SVD in Recommender Systems

Aiyana Brown, Addie Gage, Sophia Rabbanian, Hriday Raj

Real-Life Applications of SVD Recommender Systems

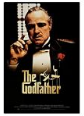
Motivating Example

Our example SVD recommender system is similar to one a streaming service would use, like Netflix. Our matrix includes a few different users and movie offerings.

Motivating Question: Given a dataset of users and movie preferences, how can we use our current information predict unknown values? That is, predict which movies a user should watch if they don't already have a preference.

Motivating Example

Items

Users						
	10	-1	8	10	9	4
	8	9	10	-1	-1	8
	10	5	4	9	-1	-1
	9	10	-1	-1	-1	3
	6	-1	-1	-1	8	10

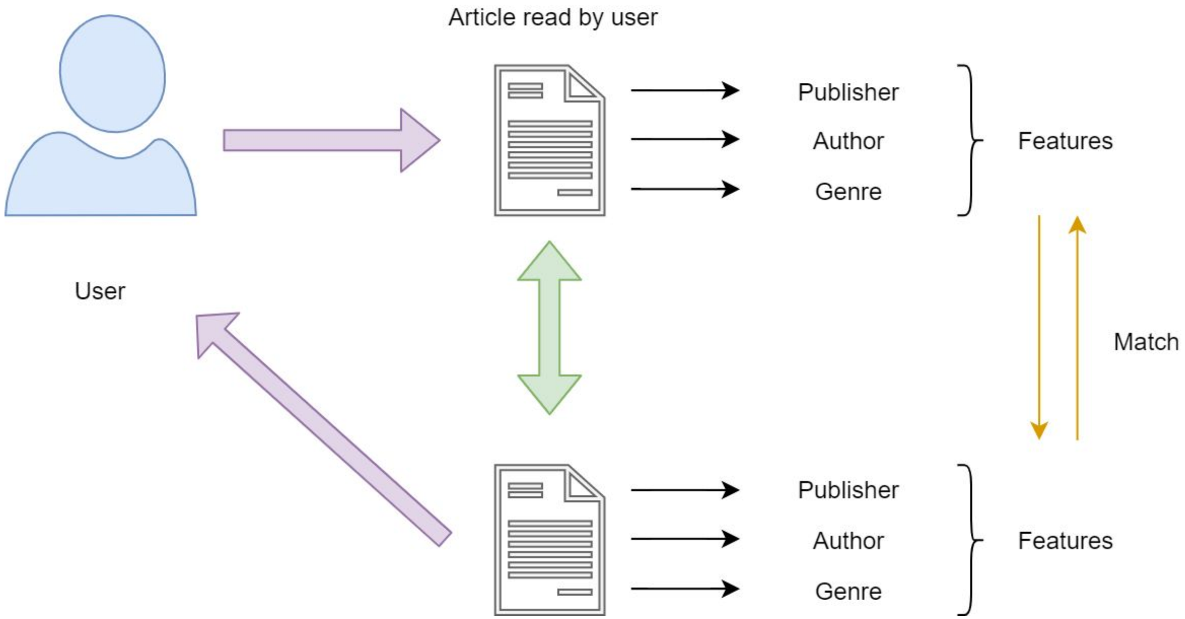
NETFLIX

Which movie should be recommended to Willie?
The Departed or Pulp Fiction?

User-item Interaction matrix

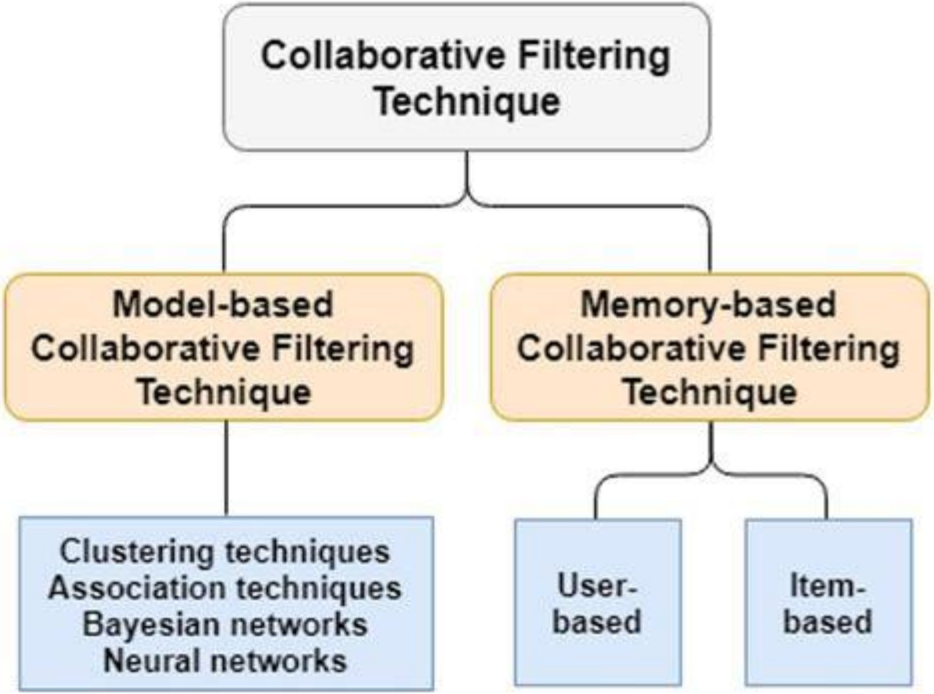
Content based filtering

- Similarity of items are rated on the item's features
- If a user likes an item, another item with similar features will be recommended



Collaborative filtering

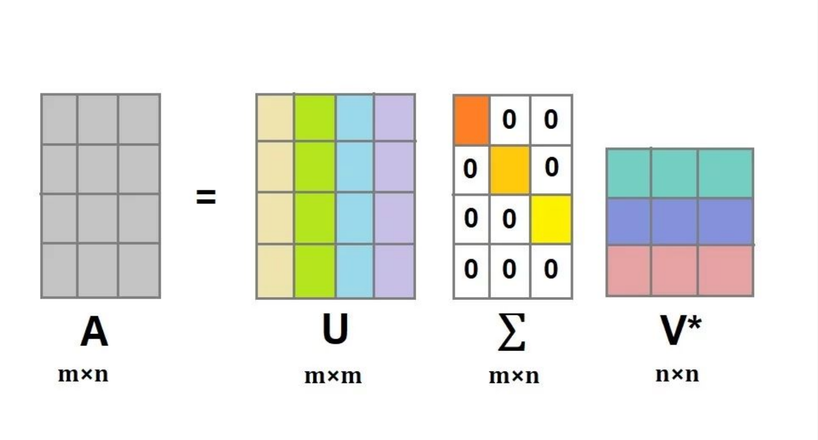
- Most popular method of recommendation
- Model-based is a black box implementation, where you feed it data and train the model to make good recommendations
- User-based memory recommendation judges how similar other users are based on a given user's history
- Item-based memory recommendation systems judges how similar items are based on a given user's history



Recap on SVD

Singular Value Decomposition

- Columns of Matrix U (left singular vectors) form a basis for the user space
- Columns of the Matrix V (right singular vectors) form a basis for the item space
- They span all linear combinations of users and items



SVD breaks down large, complex matrices into smaller, easily usable ones!

Definition: Let A be an n x m matrix. Then there exists orthogonal matrices U and V and pseudo-diagonal matrix Σ such that:
<u>All matrices have this decomposition!</u>

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

U = Left singular vectors

Σ = Singular values

V = Right singular vectors

Lets perform SVD on our matrix →

Matrix A

$$\begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix}$$

What's the first step?

SVD Example

$A = U\Sigma V^T$

Matrix

$$\begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix}$$

Find A^TA & orthogonally diagonalize

V = Right singular vectors: **unit eigenvectors**

Σ = Singular values: **√λ**

$$\mathbf{A^TA} = \begin{pmatrix} 4 & 8 \\ 11 & 7 \\ 14 & -2 \end{pmatrix}\begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix} = \begin{pmatrix} 80 & 100 & 40 \\ 100 & 170 & 140 \\ 40 & 140 & 200 \end{pmatrix}$$

Eigenvectors for $\lambda = 0$: $\begin{pmatrix} 2 \\ -2 \\ 1 \end{pmatrix}$

Eigenvectors for $\lambda = 90$: $\begin{pmatrix} -2 \\ -1 \\ 2 \end{pmatrix}$

Eigenvectors for $\lambda = 360$: $\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$

$$D = \begin{pmatrix} 360 & 0 & 0 \\ 0 & 90 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad P = \begin{pmatrix} \frac{1}{3} & -\frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & -\frac{1}{3} & -\frac{2}{3} \\ \frac{2}{3} & \frac{2}{3} & \frac{1}{3} \end{pmatrix}$$

$$\textit{Right Singular Vectors "V"} = \begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix}\begin{pmatrix} -\frac{2}{3} \\ -\frac{1}{3} \\ \frac{2}{3} \end{pmatrix}\begin{pmatrix} \frac{2}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix}$$

$$\textit{Singular Values "Z"} = \begin{pmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{pmatrix}$$

What’s the next step?

SVD Example

Find A^TA & orthogonally diagonalize
V = Right singular vectors: **unit eigenvectors**
Σ = Singular values: **√λ**
Last Step!

Left singular vectors: **U_i = 1 / σ_i(AV_i)**

Matrix A

$$\begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix}$$

Left Singular Vector

$$U1 = \frac{1}{6\sqrt{10}} \begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix} \begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{3}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} \end{pmatrix}$$

Left Singular Vector

$$U2 = \frac{1}{3\sqrt{10}} \begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix} \begin{pmatrix} \frac{-2}{3} \\ \frac{-1}{3} \\ \frac{2}{3} \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{10}} \\ -\frac{3}{\sqrt{10}} \end{pmatrix}$$

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

Matrix A $= \begin{pmatrix} 4 & 11 & 14 \\ 8 & 7 & -2 \end{pmatrix}$

$$\textit{Right Singular Vectors "V"} = \begin{pmatrix} \frac{1}{3} \\ \frac{2}{3} \\ \frac{2}{3} \end{pmatrix} \begin{pmatrix} -\frac{2}{3} \\ -\frac{1}{3} \\ \frac{2}{3} \end{pmatrix} \begin{pmatrix} \frac{2}{3} \\ -\frac{2}{3} \\ \frac{1}{3} \end{pmatrix}$$

$$\textit{Left Singular Vector "U1"} = \begin{pmatrix} \frac{3}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} \end{pmatrix}$$

$$\textit{Singular Values "Z"} = \begin{pmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{pmatrix}$$

$$\textit{Left Singular Vector "U2"} = \begin{pmatrix} \frac{1}{\sqrt{10}} \\ \frac{-3}{\sqrt{10}} \end{pmatrix}$$

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

Solution:.

$$U = \begin{bmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{bmatrix}, \quad \Sigma = \begin{bmatrix} 6\sqrt{10} & 0 & 0 \\ 0 & 3\sqrt{10} & 0 \end{bmatrix}, \quad V = \begin{bmatrix} 1/3 & -2/3 & 2/3 \\ 2/3 & -1/3 & -2/3 \\ 2/3 & 2/3 & 1/3 \end{bmatrix}$$

Motivating Example

NETFLIX

Items

Users

10	-1	8	10	9	4
8	9	10	-1	-1	8
10	5	4	9	-1	-1
9	10	-1	-1	-1	3
6	-1	-1	-1	8	10

Which movie should be recommended to Willie?
The Departed or Pulp Fiction?

User-item Interaction matrix

SVD with our dataset A^TA

$$\begin{pmatrix} 10 & 8 & 10 & 9 & 6 \\ 0 & 9 & 5 & 10 & 0 \\ 8 & 10 & 4 & 0 & 0 \\ 10 & 0 & 9 & 0 & 0 \\ 9 & 0 & 0 & 0 & 8 \\ 4 & 8 & 0 & 3 & 10 \end{pmatrix} \begin{pmatrix} 10 & 0 & 8 & 10 & 9 & 4 \\ 8 & 9 & 10 & 0 & 0 & 8 \\ 10 & 5 & 4 & 9 & 0 & 0 \\ 9 & 10 & 0 & 0 & 0 & 3 \\ 6 & 0 & 0 & 0 & 8 & 10 \end{pmatrix}$$

$$\begin{pmatrix} 381 & 212 & 200 & 190 & 138 & 191 \\ 212 & 206 & 110 & 45 & 0 & 102 \\ 200 & 110 & 180 & 116 & 72 & 112 \\ 190 & 45 & 116 & 181 & 90 & 40 \\ 138 & 0 & 72 & 90 & 145 & 116 \\ 191 & 102 & 112 & 40 & 116 & 189 \end{pmatrix}$$

$\sigma_1 = 29.47831895$

$\sigma_2 = 14.06486849$

$\sigma_3 = 12.23282051$

$\sigma_4 = 7.8110385$

SVD with our dataset $U\Sigma V^T$

$$U = \begin{pmatrix} 0.56421 & 0.625665 & -0.173491 & -0.148855 & -0.487805 \\ 0.513004 & -0.484748 & 0.207026 & -0.66971 & 0.102345 \\ 0.431453 & -0.0128919 & -0.604567 & 0.24797 & 0.621846 \\ 0.354644 & -0.540641 & 0.0225694 & 0.597584 & -0.473623 \\ 0.32645 & 0.284787 & 0.74902 & 0.332768 & 0.374916 \end{pmatrix}$$

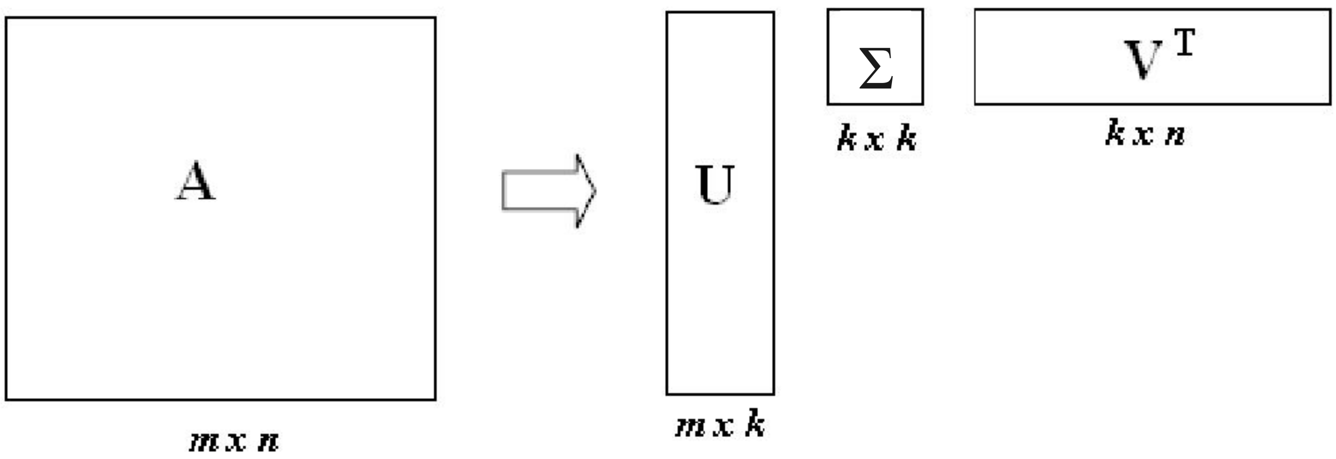
$$\Sigma = \begin{pmatrix} 29.4783 & 0 & 0 & 0 & 0 & 0 \\ 0 & 14.0649 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12.2328 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.81104 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2.134 & 0 \end{pmatrix}$$

$$V = \begin{pmatrix} 0.651705 & -0.0645083 & -0.116663 & 0.385139 & 0.0684458 & -0.635997 \\ 0.350113 & -0.699161 & -0.076344 & 0.152131 & -0.330783 & 0.500206 \\ 0.385691 & 0.00755592 & -0.141908 & -0.882861 & -0.183507 & -0.133899 \\ 0.323125 & 0.436593 & -0.586619 & 0.0951445 & 0.336723 & 0.488282 \\ 0.260852 & 0.562343 & 0.362201 & 0.169305 & -0.651787 & 0.176198 \\ 0.362616 & -0.0106208 & 0.696499 & -0.106601 & 0.560369 & 0.24064 \end{pmatrix}$$

Dimensionality Reduction

From SVD, you reduce U, Σ, and V by keeping the first **k** singular values in Σ and removing **r-k** columns from U while removing **r-k** rows from V.

By performing dimensionality reduction, we increase the computational efficiency of the recommender system and this also filter out noise.



Dimension Reduction With Our Movie Dataset

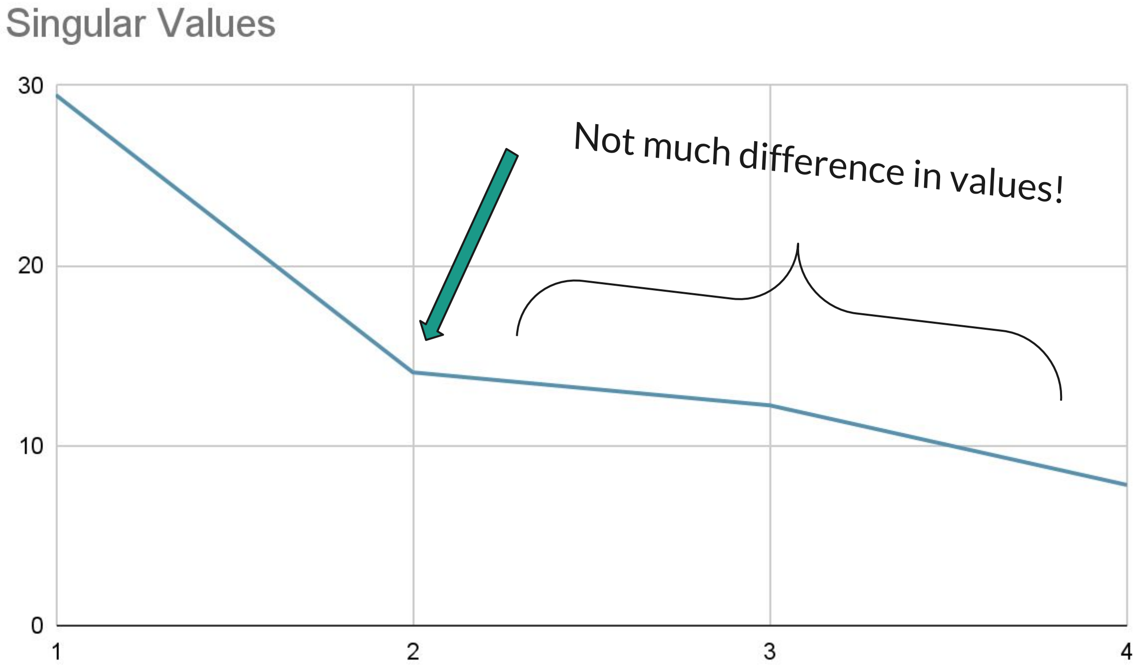
k = 2

$\sigma_1 = 29.47831895$

$\sigma_2 = 14.06486849$

$\sigma_3 = 12.23282051$

$\sigma_4 = 7.8110385$



Dimension Reduction with our dataset

$$U = \begin{pmatrix} \boxed{\begin{matrix} 0.56421 & 0.625665 \\ 0.513004 & -0.484748 \\ 0.431453 & -0.0128919 \\ 0.354644 & -0.540641 \\ 0.32645 & 0.284787 \end{matrix}} & \begin{matrix} -0.173491 & -0.148855 & -0.487805 \\ 0.207026 & -0.66971 & 0.102345 \\ -0.604567 & 0.24797 & 0.621846 \\ 0.0225694 & 0.597584 & -0.473623 \\ 0.74902 & 0.332768 & 0.374916 \end{matrix} \end{pmatrix}$$

$$\Sigma = \begin{pmatrix} \boxed{\begin{matrix} 29.4783 & 0 \\ 0 & 14.0649 \end{matrix}} & \begin{matrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \\ \begin{matrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{matrix} & \begin{matrix} 12.2328 & 0 & 0 & 0 \\ 0 & 7.81104 & 0 & 0 \\ 0 & 0 & 2.134 & 0 \end{matrix} \end{pmatrix}$$

$$V = \begin{pmatrix} \boxed{\begin{matrix} 0.651705 & -0.0645083 & -0.116663 & 0.385139 & 0.0684458 & -0.635997 \\ 0.350113 & -0.699161 & -0.076344 & 0.152131 & -0.330783 & 0.500206 \end{matrix}} \\ \begin{matrix} 0.385691 & 0.00755592 & -0.141908 & -0.882861 & -0.183507 & -0.133899 \\ 0.323125 & 0.436593 & -0.586619 & 0.0951445 & 0.336723 & 0.488282 \\ 0.260852 & 0.562343 & 0.362201 & 0.169305 & -0.651787 & 0.176198 \\ 0.362616 & -0.0106208 & 0.696499 & -0.106601 & 0.560369 & 0.24064 \end{matrix} \end{pmatrix}$$

Dimensionally reduced matrix

$$\begin{bmatrix} 10.27146 & -0.32948 & 6.48129 & 9.21617 & 9.28705 & 5.93755 \\ 10.29522 & 10.06140 & 5.78110 & 1.90980 & 0.11073 & 5.55607 \\ 8.30040 & 4.57969 & 4.90405 & 4.03050 & 3.21569 & 4.61386 \\ 7.30365 & 8.97664 & 3.97469 & 0.05818 & 1.54904 & 3.87167 \\ 6.01310 & 0.56873 & 3.74185 & 4.85826 & 4.76269 & 3.44699 \end{bmatrix}$$

User-based Similarity

Reduced matrix $\mathbf{U}_k$ is used

Each row = 1 user

Comparing two users (rows):

- Cosine-based similarity
- Euclidean distances in k-dimensional space

Cosine-based similarity: Suppose a, b are two users

Measure similarity between users a and b using:

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Item-based Similarity

Reduced matrix $\mathbf{V}_k$ is used

Each row = 1 item

Items do not change as frequently as users do :

- Available for pre-computation
- Better run-time performance

Comparing two items (rows) in the $\mathbf{V}_k$ matrix:

- Cosine-based similarity
- Euclidean distances in k-dimensional space

Cosine Similarities between users

$$\begin{pmatrix} 1 & -0.02328861 & 0.64721401 & -0.25363852 & 0.99284923 \\ -0.02328861 & 1 & 0.74702889 & 0.97294362 & 0.09622064 \\ 0.64721401 & 0.74702889 & 1 & 0.57322176 & 0.73358656 \\ -0.25363852 & 0.97294362 & 0.57322176 & 1 & -0.13635338 \\ 0.99284923 & 0.09622064 & 0.73358656 & -0.13635338 & 1 \end{pmatrix}$$

Now that we have our dimensionally reduced matrix and cosine similarity matrix, what's next?

Calculating Predicted Ratings

$$P_{ui} = \frac{\sum_{v \in I_i} S_{uv} \cdot R_{vi}}{\sum_{v \in I_i} |S_{uv}|}$$

R is the ratings matrix where R_{ui} denotes the rating given by user u to item i.
S is the user similarity matrix where S_{uv} is the similarity score between users u and v.
I_i is the set of users who have rated item i.

Numerator: $\sum_{v \in I_i} S_{uv} \cdot R_{vi}$ — This is the sum of products of the similarity between the target user u and each other user v who has rated the item i, and the rating that user v has given to item i.

Denominator: $\sum_{v \in I_i} |S_{uv}|$ — This is the sum of the absolute values of the similarities between the target user u and all other users v who have rated the item. This acts as a normalization factor.

Conditional Execution: If the denominator is zero (which might occur if no similar users have rated the item i), the predicted rating P_{ui} is typically set to zero or some default value, as similarity-based prediction cannot be computed.

Predicting ratings- our data

$$\begin{pmatrix} 1 & -0.02328861 & 0.64721401 & -0.25363852 & 0.99284923 \\ -0.02328861 & 1 & 0.74702889 & 0.97294362 & 0.09622064 \\ 0.64721401 & 0.74702889 & 1 & 0.57322176 & 0.73358656 \\ -0.25363852 & 0.97294362 & 0.57322176 & 1 & -0.13635338 \\ 0.99284923 & 0.09622064 & 0.73358656 & -0.13635338 & 1 \end{pmatrix}$$

$$\begin{bmatrix} 10.27146 & -0.32948 & 6.48129 & 9.21617 & 9.28705 & 5.93755 \\ 10.29522 & 10.06140 & 5.78110 & 1.90980 & 0.11073 & 5.55607 \\ 8.30040 & 4.57969 & 4.90405 & 4.03050 & 3.21569 & 4.61386 \\ 7.30365 & 8.97664 & 3.97469 & 0.05818 & 1.54904 & 3.87167 \\ 6.01310 & 0.56873 & 3.74185 & 4.85826 & 4.76269 & 3.44699 \end{bmatrix}$$

We use these two matrices to get our final matrix:

$$\begin{pmatrix} 6.84273083 & 0.53031657 & 6.19931379 & 9.60708566 & 8.50179411 & 5.71918365 \\ 8.65341681 & 8.25912048 & 7.23136199 & 8.42558282 & 4.68723239 & 5.6335237 \\ 8.6485932 & 7.52309998 & 6.9533467 & 9.39291434 & 8.46872374 & 6.52361492 \\ 6.52596548 & 8.49222189 & 5.55239106 & 3.17177004 & 8.65036868 & 3.55721133 \\ 7.70762653 & 3.28142629 & 6.49565452 & 9.5750861 & 8.49820589 & 6.44016902 \end{pmatrix}$$

Final Predicted Ratings

Which movie should be recommended to Willie?
The Departed or Pulp Fiction?

$$\begin{pmatrix} 6.84273083 & 0.53031657 & 6.19931379 & 9.60708566 & 8.50179411 & 5.71918365 \\ 8.65341681 & 8.25912048 & 7.23136199 & 8.42558282 & 4.68723239 & 5.6335237 \\ 8.6485932 & 7.52309998 & 6.9533467 & 9.39291434 & 8.46872374 & 6.52361492 \\ 6.52596548 & 8.49222189 & 5.55239106 & 3.17177004 & 8.65036868 & 3.55721133 \\ 7.70762653 & 3.28142629 & 6.49565452 & 9.5750861 & 8.49820589 & 6.44016902 \end{pmatrix}$$

The algorithm suggests The Departed would be a better movie for Willie

Benefits of SVD

vs

Limitations

Simplifies datasets, making patterns more discernible and calculations faster.

Enhances the accuracy of predictions, tailoring recommendations to individual preferences.

SVD complexity can strain resources, particularly with very large data sets

Struggles with new users/items (cold start) and rapidly updating preferences.

Sources

Our main source:

★ Osmanlı, O. N. (2010). *A singular value decomposition approach for recommendation systems* [M.S. - Master of Science]. Middle East Technical University. https://open.metu.edu.tr/handle/11511/19640

Others of note:

- Enhancing Recommender Systems with NLP-based Biased
- Singular Value Decomposition | Semantic Scholar
- Recommender systems: a novel approach based on singular value decomposition | Semantic Scholar
- Recommendation Algorithm with SVD - Jake Tae
- Recommender System — singular value decomposition (SVD) & truncated SVD | by Denise Chen | Towards Data Science
- https://realpython.com/build-recommendation-engine-collaborative-filtering/

Thank you!