

SVD Example & Conclusion

Mining of Massive Datasets
Leskovec, Rajaraman, and Ullman
Stanford University



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 into a concept space for finding users that like 'Matrix'.

The query matrix (Matrix) is shown as a 6x5 matrix, where rows represent genres (SciFi, Romance) and columns represent movies (Matrix, Alien, Serenity, Casablanca, Amelie). The matrix is:

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

The matrix is mapped to a concept space (represented by a 6x3 matrix) using a transformation matrix (represented by a 3x3 matrix). The transformation is shown as:

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

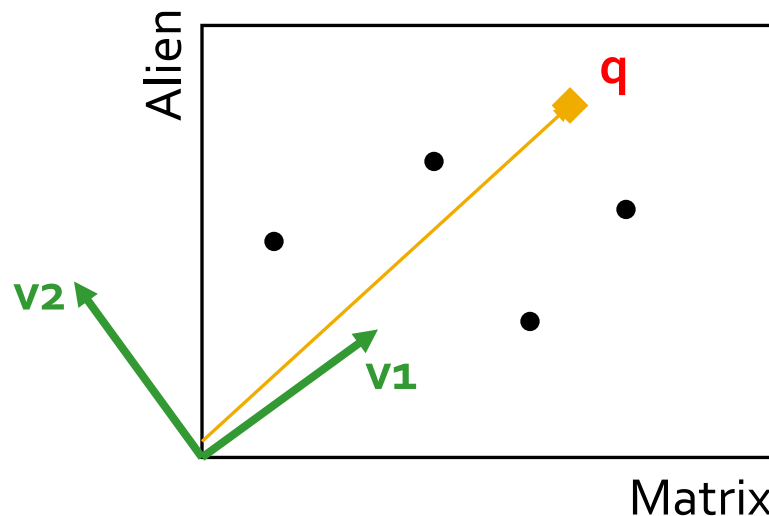
The resulting matrix (concept space) is shown as a 6x5 matrix.

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

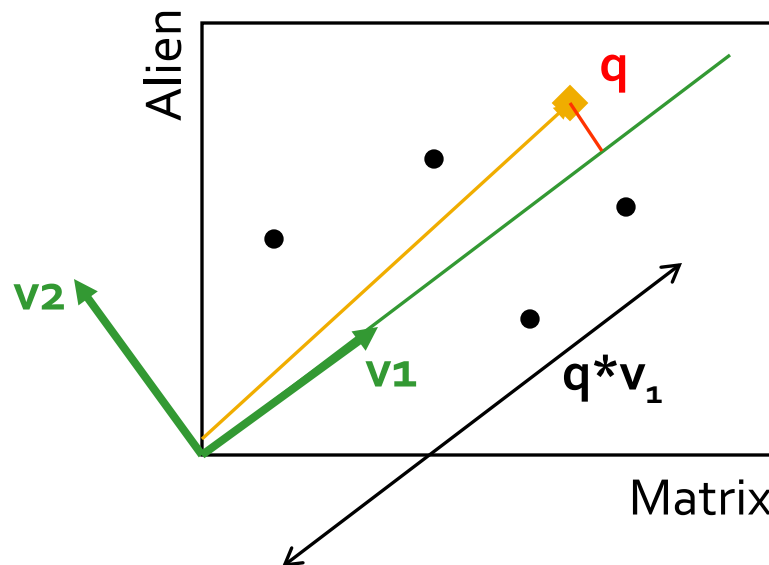


Case study: How to query?

- 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:

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

E.g.:

$$\mathbf{q} = \begin{bmatrix} \text{Matrix} \\ 5 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x} \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} = \begin{bmatrix} 2.8 & 0.6 \end{bmatrix}$$

movie-to-concept similarities (V)

SciFi-concept

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.:

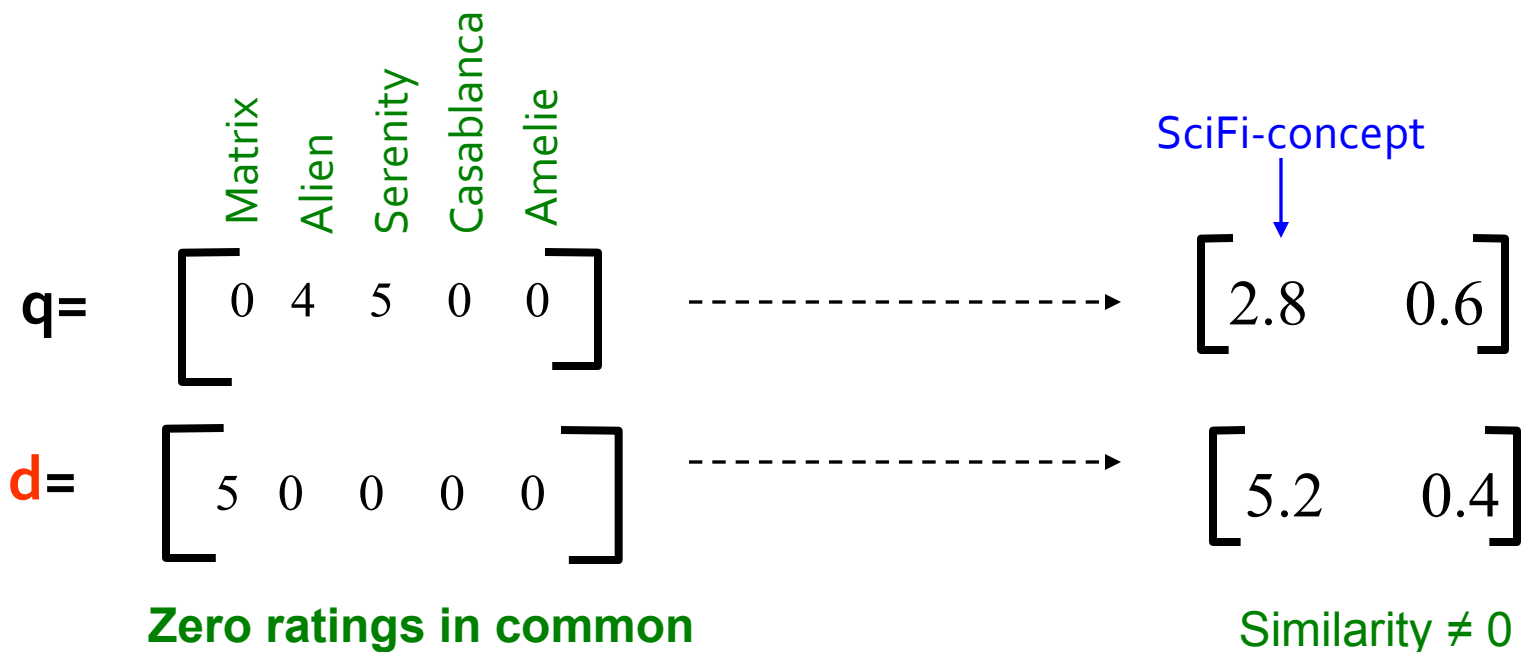
$$\mathbf{d} = \begin{matrix} & \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} & \times & \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} & = & \begin{bmatrix}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

movie-to-concept
similarities (V)

SciFi-concept
↓

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**!



Relation to Eigen-decomposition

- SVD gives us:

- $A = U \Sigma V^T$

- Eigen-decomposition:

- $A = X \Lambda X^T$

- A is symmetric

- U, V, X are orthonormal (e.g., $U^T U = I$),

- Λ, Σ are diagonal

- What is:

- $AA^T =$

- $A^T A = V \Sigma^T U^T (U \Sigma V^T) = V \Sigma \Sigma^T V^T$

Relation to Eigen-decomposition

- SVD gives us:

- $A = U \Sigma V^T$

- Eigen-decomposition:

- $A = X \Lambda X^T$

- A is symmetric

- U, V, X are orthonormal (e.g., $U^T U = I$),

- Λ, Σ are diagonal

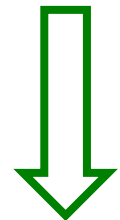
- What is:

- $AA^T = U \Sigma V^T (U \Sigma V^T)^T = U \Sigma V^T (V \Sigma^T U^T) = U \Sigma \Sigma^T U^T$

- $A^T A = V \Sigma^T U^T (U \Sigma V^T) = V \Sigma \Sigma^T V^T$

$$\begin{matrix} \uparrow & \uparrow & \uparrow \\ X & \Lambda & X^T \end{matrix}$$

Shows how to compute
SVD using eigenvalue
decomposition!



$$\begin{matrix} X & \Lambda & X^T \\ \downarrow & \downarrow & \downarrow \end{matrix}$$

So, $\lambda_i = \sigma_i^2$

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 six black dots. This is followed by an equals sign. To the right of the equals sign is a tall, narrow rectangle labeled U at the bottom, which is densely packed with black dots. To the right of U is a small square labeled Σ at the bottom, containing a few black dots. To the right of Σ is a wide, short rectangle labeled V^T at the bottom, which is also densely packed with black dots.