

# Linear Algebra for AI/ML

Jiaul Paik

# Topics


- **Recommender Systems**
  - Content Filtering
  - Collaborative Filtering
  - CF: Neighborhood Methods
  - CF: Latent Factor Methods
- **Matrix Factorization**
  - User / item vectors
  - Prediction model
  - Training by SGD


# 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









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  Cart


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



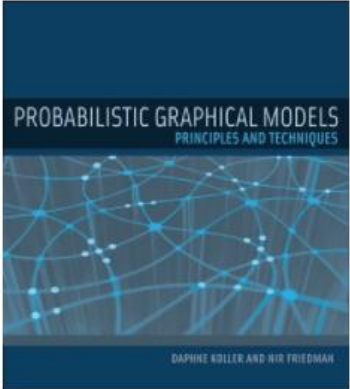
Buy It Again in Grocery  
14 ITEMS



Buy It Again in Pets  
6 ITEMS

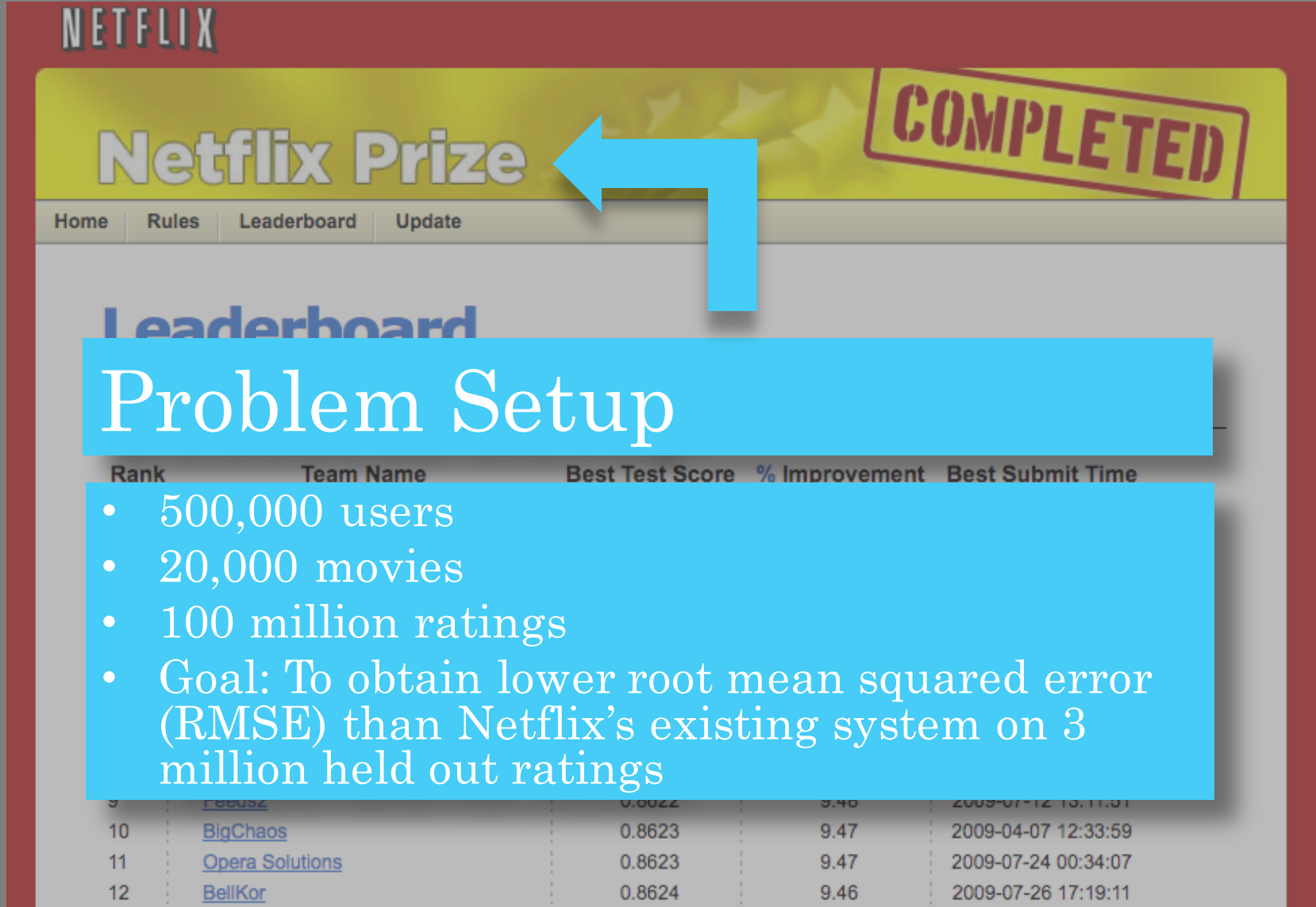


Buy It Again in Baby Products  
5 ITEMS



Engineering Books  
86 ITEMS

# Recommender Systems



The screenshot shows the Netflix Prize Leaderboard page. At the top, the Netflix logo is on the left, and a yellow banner with '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 'Home', 'Rules', 'Leaderboard', and 'Update'. The main heading is 'Leaderboard'. A blue box with white text is overlaid on the page, containing the title 'Problem Setup' and a list of details. Below the box, a table shows the top of the leaderboard with columns for Rank, Team Name, Best Test Score, % Improvement, and Best Submit Time. The table lists teams like Feus2, BigChaos, Opera Solutions, and 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	<a href="#">Feus2</a>	0.8622	9.48	2009-07-12 15:11:51
10	<a href="#">BigChaos</a>	0.8623	9.47	2009-04-07 12:33:59
11	<a href="#">Opera Solutions</a>	0.8623	9.47	2009-07-24 00:34:07
12	<a href="#">BellKor</a>	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

# Two Types of Recommender Systems

- *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
- *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

- **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?”
- ...

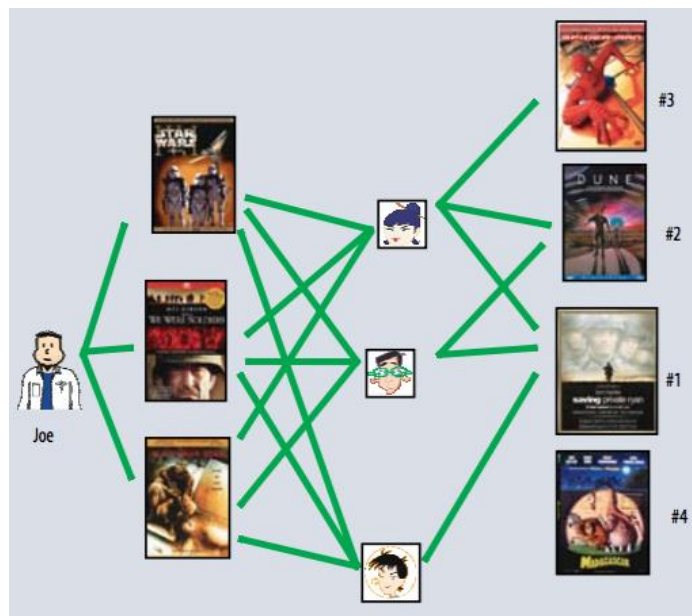
- **Common insight: personal tastes are correlated**

- 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

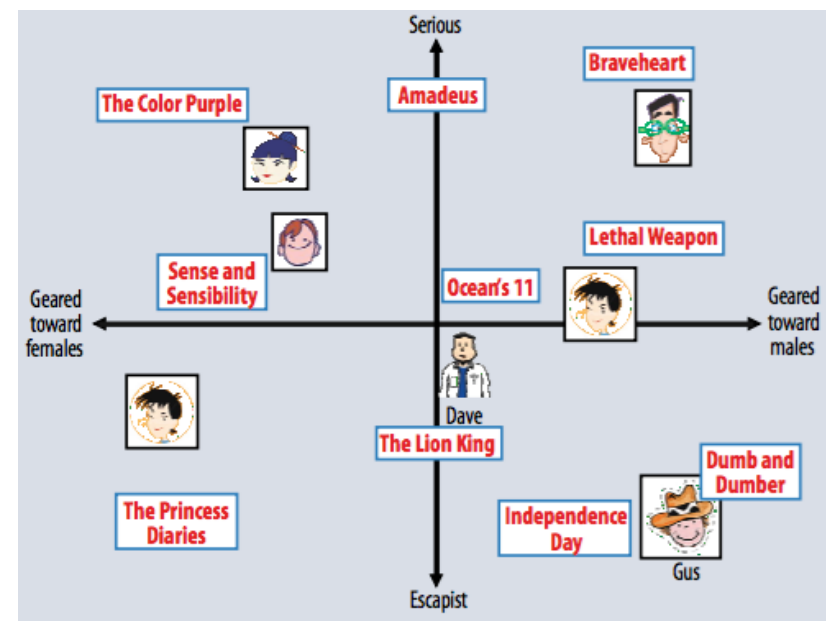


# Two Types of Collaborative Filtering

## 1. Neighborhood Methods

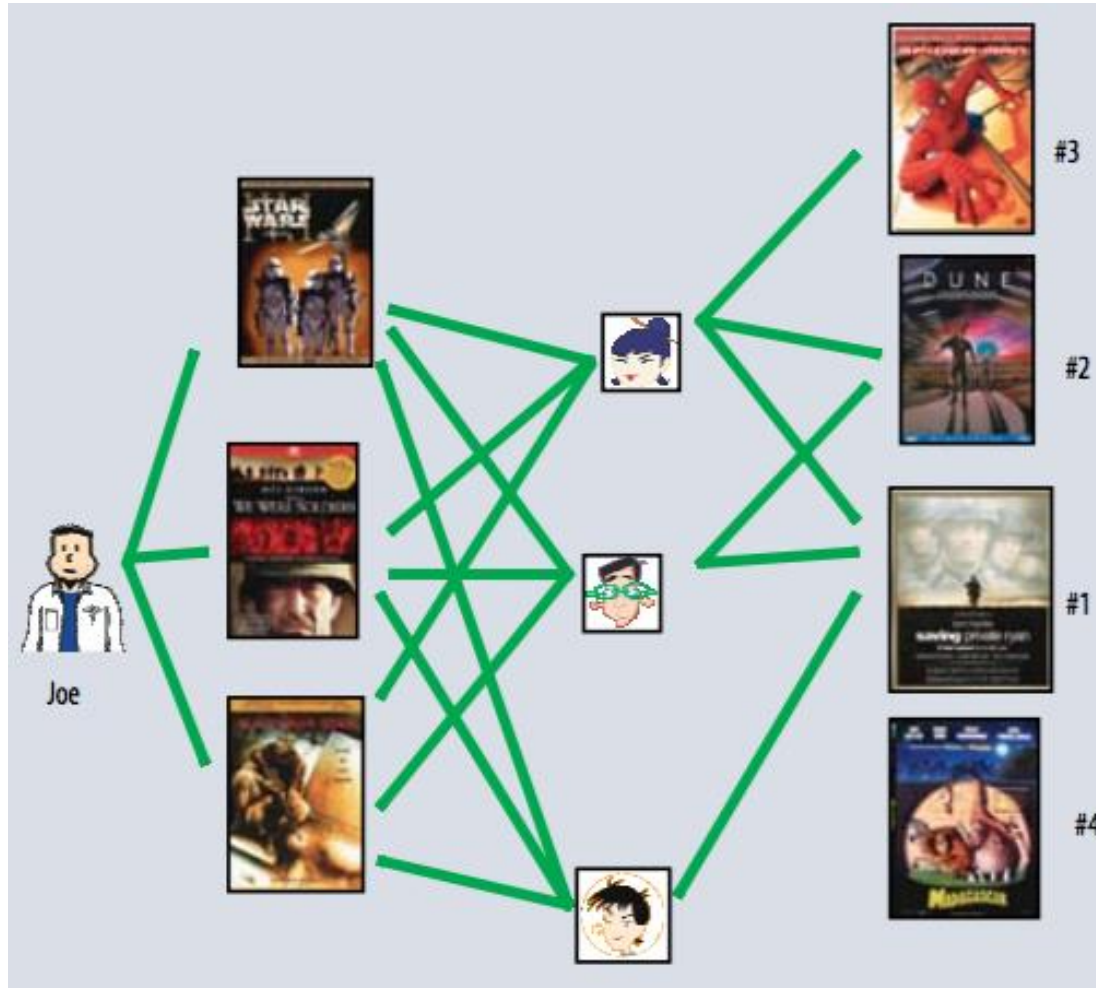


## 2. Latent Factor Methods



# Two Types of Collaborative Filtering

## 1. Neighborhood Methods



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

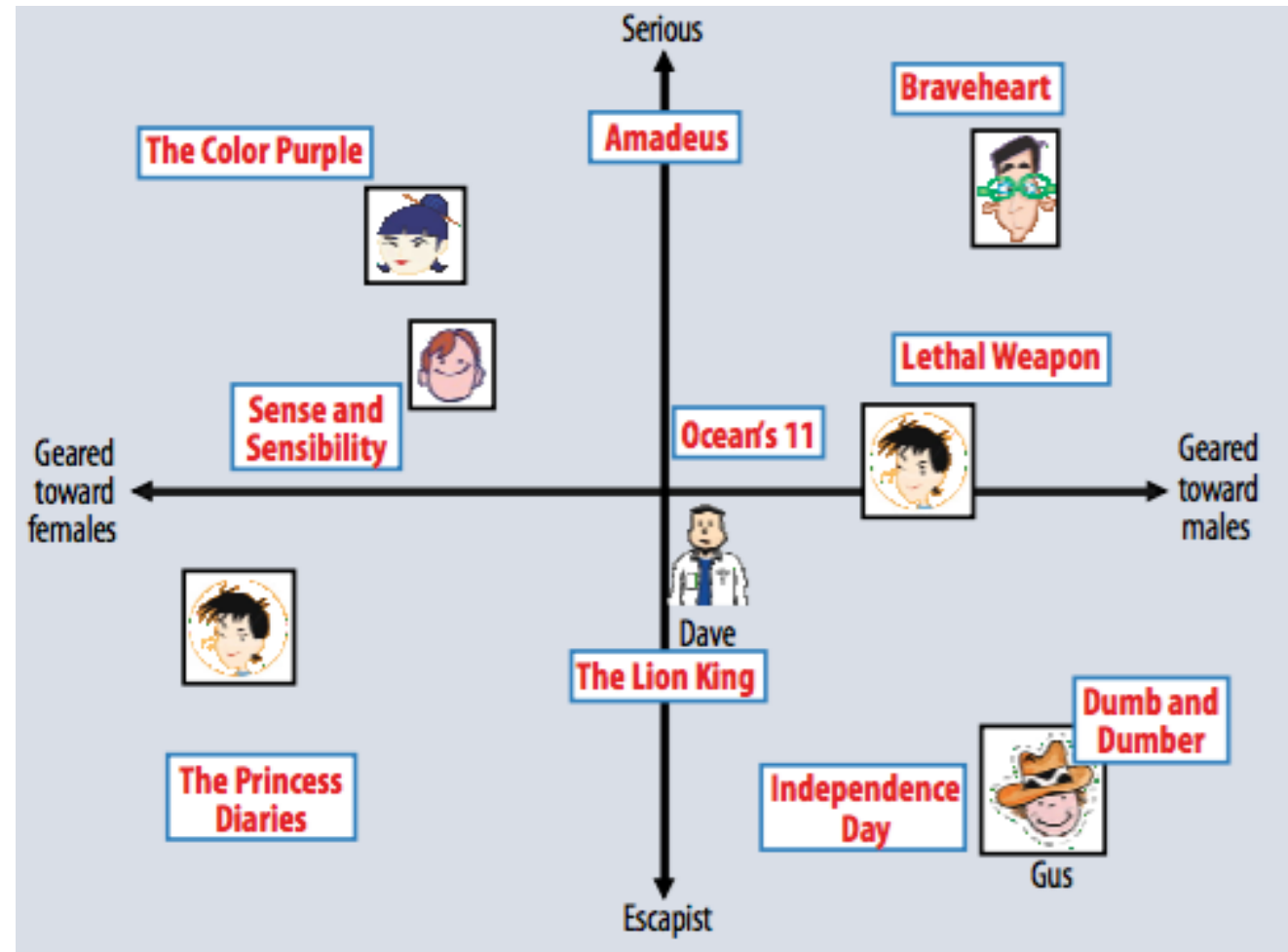
### Algorithm:

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

(with matrices)

- User vectors:

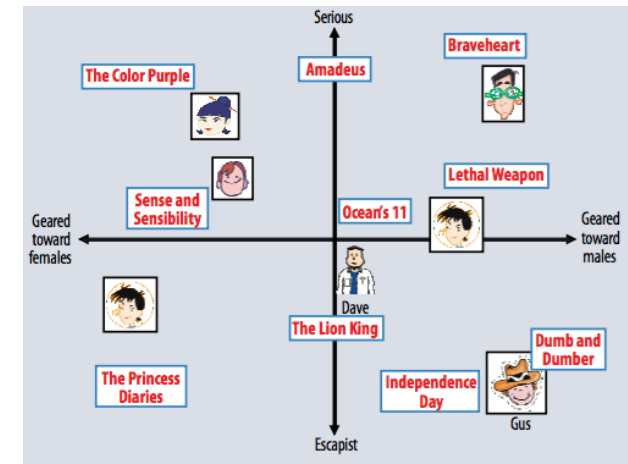
$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

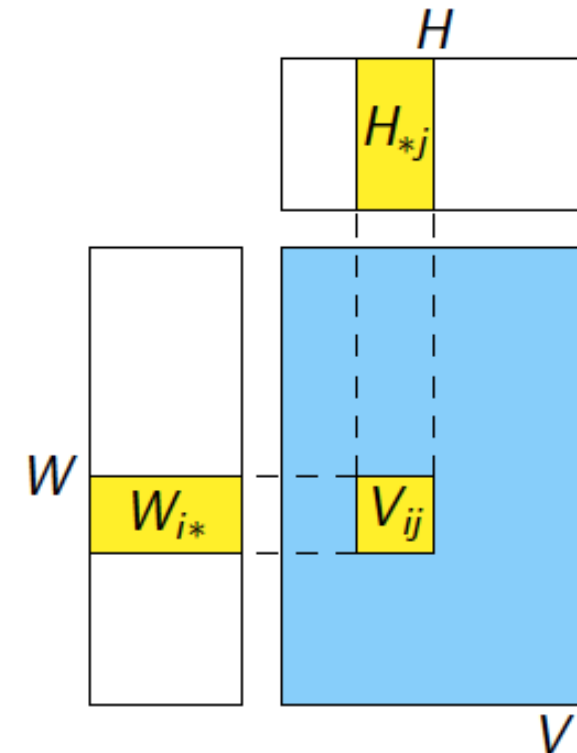
$$H_{*i} \in \mathbb{R}^r$$

- 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)

# Matrix Factorization

(with vectors)

- User vectors:

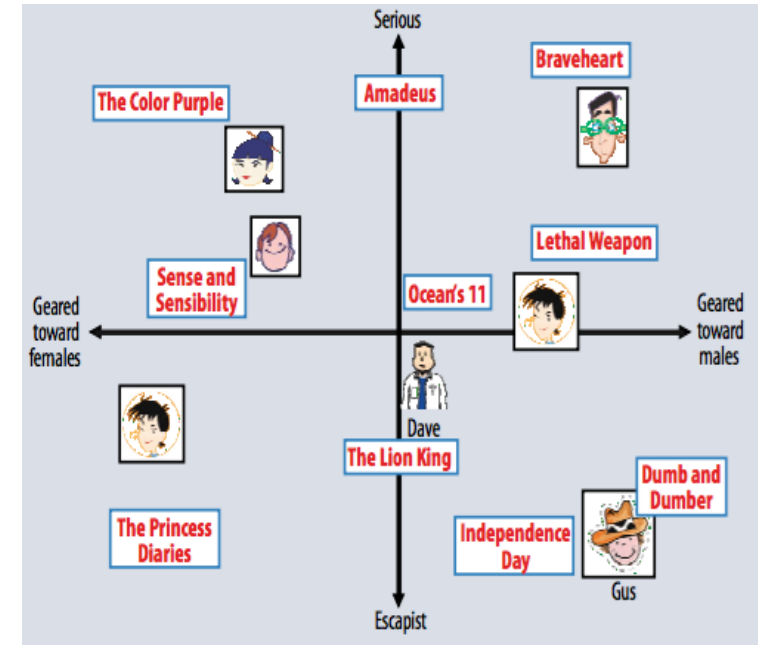
$$\mathbf{w}_u \in \mathbb{R}^r$$

- 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

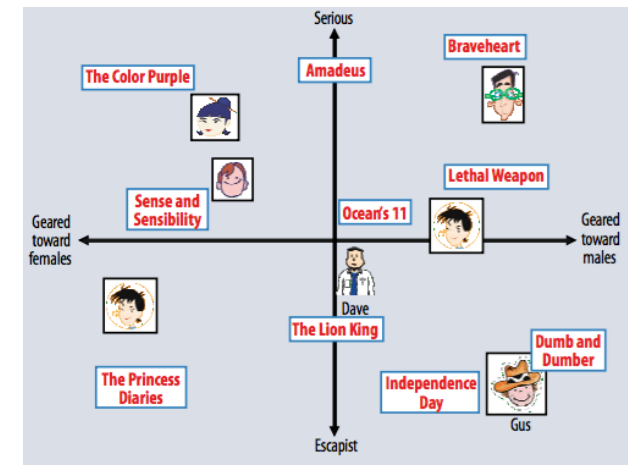
(with vectors)

- Set of non-zero entries:

$$\mathcal{Z} = \{(u, i) : v_{ui} \neq 0\}$$

- Objective:

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



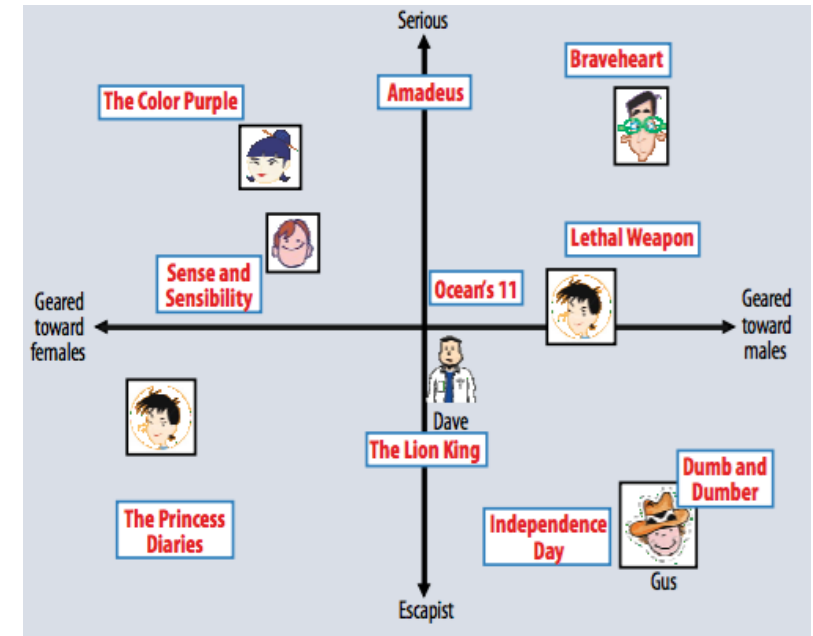
Figures from Koren et al. (2009)

# Matrix Factorization

## (with vectors)

- Regularized Objective:

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



Figures from Koren et al. (2009)



# Matrix Factorization (with vectors)

- Regularized Objective:

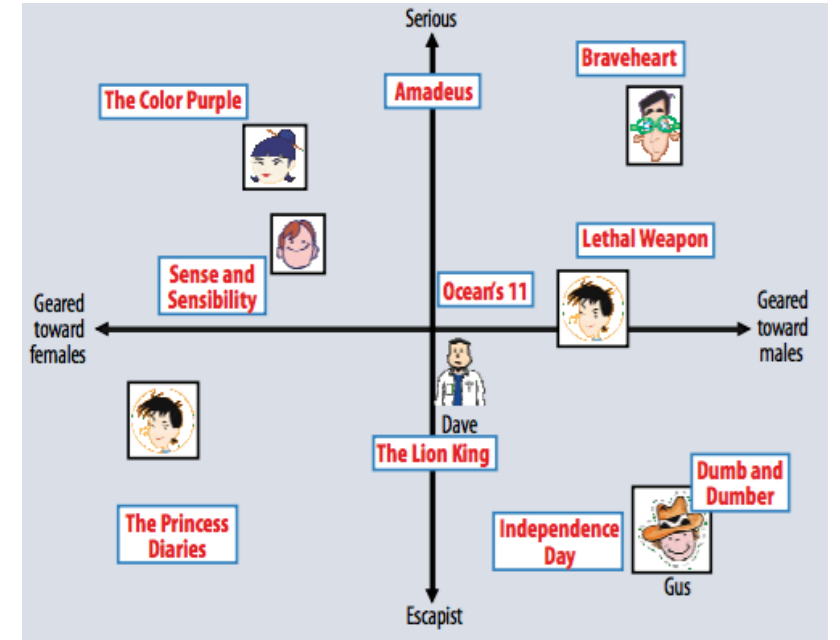
$$\operatorname{argmin}_{\mathbf{w}, \mathbf{h}} \sum_{(u,i) \in \mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 + \lambda \left( \sum_i \|\mathbf{w}_i\|^2 + \sum_u \|\mathbf{h}_u\|^2 \right)$$

- 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:**

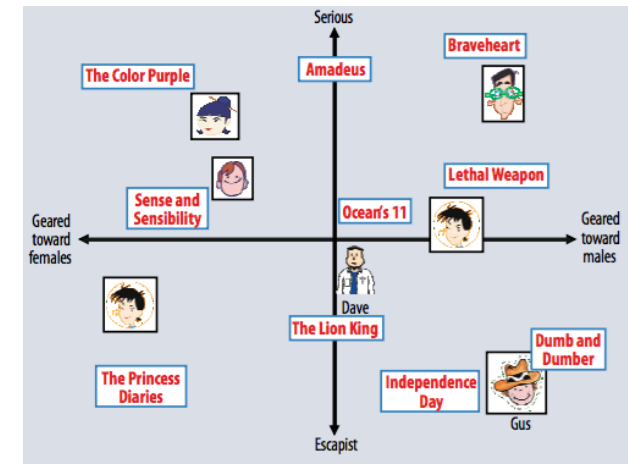
$$(W_{u*})^T \in \mathbb{R}^r$$

- **Item vectors:**

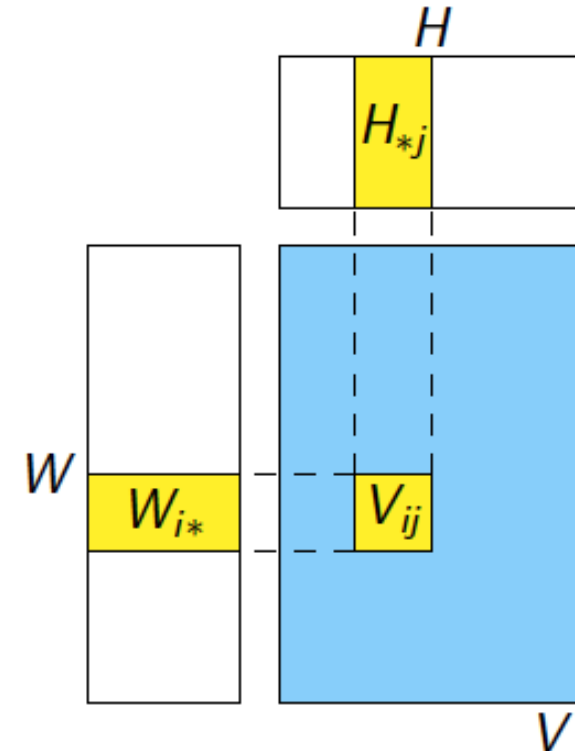
$$H_{*i} \in \mathbb{R}^r$$

- **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)

# Matrix Factorization

- SGD (with matrices)

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.

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

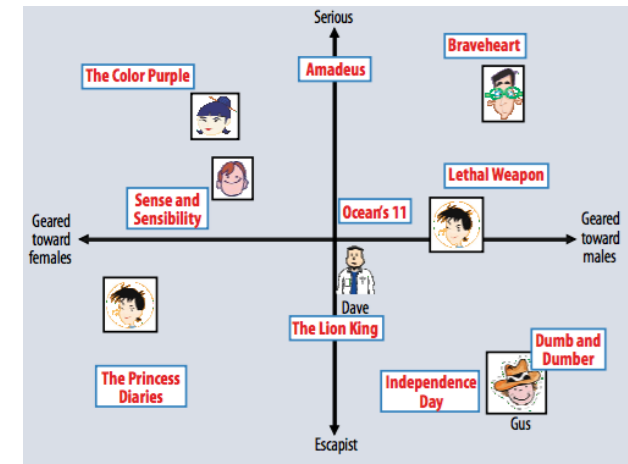
$$\mathbf{H}_{*j} \leftarrow \mathbf{H}_{*j} - \epsilon_n N \frac{\partial}{\partial \mathbf{H}_{*j}} l(\mathbf{V}_{ij}, \mathbf{W}_{i*}, \mathbf{H}_{*j})$$

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

**end while**

step size

Figure from Gemulla et al. (2011)



Figures from Koren et al. (2009)

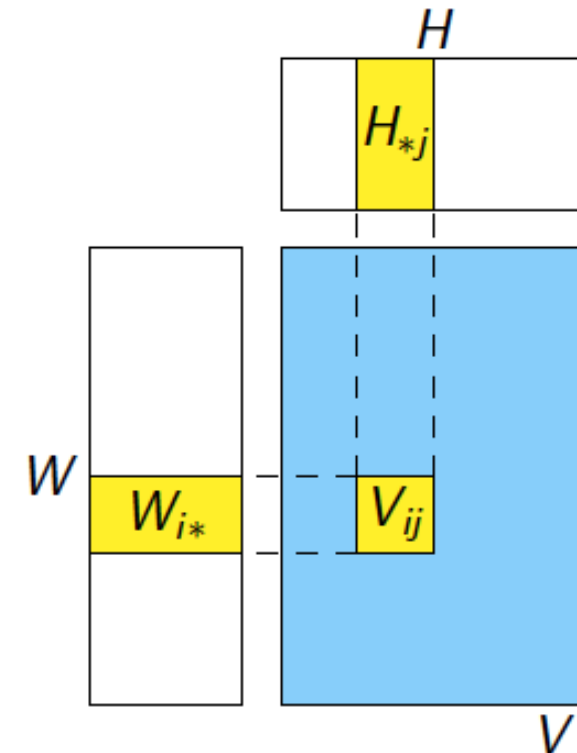


Figure from Gemulla et al.  
(2011)

Thank you!