



10-601B Introduction to Machine Learning

Matrix Factorization and Collaborative Filtering

Readings:

Koren et al. (2009)
Gemulla et al. (2011)

Matt Gormley
Lecture 26
November 30, 2016

Reminders

- Homework 7
 - due Mon., Dec. 5
- In-class Review Session
 - Mon., Dec. 5
- Final Exam
 - in-class Wed., Dec. 7

Outline

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

CYBER MONDAY DEALS WEEK

All [Search](#)

amazon Prime

Departments [Browsing History](#) [Matt's Amazon.com](#) [Cyber Monday](#) [Gift Cards & Registry](#) [Sell](#) [Help](#)

Hello, Matt
Your Account [Prime](#) [Lists](#) [Cart](#) 1

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

PROBABILISTIC GRAPHICAL MODELS
PRINCIPLES AND TECHNIQUES
DAPHNE KOLLER AND NIR FRIEDMAN

Recommender Systems



Recommender Systems

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.

Recommender Systems



The image shows a screenshot of the Netflix Prize website. At the top, there's a yellow banner with the text "Netflix Prize" and a large red "COMPLETED" stamp. Below the banner is a navigation bar with links for "Home", "Rules", "Leaderboard", and "Update". A blue arrow points from the "Leaderboard" link down to the "Problem Setup" section of the page. The main content area has a light blue background and displays the title "Leaderboard" above a large, bold "Problem Setup" heading. Below this, there's a table with columns for Rank, Team Name, Best Test Score, % Improvement, and Best Submit Time. The table lists four teams: FOCOSZ, BigChaos, Opera Solutions, and BellKor, all of which have a Best Test Score of 0.8622 and a % Improvement of 9.46. The Best Submit Time for FOCOSZ is 2009-07-12 13:11:51, for BigChaos it's 2009-04-07 12:33:59, for Opera Solutions it's 2009-07-24 00:34:07, and for BellKor it's 2009-07-26 17:19:11.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
9	FOCOSZ	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

The screenshot shows the Netflix Prize Leaderboard page. At the top, the Netflix logo is visible, followed by a large yellow banner with the text "Netflix Prize" and a large red stamp that says "COMPLETED". Below the banner, there is a navigation bar with links for "Home", "Rules", "Leaderboard", and "Update". The main title "Leaderboard" is displayed in a large blue font. Below it, a sub-instruction "Showing Test Score. [Click here to show quiz score](#)" is present. A horizontal line separates this from the table below. The table has columns for "Rank", "Team Name", "Best Test Score", "% Improvement", and "Best Submit Time". A blue header row highlights the "Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos". The data rows list 12 teams, starting with BellKor's Pragmatic Chaos at rank 1.

Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
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

The screenshot shows the Netflix Prize Leaderboard page. At the top, the Netflix logo is visible, followed by the text "Netflix Prize" and a large red stamp that says "COMPLETED". Below this, there is a navigation bar with links for "Home", "Rules", "Leaderboard", and "Update". The main section is titled "Leaderboard" in large blue letters. Below it, a sub-section says "Showing Test Score. [Click here to show quiz score](#)". The table below lists the top 12 teams along with their test scores, improvement percentages, and submission times.

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

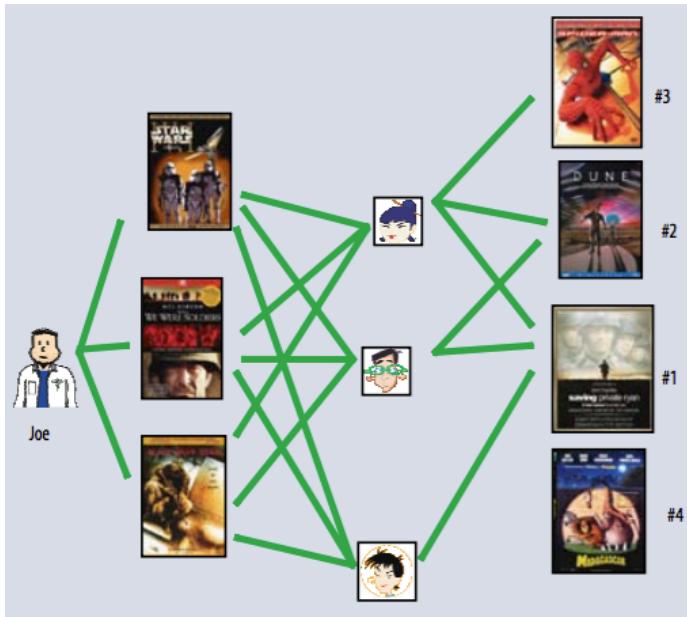
Content Filtering

Collaborative Filtering

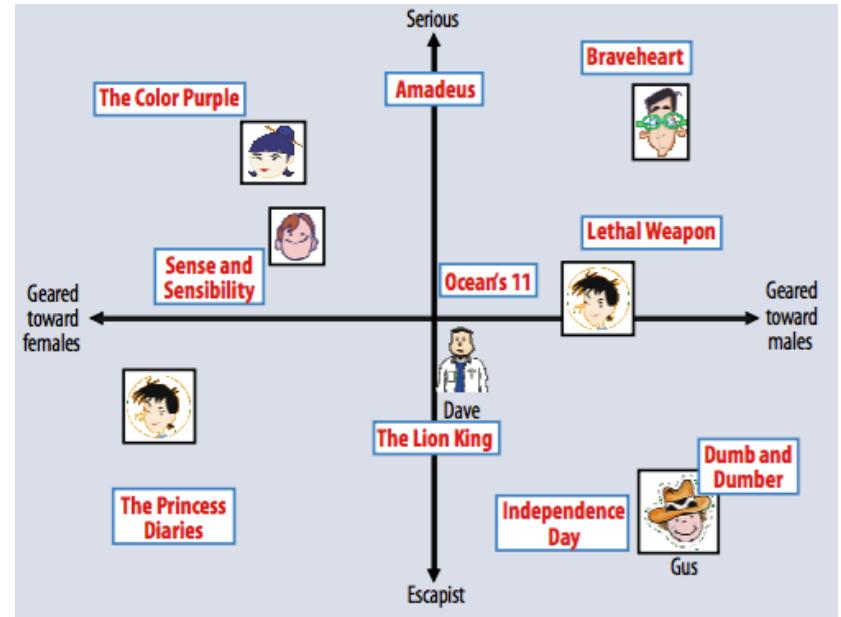
- **Everyday Examples of Collaborative Filtering...**
 - Bestseller lists
 - Top 40 music lists
 - The “recent returns” shelf at the library
 - Unmarked but well-used paths thru the woods
 - The printer room at work
 - “Read any good books lately?”
 - ...
- **Common insight:** personal tastes are correlated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

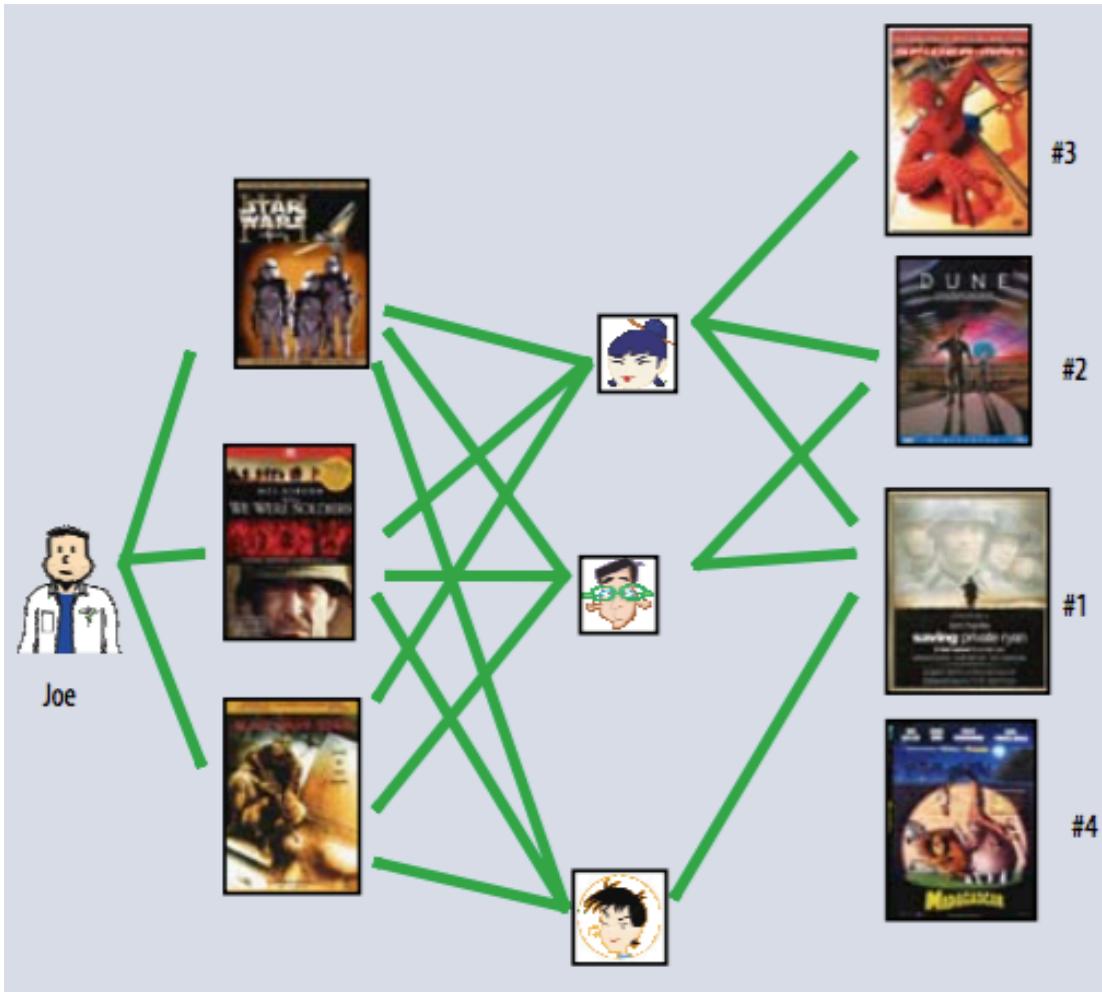


2. Latent Factor Methods



Two Types of Collaborative Filtering

1. Neighborhood Methods



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

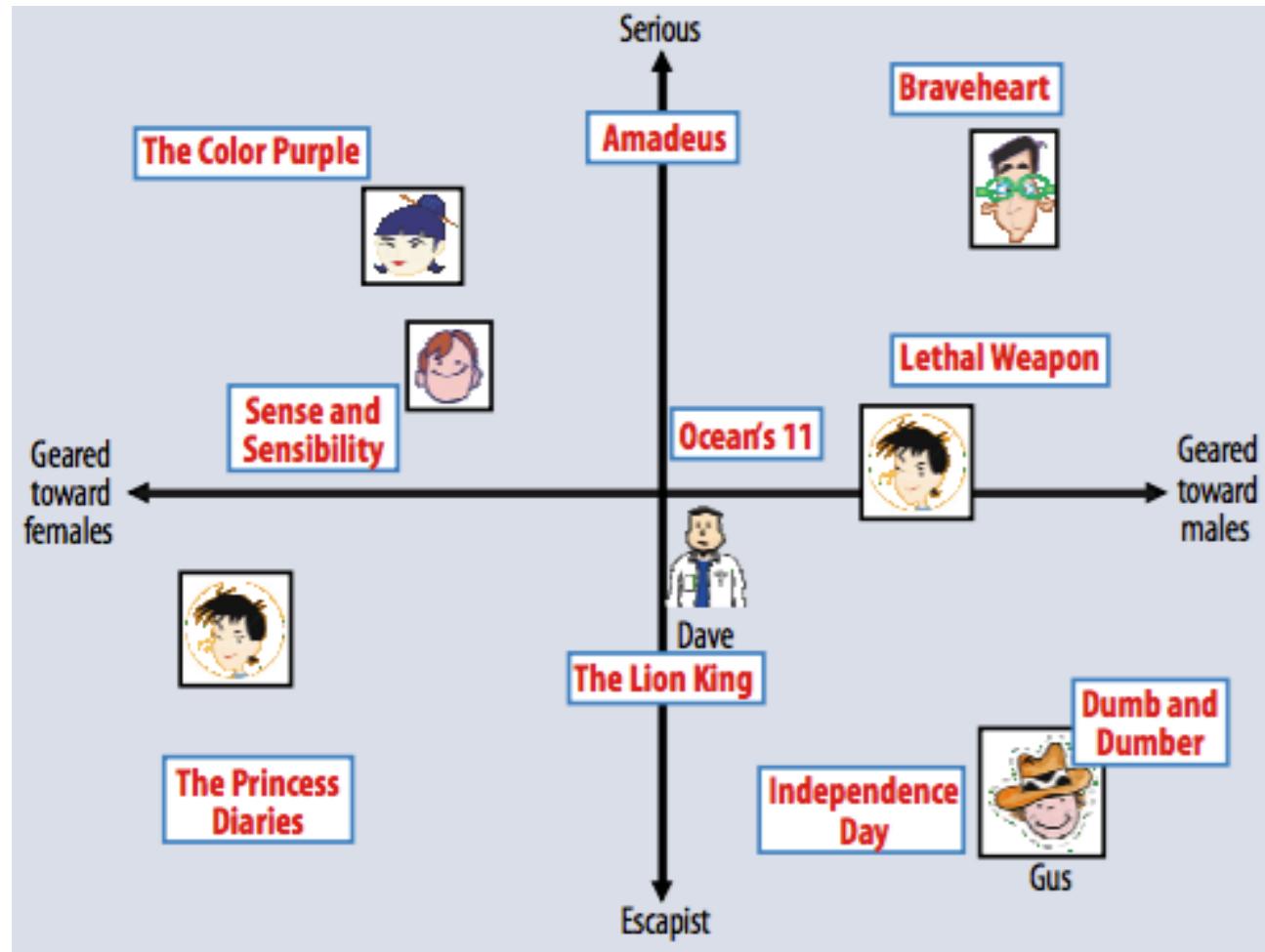
Algorithm:

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 (with matrices)

- User vectors:

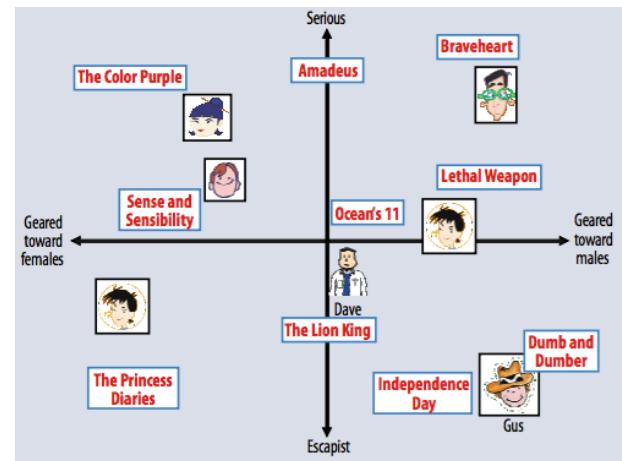
$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

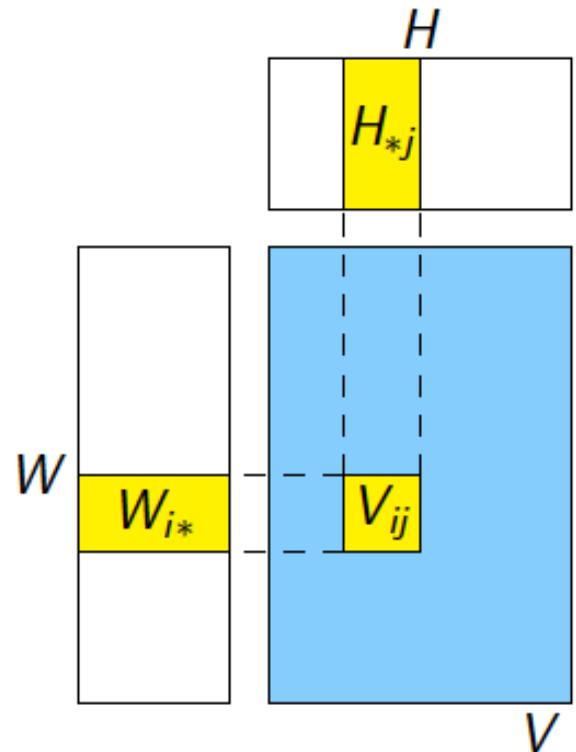
$$H_{*i} \in \mathbb{R}^r$$

- 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)₂₀

Matrix Factorization (with vectors)

- User vectors:

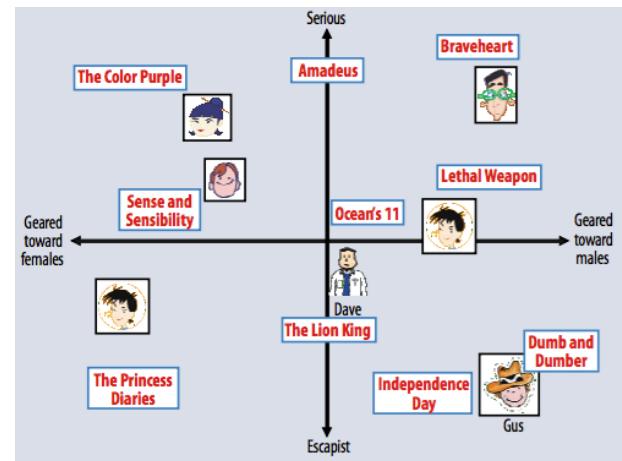
$$\mathbf{w}_u \in \mathbb{R}^r$$

- 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)

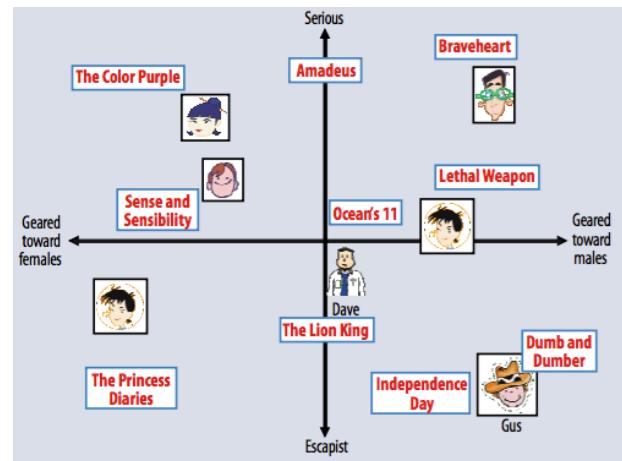
Matrix Factorization (with vectors)

- Set of non-zero entries:

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

- Objective:

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

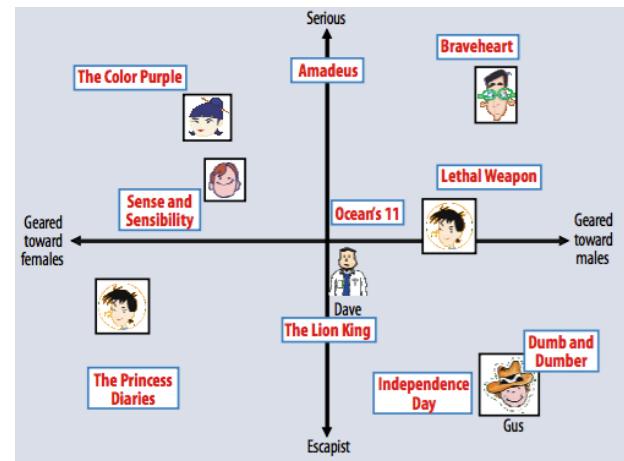


Figures from Koren et al. (2009)

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:

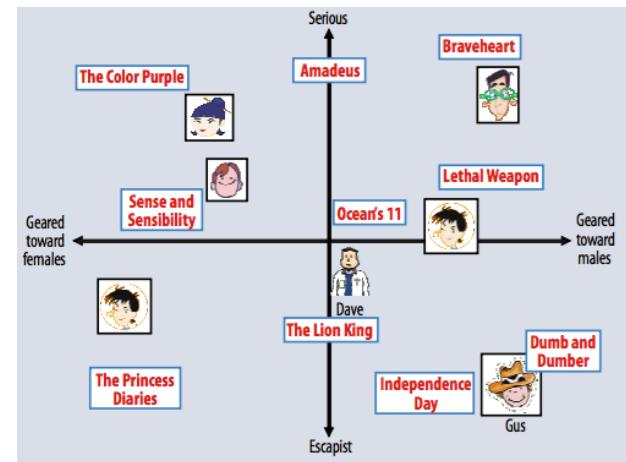
$$\operatorname{argmin}_{\mathbf{w}, \mathbf{h}} \sum_{(u,i) \in \mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 + \lambda (\sum_i \|\mathbf{w}_i\|^2 + \sum_u \|\mathbf{h}_u\|^2)$$

- 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:

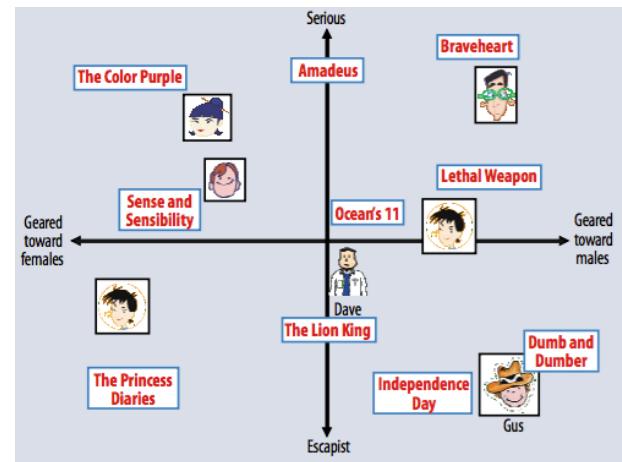
$$(W_{u*})^T \in \mathbb{R}^r$$

- Item vectors:

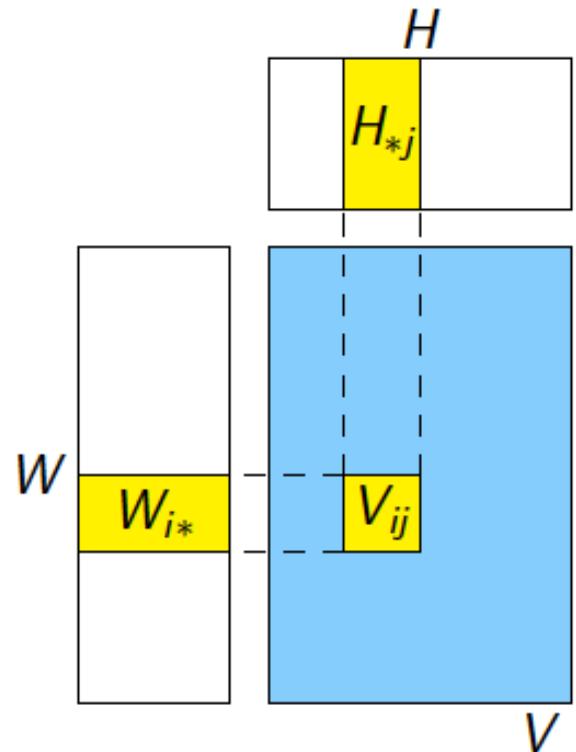
$$H_{*i} \in \mathbb{R}^r$$

- 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)₂₅

Matrix Factorization (with matrices)

- SGD

require that the loss can be written as

$$L = \sum_{(i,j) \in Z} l(V_{ij}, W_{i*}, H_{*j})$$

Algorithm 1 SGD for Matrix Factorization

Require: A training set Z , initial values W_0 and H_0
while not converged **do** {step}

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

$$W'_{i*} \leftarrow W_{i*} - \epsilon_n N \frac{\partial}{\partial W_{i*}} l(V_{ij}, W_{i*}, H_{*j})$$

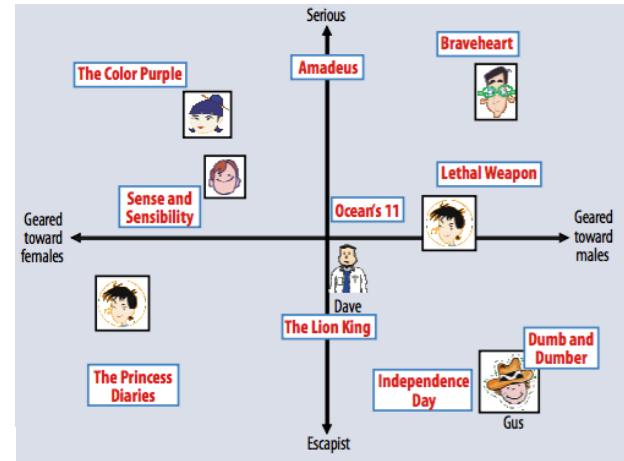
$$H_{*j} \leftarrow H_{*j} - \epsilon_n N \frac{\partial}{\partial H_{*j}} l(V_{ij}, W_{i*}, H_{*j})$$

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

end while

step size

Figure from Gemulla et al. (2011)



Figures from Koren et al. (2009)

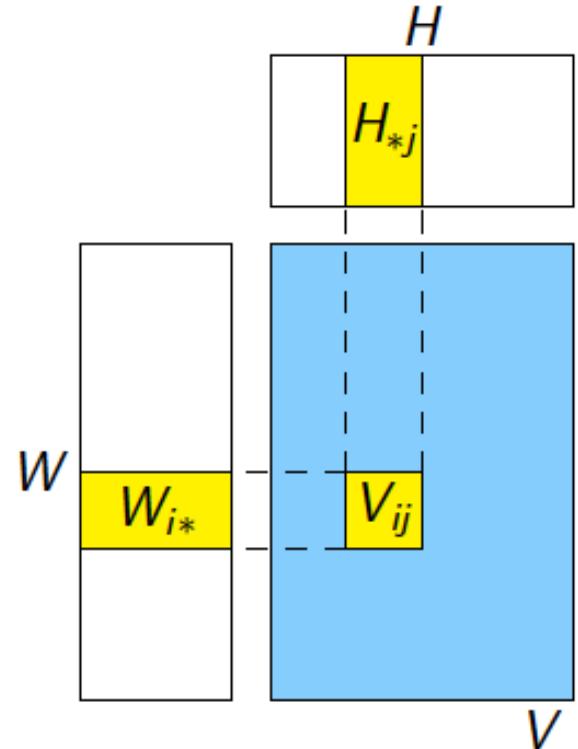


Figure from Gemulla et al. (2011)₂₆

Matrix Factorization

Example Factors

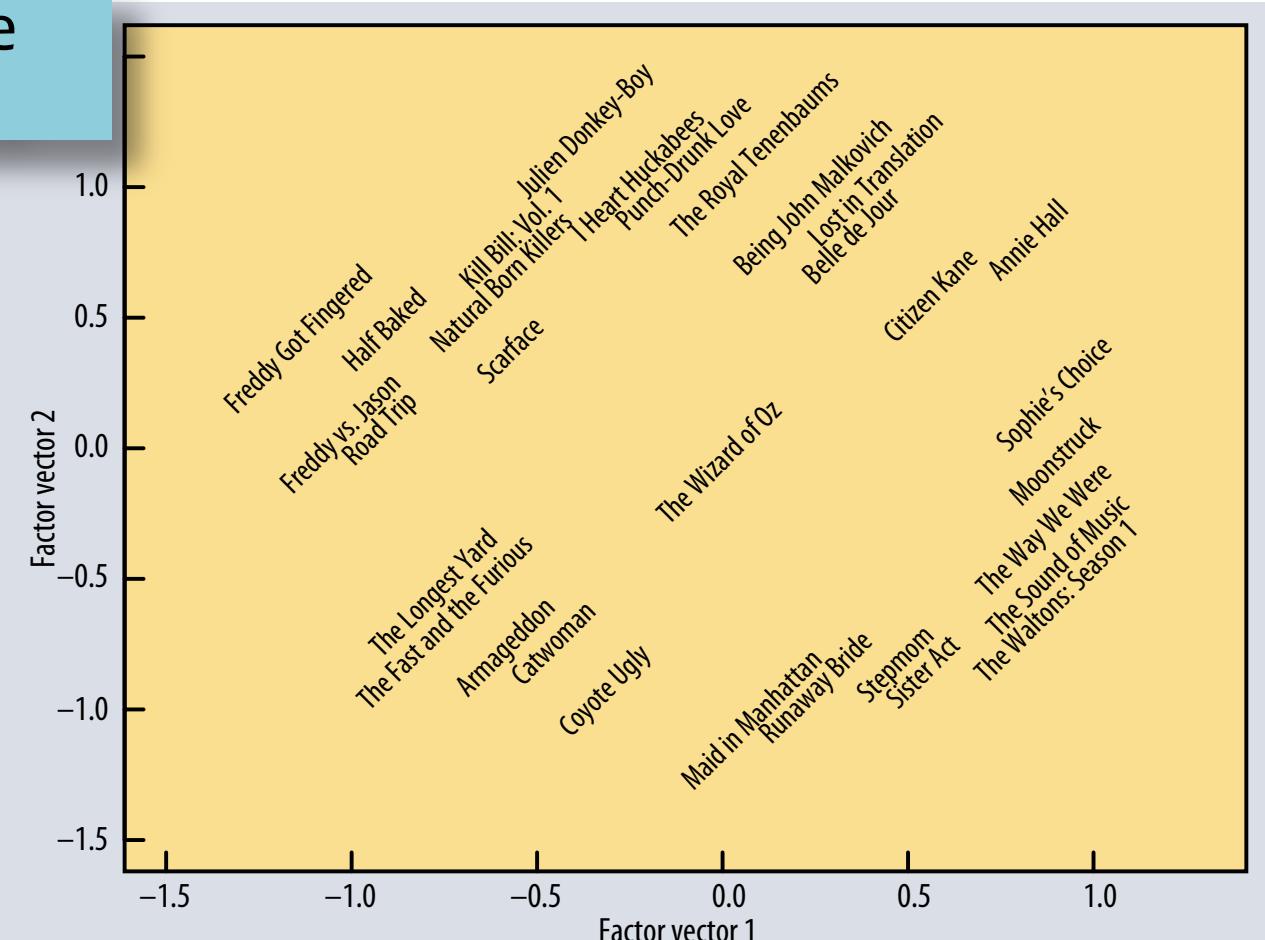
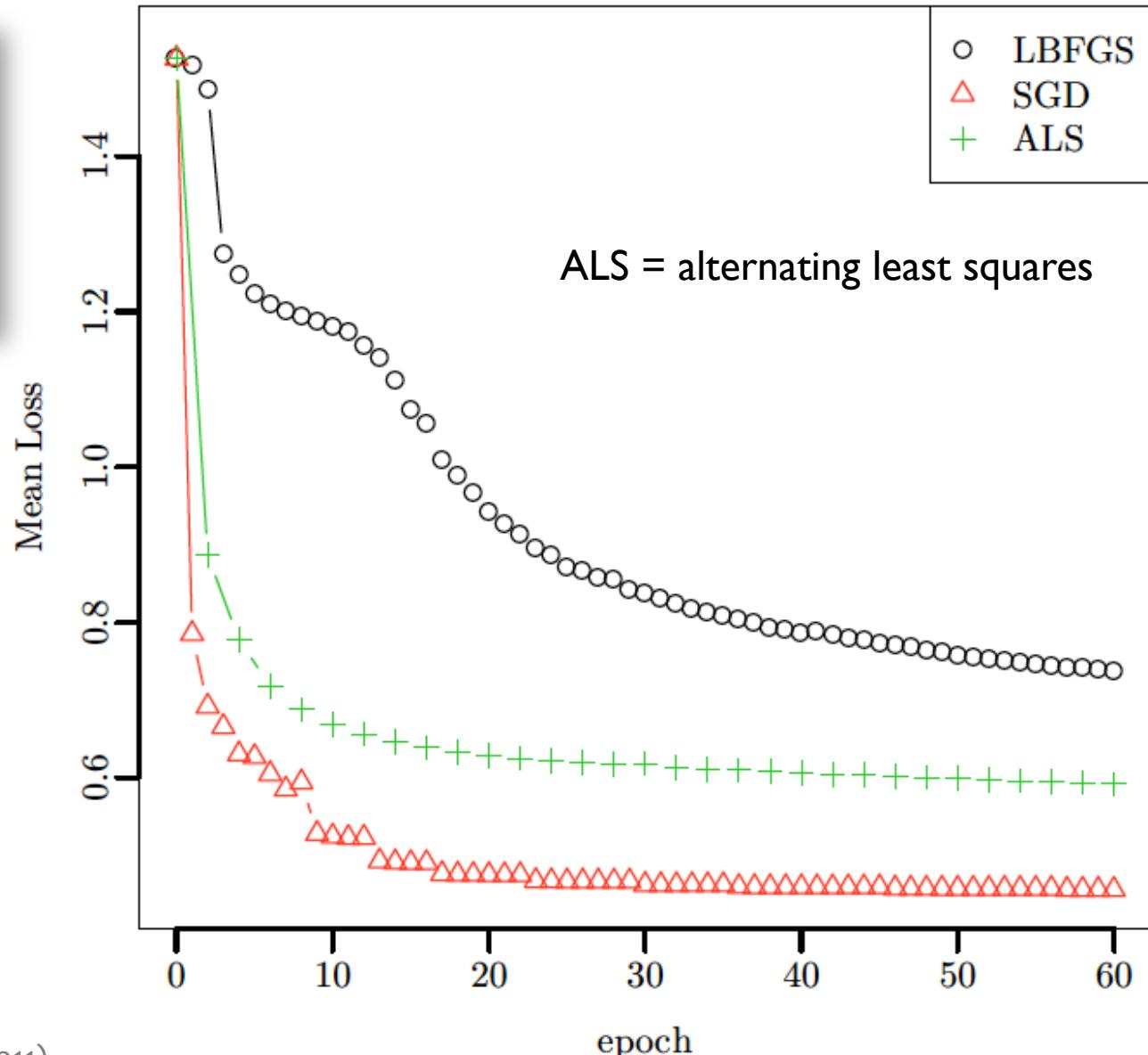


Figure 3. The first two vectors from a matrix decomposition of the Netflix Prize data. Selected movies are placed at the appropriate spot based on their factor vectors in two dimensions. The plot reveals distinct genres, including clusters of movies with strong female leads, fraternity humor, and quirky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

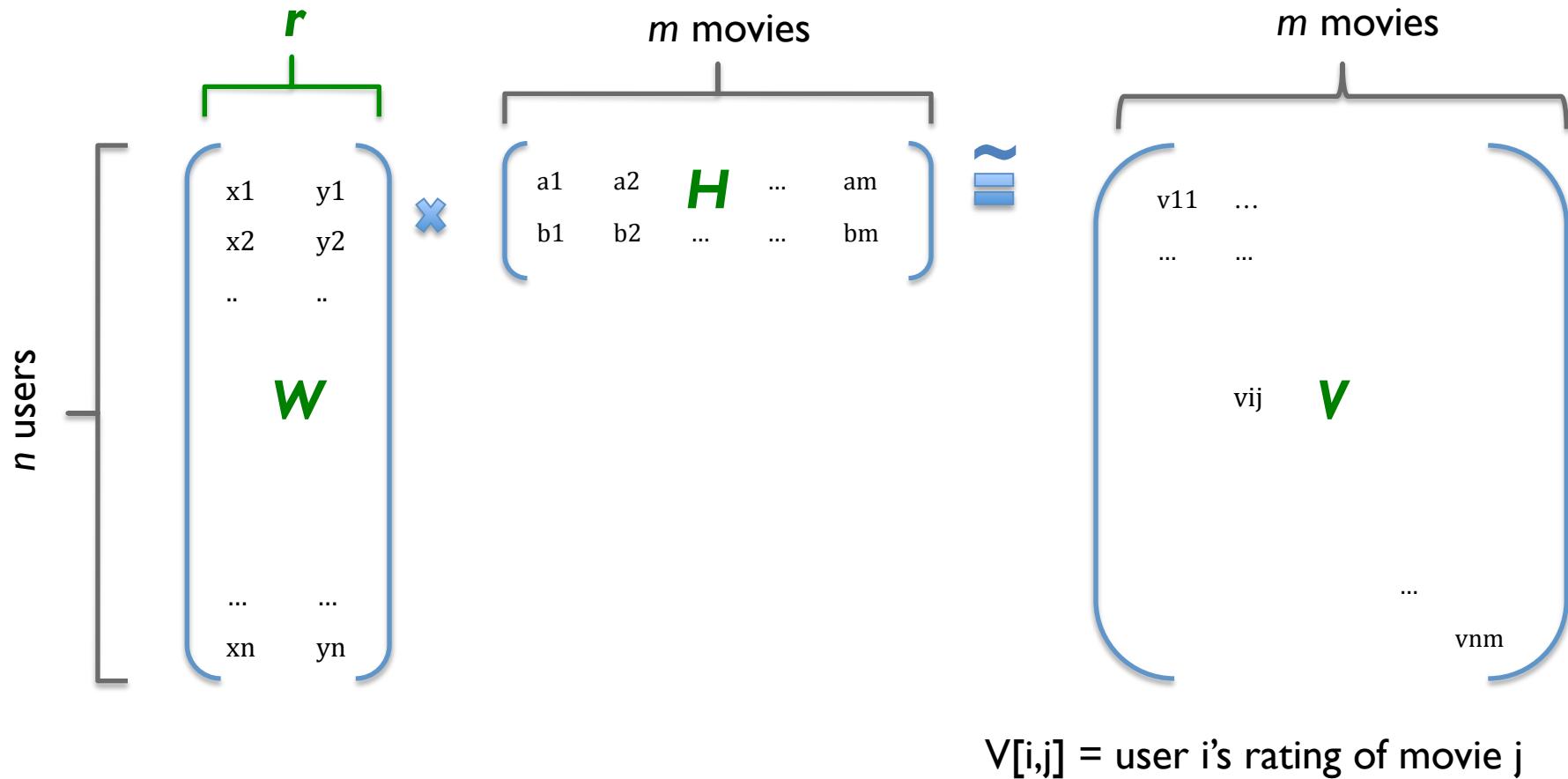
Comparison of Optimization Algorithms

Matrix Factorization

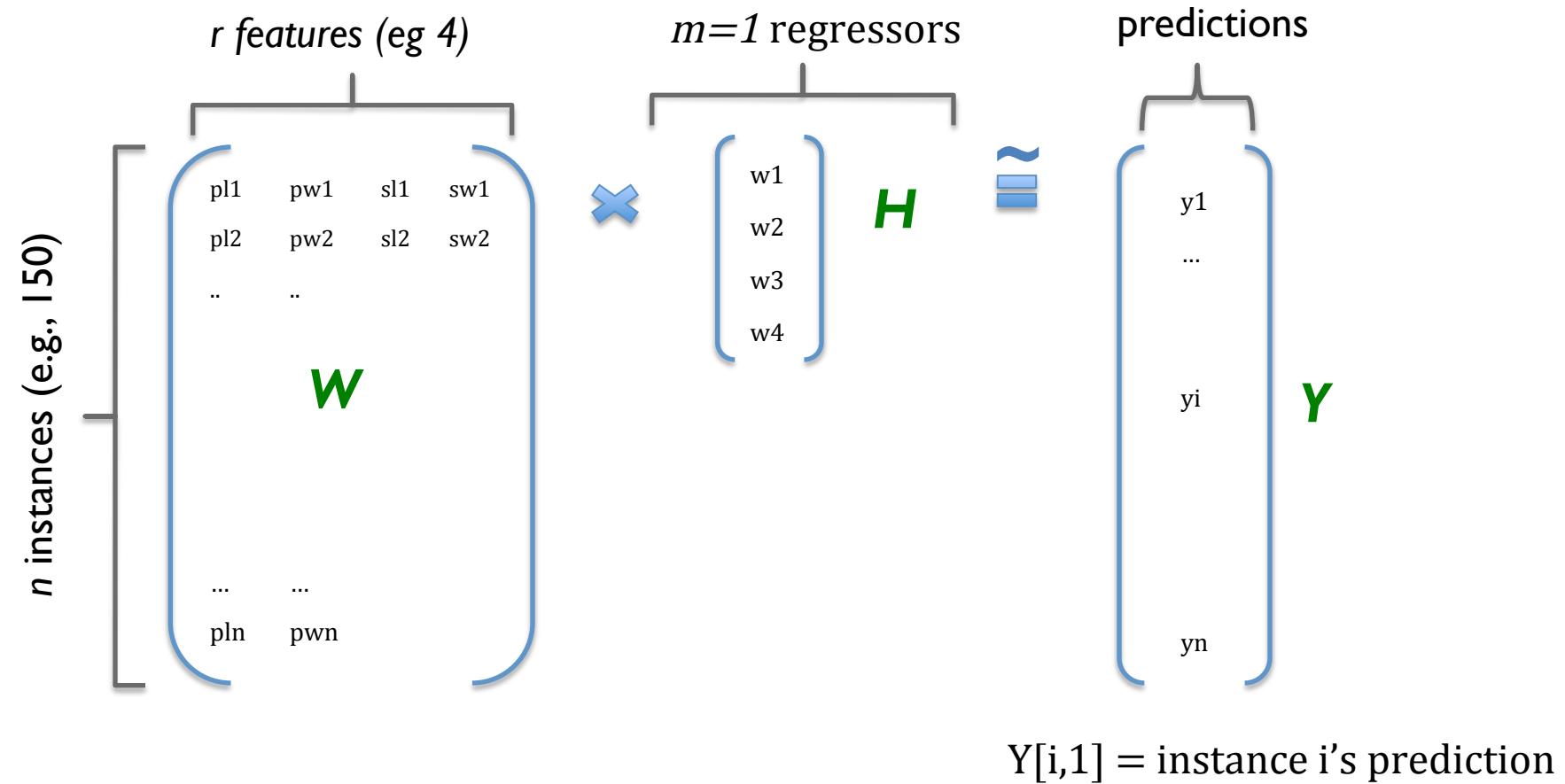


MATRIX MULTIPLICATION IN MACHINE LEARNING

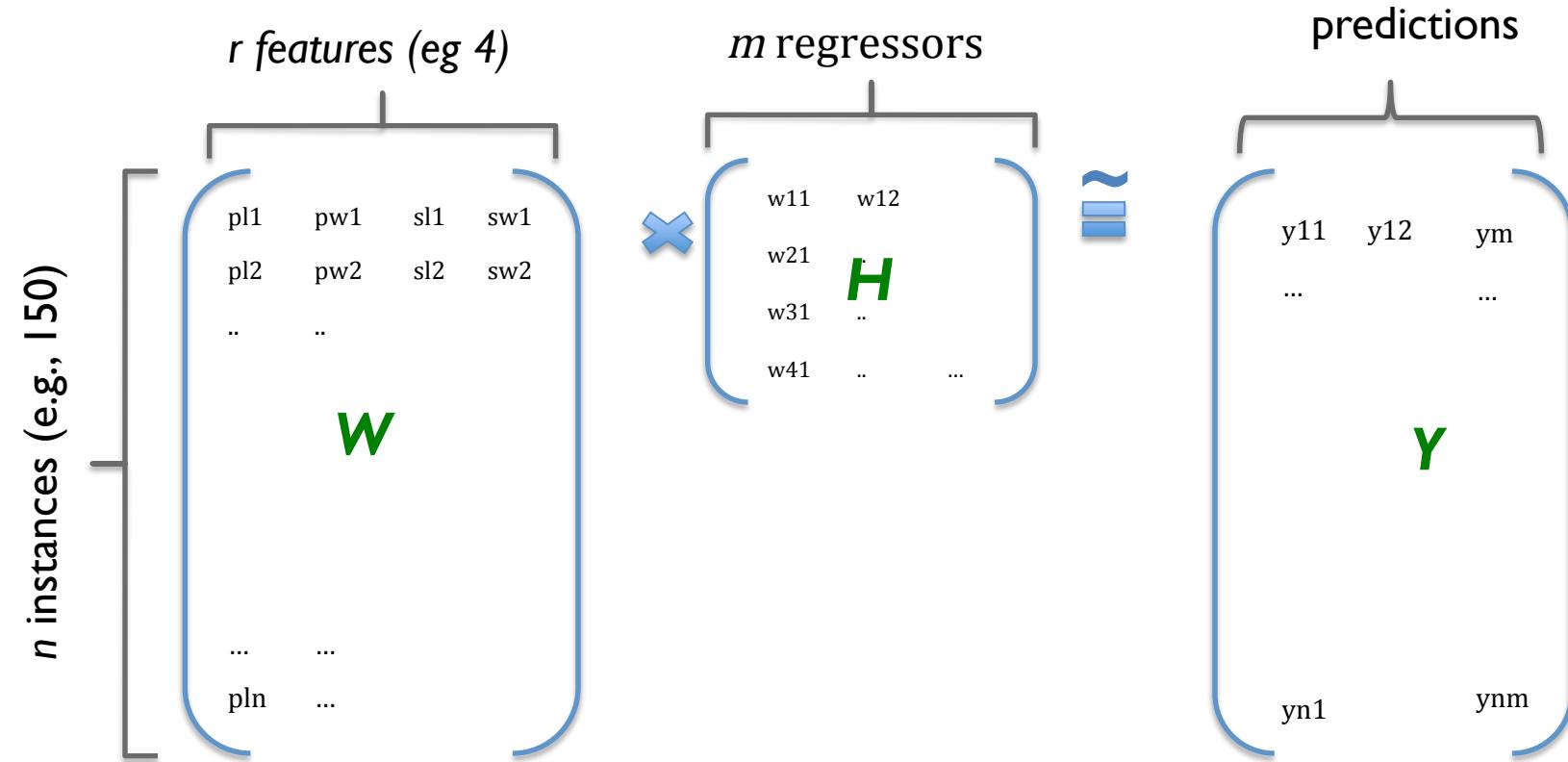
Recovering latent factors in a matrix



... is like Linear Regression



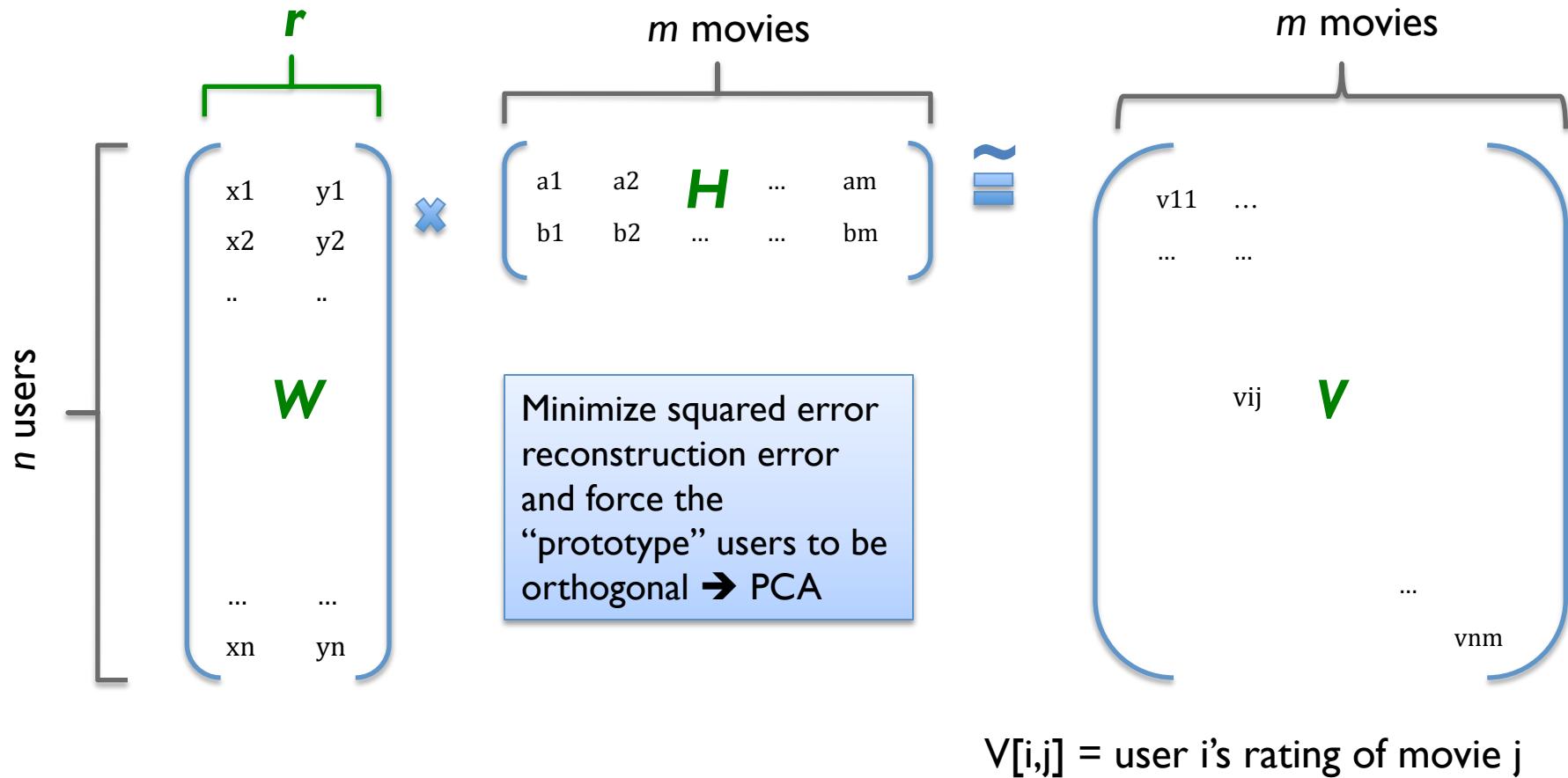
.. for many outputs at once....



... where we also have to find the dataset!

$Y[i,j]$ = instance i's prediction for regression task j

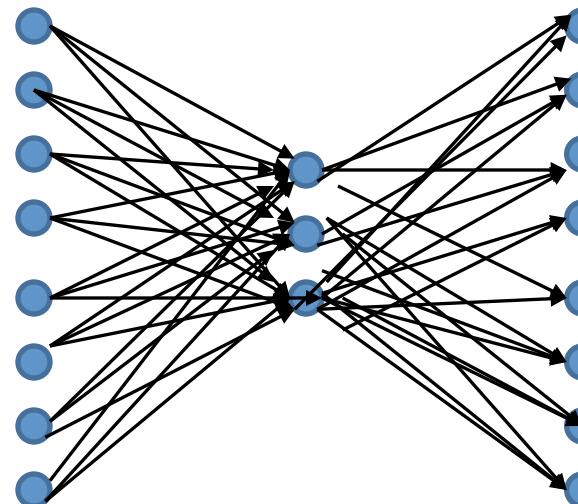
... vs PCA



... vs autoencoders & nonlinear PCA

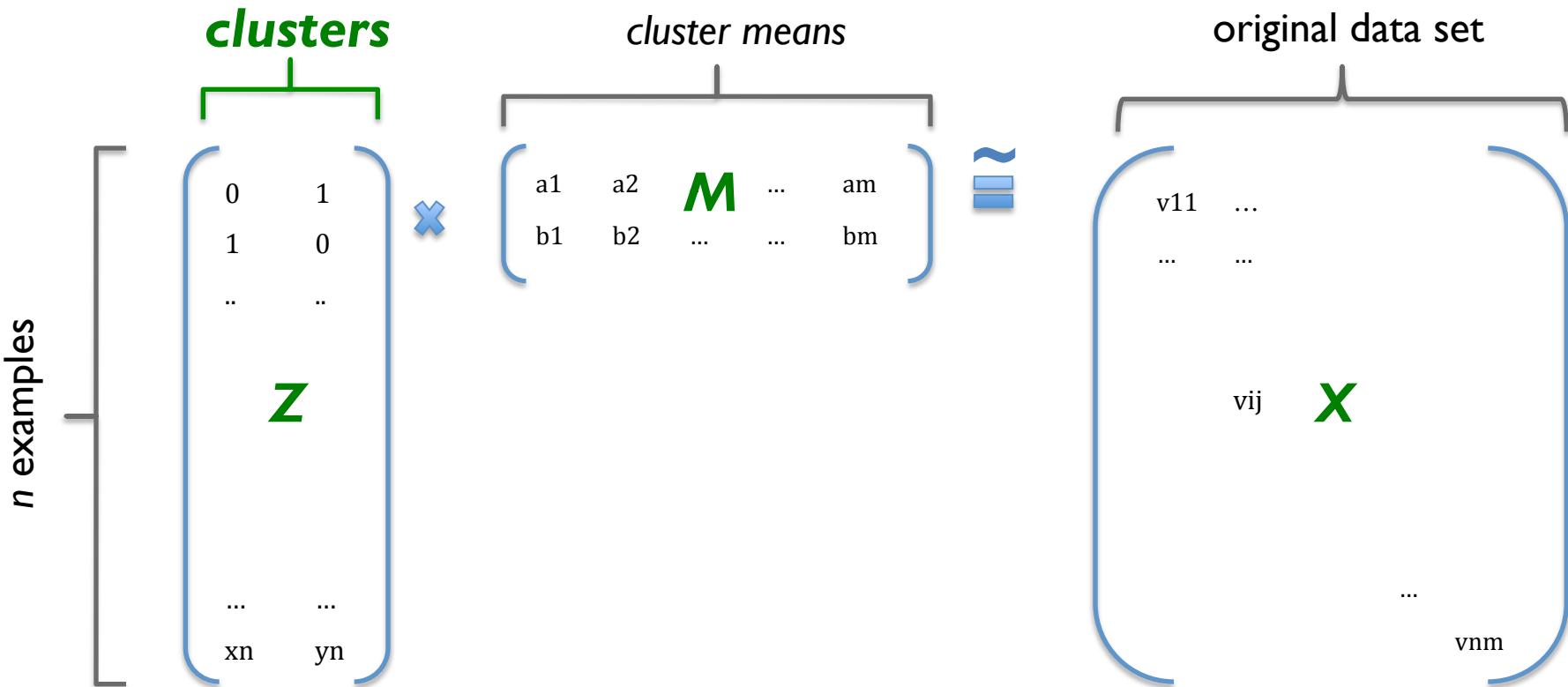
- Assume we would like to learn the following (trivial?) output function:
- Using the following network:
- With *linear* hidden units, how do the weights match up to W and H ?

Input	Output
00000001	00000001
00000010	00000010
00000101	00000100
00001000	00001000
00010000	00010000
00100000	00100000
01000000	01000000
10000000	10000000

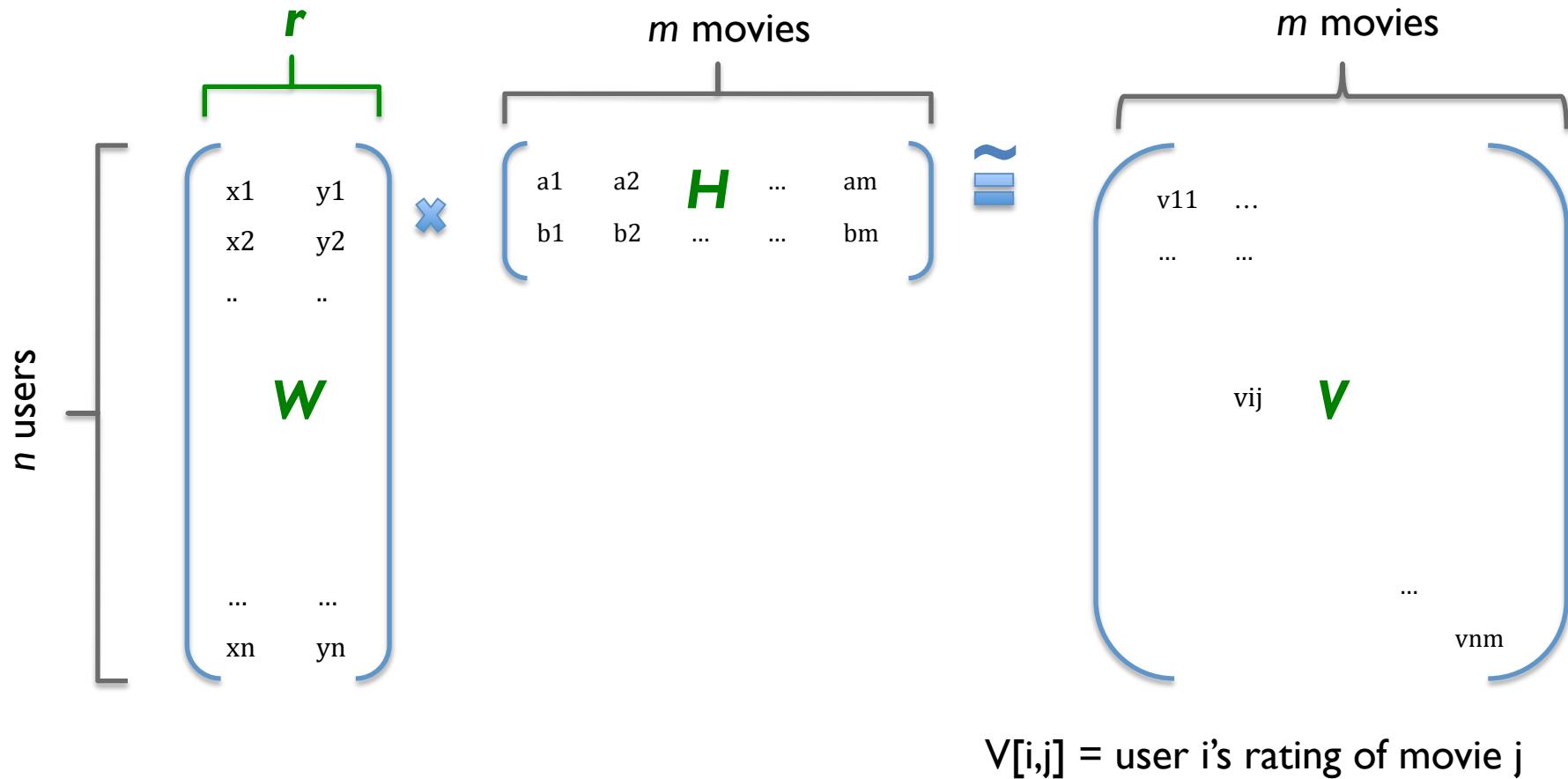


..... vs k-means

indicators for r



Recovering latent factors in a matrix



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