of
- Note: and are the dataset covariance matrices

A Brief Review of Eigen-Decomposition

- Eigenvalues and eigenvectors
  - matrix.
  - eigenvalue of , :
- Simple computati lues
  - Solve the equation
  - Example
    - Then
    - Then
    - Solve , we get

A Brief Review of Eigen-Decomposition

- Example (continued)

  – solve , we get e
  – now we compute eigenvector
    - for eigenvalue we need to find
    - solve
    - We get Since needs to be a unit vector, therefore
    - Similarly, we can compute

Computing Eigenvalues: Power Method

- Power method
  - choose an arbitrary
  - .
  - Theorem: seque ncipal eigenvector (i.e., the eigenvector corresponds to eigenvalue)
- Normalized power method
  - choose an arbitrary
  - Theorem: sequence converges to the principal eigenvector.

In-class Practice

- Go to practice

SVD Case Study: How to Query?

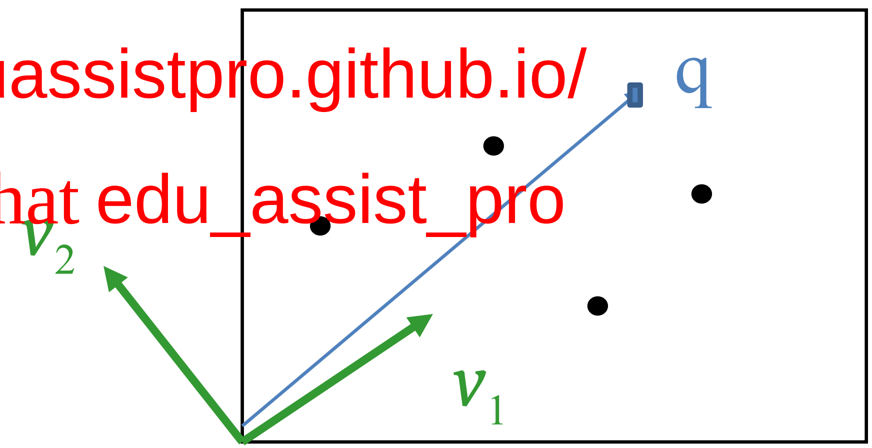
- Q: Find users that like "The Avengers"
- A: Map query into a "concept space" – how?

$$
\begin{array}{l c}
 & \begin{array}{cccc} \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \end{array} \\
\begin{array}{l} \text{SciFi} \\ \\ \\ \\ \text{Romance} \\ \\ \\ \end{array} & \left[\begin{array}{cccc} 1 & 3 & 1 & 0 \\ 3 & 4 & 3 & 1 \\ 4 & 2 & 4 & 0 \\ 5 & 4 & 5 & 1 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 5 \end{array}\right]
\end{array}
\underset{¿}{m}
\left[\begin{array}{ccc} 0.24 & 0.02 & 0.69 \\ 0.48 & -0.02 & 0.43 \\ 0.49 & -0.08 & -0.52 \\ 0.68 & -0.07 & -0.16 \\ 0.06 & 0.57 & 0.06 \\ 0.01 & 0.41 & -0.20 \\ 0.07 & 0.70 & -0.01 \end{array}\right]
\times
\left[\begin{array}{ccc} 11.9 & 0 & 0 \\ 0 & 7.1 & 0 \\ 0 & 0 & 2.5 \end{array}\right]
\times
\left[\begin{array}{cccc} 0.59 & 0.54 & 0.59 & 0.05 \\ -0.10 & 0.12 & -0.10 & 0.98 \\ -0.37 & 0.83 & -0.37 & -0.17 \end{array}\right]
$$

Case Study: How to Query?

- Q: Find users that like "The Avengers"
- A: Map query into a "concept space" – how?

$$\begin{array}{cccccc} & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\ q & [5 & 0 & 0 & 0] \end{array}$$

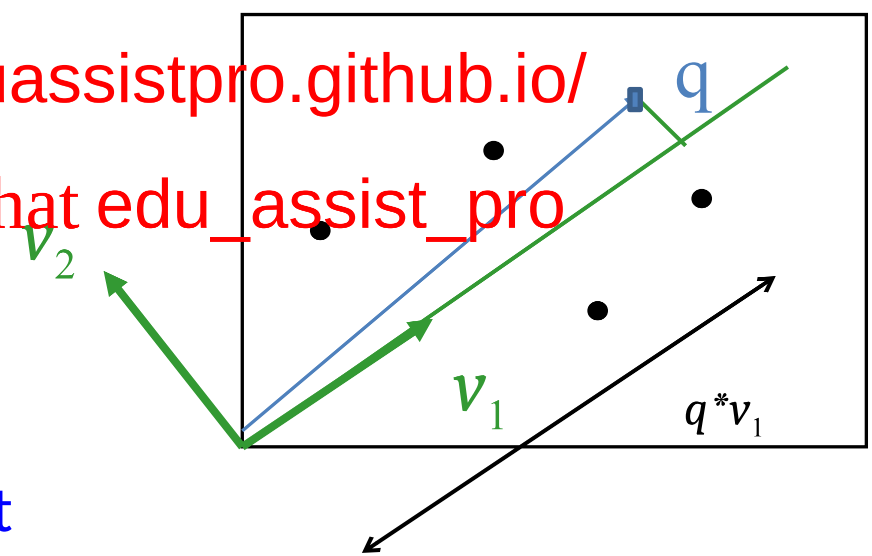


Project into concept space:
Inner product with each concept vector v_i

Case Study: How to Query?

- Q: Find users that like "The Avengers"
- A: Map query into a "concept space" – how?

$$\begin{array}{cccccc} & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\ q & [5 & 0 & 0 & 0] \end{array}$$



Project into concept space:
Inner product with each concept vector v_i

Case Study: How to Query?

- Compactly, we have
  - $q_c = qV$

ept

$$
\begin{array}{cccc} \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \end{array}
$$

$$
q \quad \begin{bmatrix} 5 & 0 & 0 & 0 \end{bmatrix} \quad \text{x} \quad \begin{bmatrix} 0.59 & -0.10 \\ 0.54 & 0.12 \\ 0.59 & -0.10 \\ 0.05 & 0.98 \end{bmatrix} \quad = \quad \begin{bmatrix} 2.95 & -0.50 \end{bmatrix}
$$

movie-to-concept
similarities ()

Case Study: How to Query?

- How would the user *d* that rated (‘Star Wars’, ‘Matrix’) be handled?
  - $d_c = d\,V$

ept

$$d \quad \begin{array}{cccc} \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\ \hline \end{array}$$

$$d \quad \begin{bmatrix} 0 & 4 & 5 & 0 \end{bmatrix} \times \begin{bmatrix} 0.59 & -0.10 \\ 0.54 & 0.12 \\ 0.59 & -0.10 \\ 0.05 & 0.98 \end{bmatrix} = \begin{bmatrix} 5.11 & -0.02 \end{bmatrix}$$

movie-to-concept similarities ()

Case Study: How to Query?

- Observation
  - User d that rated ('Star Wars') will be similar to user q that rate ('The Avengers'), although d and q have zero ratings in co

ept

$$\begin{array}{c|cccc} & \text{The Avengers} & \text{Star Wars} & \text{Matrix} & \text{Twilight} \\ \hline d & 0 & 4 & 5 & 0 \\ q & 5 & 0 & 0 & 0 \end{array}$$

$$d \quad [0 \quad 4 \quad 5 \quad 0] \dashrightarrow [5.11 \quad -0.02]$$

$$q \quad [5 \quad 0 \quad 0 \quad 0] \dashrightarrow [2.95 \quad -0.50]$$

Zero ratings in common

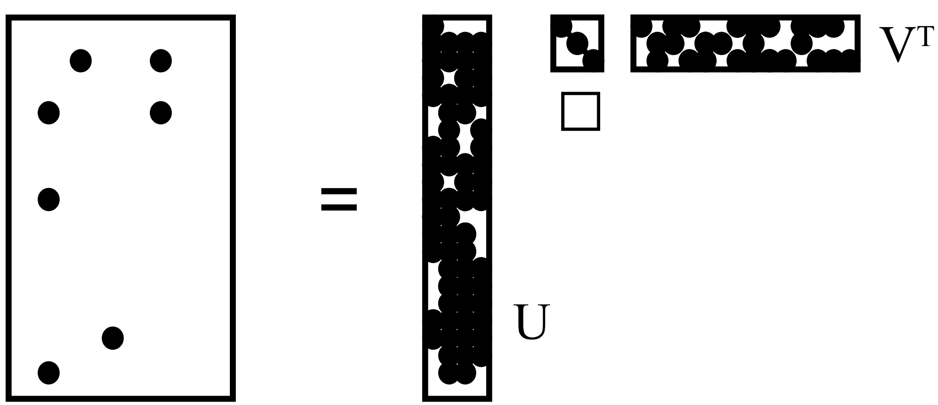
Similarity $\neq 0$

Cosine similarity: 0.99

SVD: Drawbacks

\+ **Optimal low-rank approximation** in terms of Euclidean norm

- **Interpretability problem:**
  - A singular vector specifies a linear combination rows
- **Lack of sparsity:**
  - Singular vectors are **dense!**



CUR Decomposition

- Goal: express A as a product of matrices
  - Minimize
- Constraints o

CUR Decomposition

- Goal: express A as a product of matrices
  - Minimize
- Constraints o

CUR: Good Approximation to SVD

- Let be the best rank approximation of (obtain by SVD)

- Theorem

  - CUR algorith

  - with probability at least , by picking
    - columns and
    - rows
    - (in practice, choose columns/rows)

CUR: How it Works

- Sampling columns (similarly for rows):

CUR: Computing U

- Let be the “intersection” of sampled columns C and rows R
  - Let SVD of
- Then: , where
  - is the “Moore-Penrose ps se”.
  - , if .

CUR: Pros & Cons

+ easy interpretation
  – the basis vectors are actual columns and rows

- duplicate colu
  – columns of la pled many times

CUR: Duplicate Columns

- If we want to get rid of the duplicates
  - Throw them away
  - Scale the col are root of the number of du

SVD vs CUR

Question: Large or small? Dense or sparse?

SVD vs CUR

SVD & CUR: Simple Experiments

- DBLP data
  - author-to-conference matrix
    - very sparse

  - : number of pap at conference .

  - 428k authors (r
  - 3659 conferences (column)

- Dimensionality reduction
  - Running time?
  - Space?
  - Reconstruction error?

Results: DBLP

courtesy: Sun, Faloutsos: *Less is more*, *Compact Matrix Decomposition for Large Sparse Graph*, SDM'07

- accuracy: 1-relative sum squared errors
- space ratio: # output non-zero matrix entries / # input non-zero matrix entries

The Linearity Assumption

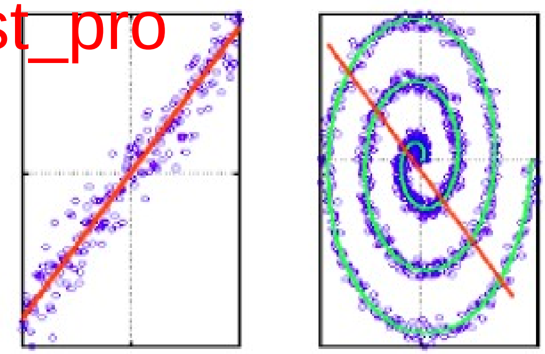
- SVD is limited to linear projections
  - Data lies on a low-dimensional linear space

- Non-linear me
  - Data lies on a

ifold

    - Non-linear

  - How?
    - Build adjacency graph
    - SVD the graph adjacency matrix
    - Further reading: wikipage of Isomap



PCA: An Application of SVD

- PCA = Principle Component Analysis

- Motivation
  - Visualization

PCA: Data Visualization

- Example:
  - Given 53 blood samples (features) from 65 people (data item or instance)

  - How can we visualize the samples

PCA: Data Visualization

How can we visualize the other variables???

… difficult to see in 4 or higher dimensional spaces …

PCA: Data Visualization

- Is there a representation better than the coordinate axes?
- Is it really necessary to show all the 53 dimensions?

  - What if there ar ween the features?
  - How could we find the smallest f the 53-D space that keeps the most informatio original data?

- A solution: Principal Component Analysis
  - An application of SVD.

PCA: Definition and Algorithms

- PCA
  - Orthogonal projection of the data onto a lower-dimensional line that
    - Maximize variance of projected data (purple line)
    - Minimize mean squared distance between
      - Data point
      - Projection (sum of blue lines)
  - Look data from a literally different angle.

PCA: Idea

- Given data points in a d-dimensional space, project them into a lower dimensional space while preserving as s possible.
  - Find best pla D data
  - Find best 12-D approximati -D data
- In particular, choose projection that minimizes squared error in reconstructing the original data.
  - Implement through SVD

PCA

- **PCA Vectors** originate from the center of mass.
- Principal component #1: points in the direction of the **largest va**

- Each subsequ nent
  - is **orthogonal** to the previo nd
  - points in the directions of the **largest variance of the residual subspace**

PCA: 2D Gaussian dataset

1st PCA axis

2nd PCA axis

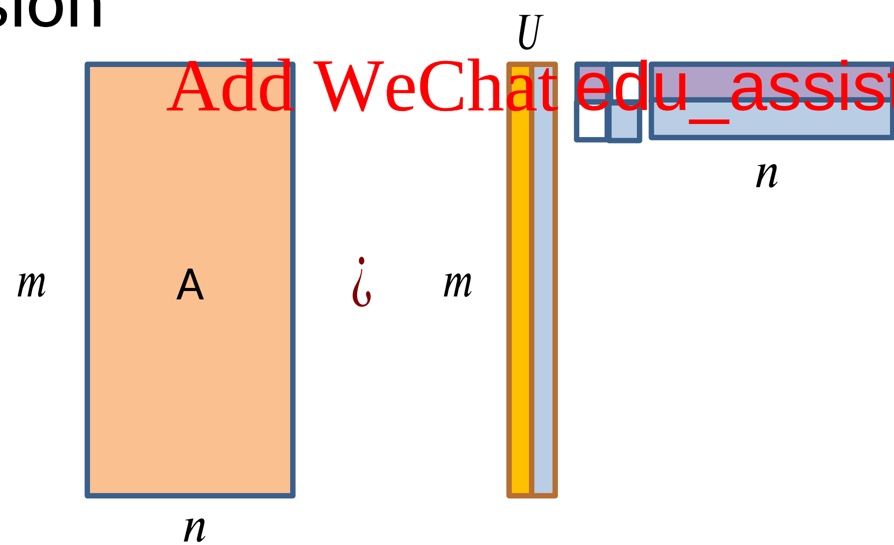
PCA: Algorithm

- Given centered data , compute principle vectors
  - 1st principle vector
  - maximize the of

PCA: Algorithm by SVD

- SVD of the centered data matrix

  - : number of i
  - : dimension



PCA: Algorithm by SVD

- Columns of
  - is exactly the principal vectors,
  - orthogonal a
- Matrix
  - Diagonal
  - Strength of each eigenvector
- Columns of
  - Coefficients for reconstructing the samples.

Application: Face Recognition

- Want to identify specific person, based on facial image
- Can't just use the given 256 x 256 pixels

Applying PCA

- **Method A:** Build a PCA subspace for each person and check which subspace can reconstruct the test image the best

- **Method B:** Buil or the whole dataset and then classify ba weights.

- Example data set: Images of faces
- Each face is …
  - values

Principal Components (Method B)

Reconstructing … (Method B)

Happiness Subspace (Method A)

Disgust Subspace (Method A)

Image Compression

- Divide the origi
  372x492 image
  patches:
  - Each patch is an instance that contains 12x12 pixels on a grid
  - View each as a 144-D vector

PCA Compression: 144D => 60D

PCA Compression: 144D => 16D

PCA Compression: 144D => 6D

6 Most Important Eigenvectors

PCA Compression: 144D => 3D

3 Most Important Eigenvectors

Noisy Image

Denoised Image using 15 PCA Components

PCA: Shortcomings

- PCA cannot capture non-linear structure
  - Similar with SVD

PCA: Shortcomings

- PCA does not know labels

PCA: Conclusions

- PCA
  - find orthonormal basis for data
  - sort dimensions in order of "strength"
  - discard low significance dimensions
- Uses
  - Get compact description
  - Ignore noise
  - Improve classification (hopefully)
- Not magic:
  - Doesn't know class labels
  - Can only capture linear variations
  - One of many tricks to reduce dimensionality!

Extra: Compute PCA Using Eigen-Decomposition

- Given centered data compute covariance matrix

- Top PCA com vectors of .

  - Equivalence position and SVD
    - SVD decomposition of ,

      - SVD-based algorithm for PCA
    - Eigen-decomposition of .
      - Eigen-based algorithm for PCA
    - The equivalence gives .

One-slide Takeaway

- Dimensionality reduction
  - compress/reduce dimension
  - reconstruct the original matrix by two or more smaller matrices

- Singular value dec
  - decompose a matr

    - : column-orthonormal. diagonal matrix.

- CUR decomposition
    - set of **c**olumns of . set of **r**ows of .
- Principle component analysis (PCA)
  - reconstruct data matrix by a smaller number of eigenvectors
  - view the data from a *literally* different angle.

In-class Practice

- **1.** Describe briefly (informally or formally) the relationship between singular value decomposition and eigenvalue decomposition.

- **2.1** Compute the eigenvectors of matrix

- **2.2** Let It is easy to check that . What are the singular values of ?
- **2.3** Obtain SVD for A where $A = U \Sigma V^T$