

# Dimensionality Reduction: SVD

Mining of Massive Datasets

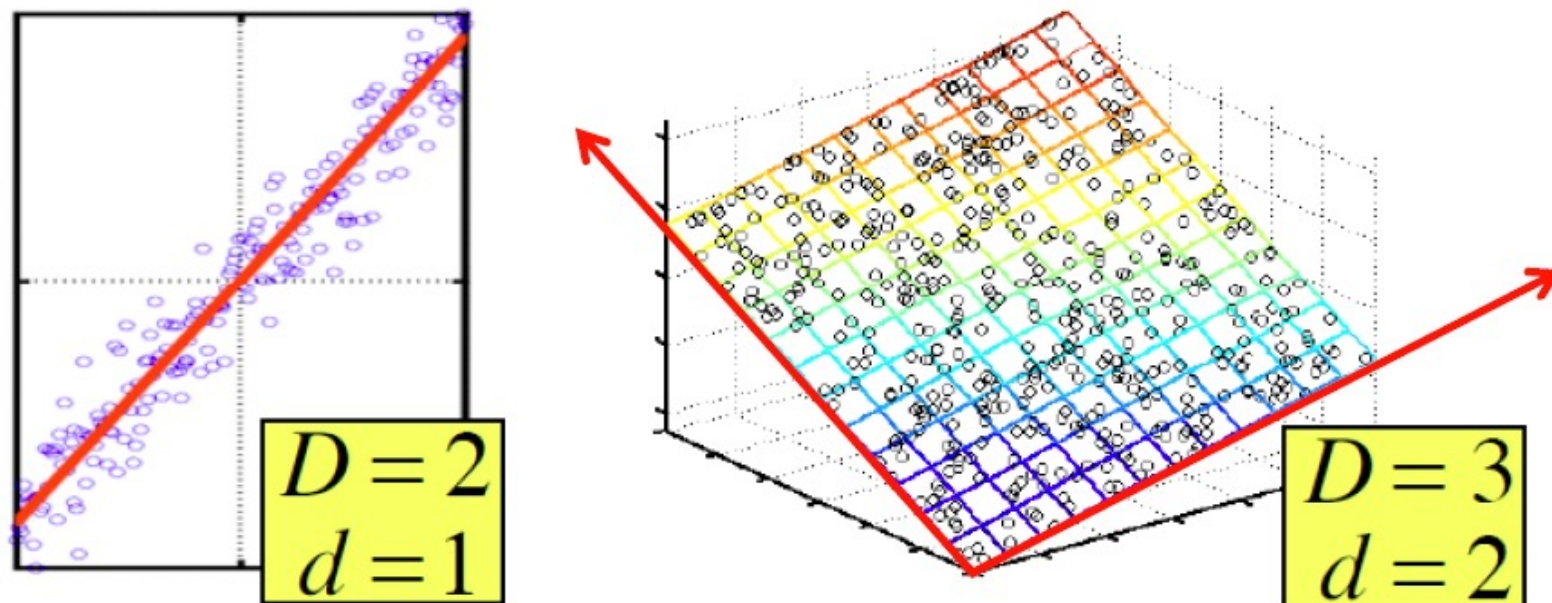
Jure Leskovec, Anand Rajaraman, Jeff Ullman

Stanford University

<http://www.mmnds.org>



# Dimensionality Reduction



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

# Dimensionality Reduction

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

day	We	Th	Fr	Sa	Su
customer	7/10/96	7/11/96	7/12/96	7/13/96	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  $[0 \ 0 \ 0 \ 1 \ 1]$

# Rank of a Matrix

- **Q:** What is **rank** of a matrix **A**?
- **A:** Number of **linearly independent** columns of **A**
- **For example:**
  - Matrix  $\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$  has rank **r=2**
    - **Why?** The first two rows are linearly independent, 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 :  $[1 \ 0] \ [0 \ 1] \ [1 \ 1]$

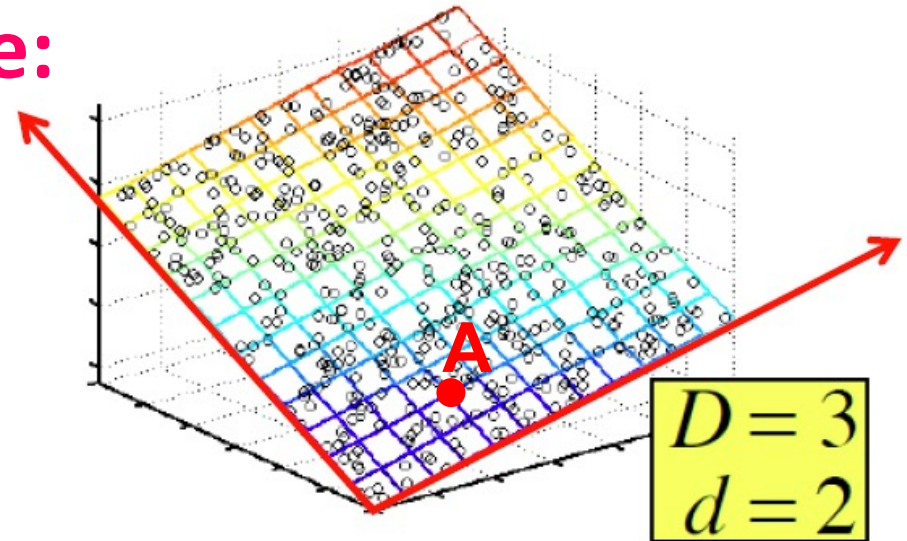
# Rank is “Dimensionality”

- **Cloud of points 3D space:**

- Think of point positions

as a matrix:  $\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$  **A**  
**B**  
**C**

1 row per point:

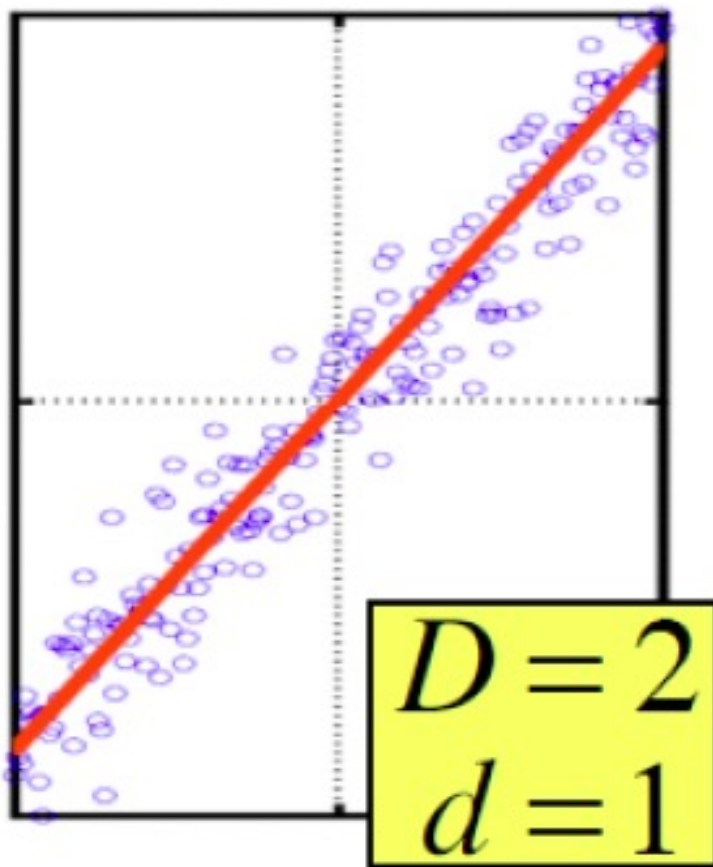


- **We can rewrite coordinates more efficiently!**

- 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!



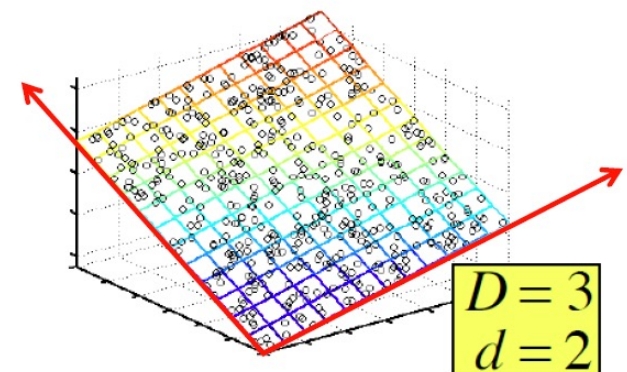
Rather than representing every point with 2 coordinates we represent each point with 1 coordinate (corresponding to the position of the point on the red line).

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?

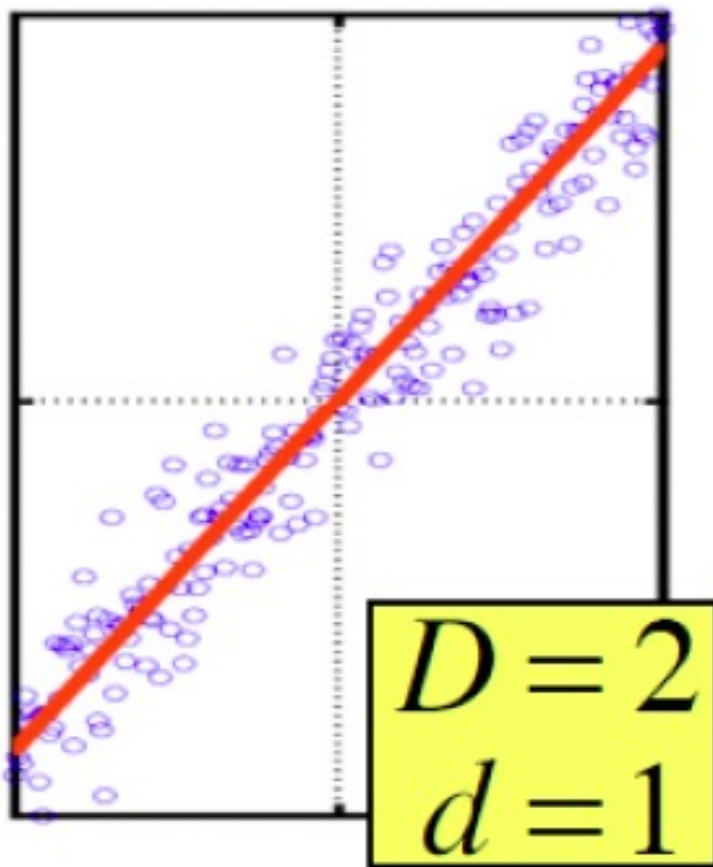
- Discover hidden correlations/topics
  - Words that occur commonly together
- Remove redundant and noisy features
  - Not all words are useful
- Interpretation and visualization
- Easier storage and processing of the data





# Dimensionality Reduction

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



How to find the axis?

- Singular value decomposition
  - All matrix dimensions
- Eigenvalue decomposition
  - Square matrix (covariance)



# SVD - Definition

$$\mathbf{A}_{[m \times n]} = \mathbf{U}_{[m \times r]} \mathbf{\Sigma}_{[r \times r]} (\mathbf{V}_{[n \times r]})^T$$

- **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)
- **$\Sigma$ : 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 - Properties

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

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

Nice proof of uniqueness: <http://www.mpi-inf.mpg.de/~bast/ir-seminar-wso4/lecture2.pdf>

# SVD – Example: Users-to-Movies

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

Matrix Alien Serenity Casablanca Amelie

SciFi ↑  
↓  
Romnce ↑  
↓

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

Each user is represented by 5 reviews

# SVD – Example: Users-to-Movies

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

Diagram illustrating the SVD decomposition of a Users-to-Movies matrix  $A$ .

The matrix  $A$  (Users to Movies) is shown as a 6x5 matrix. The rows are labeled with movie titles: Matrix, Alien, Serenity, Casablanca, Amelie. The columns are labeled with user IDs: 1, 2, 3, 4, 5. The matrix is annotated with green arrows indicating the SciFi and Romance genres for each user.

The matrix  $A$  is equal to the product of three matrices:

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

Where:

- $U$  (Users to Latent Space) is a 6x3 matrix. The columns are labeled "SciFi-concept" and "Romance-concept".
- $\Sigma$  (Latent Space) is a 3x3 diagonal matrix.
- $V^T$  (Latent Space to Movies) is a 3x5 matrix.

The resulting matrix  $A$  is shown below the decomposition:

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

Each user is represented by 5 reviews

# SVD – Example: Users-to-Movies

■  $A = U \Sigma V^T$  - example:

$U$  is “user-to-concept” similarity matrix

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

# SVD – Example: Users-to-Movies

## ■ $A = U \Sigma V^T$ - example:

Matrix   Alien   Serenity   Casablanca   Amelie

SciFi   ↑   ↓   ↑   ↓

Romnce

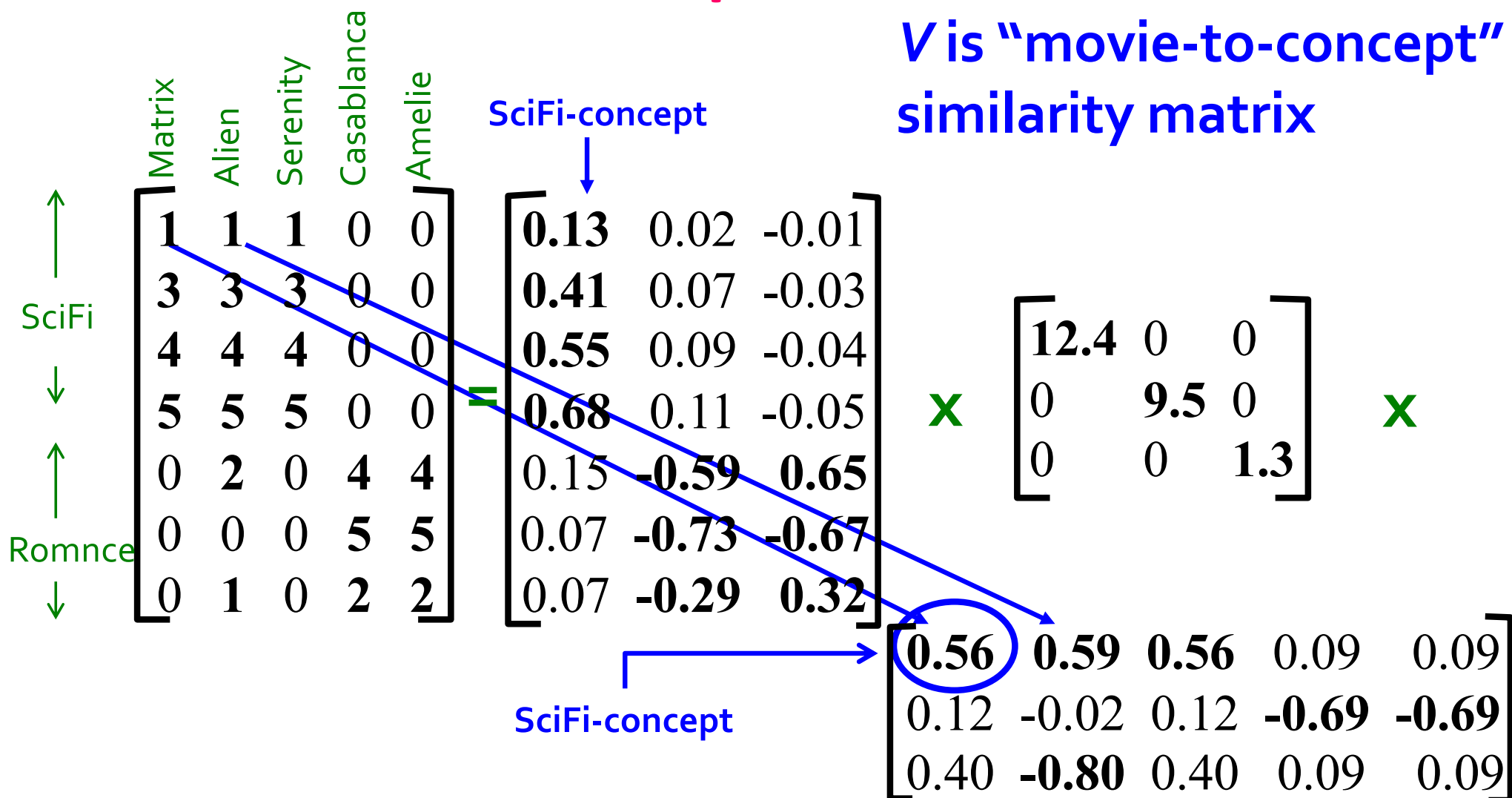
$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 0.13 & 0.02 & -0.01 \\ 0.41 & 0.07 & -0.03 \\ 0.55 & 0.09 & -0.04 \\ 0.68 & 0.11 & -0.05 \\ 0.15 & -0.59 &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

SciFi-concept

"strength" of the SciFi-concept

# SVD – Example: Users-to-Movies

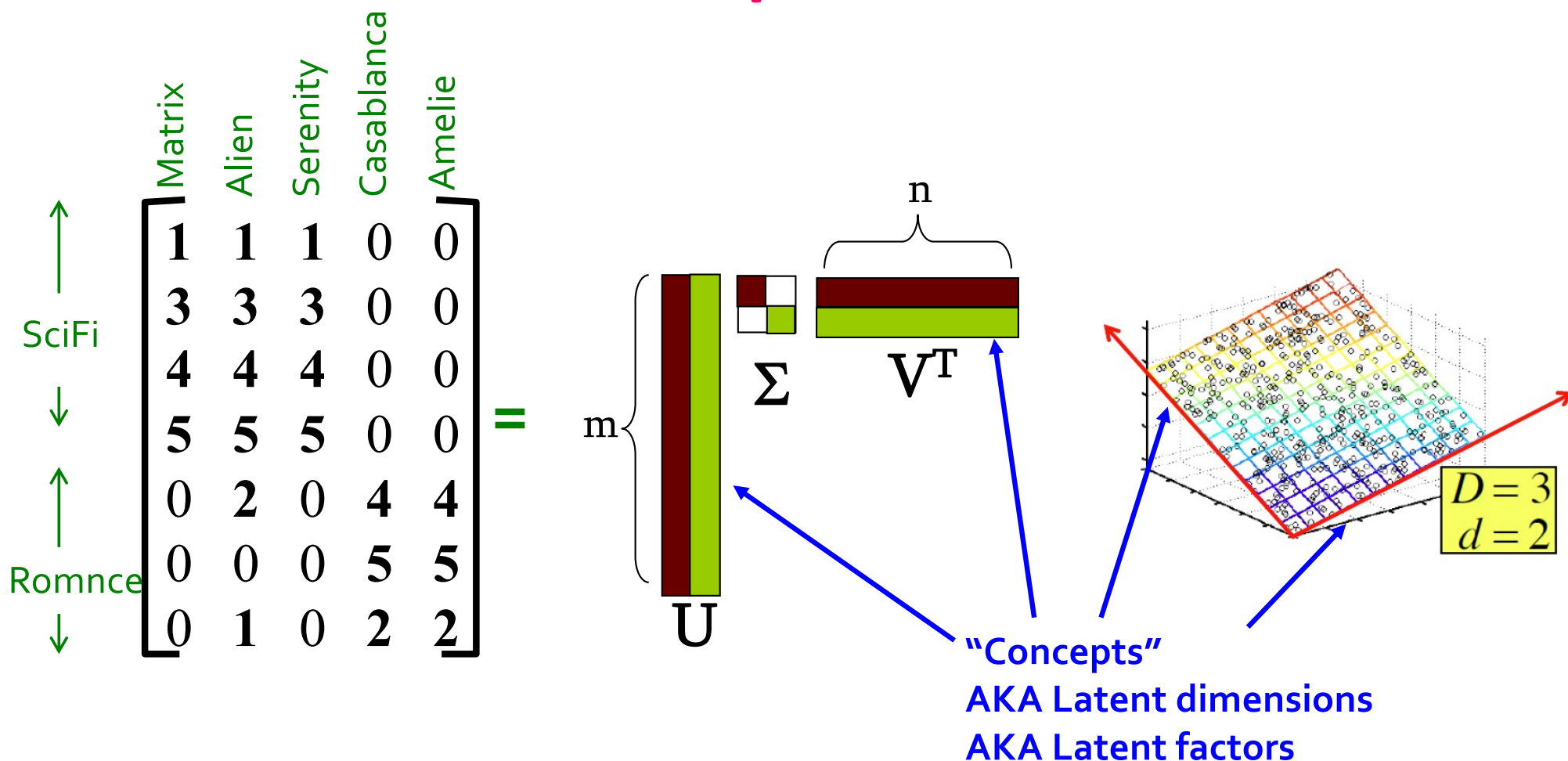
## ■ $A = U \Sigma V^T$ - example:





# SVD – Example: Users-to-Movies

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



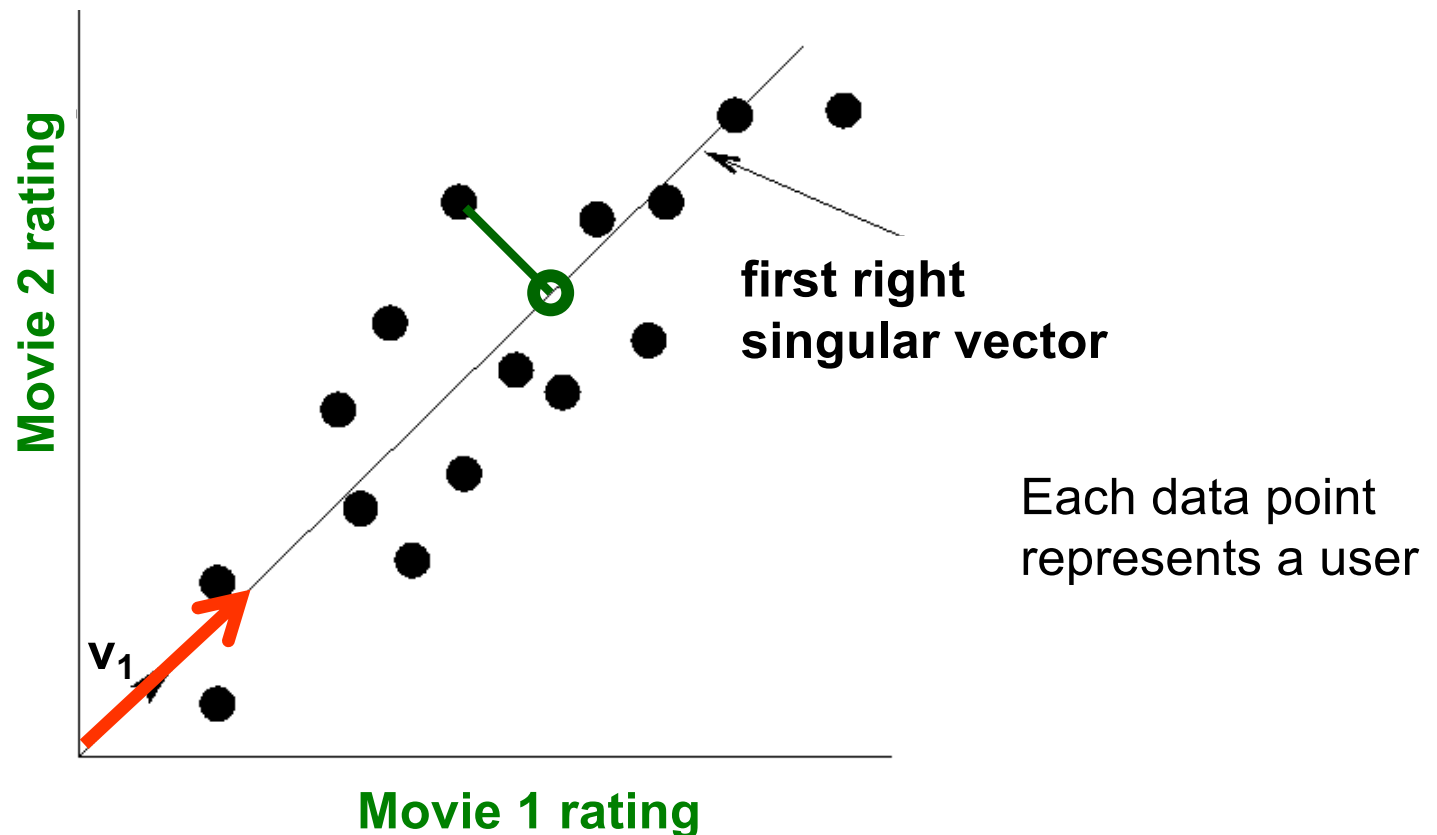
# SVD - Interpretation #1

‘**movies**’, ‘**users**’ and ‘**concepts**’:

- $U$ : user-to-concept similarity matrix
- $V$ : movie-to-concept similarity matrix
- $\Sigma$ : its diagonal elements:  
‘strength’ of each concept

# Dimensionality Reduction with SVD

# SVD – Dimensionality Reduction



- Instead of using two coordinates  $(x, y)$  to describe point locations, let's use only one coordinate  $(z)$
- Point's position is its location along vector  $v_1$
- **How to choose  $v_1$ ? Minimize reconstruction error**

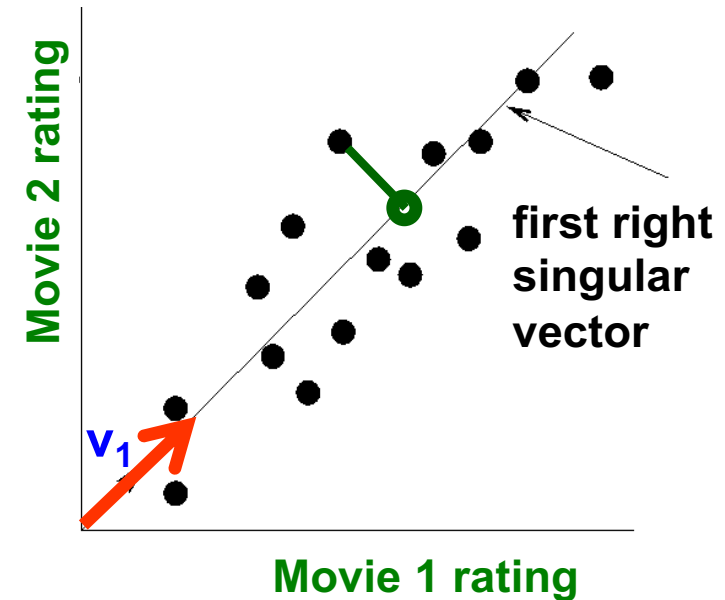
# SVD – Dimensionality Reduction

- **Goal:** Minimize the sum of reconstruction errors:

$$\sum_{i=1}^N \sum_{j=1}^D \|x_{ij} - z_{ij}\|^2$$

- where  $x_{ij}$  are the “old” and  $z_{ij}$  are the “new” coordinates

- **SVD gives ‘best’ axis to project on:**
  - ‘best’ = minimizing the reconstruction errors
- In other words, **minimum reconstruction error**



# SVD - Interpretation #2

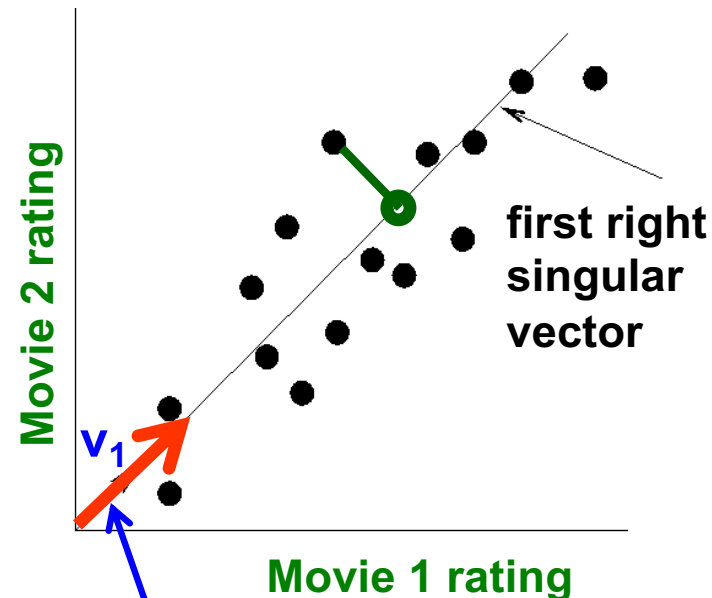
## ■ $A = U \Sigma V^T$ - example:

- $V$ : “movie-to-concept” matrix
- $U$ : “user-to-concept” matrix

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

$$\begin{bmatrix} 12.4 & 0 & 0 \\ 0 & 9.5 & 0 \\ 0 & 0 & 1.3 \end{bmatrix} \times$$

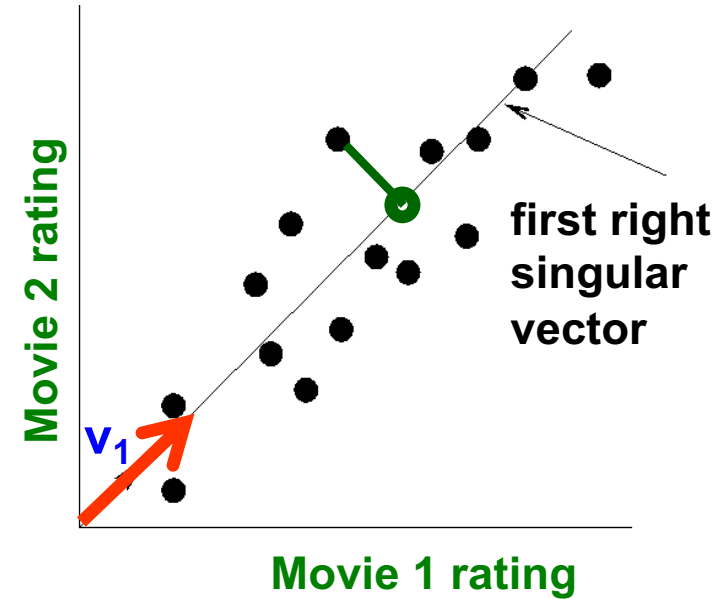
$$\begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \\ 0.40 & -0.80 & 0.40 & 0.09 & 0.09 \end{bmatrix}$$



# SVD - Interpretation #2

## ■ $A = U \Sigma V^T$ - example:

variance ('spread')  
on the  $v_1$  axis



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



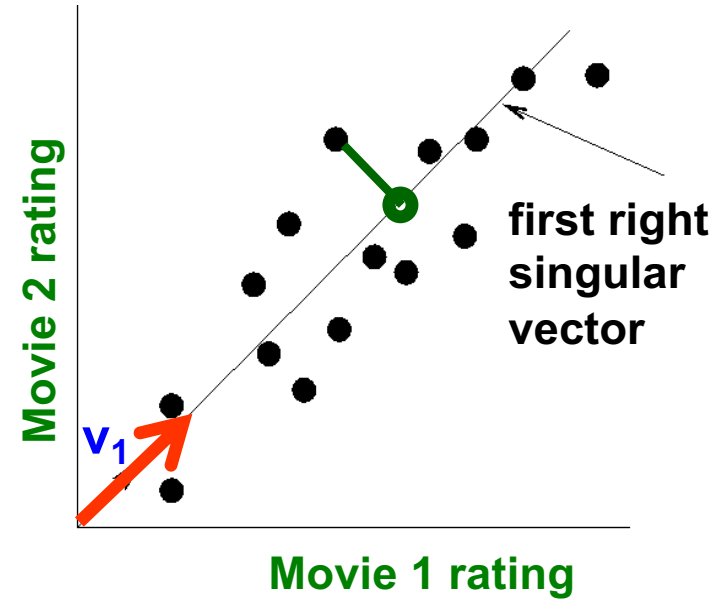
# SVD - Interpretation #2

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

- **$U \Sigma$ :** Gives the coordinates of the points in the projection axis

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

**Projection of users  
on the “Sci-Fi” axis  
( $U \Sigma$ )<sup>T</sup>:**



1.61	0.19	-0.01
5.08	0.66	-0.03
6.82	0.85	-0.05
8.43	1.04	-0.06
1.86	-5.60	0.84
0.86	-6.93	-0.87
0.86	-2.75	0.41

# SVD - Interpretation #2

## More details

- **Q:** How exactly is dim. reduction done?

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

# SVD - Interpretation #2

## More details

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

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

# SVD - Interpretation #2

## More details

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

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

# SVD - Interpretation #2

## More details

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

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

The diagram illustrates the process of dimensionality reduction using SVD. The first matrix (5x7) is approximated by the product of three matrices: a 5x3 matrix of singular vectors, a 3x3 matrix of singular values, and a 3x7 matrix of right singular vectors. The singular values matrix shows the singular values 12.4, 9.5, and 1.3. The right singular vectors matrix shows the vectors corresponding to these singular values. The singular value 1.3 is the smallest, and its corresponding vector is crossed out with a red line, indicating it is the one to be set to zero for dimensionality reduction.

# SVD - Interpretation #2

## More details

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

A user is represented by 5 ratings

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

$\approx$

$$\begin{bmatrix} 0.13 & 0.02 \\ 0.41 & 0.07 \\ 0.55 & 0.09 \\ 0.68 & 0.11 \\ 0.15 & -0.59 \\ 0.07 & -0.73 \\ 0.07 & -0.29 \end{bmatrix}$$

Two Axis: Sci-Fi and Romance

Projection: A user is represented by 2 concepts

$\times$

$$\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}$$

$\times$

$$\begin{bmatrix} 0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & -0.69 & -0.69 \end{bmatrix}$$

# Case study: How to query?

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

Diagram illustrating the mapping of a query (Matrix) into a concept space for finding similar users.

**Matrix (Query) and User Ratings:**

	Matrix	Alien	Serenity	Casablanca	Amelie
SciFi	1	1	1	0	0
	3	3	3	0	0
	4	4	4	0	0
	5	5	5	0	0
Romnce	0	2	0	4	4
	0	0	0	5	5
	0	1	0	2	2

**Concept Space Mapping:**

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



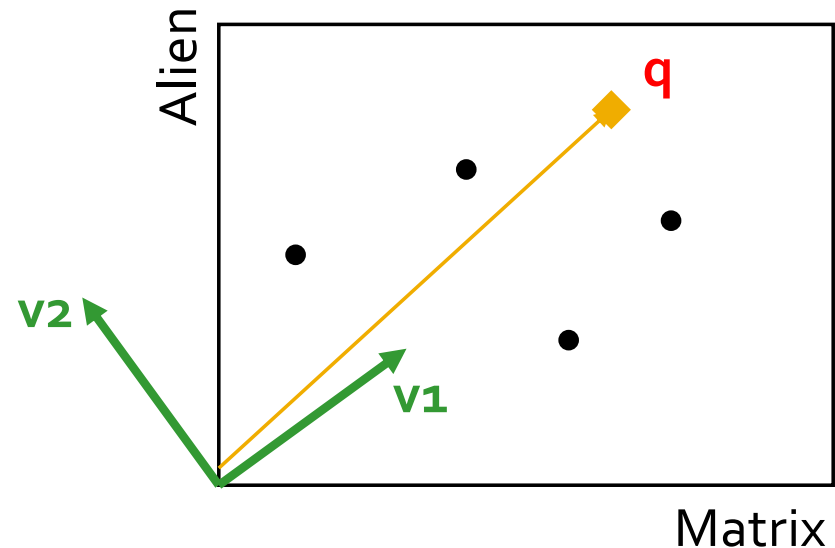
# Case study: How to query?

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

This query also represents a user who likes Matrix, and the goal is to find users with a similar taste

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

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



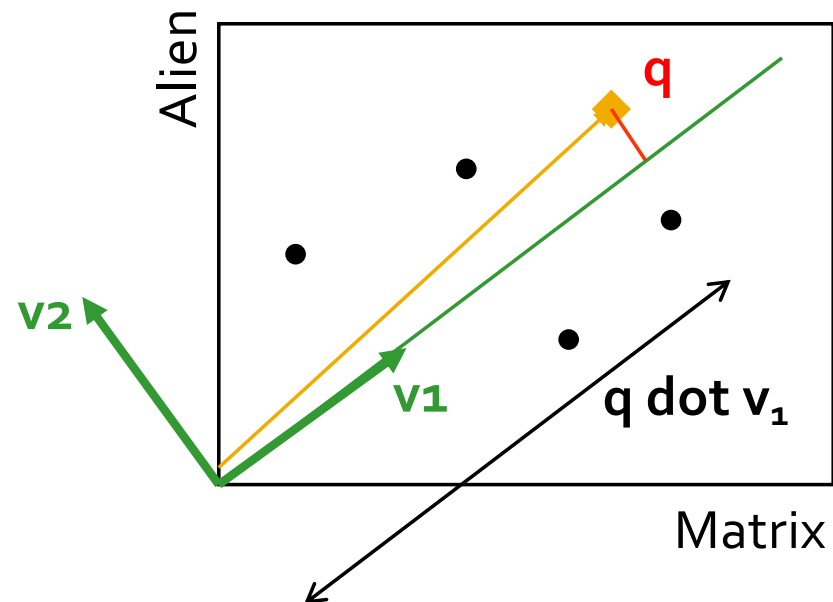
V1 and V2 are concepts

# Case study: How to query?

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

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

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



# Case study: How to query?

Compactly, we have:

$$q_{\text{concept}} = q V$$

E.g.:

$$q = \begin{matrix} & \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \begin{matrix} \text{SciFi} \\ \text{Fantasy} \end{matrix} & \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix} \times \begin{matrix} \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} \\ \text{movie-to-concept} \\ \text{similarities (V)} \end{matrix} = \begin{matrix} \text{SciFi-concept} \\ \downarrow \\ \begin{bmatrix} 2.8 & 0.6 \end{bmatrix} \end{matrix}$$

# Case study: How to query?

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

$$\mathbf{d}_{\text{concept}} = \mathbf{d} \mathbf{V}$$

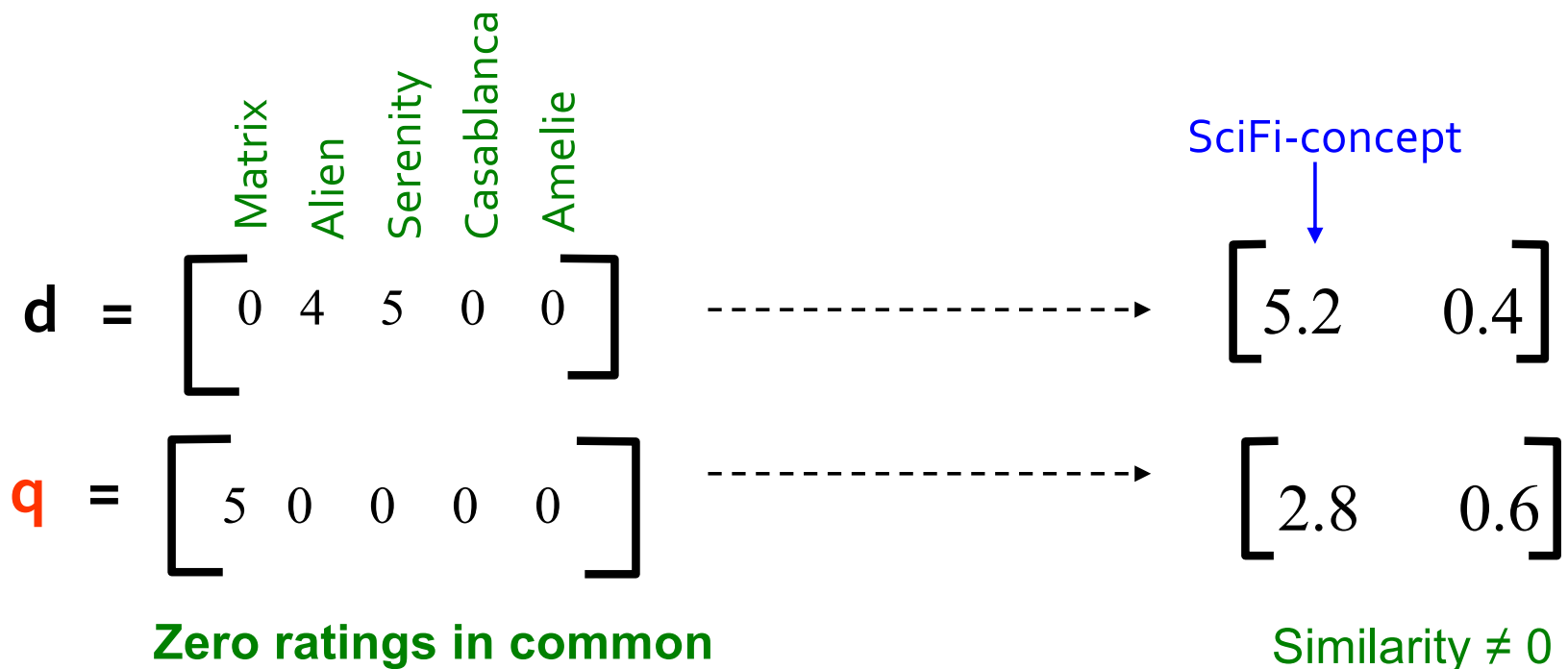
E.g.:

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

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!**



# 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!**

The diagram illustrates the SVD decomposition of a matrix. On the left is a sparse matrix represented by a rectangle containing 7 black dots. This is followed by an equals sign. To the right of the equals sign is the product of three matrices: a tall, dense matrix labeled  $U$  (filled with black dots), a small square diagonal matrix labeled  $\Sigma$  (with a few black dots on its main diagonal), and a wide, dense matrix labeled  $V^T$  (filled with black dots).

# 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

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