

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?

Background: Low Dimensional Embeddings

Why might low dimensional embeddings be useful?

- Example: MNIST digit classification with nearest neighbor

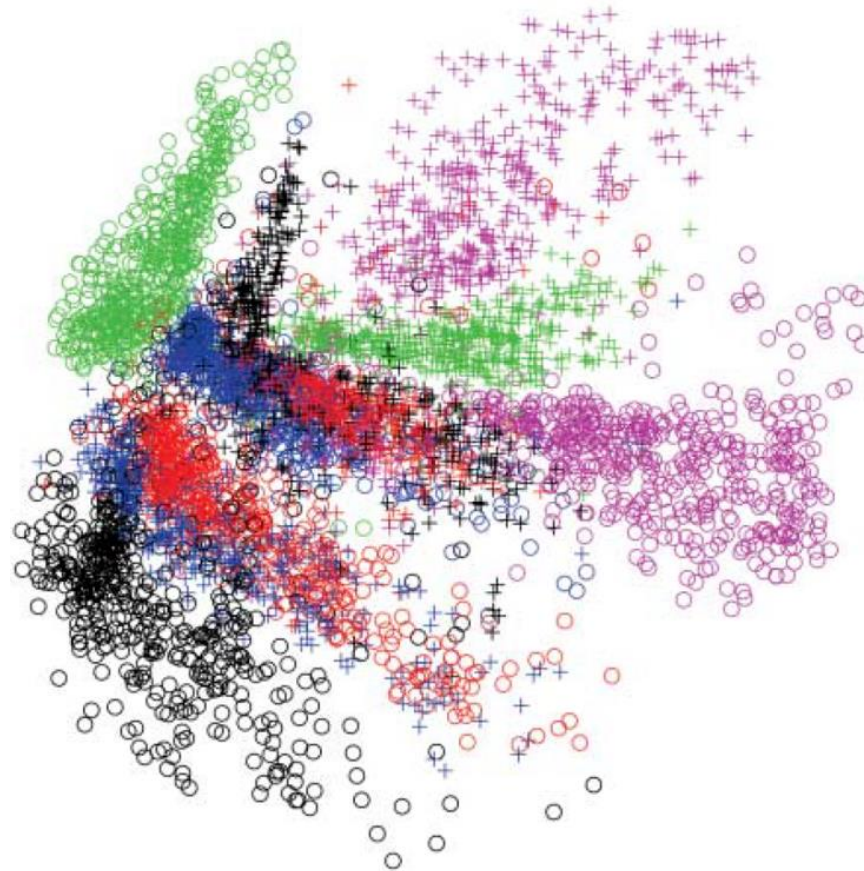
2



Background: Low Dimensional Embeddings

Why might low dimensional embeddings be useful?

- Example: MNIST digit classification with nearest neighbor



Background: Measure of Similarity

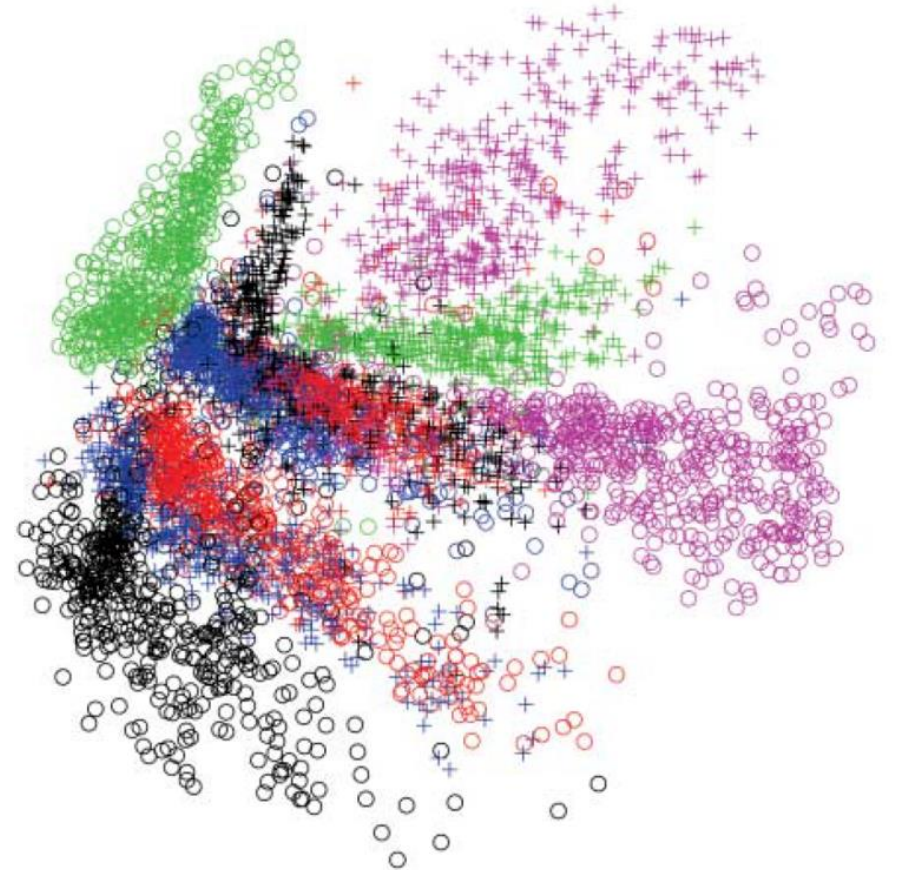
We've been using Euclidean distance

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

Cosine similarity

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

- $d(\mathbf{x}, \mathbf{z}) = \mathbf{x}^T \mathbf{z}$



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, maple syrup, and other grocery items.
- Buy It Again in Pets** (6 ITEMS): Includes images of pet products like Advantage II flea treatment and pet food.
- Buy It Again in Baby Products** (5 ITEMS): Includes images of baby products like crayons, pacifiers, and baby wipes.
- Engineering Books** (86 ITEMS): Includes the book cover for "Probabilistic Graphical Models: Principles and Techniques" by Daphne Koller and Nir Friedman.

NETFLIX

COMPLETED

[Home](#)
[Rules](#)
[Leaderboard](#)
[Update](#)

Congratulations!

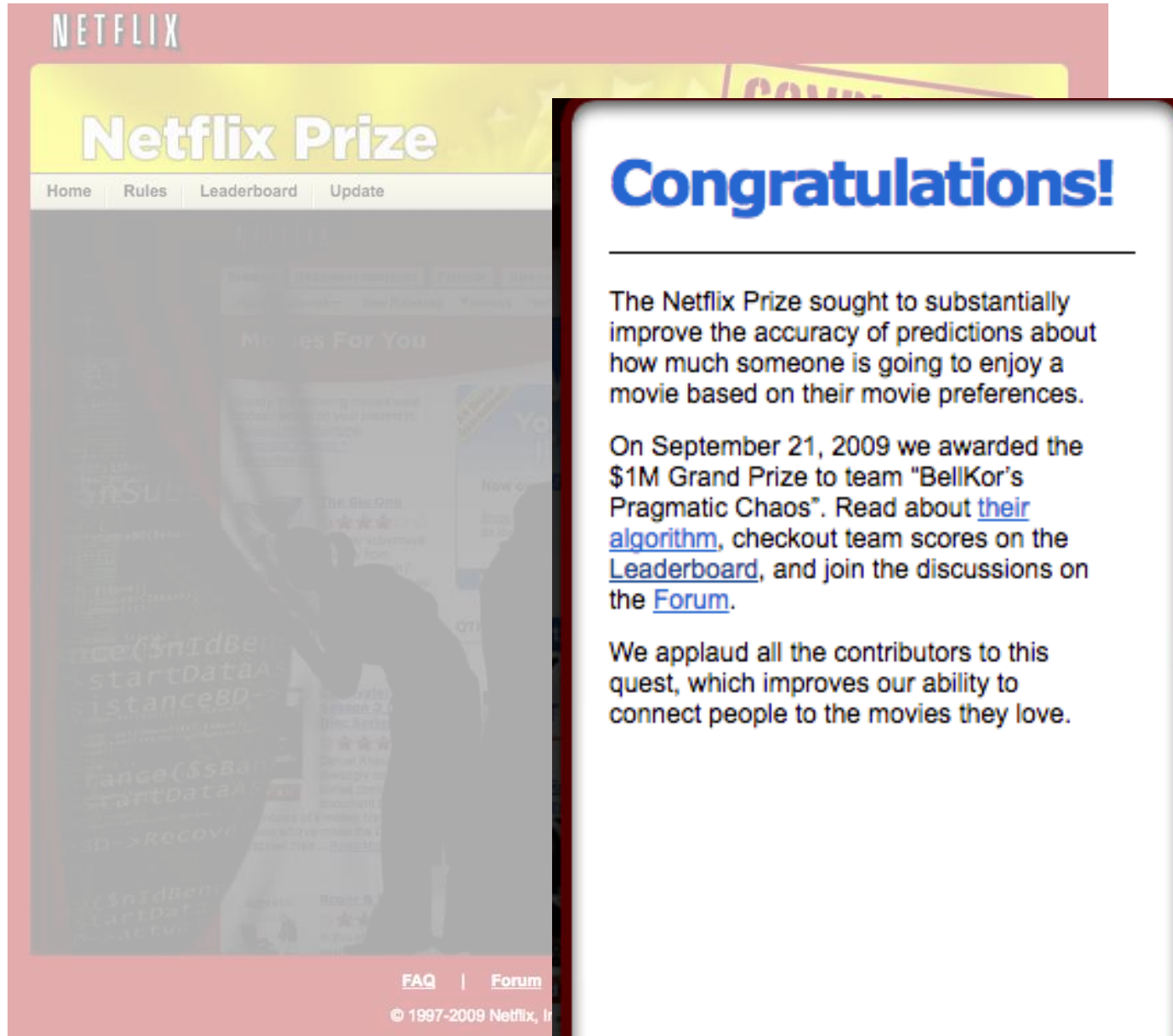
On September 21, 2009 we awarded the \$1M Grand Prize to team "BellKor's Pragmatic Chaos". Read about [their algorithm](#), checkout team scores on the [Leaderboard](#), and join the discussions on the [Forum](#).

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

[FAQ](#) | [Forum](#) | [Netflix Home](#)

© 1997-2009 Netflix, Inc. All rights reserved.

Recommender Systems



The image shows a screenshot of the Netflix Prize website. The background is a faded view of the site's interface, including the 'Netflix Prize' title, navigation links like 'Home', 'Rules', 'Leaderboard', and 'Update', and a section titled 'Movies For You'. Overlaid on the right side of the screenshot is a white box with a dark red border containing a 'Congratulations!' message. The message text is as follows:

Congratulations!

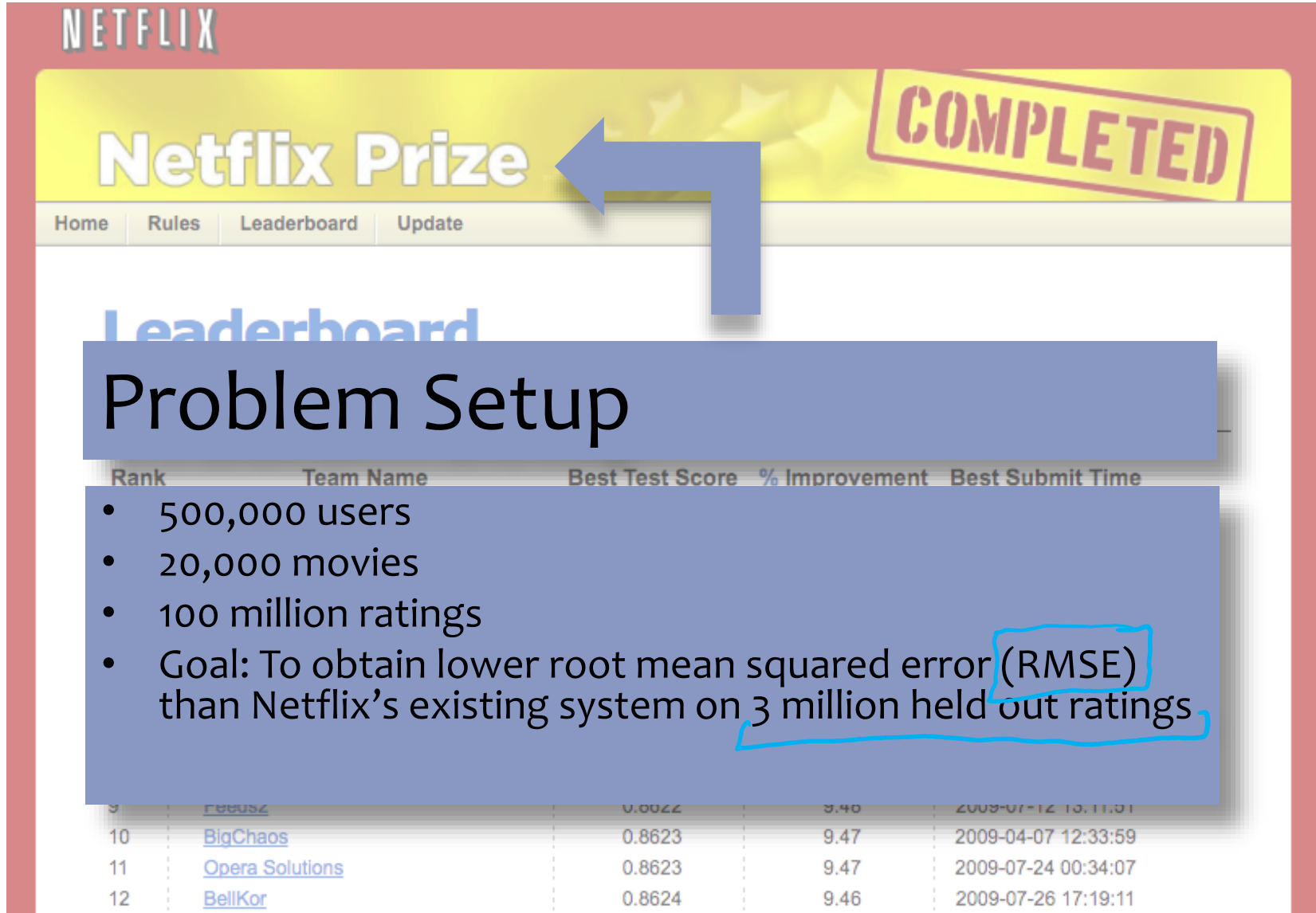
The Netflix Prize sought to substantially improve the accuracy of predictions about how much someone is going to enjoy a movie based on their movie preferences.

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.

At the bottom of the screenshot, the footer includes links for 'FAQ' and 'Forum', and a copyright notice: '© 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 the top 12 teams, with the first team being "Feus2" and the last team being "BellKor".

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

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

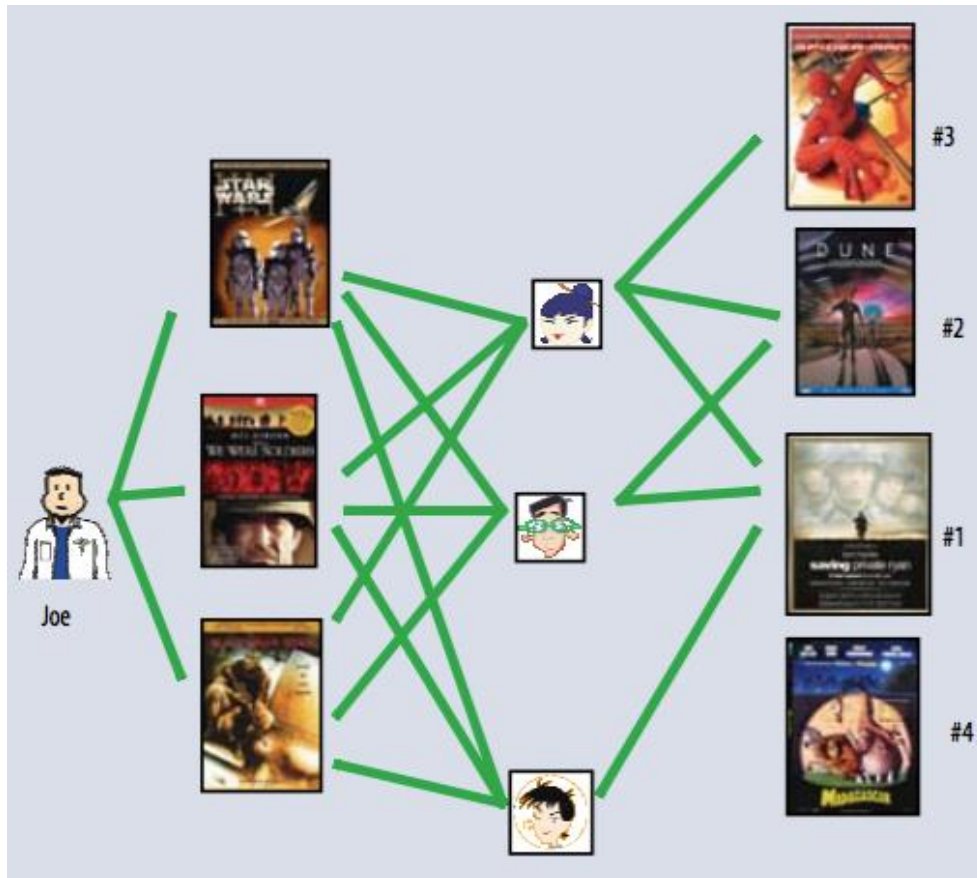
COLLABORATIVE FILTERING

Collaborative Filtering

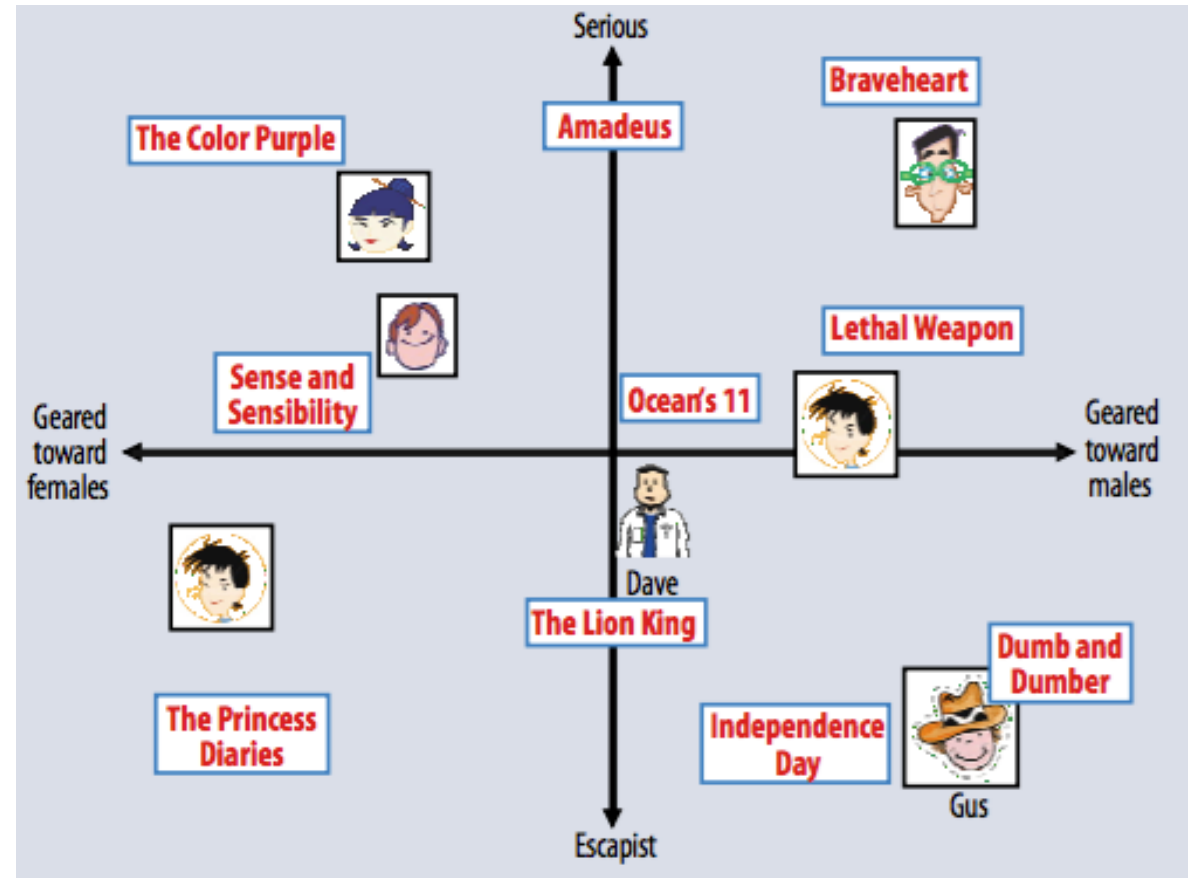
- **Everyday Examples of Collaborative Filtering...**
 - Bestseller lists
 - Top 40 music lists
 - The “recent returns” shelf at the library
 - Unmarked but well-used paths thru the woods
 - The printer room at work
 - “Read any good books lately?”
 - ...
- **Common insight:** personal tastes are correlated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

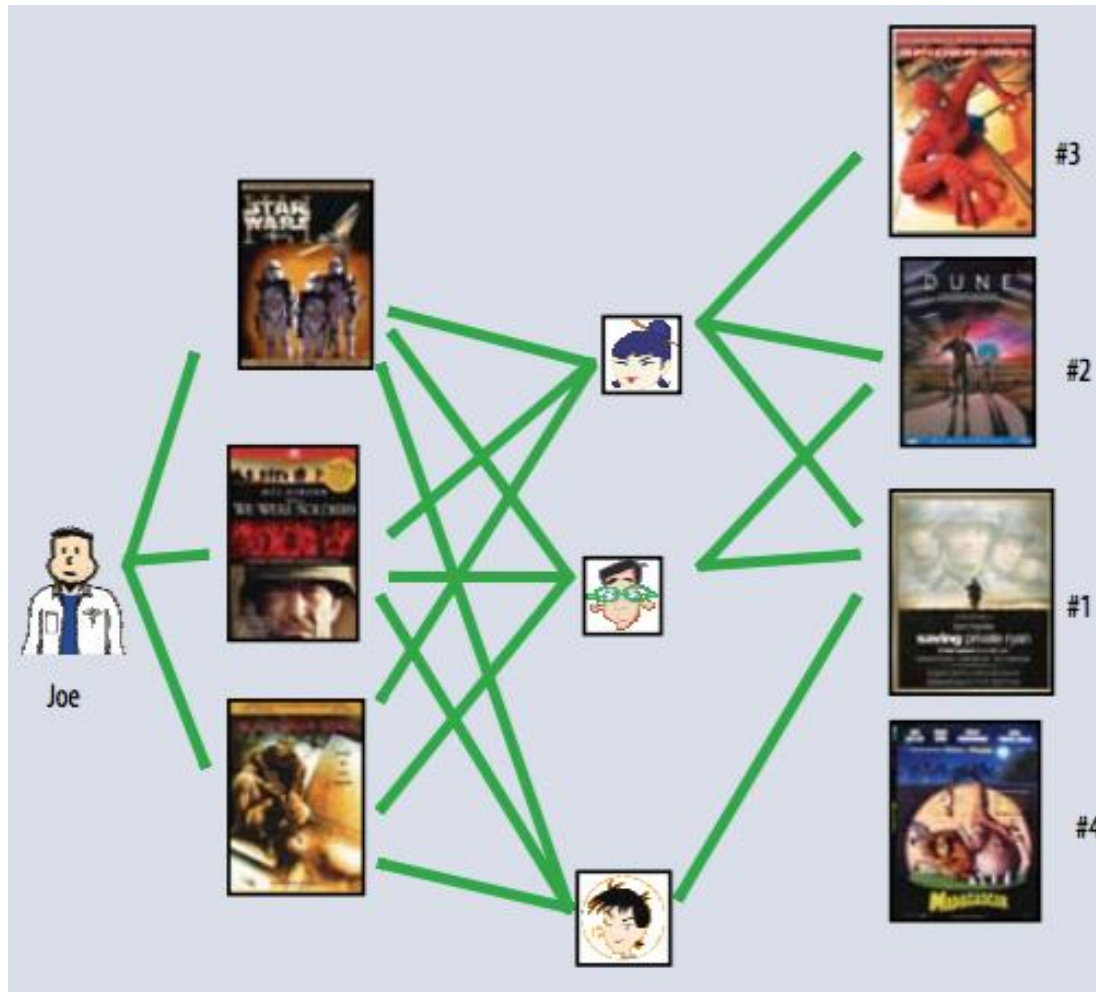


2. Latent Factor Methods



Two Types of Collaborative Filtering

1. Neighborhood Methods



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

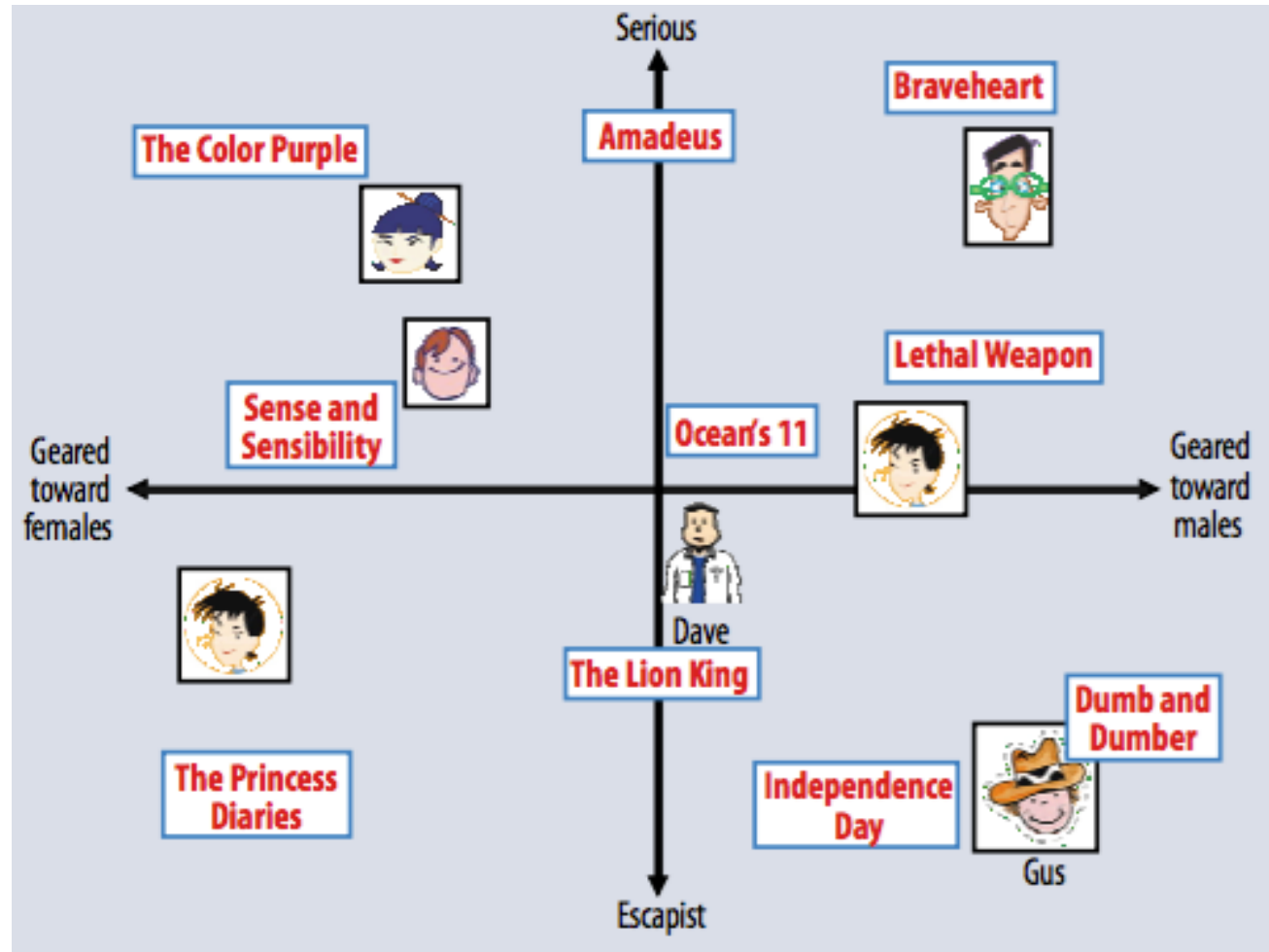
Algorithm:

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

Two Types of Collaborative Filtering

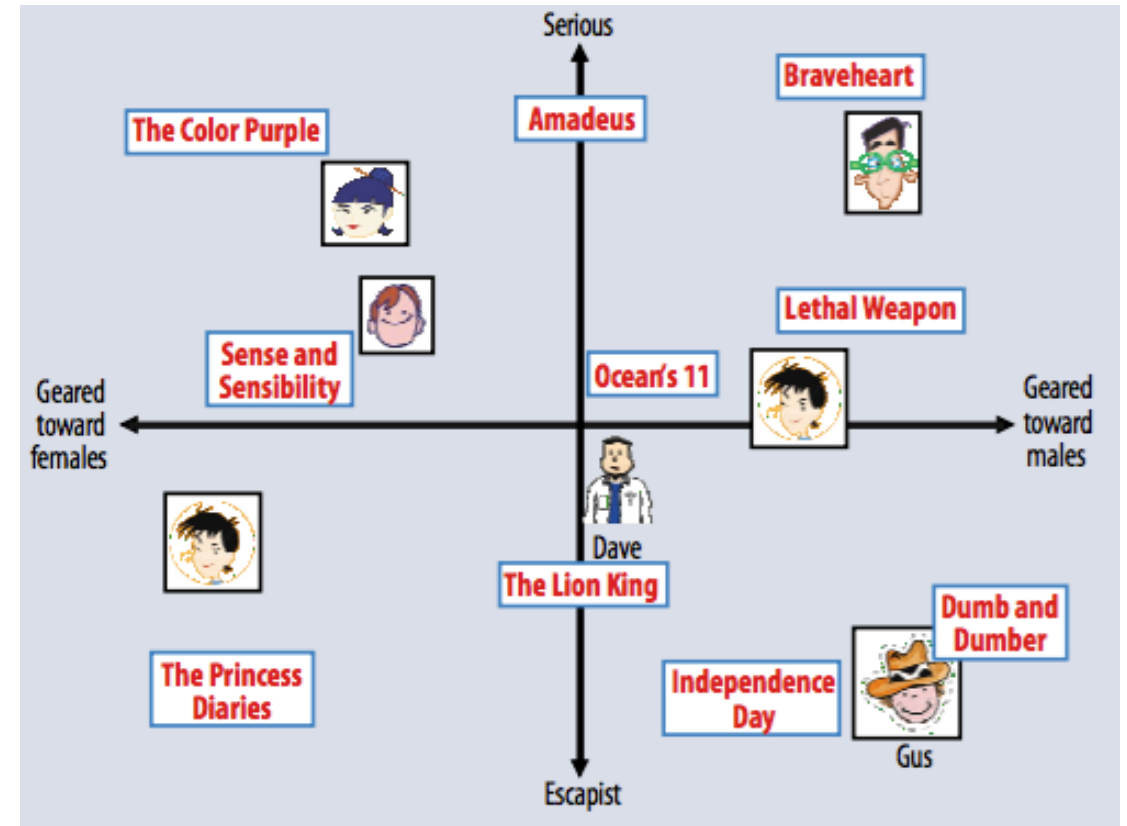
2. Latent Factor Methods

- 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



Recommender System: Matrix Factorization

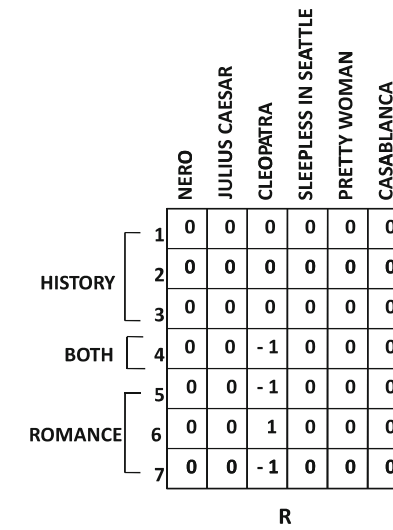
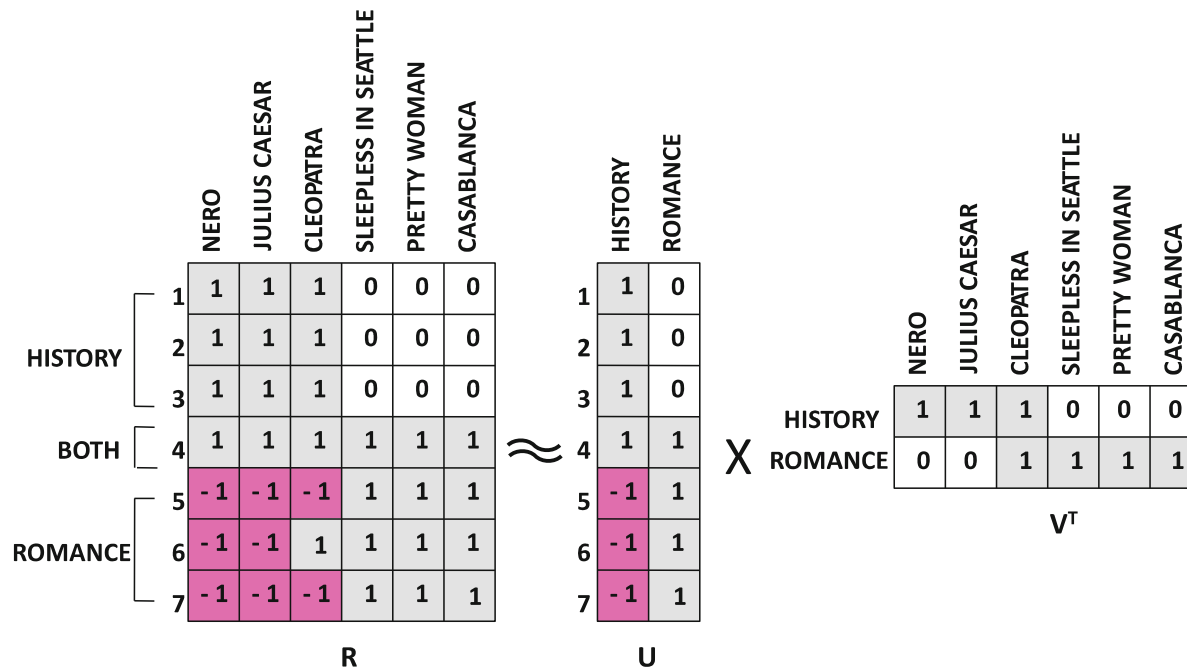
Optimization

Recommender System: Matrix Factorization

Sparse labels 😞

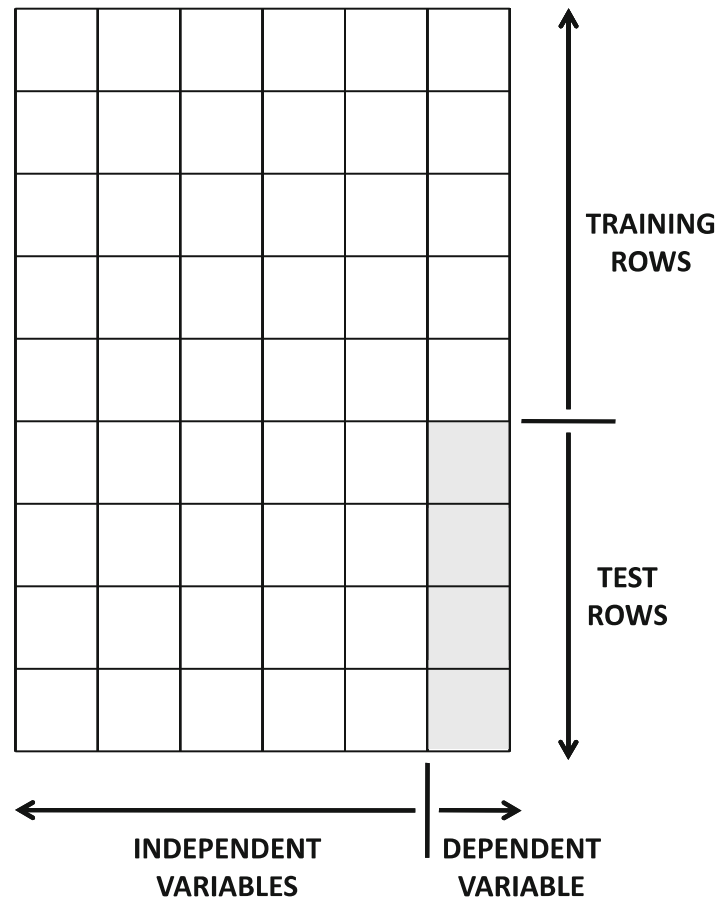
	Doctor Strange	Star Trek: Beyond	Zootopia
Alita	1		5
BB-8	3	4	
C-3Po	3	5	2

Example: MF for Netflix Problem

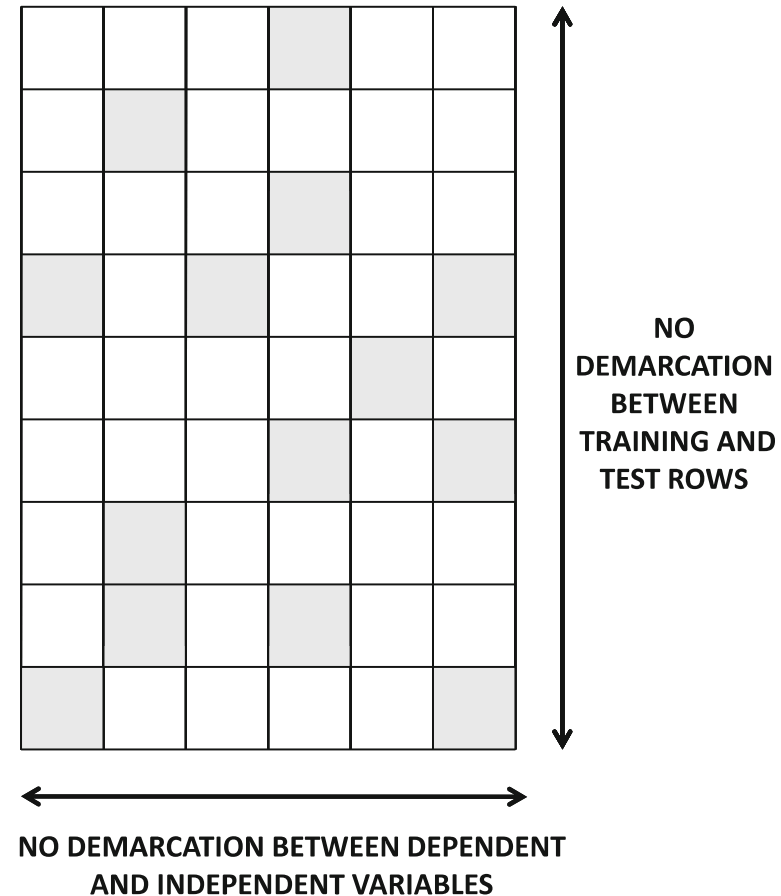


Regression vs. Collaborative Filtering

Regression



Collaborative Filtering



Matrix Factorization: SVD

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

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

Alita			
BB-8			
C-3Po			

Matrix Factorization: SGD

Objective function using only the labels we have

Piazza Poll 1

Is the following optimization a quadratic optimization?

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

A.

B.

C.

Matrix Factorization: SGD

Method of *alternating minimization*

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

Matrix Factorization: SGD

Method of *alternating minimization*

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

Matrix Factorization: SGD

Add regularization to avoid overfitting

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

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