BROWN
Computer Science

# Lecture 20: Dimensionality Reduction:

Based on the lecture notes from:
Mining of Massive Datasets
Jure Leskovec, Anand Rajaraman, Jeff Ullman
Stanford University
http://www.mmds.org

# What is dimensionality reduction?

- Often 1000s or (100s of 1000s) of features
- Many (most) are useless or redundant
- Keeping track of all of them would explode the complexity of our models

| Clicks | Recency | Reading Level | Photo | Title: "new" | Title: "tax" | Title: "this" | … |
|---|---|---|---|---|---|---|---|
| 10 | 1.3 | 11 | 1 | 1 | 0 | 0 | … |
| 1000 | 1.7 | 3 | 1 | 0 | 0 | 1 | … |
| 1000000 | 2.4 | 2 | 1 | 0 | 0 | 1 | … |
| 1 | 5.9 | 19 | 0 | 0 | 0 | 0 | … |

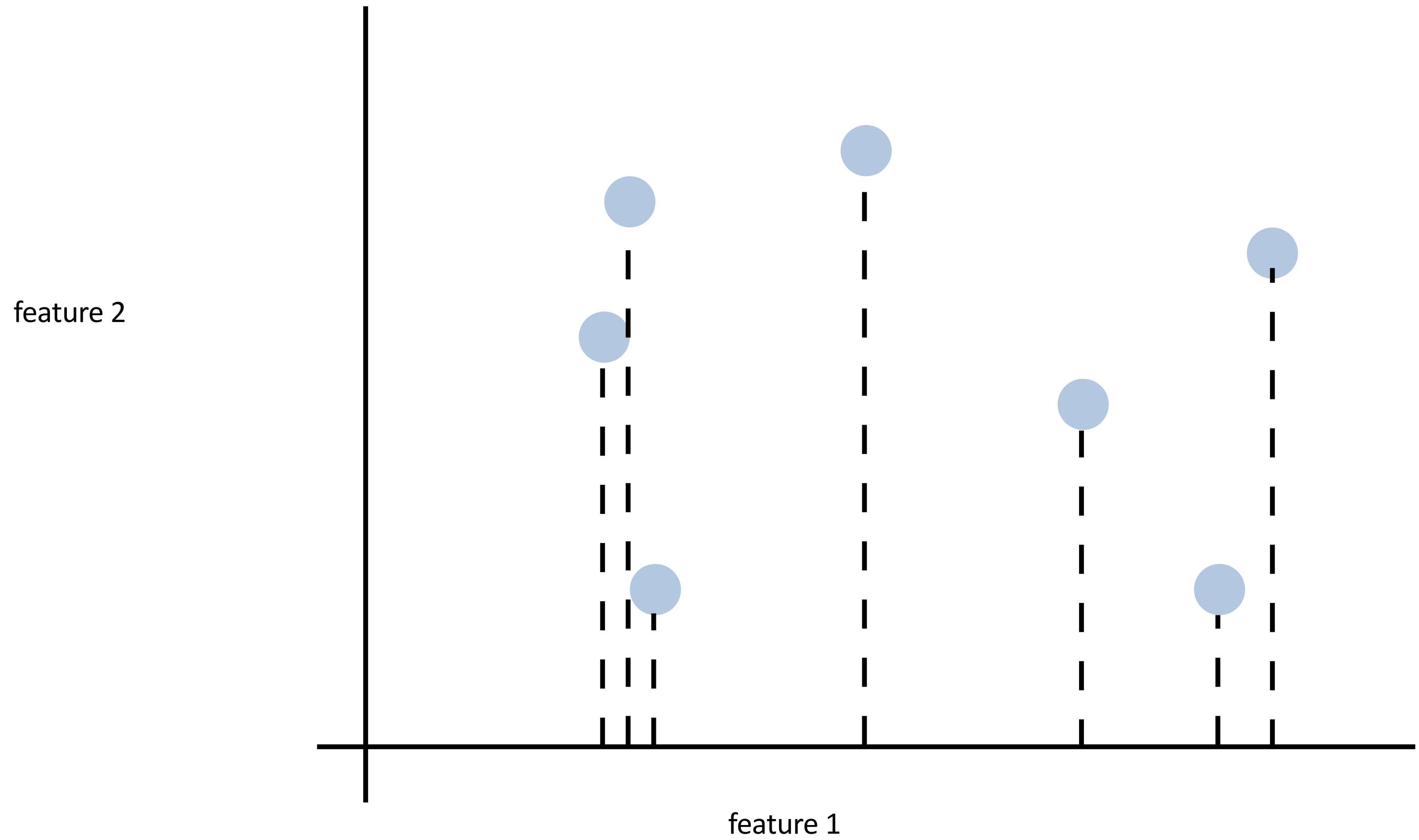
# What is dimensionality reduction?

Keeping track of all of them would explode the complexity of our models

- slower to train
- easier to overfit
- harder to visualize/interpret

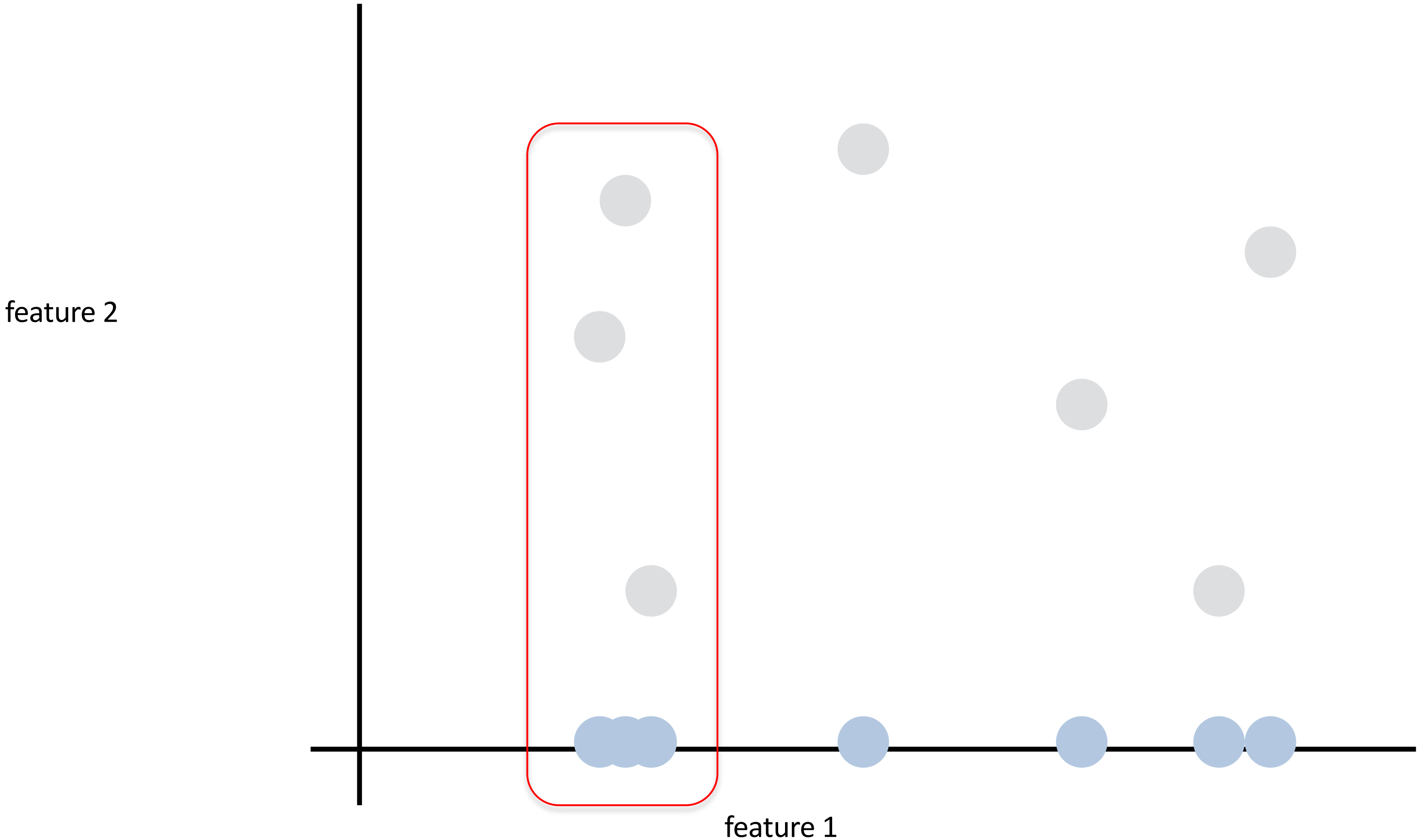
| Clicks | Recency | Reading Level | Photo | Title: "new" | Title: "tax" | Title: "this" | … |
|---|---|---|---|---|---|---|---|
| 10 | 1.3 | 11 | 1 | 1 | 0 | 0 | … |
| 1000 | 1.7 | 3 | 1 | 0 | 0 | 1 | … |
| 1000000 | 2.4 | 2 | 1 | 0 | 0 | 1 | … |
| 1 | 5.9 | 19 | 0 | 0 | 0 | 0 | … |

# Main idea



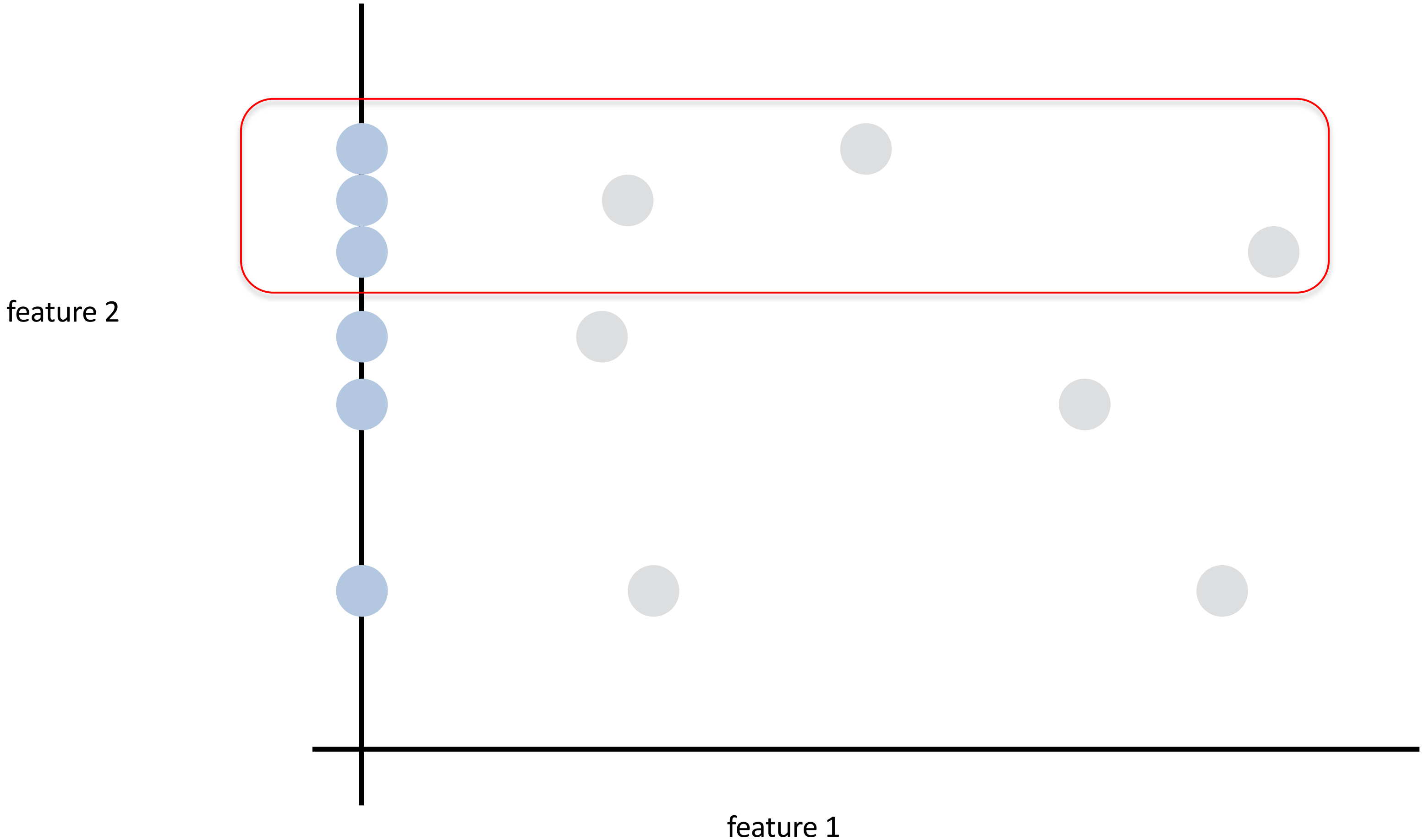
# Main idea

- Reduce the number of features/dimensions
- Losing some distinction between points



# Main idea

- We can try different ways of reducing dimension
- Some may lead to losing more distinction/differences than others

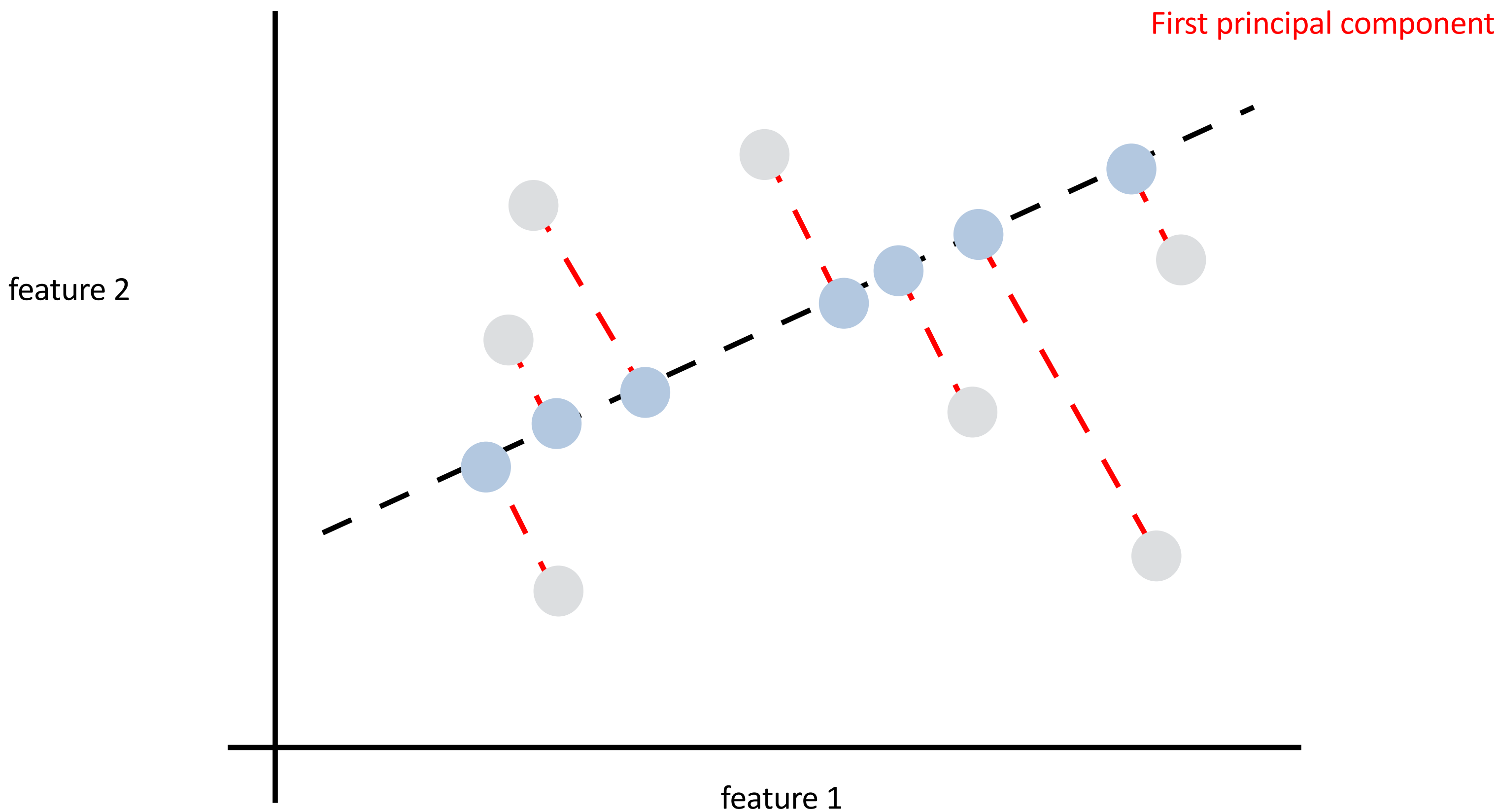


# Principle Component Analysis (PCA)

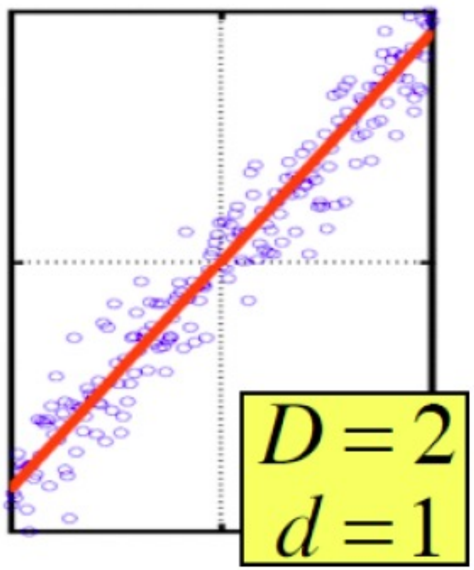
Pick a line…

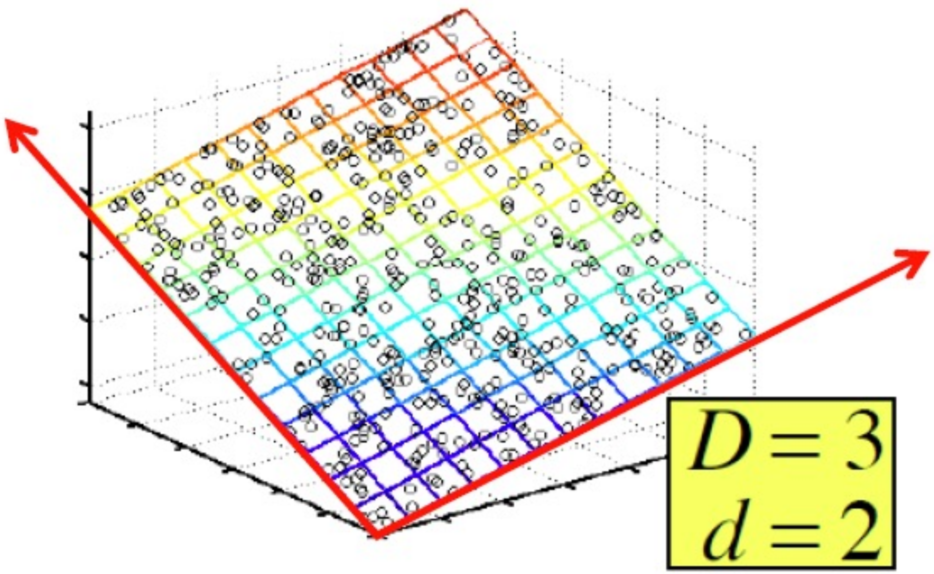
… such that the projection of the other points on it…

… has maximum distinction / variance



# 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

| day / customer | We 7/10/96 | Th 7/11/96 | Fr 7/12/96 | Sa 7/13/96 | Su 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]

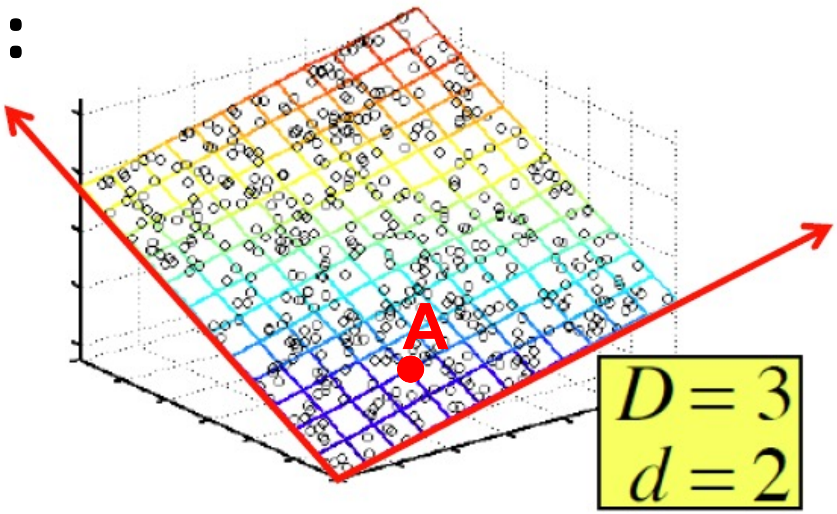
# Rank of a Matrix

- The rank of a matrix is the number of its linearly independent columns
- 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:

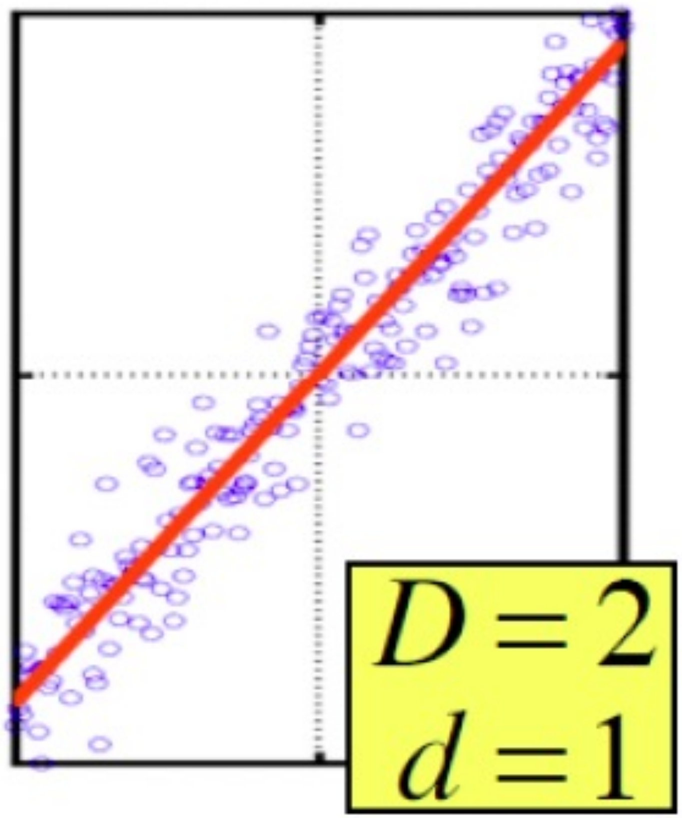
**1 row per point:** $\begin{array}{cc} \begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix} & \begin{array}{c} \mathbf{A} \\ \mathbf{B} \\ \mathbf{C} \end{array} \end{array}$




- **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]
    - **We reduced the number of coordinates/dimensions of the representation!**

# Dimensionality Reduction

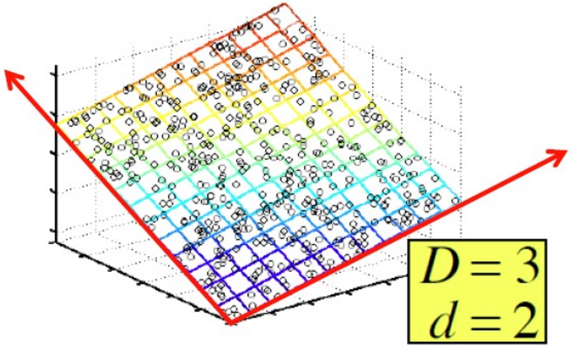
- Goal of dimensionality reduction is to discover the axis of data!




- 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?

- 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



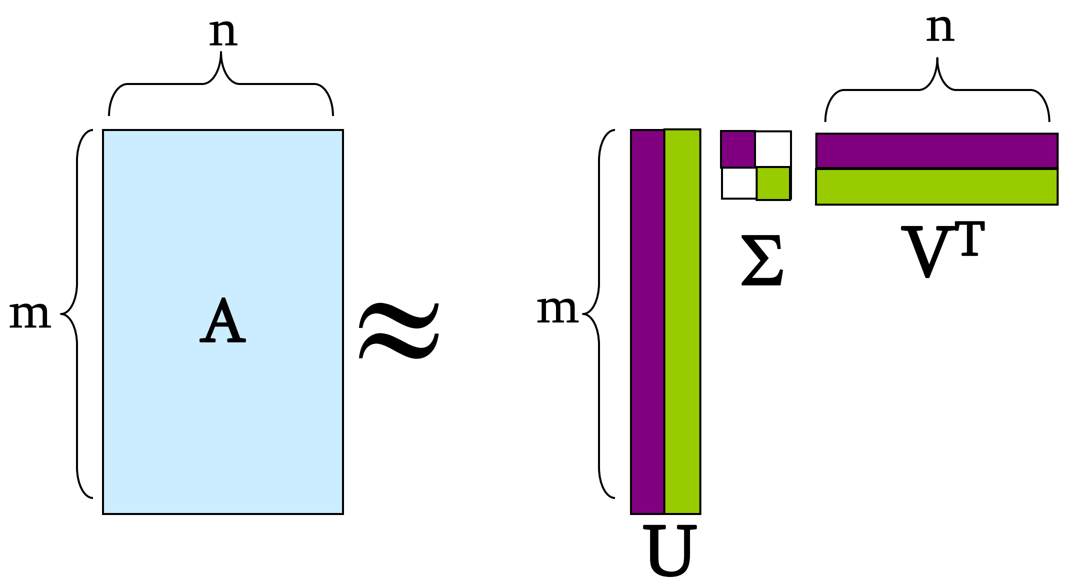
# SVD - Definition

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

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

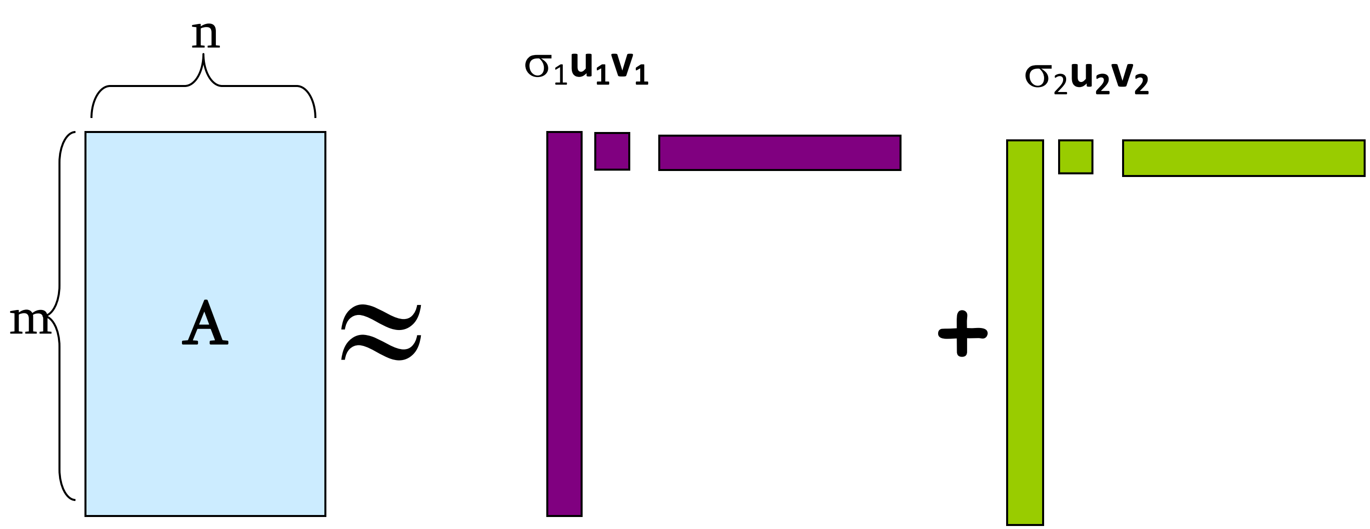
# SVD

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



# SVD

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




**$\sigma_i$ … scalar**

**$u_i$ … vector**

**$v_i$ … vector**

# SVD - Properties

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

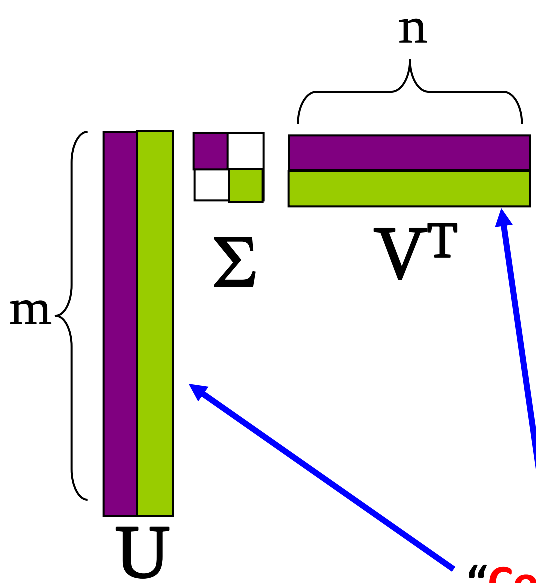
- $\boldsymbol{U}, \Sigma, \boldsymbol{V}$: unique
- $\boldsymbol{U}, \boldsymbol{V}$: column orthonormal
  - $\boldsymbol{U}^T \boldsymbol{U} = \boldsymbol{I}; \boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}$ ($\boldsymbol{I}$: identity matrix)
  - Columns are orthogonal unit vectors
- $\Sigma$: diagonal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

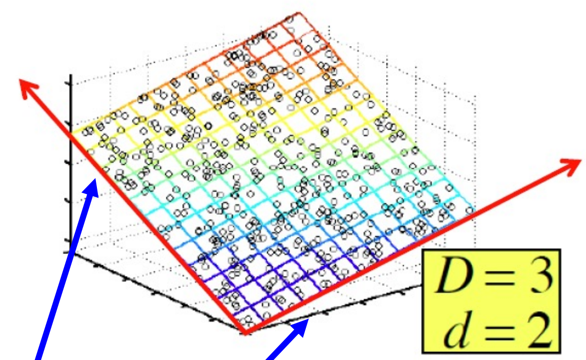
Nice proof of uniqueness: http://www.mpi-inf.mpg.de/~bast/ir-seminar-ws04/lecture2.pdf

# SVD – Example: Users-to-Movies

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

$$
\begin{array}{c}
 \\ \\ \text{SciFi} \\ \\ \\ \text{Romnce} \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
$$




**"Concepts"**
**AKA Latent dimensions**
**AKA Latent factors**

# SVD – Example: Users-to-Movies

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

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

SciFi-concept

Romance-concepts

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

# SVD – Example: Users-to-Movies

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

SciFi-concept

“strength” of the SciFi-concept

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

# SVD – Example: Users-to-Movies

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

***V* is "movie-to-concept" similarity matrix**

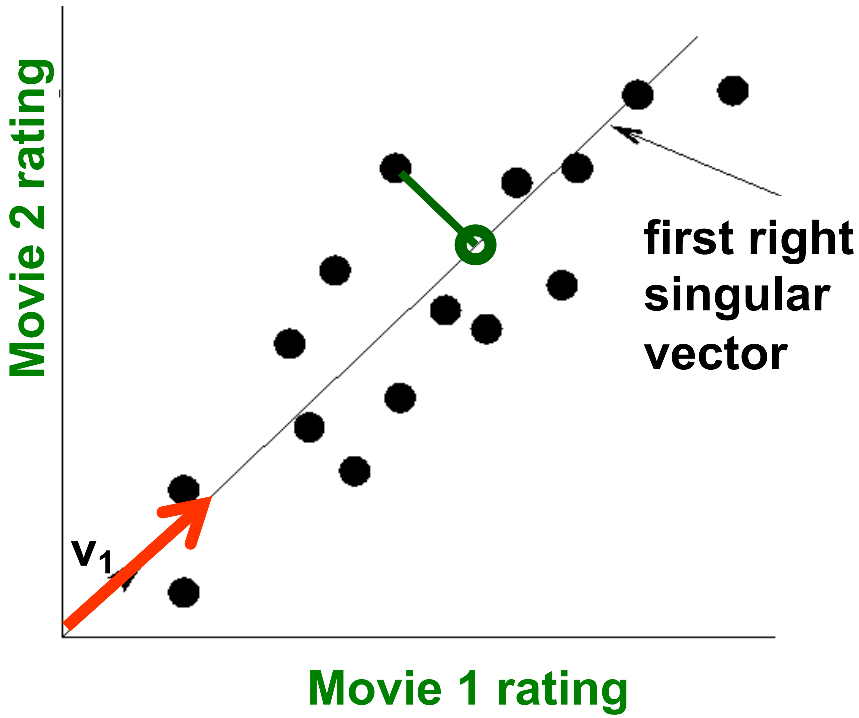
$$
\begin{array}{cc}
 & \begin{matrix}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{matrix} \\
\begin{matrix} \\ \text{SciFi} \\ \\ \\ \\ \text{Romnce} \\ \\ \end{matrix} &
\begin{bmatrix}
\mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\
\mathbf{3} & \mathbf{3}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

# 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

# SVD – Dimensionality Reduction



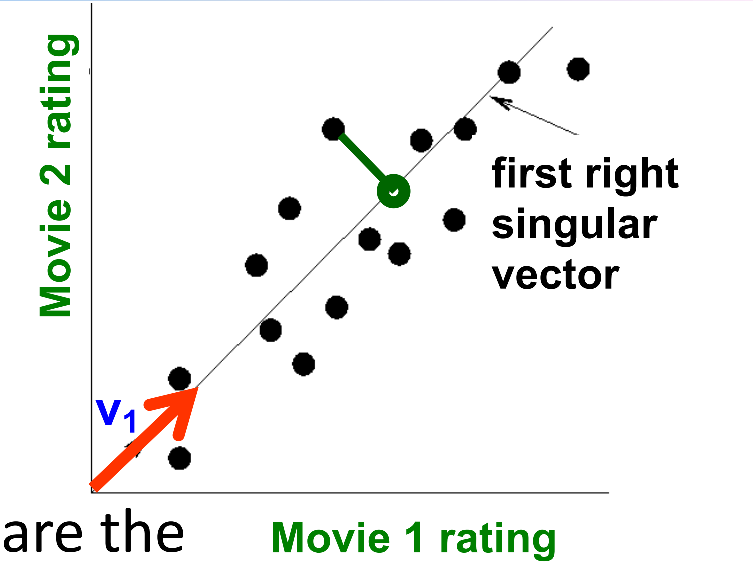

- Instead of using two coordinates $(x, y)$ to describe point locations, let's use only one coordinate $(z)$
- Point's position is its location along vector $v_1$
- **How to choose $v_1$? Minimize reconstruction error**

# SVD – Dimensionality Reduction

- Goal: Minimize the sum of reconstruction errors:

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

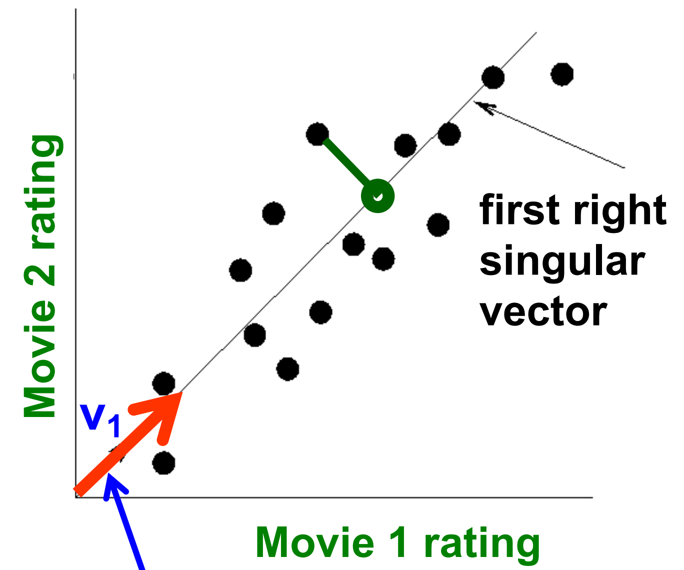
  - where $\boldsymbol{x_{ij}}$ are the "old" and $\boldsymbol{z_{ij}}$ are the "new" coordinates




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

# SVD - Interpretation #2

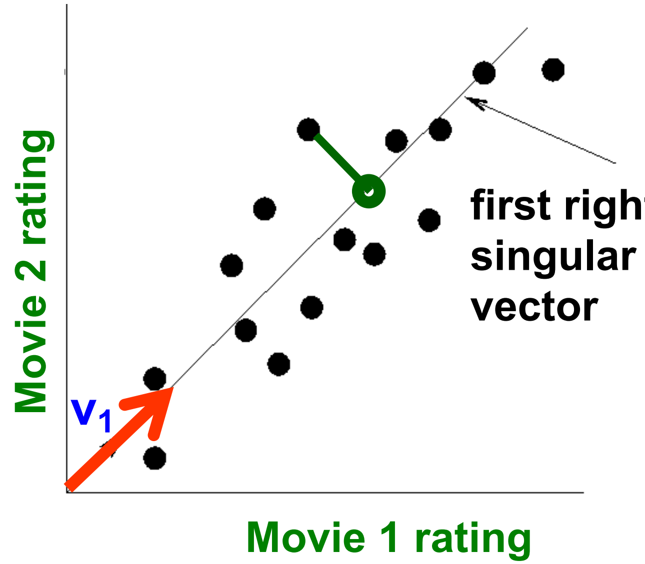
- **$A = U \Sigma V^T$ - example:**
  - **V**: “movie-to-concept” matrix
  - **U**: “user-to-concept” matrix




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

# SVD - Interpretation #2

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

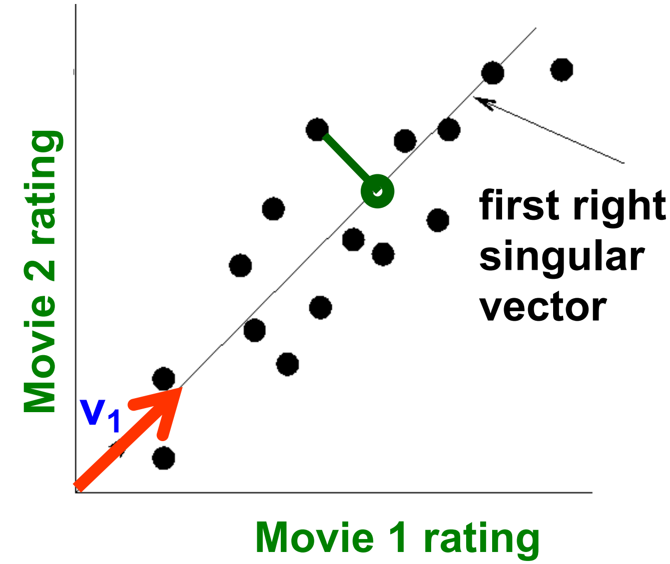



variance ('spread')
on the $v_1$ axis

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

# SVD - Interpretation #2

- **U Σ**: 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 Σ)$^T$:**

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

# SVD - Interpretation #2

- How exactly is dim. reduction done?
- Set smallest singular values to zero

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

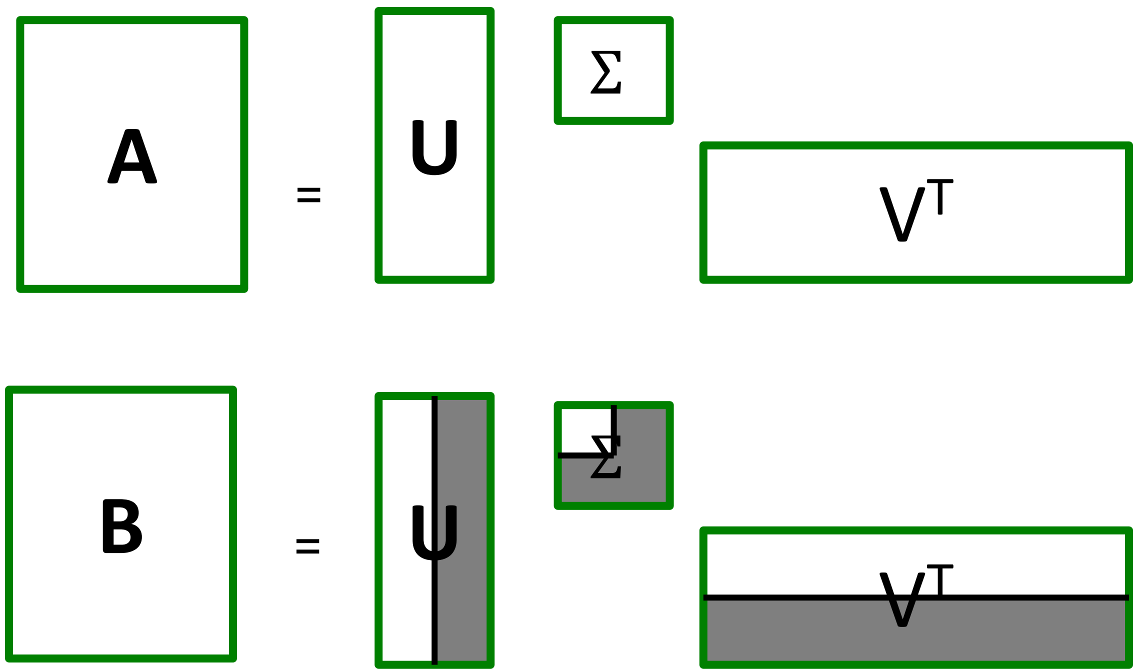
27

# SVD - Interpretation #2

- How exactly is dim. reduction done?
- Set smallest singular values to zero

$$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ \mathbf{3} & \mathbf{3} & \mathbf{3} & 0 & 0 \\ \mathbf{4} & \mathbf{4} & \mathbf{4} & 0 & 0 \\ \mathbf{5} & \mathbf{5} & \mathbf{5} & 0 & 0 \\ 0 & \mathbf{2} & 0 & \mathbf{4} & \mathbf{4} \\ 0 & 0 & 0 & \mathbf{5} & \mathbf{5} \\ 0 & \mathbf{1} & 0 & \mathbf{2} & \mathbf{2} \end{bmatrix} \approx \begin{bmatrix} \mathbf{0.13} & 0.02 \\ \mathbf{0.41} & 0.07 \\ \mathbf{0.55} & 0.09 \\ \mathbf{0.68} & 0.11 \\ 0.15 & \mathbf{-0.59} \\ 0.07 & \mathbf{-0.73} \\ 0.07 & \mathbf{-0.29} \end{bmatrix} \times \begin{bmatrix} \mathbf{12.4} & 0 \\ 0 & \mathbf{9.5} \end{bmatrix} \times \begin{bmatrix} \mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\ 0.12 & -0.02 & 0.12 & \mathbf{-0.69} & \mathbf{-0.69} \end{bmatrix}$$

# SVD – Best Low Rank Approx.




B is a good approximation of A

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

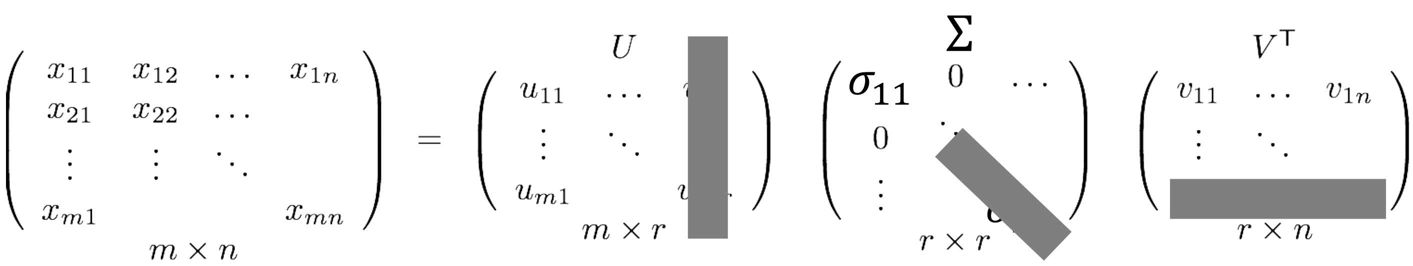
# SVD – Best Low Rank Approx.

**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**)=$k$ approxIMATION to $A$

What do we mean by "best":

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



# SVD

**Q: How many σs to keep?**

**A:** Rule-of-a thumb:

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

# Dimensionality Reduction

- "Low Rank Assumption": we typically assume that our features contain a large amount of redundant information
- We can throw away a lot of principle components without losing too much of the signal needed for our task

# Matrices IRL

- Data is noisy, so M is most likely full-rank
- We assume that M is *close to* a low rank matrix, and we approximate the matrix it is close to
- Viewed as a "de-noised" version of M
- "Original matrix exhibits redundancy and noise, low-rank reconstruction exploits the former to remove the latter"*

# Matrices IRL

- Data is also often incomplete…missing values, new observations, etc.
- Can we use SVD for this?
- Yes! Though we need to make a few changes to complete the matrix

# SVD - Complexity

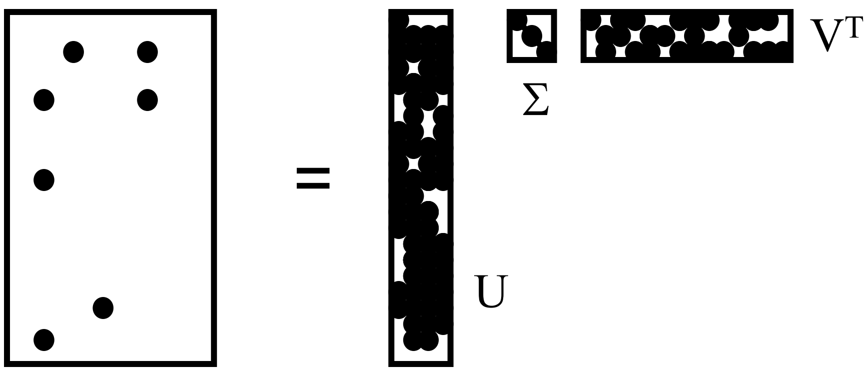
- To compute SVD:
  - $O(nm^2)$ or $O(n^2m)$ (whichever is less)
  - https://www.cs.utexas.edu/users/inderjit/public_papers/HLA_SVD.pdf
- But:
  - Less work, if we just want singular values
  - or if we want first *k* singular vectors
  - or if the matrix is sparse
- Implemented in linear algebra packages like
  - LINPACK, Matlab, SPlus, Mathematica ...

# SVD - Conclusions so far

- SVD: $\mathbf{A} = \mathbf{U}\,\Sigma\,\mathbf{V}^T$: **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

# SVD: Pros and Cons

- \+ **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**



# 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 (**U^T^U=I**),
    - $\Lambda$, $\Sigma$ are diagonal
- **Now let's calculate:**
  - **$AA^T = U\Sigma\,V^T(U\Sigma\,V^T)^T =$**
  - **$A^TA = V\,\Sigma^T\,U^T\,(U\Sigma\,V^T) =$**

Shows how to compute SVD using eigenvalue decomposition!

$X\,\Lambda^2\;\;X^T$

$X\,\Lambda^2\;\;X^T$

# Case study: How to query?

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

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

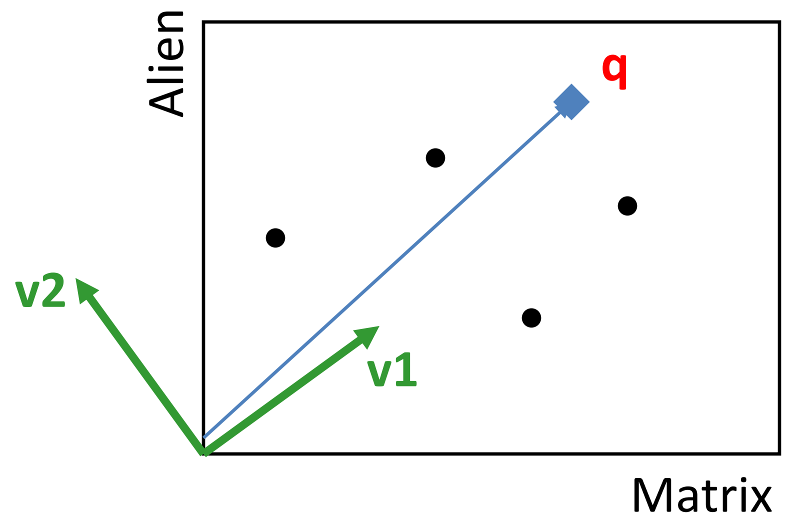
# Case study: How to query?

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

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

$$\mathbf{q} = \begin{bmatrix} 5 & 0 & 0 & 0 & 0 \end{bmatrix}$$

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

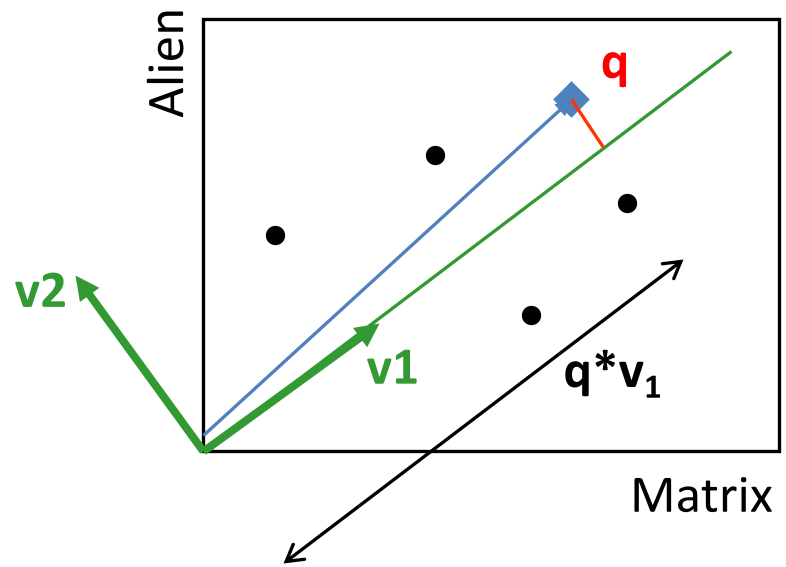


# Case study: How to query?

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

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

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



# Case study: How to query?

Compactly, we have:

$q_{concept} = q \times V$

**E.g.:**

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

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

**movie-to-concept similarities (V)**

SciFi-concept

# Case study: How to query?

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

$$\mathbf{d} = \begin{array}{c} \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\ \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \end{array} \mathbf{x} \begin{bmatrix} 0.56 & 0.12 \\ 0.59 & -0.02 \\ 0.56 & 0.12 \\ 0.09 & -0.69 \\ 0.09 & -0.69 \end{bmatrix} = \begin{array}{c} \begin{array}{cc} \text{SciFi-concept} & \end{array} \\ \begin{bmatrix} 5.2 & 0.4 \end{bmatrix} \end{array}$$

**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**!

$$
\begin{array}{rccccc}
 & \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \\
\mathbf{d} = & 0 & 4 & 5 & 0 & 0 \\
\mathbf{q} = & 5 & 0 & 0 & 0 & 0
\end{array}
$$

$$
\mathbf{d} = \begin{bmatrix} 0 & 4 & 5 & 0 & 0 \end{bmatrix} \dashrightarrow \begin{bmatrix} \overset{\text{SciFi-concept}}{5.2} & 0.4 \end{bmatrix}
$$

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

**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!**