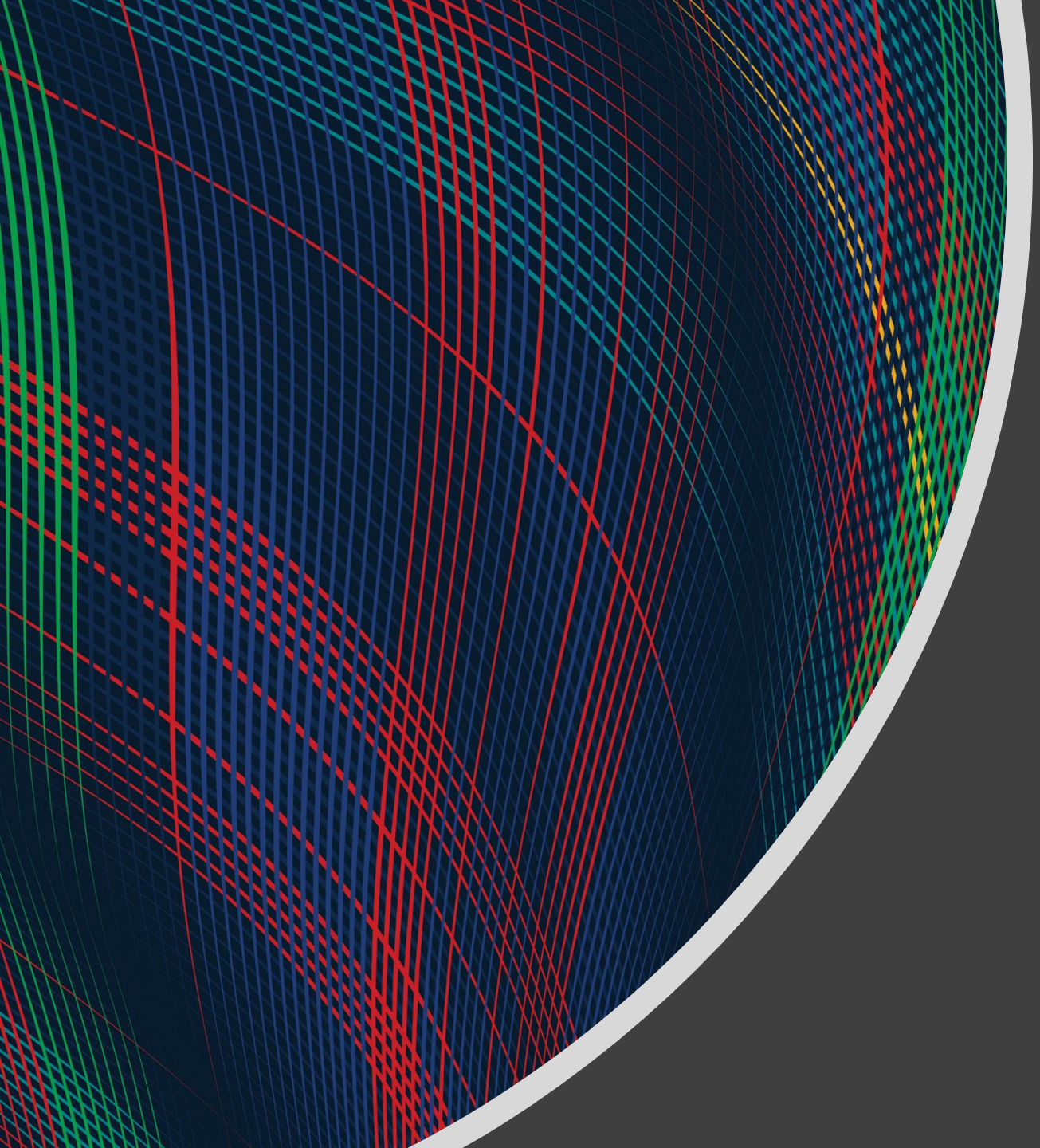
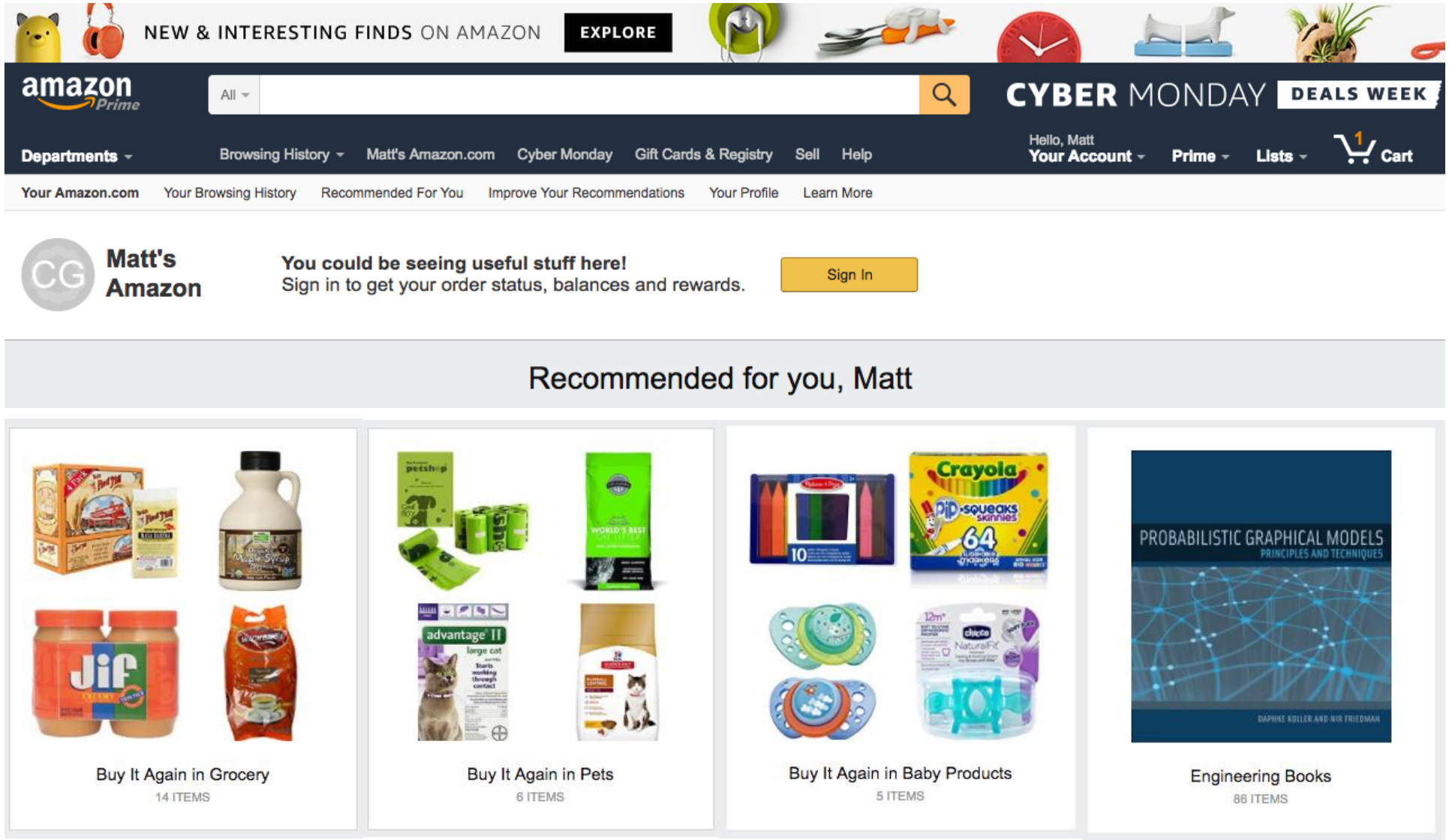


10-315 Introduction to ML

Recommender Systems

Instructor: Pat Virtue

Recommender Systems



Slide credit: CMU MLD Matt Gormley

Recommender Systems

Slide credit: CMU MLD Matt Gormley

ML System Design: Movie Recommendation

Task

$h(i, j) \rightarrow \hat{r}$ (userid, itemid, rating)

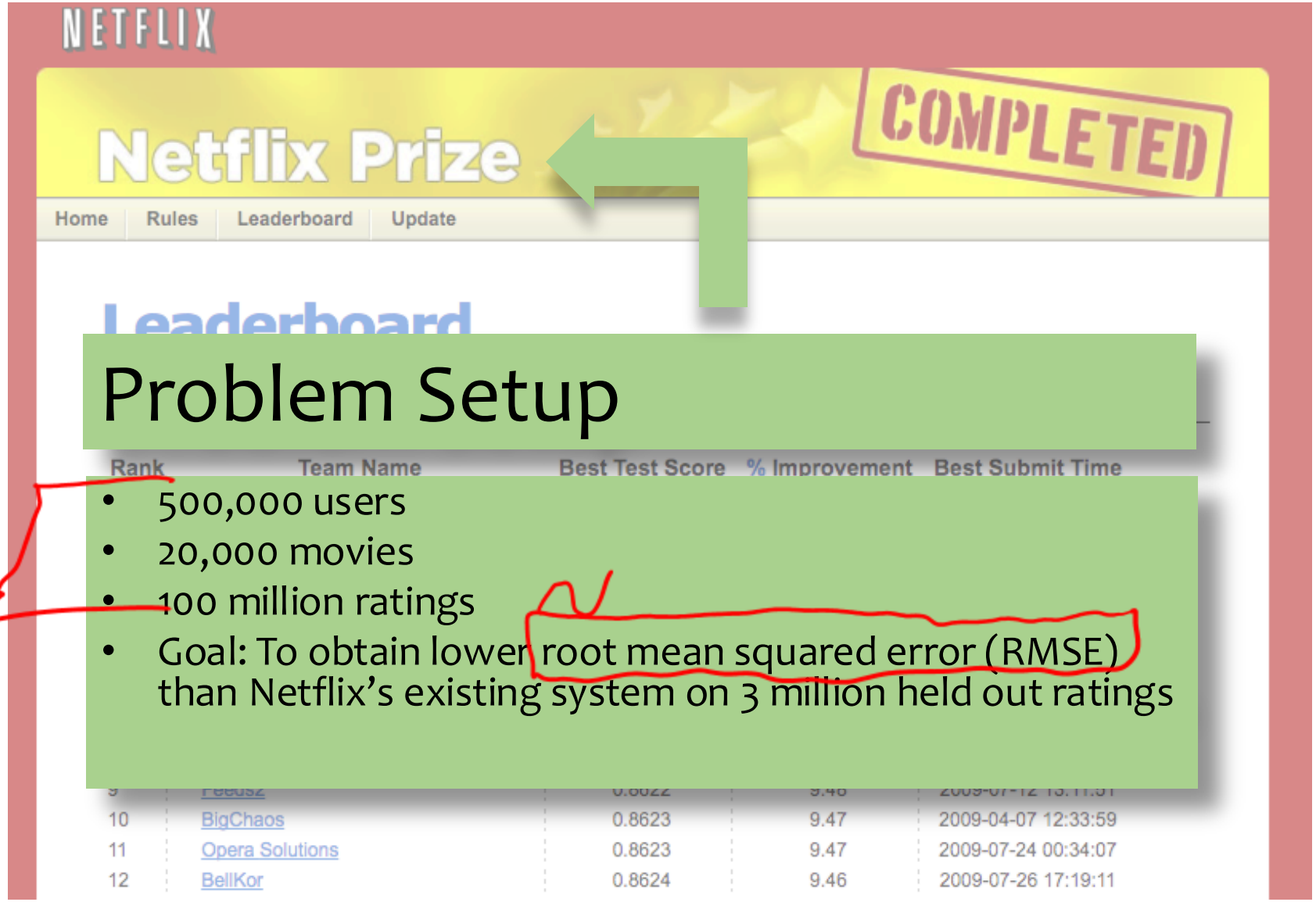
Experience

$\{(i, j, r)\}_{i=1}^{N}$

Performance measure

$\ell(r, \hat{r}) = (r - \hat{r})^2$ (4, 3, 2)

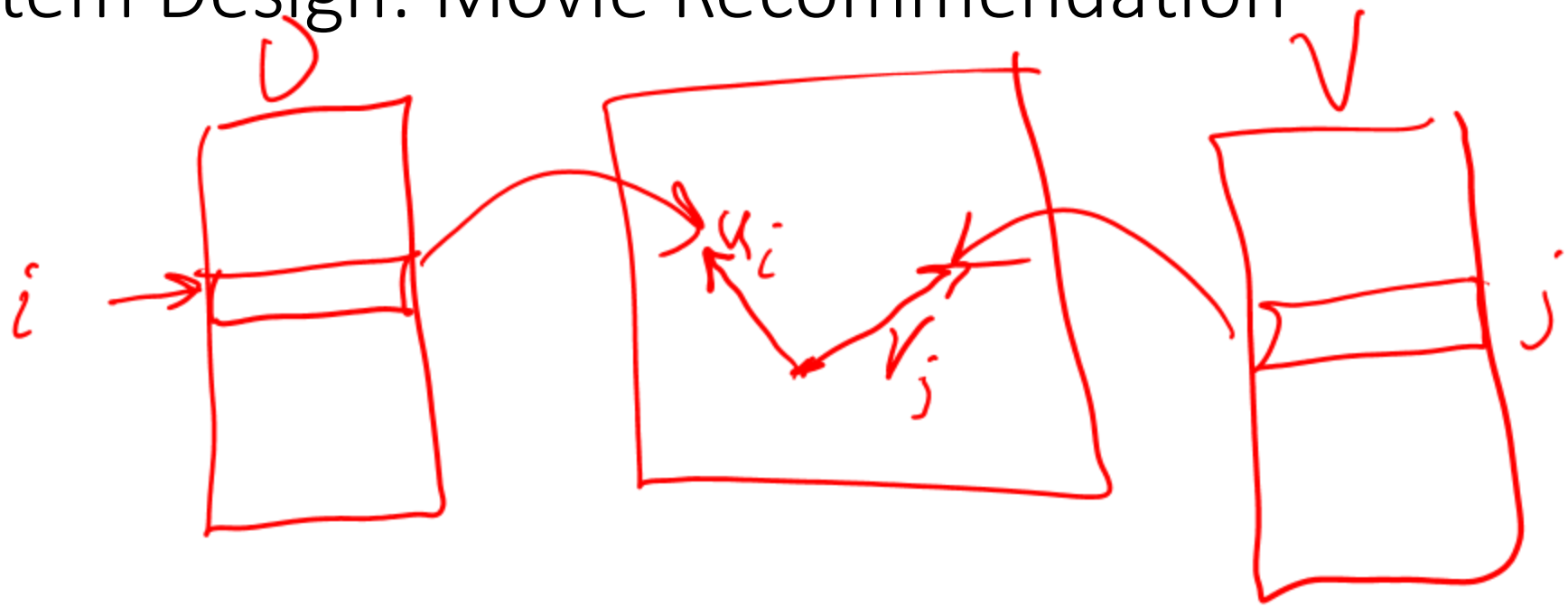
Recommender Systems



Slide credit: CMU MLD Matt Gormley

ML System Design: Movie Recommendation

Model



$$\hat{r}_{ij} \leftarrow h(i,j) = \vec{u}_i^{\,T} \vec{v}_j$$

$$J(U,V) = \sum_{ij\,r \in S} \left(\| r_{ij} - \vec{u}_i^{\,T} v_j \| \right)^2$$

ML System Design: Movie Recommendation

Model

$$\|R - UV^T\|_F^2$$

Recommender Systems

Setup:

- Items:
  movies, songs, products, etc.
  (often many thousands)
- Users:
  watchers, listeners, purchasers, etc.
  (often many millions)
- Feedback:
  5-star ratings, not-clicking ‘next’, purchases, etc.

Challenge:

- Users only rate a small number of items
  (the user/item rating data is sparse)

	Doctor Strange	Star Trek: Beyond	Zootopia
Alita	1	?	5
BB-8	3	4	?
C-3P0	3	5	2

$R_{2,1} = 3$

Slide credit: CMU MLD Matt Gormley

Different Approaches

Item-based (*Content filtering*)

- Features about each item
- Given an item, other “close” items have similar values
- e.g. Pandora.com, music genome project

Different Approaches

Item-based (*Content filtering*)

- Features about each item
- Given an item, other "close" items have similar values
- e.g. Pandora.com, music genome project

User-based

- Features about each user
- Given a user, other "close" users have similar preferences
- *Market segmentation*

Learning user-item relationship

- Can be done without features on either user or item
- Collaborative filtering techniques

Collaborative Filtering

Collaborative Filtering

Everyday Examples of Collaborative Filtering...

- Bestseller lists
- Top 40 music lists
- The “recent returns” shelf at the library
- Unmarked but well-used paths thru the woods
- The printer room at work
- “Read any good books lately?”
- …

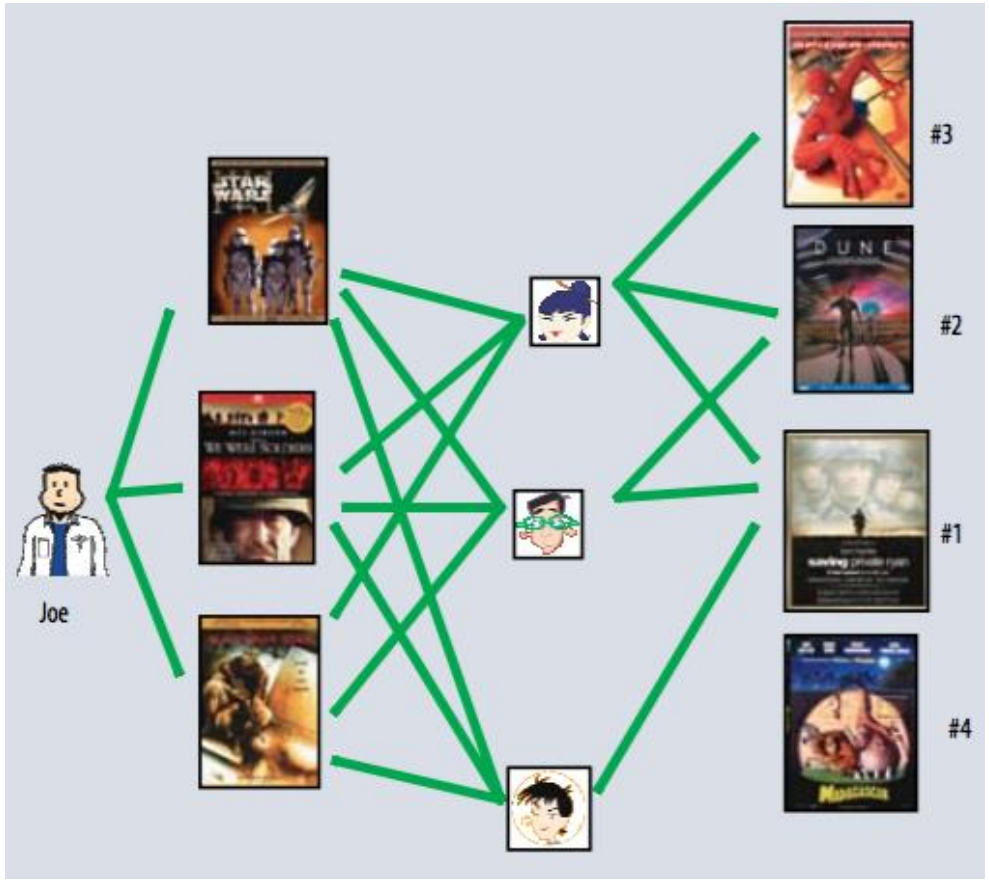
Common insight: personal tastes are correlated

- If Alita and BB-8 both like X and Alita likes Y then BB-8 is more likely to like Y
- especially (perhaps) if BB-8 knows Alita

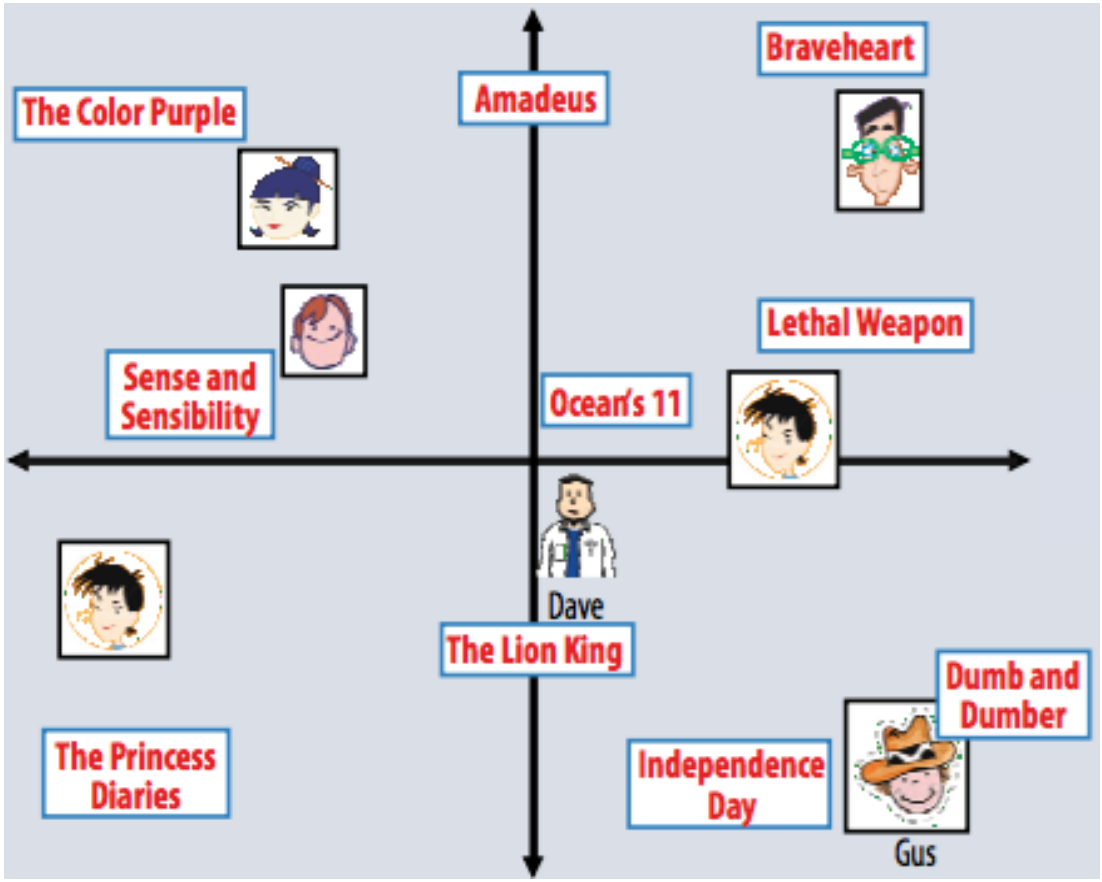
Slide from William Cohen

Two Types of Collaborative Filtering

1. Neighborhood Methods



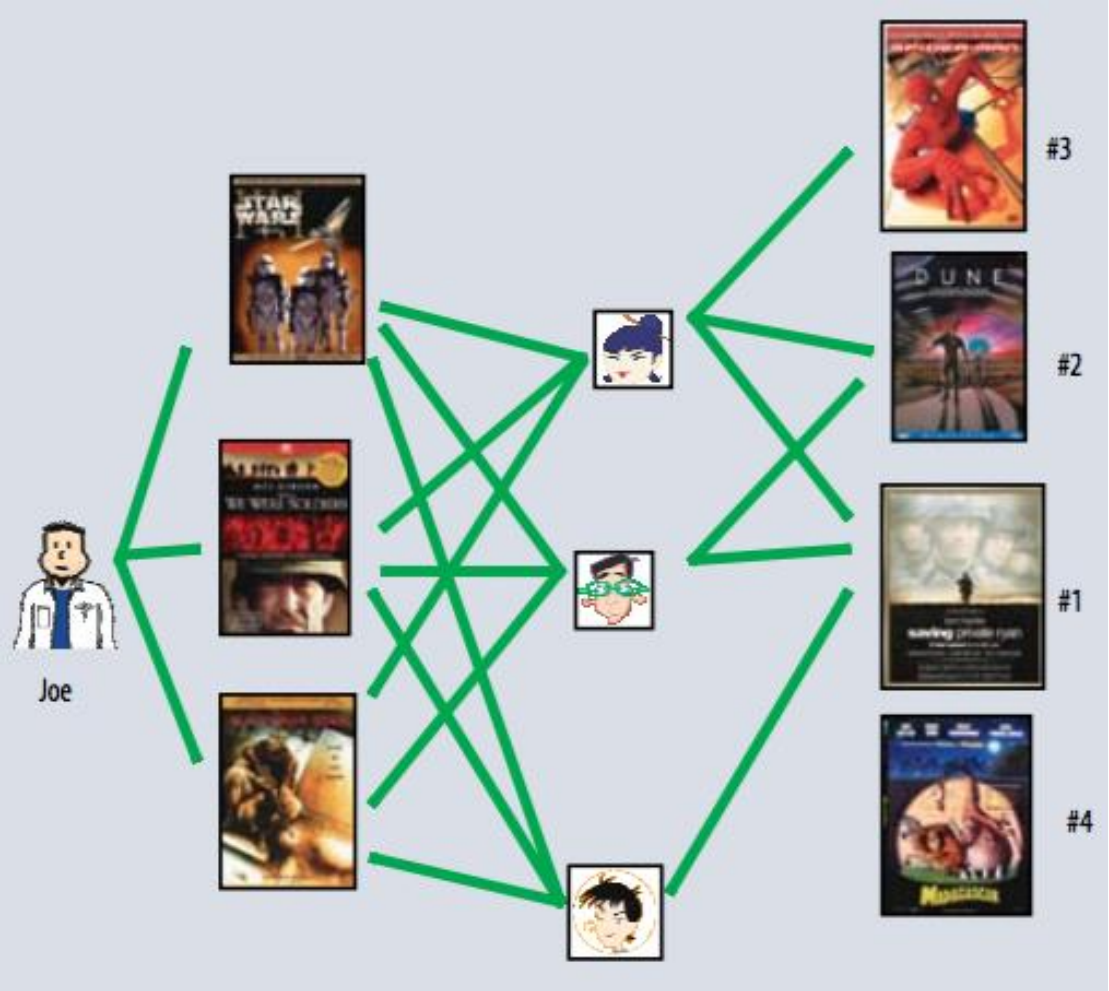
2. Latent Factor Methods



Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

1. Neighborhood Methods



In the figure, assume that a green line indicates the movie was **watched**

Algorithm:

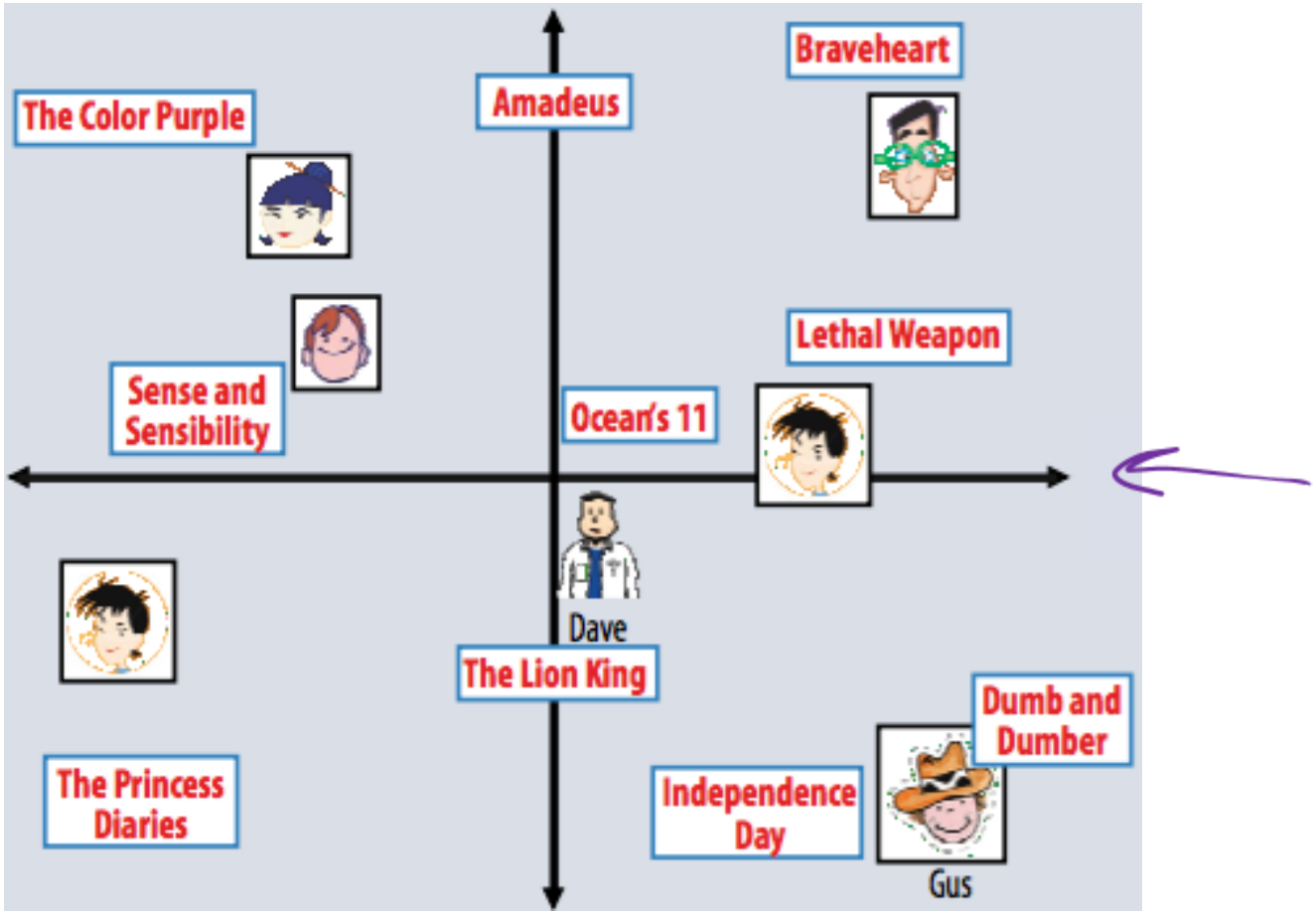
1. **Find neighbors** based on similarity of movie preferences
2. **Recommend** movies that those neighbors watched

Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

2. Latent Factor Methods

- Assume that both movies and users can be mapped to the same **feature space**
- **Recommend** a movie based on its **proximity** to the user in the feature (latent) space
- **Example algorithm**: Latent factor method

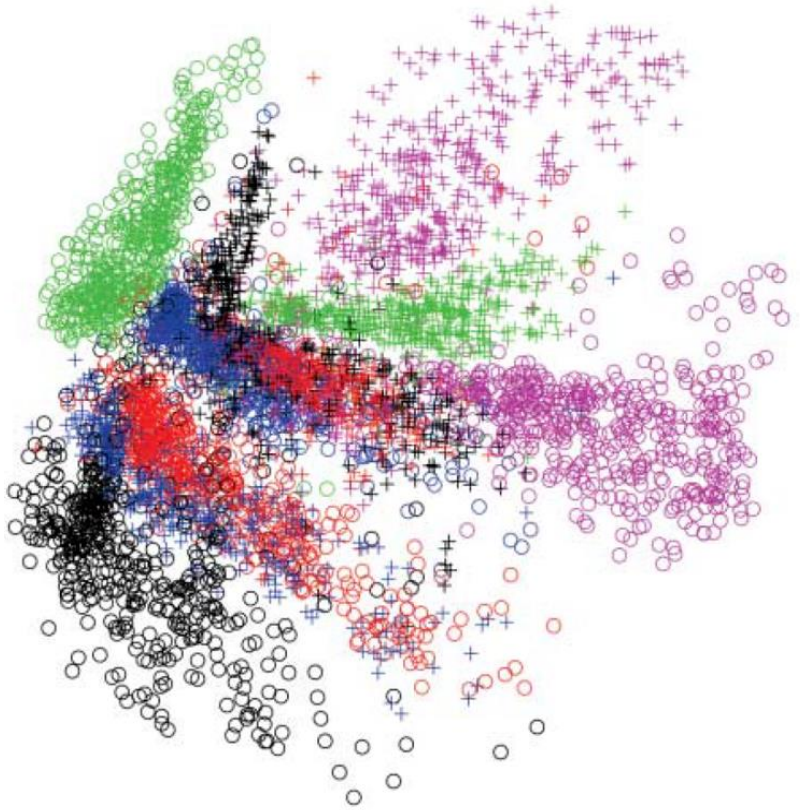


Figures from Koren et al. (2009)

Collaborative Filtering:
Latent Factor (Matrix Factorization) Method

Background: Learn a Feature Space

Autoencoders



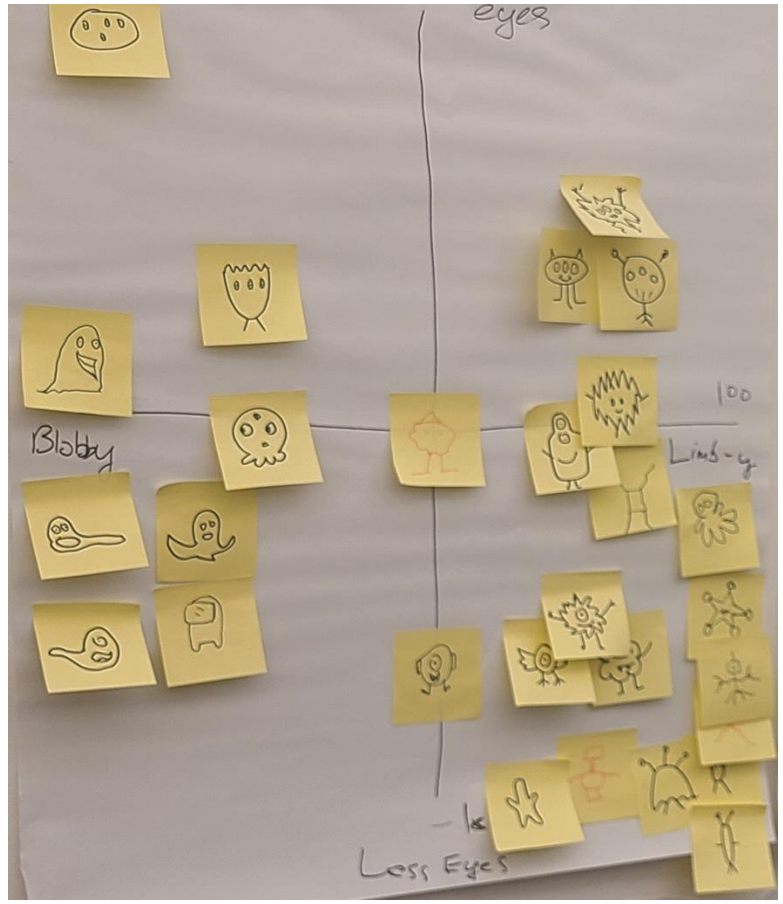
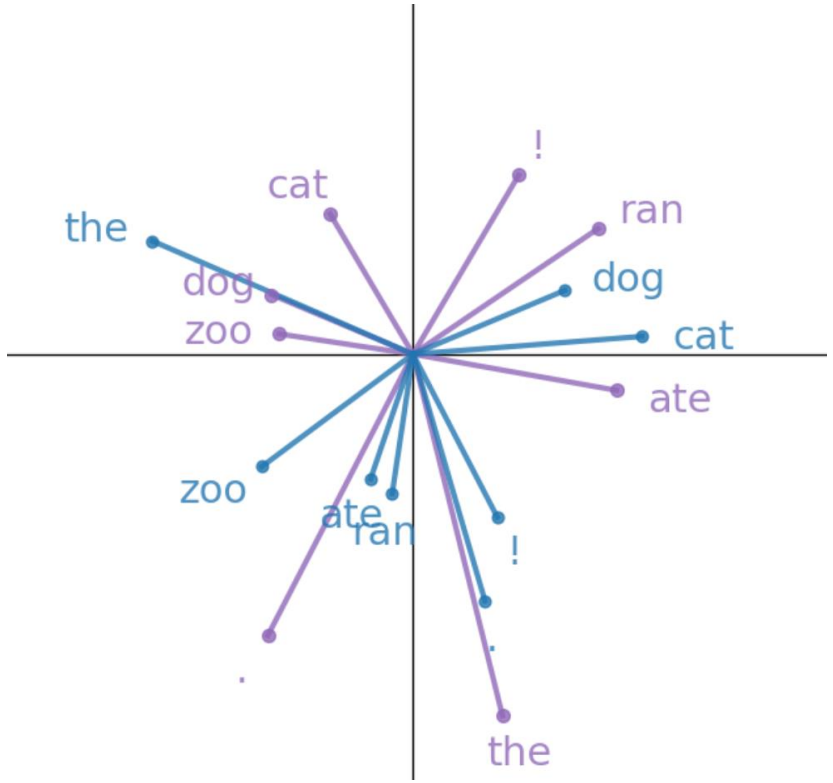


Image: Hinton & Salakhutdinov. *Science* 313.5786 (2006): 504-507.

Background: Learn a Feature Space

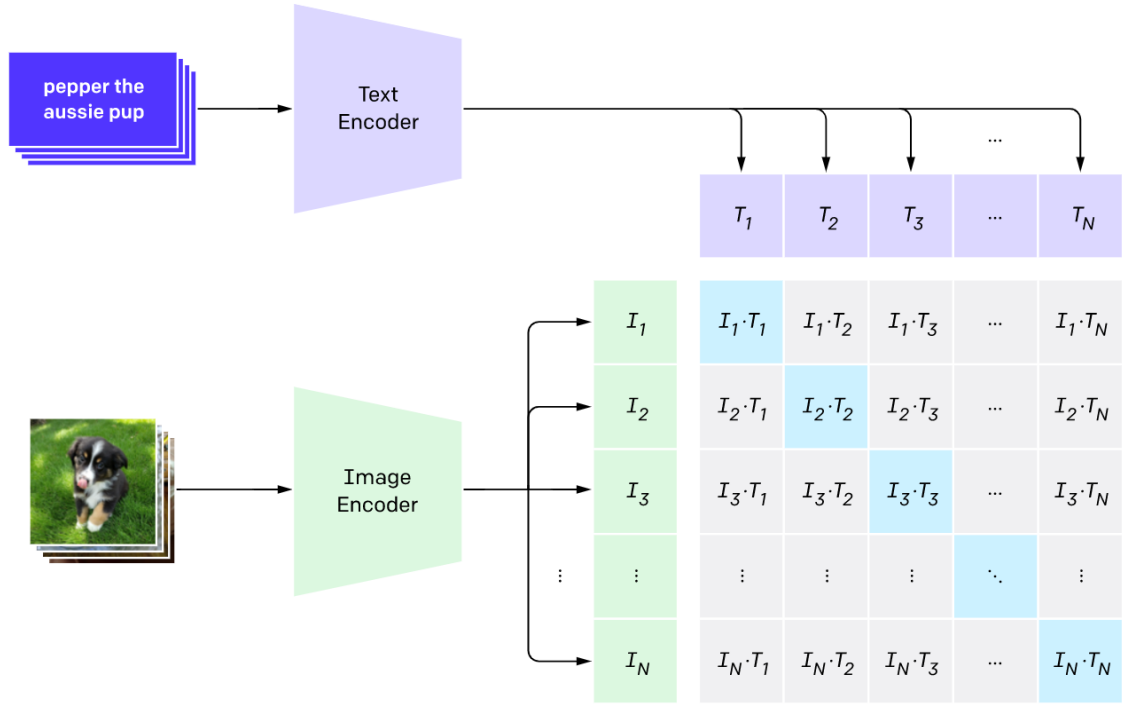
Word2vec:

Feature space for words



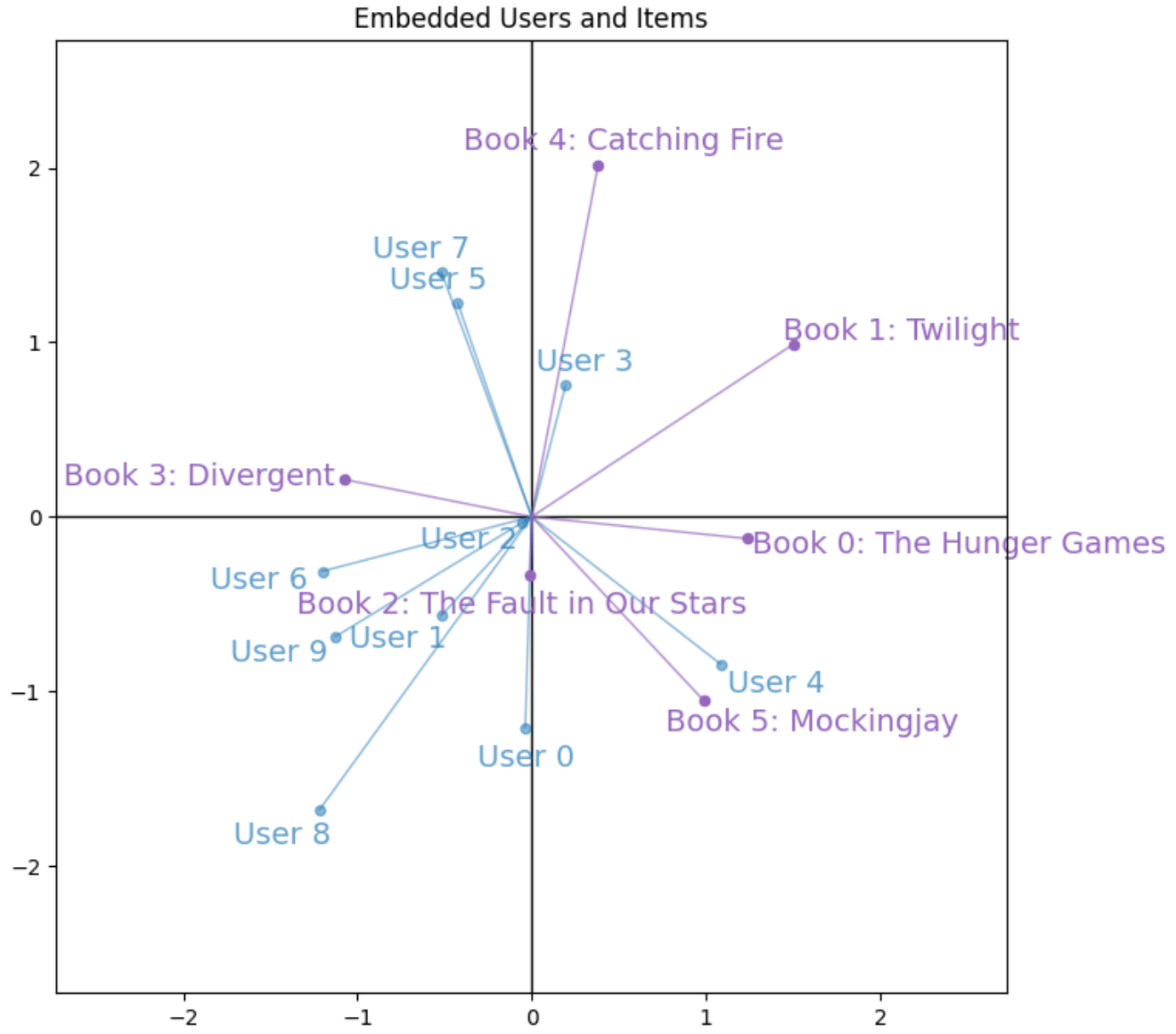
CLIP:

Common feature space for text and images



Recommender System: Latent Factor Model

Learning to map items and users to the same lower dimensional space



Recommender System: Latent Factor Model

Optimization: Objective function using only the labels we have

$$J(U,V) = \sum_{i,j,r \in \mathcal{D}} \left(r - \mathbf{u}^{(i)^T} \mathbf{v}^{(j)} \right)^2$$

Notation alert!

- Let $\mathbf{u}^{(i)}$ be the transpose of the i-the row of U
- Let $\mathbf{v}^{(j)}$ be the transpose of the j-the row of V

Latent Factor (Matrix Factorization) Method

SGD Optimization

Recommender System: Matrix Factorization

Optimization: Objective function using only the labels we have

$$J(U,V) = \sum_{i,j,r \in \mathcal{D}} \left(r - \mathbf{u}^{(i)^T}\mathbf{v}^{(j)}\right)^2$$

SGD objective function: Just one tuple in $\mathcal{D}$: (r_{ij}, i, j)

$$J\left(\mathbf{u}^{(i)}, \mathbf{v}^{(j)}\right) = \frac{1}{2}\left(r_{ij} - \mathbf{u}^{(i)^T}\mathbf{v}^{(j)}\right)^2$$

SGD gradient steps

$$\frac{\partial J}{\partial \boldsymbol{u}^{(j)}} = -\left(r_{ij} - \mathbf{u}^{(i)^T}\mathbf{v}^{(j)}\right)\mathbf{v}^{(j)}$$

$$\frac{\partial J}{\partial \mathbf{v}^{(j)}} = -\left(r_{ij} - \mathbf{u}^{(i)^T}\mathbf{v}^{(j)}\right)\mathbf{u}^{(j)}$$

Notation alert!

- Let $\mathbf{u}^{(i)}$ be the transpose of the i-the row of U
- Let $\mathbf{v}^{(j)}$ be the transpose of the j-the row of V

Latent Factor Method

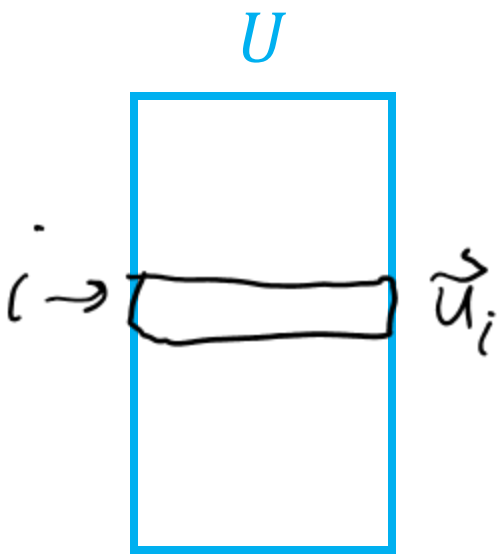
Inference

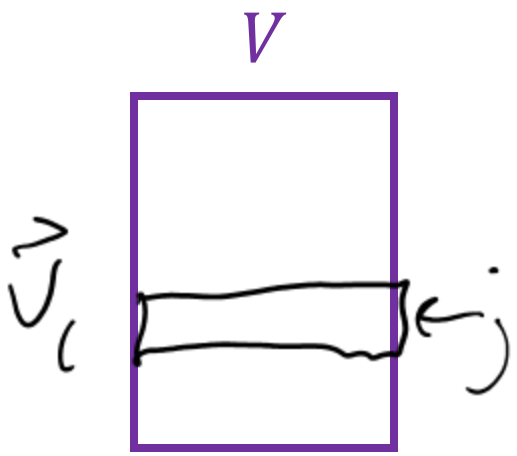
Inference for Recommender Systems

First solve:

$$\min_{U,V} J(U,V) \qquad J(U,V) = \sum_{i,j,r \in \mathcal{D}} \left(r - \mathbf{u}^{(i)^T} \mathbf{v}^{(j)}\right)^2$$

What then?

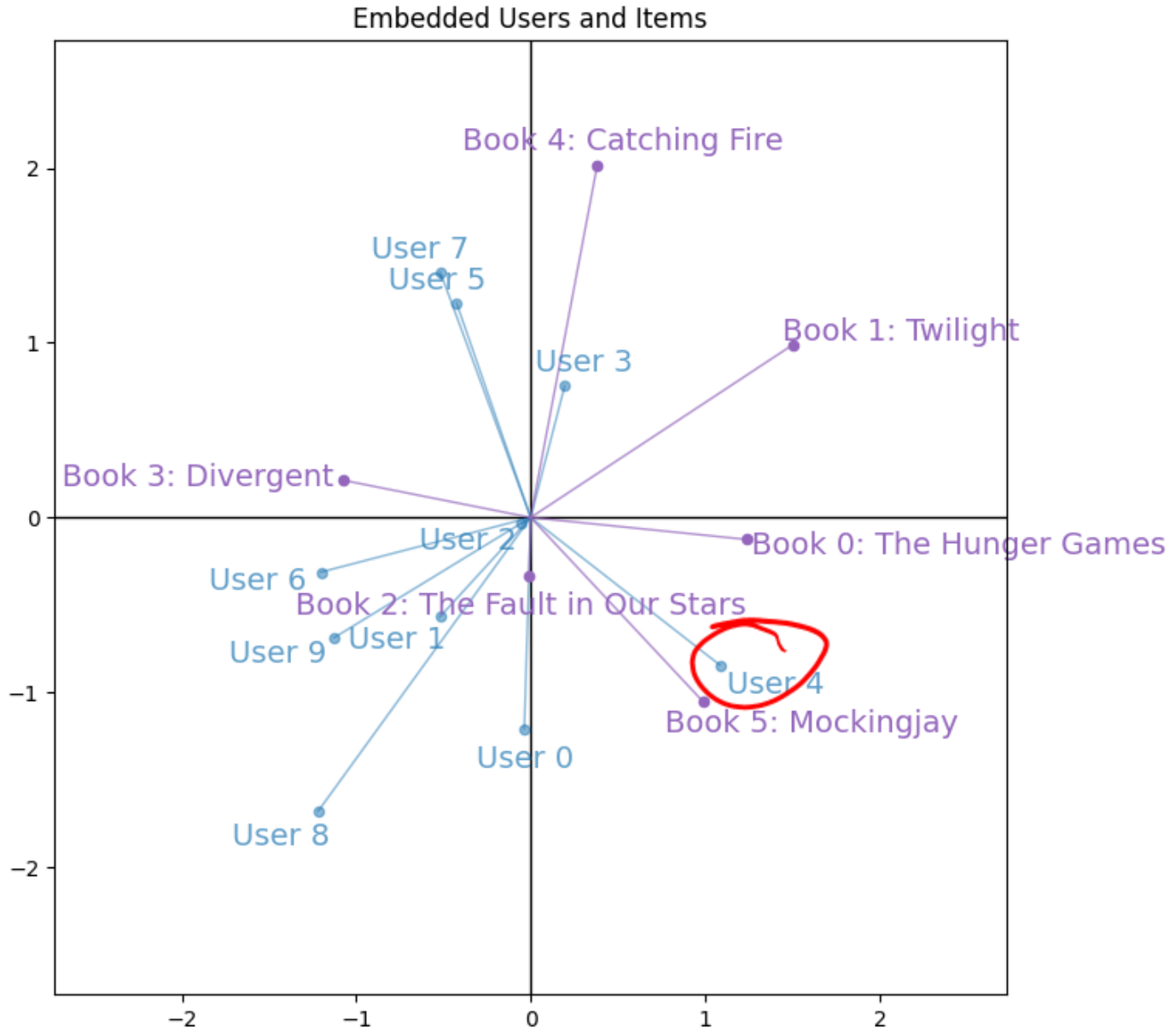




	Doctor Strange	Star Trek: Beyond	Zootopia
Alita	1		5
BB-8	3	4	$\hat{r}$
C-3Po	3	5	2

Inference for Recommender Systems

Recommendations



Latent Factor Method

Practical Considerations

Latent Factor Regularization

Add regularization to avoid overfitting

$$\min_{U,V} J(U,V)$$

$$J(U,V) = \sum_{i,j \in \mathcal{S}} \left(r_{ij} - \mathbf{u}^{(i)^T} \mathbf{v}^{(j)} \right)^2 + \lambda \|u_i\|_2^2 + \lambda \|v_j\|_2^2$$

Bias in Recommender Systems

What high-level problems can occur from recommender systems?

Summary

Recommender systems solve many **real-world** (*large-scale) **problems**

Collaborative filtering using a latent factor model can be done without any features about the users or items other than their past ratings

Solving for latent factors is just another example of a **common recipe**:

1. define a model
2. define an objective function
3. optimize

Optimization

- Use SGD
- Add regularization to avoid overfitting