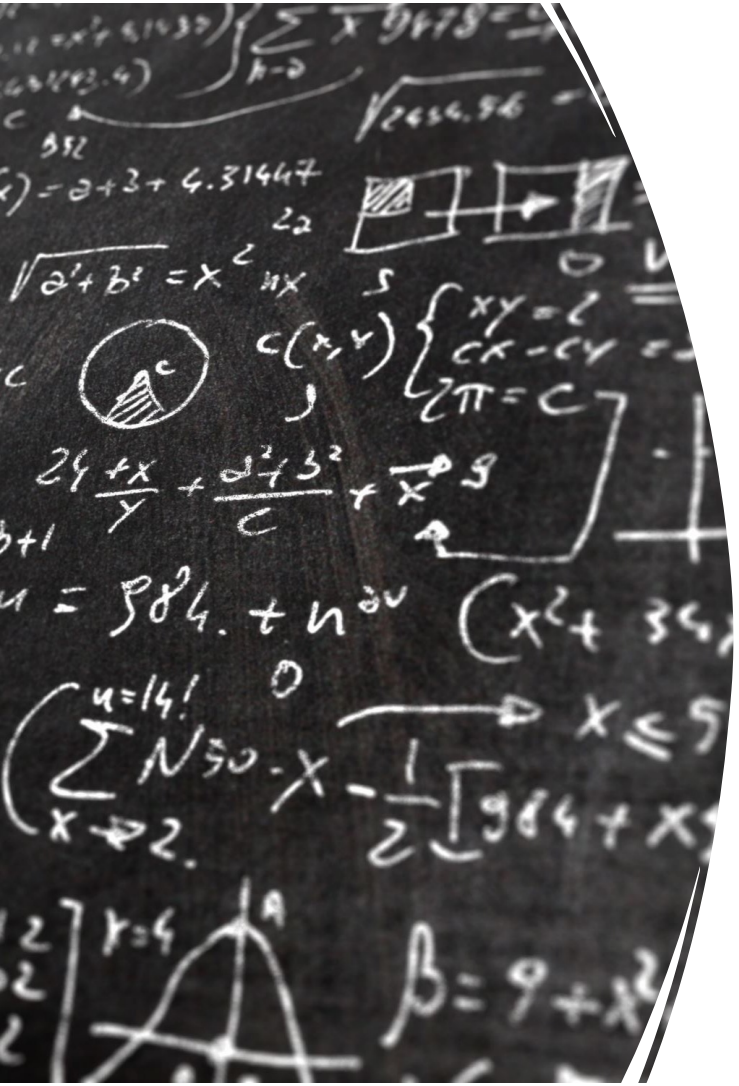


CS 383 – Computational Text Analysis

Lecture 4
Document Representation, Matrix Factorization

Adam Poliak

01/30/2023

Slides adapted from Dan Jurafsky, Dirk Hovy, Jure Leskovec

Announcements (1/2)

- Office Hours:
  - This week: Thursday 3:30-4:30pm
- HW01 due tonight Monday 01/30
  - Based on Monday's lecture
- HW02 released tonight, due Monday 02/06

Outline

- Document Representations - recap
- tf-idf
- Linear Algebra:
  - Matrix multiplication
  - Matrix factorization
    - SVD
- Latent Semantic Analysis
- PCA

Recap so far

The first class was all about counting words

2nd class was about the power of counting words.

By counting words we can ___ ___

learn about language

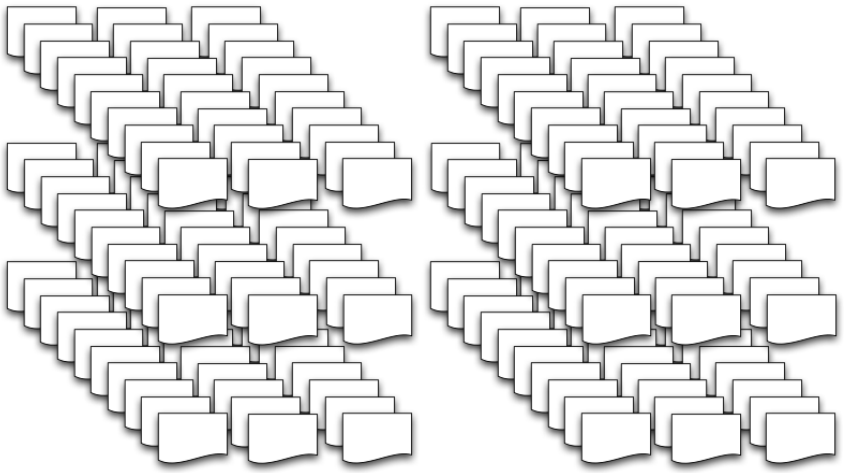
generate language

categorize language

represent documents as vectors

Terminology

- **Corpus**:
  - A collection of documents
  - ***Corpora*** – plural of corpus



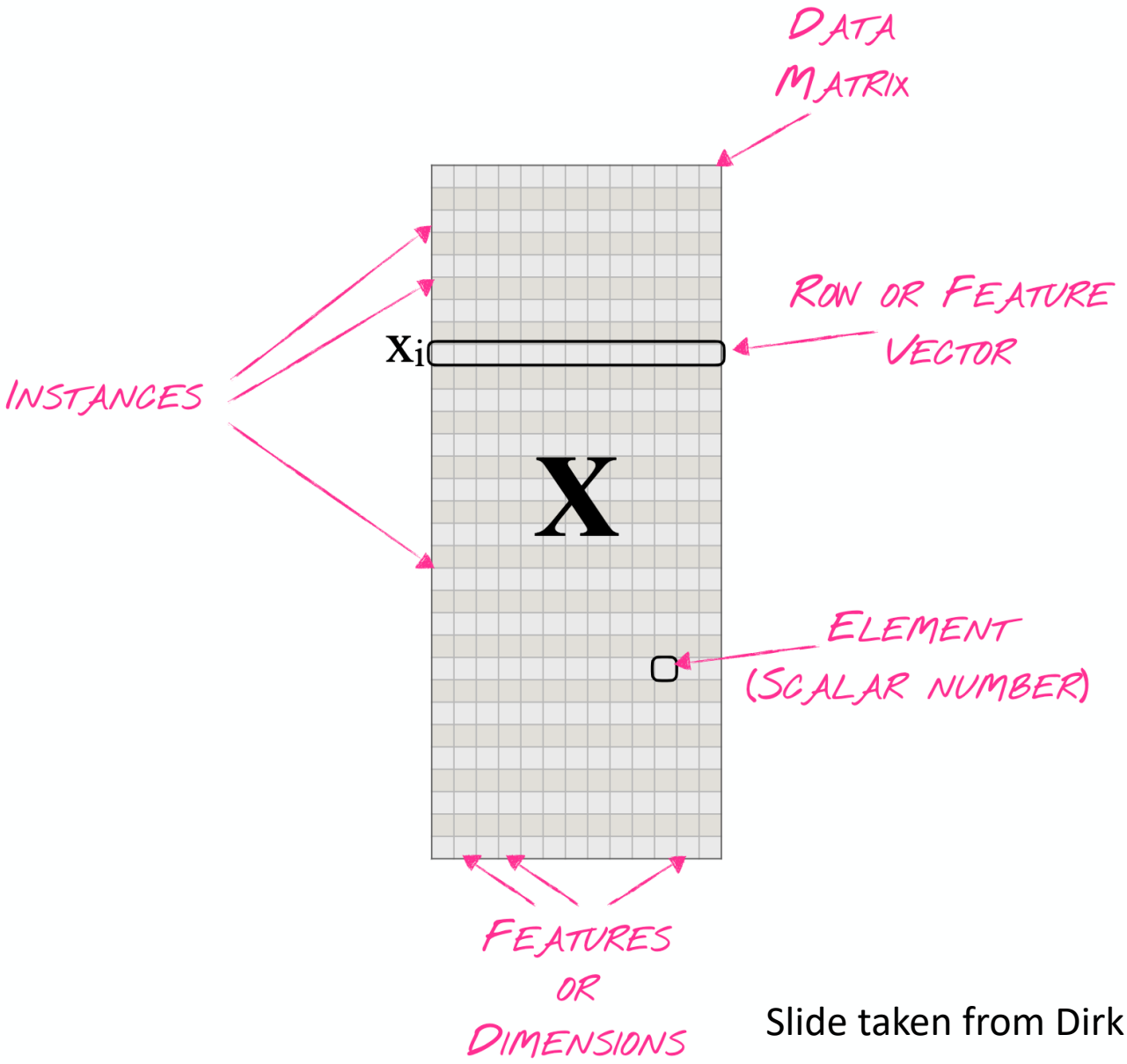
- **Document**:
  - Often unit of text of interest (dependent on RQ)
  - Often represents one data point

Bag of Words

Very good drama although it appeared to have a few blank areas leaving the viewers to fill in the action for themselves. I can imagine life being this way for someone who can neither read nor write. This film simply smacked of the real world: the wife who is suddenly the sole supporter, the live-in relatives and their quarrels, the troubled child who gets knocked up and then, typically, drops out of school, a jackass husband who takes the nest egg and buys beer with it. 2 thumbs up… very very very good movie.

('the', 8),
(',', 5),
('very', 4),
('.', 4),
('who', 4),
('and', 3),
('good', 2),
('it', 2),
('to', 2),
('a', 2),
('for', 2),
('can', 2),
('this', 2),
('of', 2),
('drama', 1),
('although', 1),
('appeared', 1),
('have', 1),
('few', 1),
('blank', 1)
…..

Document Matrix



Slide taken from Dirk Hovy

Outline

- Document Representations - recap
- tf-idf
- Linear Algebra:
  - Matrix multiplication
  - Matrix factorization
    - SVD
- Latent Semantic Analysis
- PCA

What to count?
How to count?

Next lecture

HW02

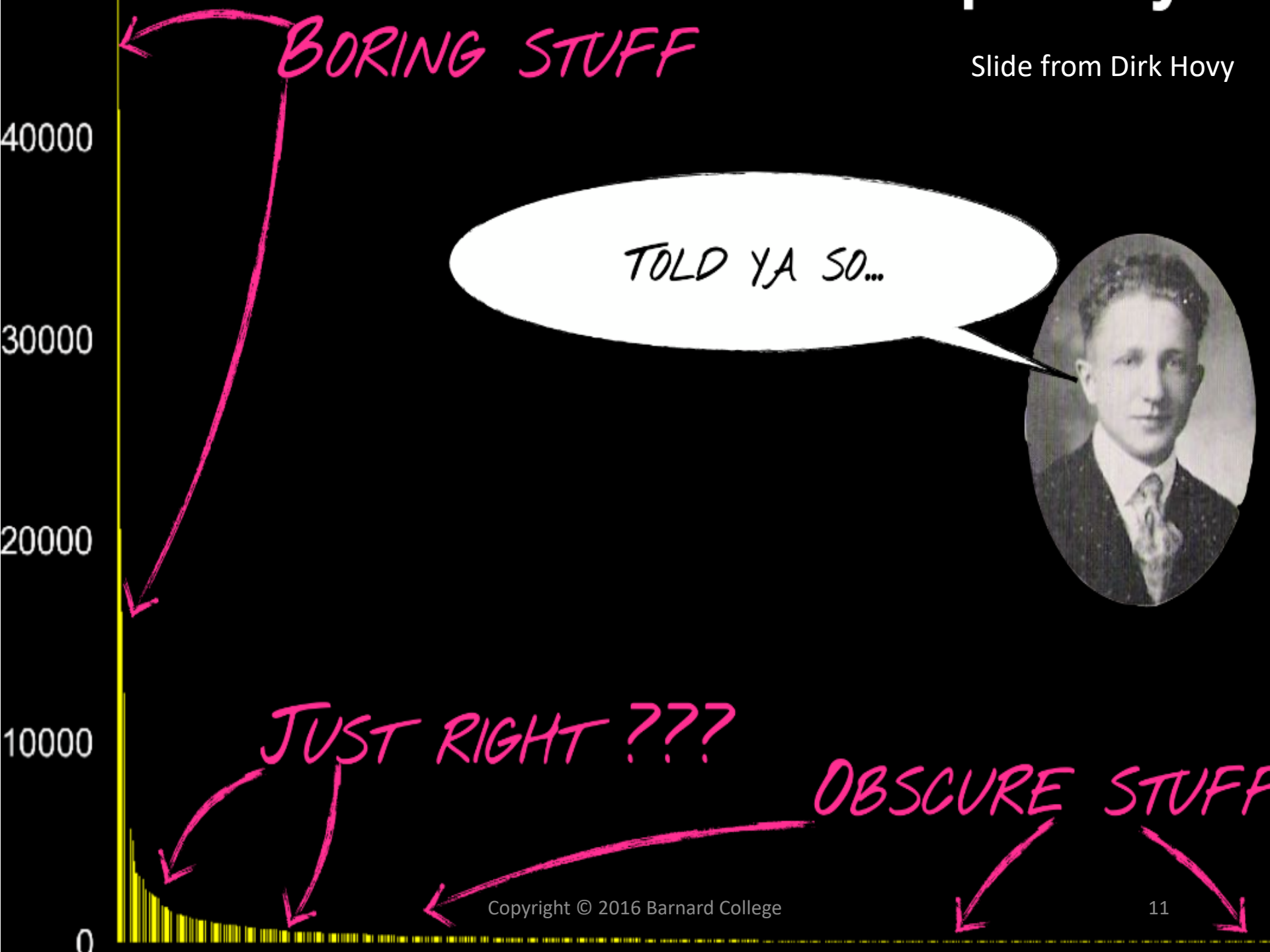
Term Frequency (tf):

tf of word ***w*** in document ***d***:

$$\frac{|w|}{|\,Document\,|}$$

*number of times **w** appears in **D***

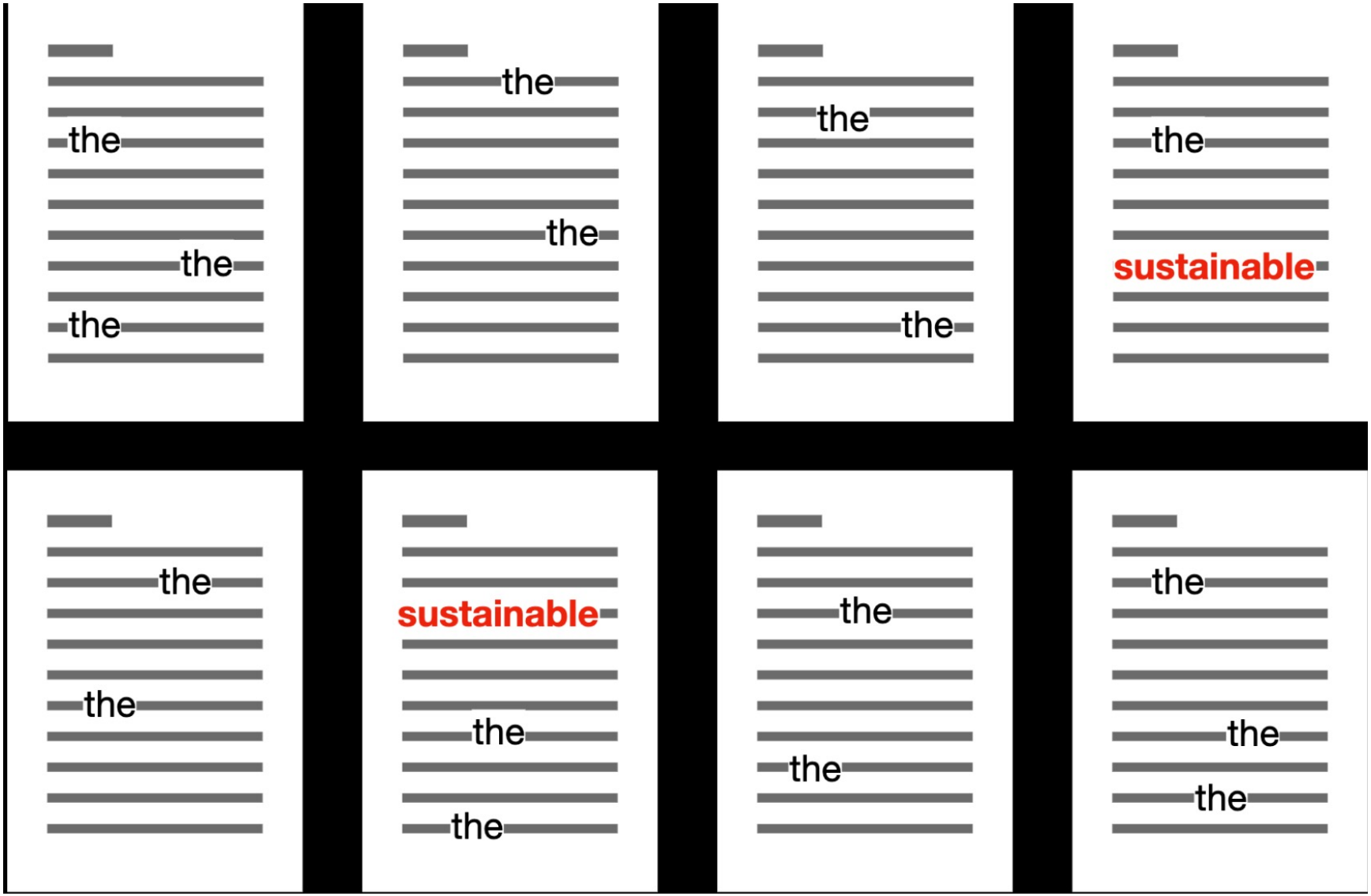
*divided by of number tokens in **D***



Slide from Dirk Hovy

Inverse Document Frequency

Some words are more interesting



Slide taken from Dirk Hovy

Inverse Document Frequency (idf)

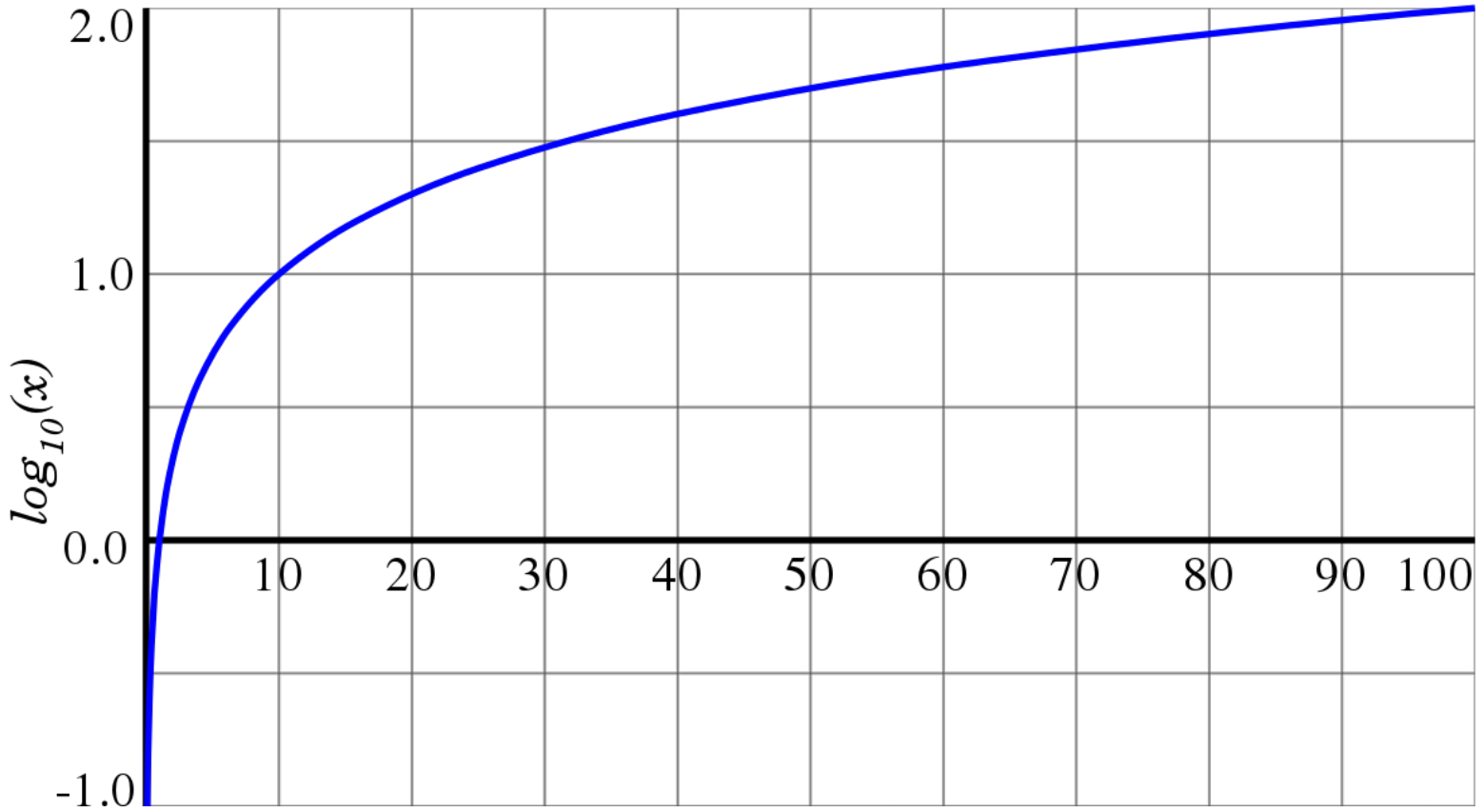
idf of word ***w*** in document ***D***:

$$\log \frac{|D|}{|\ tf(w,d) \neq 0\ |}$$

number of documents divided by number of documents that contain ***w***

Scaling down IDF

log function is a way to scale down idf



TF-IDF

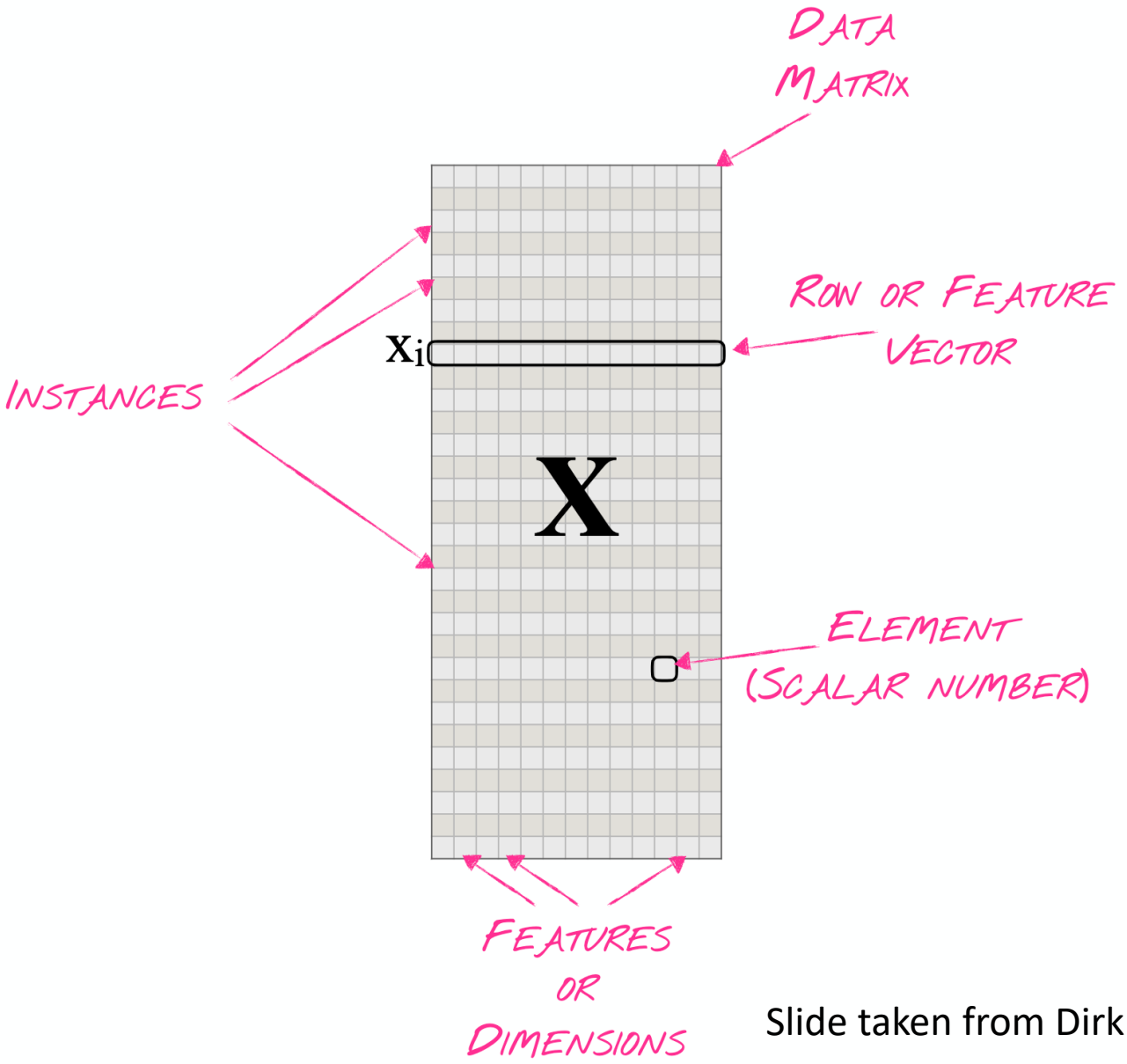
TF-IDF:
Term Frequency - Inverse Document Frequency

TF-IDF of word ***w*** in document ***D***:

Term Frequency * Inverse Document Frequency

Captures terms that are frequent in a document and specific to the document in the corpus

Document Matrix



Slide taken from Dirk Hovy

Document Matrix Characteristics

- Sparse:
  - Most common value is 0
- Size:
  - |D| x |V|
  - Vocab can be very large, especially if we use n-grams
- Values are positivie

Outline

- Document Representations - recap
- tf-idf
- Dimensionality Reduction
  - Linear Algebra Review:
    - Matrix multiplication
  - Matrix factorization
    - SVD
- Latent Semantic Analysis
- PCA

Matrix Multiplication

We can multiply two matrices A and B if ….

number of columns in A = number of rows in B

The size of the resulting matrix is ….

number of rows in A & the number of columns in B

Khan Academy

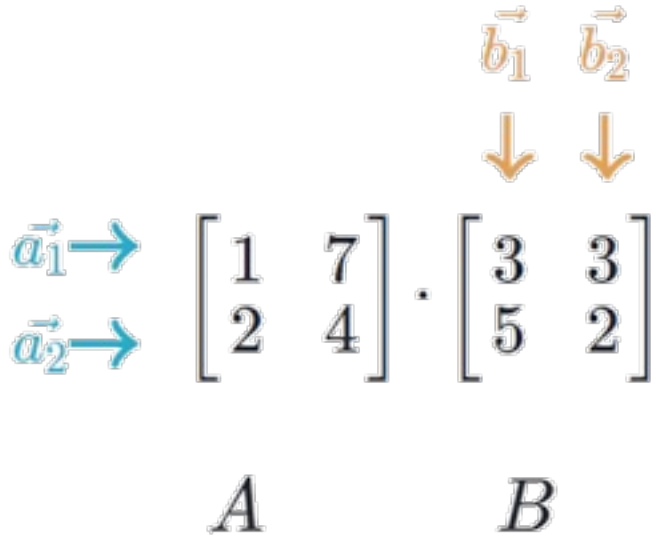
Matrix Multiplication

$$\begin{bmatrix} 1 & 7 \\ 2 & 4 \end{bmatrix} \cdot \begin{bmatrix} 3 & 3 \\ 5 & 2 \end{bmatrix}$$

A B

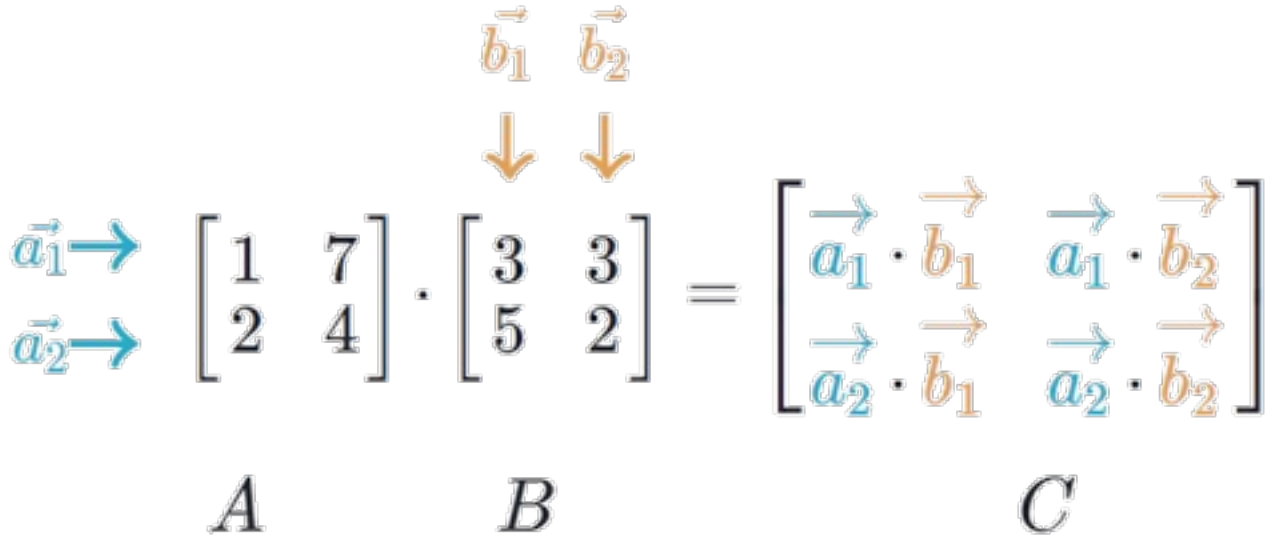
Khan Academy

Matrix Multiplication



Khan Academy

Matrix Multiplication



Khan Academy

Rank of a Matrix

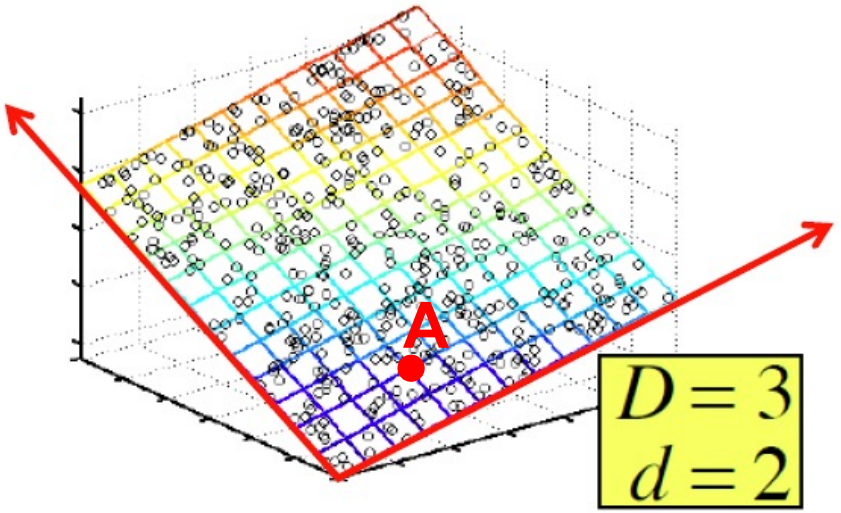
- **Q:** What is **rank** of a matrix **A**?
- **A:** Number of **linearly independent** columns of **A**
- **For example:**
  - Matrix **A** = $\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$ has rank **r=2**
    - **Why?** The first two rows are linearly independent, so the rank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

1 row per point:

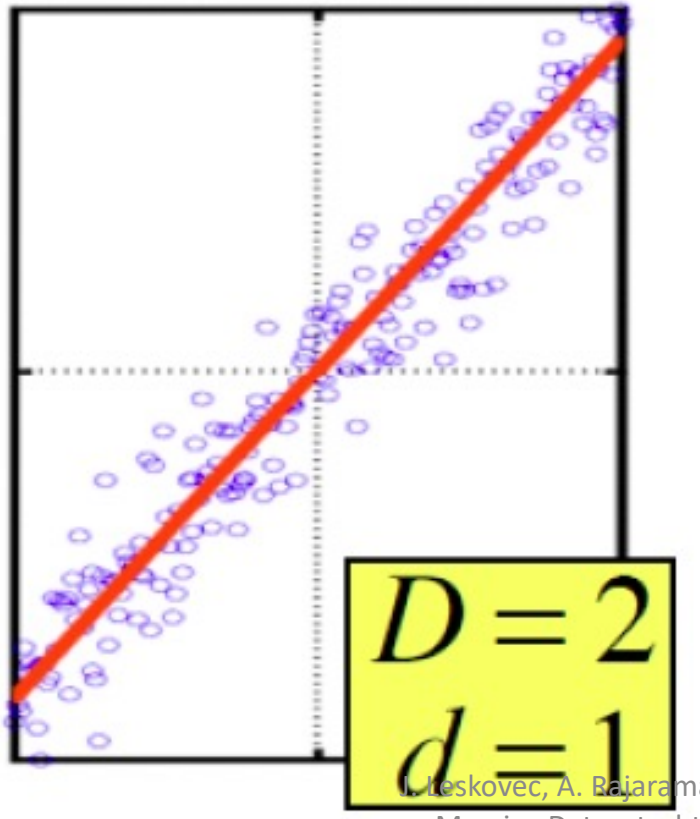
$$\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix} \begin{matrix} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{matrix}$$



- **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!**

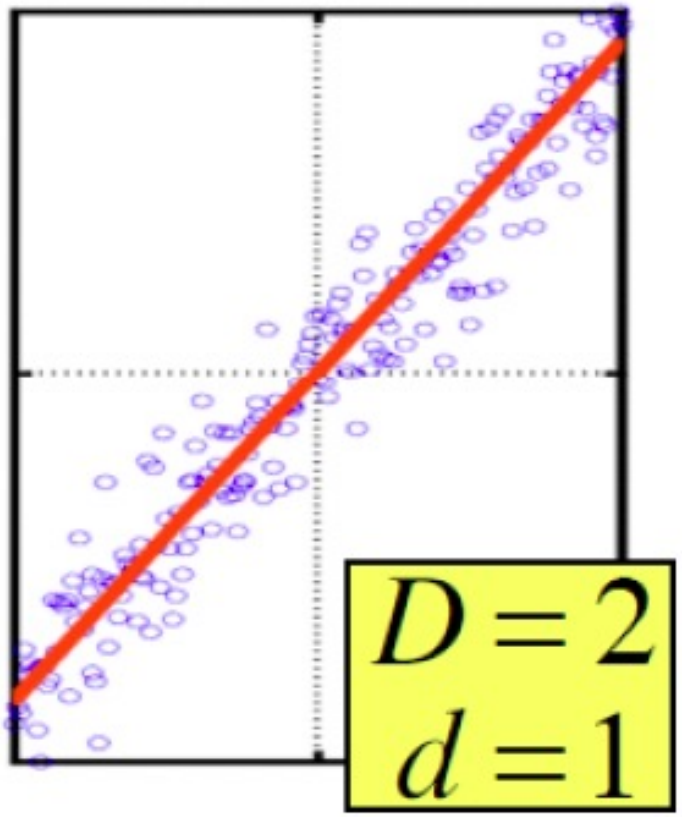


Rather than representing every point with 2 coordinates we represent each point with 1 coordinate (corresponding to the position of the point on the red line).

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

Dimensionality Reduction in NLP

- **Goal of dimensionality reduction is to discover axes of data!**



Rather than representing every point with |V| coordinates, we represent each point with k coordinates (corresponding to the position of the point on the red line).

What are the k coordinates?

Why Reduce Dimensions?

Discover hidden correlations/topics

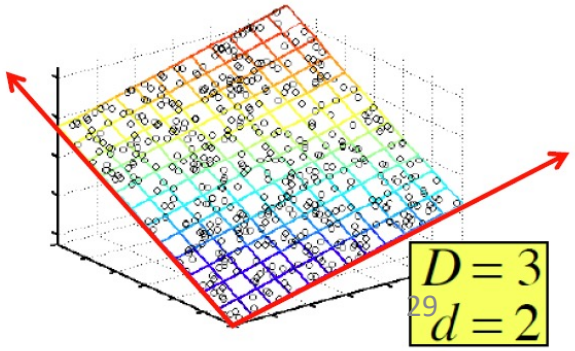
- Words that occur commonly together

Remove redundant and noisy features

- Not all words are useful

Interpretation and visualization

Easier storage and processing of the data



SVD - Definition

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

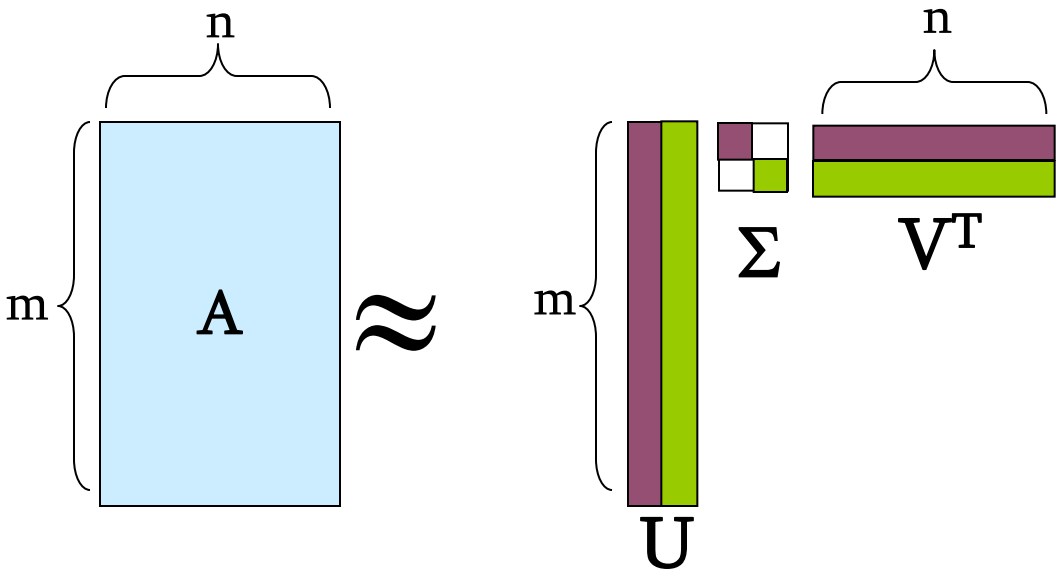
- **A**: **Input data matrix**
  - *m* x *n* matrix (e.g., *m* documents, *n* terms)
- **U**: **Left singular vectors**
  - *m* x *k* matrix (*m* documents, *r* concepts)

$$r << n$$

- Σ: **Singular values**
  - *r* x *r* diagonal matrix (strength of each 'concept')
    (*r* : rank of the matrix **A**)
- **V**: **Right singular vectors**
  - *n* x *r* matrix (*n* terms, *r* concepts)

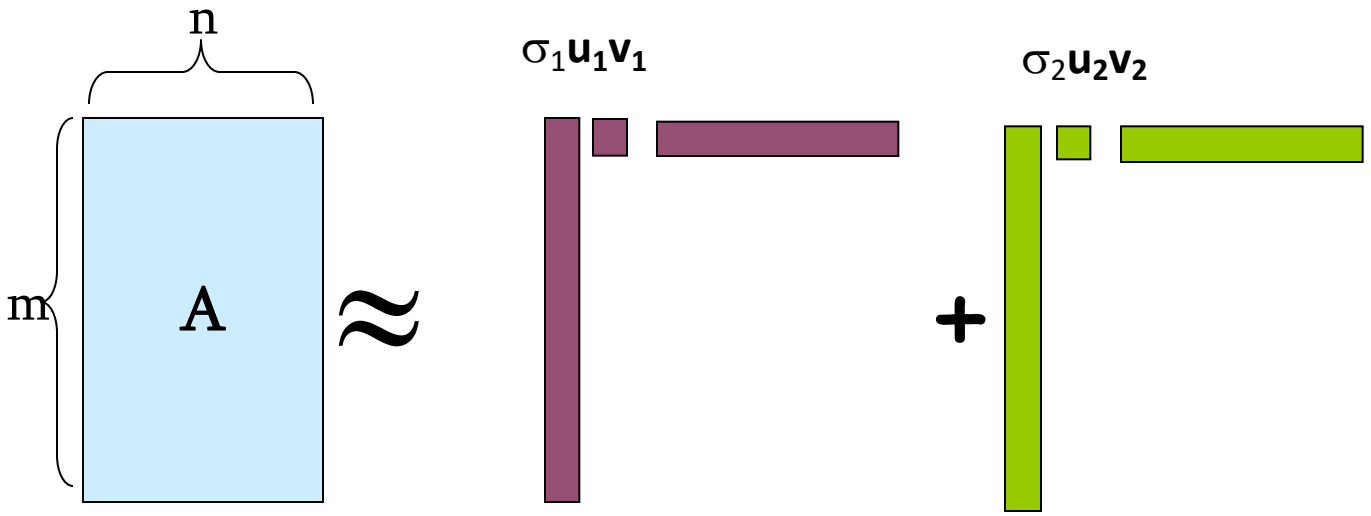
SVD

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



SVD

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



σ_i … scalar
u_i … vector
v_i … vector

SVD - Properties

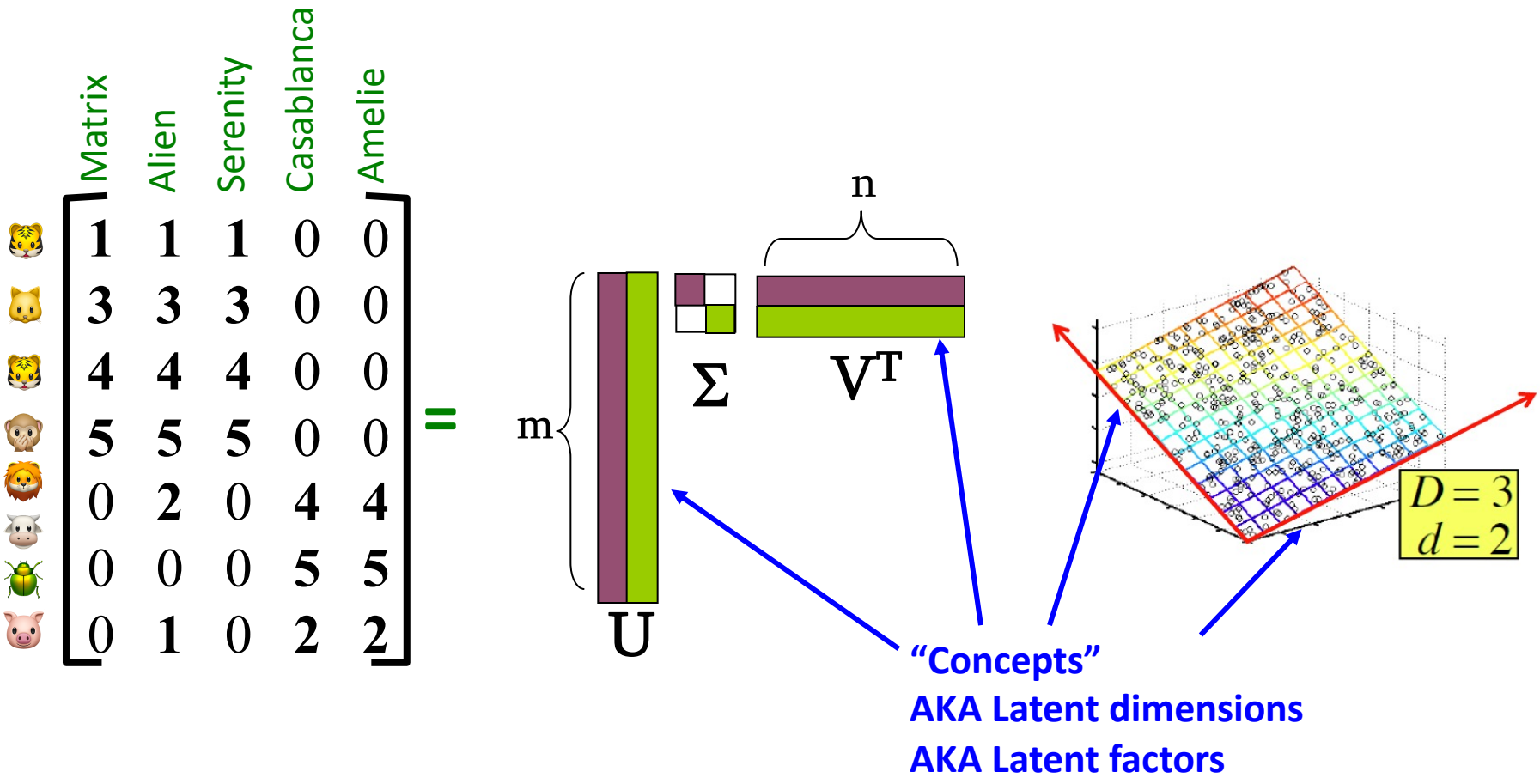
It is **always** possible to decompose a real matrix A into $A = U \Sigma V^T$, where

- 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 ($\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$)

Nice proof of uniqueness: http://www.mpi-inf.mpg.de/~bast/ir-seminar-ws04/lecture2.pdf

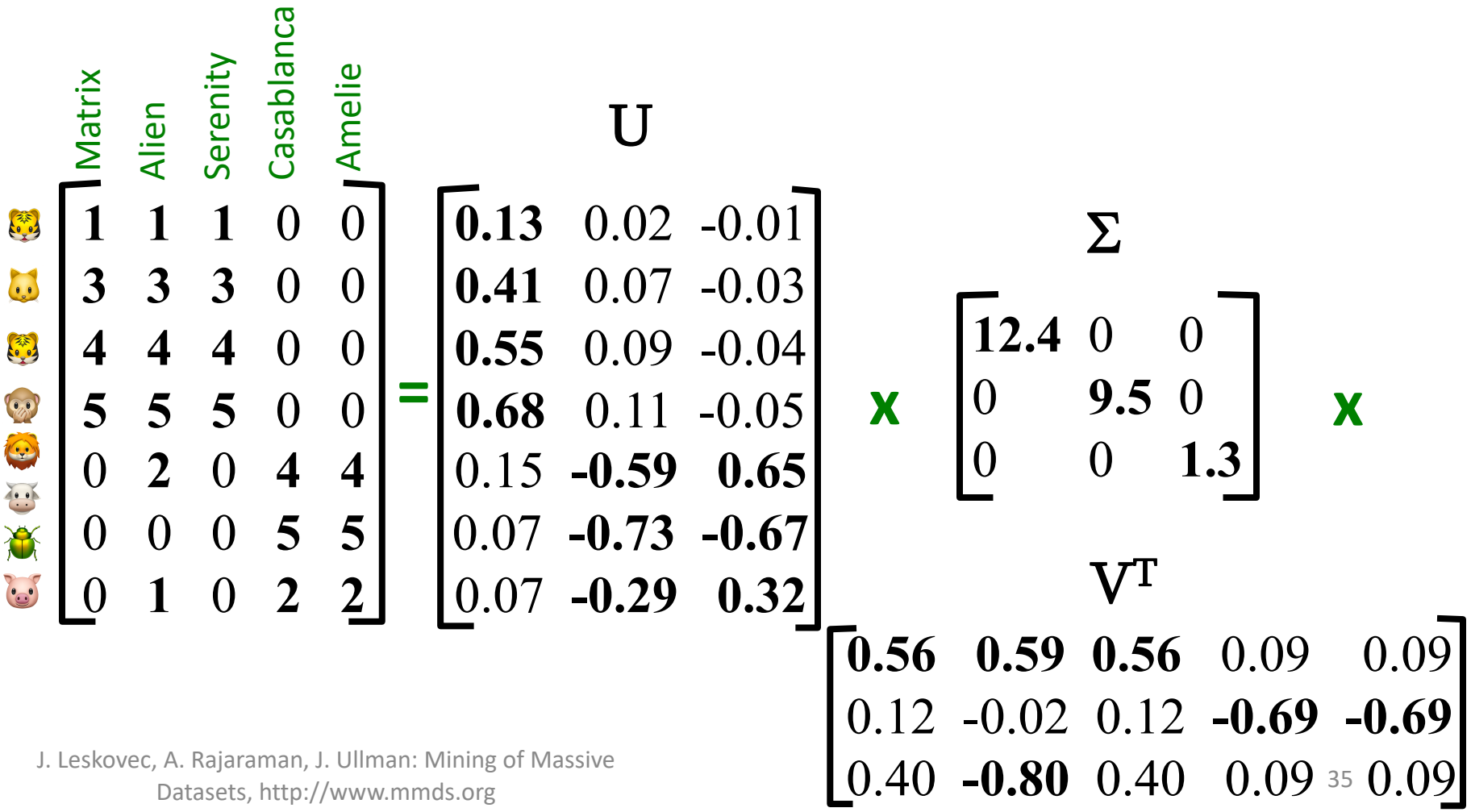
SVD – Example: Users-to-Movies

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



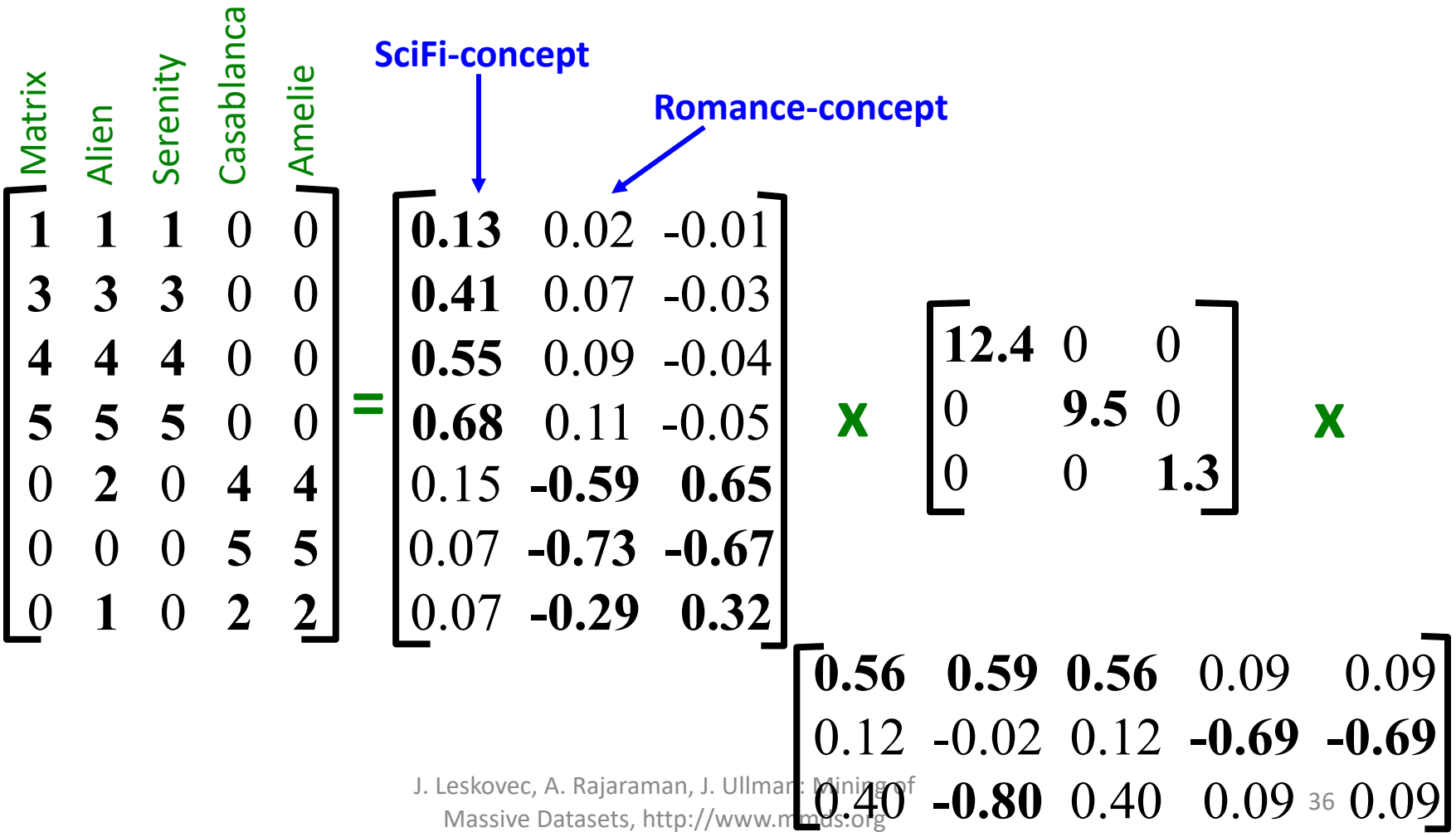
SVD – Example: Users-to-Movies

- **A = U Σ V^T - example: Users to Movies**



SVD – Example: Users-to-Movies

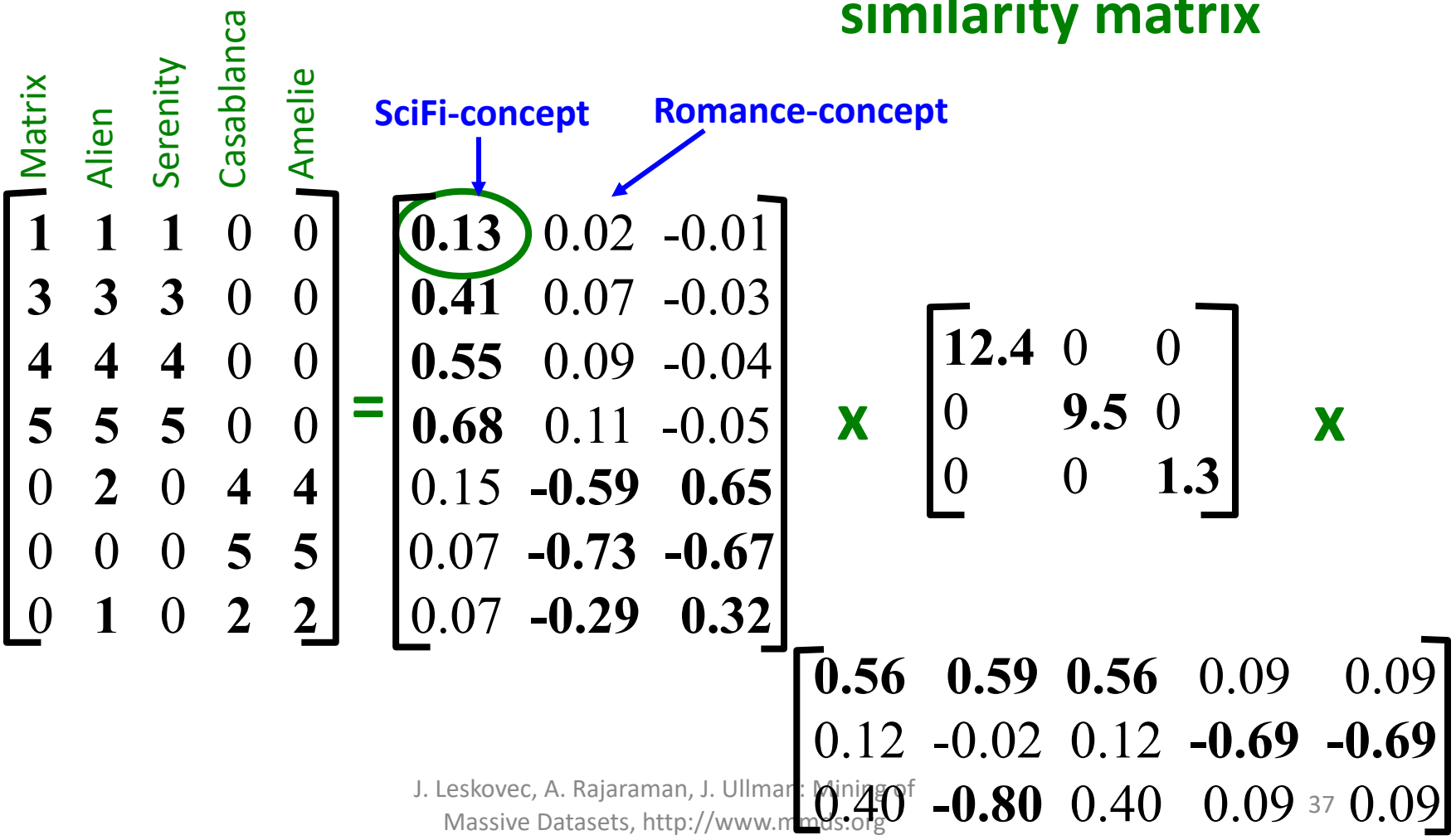
- **A = U Σ V^T - example: Users to Movies**



SVD – Example: Users-to-Movies

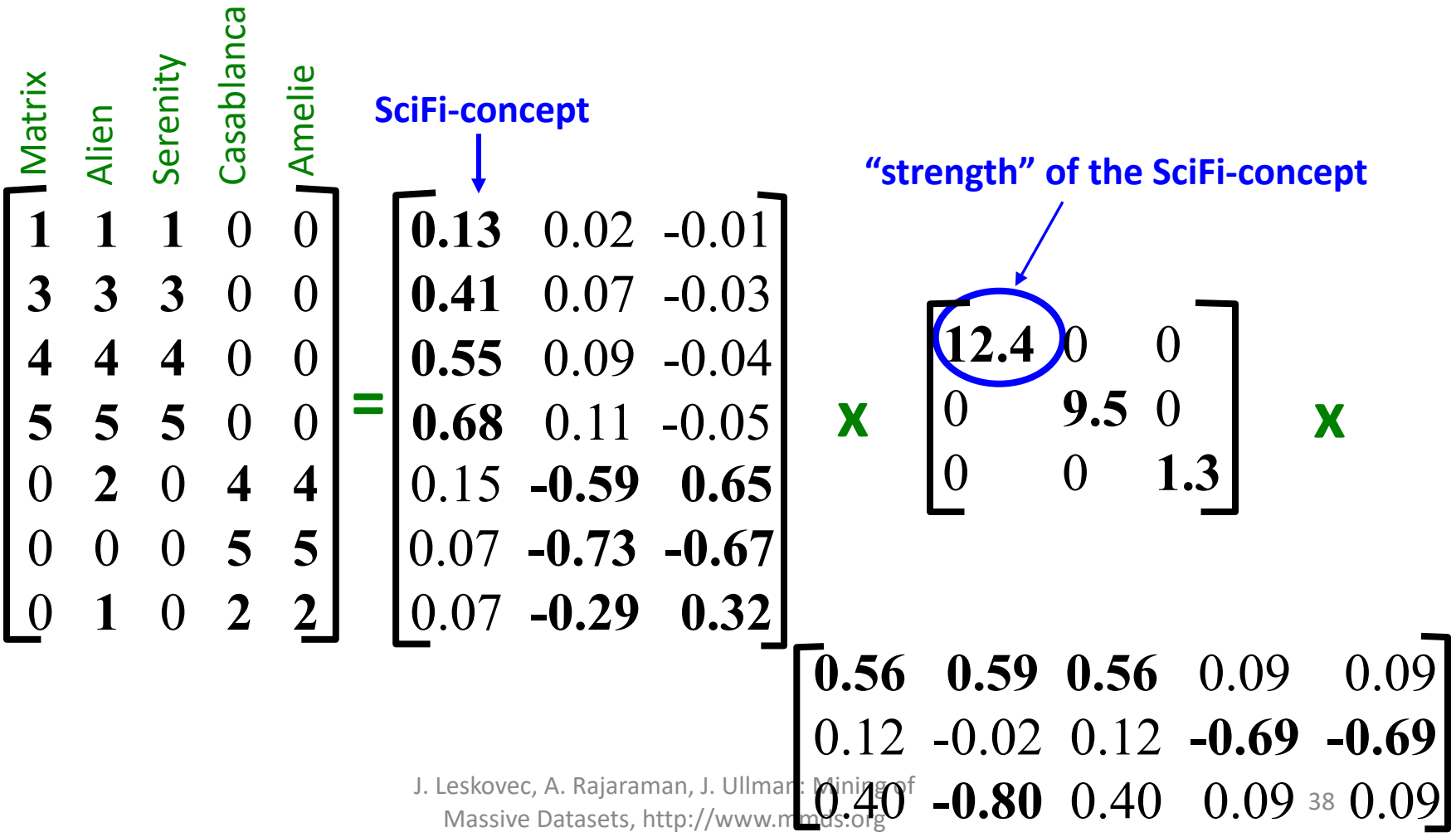
- **$A = U \Sigma V^T$ - example:**

_U_ is "user-to-concept" similarity matrix



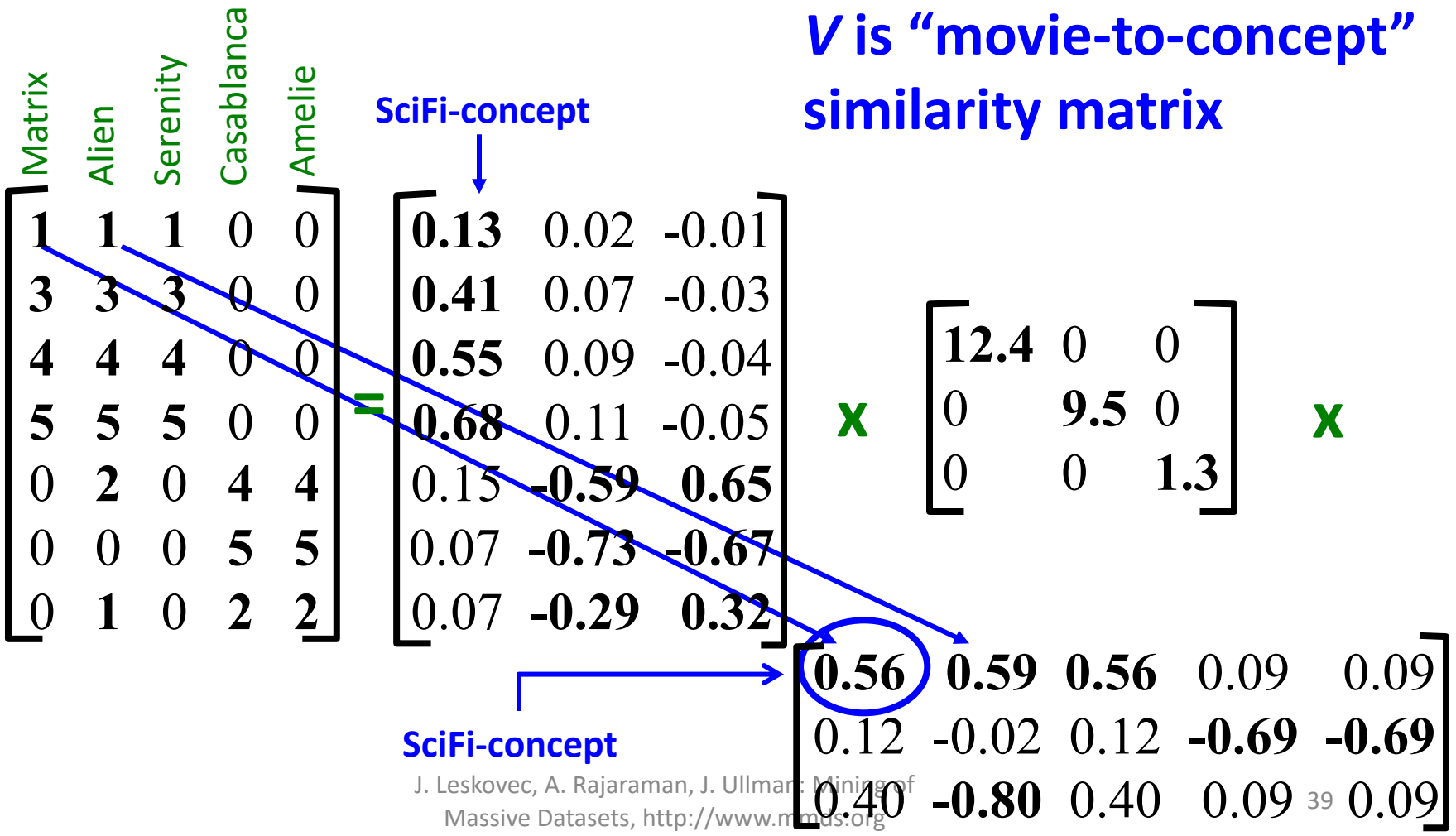
SVD – Example: Users-to-Movies

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



SVD – Example: Users-to-Movies

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

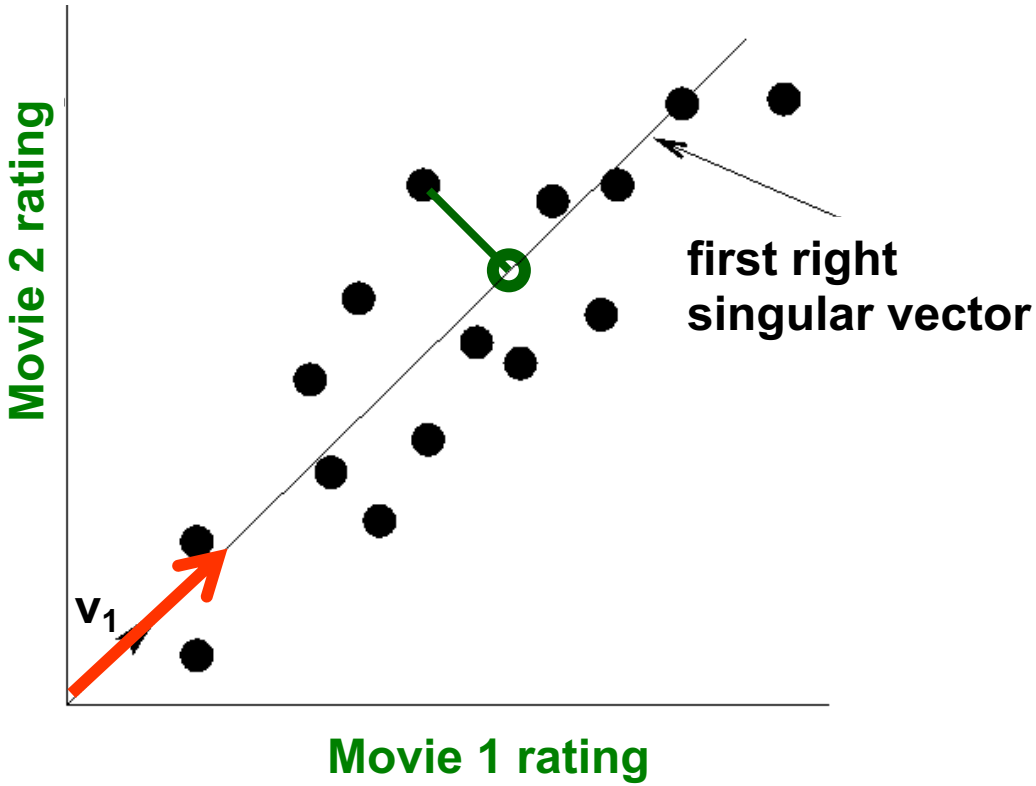


SVD - Interpretation #1

'**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 – Dimensionality Reduction



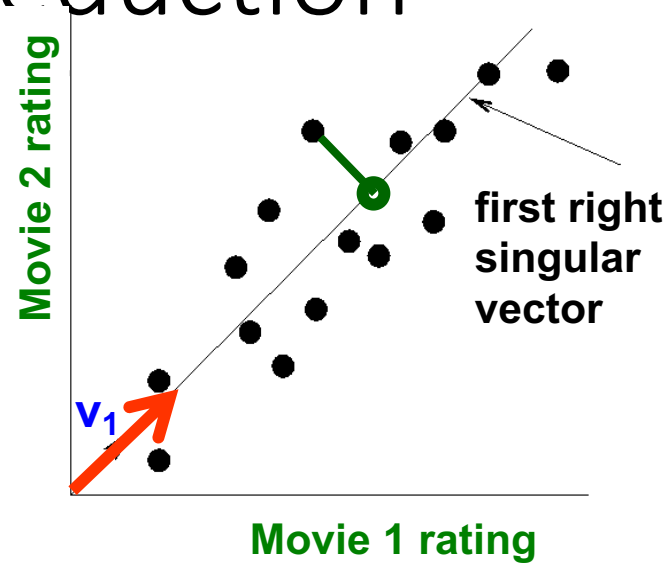
J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Datasets, http://www.mmds.org

- 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:**

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

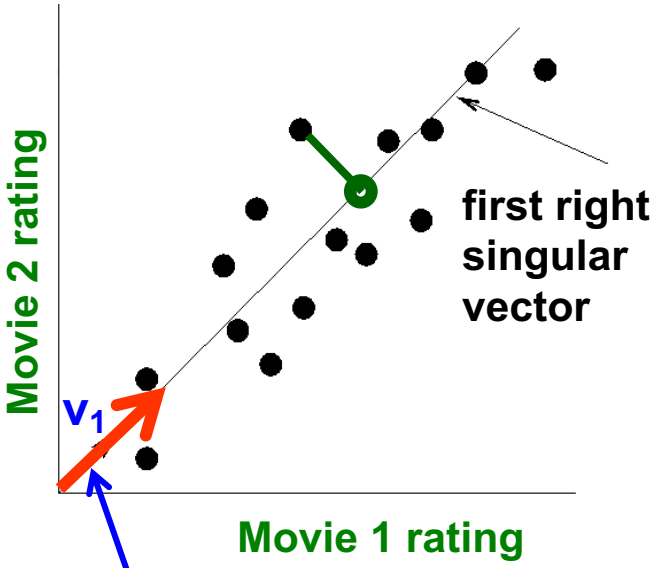


  - where x_{ij} are the "old" and z_{ij} are the "new" coordinates

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

SVD - Interpretation #2

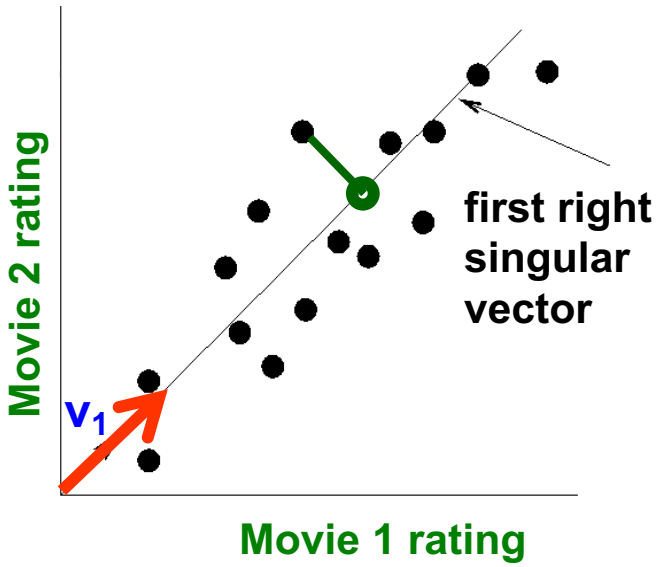
- **$A = U \Sigma V^T$ - example:**
  - **V**: "movie-to-concept" matrix
  - **U**: "user-to-concept" matrix



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

SVD - Interpretation #2

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



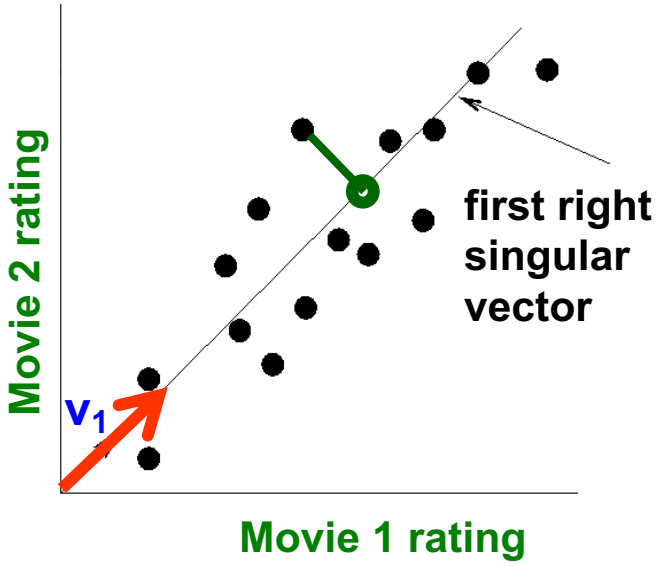
variance ('spread') on the v_1 axis

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

SVD - Interpretation #2

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

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



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

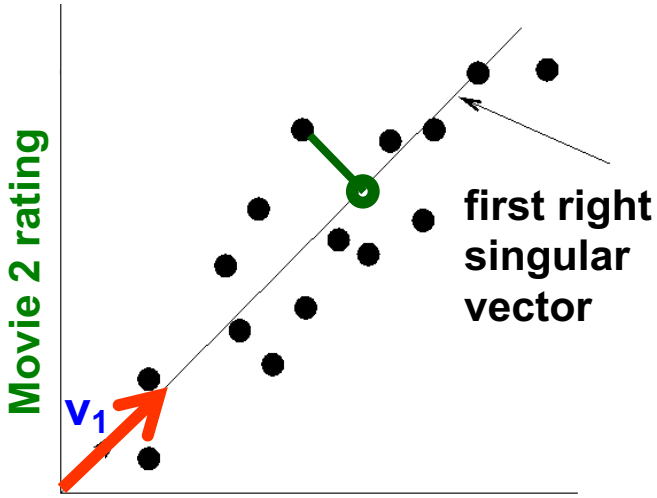
Projection of users on the "Sci-Fi" axis $(U \Sigma)^T$:

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

SVD - Interpretation #2

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

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



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

Low variation around the third axis

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

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**

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

SVD - Interpretation #2

More details

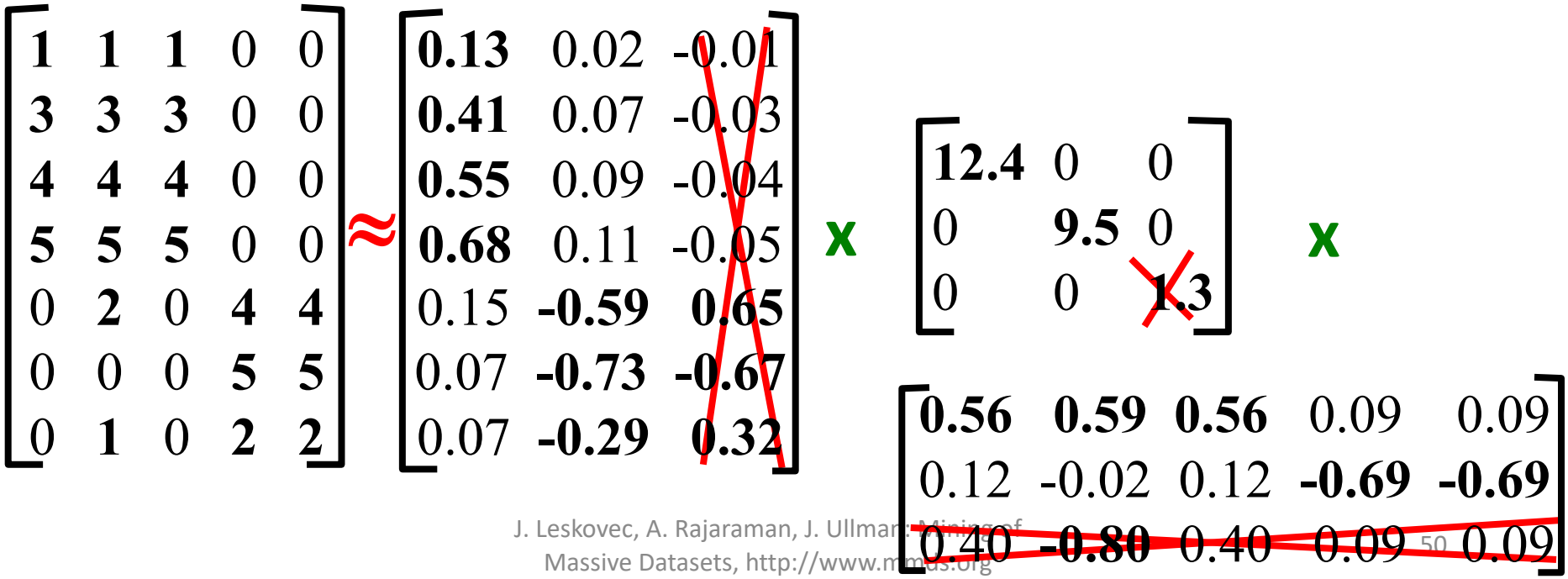
- Q: **How exactly is dim. reduction done?**
- **A: Set smallest singular values to zero**

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

SVD - Interpretation #2

More details

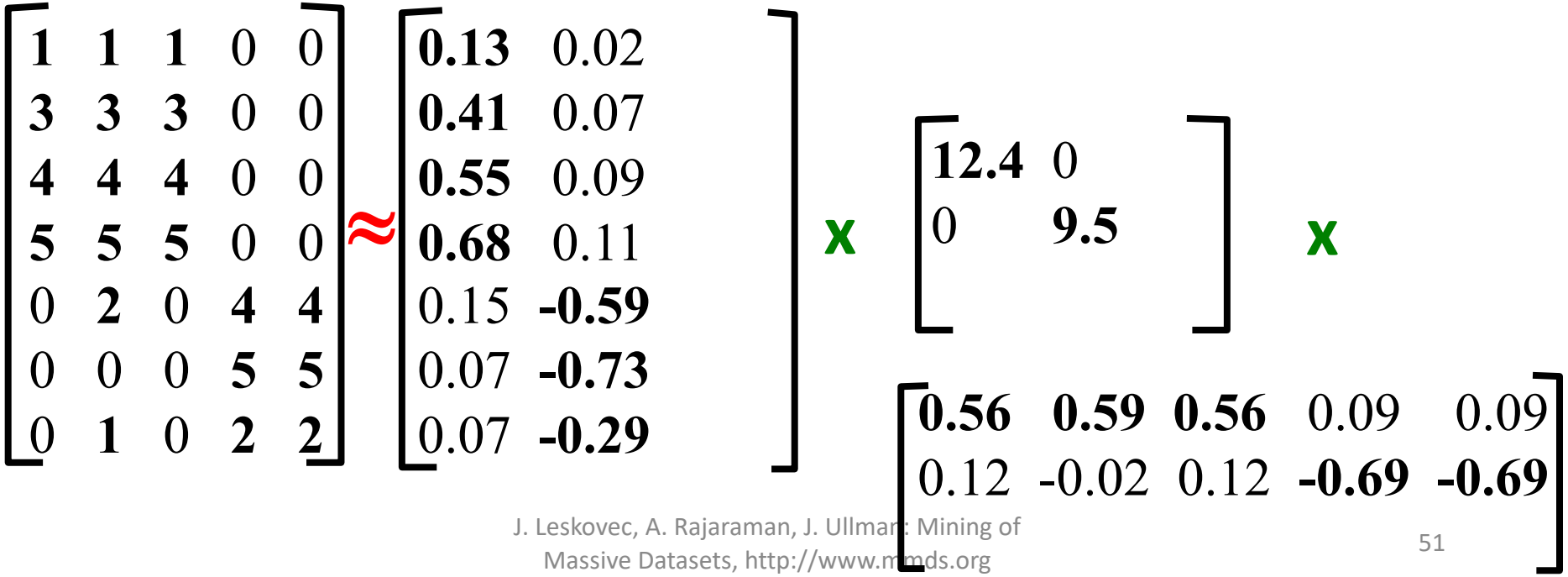
- Q: **How exactly is dim. reduction done?**
- **A: Set smallest singular values to zero**



SVD - Interpretation #2

More details

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



SVD - Interpretation #2

More details

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

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

Frobenius norm:

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

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

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

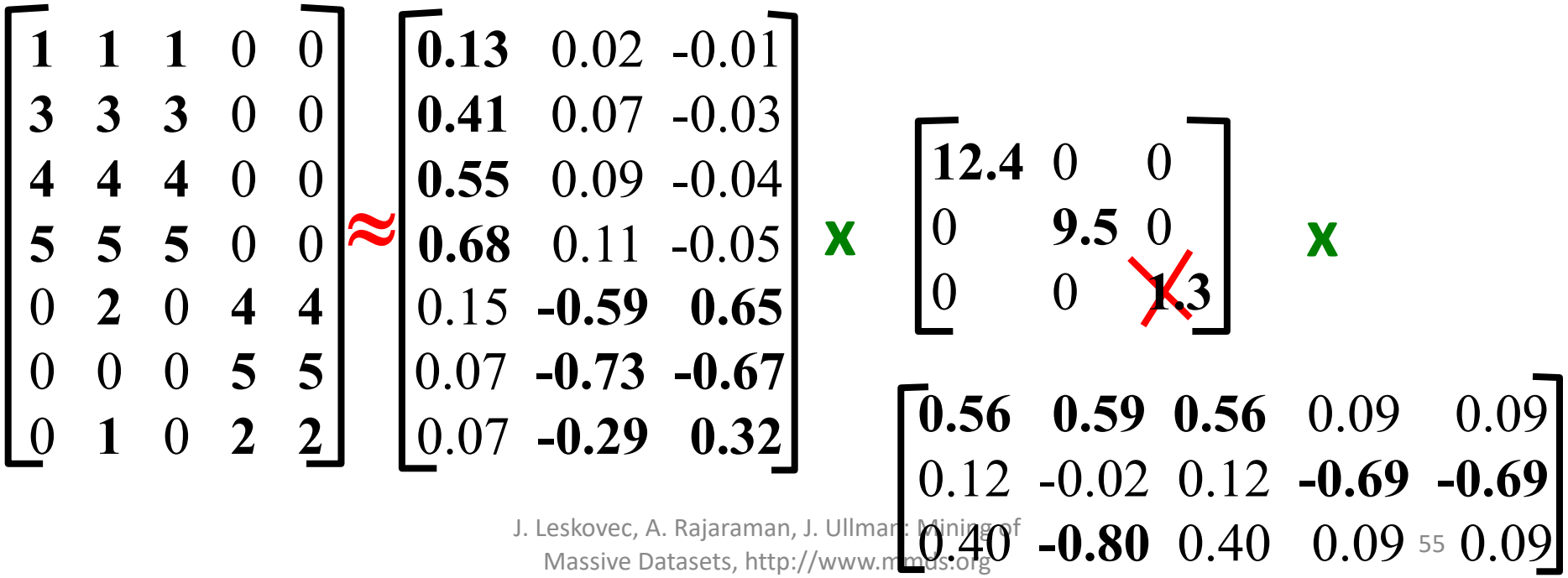
- Q: **How exactly is dim. reduction done?**
- **A: Set smallest singular values to zero**

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

SVD - Interpretation #2

More details

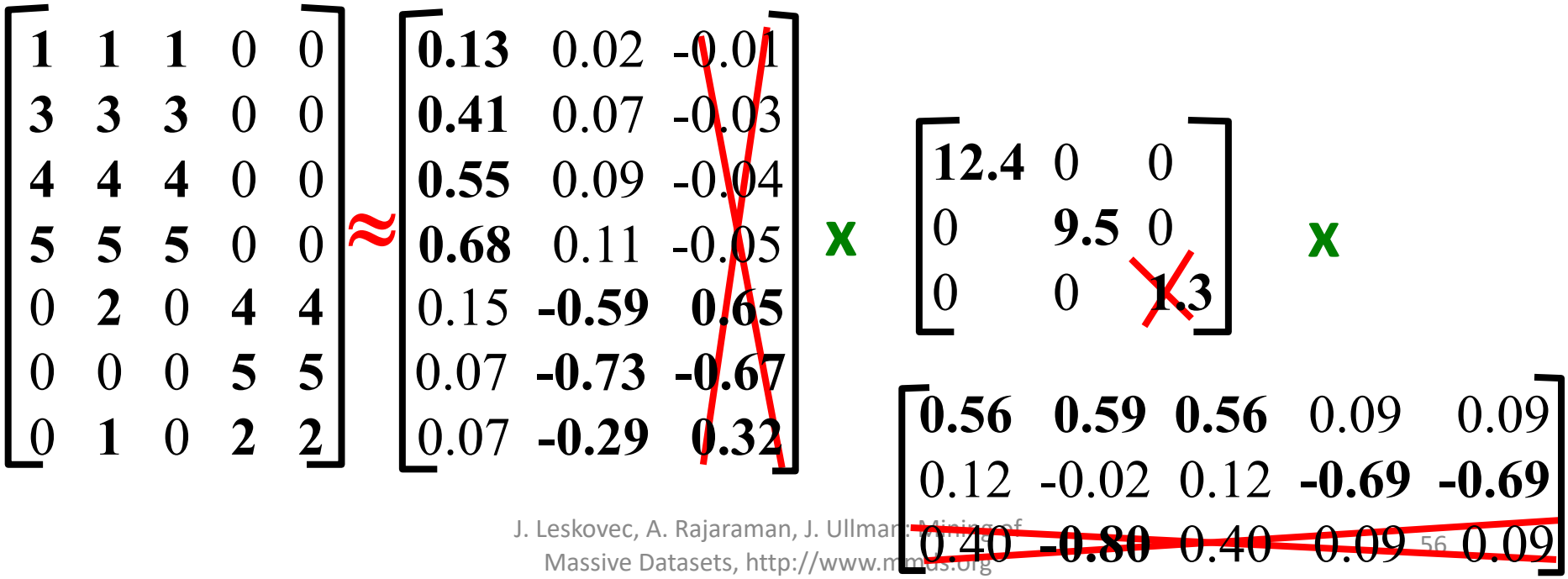
- Q: **How exactly is dim. reduction done?**
- **A: Set smallest singular values to zero**



SVD - Interpretation #2

More details

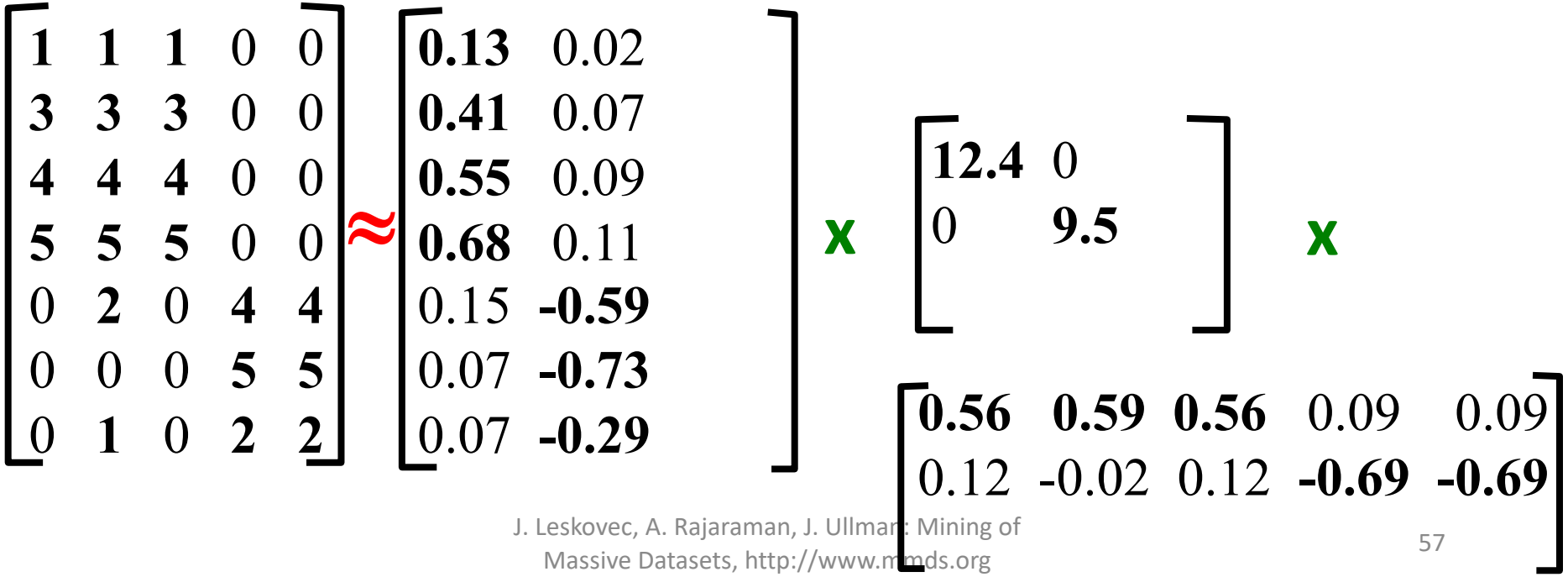
- Q: **How exactly is dim. reduction done?**
- **A: Set smallest singular values to zero**



SVD - Interpretation #2

More details

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



SVD - Interpretation #2

More details

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

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

A B

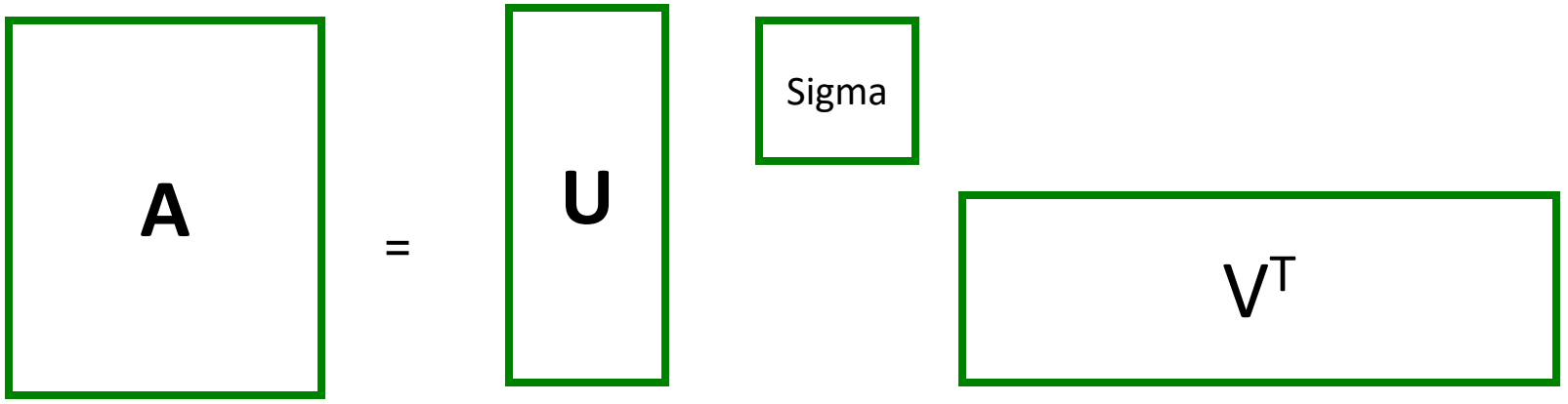
Frobenius norm:

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

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

is "small"

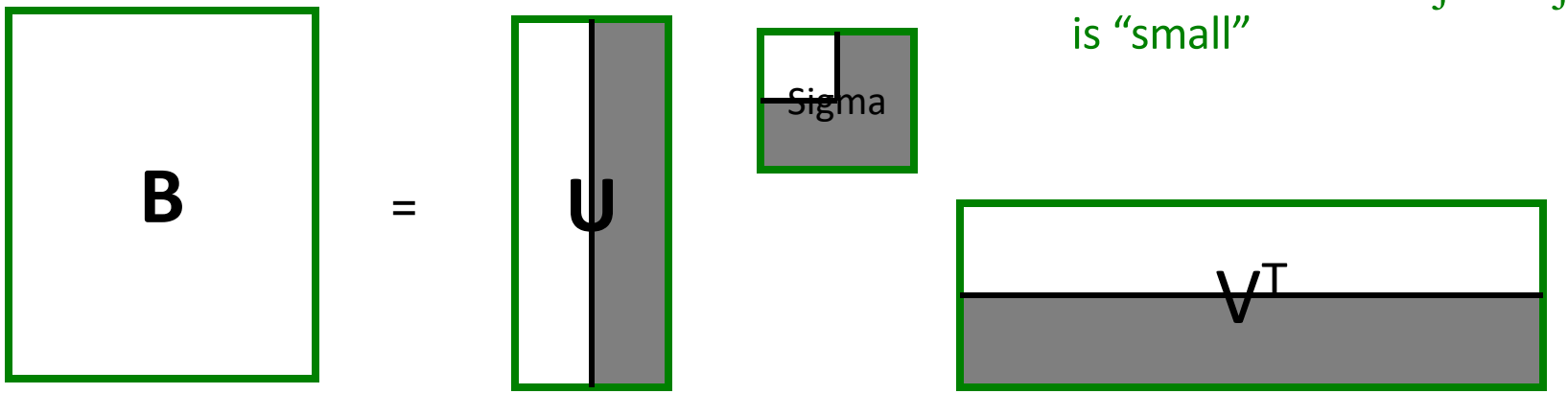
SVD – Best Low Rank Approx.



B is best approximation of A

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

is "small"

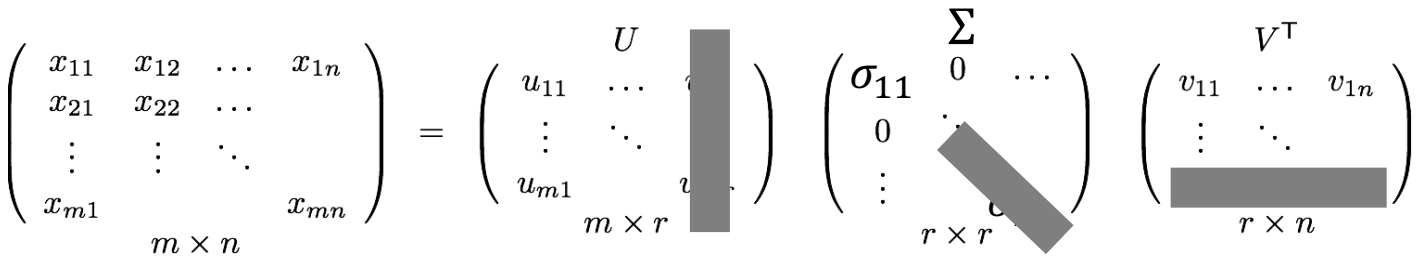


SVD – Best Low Rank Approx.

- **Theorem:**
  Let **A = U Σ V^T** and **B = U S V^T** where
  S = diagonal *r*x*r* matrix with $s_i=\sigma_i$ ($i=1\ldots k$) else $s_i=0$
  then **B** is a **best** rank(**B**)=***k*** approx. to ***A***

What do we mean by "best":

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

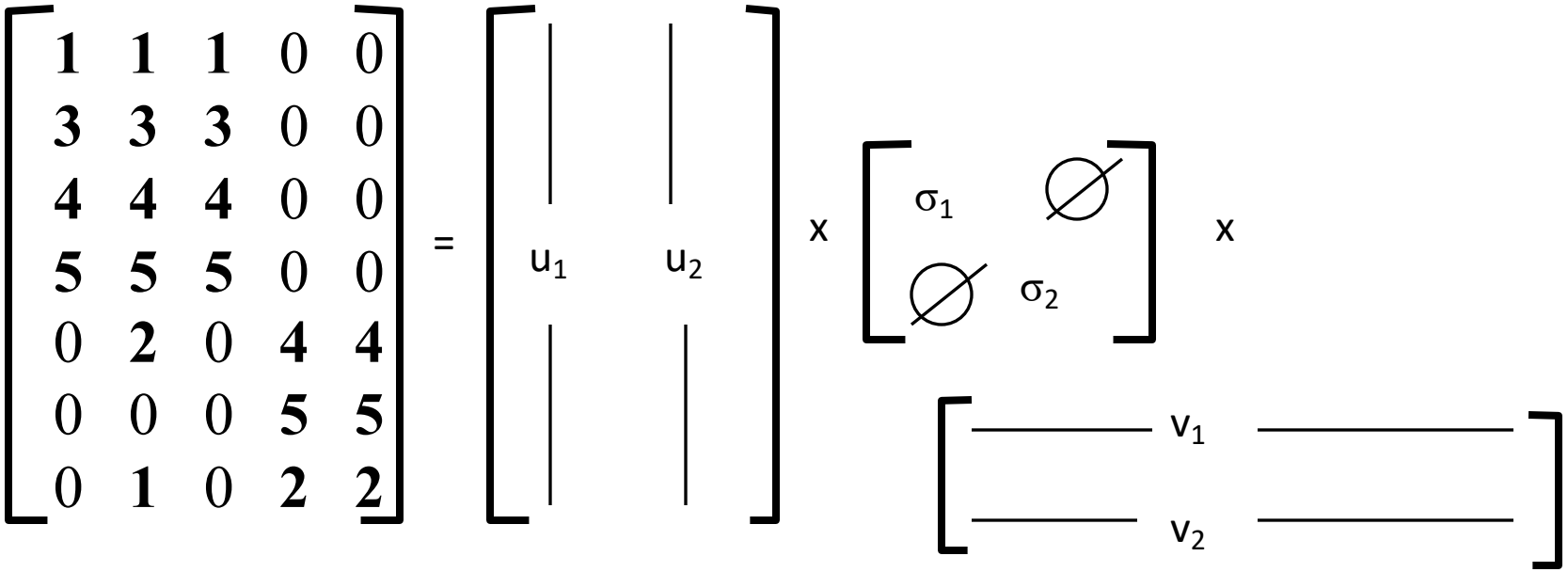


$$\|A-B\|_F = \sqrt{\sum_{ij}\left(A_{ij}-B_{ij}\right)^2}$$

SVD - Interpretation #2

Equivalent:

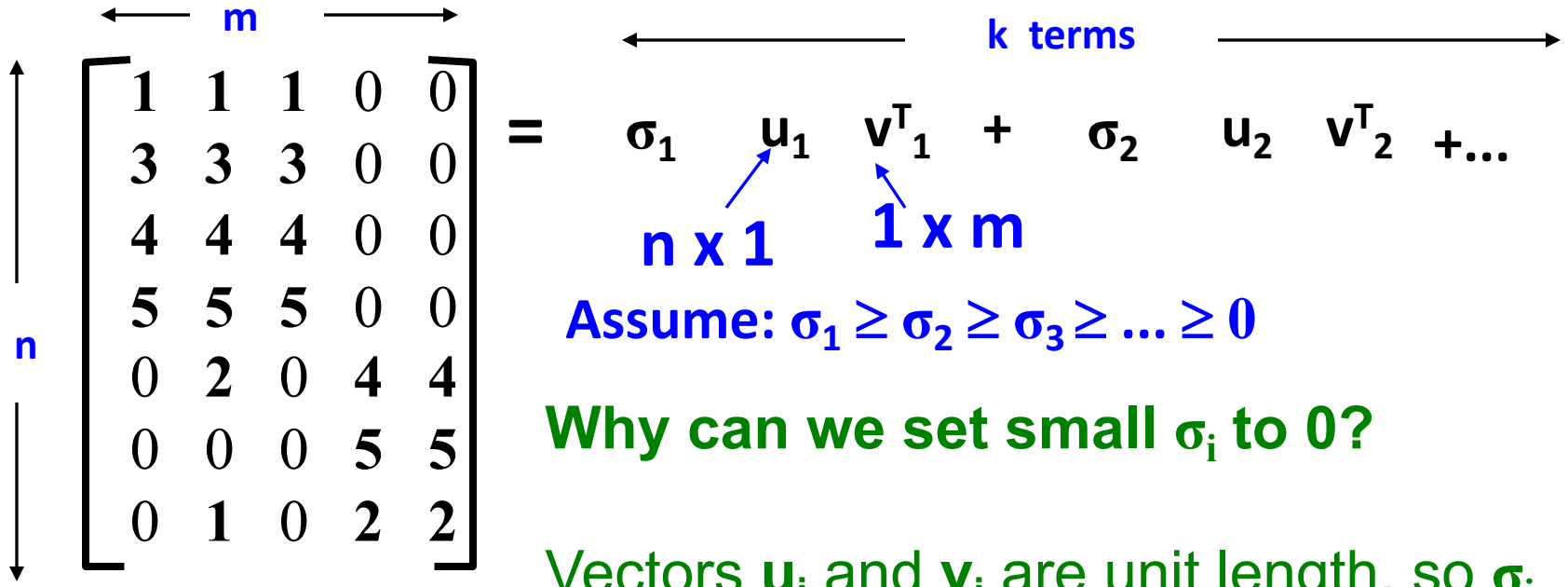
'spectral decomposition' of the matrix:



SVD - Interpretation #2

Equivalent:

'spectral decomposition' of the matrix



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

Why can we set small σ_i to 0?

Vectors $\mathbf{u_i}$ and $\mathbf{v_i}$ are unit length, so σ_i scales them.
So, zeroing small σ_i introduces less error.

SVD - Interpretation #2

Q: How many σ_s to keep?

A: Rule-of-a thumb:

keep 80-90% of 'energy' $= \sum_i \sigma_i^2$

m

n

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 2 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 1 & 0 & 2 & 2 \end{bmatrix} = \sigma_1 \; u_1 \; v^T_1 + \sigma_2 \; u_2 \; v^T_2 + \ldots$$

$$\Big/ \sum_i \sigma_i^2 \geq .80 \quad \text{for } \sum_i^k \sigma_i^2$$

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, Sklearn ...

SVD - Conclusions so far

- SVD: **A= U Σ 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

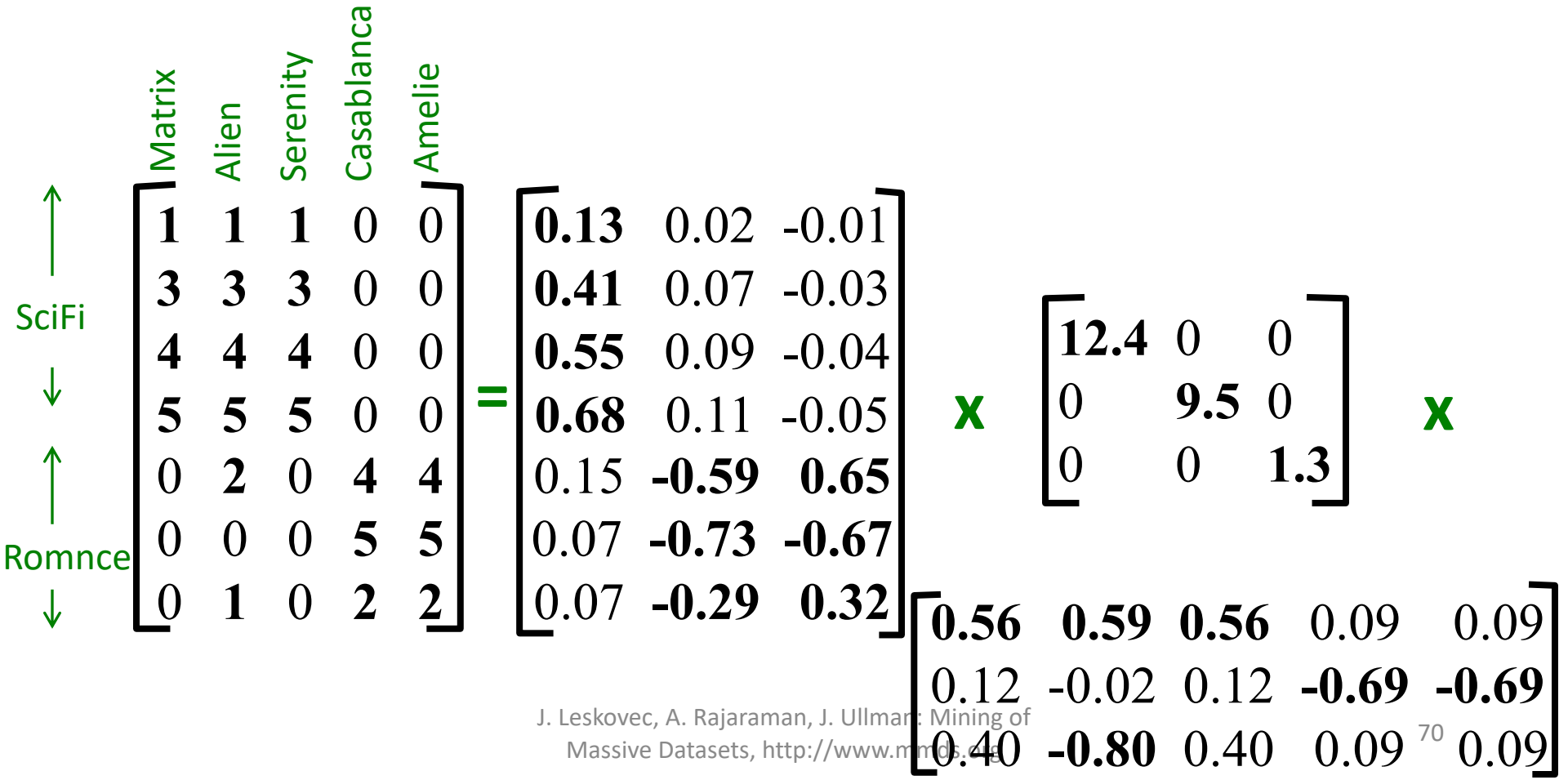
SVD - for Document-Term Matrix

- SVD: **A= U Σ V^T**: **unique**
  - **U**: document-to-concept similarities
  - **V**: term-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

Case study: How to query?

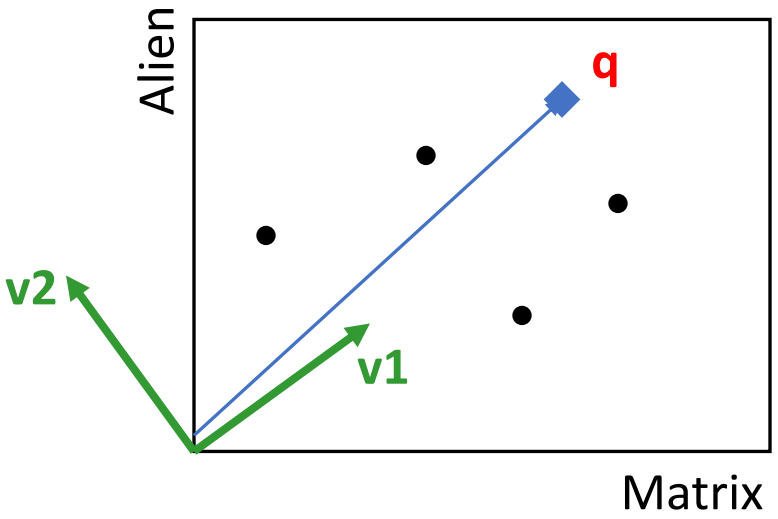
- **Q: Find users that like 'Matrix'**
- **A: Map query into a 'concept space' – how?**



Case study: How to query?

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

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

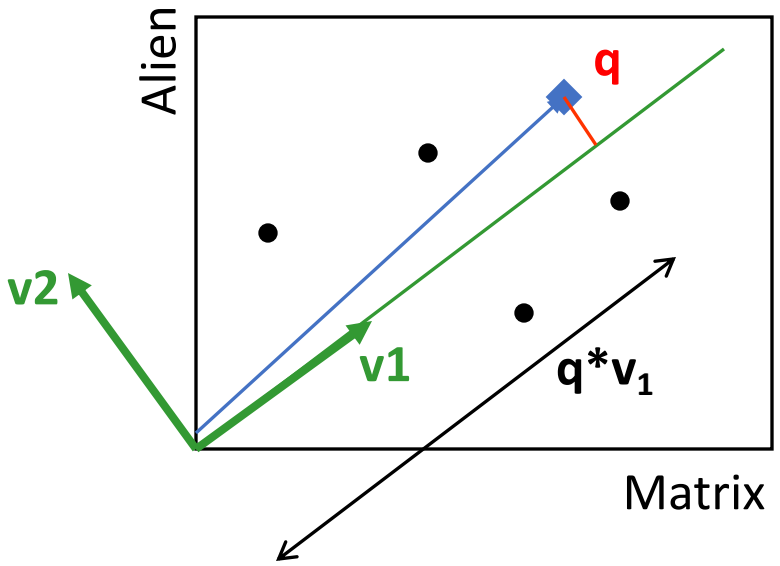


Project into concept space:
Inner product with each 'concept' vector $\mathbf{v}_i$

Case study: How to query?

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

	Matrix	Alien	Serenity	Casablanca	Amelie
q =	5	0	0	0	0



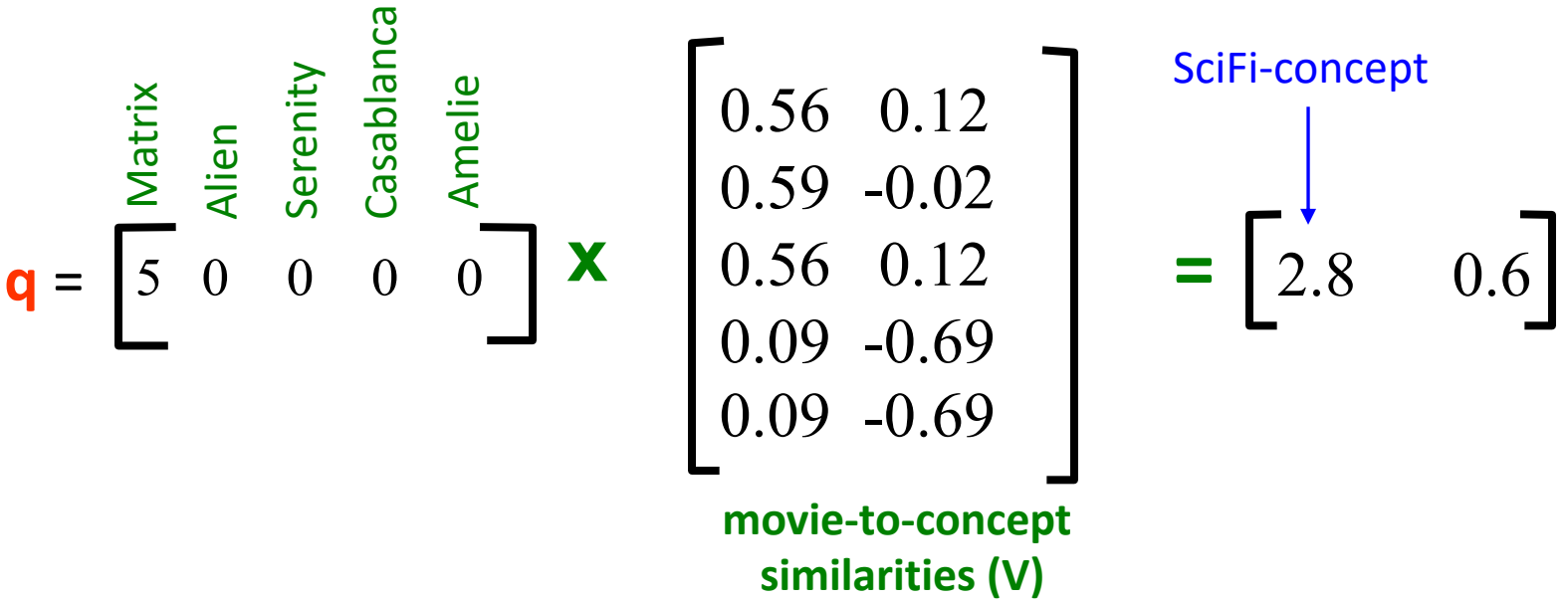
Project into concept space:
Inner product with each 'concept' vector $\mathbf{v_i}$

Case study: How to query?

Compactly, we have:

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

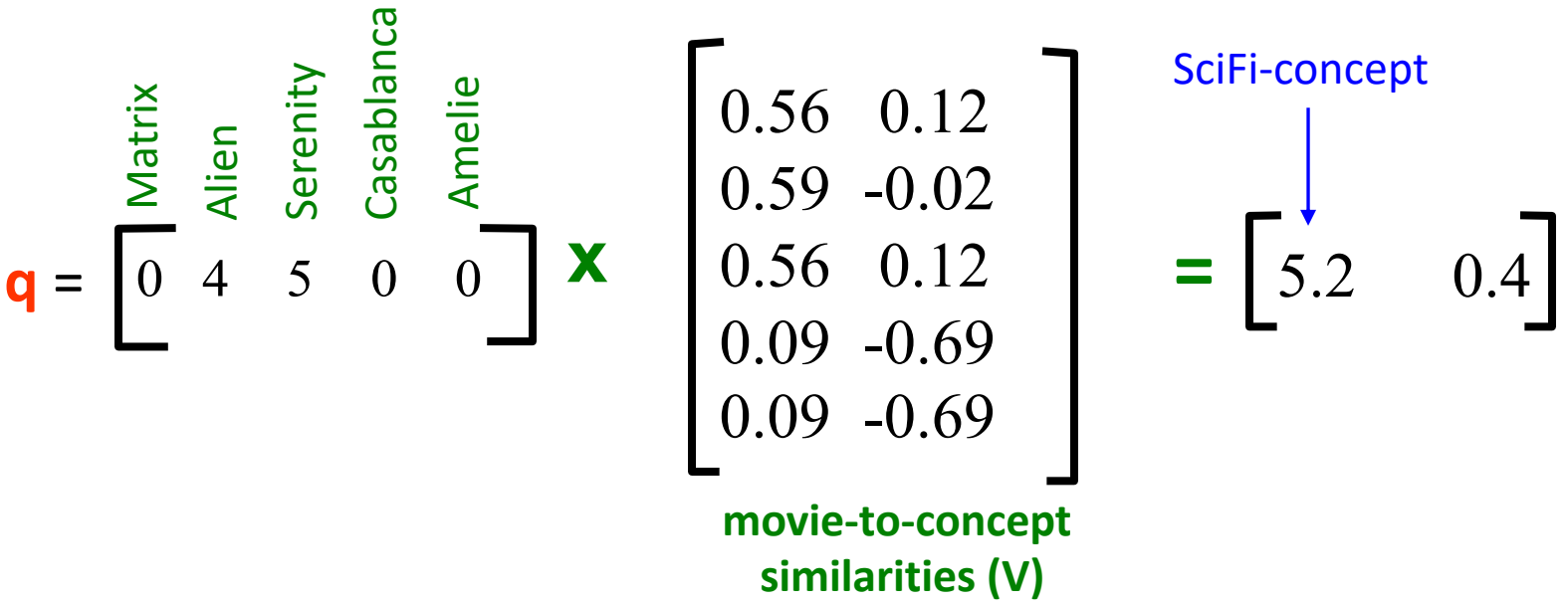
E.g.:



Case study: How to query?

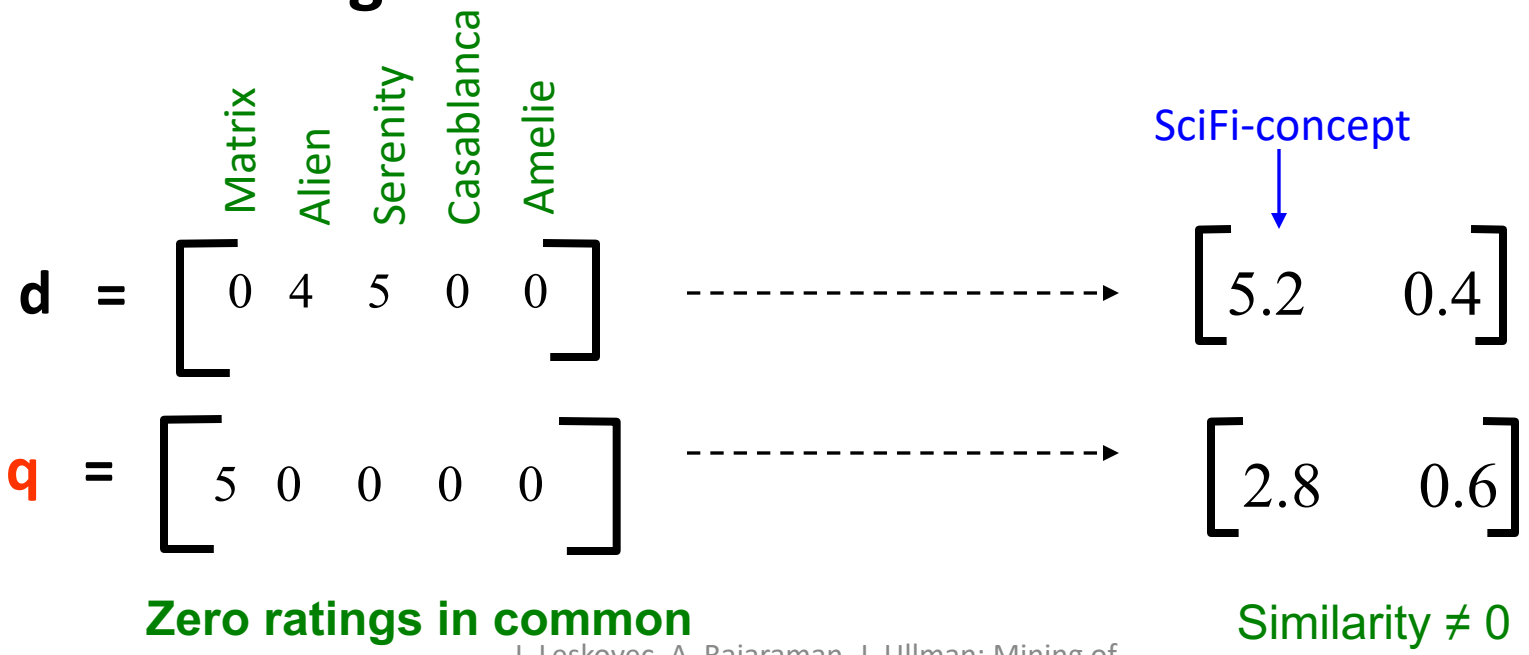
- **How would the user *d* that rated ('Alien', 'Serenity') be handled?**
  $d_{concept} = d\ V$

E.g.:



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

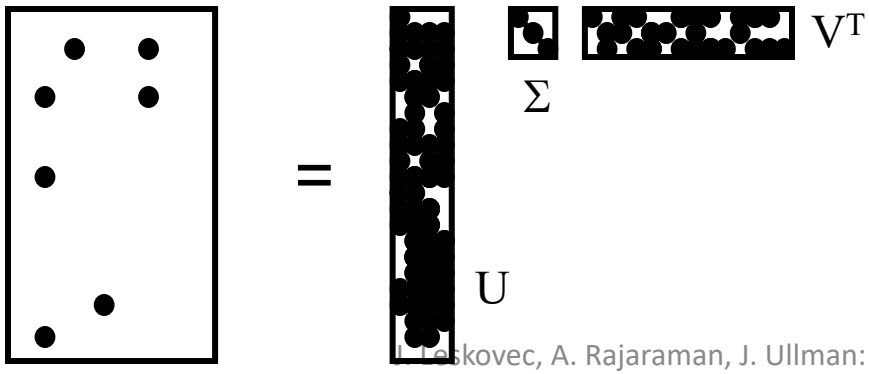
+ **Optimal low-rank approximation**
  in terms of Frobenius norm

− **Interpretability problem:**

  - A singular vector specifies a linear combination of all input columns or rows

− **Lack of sparsity:**

  - Singular vectors are **dense!**



Latent Semantic Analysis

Applying SVD to the Document Term Matrix