



10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Matrix Factorization and Collaborative Filtering

MF Readings:

(Koren et al., 2009)

Matt Gormley
Lecture 25
April 19, 2017

Reminders

- **Homework 8: Graphical Models**
 - Release: Mon, Apr. 17
 - Due: Mon, Apr. 24 at 11:59pm
- **Homework 9: Applications of ML**
 - Release: Mon, Apr. 24
 - Due: Wed, May 3 at 11:59pm

Outline

- **Recommender Systems**
 - Content Filtering
 - Collaborative Filtering (CF)
 - CF: Neighborhood Methods
 - CF: Latent Factor Methods
- **Matrix Factorization**
 - Background: Low-rank Factorizations
 - Residual matrix
 - Unconstrained Matrix Factorization
 - Optimization problem
 - Gradient Descent, SGD, Alternating Least Squares
 - User/item bias terms (matrix trick)
 - Singular Value Decomposition (SVD)
 - Non-negative Matrix Factorization
- **Extra: Matrix Multiplication in ML**
 - Matrix Factorization
 - Linear Regression
 - PCA
 - (Autoencoders)
 - K-means

RECOMMENDER SYSTEMS

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

NEW & INTERESTING FINDS ON AMAZON

EXPLORE

amazon Prime

All ▾



CYBER MONDAY DEALS WEEK

Departments ▾

Browsing History ▾

Matt's Amazon.com

Cyber Monday

Gift Cards & Registry

Sell

Help

Hello, Matt

Your Account ▾

Prime ▾

Lists ▾



Your Amazon.com

Your Browsing History

Recommended For You

Improve Your Recommendations

Your Profile

Learn More



Matt's Amazon

You could be seeing useful stuff here!

Sign in to get your order status, balances and rewards.

Sign In

Recommended for you, Matt



Buy It Again in Grocery

14 ITEMS



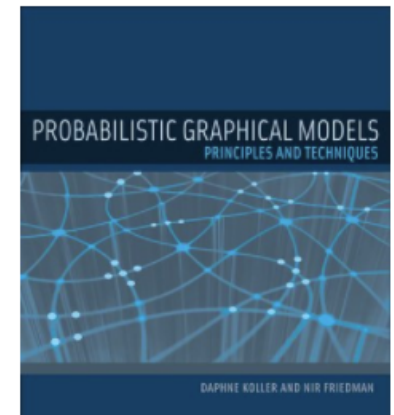
Buy It Again in Pets

6 ITEMS



Buy It Again in Baby Products

5 ITEMS



Engineering Books

86 ITEMS

Recommender Systems



The image shows a screenshot of the Netflix website during the Netflix Prize competition. A large yellow banner at the top features the text "Netflix Prize" and a red stamp that says "COMPLETED". Below the banner is a navigation menu with links for "Home", "Rules", "Leaderboard", and "Update". The main content area is dimmed, showing a "Movies For You" section with various movie recommendations. A white box with a red border is overlaid on the right side of the page, containing a "Congratulations!" message and details about the \$1M Grand Prize awarded to team "BellKor's Pragmatic Chaos".

NETFLIX

Netflix Prize

COMPLETED

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 | Netflix Home

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

Recommender Systems



The image shows a screenshot of the Netflix Prize website. The main page features the Netflix logo at the top left, followed by a yellow banner with the text "Netflix Prize". Below this is a navigation menu with links for "Home", "Rules", "Leaderboard", and "Update". The main content area is titled "Movies For You" and displays a list of recommended movies. An inset box on the right side of the page contains a congratulatory message.

NETFLIX

Netflix Prize

Home Rules Leaderboard Update

NETFLIX

Home Recommendations Friends Clubs

Home Search New Releases Reviews

Movies For You

Handy: The following movies were chosen based on your interest in watching the Columbia TriStar Season 3

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.

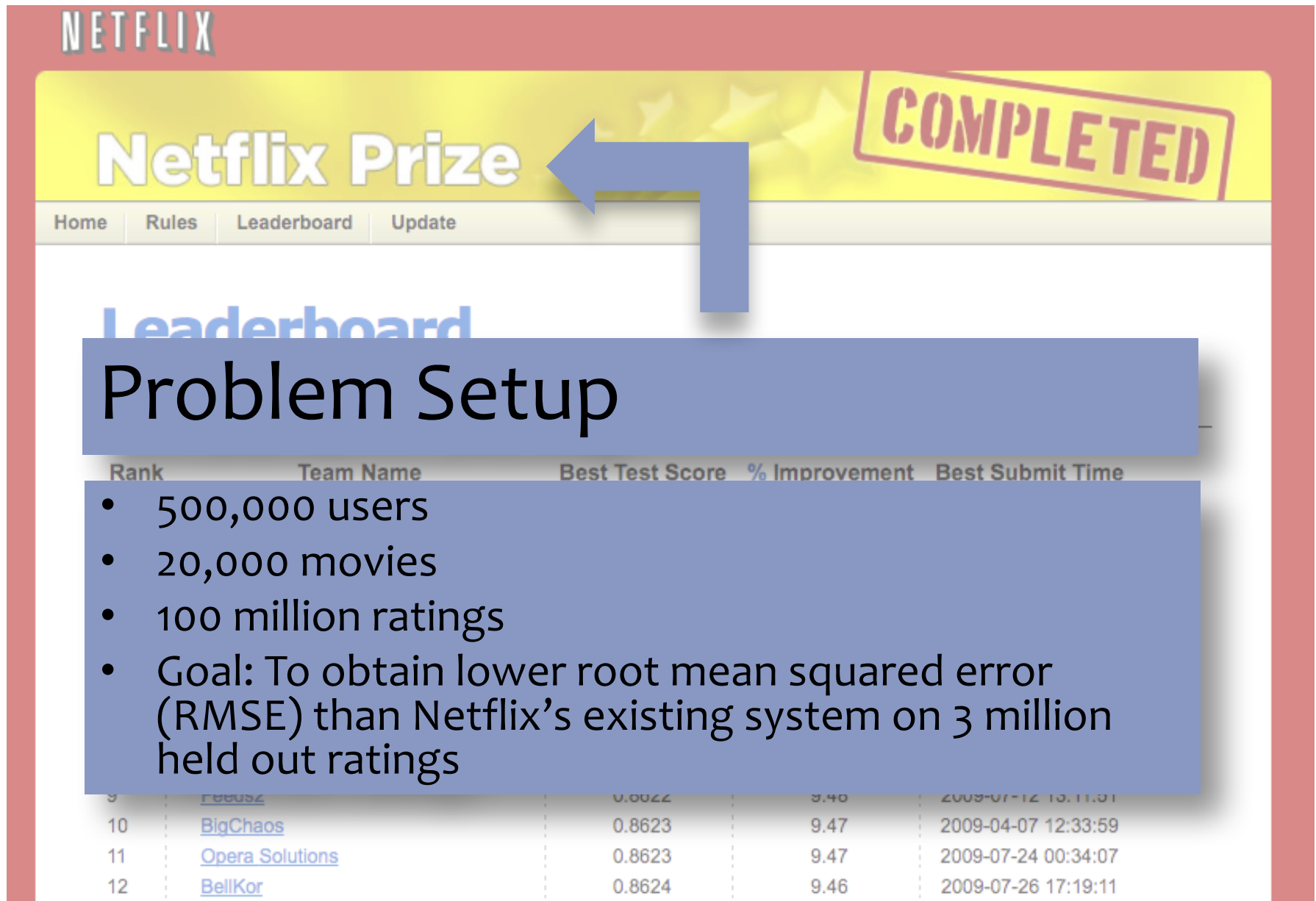
On September 21, 2009 we awarded the \$1M Grand Prize to team "BellKor's Pragmatic Chaos". Read about [their algorithm](#), checkout team scores on the [Leaderboard](#), and join the discussions on the [Forum](#).

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



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	9.46	2009-07-12 13:11:51
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

NETFLIX

Netfix Prize

COMPLETED

Home Rules Leaderboard Update

Leaderboard

Showing Test Score. [Click here to show quiz score](#)

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
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
Alice	1		5
Bob	3	4	
Charlie	3	5	2

Recommender Systems

NETFLIX

Netfix Prize

COMPLETED

Home Rules Leaderboard Update

Leaderboard

Showing Test Score. [Click here to show quiz score](#)

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.8594	9.77	2009-06-24 12:06:56
7	BellKor in BigChaos	0.8601	9.70	2009-05-13 08:14:09
8	Dace	0.8612	9.59	2009-07-24 17:18:43
9	Feeds2	0.8622	9.48	2009-07-12 13:11:51
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

Two Types of Recommender Systems

Content Filtering

- *Example:* **Pandora.com** music recommendations (Music Genome Project)
- **Con:** Assumes access to **side information** about items (e.g. properties of a song)
- **Pro:** Got a **new item** to add? No problem, just be sure to include the side information

Collaborative Filtering

- *Example:* **Netflix** movie recommendations
- **Pro:** Does not assume access to **side information** about items (e.g. does not need to know about movie genres)
- **Con:** Does not work on **new items** that have no ratings

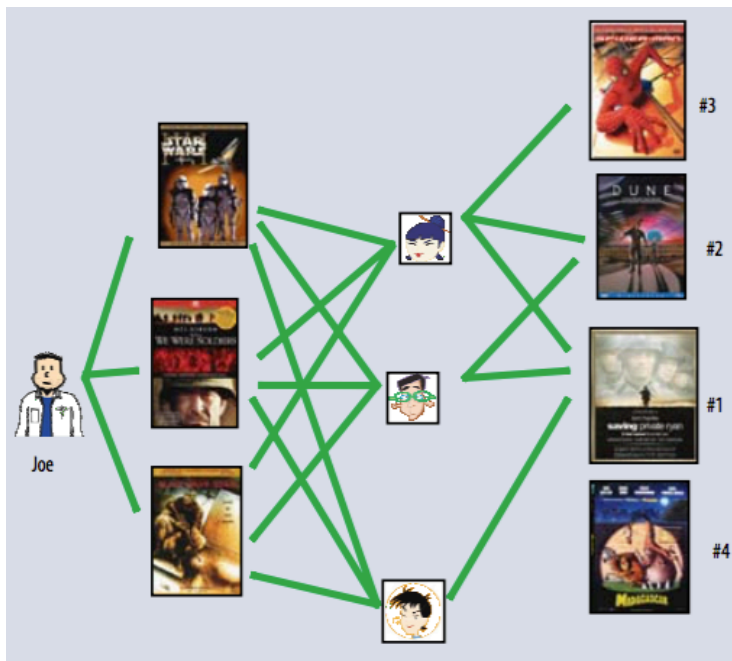
COLLABORATIVE FILTERING

Collaborative Filtering

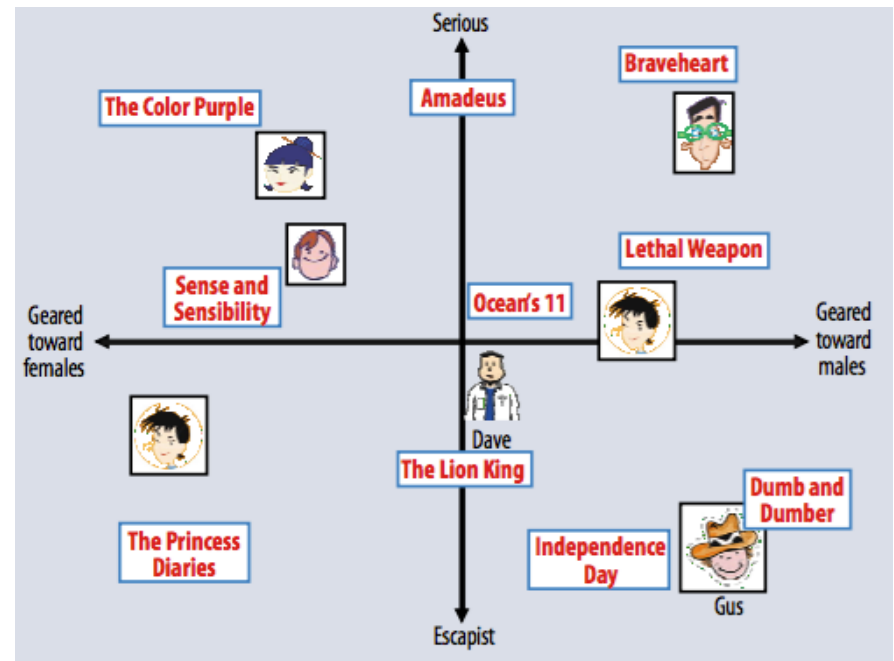
- **Everyday Examples of Collaborative Filtering...**
 - Bestseller lists
 - Top 40 music lists
 - The “recent returns” shelf at the library
 - Unmarked but well-used paths thru the woods
 - The printer room at work
 - “Read any good books lately?”
 - ...
- **Common insight:** personal tastes are correlated
 - If Alice and Bob both like X and Alice likes Y then Bob is more likely to like Y
 - especially (perhaps) if Bob knows Alice

Two Types of Collaborative Filtering

1. Neighborhood Methods

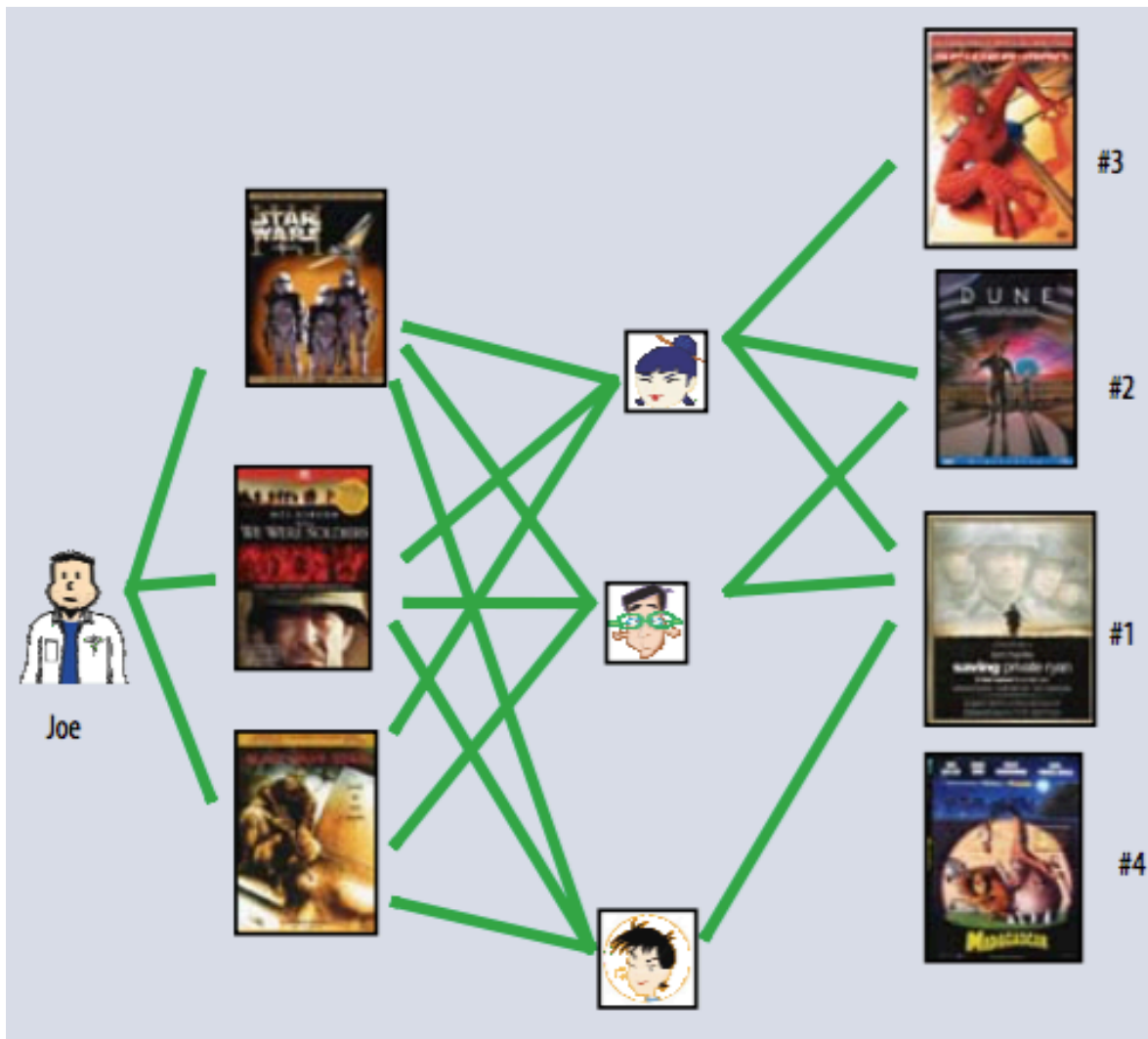


2. Latent Factor Methods



Two Types of Collaborative Filtering

1. Neighborhood Methods



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

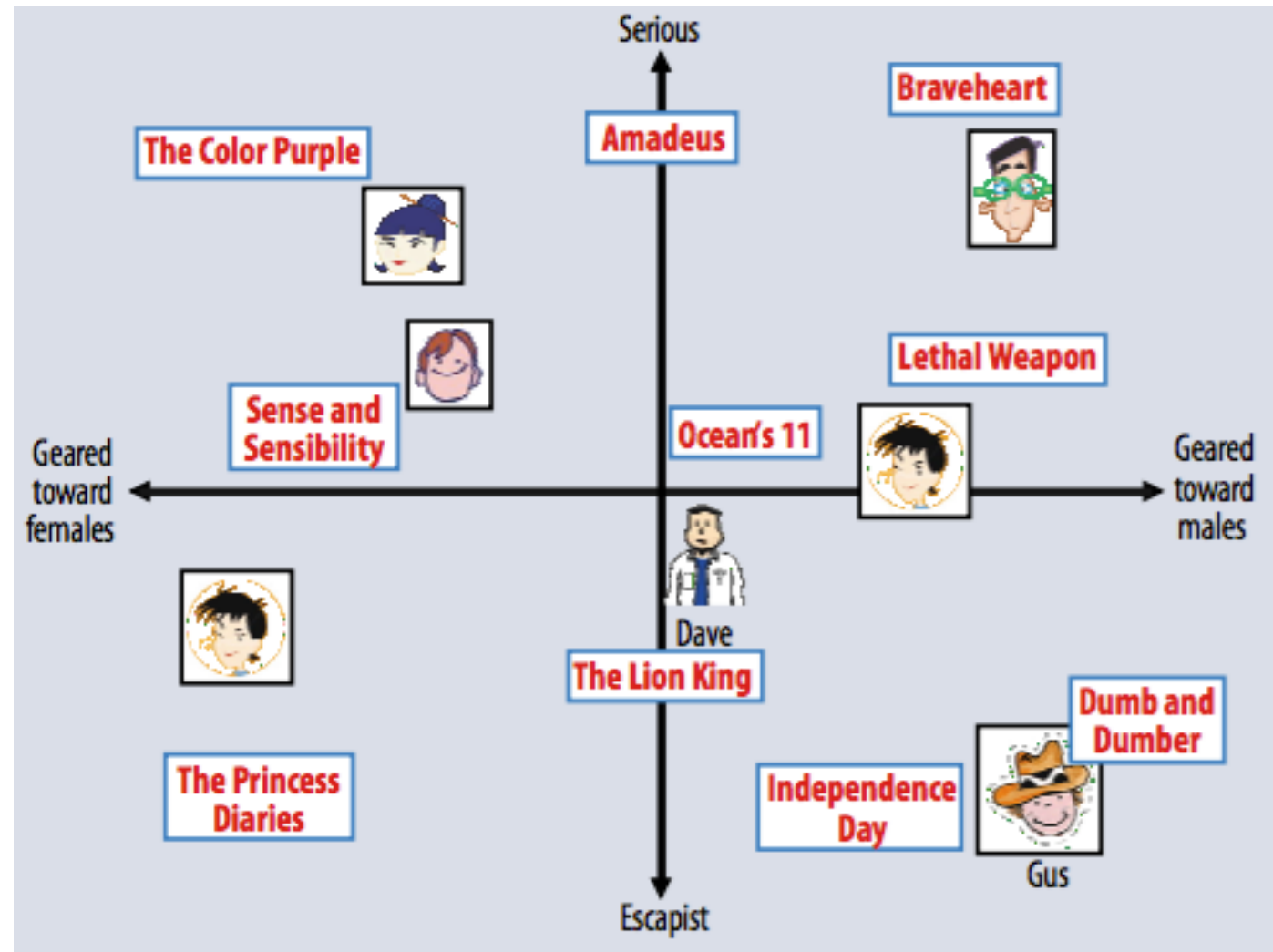
Algorithm:

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



MATRIX FACTORIZATION

Matrix Factorization

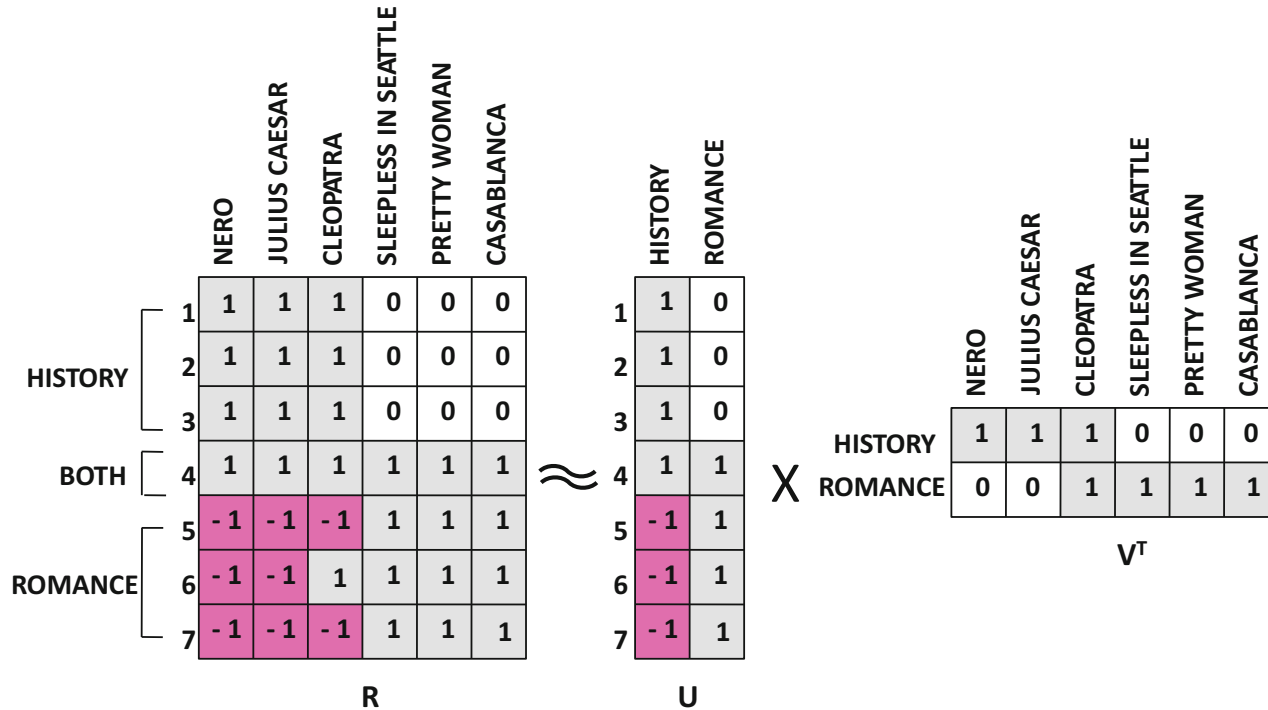
- Many different ways of factorizing a matrix
- We'll consider three:
 1. Unconstrained Matrix Factorization
 2. Singular Value Decomposition
 3. Non-negative Matrix Factorization
- MF is just another example of a **common recipe**:
 1. define a model
 2. define an objective function
 3. optimize with SGD

Matrix Factorization

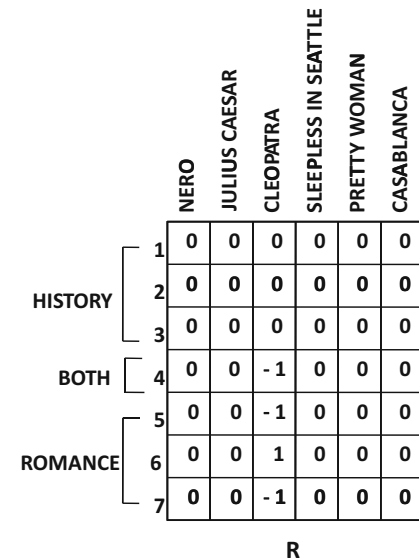
Whiteboard

- Background: Low-rank Factorizations
- Residual matrix

Example: MF for Netflix Problem



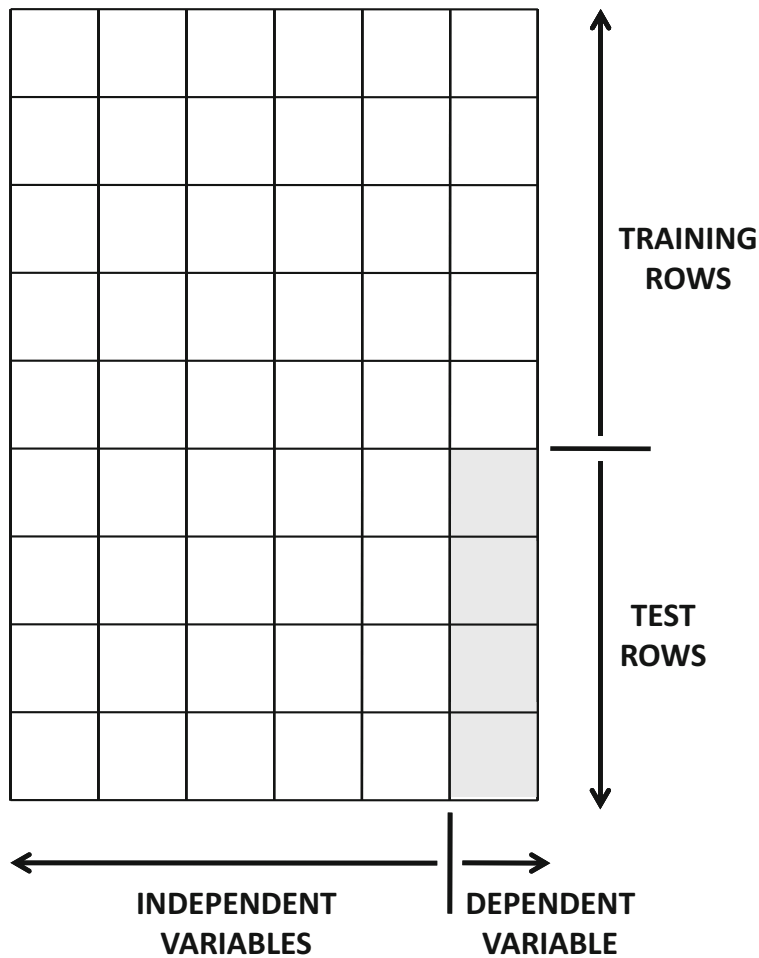
(a) Example of rank-2 matrix factorization



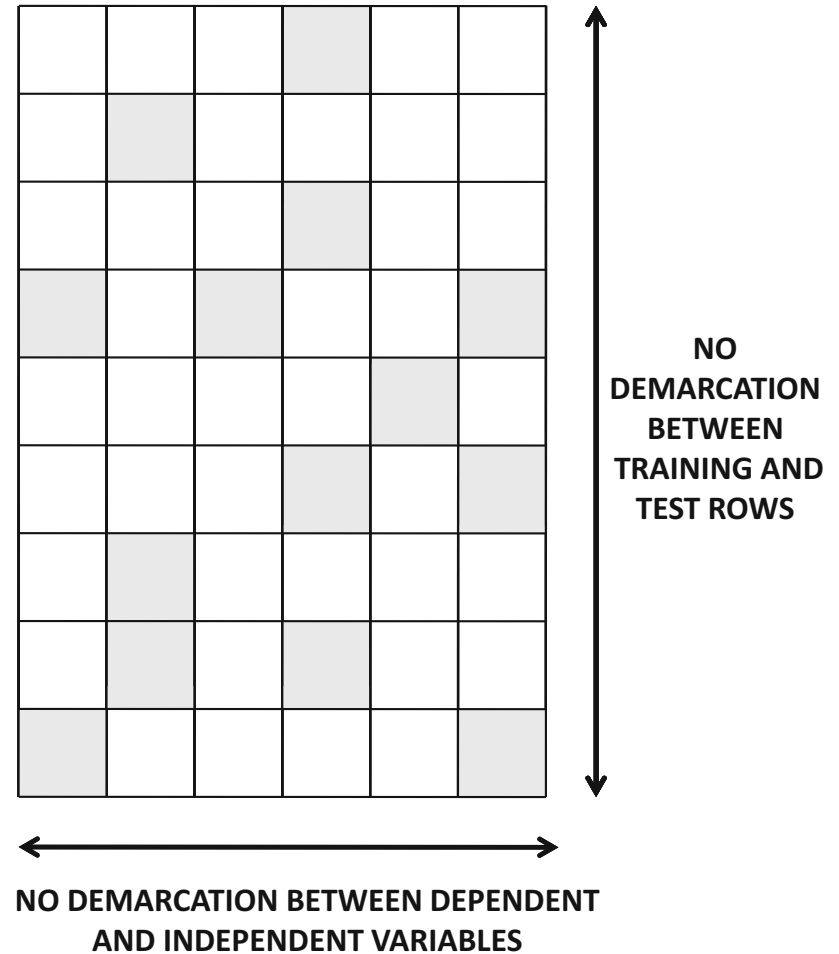
(b) Residual matrix

Regression vs. Collaborative Filtering

Regression



Collaborative Filtering



UNCONSTRAINED MATRIX FACTORIZATION

Unconstrained Matrix Factorization

Whiteboard

- Optimization problem
- SGD
- SGD with Regularization
- Alternating Least Squares
- User/item bias terms (matrix trick)

Unconstrained Matrix Factorization

In-Class Exercise

Derive a block coordinate descent algorithm for the Unconstrained Matrix Factorization problem.

- User vectors:

$$\mathbf{w}_u \in \mathbb{R}^r$$

- Item vectors:

$$\mathbf{h}_i \in \mathbb{R}^r$$

- Rating prediction:

$$v_{ui} = \mathbf{w}_u^T \mathbf{h}_i$$

- Set of non-missing entries

$$\mathcal{Z} = \{(u, i) : v_{ui} \text{ is observed}\}$$

- Objective:

$$\operatorname{argmin}_{\mathbf{w}, \mathbf{h}} \sum_{(u, i) \in \mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2$$

Matrix Factorization (with matrices)

- User vectors:

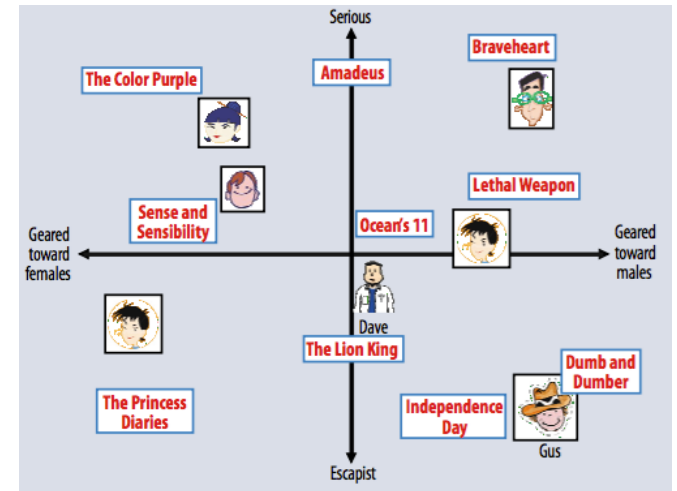
$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

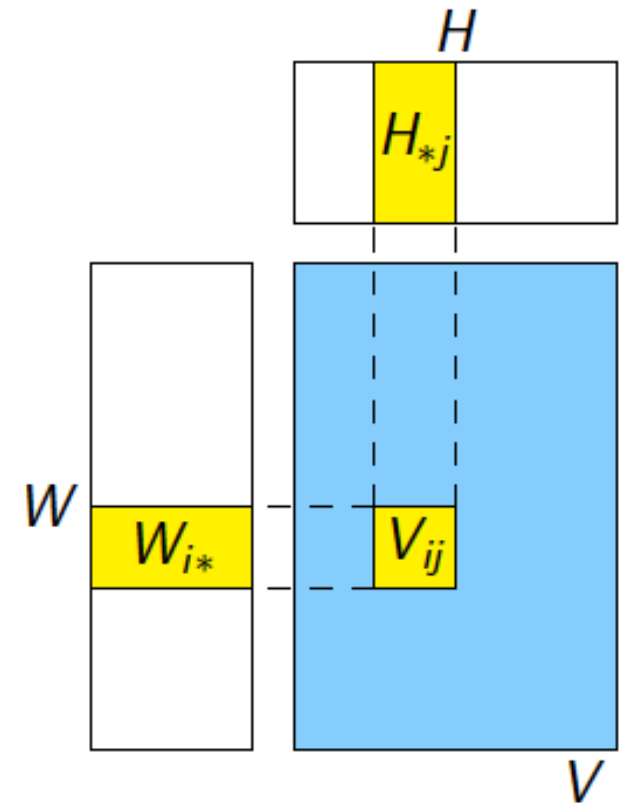
$$H_{*i} \in \mathbb{R}^r$$

- Rating prediction:

$$\begin{aligned} V_{ui} &= W_{u*} H_{*i} \\ &= [WH]_{ui} \end{aligned}$$



Figures from Koren et al. (2009)



Figures from Gemulla et al. (2011)₃₃

Matrix Factorization (with vectors)

- User vectors:

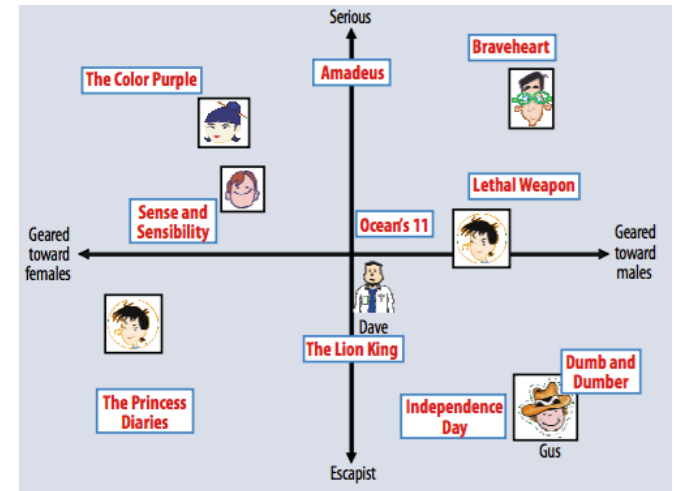
$$\mathbf{w}_u \in \mathbb{R}^r$$

- Item vectors:

$$\mathbf{h}_i \in \mathbb{R}^r$$

- Rating prediction:

$$v_{ui} = \mathbf{w}_u^T \mathbf{h}_i$$

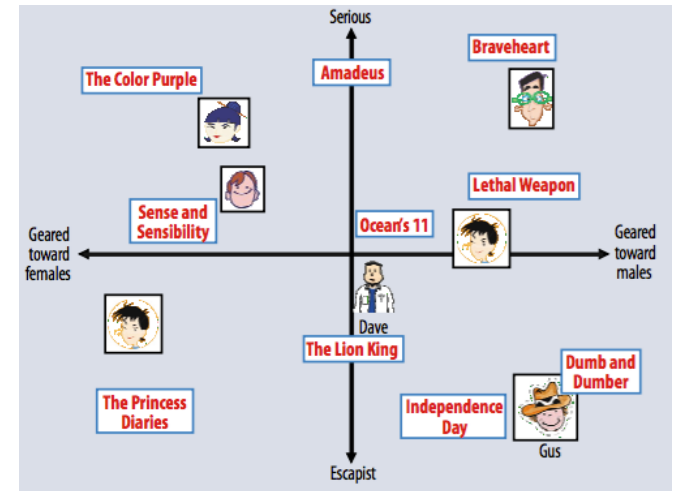


Figures from Koren et al. (2009)

Matrix Factorization (with vectors)

- Set of non-missing entries:
 $\mathcal{Z} = \{(u, i) : v_{ui} \text{ is observed}\}$
- Objective:

$$\operatorname{argmin}_{\mathbf{w}, \mathbf{h}} \sum_{(u, i) \in \mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2$$

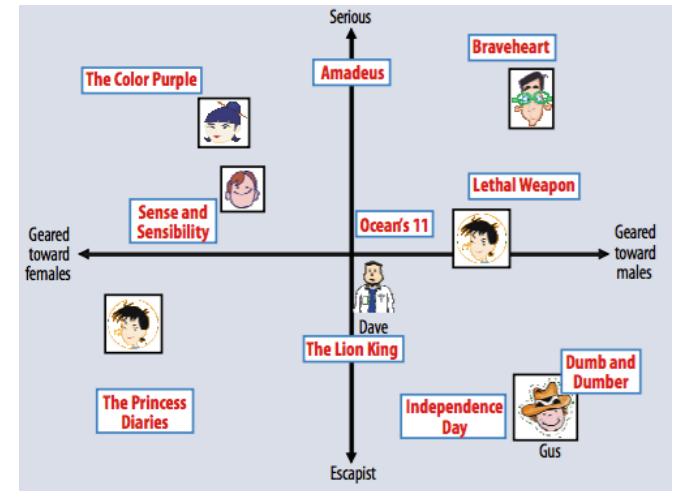


Figures from Koren et al. (2009)

Matrix Factorization (with vectors)

- Regularized Objective:

$$\begin{aligned} \operatorname{argmin}_{\mathbf{w}, \mathbf{h}} \sum_{(u, i) \in \mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 \\ + \lambda \left(\sum_i \|\mathbf{w}_i\|^2 + \sum_u \|\mathbf{h}_u\|^2 \right) \end{aligned}$$



Figures from Koren et al. (2009)

Matrix Factorization (with vectors)

- Regularized Objective:

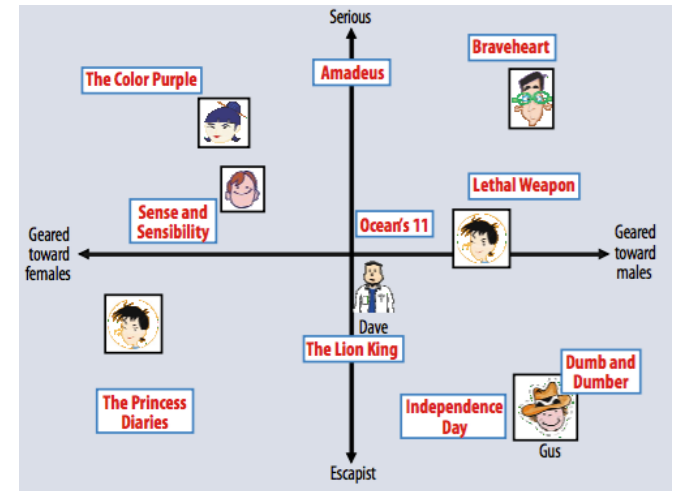
$$\begin{aligned} \operatorname{argmin}_{\mathbf{w}, \mathbf{h}} \sum_{(u,i) \in \mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 \\ + \lambda \left(\sum_i \|\mathbf{w}_i\|^2 + \sum_u \|\mathbf{h}_u\|^2 \right) \end{aligned}$$

- SGD update for random (u,i) :

$$e_{ui} \leftarrow v_{ui} - \mathbf{w}_u^T \mathbf{h}_i$$

$$\mathbf{w}_u \leftarrow \mathbf{w}_u + \gamma (e_{ui} \mathbf{h}_i - \lambda \mathbf{w}_u)$$

$$\mathbf{h}_i \leftarrow \mathbf{h}_i + \gamma (e_{ui} \mathbf{w}_u - \lambda \mathbf{h}_i)$$



Figures from Koren et al. (2009)

Matrix Factorization (with matrices)

- User vectors:

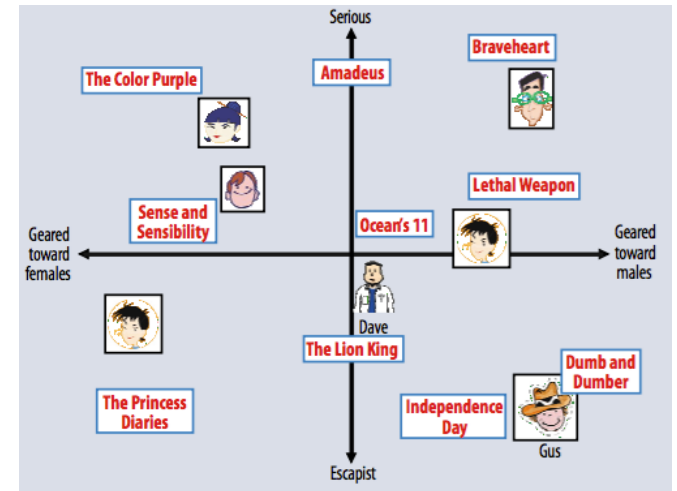
$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

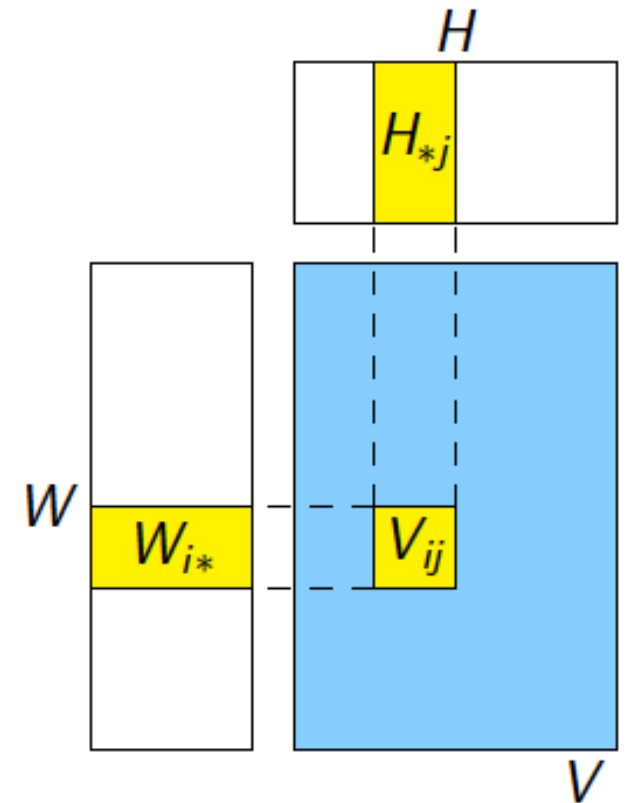
$$H_{*i} \in \mathbb{R}^r$$

- Rating prediction:

$$\begin{aligned} V_{ui} &= W_{u*} H_{*i} \\ &= [WH]_{ui} \end{aligned}$$



Figures from Koren et al. (2009)



Figures from Gemulla et al. (2011)₃₈

Matrix Factorization (with matrices)

- SGD

require that the loss can be written as

$$L = \sum_{(i,j) \in Z} l(\mathbf{V}_{ij}, \mathbf{W}_{i*}, \mathbf{H}_{*j})$$

Algorithm 1 SGD for Matrix Factorization

Require: A training set Z , initial values \mathbf{W}_0 and \mathbf{H}_0

while not converged **do** {step}

 Select a training point $(i, j) \in Z$ uniformly at random.

$$\mathbf{W}'_{i*} \leftarrow \mathbf{W}_{i*} - \epsilon_n N \frac{\partial}{\partial \mathbf{W}_{i*}} l(\mathbf{V}_{ij}, \mathbf{W}_{i*}, \mathbf{H}_{*j})$$

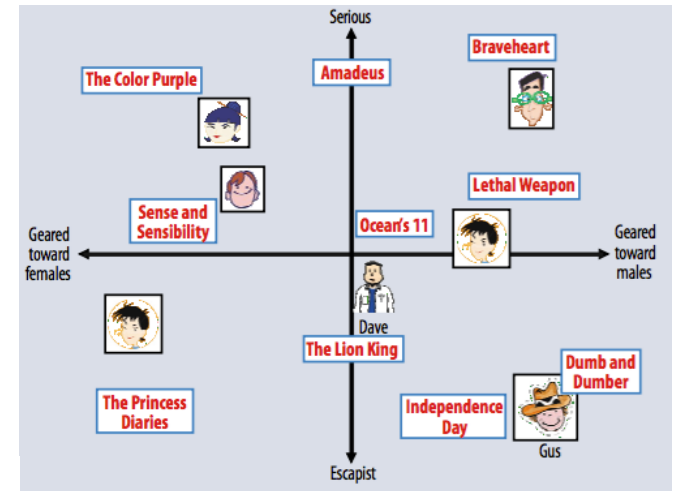
$$\mathbf{H}_{*j} \leftarrow \mathbf{H}_{*j} - \epsilon_n N \frac{\partial}{\partial \mathbf{H}_{*j}} l(\mathbf{V}_{ij}, \mathbf{W}_{i*}, \mathbf{H}_{*j})$$

$$\mathbf{W}_{i*} \leftarrow \mathbf{W}'_{i*}$$

end while

← step size

Figure from Gemulla et al. (2011)



Figures from Koren et al. (2009)

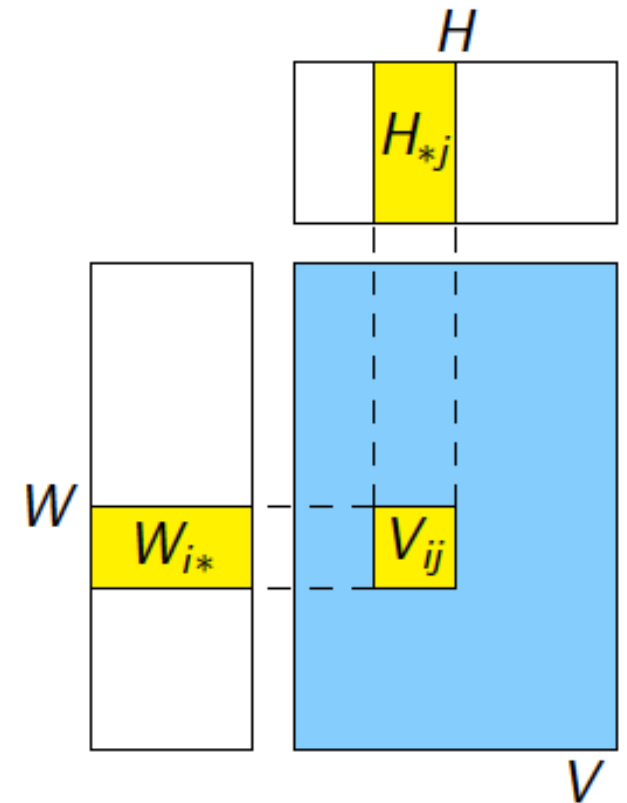


Figure from Gemulla et al. (2011)₃₉

Matrix Factorization

Example Factors

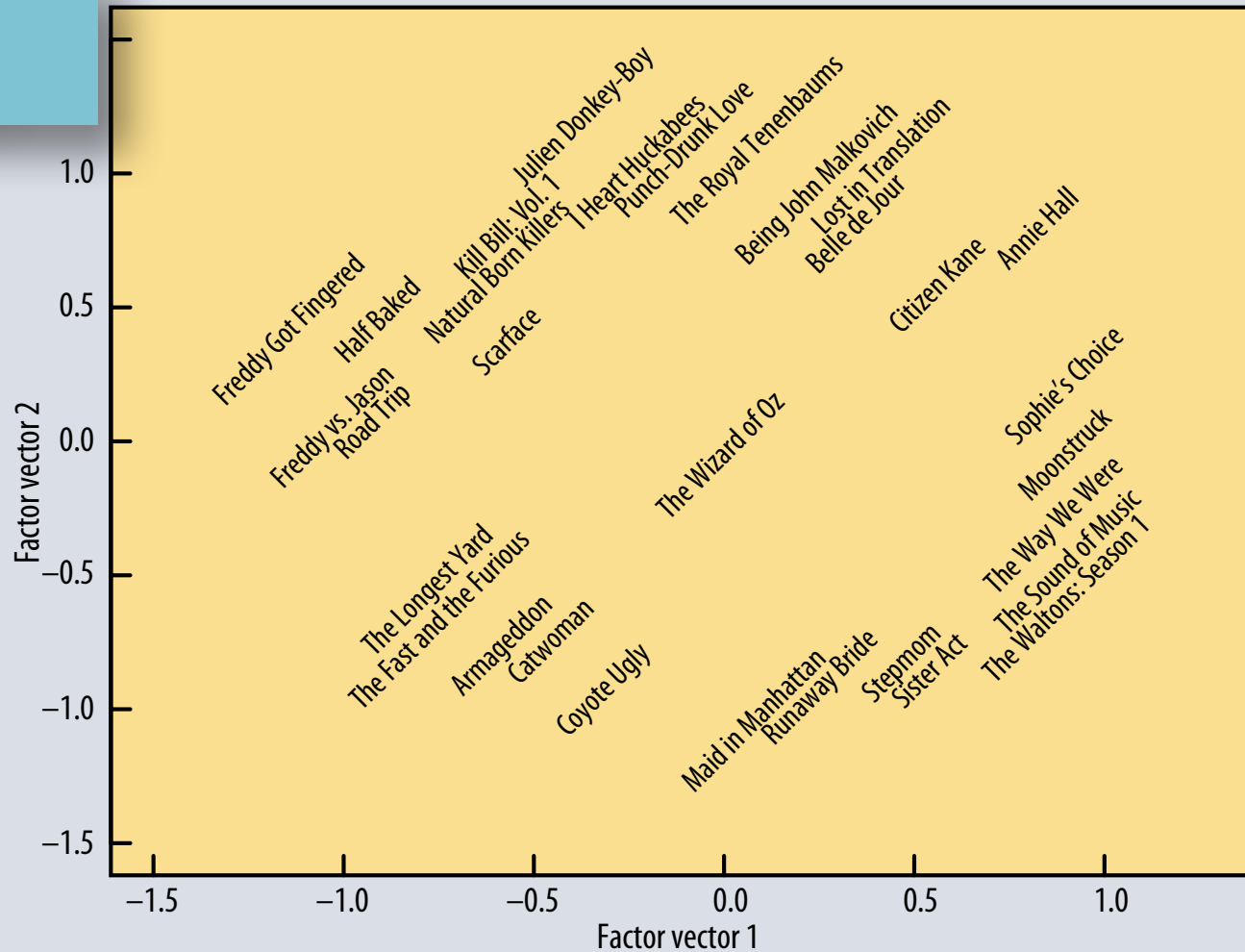


Figure 3. The first two vectors from a matrix decomposition of the Netflix Prize data. Selected movies are placed at the appropriate spot based on their factor vectors in two dimensions. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky independent films.

Matrix Factorization

Comparison of Optimization Algorithms

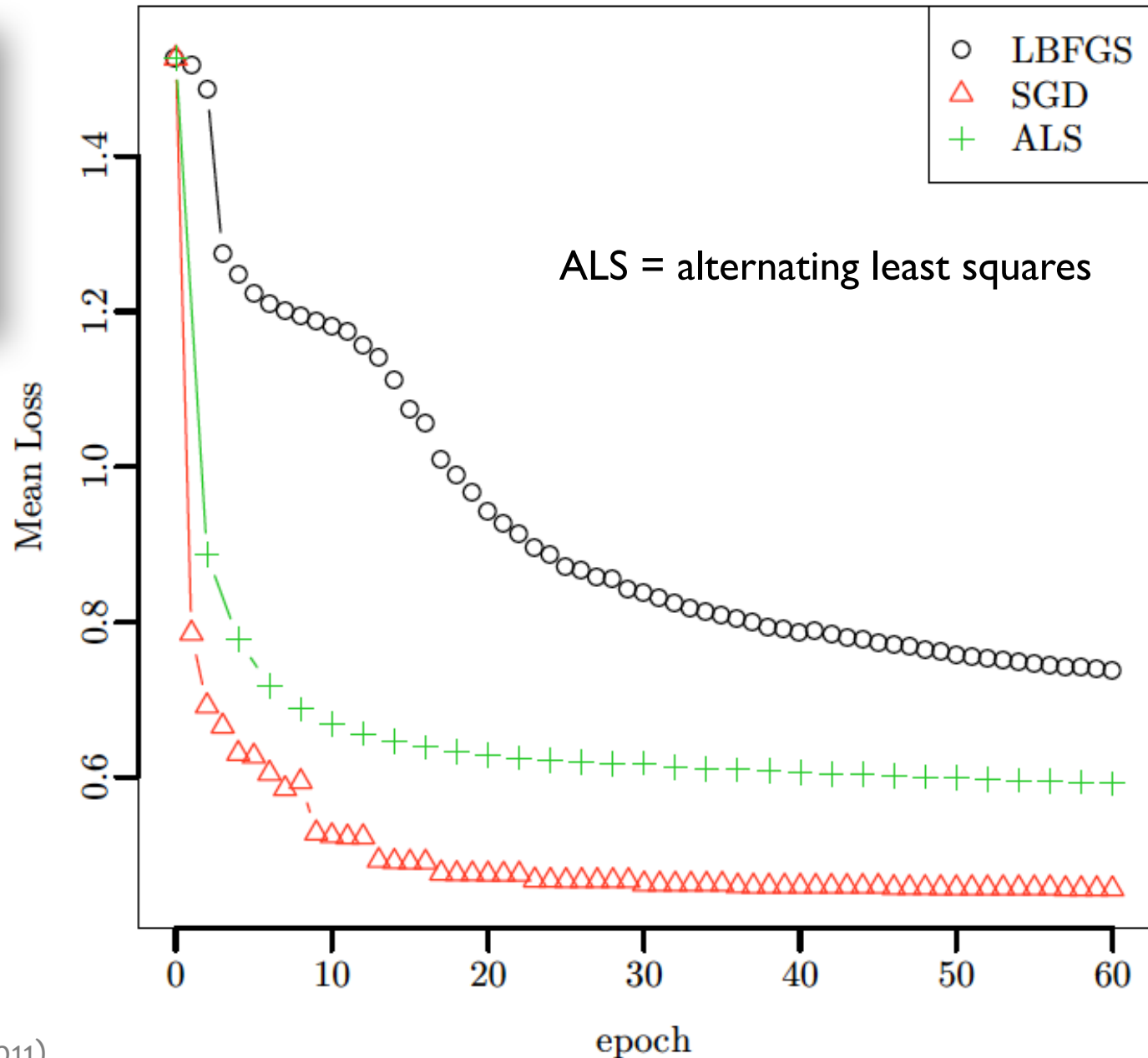


Figure from Gemulla et al. (2011)

SVD FOR COLLABORATIVE FILTERING

Singular Value Decomposition for Collaborative Filtering

Whiteboard

- Optimization problem
- Equivalence to Unconstrained Matrix Factorization (fully specified, no regularization)

NON-NEGATIVE MATRIX FACTORIZATION

Implicit Feedback Datasets

- What information does a five-star rating contain?



- Implicit Feedback Datasets:
 - In many settings, users don't have a way of expressing *dislike* for an item (e.g. can't provide negative ratings)
 - The only mechanism for feedback is to “like” something
- Examples:
 - Facebook has a “Like” button, but no “Dislike” button
 - Google's “+1” button
 - Pinterest pins
 - Purchasing an item on Amazon indicates a preference for it, but there are many reasons you might *not* purchase an item (besides dislike)
 - Search engines collect click data but don't have a clear mechanism for observing dislike of a webpage

Non-negative Matrix Factorization

Whiteboard

- Optimization problem
- Multiplicative updates

Summary

- Recommender systems solve many **real-world** (*large-scale) **problems**
- Collaborative filtering by Matrix Factorization (MF) is an **efficient** and **effective** approach
- MF is just another example of a **common recipe**:
 1. define a model
 2. define an objective function
 3. optimize with SGD