中国科学院大学
University of Chinese Academy of Sciences

Mining Massive Datasets Dimensionality Reduction SVD&CUR

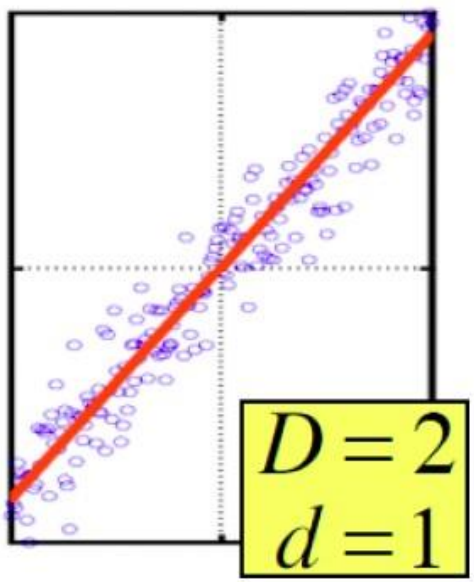
Dimensionality Reduction

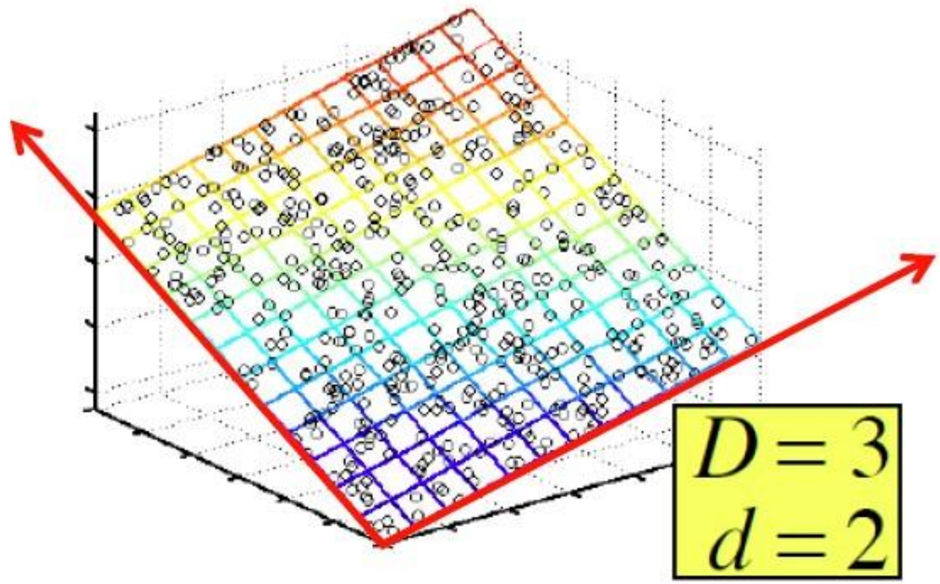
- **Compress / reduce dimensionality:**
  - 10^6 rows; 10^3 columns; no updates
  - Random access to any cell(s); **small error: OK**

customer \ day	We 7/10/96	Th 7/11/96	Fr 7/12/96	Sa 7/13/96	Su 7/14/96
ABC Inc.	1	1	1	0	0
DEF Ltd.	2	2	2	0	0
GHI Inc.	1	1	1	0	0
KLM Co.	5	5	5	0	0
Smith	0	0	0	2	2
Johnson	0	0	0	3	3
Thompson	0	0	0	1	1

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]

Dimensionality Reduction





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

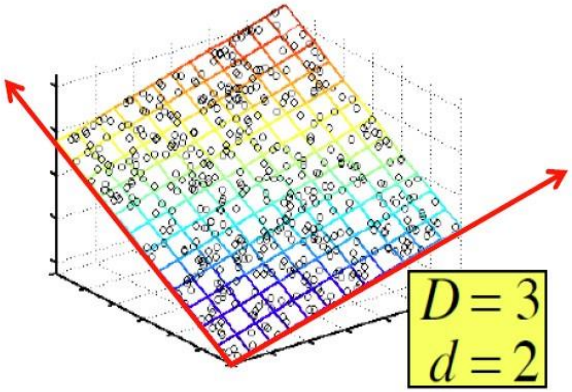
Why Reduce Dimensions?

- Some features may be irrelevant
- We want to visualize high dimensional data
- "Intrinsic" dimensionality may be smaller than the number of features
- In particular, choose projection that minimizes the squared error in reconstructing original data

Why Reduce Dimensions?

Why reduce dimensions?

- **Discover hidden correlations/topics**
  - Words that occur commonly together
- **Remove redundant and noisy features**
  - Not all words are useful
- **Interpretation and visualization**
- **Easier storage and processing of the data**



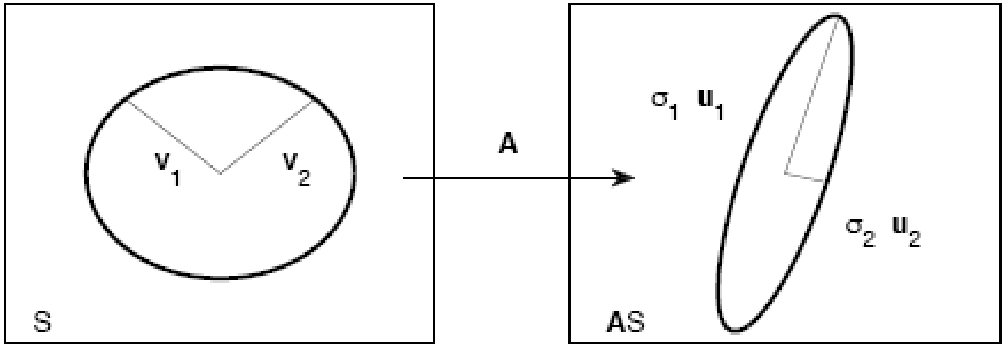
SVD-Definition

$$A_{[m \times n]} = U_{[m \times r]} \Sigma_{[r \times r]} (V_{[n \times r]})^T$$

- **A: Input data matrix**
  - $m \times n$ matrix (e.g., m documents, n terms)
- **U: Left singular vectors**
  - $m \times r$ matrix (m documents, r concepts)
- **Σ: Singular values**
  - $r \times r$ diagonal matrix (strength of each 'concept') (r : rank of the matrix **A**)
- **V: Right singular vectors**
  - $n \times r$ matrix (n terms, r concepts)

SVD-decomposition

- The SVD, much as illustrated in the following figure, is essentially a transformation that stretches/compresses and rotates a given set of vectors



the transformation from the unit sphere to the hyperellipse

$$\mathbf{A}\mathbf{v}_j = \sigma_j \mathbf{u}_j,$$

$$\mathbf{A}^T \mathbf{u}_i = \sigma_j \mathbf{v}_j.$$

SVD–Eigenvalue & Eigenvector

Given a n x n matrix A^TA, for any σ and v, if

$$A^T A v_j = \sigma_j v_j$$

Then σ is called eigenvalue, and w is called eigenvector.

- To gain insight into the SVD, treat the rows of an $m \times n$ matrix A as n points in a n-dimensional space and consider the problem of finding the best r-dimensional subspace with respect to the set of points. Here best means minimize the sum of the squares of the perpendicular distances of the points to the subspace.

SVD-decomposition

- The objective of the rotation transformation is to find the maximal variance. We Projection of data along v is Av. Variance:

$$\sigma^2 = (Av)^T (Av) = v^T A^T Av$$

where $A^T A$ is the covariance matrix of the data

Objective: maximize variance subject to constraint $v^T v=1$.

Maximize $\qquad f = v^T A^T Av - \lambda(v^T v - 1)$

λ is the Lagrange multiplier, Differentiating with respect to v yields Eigenvalue equation:

$$A^T Av = \lambda v$$

SVD-decomposition

- The objective of the rotation transformation is to find the maximal variance. We Projection of data along v is Av.

Variance:

$$\sigma^2 = (Av)^T (Av) = v^T A^T Av$$

where $A^T A$ is the covariance matrix of the data

Objective: maximize variance subject to constraint $v^T v=1$.

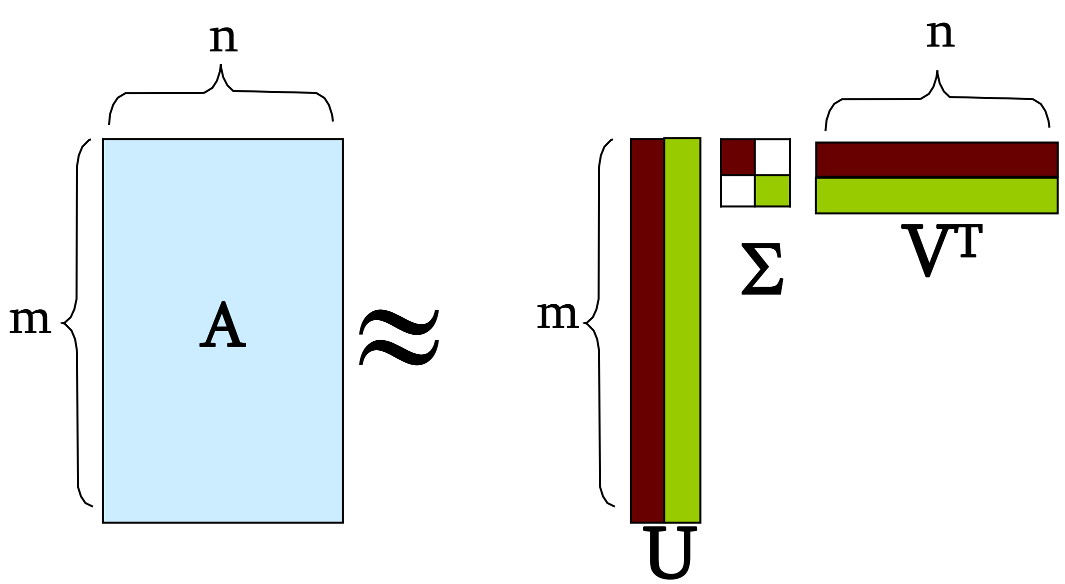
Maximize $$f = v^T A^T Av - \sigma(v^T v - 1)$$

σ is the Lagrange multiplier, Differentiating with respect to v yields Eigenvalue equation:

$$A^T Av = \lambda v$$

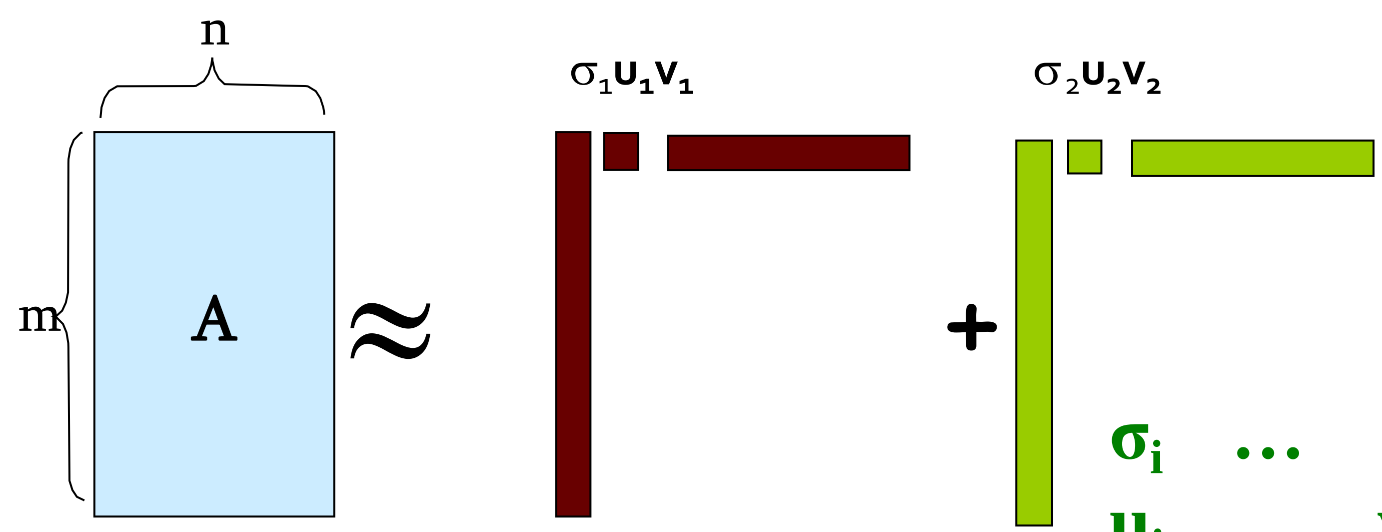
SVD

$$\mathbf{A} \approx \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^{\mathrm{T}}$$



SVD

$$\mathbf{A} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_i \sigma_i \mathbf{u}_i \circ \mathbf{v}_i^{\mathrm{T}}$$



σ_i … scalar

u_i … vector

v_i … vector

SVD

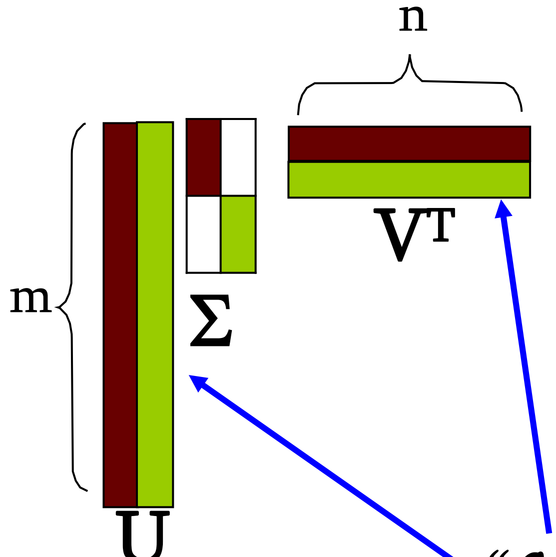
It is **always** possible to decompose a real matrix A into $A = U \Sigma V^T$, where

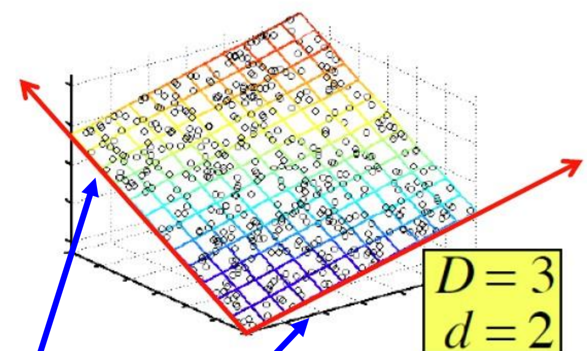
- U, Σ, V: unique
- U, V: column orthonormal
  - $U^T U = I$; $V^T V = I$ (I: identity matrix)
  - (Columns are orthogonal unit vectors)
- Σ: diagonal
  - Entries (**singular values**) are positive, and sorted in decreasing order ($\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$)

SVD-Example:Users-to-Movies

- $A = U\Sigma V^T$ - example: Users to Movies

$$
\begin{array}{c|ccccc}
 & \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanc} & \text{a Amelie} \\
\hline
 & & & & & \\
\text{SciFi} & & & & & \\
 & & & & & \\
\text{Romnce} & & & & & \\
\end{array}
\qquad
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
=
$$





"Concepts" AKA Latent dimensions AKA Latent factors

SVD-Example:Users-to-Movies

- $A = U\Sigma V^T$ - example: Users to Movies

$$
\begin{array}{cc}
 & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} \\
\begin{array}{c}\text{SciFi} \\ \\ \\ \\ \text{Romnce}\end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

SVD-Example:Users-to-Movies

- $A = U\Sigma V^T$ - example: Users to Movies

SciFi-concept

Romance-concept

$$
\begin{array}{c}
 \\ \text{SciFi} \\ \\ \\ \\ \text{Romnce} \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanc} & \text{Amelie} \end{array} \\
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

SVD-Example:Users-to-Movies

- $A = U\Sigma V^T$ - example: U is "user-to-concept" similarity matrix

SciFi-concept Romance-concept

$$
\begin{array}{c c}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanc} & \text{a Amelie} \end{array} \\
\begin{array}{c} \\ \text{SciFi} \\ \\ \\ \\ \text{Romnce} \\ \\ \end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

SVD-Example:Users-to-Movies

- $A = U \Sigma V^T$ - example:

SciFi-concept

"strength" of the SciFi-concept

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanc} & \text{a Amelie} \end{array} \\
\begin{array}{c} \\ \text{SciFi} \\ \\ \\ \\ \text{Romnce} \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

SVD-Example:Users-to-Movies

- $A = U\Sigma V^T$ - example: V is "movie-to-concept" similarity matrix

SciFi-concept

$$
\begin{array}{c}
\\ \text{SciFi} \\ \\ \\ \\ \text{Romnce} \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanc} & \text{a Amelie} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

SciFi-concept

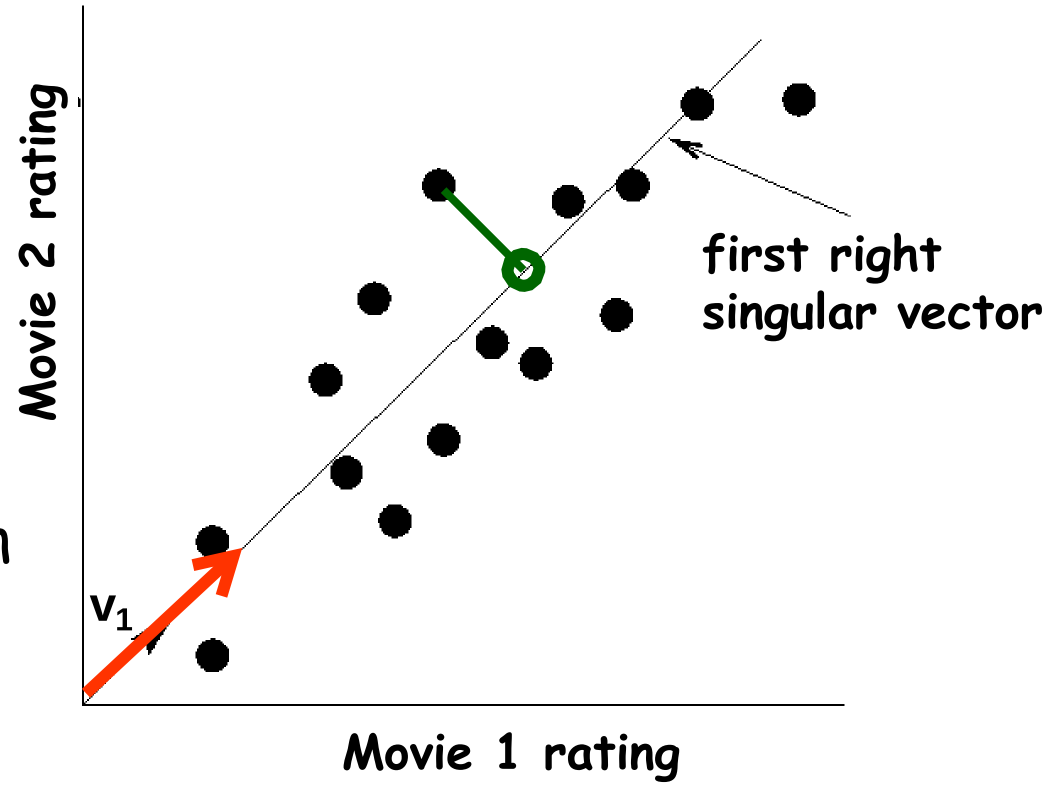
SVD-Interpretation #1

'**movies**', '**users**' and '**concepts**':

- *U*: user-to-concept similarity matrix
- *V*: movie-to-concept similarity matrix
- Σ: its diagonal elements: 'strength' of each concept

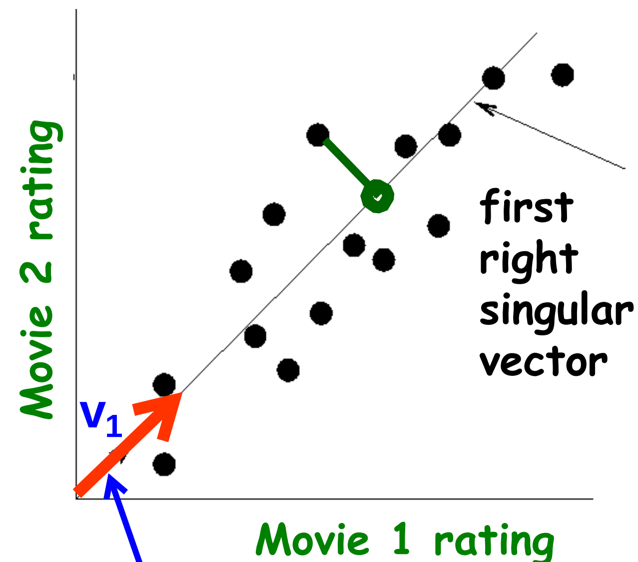
SVD-Interpretation #2

- **SVD gives 'best' axis to project on:**
  - '**best**' = min sum of squares of projection errors
  - In other words, minimum reconstruction error



SVD-Interpretation #2

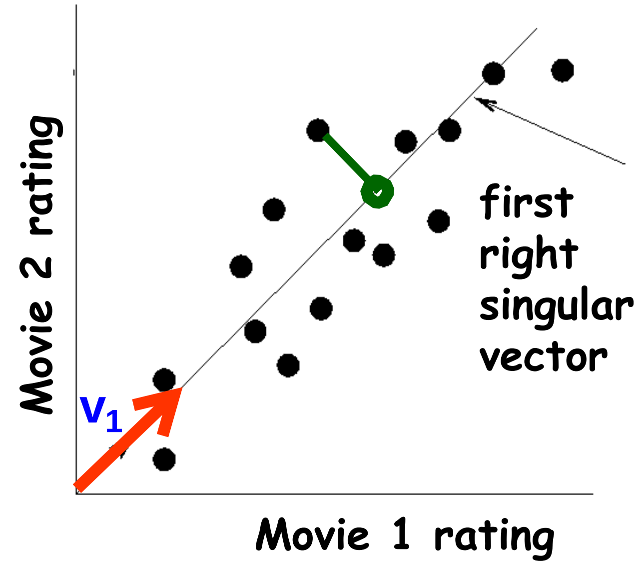
- $A = U\Sigma V^T$ - example:
  - V: "movie-to-concept" matrix
  - U: "user-to-concept" matrix



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

SVD-Interpretation #2

- $A = U\Sigma V^T$ - example:

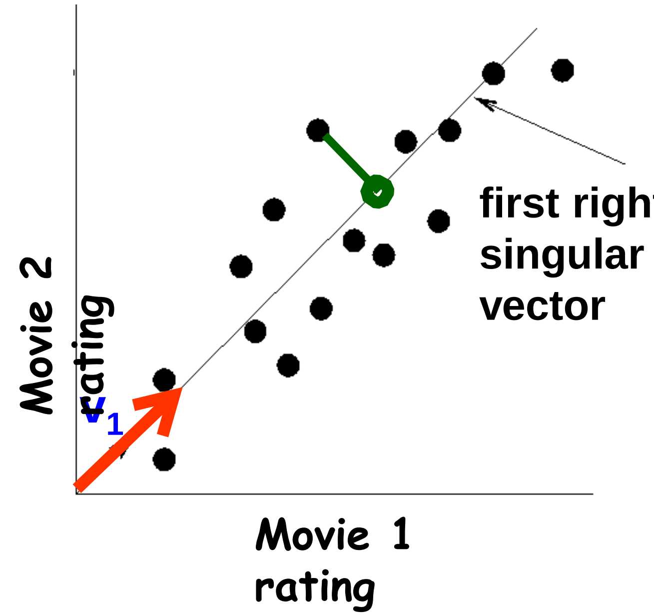


variance ('spread') on the v_1 axis

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

SVD-Interpretation #2

- $A = U\Sigma V^T$ - example:
  - $U\Sigma$: Gives the coordinates of the points in the projection axis



$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}$$

Projection of users on the "Sci-Fi" axis $((U\Sigma)^T)$

$$\begin{bmatrix} \boxed{\begin{matrix} 1.61 \\ 5.08 \\ 6.82 \\ 8.43 \\ 1.86 \\ 0.86 \\ 0.86 \end{matrix}} & \begin{matrix} 0.19 \\ 0.66 \\ 0.85 \\ 1.04 \\ -5.60 \\ -6.93 \\ -2.75 \end{matrix} & \begin{matrix} -0.01 \\ -0.03 \\ -0.05 \\ -0.06 \\ 0.84 \\ -0.87 \\ 0.41 \end{matrix} \end{bmatrix}$$

SVD-Interpretation #2

More details

- Q: How exactly is dim. reduction done?

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

SVD-Interpretation #2

More details

- **Q: How exactly is dim. reduction done?**
- **A: Set smallest singular values to zero**

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

SVD-Interpretation #2

More details

- Q: How exactly is dim. reduction done?
- A: Set smallest singular values to zero

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

SVD-Interpretation #2

More details

- Q: How exactly is dim. reduction done?
- A: Set smallest singular values to zero

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

SVD-Interpretation #2

More details

- Q: How exactly is dim. reduction done?
- A: Set smallest singular values to zero

$$
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{bmatrix}
\approx
\begin{bmatrix}
\mathbf{0.13} & 0.02 \\
\mathbf{0.41} & 0.07 \\
\mathbf{0.55} & 0.09 \\
\mathbf{0.68} & 0.11 \\
0.15 & \mathbf{-0.59} \\
0.07 & \mathbf{-0.73} \\
0.07 & \mathbf{-0.29}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 \\
0 & \mathbf{9.5}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69}
\end{bmatrix}
$$

SVD-Interpretation #2

More details

- Q: How exactly is dim. reduction done?
- A: Set smallest singular values to zero

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.92} & \mathbf{0.95} & \mathbf{0.92} & 0.01 & 0.01 \\ \mathbf{2.91} & \mathbf{3.01} & \mathbf{2.91} & -0.01 & -0.01 \\ \mathbf{3.90} & \mathbf{4.04} & \mathbf{3.90} & 0.01 & 0.01 \\ \mathbf{4.82} & \mathbf{5.00} & \mathbf{4.82} & 0.03 & 0.03 \\ 0.70 & \mathbf{0.53} & 0.70 & \mathbf{4.11} & \mathbf{4.11} \\ -0.69 & 1.34 & -0.69 & \mathbf{4.78} & \mathbf{4.78} \\ 0.32 & \mathbf{0.23} & 0.32 & \mathbf{2.01} & \mathbf{2.01} \end{bmatrix}$$

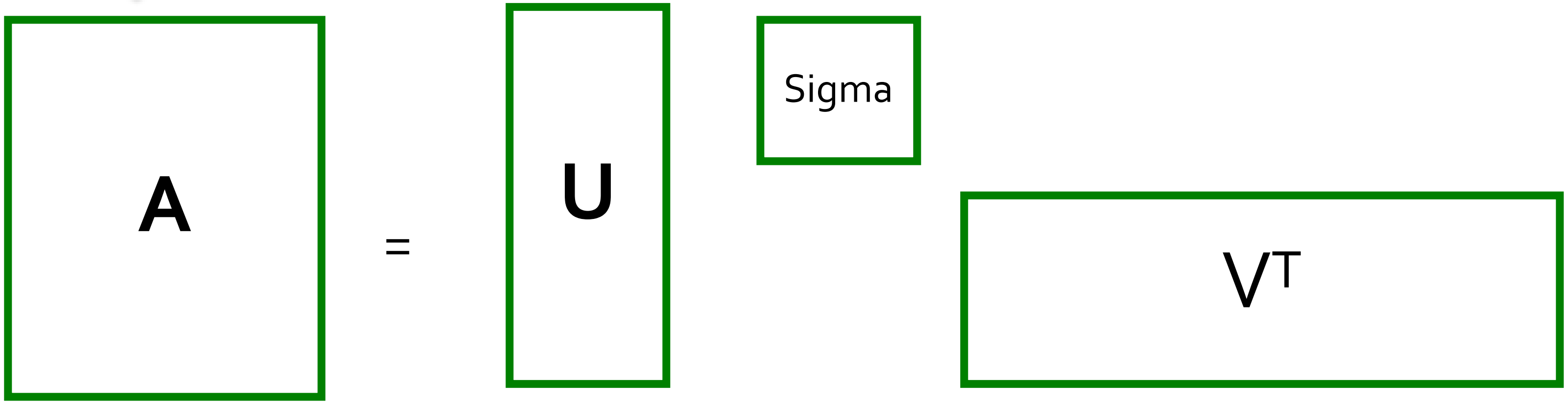
Frobenius norm:

$$\|M\|_F = \sqrt{\Sigma_{ij} M_{ij}^2}$$

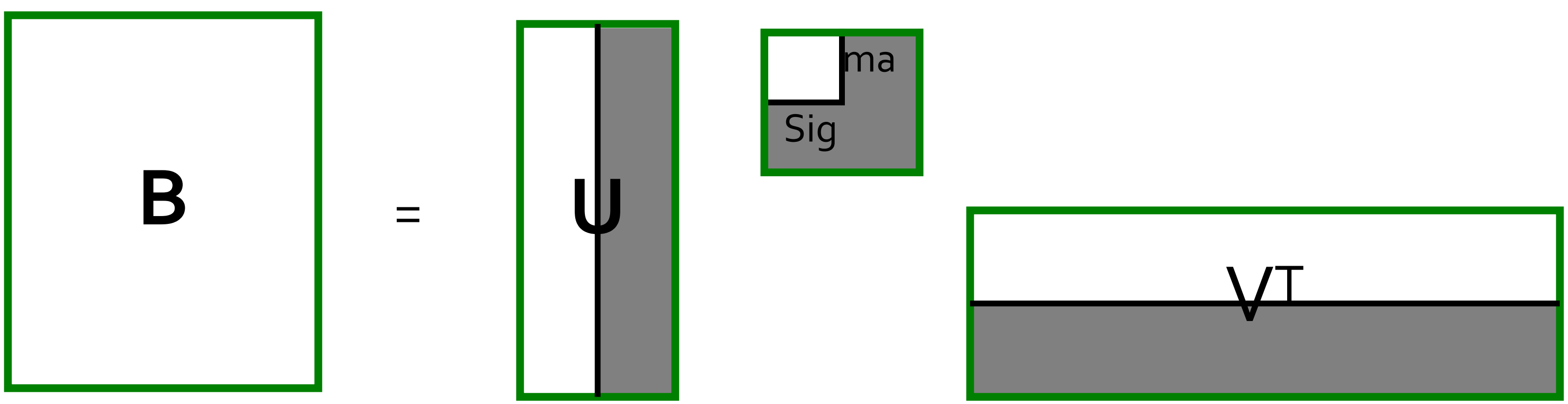
$$\|A-B\|_F = \sqrt{\Sigma_{ij} (A_{ij}-B_{ij})^2}$$

is "small"

SVD-Best Low Rank Approx.



B is best approximation of A

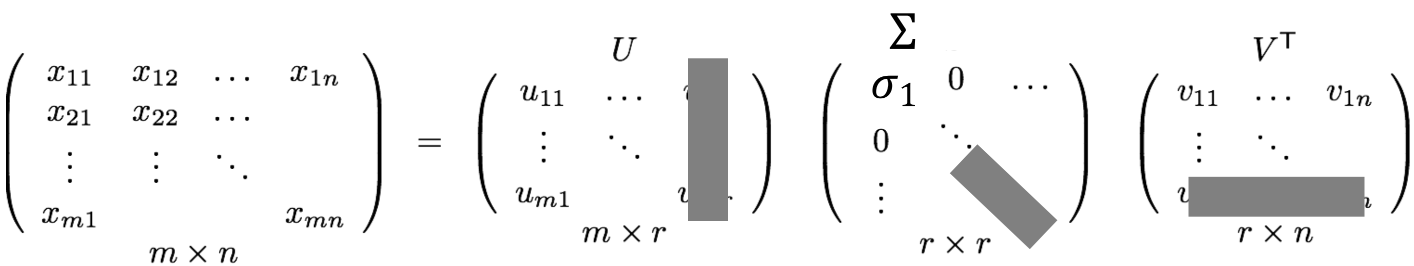


SVD-Best Low Rank Approx.

- <u>Theorem:</u> Let $A = U\,\Sigma\,V^T$ $\quad(\sigma_1 \geq \sigma_2 \geq \ldots,\ \mathrm{rank}(A)=r)$
  then $B = U\,S\,V^T$
  - S = diagonal nxn matrix where $s_i=\sigma_i$ $(i=1\ldots k)$ else $s_i=0$

is a best rank-k approximation to *A*:

- B is a solution to $\min_B \|A-B\|_F$ where $\mathrm{rank}(B)=k$



- **We will need 2 facts:**
  - $\|M\|_F = \sum_i (q_{ii})^2$ where ***M*** = ***P Q R*** is SVD of ***M***
  - $\mathbf{U\,\Sigma V^T - U\,S\,V^T = U\,(\Sigma - S)\,V^T}$

SVD-Best Low Rank Approx.

- **We will need 2 facts:**
  - $\left\| M \right\|_F = \sum_k (q_{kk})^2$ where ***M*** = ***P Q R*** is SVD of ***M***

$$\|M\| = \sum_i \sum_j (m_{ij})^2 = \sum_i \sum_j \left(\sum_k \sum_\ell p_{ik} q_{k\ell} r_{\ell j} \right)^2$$

$$\|M\| = \sum_i \sum_j \sum_k \sum_\ell \sum_n \sum_m p_{ik} q_{k\ell} r_{\ell j} p_{in} q_{nm} r_{mj}$$

$\sum_i p_{ik} p_{in}$ is 1 if $k = n$ and 0 otherwise

We apply:
-- P column orthonormal
-- R row orthonormal
-- Q is diagonal

- **U ΣV^T - U S V^T = U (Σ- S) V^T**

SVD-Best Low Rank Approx.

- **A = U Σ V^T , B = U S V^T** $(\sigma_1 \geq \sigma_2 \geq \ldots \geq 0, \text{rank}(\boldsymbol{A})=\boldsymbol{r})$
  - **S** = diagonal $n_x n$ matrix where $s_i=\sigma_i$ (i=1…k) else s_i=0

then $\boldsymbol{B}$ is solution to $\min_{\boldsymbol{B}} \|\boldsymbol{A}\text{-}\boldsymbol{B}\|_F$, $\text{rank}(\boldsymbol{B})=\boldsymbol{k}$

- **Why?**

$$\min_{B, rank(B)=k} \|A - B\|_F = \min \|\Sigma - S\|_F = \min_s \sum_{i=1}^{r} (\sigma_i^2 - s_i^2)$$

We used: U ΣV^T - U S V^T = U (Σ- S) V^T

- We want to choose $\boldsymbol{s_i}$ to minimize $\sum(\sigma_i - s_i)^2$
  - We set $\boldsymbol{s_i=\sigma_i}$ (i=1…k) and other $\boldsymbol{s_i=0}$

$$= \min_{s_i} \sum_{i=1} (\sigma_i - s_i)^2 + \sum_{i=k+1} \sigma_i^2 = \sum_{i=k+1} \sigma_i^2$$

SVD-Interpretation #2

Equivalent:
'spectral decomposition' of the matrix:

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} | & | \\ u_1 & u_2 \\ | & | \end{bmatrix} \times \begin{bmatrix} \sigma_1 & \varnothing \\ \varnothing & \sigma_2 \end{bmatrix} \times \begin{bmatrix} \text{———} \; v_1 \; \text{———} \\ \text{———} \; v_2 \; \text{———} \end{bmatrix}$$

SVD-Interpretation #2

Equivalent:

'spectral decomposition' of the matrix:

$$
\begin{array}{cc}
 & \longleftarrow \; m \; \longrightarrow \\
n & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix}
\end{array}
= \overbrace{\sigma_1 \; u_1 \; v^T_1 + \sigma_2 \; u_2 \; v^T_2 + \ldots}^{k \text{ terms}}
$$

u_1: n x 1

v^T_1: 1 x m

Assume: $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots \geq 0$

Why is setting small σ_i to 0 the right thing to do?

Vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are unit length, so $\boldsymbol{\sigma}_i$ scales them.

So, zeroing small $\boldsymbol{\sigma}_i$ introduces less error.

SVD-Interpretation #2

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

keep 80-90% of 'energy' $(=\sum \sigma_i^2)$

$$
\begin{array}{c c}
 & \xleftarrow{\quad\quad} m \xrightarrow{\quad\quad} \\
n & \begin{vmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{vmatrix}
\end{array}
= \quad \sigma_1 \quad u_1 \quad v^T_1 \quad + \quad \sigma_2 \quad u_2 \quad v^T_2 \quad + \ldots
$$

Assume: $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \ldots$

SVD-Complexity

- **To compute SVD:**
  - **$O(nm^2)$** or **$O(n^2m)$** (whichever is less)
  - **But:**
  - Less work, if we just want singular values
  - or if we want first *k* singular vectors
  - or if the matrix is sparse
  - **Implemented in** linear algebra packages like
  - LINPACK, Matlab, SPlus, Mathematica ...

SVD-Conclusions so far

- **SVD: $A = U \Sigma V^T$: unique**
  - **U**: user-to-concept similarities
  - **V**: movie-to-concept similarities
  - Σ : strength of each concept
- **Dimensionality reduction:**
  - keep the few largest singular values (80-90% of 'energy')
  - SVD: picks up linear correlations

Relation to Eigen-decompositon

- **SVD gives us:**
  - $A = U\,\Sigma V^T$
- **Eigen-decomposition:**
  - $A = X\,\Lambda X^T$
    - A is symmetric
    - U, V, X are orthonormal (**$U^TU=I$**),
    - Λ, Σ are diagonal
- **What is:**
  - $AA^T = U\Sigma V^T(U\Sigma V^T)^T \quad = U\Sigma V^T(V\Sigma^T\Sigma U^T) = U^T\Sigma\Sigma^T U^T$
  - $A^TA = V\,\Sigma^T\,U^T\,(U\Sigma V^T) = V\,\Sigma\Sigma^T\,V^T$

Relation to Eigen-decompositon

- **SVD gives us:**
  - $A = U\,\Sigma V^T$
- **Eigen-decomposition:**
  - $A = X\,\Lambda X^T$
    - A is symmetric
    - U, V, X are orthonormal (**$U^TU=I$**),
    - Λ, Σ are diagonal
- **What is:**
  - $\mathbf{AA^T= U\Sigma V^T(U\Sigma V^T)^T = U\Sigma V^T(V\Sigma^T U^T) = U\Sigma\Sigma^T\ U^T}$
  - $\mathbf{A^TA = V\ \Sigma^T\ U^T\ (U\Sigma V^T) = V\ \Sigma^T\Sigma\ V^T}$

Shows how to compute SVD using eigenvalue decomposition!

$X\ \Lambda\ X^T$

$X\ \Lambda\ X^T$

So, $\lambda_i = \sigma_i^2$

Case study: How to query?

$$
\begin{array}{c}
\\
\text{SciFi} \\
\\
\\
\text{Romnce} \\
\\
\end{array}
\begin{array}{ccccc}
\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\
\mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\
\mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\
0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\
0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\
0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2}
\end{array}
=
\begin{bmatrix}
\mathbf{0.13} & 0.02 & -0.01 \\
\mathbf{0.41} & 0.07 & -0.03 \\
\mathbf{0.55} & 0.09 & -0.04 \\
\mathbf{0.68} & 0.11 & -0.05 \\
0.15 & \mathbf{-0.59} & \mathbf{0.65} \\
0.07 & \mathbf{-0.73} & \mathbf{-0.67} \\
0.07 & \mathbf{-0.29} & \mathbf{0.32}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \mathbf{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \\
0.40 & \mathbf{-0.80} & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

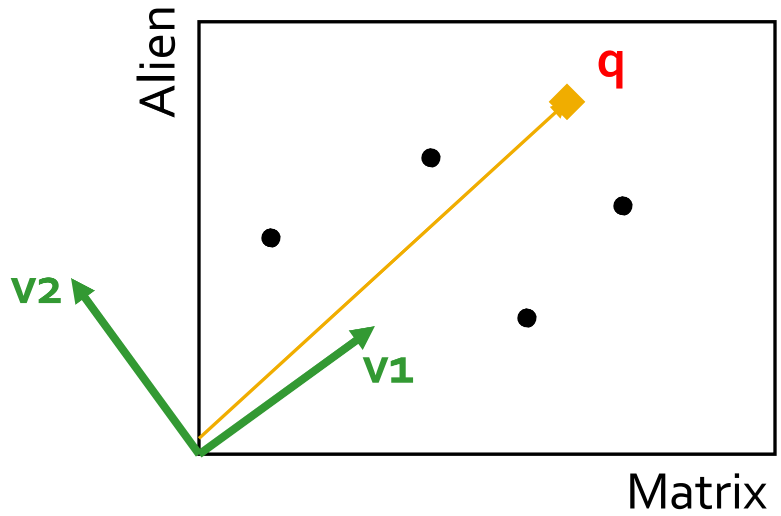
- Q: Find users that like 'Matrix'
- A: Map query into a 'concept space' – how?

Case study: How to query?

- **Q: Find users that like 'Matrix'**
- ***A: Map query into a 'concept space' – how?***

$$
\mathbf{q} = \begin{array}{c} \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\ \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}
$$

Project into concept space:
Inner product with each 'concept' vector $\mathbf{v}_i$

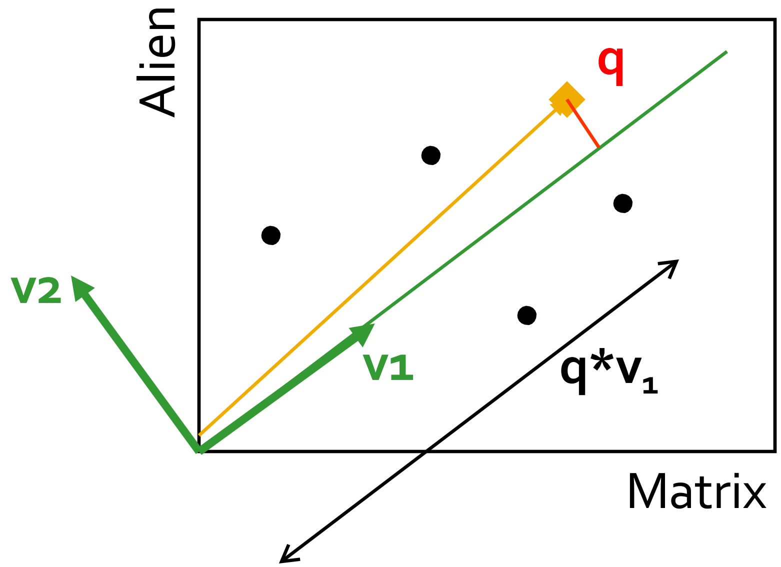


Case study: How to query?

- **Q: Find users that like 'Matrix'**
- ***A*: Map query into a 'concept space' – how?**

$$\mathbf{q} = \begin{array}{c} \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\ \left[\begin{array}{ccccc} 5 & 0 & 0 & 0 & 0 \end{array} \right] \end{array}$$

Project into concept space:
Inner product with each 'concept' vector $\mathbf{v}_i$



Case study: How to query?

Compactly, we have: $q_{concept} = q\ V$

E.g.:

$$\mathbf{q} = \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \hline \end{array}$$

$$\mathbf{q} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} = \begin{array}{cc} \text{SciFi-concept} & \\ \begin{bmatrix} 2.8 & 0.6 \end{bmatrix} \end{array}$$

movie-to-concept similarities (V)

Case study: How to query?

- **How would the user *d* that rated ('Alien', 'Serenity') be handled?**

$$\mathbf{d_{concept} = d\ V}$$

E.g.:

$$q = \begin{array}{c} \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\ \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \end{array} \times \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} = \begin{array}{c} \begin{array}{cc} \text{SciFi-concept} & \end{array} \\ \begin{bmatrix} 5.2 & 0.4 \end{bmatrix} \end{array}$$

movie-to-concept similarities (V)

Case study: How to query?

- **Observation:** User ***d*** that rated ('*Alien*', '*Serenity*') will be **similar** to user **q** that rated ('*Matrix*'), although ***d*** and **q** have **zero ratings in common**!

$$\begin{array}{ccc} & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} & \begin{array}{c}\text{SciFi-concept}\end{array} \\ \mathbf{d}= & \begin{bmatrix}0 & 4 & 5 & 0 & 0\end{bmatrix} \dashrightarrow & \begin{bmatrix}2.8 & 0.6\end{bmatrix} \\ \mathbf{q}= & \begin{bmatrix}5 & 0 & 0 & 0 & 0\end{bmatrix} \dashrightarrow & \begin{bmatrix}5.2 & 0.4\end{bmatrix}\end{array}$$

Zero ratings in common

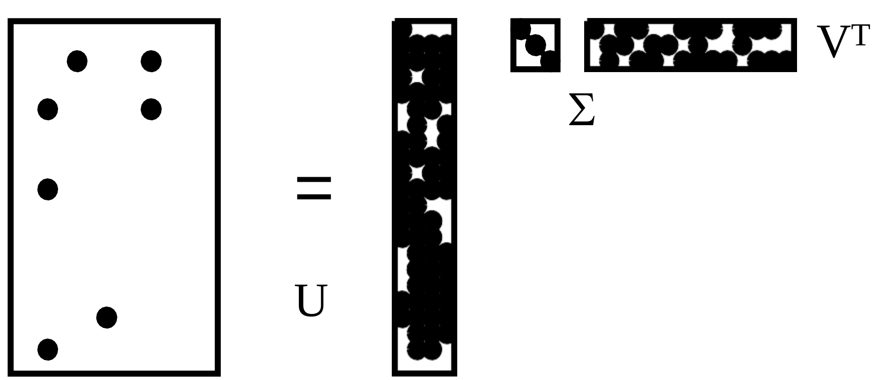
Similarity ≠ 0

SVD: Drawbacks

Optimal low-rank approximation

in terms of Frobenius norm

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



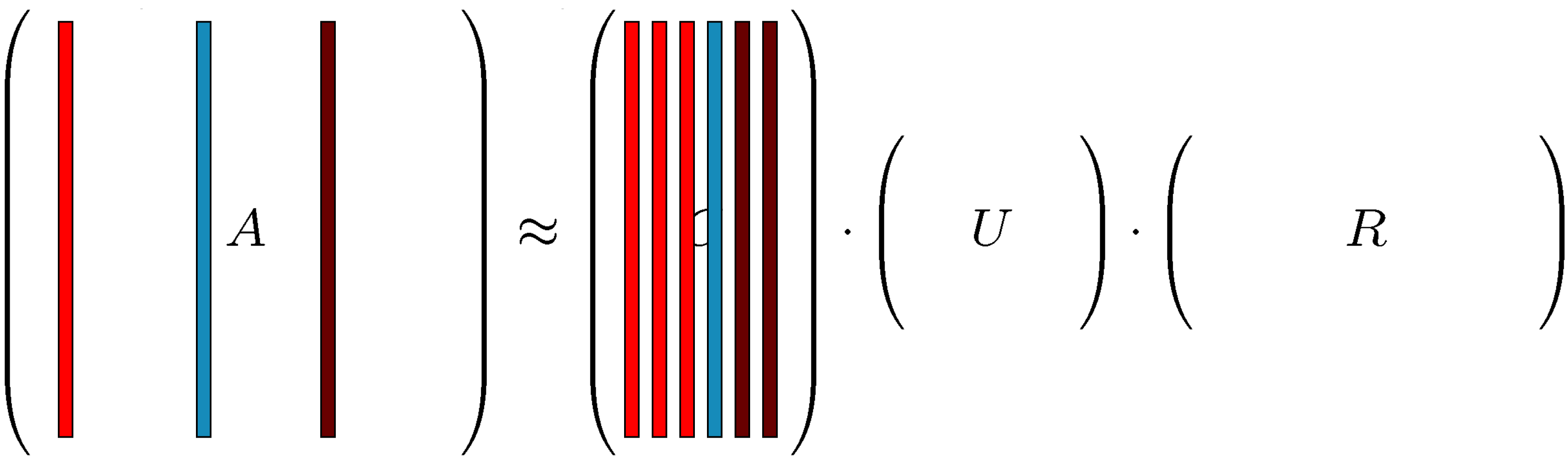
CUR Decomposition

CUR Decomposition

Frobenius norm:

$\|X\|_F = \Sigma_{ij}\ X_{ij}\ \ ^2$

- **Goal: Express A as a product of matrices C, U, R Make $\|A-C\cdot U\cdot R\|_F$ small**



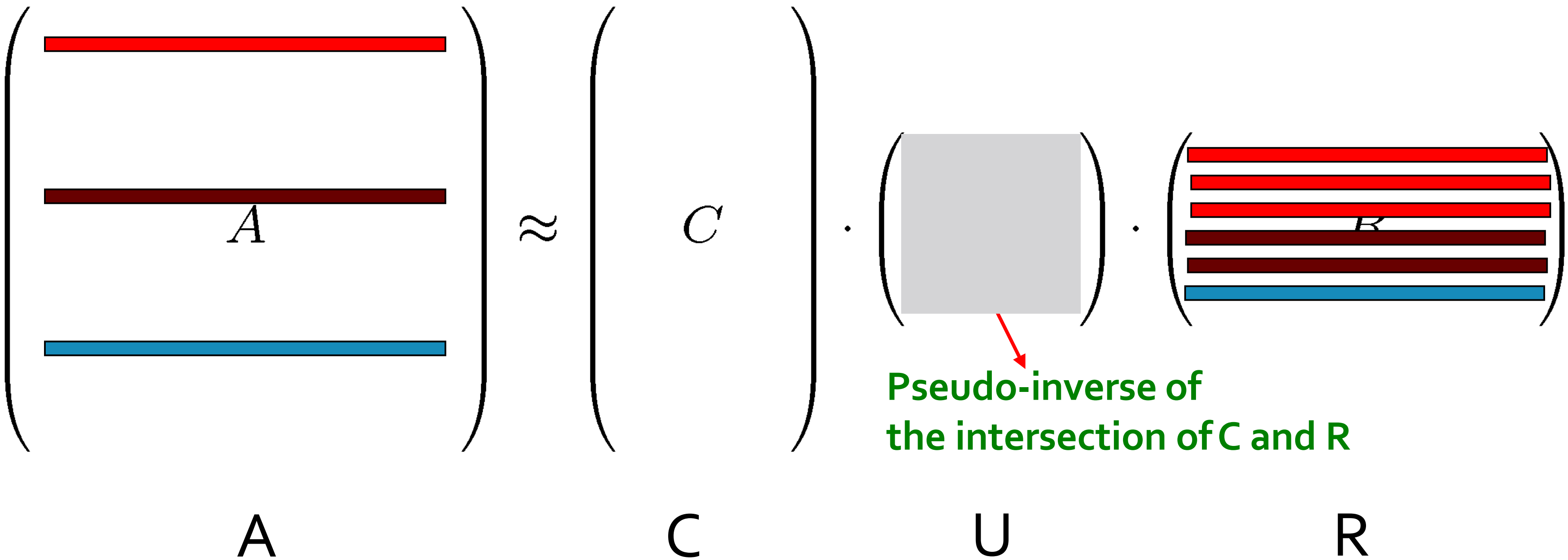
A C U R

CUR Decomposition

Frobenius norm:

$\|X\|_F = \Sigma_{ij} X_{ij}^2$

- **Goal: Express A as a product of matrices C,U,R Make $\|A-C\cdot U\cdot R\|_F$ small**
- **"Constraints" on C and R:**



CUR: Provably good approx.to SVD

- **Let:**

A_k be the "best" rank ***k*** approximation to ***A*** (that is, A_k is SVD of *A*)

<u>Theorem</u> [Drineas et al.]

CUR in O(**m·n**) time achieves

- $\|\mathbf{A-CUR}\|_F \leq \|\mathbf{A-A_k}\|_F + \varepsilon \|\mathbf{A}\|_F$

with probability at least **1-δ** by picking

- **O(k log(1/δ) /ε²)** columns, and
  - **O(k² log³(1/δ)/ε⁶)** rows

In practice:
Pick 4*k* cols/rows

CUR: How it Works

- **Sampling columns (similarly for rows):**

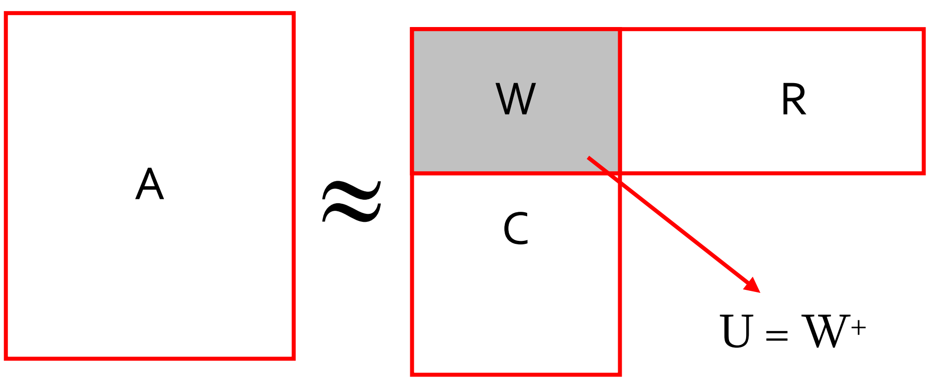
Input: matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, sample size c

Output: $\mathbf{C}_d \in \mathbb{R}^{m \times c}$

1. for $x = 1 : n$ [column distribution]
2. $P(x) = \sum_i \mathbf{A}(i,x)^2 / \sum_{i,j} \mathbf{A}(i,j)^2$
3. for $i = 1 : c$ [sample columns]
4. Pick $j \in 1 : n$ based on distribution $P(x)$
5. Compute $\mathbf{C}_d(:,i) = \mathbf{A}(:,j) / \sqrt{cP(j)}$

Computing U

- Let **W** be the "intersection" of sampled columns **C** and rows **R**
  - Let SVD of **W** = **X Z Y**^{T}
- **Then: U = W$^+$ = Y Z$^+$ X^T**
  - Z^+: **reciprocals of non-zero singular values:** $Z^+_{ii} = 1/Z_{ii}$
  - W$^+$ is the "**pseudoinverse**"



Why pseudoinverse works?
W = X Z Y then W^{-1} = X^{-1} Z^{-1} Y^{-1}
Due to orthonomality X^{-1}=X^T and Y^{-1}=Y^T
Since Z is diagonal Z = 1/Z$_{ii}$
Thus, if **W** is nonsingular, pseudoinverse is the true inverse

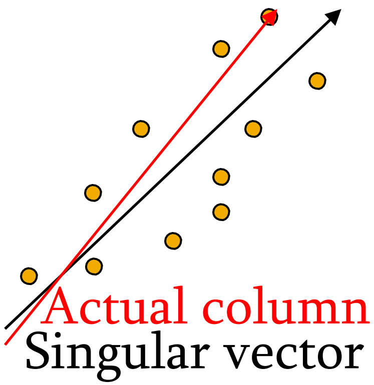
CUR: Pros&Cons

+ Easy interpretation

- Since the basis vectors are actual columns and rows

+ **Sparse basis**

- Since the basis vectors are actual columns and rows

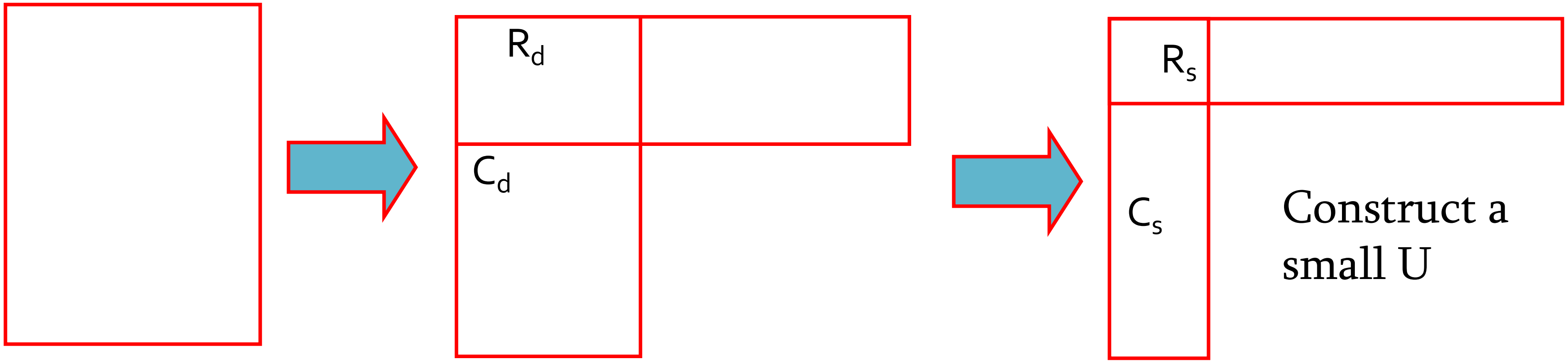


– **Duplicate columns and rows**

- Columns of large norms will be sampled many times

Solution

- **If we want to get rid of the duplicates:**
  - Throw them away
  - Scale (multiply) the columns/rows by the square root of the number of duplicates



SVD VS. CUR

sparse and small

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

Huge but sparse

Big and dense

dense but small

CUR: $A = C U R$

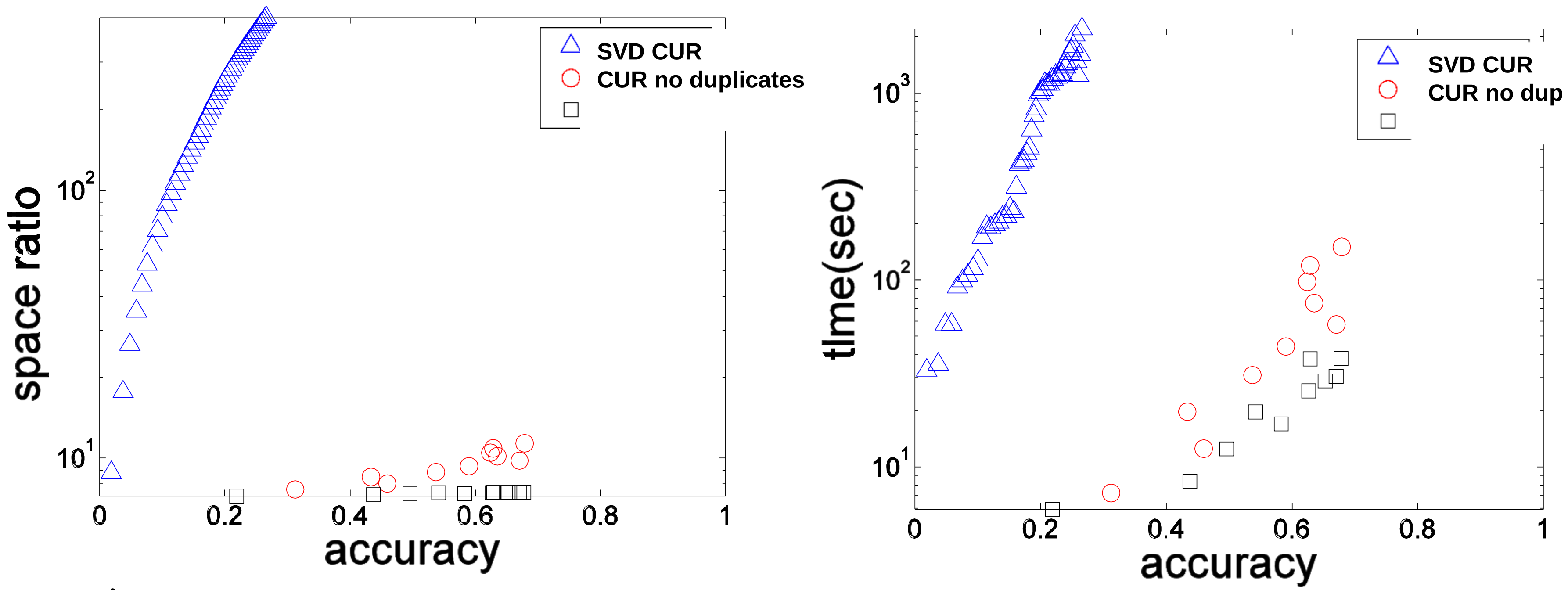
Huge but sparse

Big but sparse

Simple Experiment

- **DBLP bibliographic data**
  - Author-to-conference big sparse matrix
  - A_{ij}: Number of papers published by author *i* at conference *j*
  - 428K authors (rows), 3659 conferences (columns)
    - Very sparse
  - **Want to reduce dimensionality**
    - How much time does it take?
    - What is the reconstruction error?
    - How much space do we need?

Results：DBLP-big sparse matrix

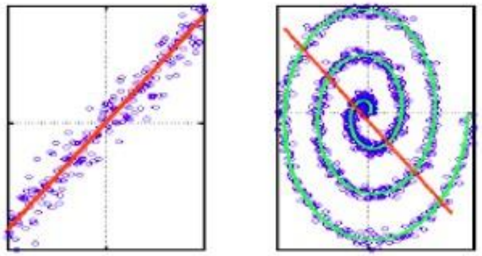


- **Accuracy:**
  - 1 – relative sum squared errors
- **Space ratio:**
  - #output matrix entries / #input matrix entries
- **CPU time**

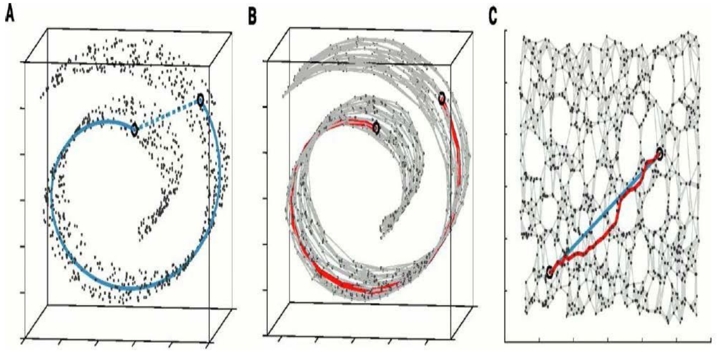
Sun, Faloutsos: *Less is More: Compact Matrix Decomposition for Large Sparse Graphs*, SDM '07.

What about linearity assumption?

- **SVD is limited to linear projections:**
  - Lower-dimensional linear projection that preserves Euclidean distances



  - Non-linear methods: **Isomap**
  - Data lies on a nonlinear low-dim curve aka manifold
    - Use the distance as measured along the manifold
  - **How?**
    - Build adjacency graph
    - Geodesic distance is graph distance
    - SVD/PCA the graph pairwise distance matrix



Further Reading: CUR

- Drineas et al., *Fast Monte Carlo Algorithms for Matrices III: Computing a Compressed Approximate Matrix Decomposition*, SIAM Journal on Computing, 2006.
- J. Sun, Y. Xie, H. Zhang, C. Faloutsos: *Less is More: Compact Matrix Decomposition for Large Sparse Graphs*, SDM 2007
- *Intra- and interpopulation genotype reconstruction from tagging SNPs*, P. Paschou, M. W. Mahoney, A. Javed, J. R. Kidd, A. J. Pakstis, S. Gu, K. K. Kidd, and P. Drineas, Genome Research, 17(1), 96-107 (2007)
- *Tensor-CUR Decompositions For Tensor-Based Data*, M. W. Mahoney, M. Maggioni, and P. Drineas, Proc. 12-th Annual SIGKDD, 327-336 (2006)

Tell me and I forget.
Show me and I remember.
Involve me and I understand.

Thank you!　Q&A