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?

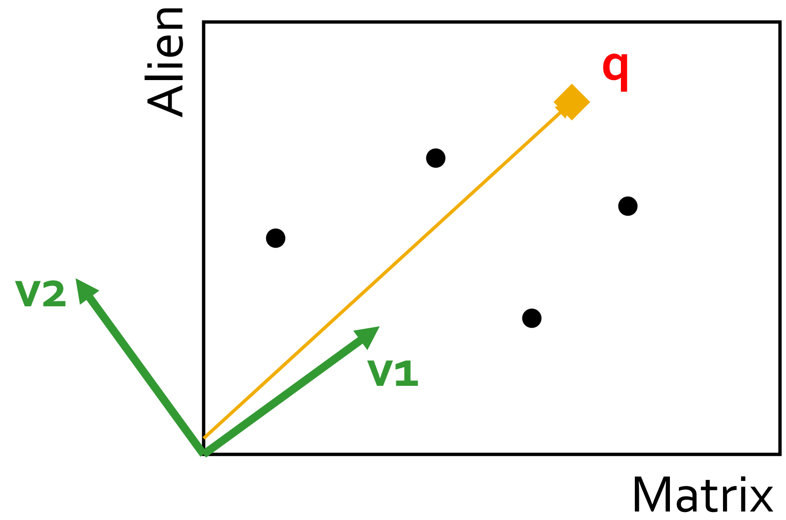
$$
\begin{array}{c c}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{array}{c} \\ \\ \text{SciFi} \\ \\ \\ \text{Romnce} \\ \\ \end{array} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3}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

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 $\mathbf{v_i}$

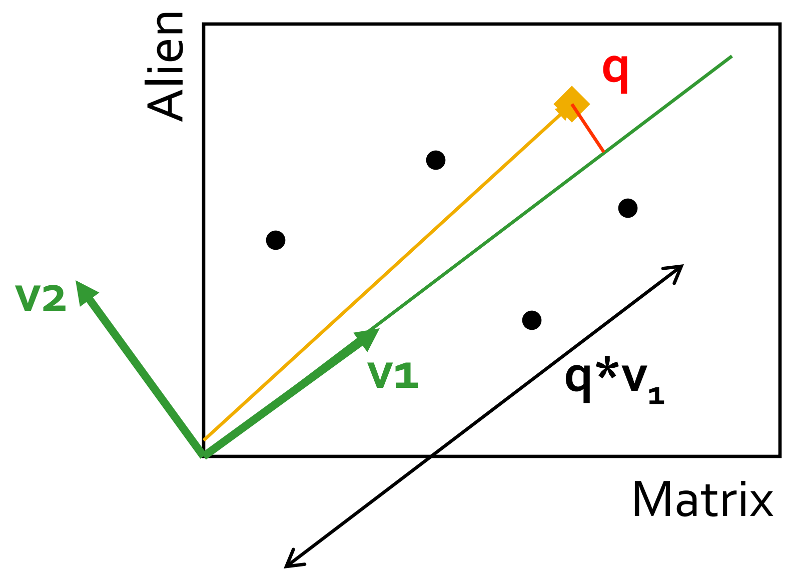


Case study: How to query?

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

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

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



Case study: How to query?

Compactly, we have:

$q_{concept} = q\ V$

E.g.:

$$
q = \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\ \hline 5 & 0 & 0 & 0 & 0 \end{array}
\ \mathbf{x} \
\begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix}
= \begin{array}{cc} \text{SciFi-concept} & \\ 2.8 & 0.6 \end{array}
$$

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.:

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

$$\mathbf{d} = \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \mathbf{x} \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} \mathbf{=} \begin{array}{cc} \text{SciFi-concept} & \\ \downarrow & \\ \end{array}$$

$$\begin{bmatrix} 5.2 & 0.4 \end{bmatrix}$$

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**!

$$
\begin{array}{cccccc}
 & \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}
\end{array}
\qquad\qquad
\begin{array}{cc}
\text{SciFi-concept} & \\
\downarrow &
\end{array}
$$

$$
\mathbf{q} = \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \dashrightarrow \begin{bmatrix} 2.8 & 0.6 \end{bmatrix}
$$

$$
\mathbf{d} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix} \dashrightarrow \begin{bmatrix} 5.2 & 0.4 \end{bmatrix}
$$

Zero ratings in common

Similarity ≠ 0

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., $\mathbf{U^TU=I}$),
    - Λ, Σ are diagonal
- **What is:**
  - $\mathbf{AA^T=}$
  - $\mathbf{A^TA = V\,\Sigma^T\,U^T\,(U\Sigma\,V^T) = V\,\Sigma\Sigma^T\,V^T}$

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., $\mathbf{U^T U = I}$),
    - Λ, Σ are diagonal
- **What is:**
  - $\mathbf{AA^T = U\Sigma\,V^T(U\Sigma\,V^T)^T = U\Sigma\,V^T(V\Sigma^T U^T) = \overset{X}{\downarrow \atop U}\,\overset{\Lambda}{\downarrow \atop \Sigma\Sigma^T}\,\overset{X^T}{\downarrow \atop U^T}}$
  - $\mathbf{A^T A = V\,\Sigma^T\,U^T\,(U\Sigma\,V^T) = \underset{X}{V \atop \uparrow}\,\underset{\Lambda}{\Sigma\Sigma^T \atop \uparrow}\,\underset{X^T}{V^T \atop \uparrow}}$

Shows how to compute SVD using eigenvalue decomposition!

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!**