

Announcements

Assignments

- HW8 (written + programming)
 - Due Thu 4/9, 11:59 pm

Introduction to Machine Learning Recommender Systems

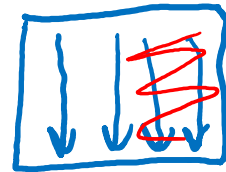
Instructor: Pat Virtue

Background: Low Dimensional Embeddings

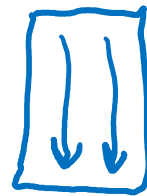
PCA: What did we do?

$$X \in \mathbb{R}^{N \times M}$$

$$\text{eig}(X^T X) \rightarrow V \in \mathbb{R}^{M \times M}$$



$$\rightarrow V_k \in \mathbb{R}^{M \times K}$$



$$\vec{x} \in \mathbb{R}^M$$

$$\vec{z} = V_k^T \vec{x}$$

$$\vec{z} \in \mathbb{R}^K$$

$$K < M$$

$$\vec{x}' = V_k \vec{z}$$


$$\vec{x}' = V_k V_k^T \vec{x}$$


$$\min_V \|\vec{x} - \vec{x}'\|$$

Background: Low Dimensional Embeddings

Why might low dimensional embeddings be useful?

- Example: MNIST digit classification with nearest neighbor

$$\vec{x} \in \mathbb{R}^{784}$$


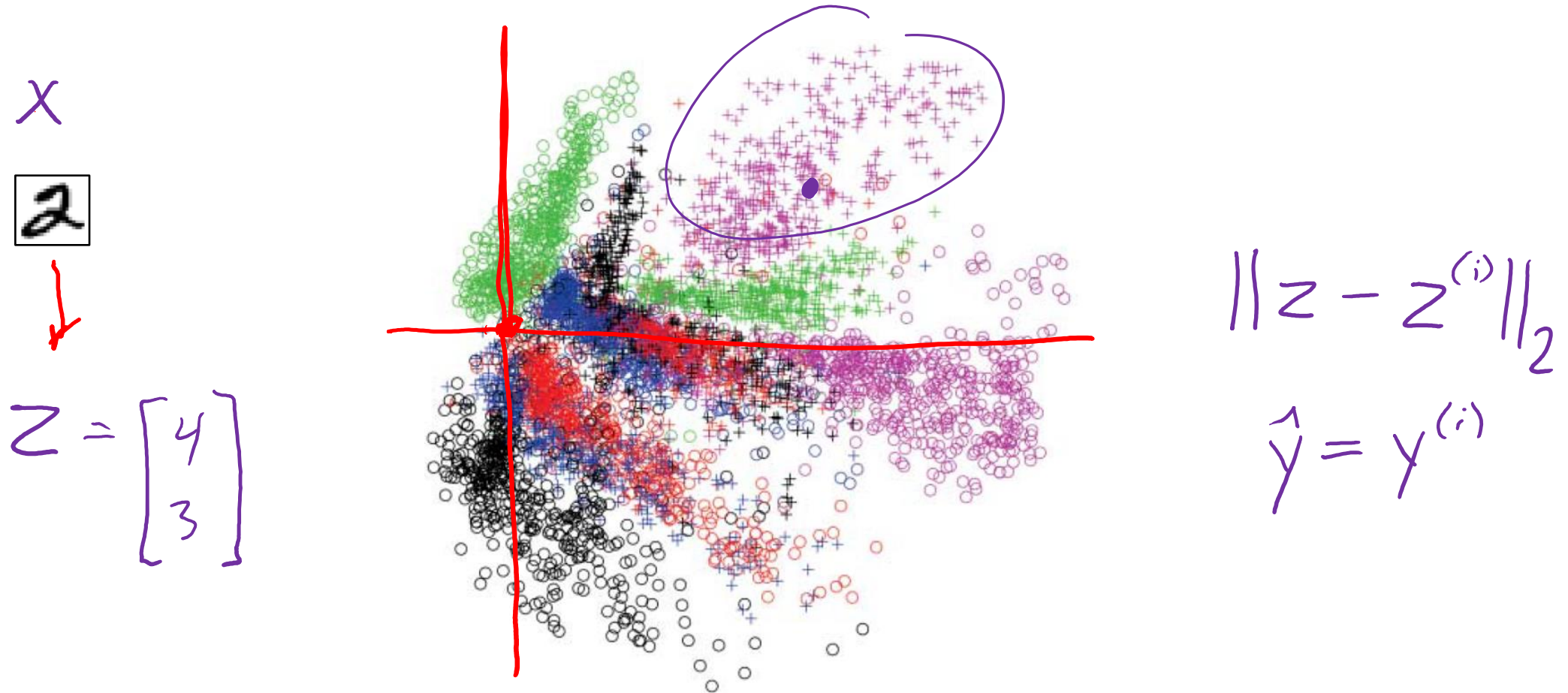
$$X_{\text{train}} \in \mathbb{R}^{N \times 784}$$


$$\|\vec{x} - \vec{x}^{(i)}\|_2$$
$$\hat{y} = y^{(i)}$$

Background: Low Dimensional Embeddings

Why might low dimensional embeddings be useful?

- Example: MNIST digit classification with nearest neighbor



Background: Measure of Similarity

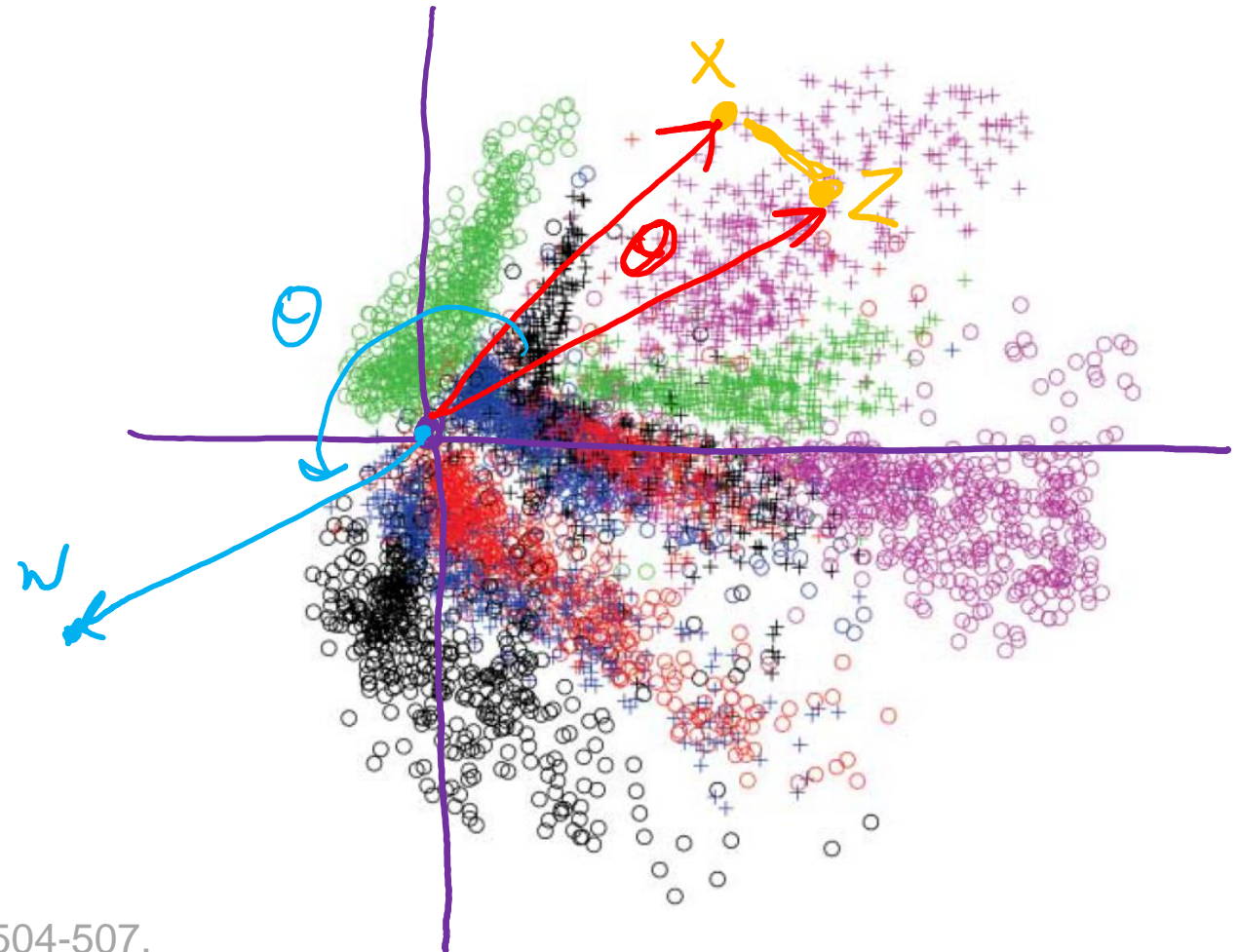
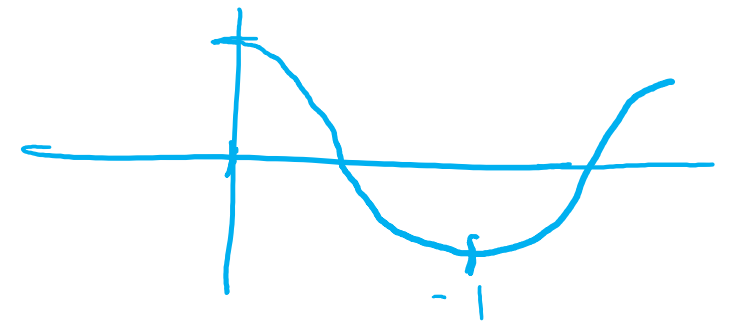
We've been using Euclidean distance

- $d(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\|_2$

Cosine similarity

- Two vectors are similar if the angle between them is small

- $d(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T \mathbf{z} = \mathbf{x} \cdot \mathbf{z}$
 $= \|\mathbf{x}\| \|\mathbf{z}\| \cos \theta$



Recommender Systems

A Common Challenge:

- Assume you're a company selling **items** of some sort: movies, songs, products, etc.
- Company collects millions of **ratings** from **users** of their **items**
- To maximize profit / user happiness, you want to **recommend** items that users are likely to want

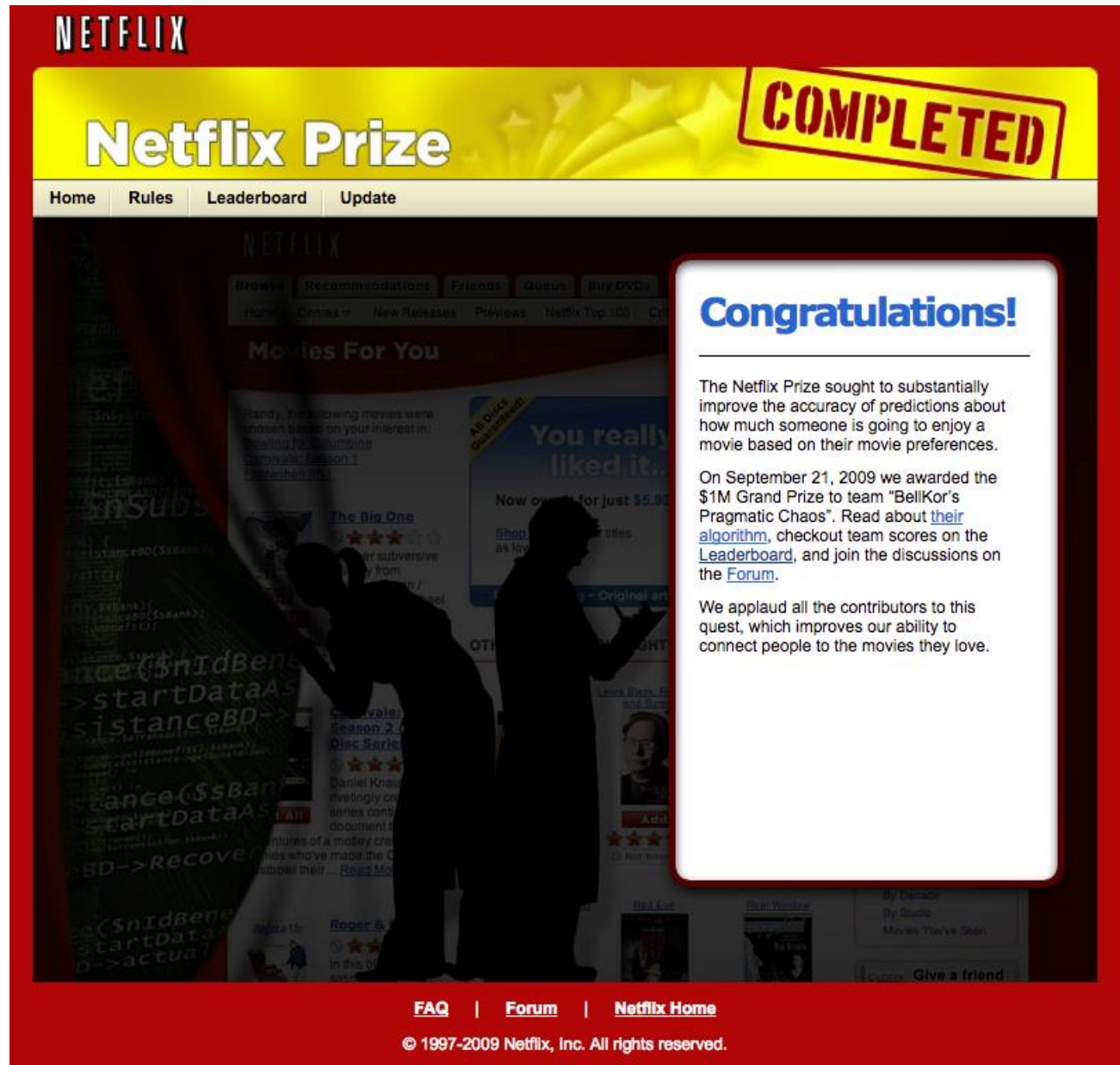
Recommender Systems

The screenshot shows the Amazon homepage with a personalized recommendation section for a user named Matt. The header includes the Amazon Prime logo, a search bar, and navigation links. The recommendation section is titled "Recommended for you, Matt" and displays four categories of products:

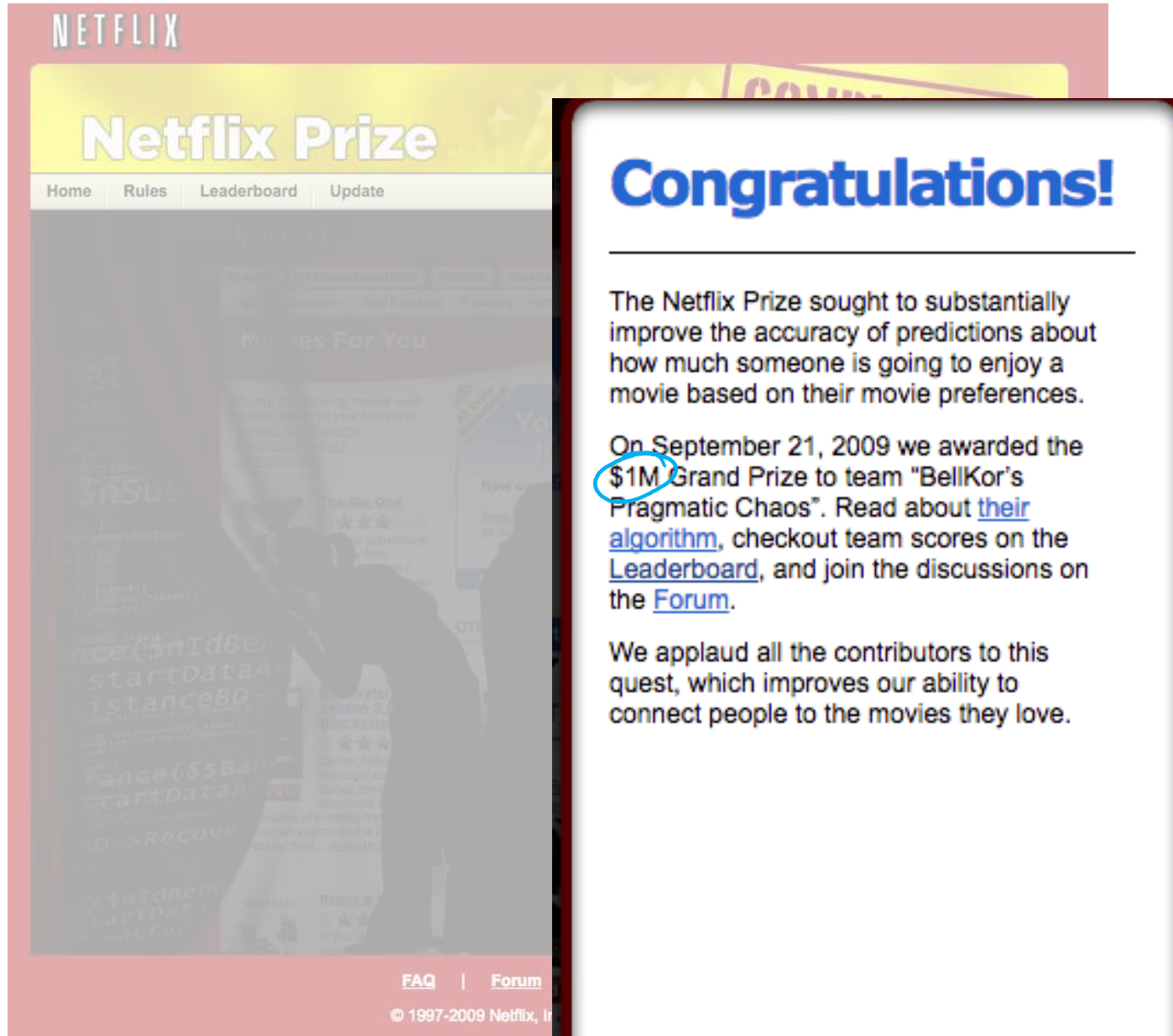
- Buy It Again in Grocery** (14 ITEMS): Includes images of Jif peanut butter, a bottle of maple syrup, and a box of Post-it notes.
- Buy It Again in Pets** (6 ITEMS): Includes images of pet products like Advantage II flea treatment, pet shampoos, and cat food.
- Buy It Again in Baby Products** (5 ITEMS): Includes images of baby products like Crayola crayons, baby bibs, and baby toys.
- Engineering Books** (86 ITEMS): Includes the book cover for "Probabilistic Graphical Models: Principles and Techniques" by Daphne Koller and Nir Friedman.

Recommender Systems

$$u^{(i)} \quad v^{(j)} \\ \downarrow \\ R_{ij}$$



Recommender Systems



The image shows a screenshot of the Netflix Prize website. The main page has a yellow header with the 'Netflix Prize' title and navigation links for 'Home', 'Rules', 'Leaderboard', and 'Update'. Below this, there's a section titled 'Movies For You' with a list of movie recommendations. An inset box on the right side of the image shows a 'Congratulations!' message. The message states that the Netflix Prize sought to improve prediction accuracy and that the \$1M Grand Prize was awarded to the team 'BellKor's Pragmatic Chaos' on September 21, 2009. It also provides links to read about their algorithm, view team scores on the leaderboard, and join discussions on the forum. The footer of the website includes links for 'FAQ' and 'Forum' and a copyright notice for 1997-2009 Netflix, Inc.

NETFLIX

Netflix Prize

Home Rules Leaderboard Update

Congratulations!

The Netflix Prize sought to substantially improve the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences.

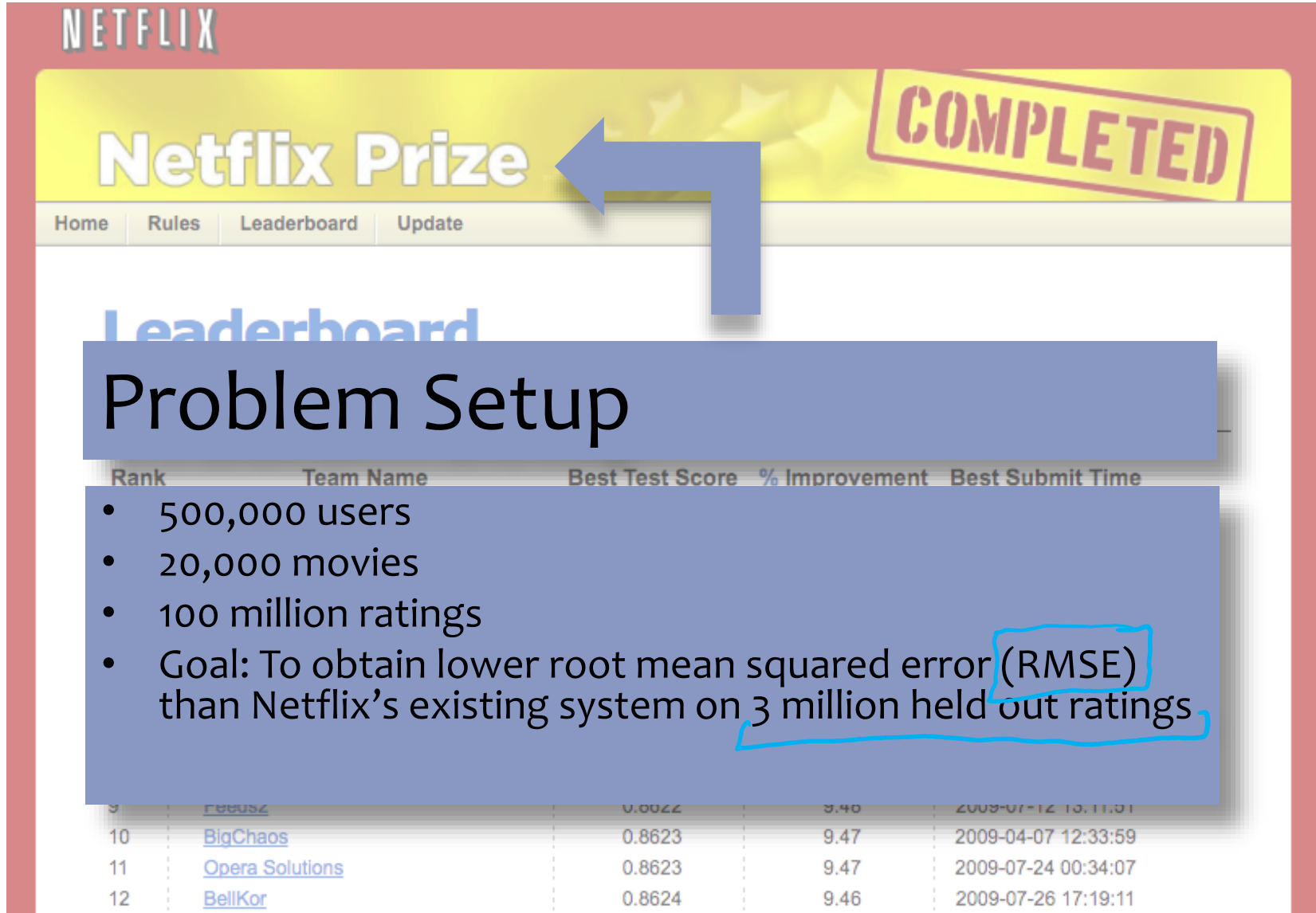
On September 21, 2009 we awarded the **\$1M** Grand Prize to team "BellKor's Pragmatic Chaos". Read about [their algorithm](#), checkout team scores on the [Leaderboard](#), and join the discussions on the [Forum](#).

We applaud all the contributors to this quest, which improves our ability to connect people to the movies they love.

FAQ | Forum

© 1997-2009 Netflix, Inc.

Recommender Systems



The image is a screenshot of the Netflix Prize website. At the top, the Netflix logo is on the left, and a yellow banner with the text "Netflix Prize" and a "COMPLETED" stamp is on the right. A blue arrow points from the "COMPLETED" stamp to the "Netflix Prize" text. Below the banner is a navigation bar with links: Home, Rules, Leaderboard, and Update. The "Leaderboard" link is highlighted. Below the navigation bar, the word "Leaderboard" is written in large blue letters. A large blue box with the text "Problem Setup" is overlaid on the page. Below this box, a table shows the leaderboard data. The table has five columns: Rank, Team Name, Best Test Score, % Improvement, and Best Submit Time. The table lists teams from rank 9 to 12. A blue box highlights the text "Goal: To obtain lower root mean squared error (RMSE) than Netflix's existing system on 3 million held out ratings" in the table. The "RMSE" is circled in blue.

NETFLIX

Netflix Prize

COMPLETED

Home Rules Leaderboard Update

Leaderboard

Problem Setup

- 500,000 users
- 20,000 movies
- 100 million ratings
- Goal: To obtain lower root mean squared error (RMSE) than Netflix's existing system on 3 million held out ratings

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
9	Feus2	0.8622	9.48	2009-07-12 15:11:51
10	BigChaos	0.8623	9.47	2009-04-07 12:33:59
11	Opera Solutions	0.8623	9.47	2009-07-24 00:34:07
12	BellKor	0.8624	9.46	2009-07-26 17:19:11

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.
- **Key Assumptions:**
 - Can represent ratings numerically
as a user/item matrix
 - Users only rate a small number of
items (the matrix is sparse)



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

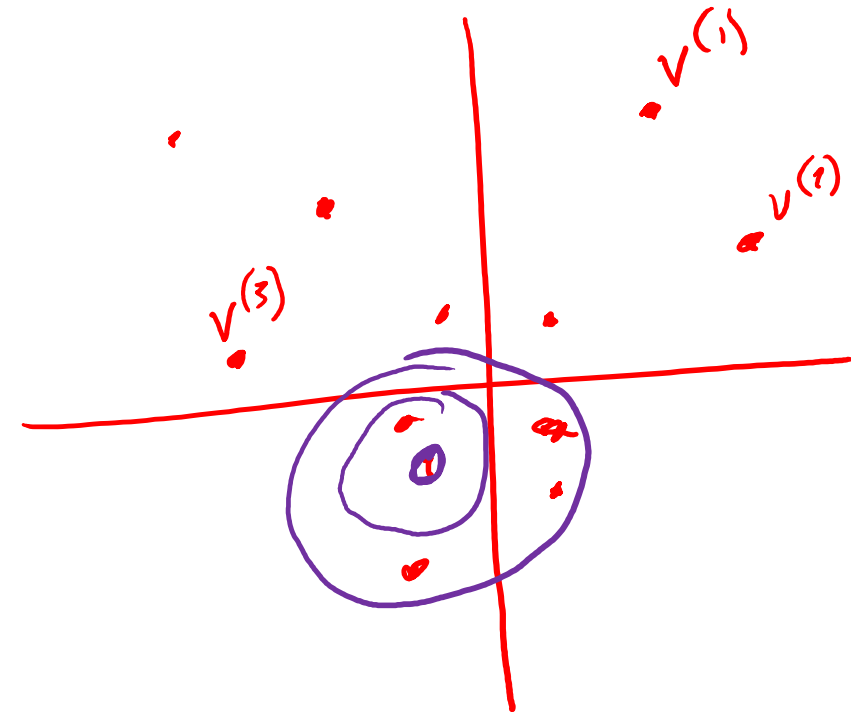


Different Approaches

Item-based (*Content filtering*)

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

$$v^{(i)} \in \mathbb{R}^M \Rightarrow \mathbb{R}^k$$



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

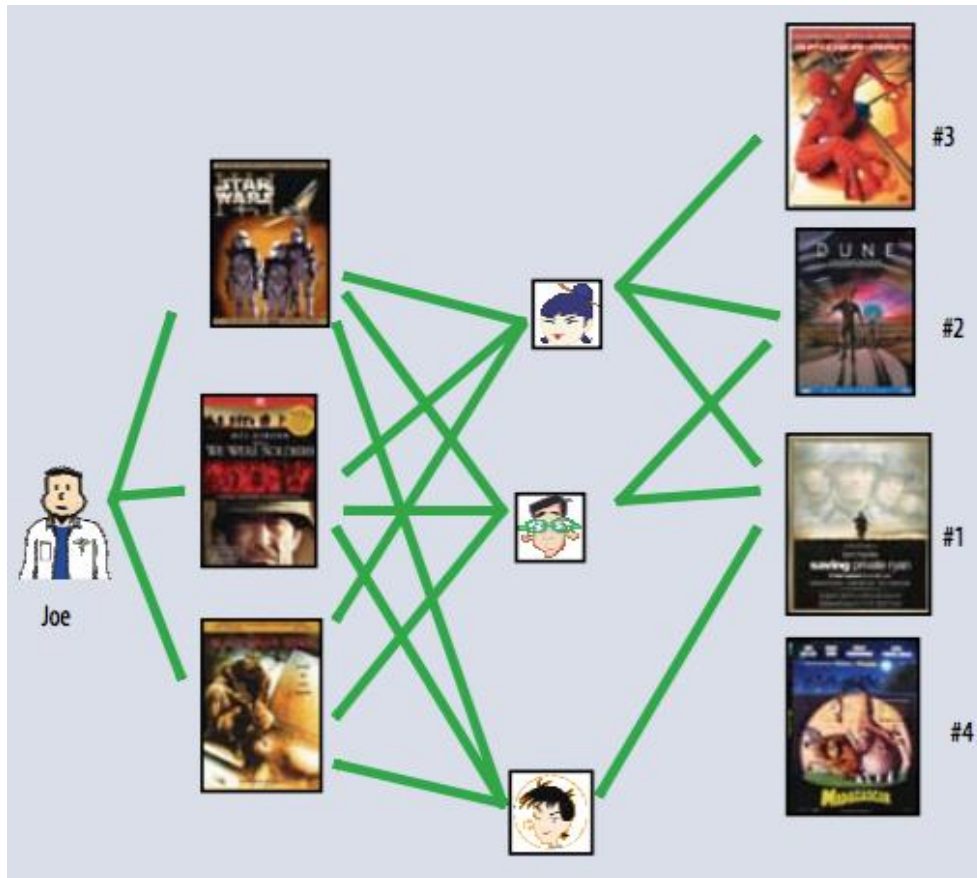
COLLABORATIVE FILTERING

Collaborative Filtering

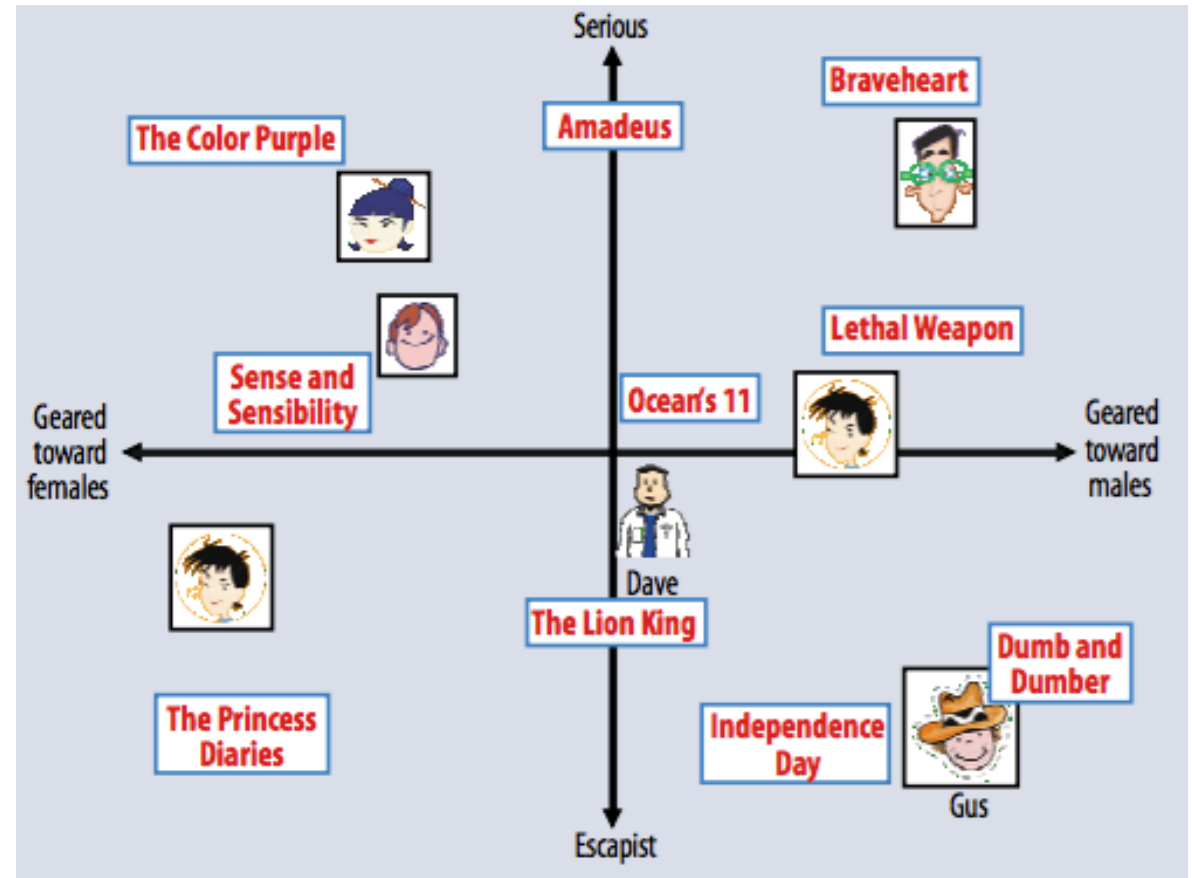
- **Everyday Examples of Collaborative Filtering...**
 - Bestseller lists
 - Top 40 music lists
 - The “recent returns” shelf at the library
 - Unmarked but well-used paths thru the woods
 - The printer room at work
 -  – “Read any good books lately?”
 - ...
- **Common insight:** personal tastes are 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

Two Types of Collaborative Filtering

1. Neighborhood Methods

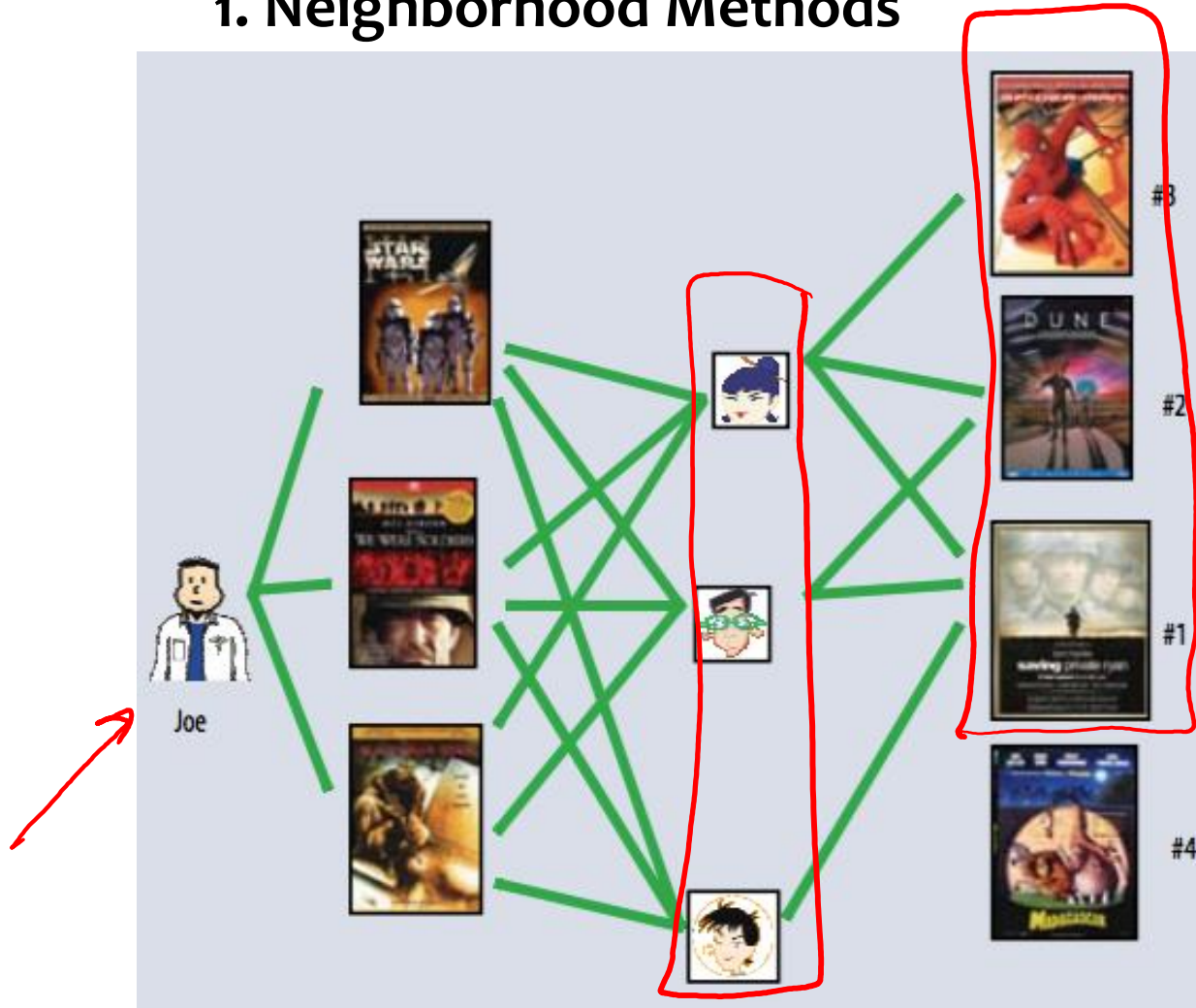


2. Latent Factor Methods



Two Types of Collaborative Filtering

1. Neighborhood Methods



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

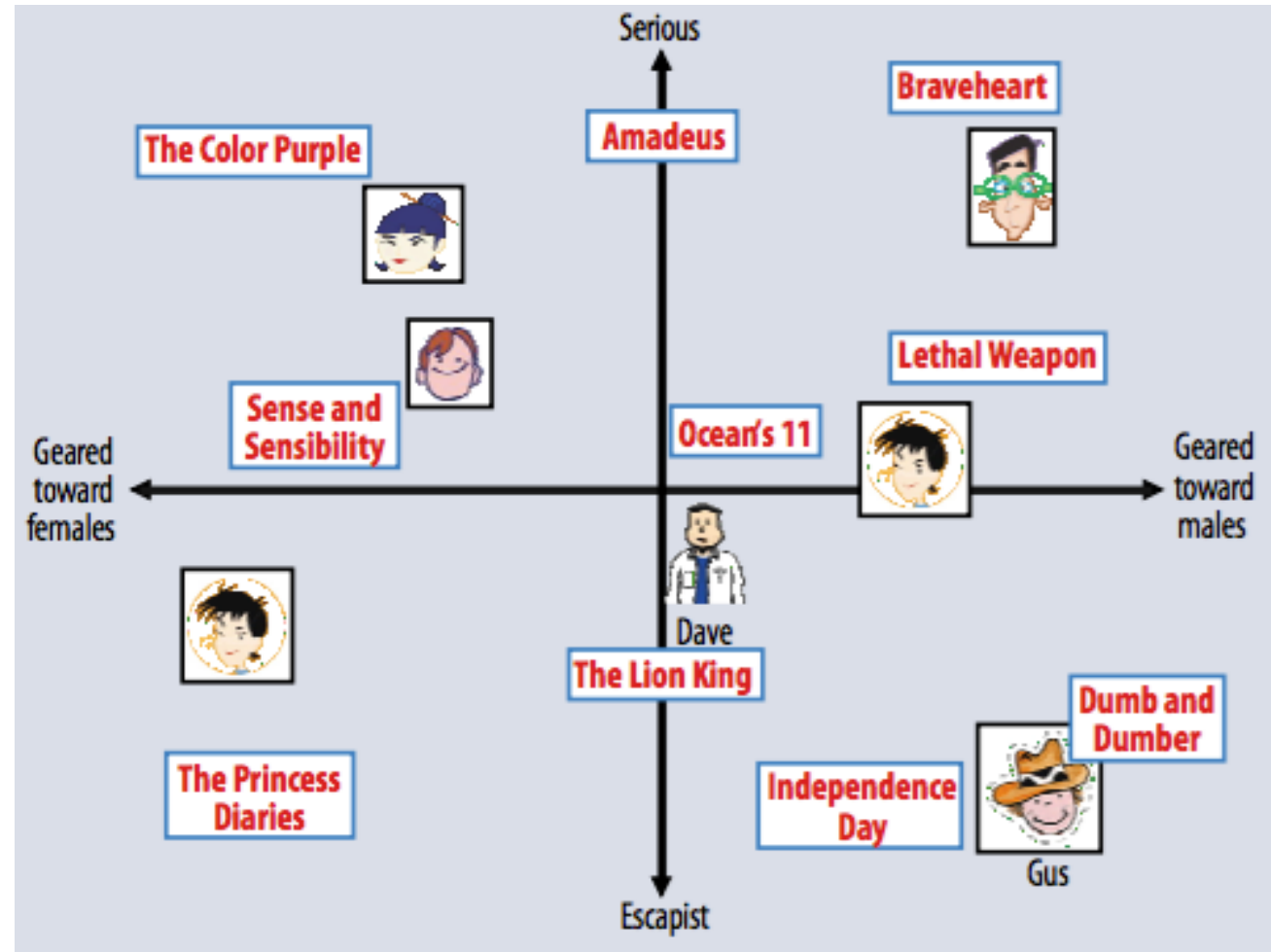
Algorithm:

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

Two Types of Collaborative Filtering

2. Latent Factor Methods

- Assume that both movies and users live in some **low-dimensional space** describing their properties
- **Recommend** a movie based on its **proximity** to the user in the latent space
- **Example Algorithm:** Matrix Factorization



Recommender System: Matrix Factorization

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

$$u^{(i)} \in \mathbb{R}^K$$

$$v^{(j)} \in \mathbb{R}^K$$

Similarity

■ Euclidean

$$\|\vec{u}^{(i)} - \vec{v}^{(j)}\|_2$$

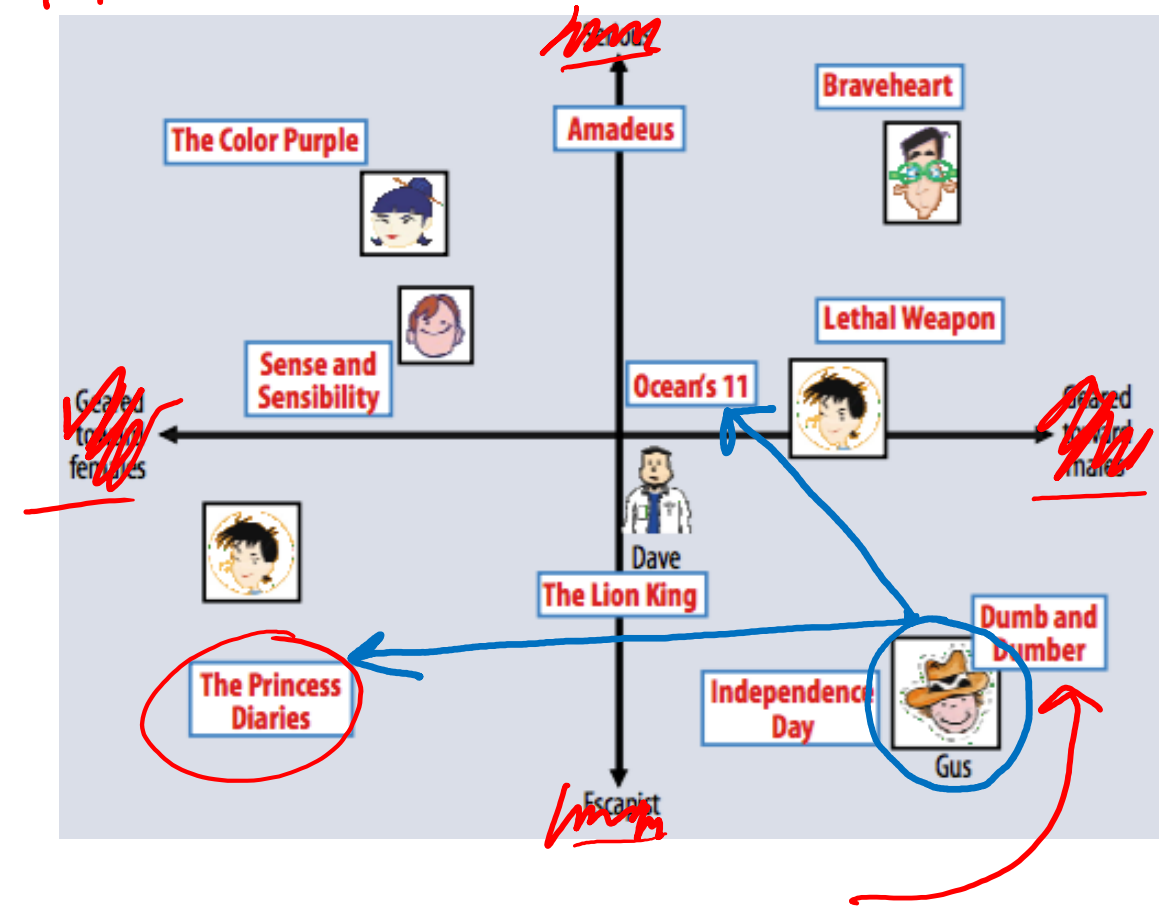
■ Cosine

$$\vec{u}^{(i)T} \vec{v}^{(j)}$$

Label

R_{ij} rating user i on item j

$K=2$



Recommender System: Matrix Factorization

Optimization

$$\hat{r}_{ij} = \vec{u}^{(i)T} \vec{v}^{(j)}$$

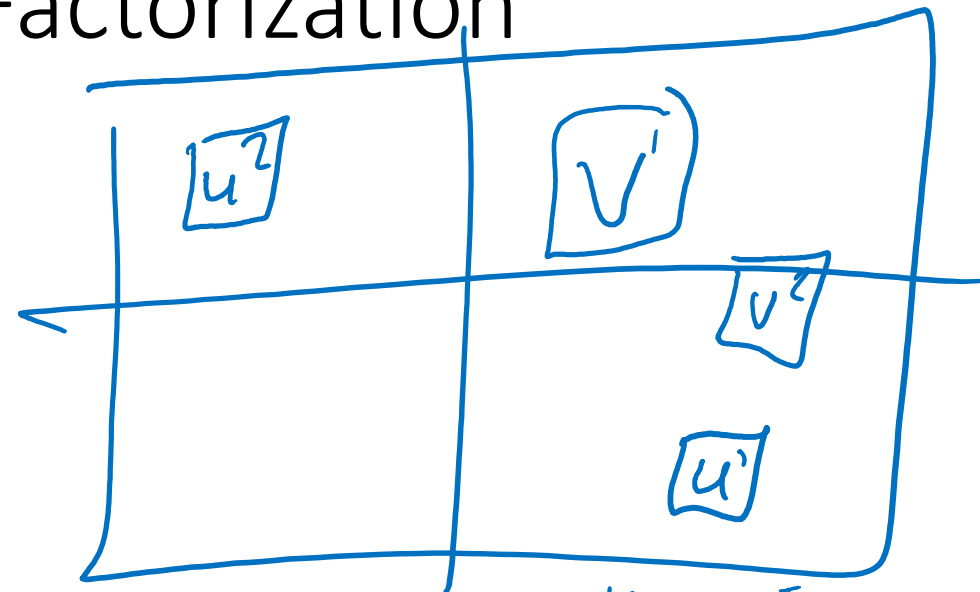
$$\min_{U, V} \sum_{ij} (R_{ij} - \vec{u}^{(i)T} \vec{v}^{(j)})^2$$

$$\min_{U, V} \|R - UV^T\|_F^2$$

$$X^T X \approx V_k \Lambda V_k^T$$

$$X = U S V^T$$

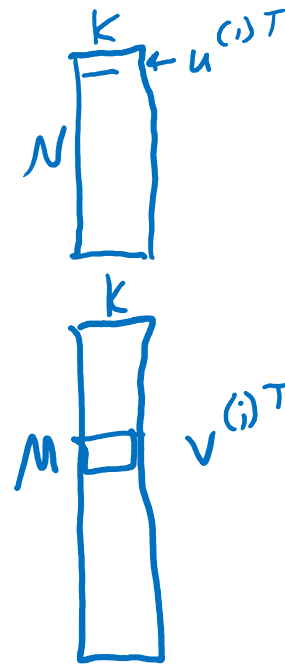
$$\tilde{X} = U_k \underline{S}_k V_k^T$$



$$U \in \mathbb{R}^{N \times K}$$

$$V \in \mathbb{R}^{M \times K}$$

$$R \in \mathbb{R}^{N \times M}$$



Recommender System: Matrix Factorization

Sparse labels 😞

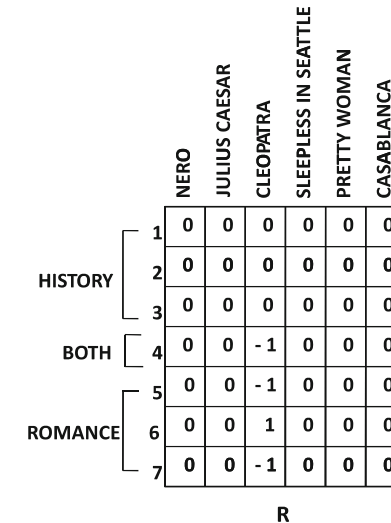
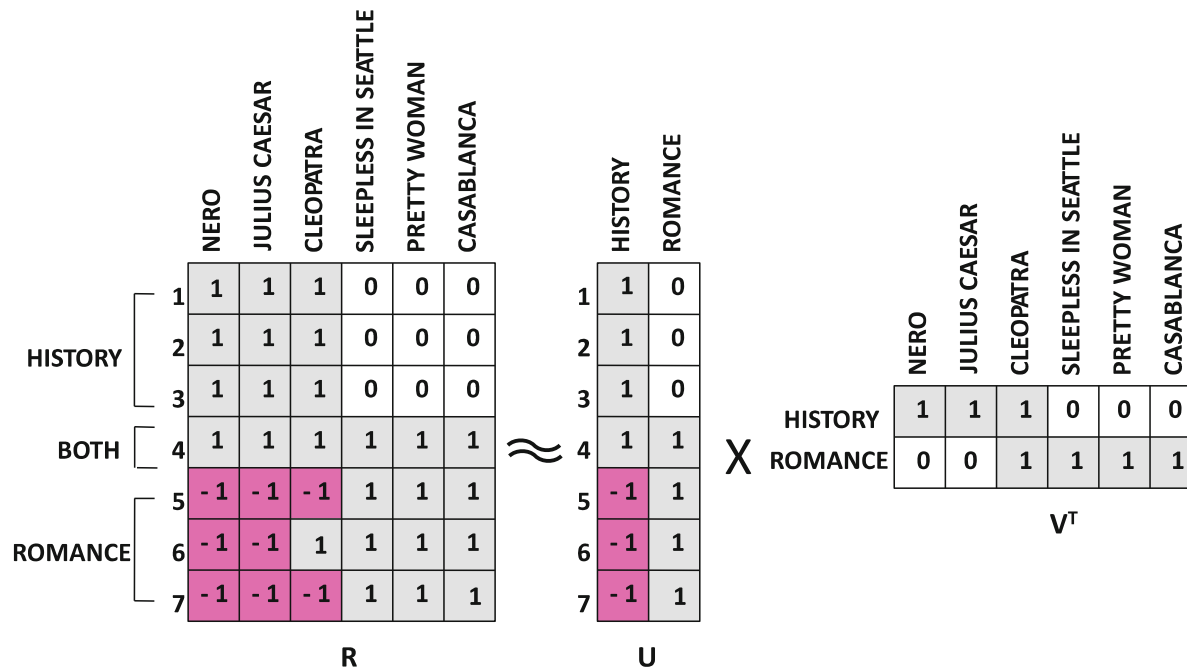
R

M

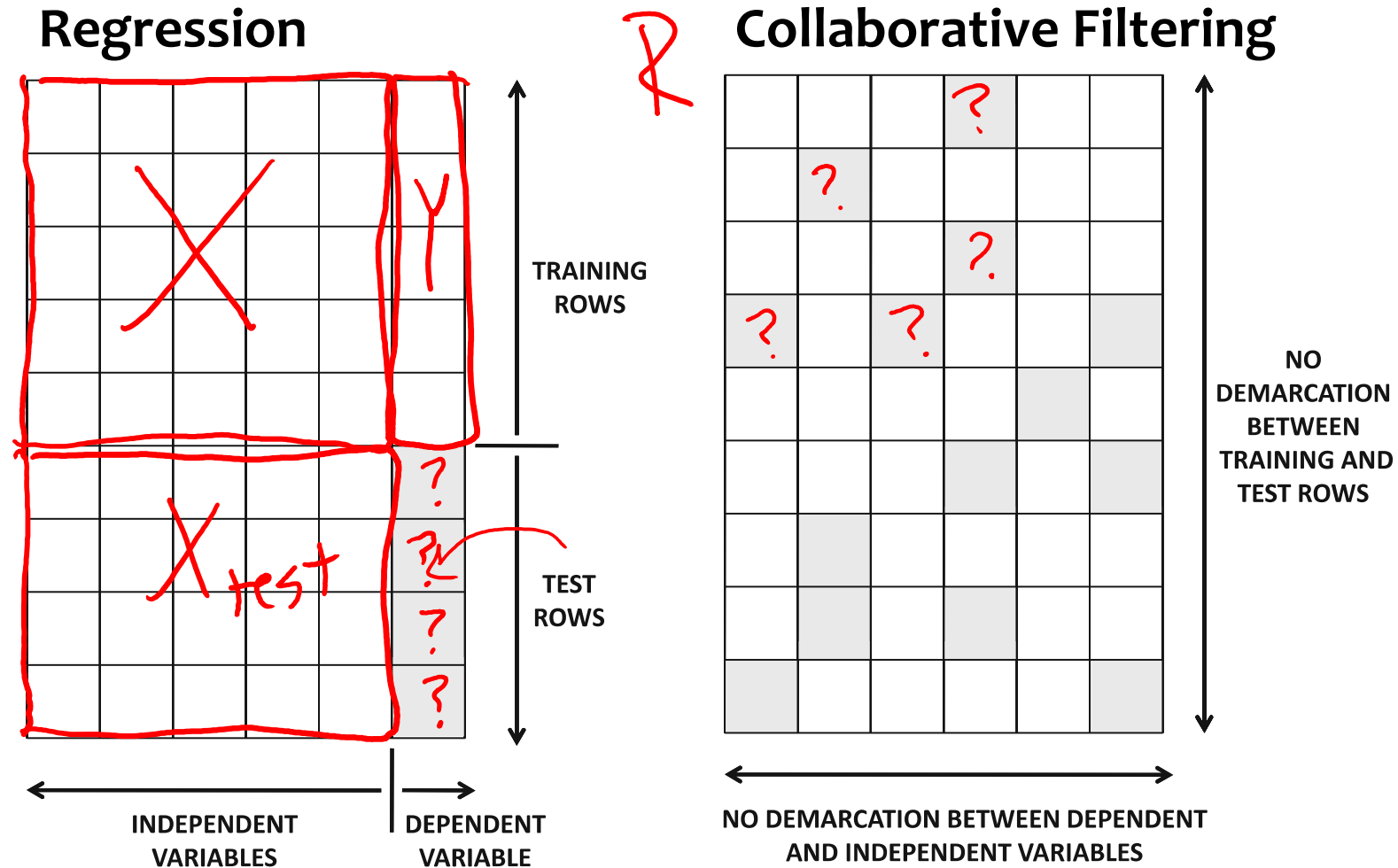
N

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

Example: MF for Netflix Problem



Regression vs. Collaborative Filtering



Matrix Factorization: SVD

We can use SVD, but as you'll see it has issues

$$\text{SVD}(R) \rightarrow U, S, V$$

$$\rightarrow \tilde{U}_k = U_k S_k, \tilde{V}_k = V_k$$

$$\tilde{R} = \tilde{U}_k \tilde{V}_k^T$$

$$\min_{U, V} \|R - \tilde{R}\|$$

U, V orth

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

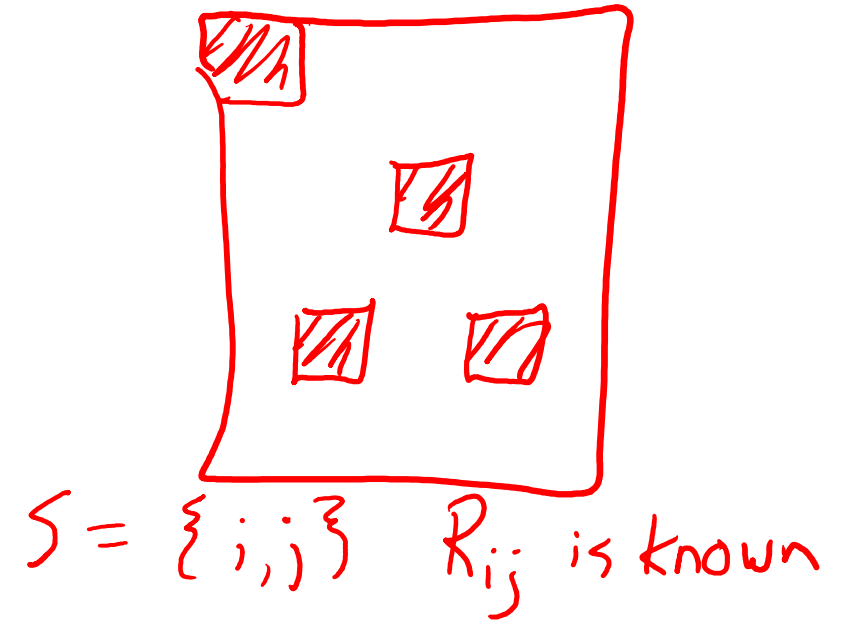
Alita	-1	0	1
BB-8	0	0,5	0
C-3Po	0	1	-0,5

Matrix Factorization: SGD

Objective function using only the labels we have

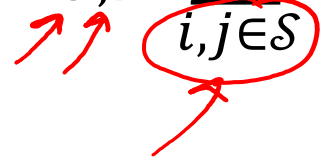
$$\min_{U, V} \|R - UV^T\|_F^2$$

$$\min_{U, V} \sum_{i, j \in S} (R_{ij} - u^{(i)T} v^{(j)})^2$$



Piazza Poll 1

Is the following optimization a quadratic optimization?

$$\min_{\underline{u}, \underline{v}} \sum_{i,j \in \mathcal{S}} \left(\underline{R}_{ij} - \underline{u}^{(i)T} \underline{v}^{(j)} \right)^2$$


- A. Yes
- B. Calamity
- ☒ C. No

Matrix Factorization: SGD

Method of *alternating minimization*

$$\min_{\mathbf{U}, \mathbf{V}} J(\mathbf{U}, \mathbf{V}), \quad J(\mathbf{U}, \mathbf{V}) = \sum_{i,j \in \mathcal{S}} \left(R_{ij} - \mathbf{u}^{(i)T} \mathbf{v}^{(j)} \right)^2$$

$$\textcircled{1} \min_{\mathbf{U}} J(\mathbf{U}, \underline{\mathbf{V}})$$

$$\mathbf{U}^{t+1} \leftarrow \mathbf{U}^t - \alpha \nabla_{\mathbf{U}} J$$

$$\textcircled{2} \min_{\mathbf{V}} J(\underline{\mathbf{U}}, \mathbf{V})$$

$$\mathbf{V} \leftarrow \mathbf{V} - \alpha \nabla_{\mathbf{V}} J$$

Matrix Factorization: SGD

Method of *alternating minimization*

$$\min_{\mathbf{U}, \mathbf{V}} J(\mathbf{U}, \mathbf{V}) \quad J(\mathbf{U}, \mathbf{V}) = \sum_{i,j \in \mathcal{S}} \underbrace{\left(R_{ij} - \mathbf{u}^{(i)T} \mathbf{v}^{(j)} \right)^2}_{\text{red bracket}}$$

$$\begin{array}{l} \textcircled{1} \left[\min_{\mathbf{U}} J_{ij}(\mathbf{U}, \mathbf{V}) \right. \\ \textcircled{2} \left[\min_{\mathbf{V}} J_{ij} \right. \end{array} \quad \begin{array}{l} J_{ij}(\mathbf{U}, \mathbf{V}) = \underbrace{\left(R_{ij} - \mathbf{u}^{iT} \mathbf{v}^j \right)^2}_{\text{red bracket}} \\ \nabla_{\mathbf{u}} J_{ij} = - \left(R_{ij} - \mathbf{u}^{iT} \mathbf{v}^j \right) \mathbf{v}^j \end{array}$$

Matrix Factorization: SGD

Add regularization to avoid overfitting

$$\min_{\mathbf{U}, \mathbf{V}} J(\mathbf{U}, \mathbf{V}) \quad J(\mathbf{U}, \mathbf{V}) = \sum_{i,j \in \mathcal{S}} \left[\left(R_{ij} - \mathbf{u}^{(i)T} \mathbf{v}^{(j)} \right)^2 + \lambda \|\mathbf{v}^{(j)}\|_2^2 + \lambda \|\mathbf{u}^{(i)}\|_2^2 \right]$$

Summary

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

Collaborative filtering by Matrix Factorization (MF) is an **efficient** and **effective** approach

(SVD for MF is a bit broken)

MF is just another example of a **common recipe**:

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

Optimization

- Need alternating minimization
- Add regularization to avoid overfitting