Matrix Factorization & Collaborative Filtering

Iran University of Science and Technology

M. S. Tahaei PhD.

Fall 2024

Machine Learning, Carnegie Mellon University

Outline

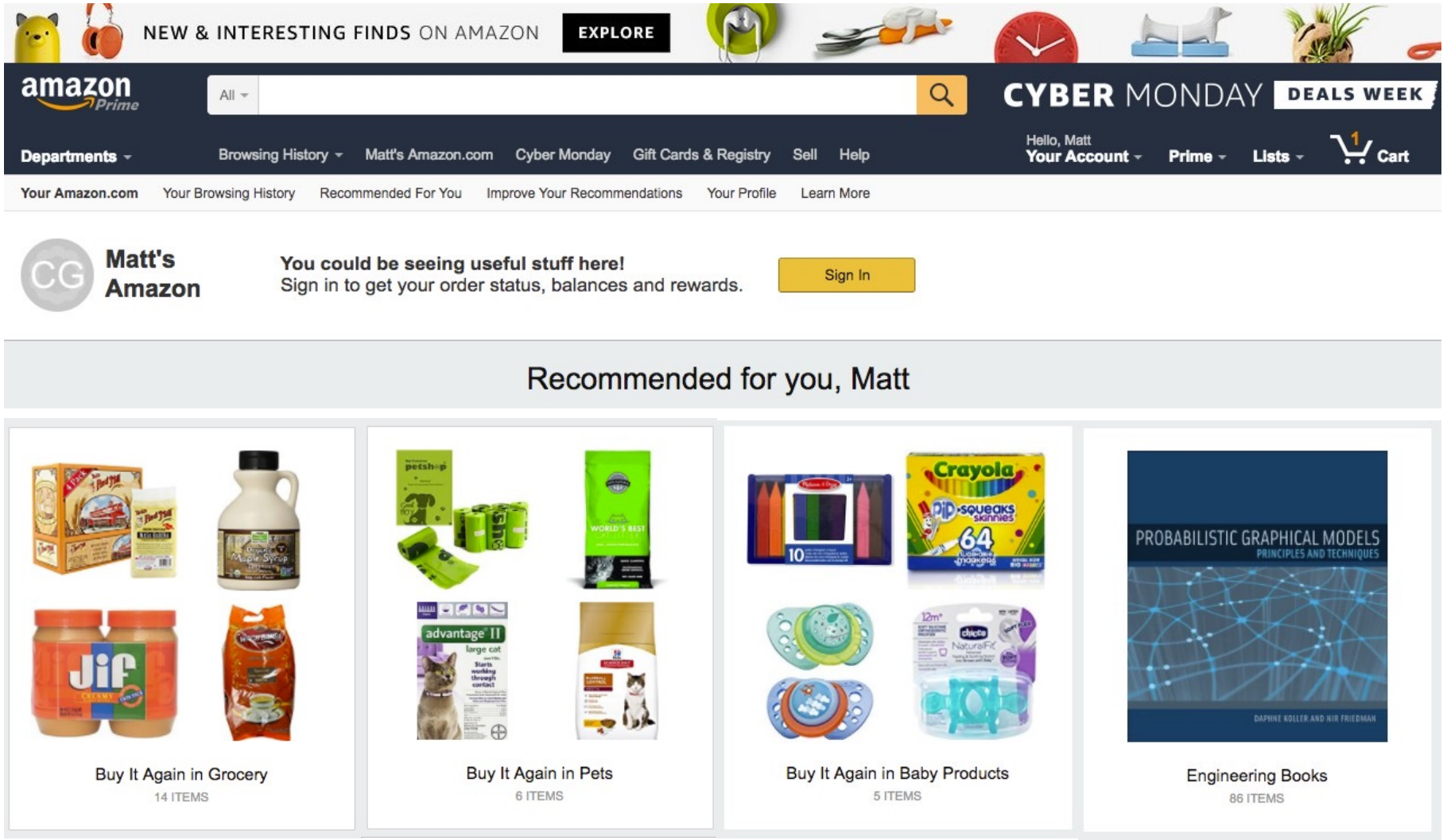
- **Recommender Systems**
  - Content Filtering
  - Collaborative Filtering (CF)
  - CF: Neighborhood Methods
  - CF: Latent Factor Methods
- **Matrix Factorization**
  - Optimization problem – SGD
  - SGD with Regularization
  - Alternating Least Squares
  - User/item bias terms (matrix trick)

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



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

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

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

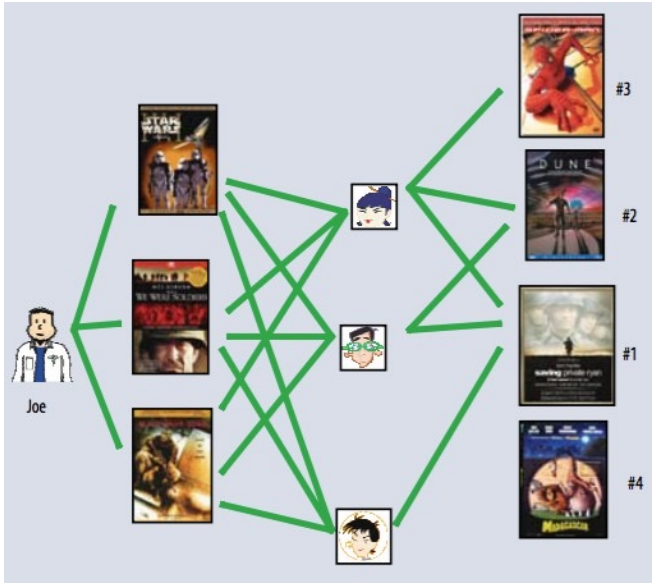
Collaborative Filtering

- **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

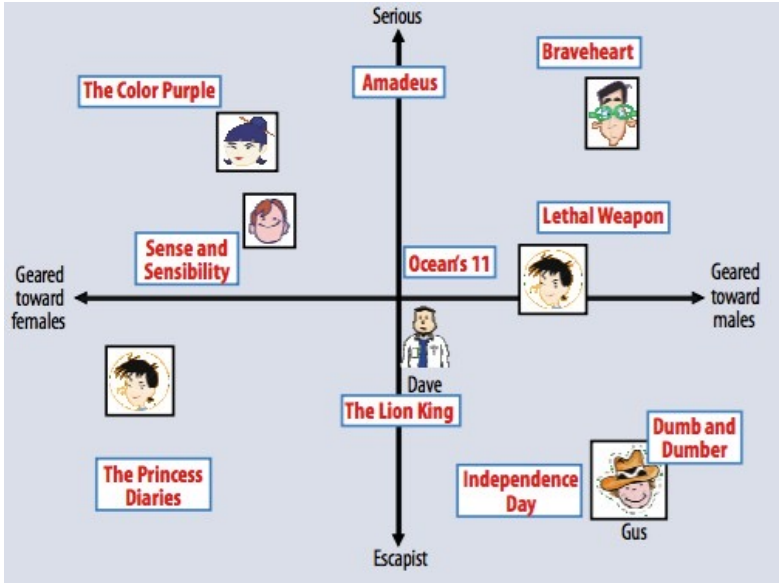
Slide from William Cohen

Two Types of Collaborative Filtering

1. Neighborhood Methods



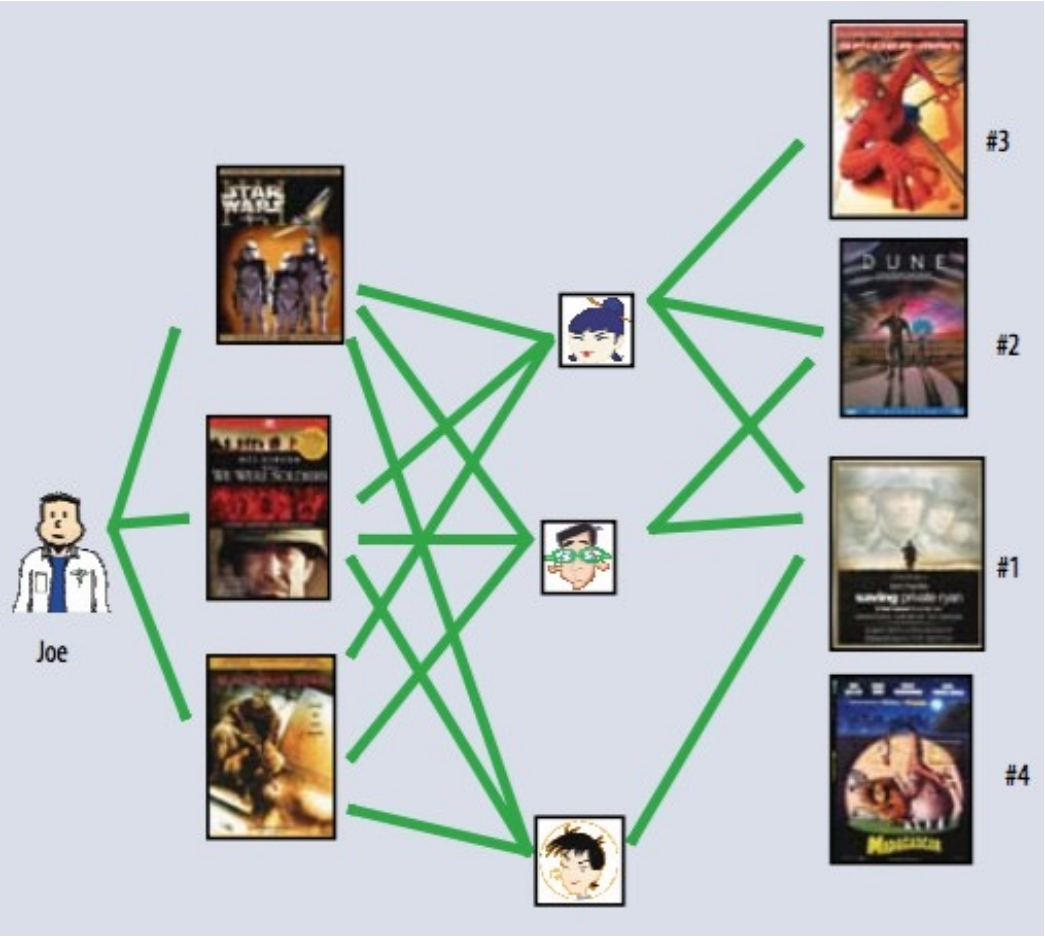
2. Latent Factor Methods



Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

1. Neighborhood Methods



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

- **Algorithm:**

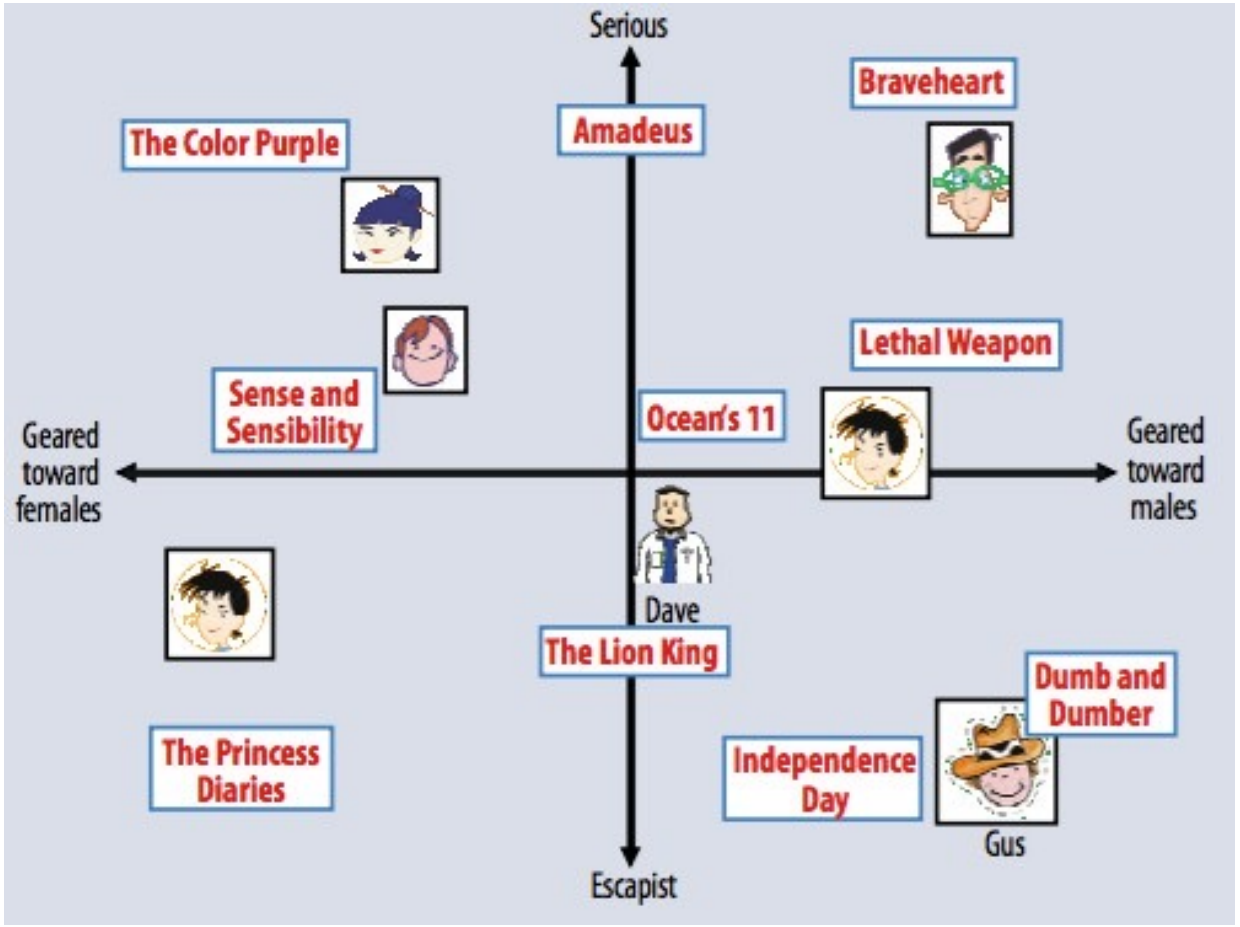
1. **Find neighbors** based on similarity of movie preferences
2. **Recommend** movies that those neighbors watched

Figures from Koren et al. (2009)

Two Types of Collaborative Filtering

- 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

2. Latent Factor Methods



Figures from Koren et al. (2009)

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

Question:

Applied to the Netflix Prize problem, which of the following methods *always* requires side information about the users and movies?

Select all that apply

1.K-Means

2.collaborative filtering

3.latent factor methods

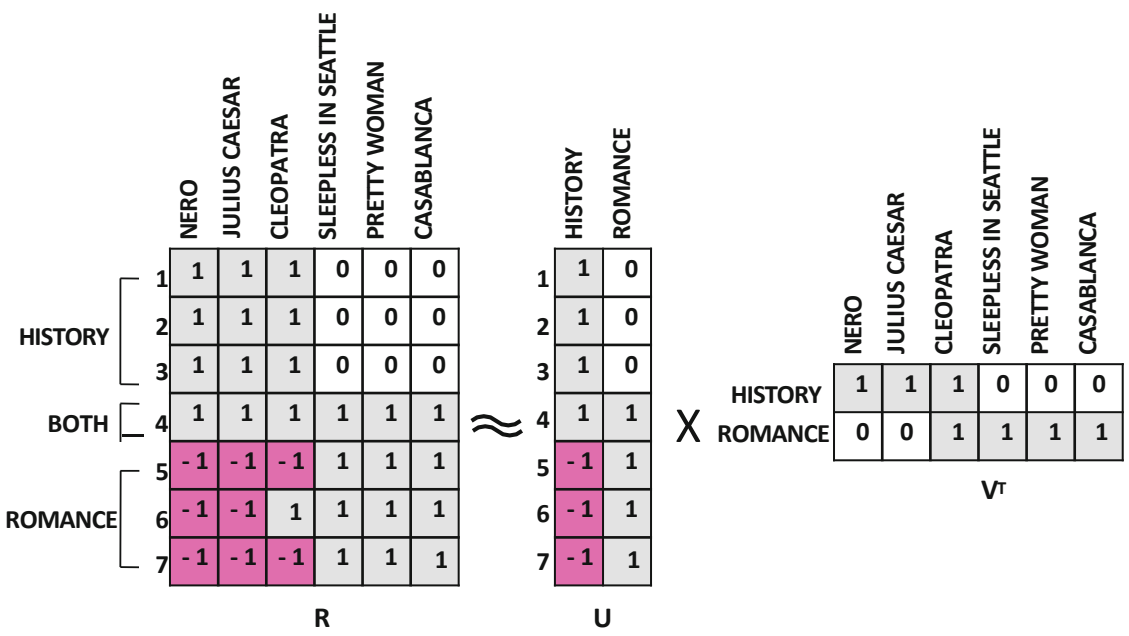
4.ensemble methods

5.content filtering

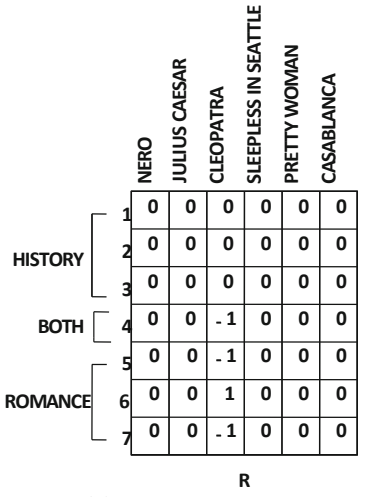
6.neighborhood methods

7.recommender systems

Example: MF for Netflix Problem



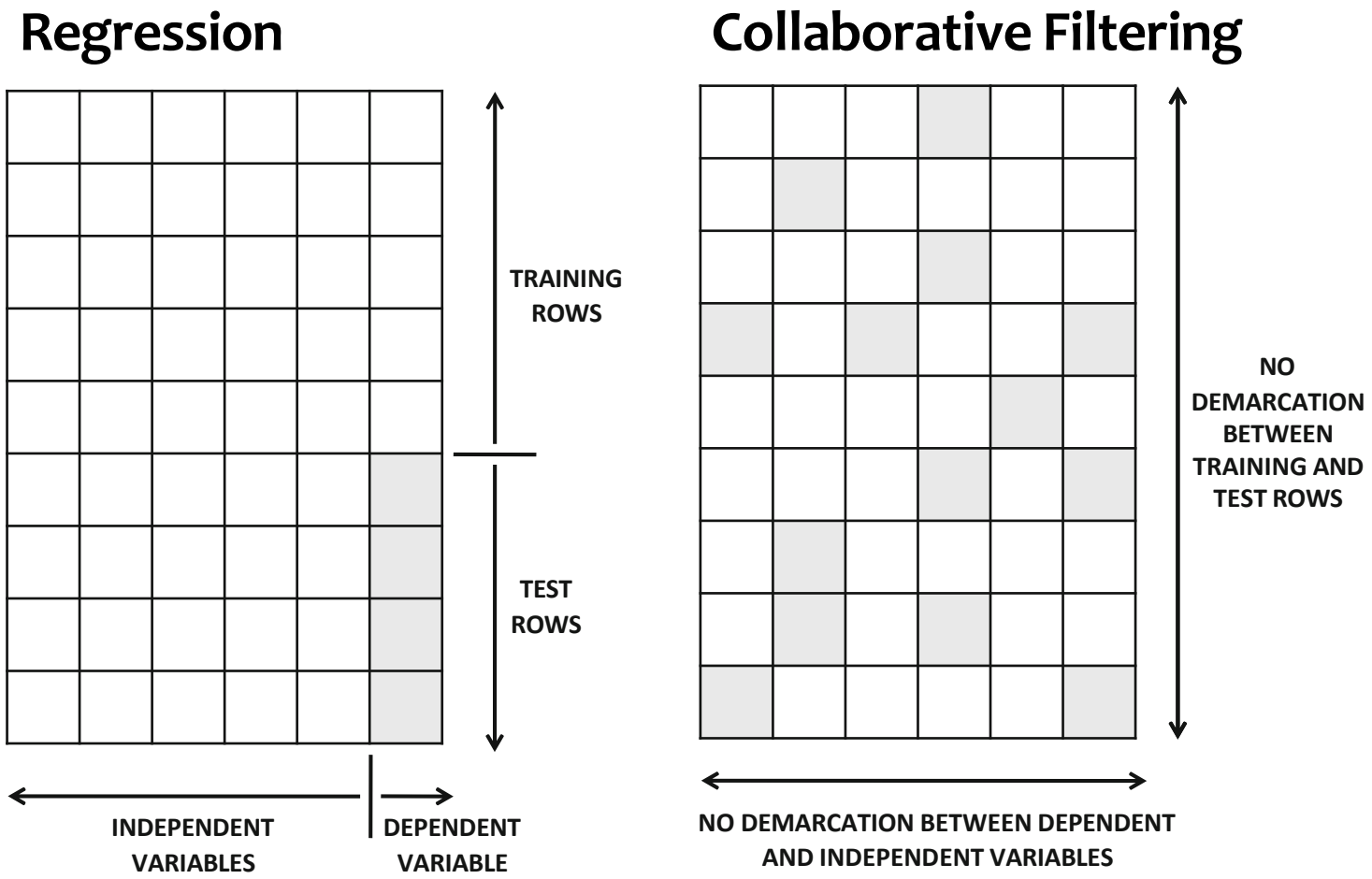
(a) Example of rank-2 matrix factorization



(b) Residual matrix

Figures from Aggarwal (2016)

Regression vs. Collaborative Filtering



Figures from Aggarwal (2016)

Unconstrained Matrix Factorization

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

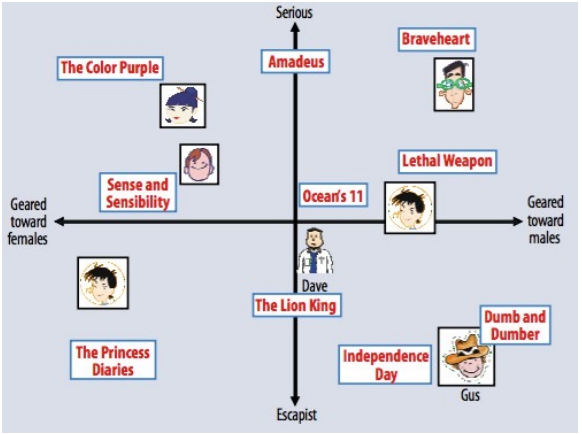
Unconstrained Matrix Factorization

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

Matrix Factorization
(with vectors)



Figures from Koren et al. (2009)

- 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:
  $$\underset{\mathbf{w},\mathbf{h}}{\operatorname{argmin}} \sum_{(u,i)\in\mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2$$
- SGD update for random (u,i):
  $$e_{ui} \leftarrow v_{ui} - \mathbf{w}_u^T \mathbf{h}_i$$
  $$\mathbf{w}_u \leftarrow \mathbf{w}_u + \gamma(e_{ui}\mathbf{h}_i - \lambda \mathbf{w}_u)$$
  $$\mathbf{h}_i \leftarrow \mathbf{h}_i + \gamma(e_{ui}\mathbf{w}_u - \lambda \mathbf{h}_i)$$
- Regularized Objective:
  $$\underset{\mathbf{w},\mathbf{h}}{\operatorname{argmin}} \sum_{(u,i)\in\mathcal{Z}} (v_{ui} - \mathbf{w}_u^T \mathbf{h}_i)^2 + \lambda\left(\sum_i ||\mathbf{w}_i||^2 + \sum_u ||\mathbf{h}_u||^2\right)$$

Matrix Factorization

(with matrices)

- User vectors:

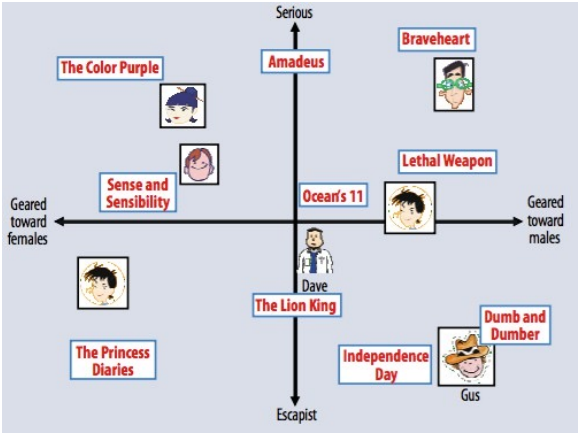
$$(W_u)^T \in \mathbb{R}^r$$

- Item vectors:

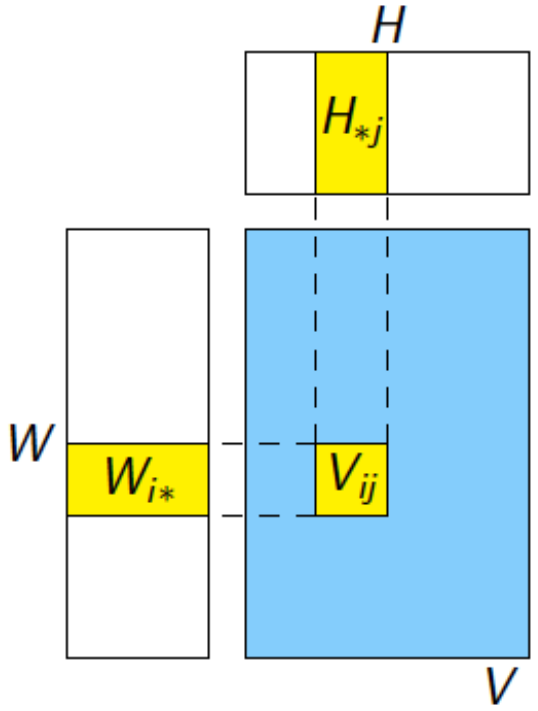
$$H_i \in \mathbb{R}^r$$

- Rating prediction:

$$V_{ui} = W_u H_i$$
$$= [WH]_{ui}$$



Figures from Koren et al. (2009)



Matrix Factorization

(with matrices)

- SGD

require that the loss can be written as

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

Algorithm 1 SGD for Matrix Factorization

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

while not converged **do** {step}

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

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

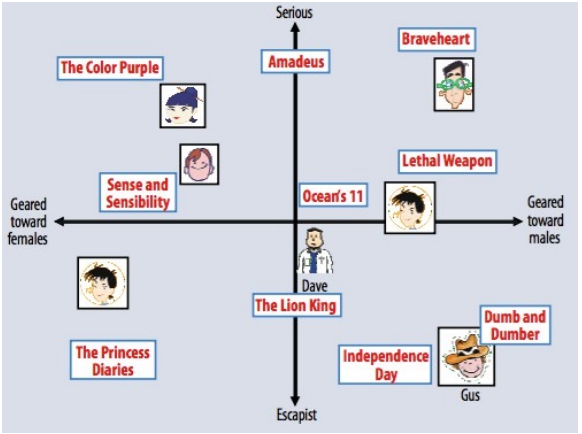
$$\boldsymbol{H}_{*j} \leftarrow \boldsymbol{H}_{*j} - \epsilon_n N \frac{\partial}{\partial \boldsymbol{H}_{*j}} l(\boldsymbol{V}_{ij}, \boldsymbol{W}_{i*}, \boldsymbol{H}_{*j})$$

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

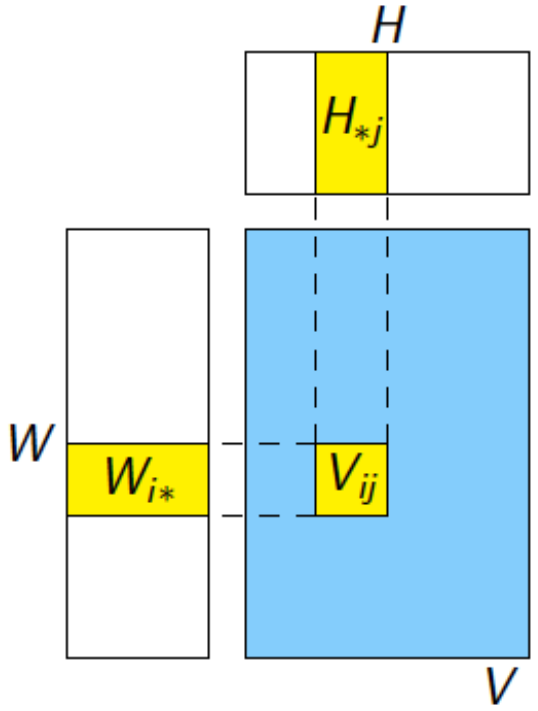
end while

step size

Figure from Gemulla et al. (2011)



Figures from Koren et al. (2009)



Singular Value Decomposition for Collaborative Filtering

For any arbitrary matrix $\mathbf{A}$, SVD gives a decomposition:

$$\mathbf{A} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$$

where $\mathbf{\Lambda}$ is a diagonal matrix, and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices.

Suppose we have the SVD of our ratings matrix

$$R = Q\Sigma P^T,$$

but then we truncate each of Q, Σ, and P s.t. Q and P have only k columns and Σ is $k \times k$:

$$R \approx Q_k \Sigma_k P_k^T$$

For collaborative filtering, let:

$$U \triangleq Q_k \Sigma_k$$

$$V \triangleq P_k$$

$$\Rightarrow U, V = \underset{U,V}{\operatorname{argmin}} \frac{1}{2} ||R - UV^T||_2^2$$

s.t. columns of U are mutually orthogonal

s.t. columns of V are mutually orthogonal

Non-negative Matrix Factorization

Constrained Optimization Problem:

$$U, V = \underset{U,V}{\operatorname{argmin}} \frac{1}{2}||R - UV^T||_2^2$$

$$\text{s.t. } U_{ij} \geq 0$$

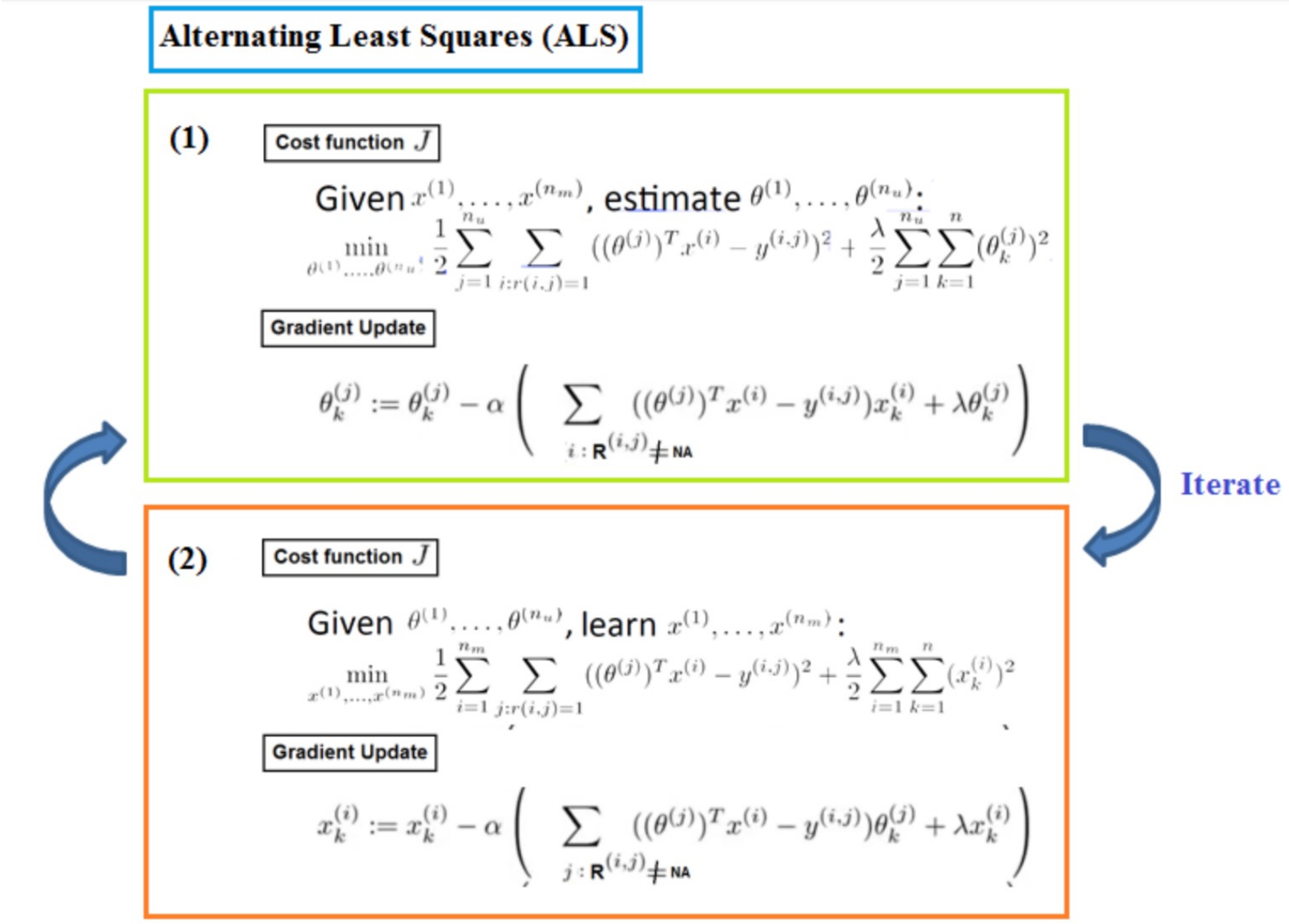
$$\text{s.t. } V_{ij} \geq 0$$

Multiplicative Updates: simple iterative algorithm for solving just involves multiplying a few entries together

SVD vs MF

Theorem: If R fully observed and no regularization, the optimal UV^T from SVD equals the optimal UV^T from Unconstrained MF

Alternating Least Squares (ALS) for UMF:



Matrix Factorization

Comparison of Optimization Algorithms

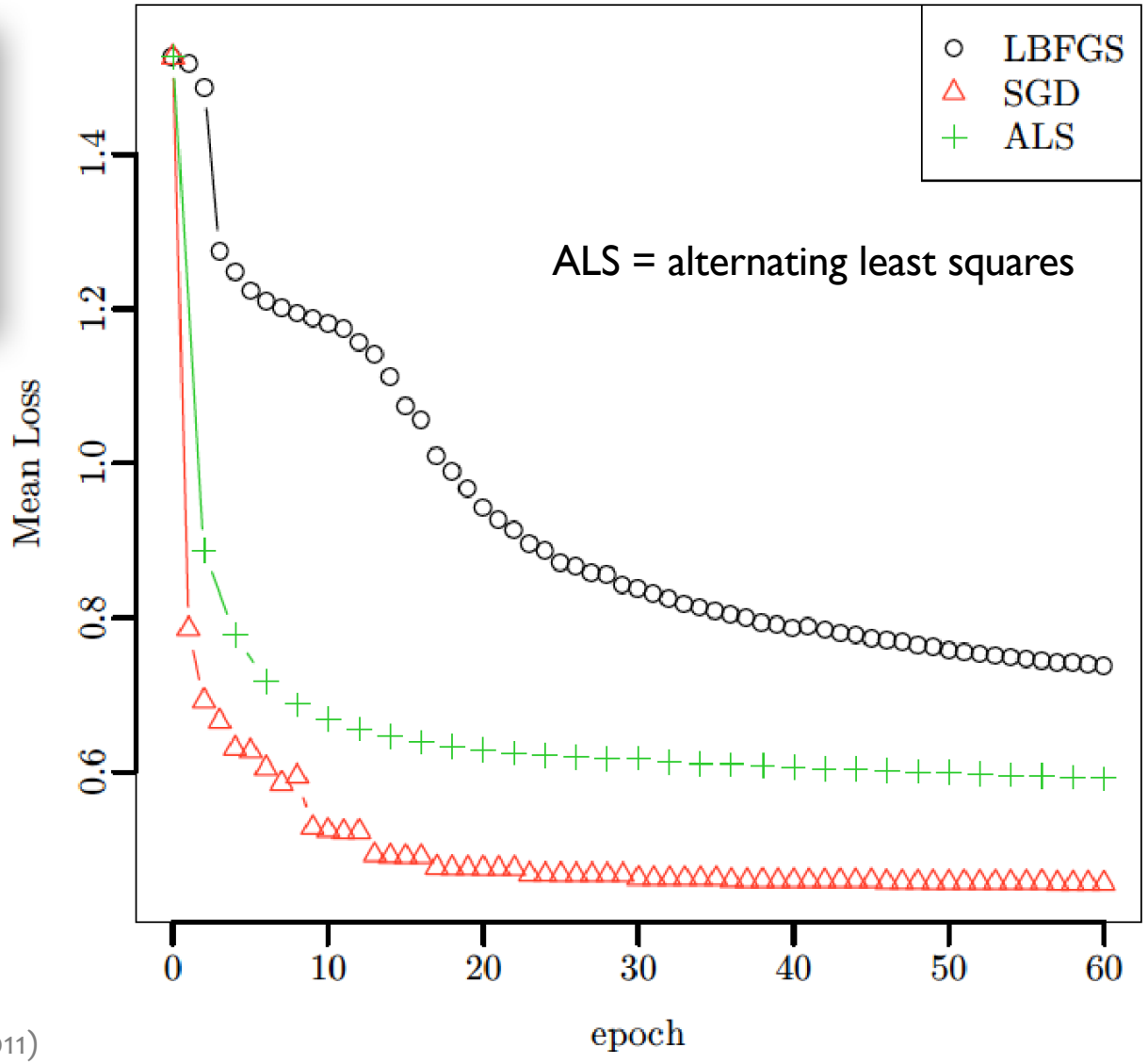


Figure from Gemulla et al. (2011)

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

Examples from Aggarwal (2016)

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