6.867 Machine Learning Fall 2017

Lecture 21. Matrix Estimation

http://stellar.mit.edu/S/course/6/fa17/6.867/

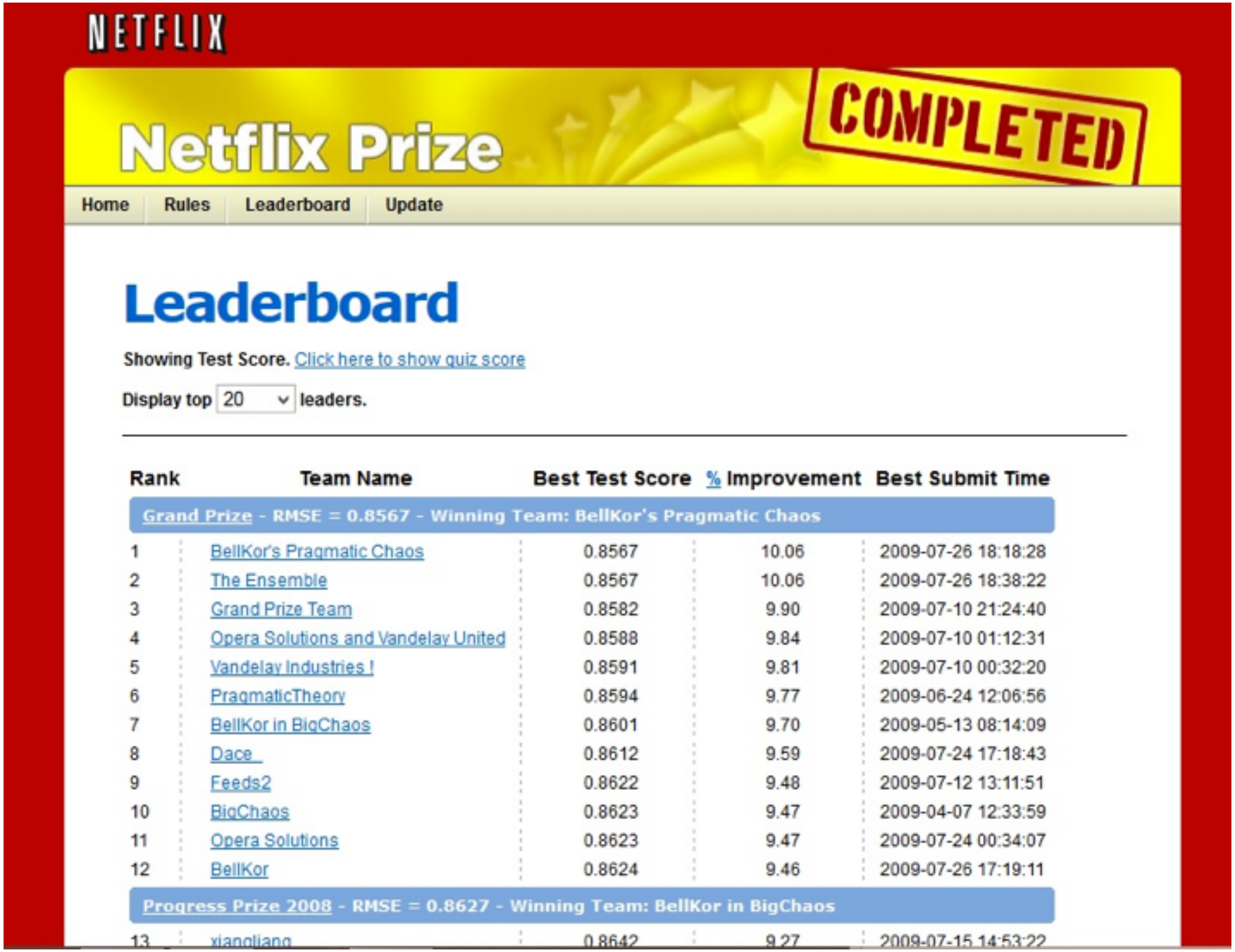
Announcements

- Trust Thanks Giving Break Was Good!
- Quiz 2
  - Thursday, November 30 7pm-9pm
    - No Lecture on that day
  - Make up:
    - TODAY: Tuesday, November 28, 4pm-6pm
  - Quiz Review:
    - Posted online with its solutions
    - Tomorrow Wednesday, November 29 during TA OH
- Exercise 11 will be posted soon: covers PCA and today's lecture

Outline

- Matrix Estimation
  - An Example Application
  - Formulation
    - Singular Value Thresholding
    - Collaborative Filtering
  - Probabilistic Latent Variable Model
  - Alternative Least Squares and Taylor's Expansion
- More Applications
- BenchMark Data Sets

Netflix Challenge circa 2008



NETFLIX

Netflix Prize

COMPLETED

Home | Rules | Leaderboard | Update

Leaderboard

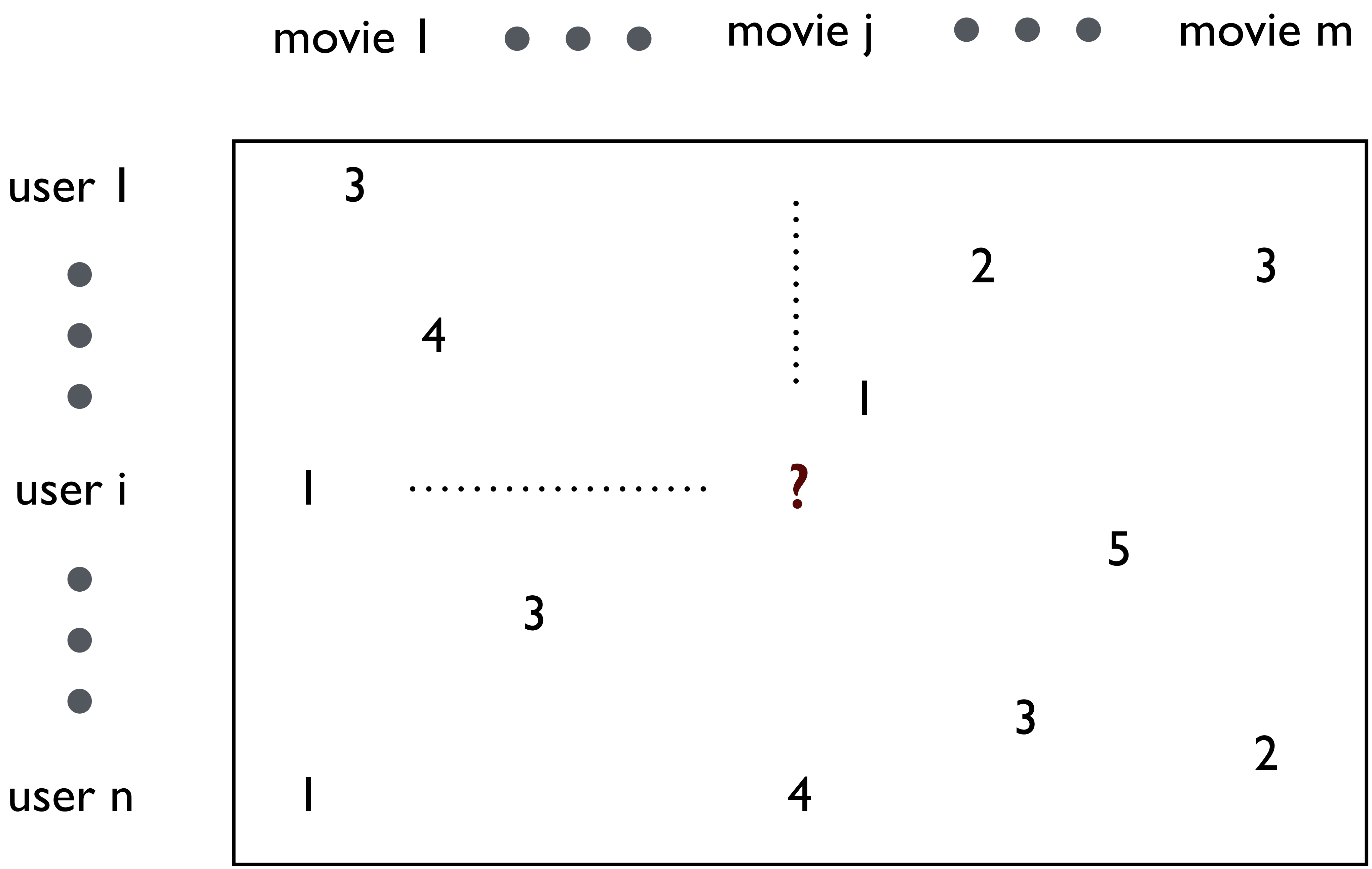
Showing Test Score. Click here to show quiz score

Display top 20 leaders.

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
Progress Prize 2008 - RMSE = 0.8627 - Winning Team: BellKor in BigChaos				
13	xiangliang	0.8642	9.27	2009-07-15 14:53:22

Matrix factorization techniques for recommender systems by Koren, Bell and Volinsky Computer 42:8, 2009

Recommendation system



Rating Matrix A

Matrix Estimation

- Ground Truth Matrix

$$A = [A_{ij}] \in \mathbb{R}^{m \times n}$$

- Observation Matrix $Y = [Y_{ij}] \in \mathbb{R}^{m \times n}$

$$Y_{ij} = \begin{cases} \text{noisy } A_{ij} & \text{if observed} \\ \star & \text{otherwise} \end{cases}$$

- Goal: produce estimate $\hat{A} = [\hat{A}_{ij}]$ so that prediction error is small

$$\frac{1}{mn} \sum_{ij} \left(\hat{A}_{ij} - A_{ij}\right)^2$$

Structure in a Matrix

- $A \in \mathbb{R}^{m \times n}$ has singular value decomposition: for $r = \min\{m, n\}$

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$$\text{where} \quad \sigma_i \geq 0,\ u_i \in \mathbb{R}^m,\ v_i \in \mathbb{R}^n$$

- Equivalently:

$$A = U \Sigma V^T$$

$$\text{where} \quad U \in \mathbb{R}^{m \times r},\ \Sigma \in \mathbb{R}^{r \times r},\ V \in \mathbb{R}^{n \times r}$$

- That is

$$A = U \tilde{V}^T$$

$$\text{where} \quad U \in \mathbb{R}^{m \times r}, \tilde{V} \in \mathbb{R}^{n \times r}$$

Exploiting Structure in a Matrix

- A natural estimation algorithm exploiting structure

$$\text{minimize} \sum_{(i,j)\in\mathcal{O}} \left(Y_{ij} - U_{i\cdot}^T V_{j\cdot}\right)^2$$

$$\text{over} \quad U \in \mathbb{R}^{m\times r}$$

$$V \in \mathbb{R}^{n\times r}$$

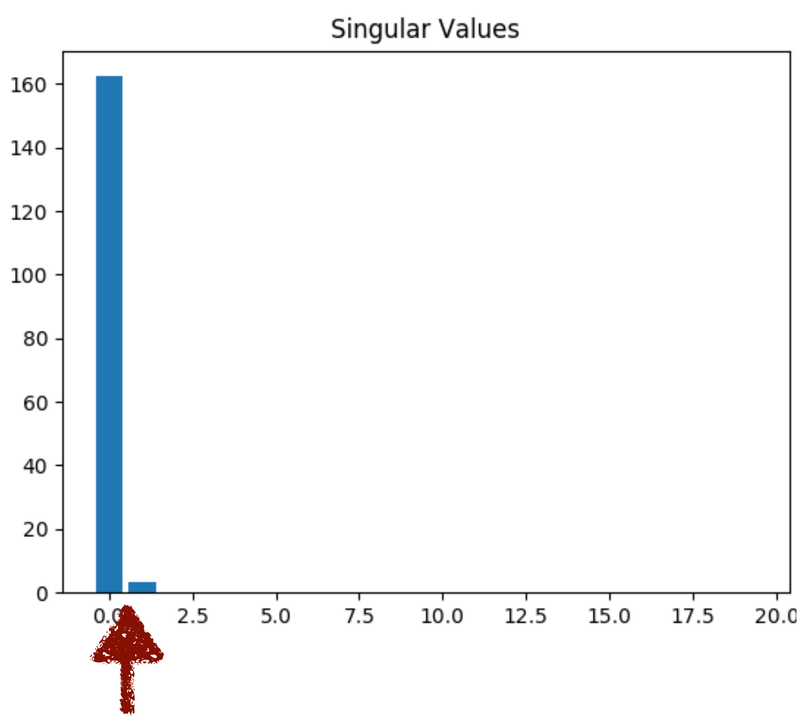
- In above $\mathcal{O} \subset [m] \times [n]$ set of entries for which entries are observed
- And number of unknowns is (m+n) r
  - So if $|\mathcal{O}|$ is small, we can not expect r to be large
- In general, this isn't computationally *easy* optimization problem

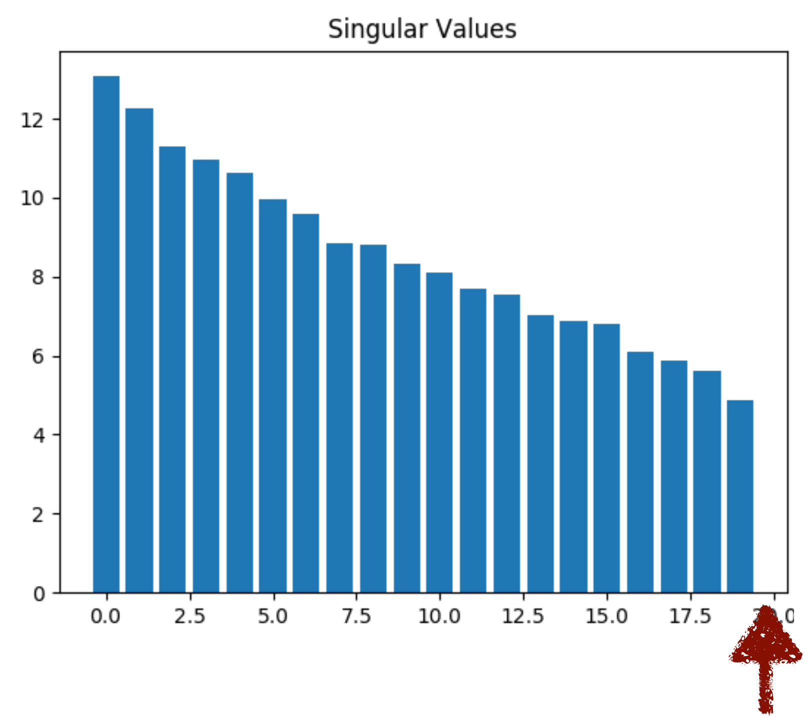
Singular Value Thresholding

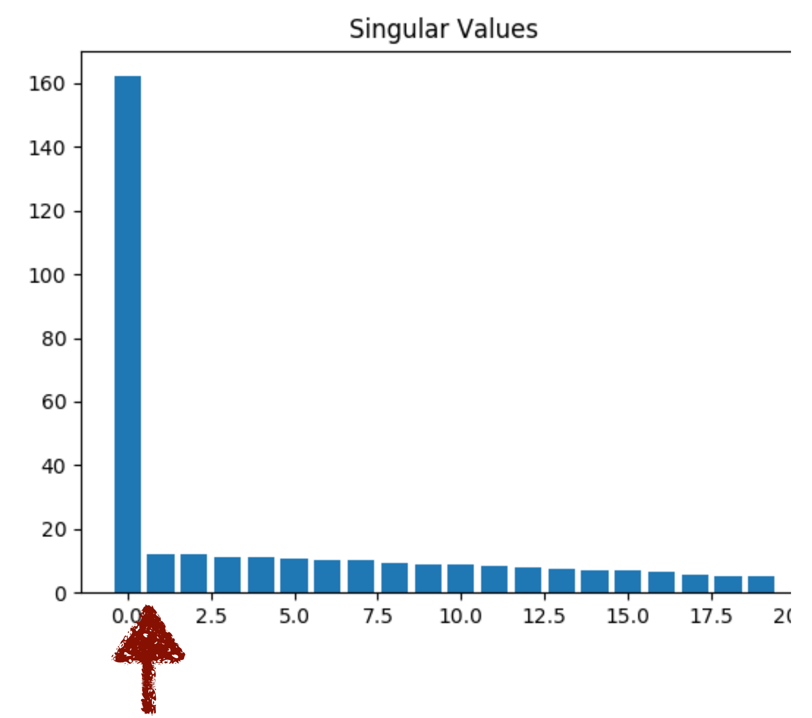
- An extremely *simple* algorithm
  - Define $\hat{Y} = [\hat{Y}_{ij}]$ as
$$\hat{Y}_{ij} = \begin{cases} Y_{ij} & \text{if } (i,j) \in \mathcal{O} \\ 0 & \text{otherwise} \end{cases}$$
  - Compute Singular Value Decomposition of $\hat{Y}$
$$\hat{Y} = \sum_{i=1}^{r} \hat{\sigma}_i \hat{u}_i \hat{v}_i^T$$
  - Estimated matrix $\hat{A} = \frac{1}{\hat{p}} \sum_{i \in S} \hat{\sigma}_i \hat{u}_i \hat{v}_i^T$
  - where $S = \{j : \sigma_j \geq \mu\},\ \hat{p} = \frac{|\mathcal{O}|}{mn}$ for some threshold μ

Singular Value Thresholding

- What threshold to choose?
  - Universal threshold: $\mu = 2\sqrt{\max\{m,n\}\hat{p}}$
  - In practice, plot the *spectrum* and look for *knee*

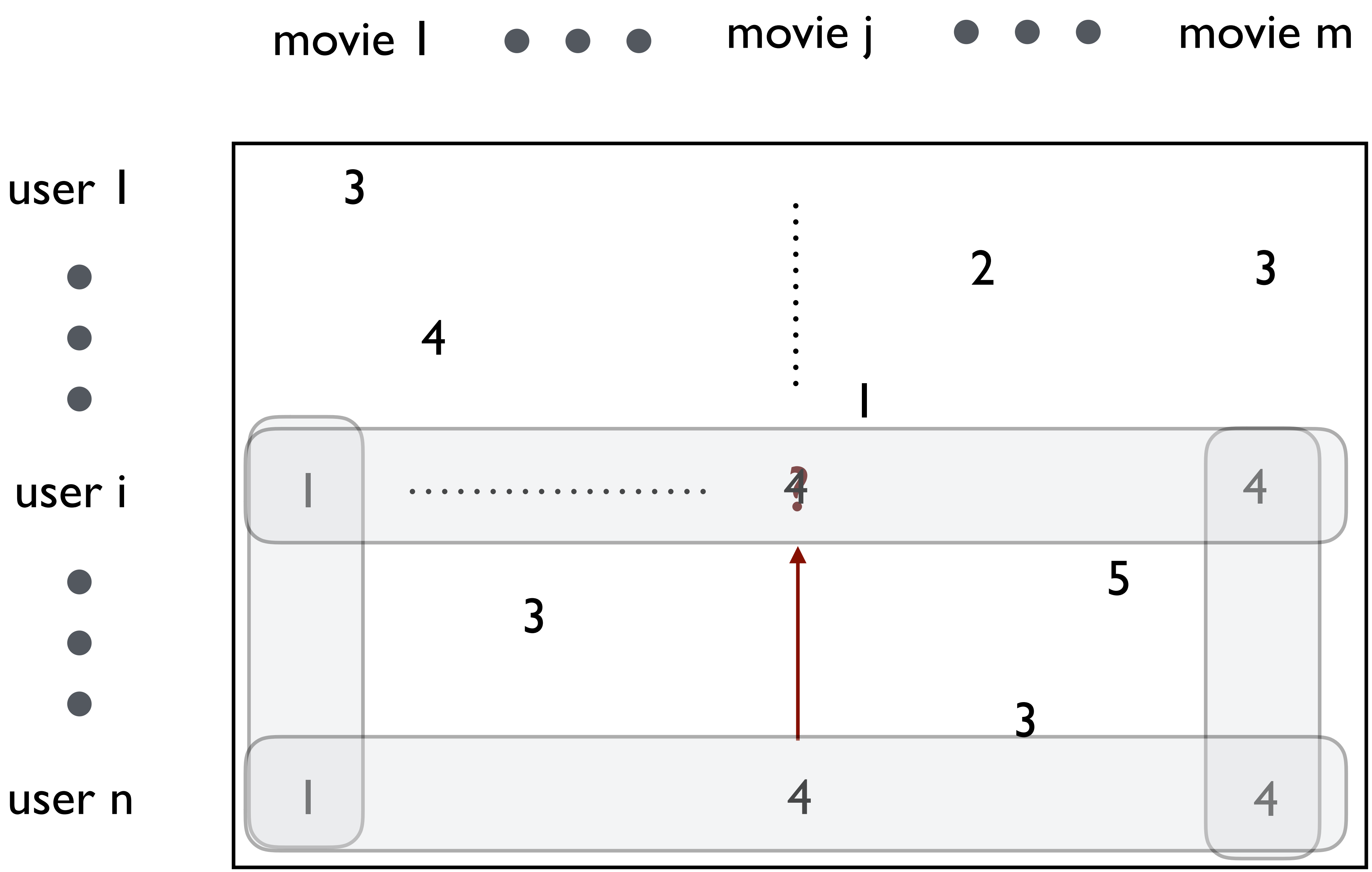






Exploiting Similarities

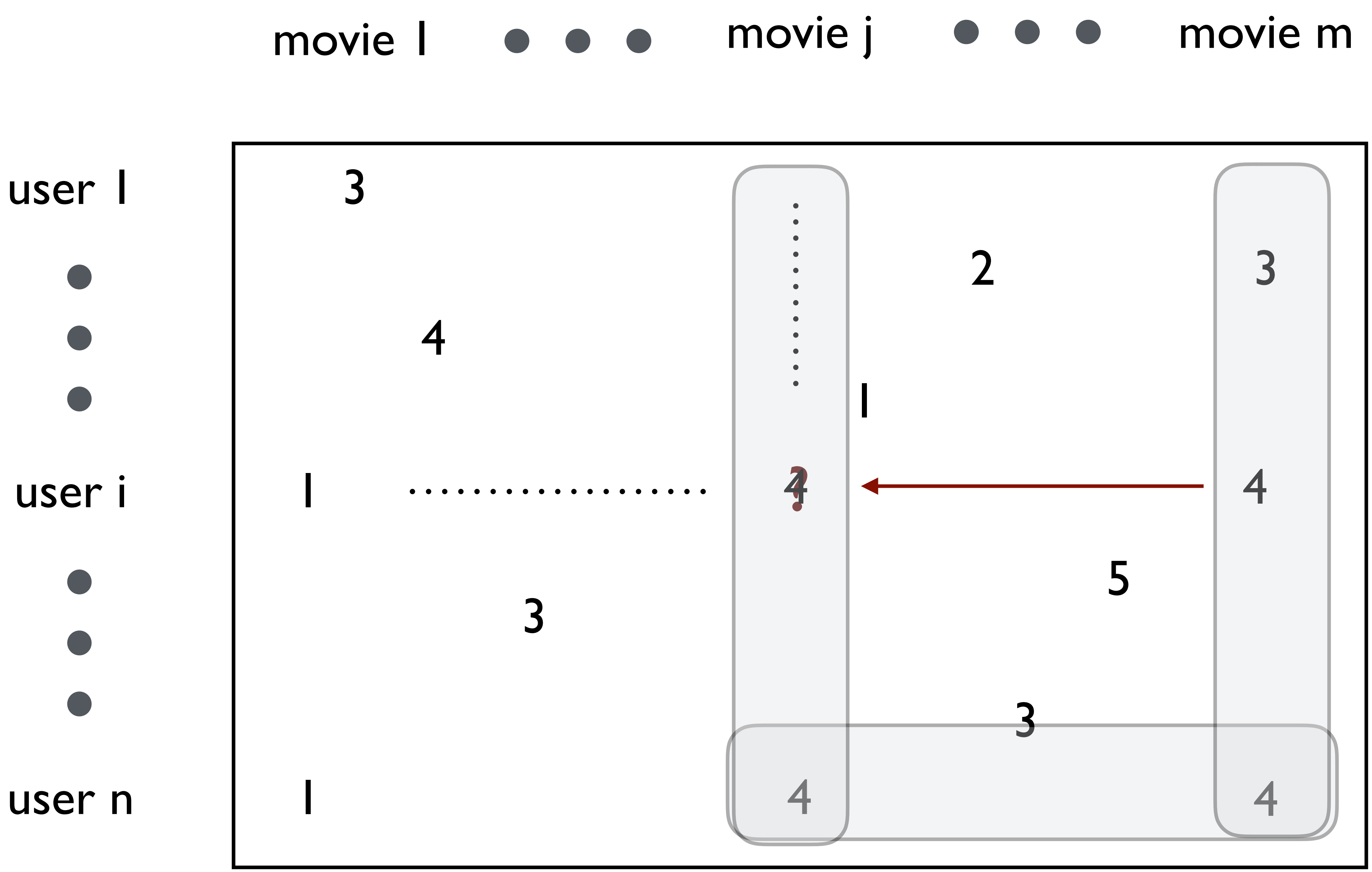
Collaborative filtering [Goldberg et al 92]



user-user collaborative filtering

Exploiting Similarities

Collaborative filtering [Goldberg et al 92]



item-item collaborative filtering

Exploiting Similarities

Collaborative filtering (CF)

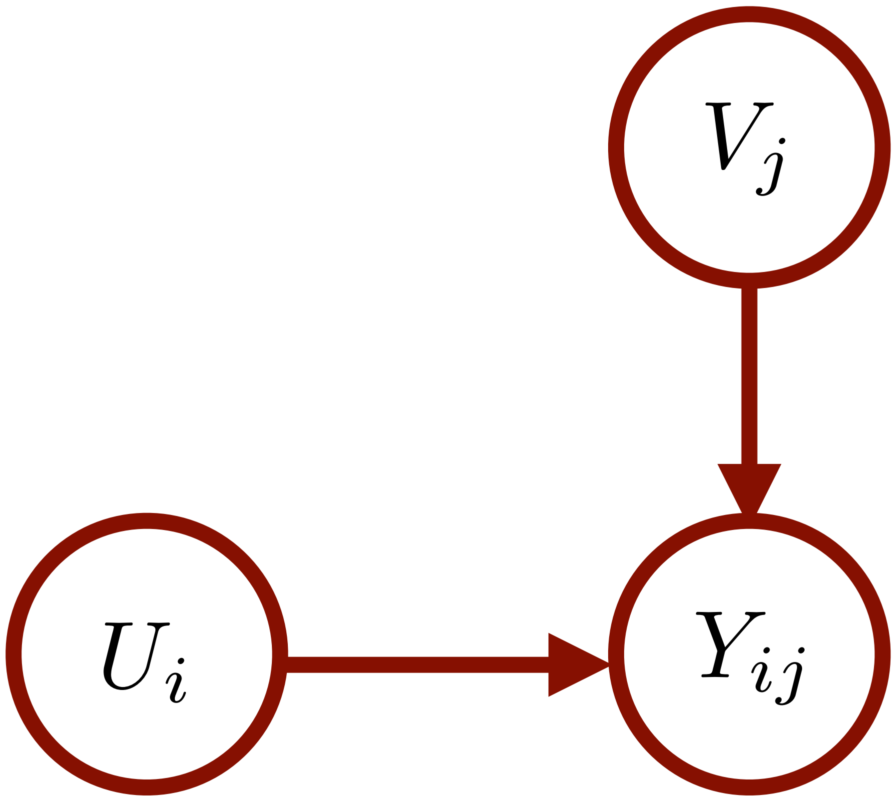
- extensively utilized in practice
  - scalable, incremental, robust and interpretable
- [Melville et al 02], [Wang et al 06], [Bell-Koren 07], [Koren et al 09]
- conceptual relationship to nearest neighbors
  - mixture distribution model for preferences across users/movies
- [Kleinberg-Sandler 04], [Dabeer 13] [Xu et al 13] [Bresler et al 14, 16]

Exploiting Similarities

- Collaborative Filtering Algorithm and It's Variations
  - Extensively used in practice
  - Scalable
    - Using "approximate nearest neighbor" data structure
  - Incremental
    - New data can be easily incorporated incrementally
  - Interpretable
    - Watch *Godfather* because you liked *Goodfellas*
  - Relationship to *nearest-neighbor* or *Kernel* based algorithm

Probabilistic Model

- Latent Variable Model



Probabilistic Model

- Latent Variable Model
  - Latent variable of Row i, U_i is drawn i.i.d. from distribution $\mathcal{U}$
  - Latent variable for column j, V_j is drawn i.i.d. from distribution $\mathcal{V}$
  - Ground truth entry $A_{ij} = f(U_i, V_j)$ for all i, j
    - for some latent function ***f***
  - If observed, Y_{ij} is independent random variable such that

$$\mathbb{E}[Y_{ij} | U_i, V_j] = A_{ij} = f(U_i, V_j)$$

- This is closely related to canonical representation for
  - "Row-Column Exchangeable" random variables [Hoover 79, 82], [Aldous 81, 82, 85]

Alternative Least Squares (ALS)

- Let the latent function be bilinear $f(U_i, V_j) = U_i^T V_j$
- An EM-like or Alternative Minimization Algorithm for solving

$$\text{minimize} \sum_{(i,j)\in\mathcal{O}} \left(Y_{ij} - U_{i\cdot}^T V_{j\cdot}\right)^2$$

$$\text{over} \quad U \in \mathbb{R}^{m\times r}$$

$$V \in \mathbb{R}^{n\times r}$$

- Assuming Vs fixed, solving for Us decomposes per row: for row *i*

$$\text{minimize} \sum_{j:(i,j)\in\mathcal{O}} \left(Y_{ij} - U_{i\cdot}^T V_{j\cdot}\right)^2$$

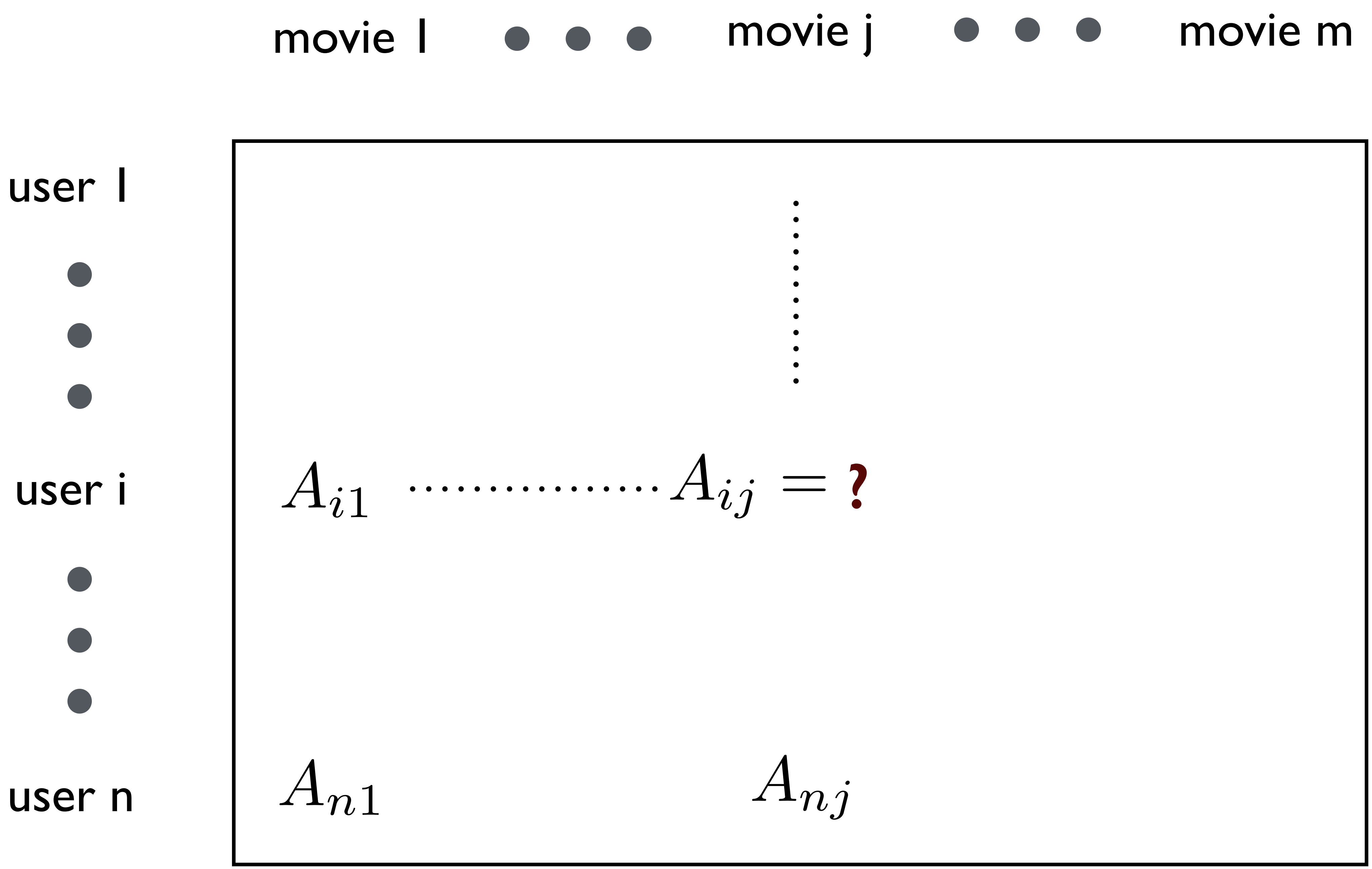
$$\text{over} \quad U_{i\cdot} \in \mathbb{R}^{m}$$

- This is classical *Regression* or *Ordinary Least Squares* problem!

Alternative Least Squares (ALS)

- In summary
  - Initialize $U^0 \in \mathbb{R}^{m \times r},\ V^0 \in \mathbb{R}^{n \times r}$ appropriately
  - Iteratively:
    - set U^{t+1} assuming V^t fixed
      - This is requires solving *m* different least squares problems
    - set V^{t+1} assuming U^{t+1} fixed
      - This requires solving *n* different least squares problems
  - Stop upon "convergence"

Taylor's Expansion



Taylor's Expansion

movie 1 • • • movie j • • • movie m

(x_2') (x_2)

user 1

•
•
•

user i (x_1) $f(x_1, x_2')$ A_{i1} $\cdots$ $f(x_1, x_2)$ A_{ij} = ?

•
•
•

user n (x_1') $f(x_1', x_2')$ A_{n1} $f(x_1', x_2)$ A_{nj}

Taylor's Expansion

For simplicity, assume $x_1' = \mathbf{0},\ x_2' = \mathbf{0}$

$$f(x_1, x_2) = f(\mathbf{0}, \mathbf{0}) + x_1 \frac{\partial f(\mathbf{0}, \mathbf{0})}{\partial x_1} + x_2 \frac{\partial f(\mathbf{0}, \mathbf{0})}{\partial x_2}$$

$$f(x_1, \mathbf{0}) = f(\mathbf{0}, \mathbf{0}) + x_1 \frac{\partial f(\mathbf{0}, \mathbf{0})}{\partial x_1}$$

$$f(\mathbf{0}, x_2) = f(\mathbf{0}, \mathbf{0}) + x_2 \frac{\partial f(\mathbf{0}, \mathbf{0})}{\partial x_2}$$

$$f(x_1, x_2) = f(x_1, \mathbf{0}) + f(\mathbf{0}, x_2) - f(\mathbf{0}, \mathbf{0})$$

Taylor's Expansion

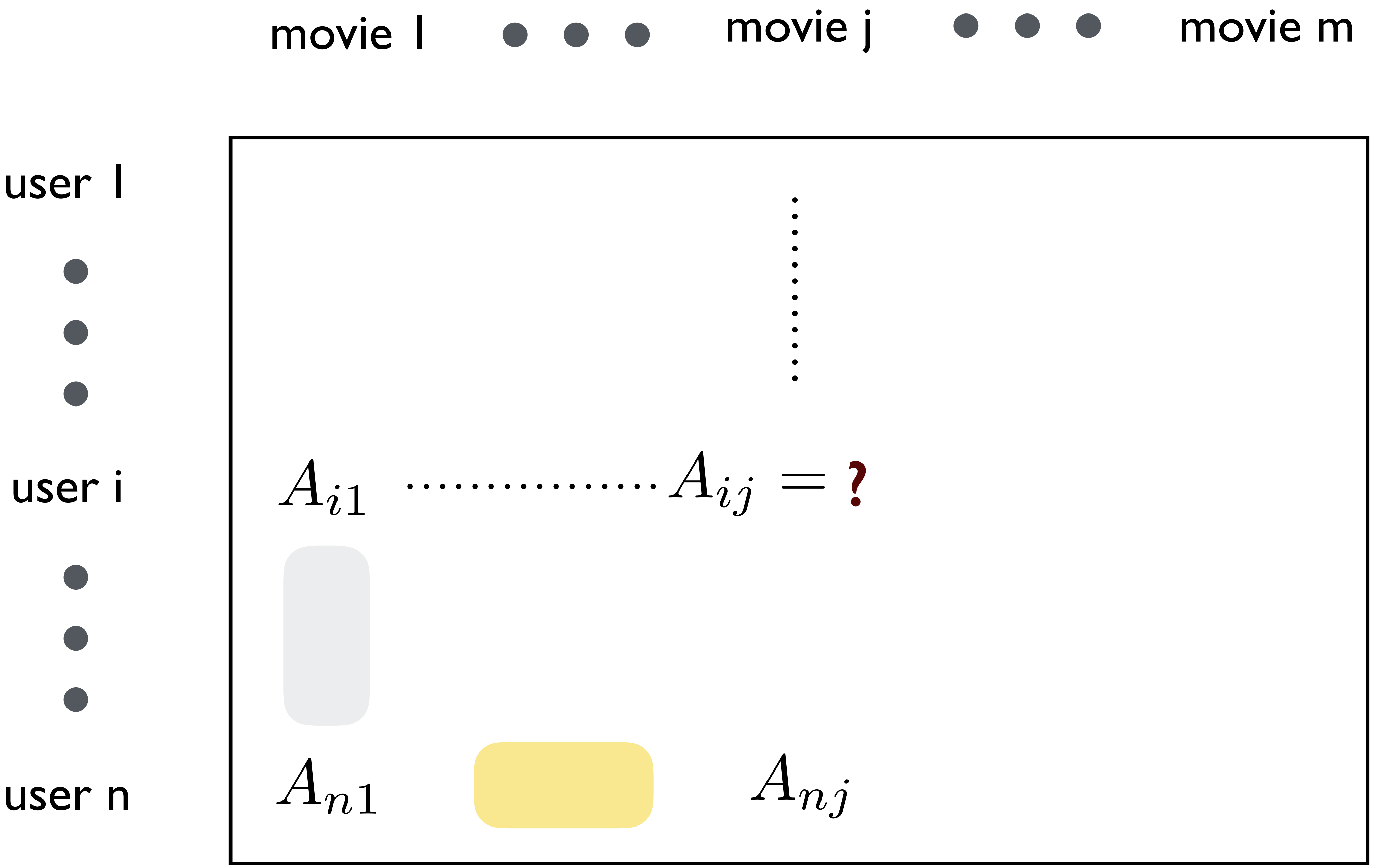
For simplicity, assume $x_1' = \mathbf{0},\ x_2' = \mathbf{0}$

$$f(x_1, x_2) = f(x_1, \mathbf{0}) + f(\mathbf{0}, x_2) - f(\mathbf{0}, \mathbf{0})$$

$$A_{ij} = A_{i1} + A_{nj} - A_{n1}$$

$$A_{ij} = A_{n1} + (A_{i1} - A_{n1}) + (A_{nj} - A_{n1})$$

Taylor's Expansion



$$A_{ij} = A_{n1} + (A_{i1} - A_{n1}) + (A_{nj} - A_{n1})$$

Taylor's Expansion

$$A_{ij} = A_{n1} + (A_{i1} - A_{n1}) + (A_{nj} - A_{n1})$$

or

$$f(x_1, x_2) = f(x_1, \mathbf{0}) + f(\mathbf{0}, x_2) - f(\mathbf{0}, \mathbf{0})$$

This assumes that

$$x_1 \approx \mathbf{0} \qquad x_2 \approx \mathbf{0}$$

Hard to verify this condition

since we do not observe features

Taylor's Expansion

A proxy: use rows and columns that minimize prediction error

error due to row selection

$$\mathbb{E}\left[\text{error}^2 \mid x_1, x_1'\right] = \text{Var}_{\mathbf{x}}[f(x_1, \mathbf{x}) - f(x_1', \mathbf{x})]$$

error due to column selection

$$\mathbb{E}\left[\text{error}^2 \mid x_2, x_2'\right] = \text{Var}_{\mathbf{x}}[f(\mathbf{x}, x_2) - f(\mathbf{x}, x_2')]$$

Taylor's Expansion

Predict rating of entry (i,j):

$$\hat{A}_{ij} = A_{kl} + (A_{il} - A_{kl}) + (A_{kj} - A_{kl})$$

where

dist(i, k) and **dist**(j,l) are *small*

all necessary entries are revealed

multiple such predictions are combined by

weighing each of them as per Gaussian Kernel using **dist**

Taylor's Expansion vs Collaborative Filtering

Predict rating of entry (i,j):

$$\hat{A}_{ij} = A_{kl} + (A_{il} - A_{kl}) + (A_{kj} - A_{kl})$$

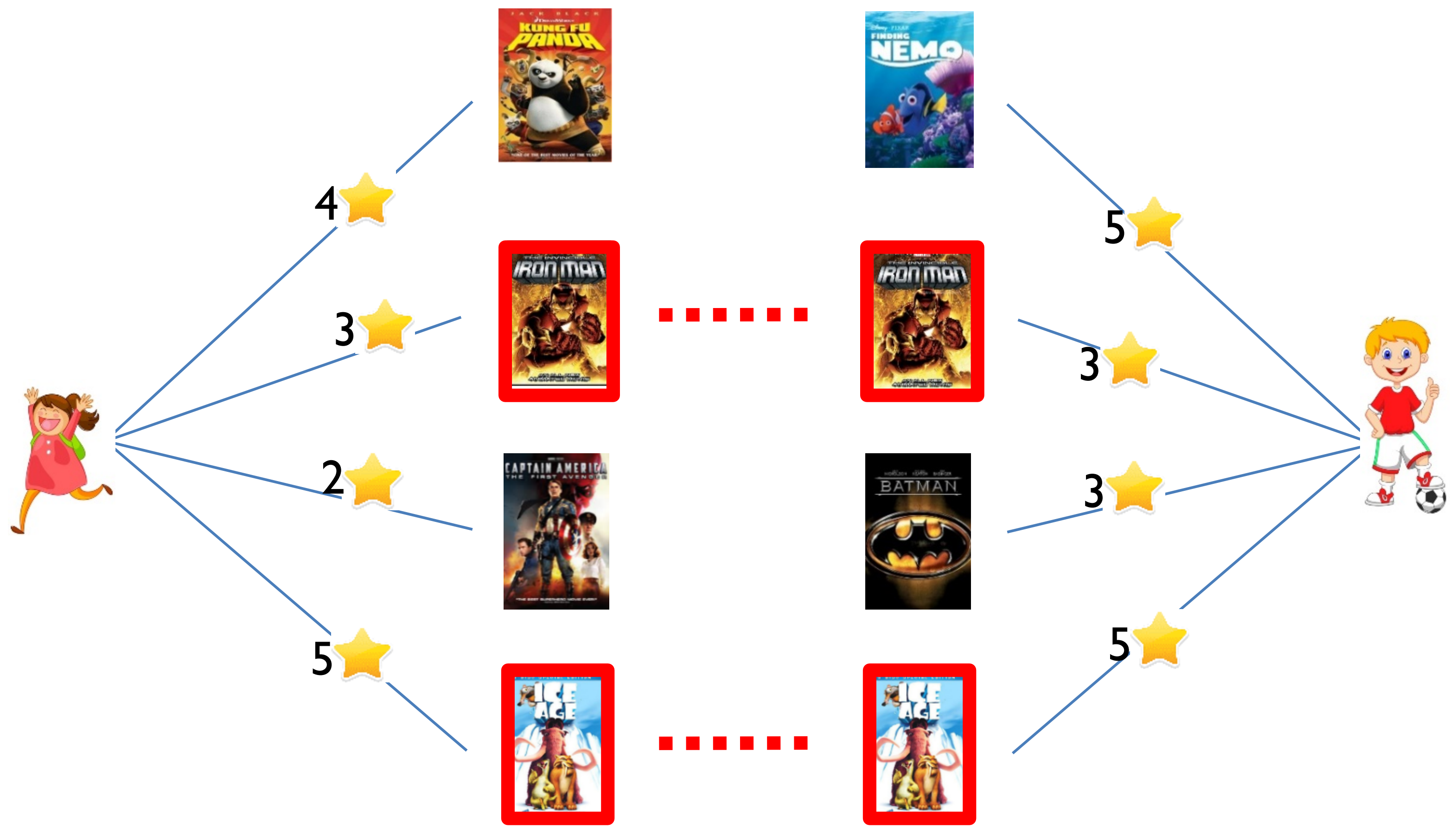
user-user CF:

$$\hat{A}_{ij} = A_{kl} + (A_{il} - A_{kl}) + (A_{kj} - A_{kl})$$

item-item CF:

$$\hat{A}_{ij} = A_{kl} + (A_{il} - A_{kl}) + (A_{kj} - A_{kl})$$

Taylor's Expansion



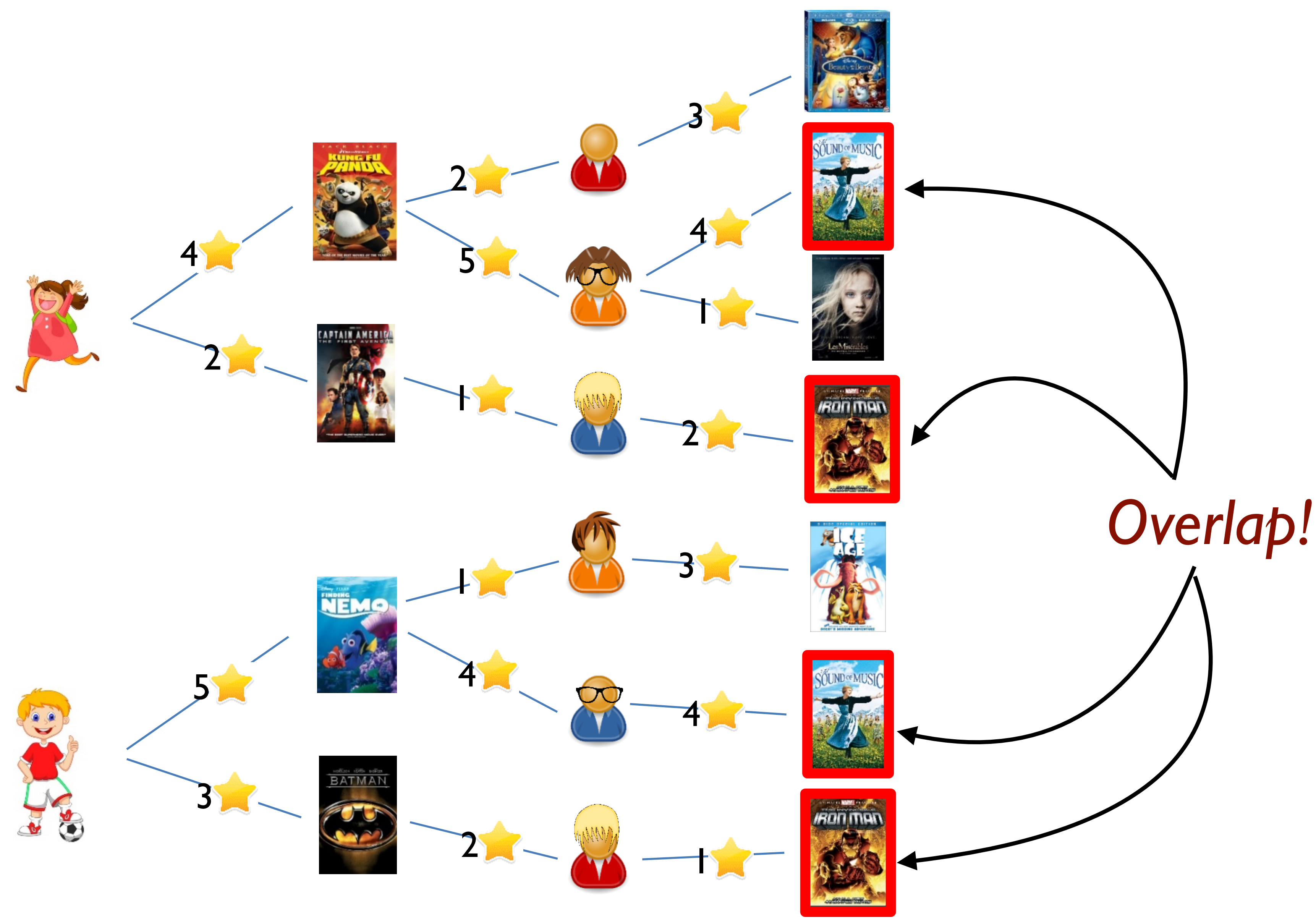
Computing similarity requires overlap

Birthday paradox leads to sample complexity $\tilde{\Omega}(n^{3/2})$

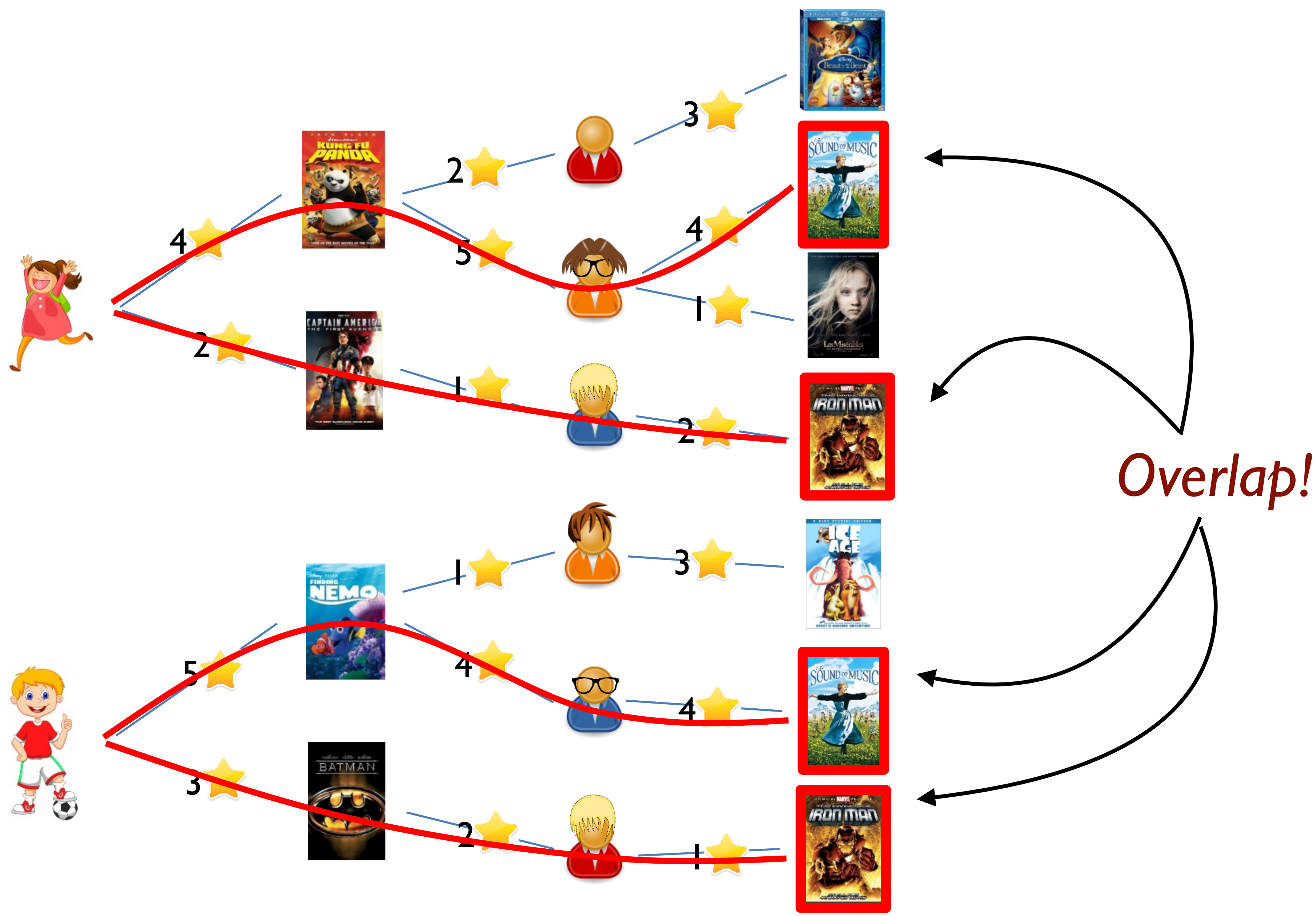
Does not work for *Sparser* setting

+ limited to additive noise model

Thy Friend is Mine

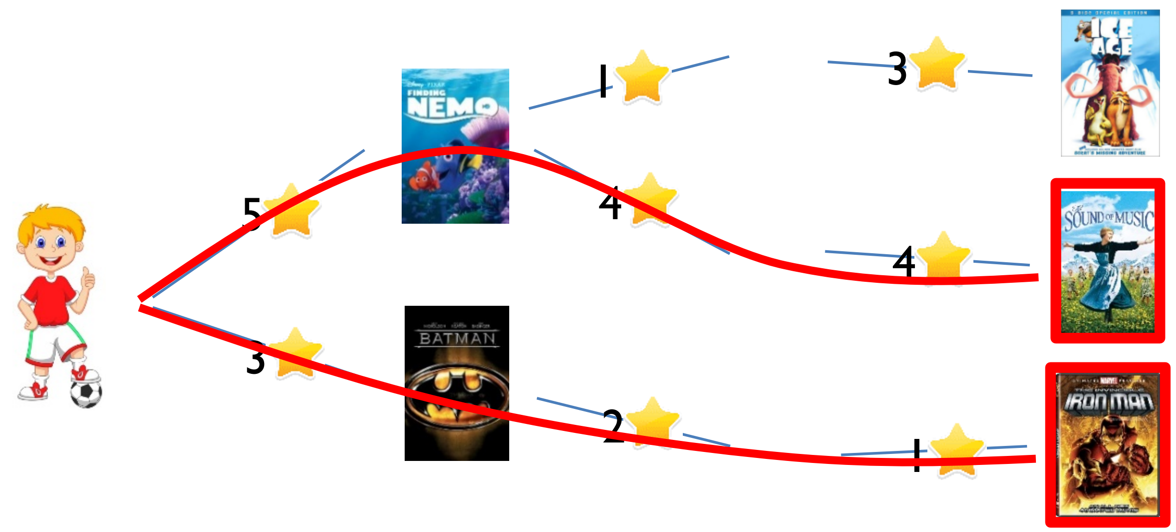


Thy Friend is Mine



Use product of ratings along path

Thy Friend is Mine



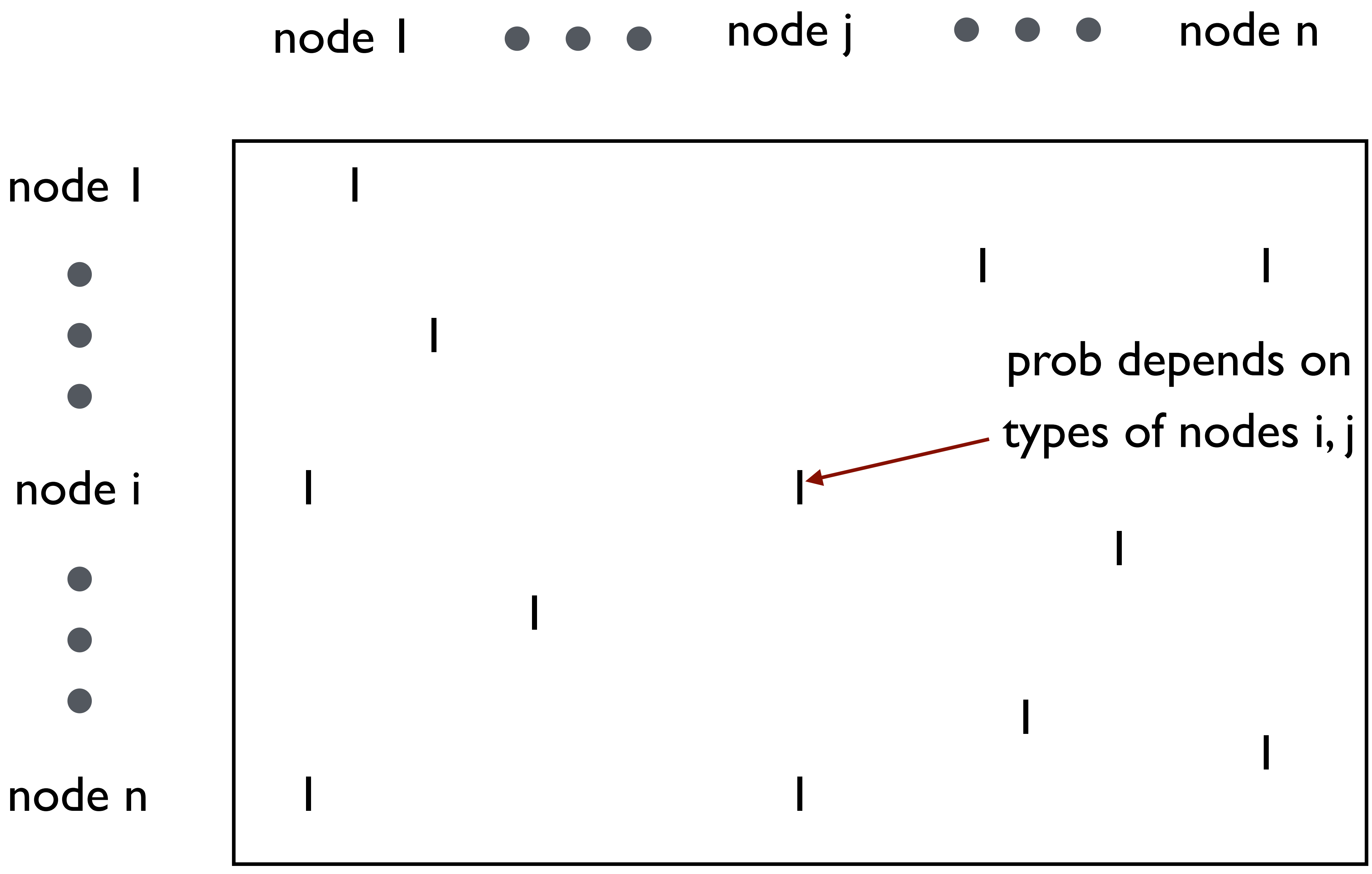
$$\mathbb{E}[Y] = U\Sigma V^T$$

$$Y \quad \bullet \quad Y^T \quad \bullet \quad Y \qquad \approx U\Sigma^3 V^T$$

Compare direct neighbors $\sim \|(u - u)\Sigma\|_2^2$

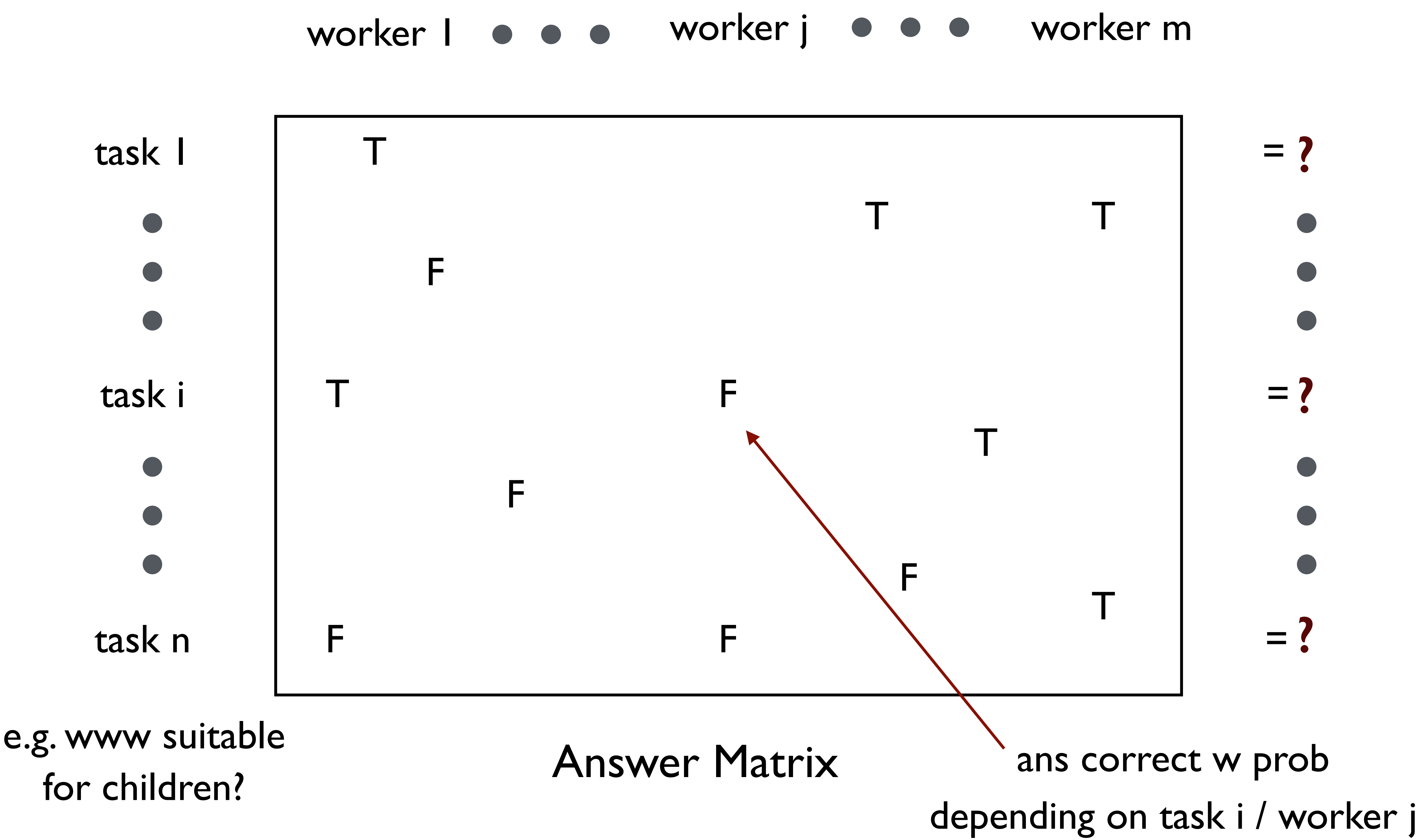
Compare r boundary neighbors $\sim \|(u - u)\Sigma^r\|_2^2$

Community detection, Graphon

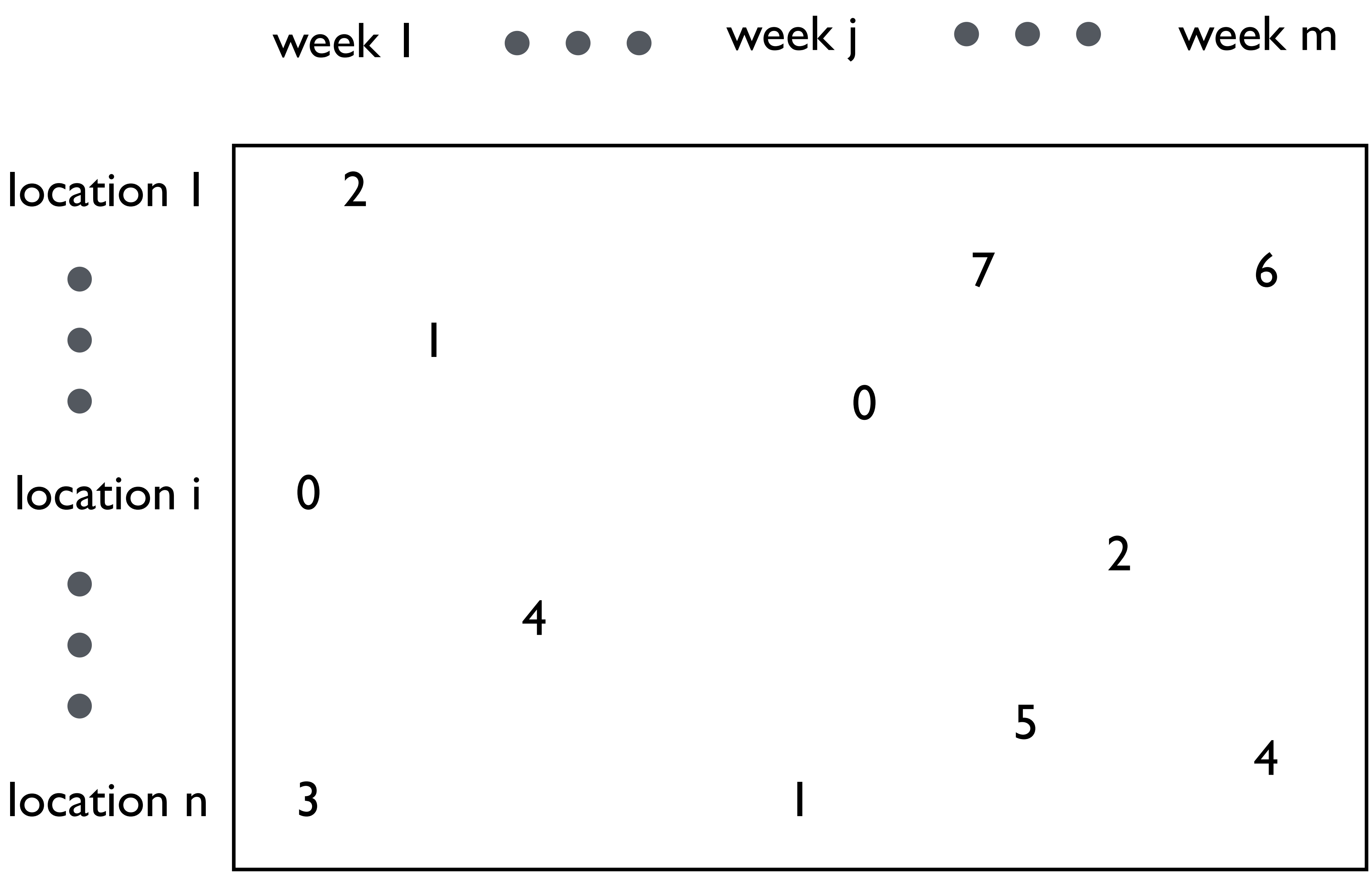


Adjacency Matrix

Low-Cost Crowd-Sourcing
(Generalized Dawid-Skene Model)



Censored Demand Prediction
(Hidden Markov Process with Censoring)



e.g. what rate umbrellas are being sold?

Censored Demand

Discussion: With Sample Data

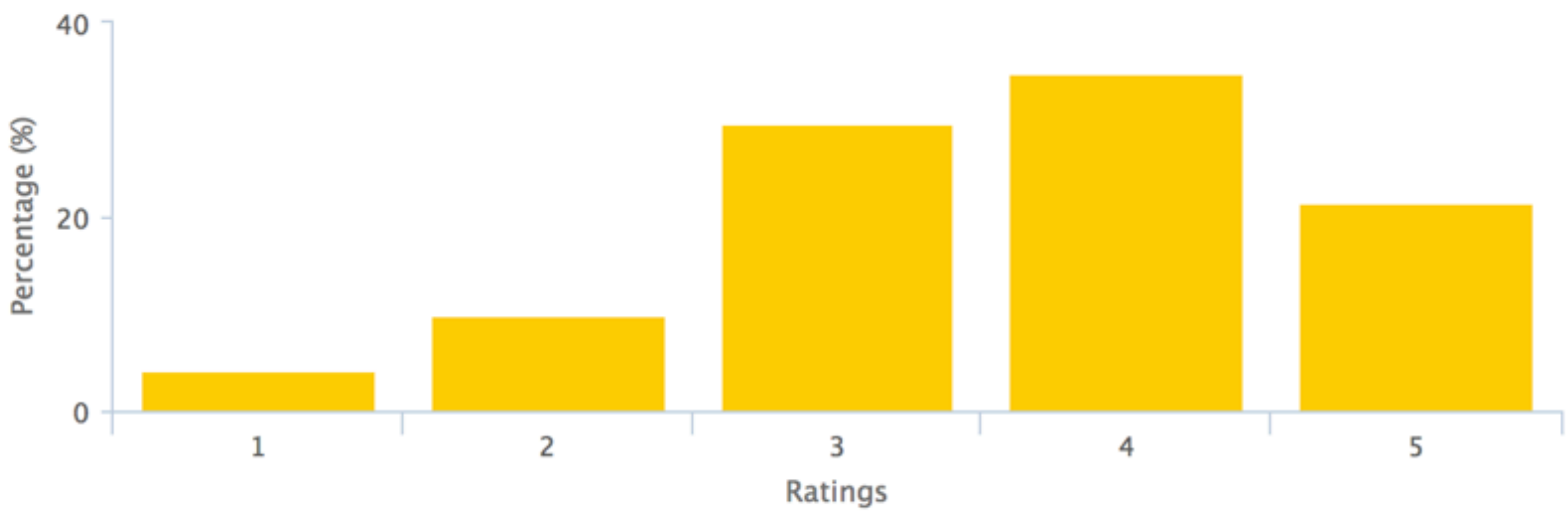
Dataset: MovieLens.

number of movies (m): 11000+

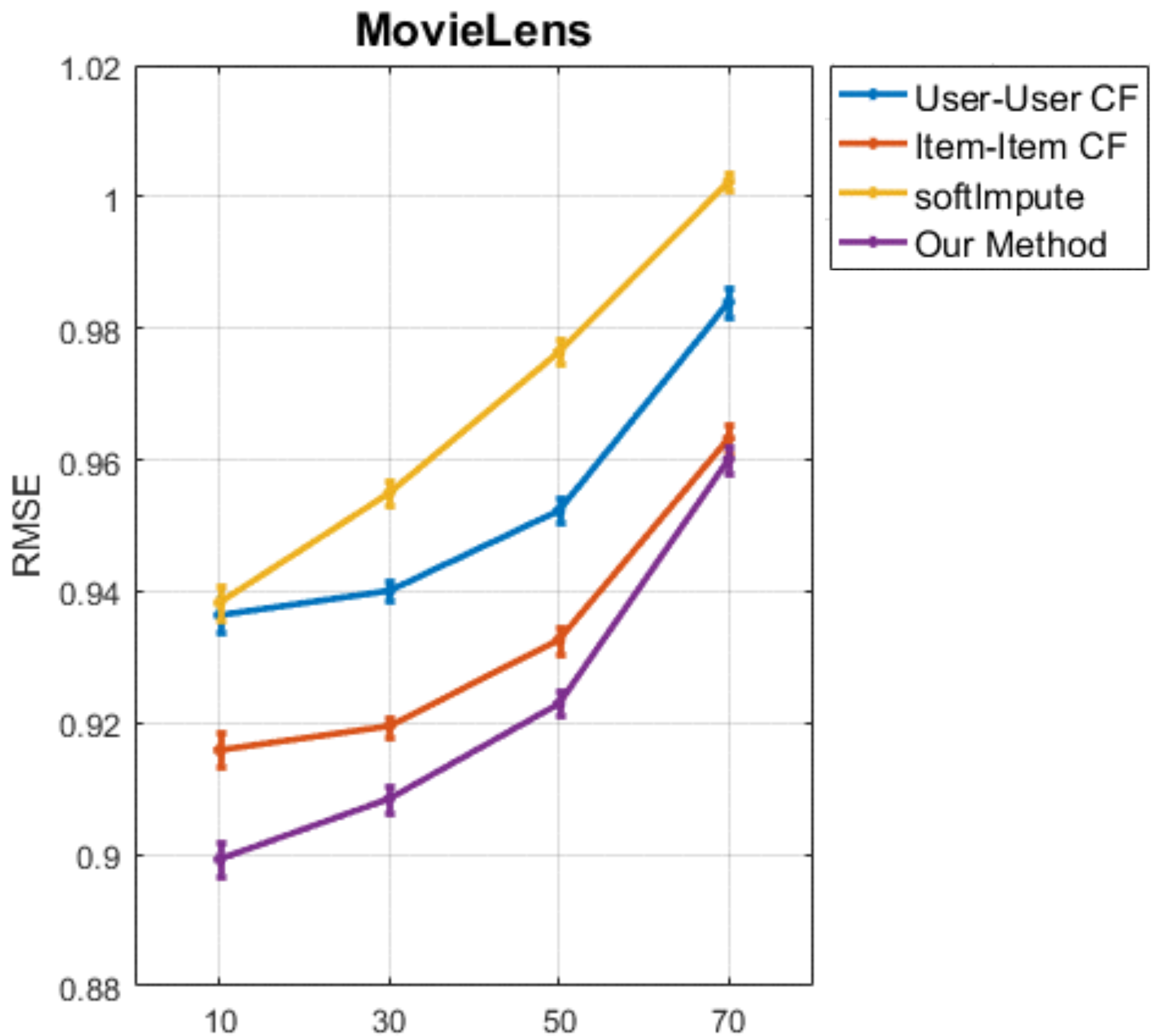
number of users (n): 30000+

average ratings per user: 3.6

around 3% of matrix is *filled*



Discussion: With Sample Data



percentage removed

Discussion: Beyond Matrices

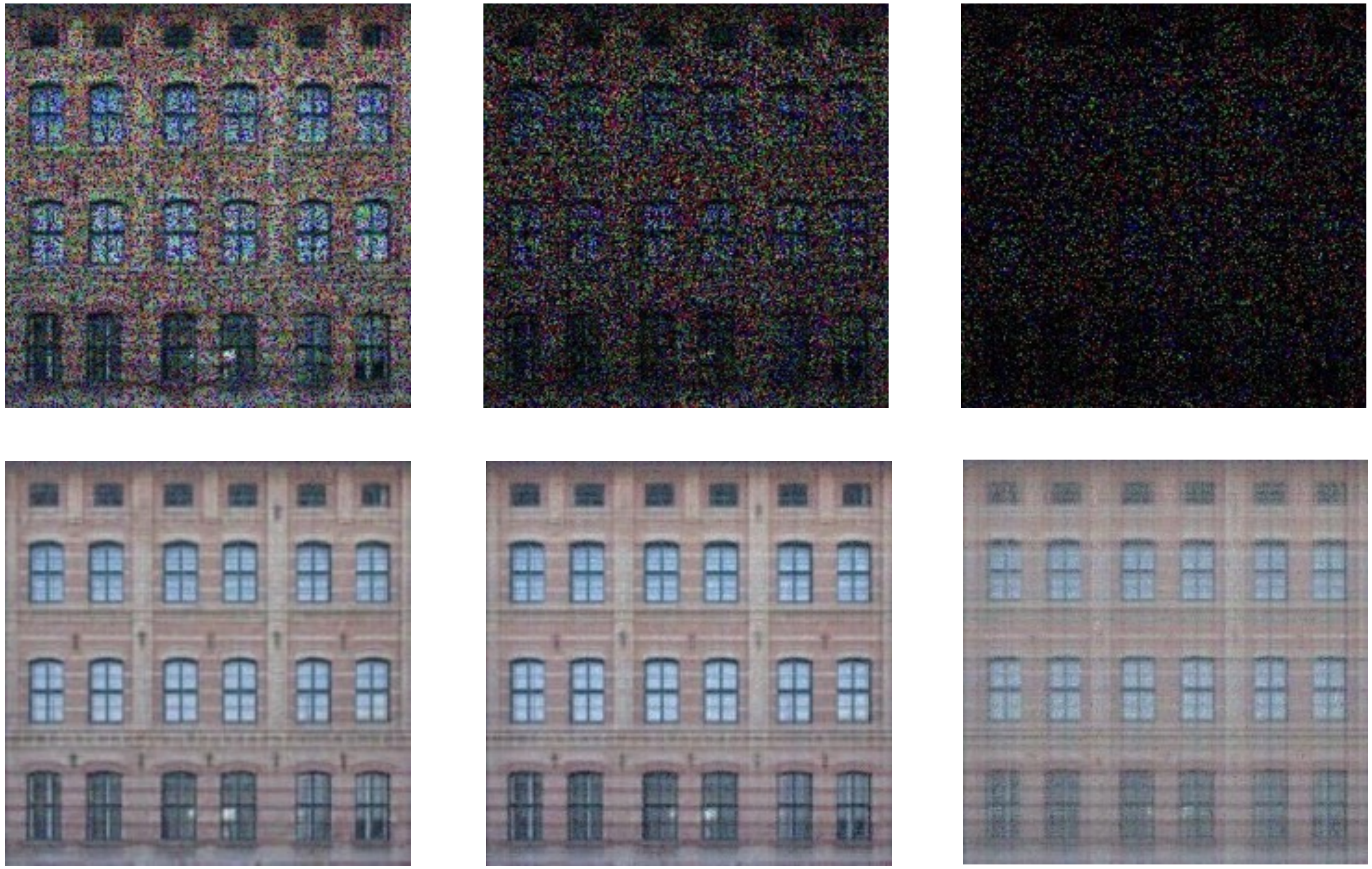
Image Data Set

3-order Tensor: rows x columns x RGB

Discussion: Beyond Matrices

actual

completion

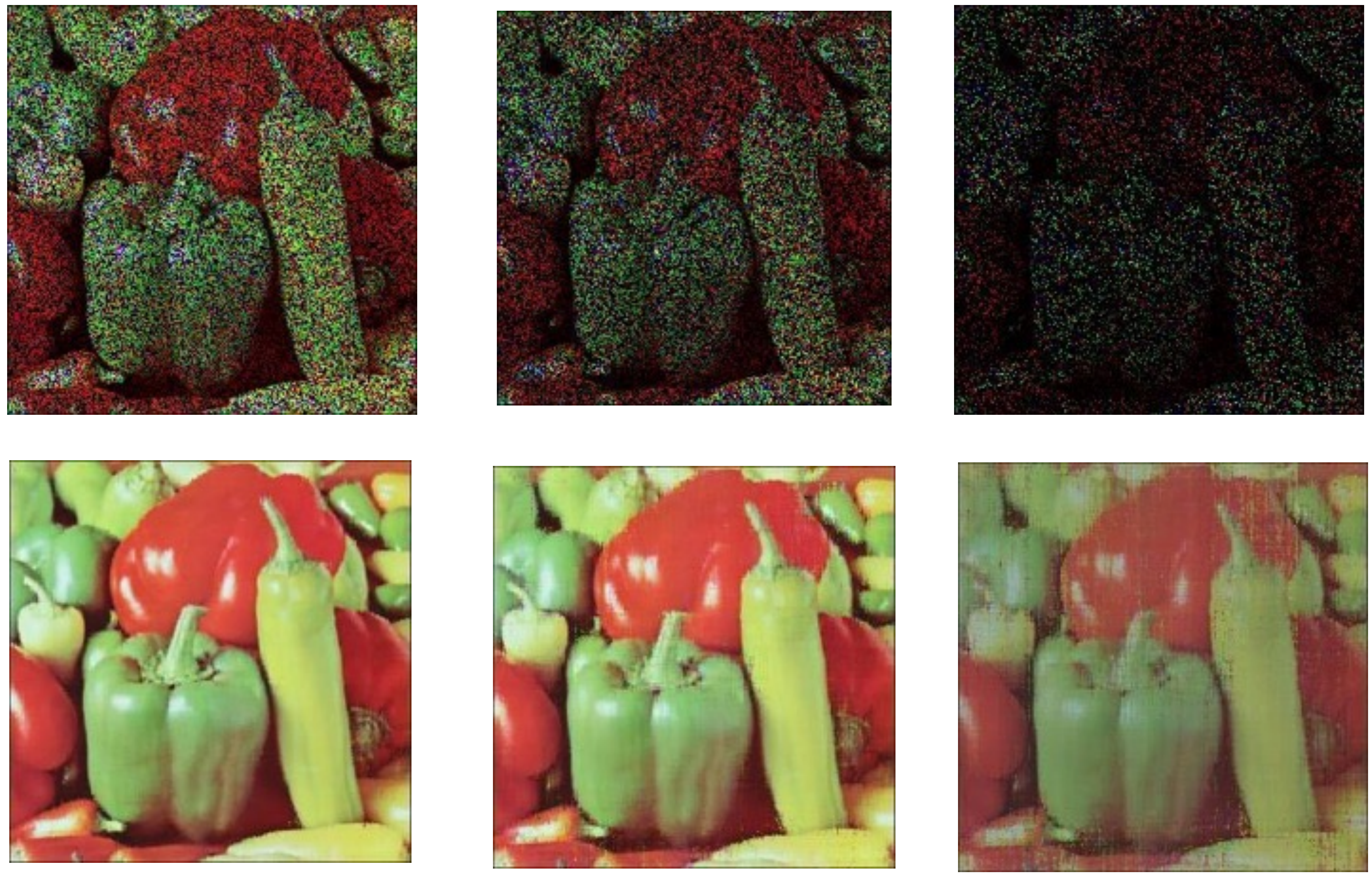


50% removed　　70% removed　　90% removed

Discussion: Beyond Matrices

actual

completion



50% removed

70% removed

90% removed

Discussion: Beyond Matrices

Image Data Set

		
ours RSE	0.086	0.1091
best-in-lit RSE	0.092	0.110

Readings

- D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," Commun. ACM, 1992
- Linden, G., Smith, B. and York, J. Amazon.Com Recommendations: Item-to Item Collaborative Filtering. IEEE Internet Computing, 2003.
- N. Srebro, N. Alon, and T. S. Jaakkola, "Generalization error bounds for collaborative prediction with low-rank matrices," in NIPS, 2004.
- Koren, Y. and Bell, R.. Advances in Collaborative Filtering. In Recommender Systems Handbook 145-186. Springer US, 2011.
- S. Chatterjee, "Matrix estimation by universal singular value thresholding," The Annals of Statistics, vol. 43, no. 1, pp. 177–214, 2015.
- Lee, C. E., Li, Y., Shah, D. and Song, D.. Blind Regression: Nonparametric Regression for Latent Variable Models via Collaborative Filtering. In NIPS 2016.
- Borgs, C., Chayes, J., Lee, C. E. and Shah, D. Thy Friend is My Friend: Iterative Collaborative Filtering for Sparse Matrix Estimation. In NIPS 2017.