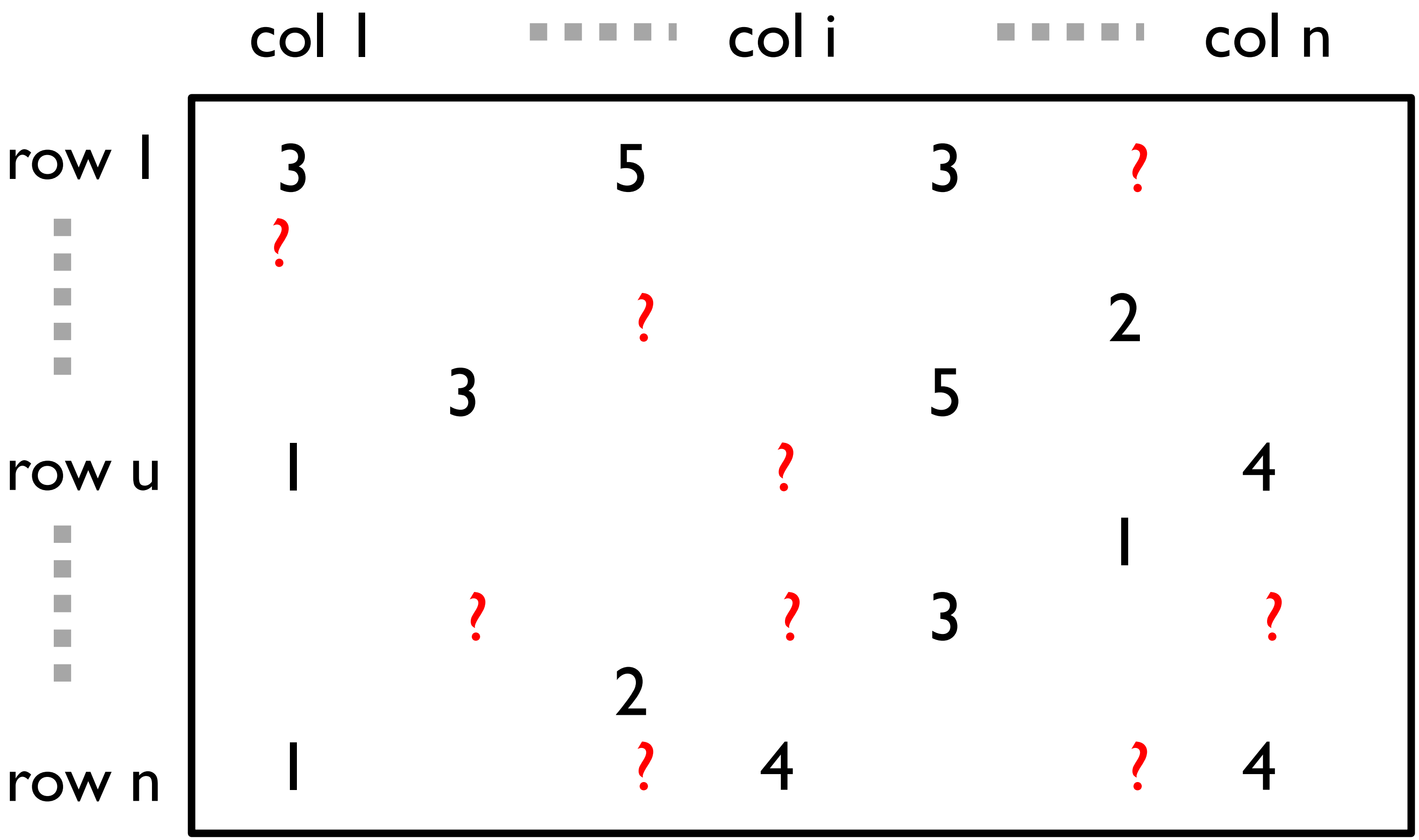
Iterative Collaborative Filtering for Sparse Matrix Estimation

Christina Lee
Microsoft Research New England

Christian Borgs (MSR), Jennifer Chayes (MSR), Devavrat Shah (MIT)

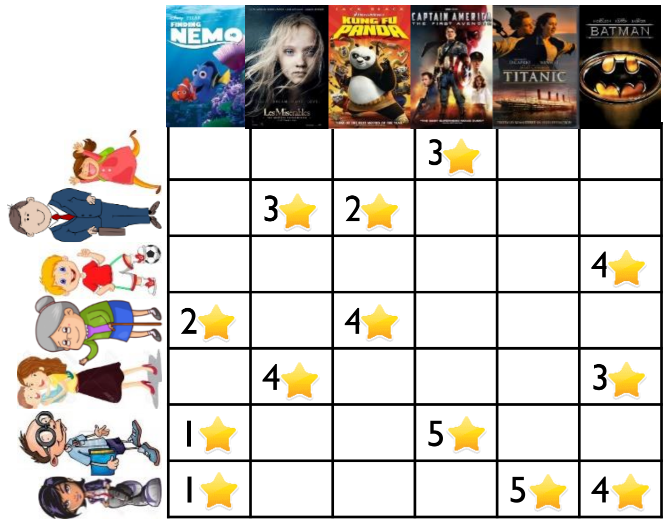
Matrix Estimation



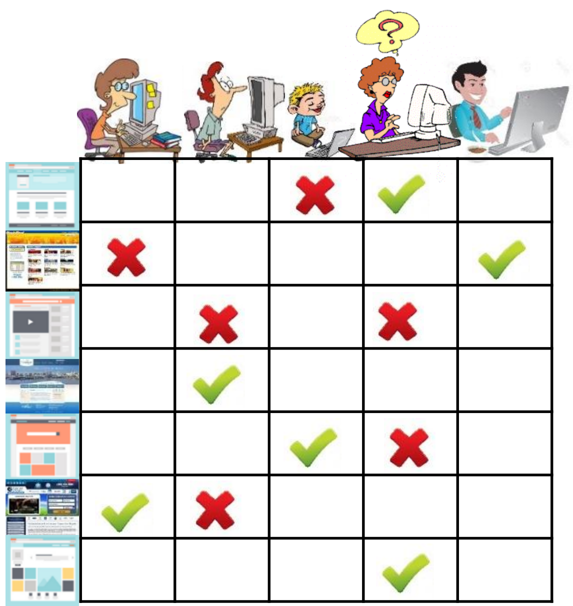
Matrix Estimation

- Observations
  - Ground truth matrix F we want to estimate
  - Obtain noisy observations $\{y_{ui}\}_{(u,i)\in E}$ for a subset E of entries subject to some noise model
$$\mathbb{E}[y_{ui}] = F_{ui}$$
- Goal
  - Produce an estimated matrix $\hat{F}$ so that the prediction error (e.g MSE) is small

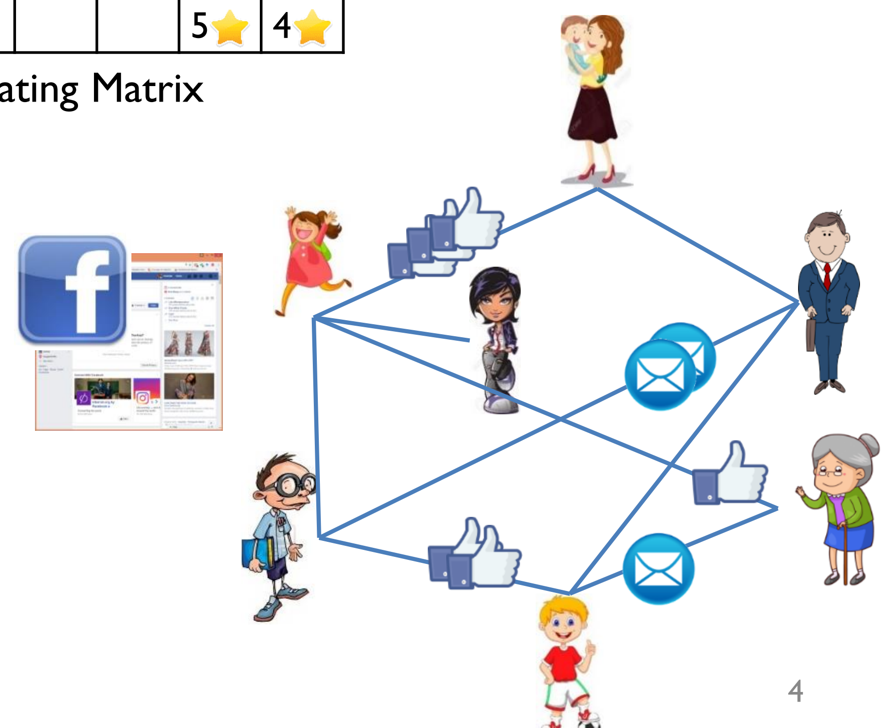
Applications



Rating Matrix



Response Matrix



Latent Variable Model

col 1: β_1 col i : β_i col n : β_n

row 1: α_1

row u: α_u

row n: α_n

$$\mathbb{E}[y_{ui}] = F_{ui} = f(\alpha_u, \beta_i)$$

Observe each entry
indep with prob p

Latent variables and entrywise noise assumed to be independent

Latent Variable Model

- Uniform observation sampling with probability p
- Row and column latent features sampled iid from compact bounded spaces, $\alpha_u \sim P_{X_1}, \beta_i \sim P_{X_2}$
- Mean of observed data entry described by latent fn f

$$\mathbb{E}[y_{ui}|\alpha_u, \beta_i] = f(\alpha_u, \beta_i) \in [0,1]$$

with bounded entries $y_{ui} \in [0,1]$

- Latent function has finite spectrum with rank d

$$f(\alpha_u, \beta_i) = \sum_{k=1}^{d} \lambda_k q_k(\alpha_u) q'_k(\beta_i)$$

Performance Metric

- Given partial observation of noisy matrix, produce an estimation matrix so the prediction error is small

$$\text{MSE} = \mathbb{E}\left[\frac{1}{nm}\sum_{ui}\left(\hat{F}_{ui} - f(\alpha_u, \beta_i)\right)^2\right]$$

- Minimize fraction p of matrix that needs to be observed (at random) to guarantee estimator is consistent

$$\lim_{n,m\to\infty} \text{MSE} = 0$$

Highlighted Results

(many remarkable results not reported here)

Paper	Sample Complexity	Noise Model	Function Class
Keshavan MontentariOh10	$\Omega(dn \max(\log n, d))$	Additive, iid Gaussian	Rank d
DavenportPlan BergWooters14	$\Omega(dn \max(\log n, d))$	Binary entries	Rank d
Chatterjee14	$\Omega(dn \log^6 n)$	Independent bounded	Rank d
XuMassoulie Lalarge14	$\Omega(n \log n)^*$	Binary entries	Rank d
AbbeSandon15	$\omega(n)^*$	Binary entries	piecewise constant (d blocks)
BorgsChayes LeeShah17	$\omega(d^5 n)$	Independent bounded	Rank d

*does not indicate dependence on d

Our Result [Borgs, Chayes, Lee, Shah '17]

Assume

- uniform sampling with prob p
- independent noise, bounded entries
- f has finite spectrum (rank d)

If $p = \omega\,(d^5 n^{-1})$,

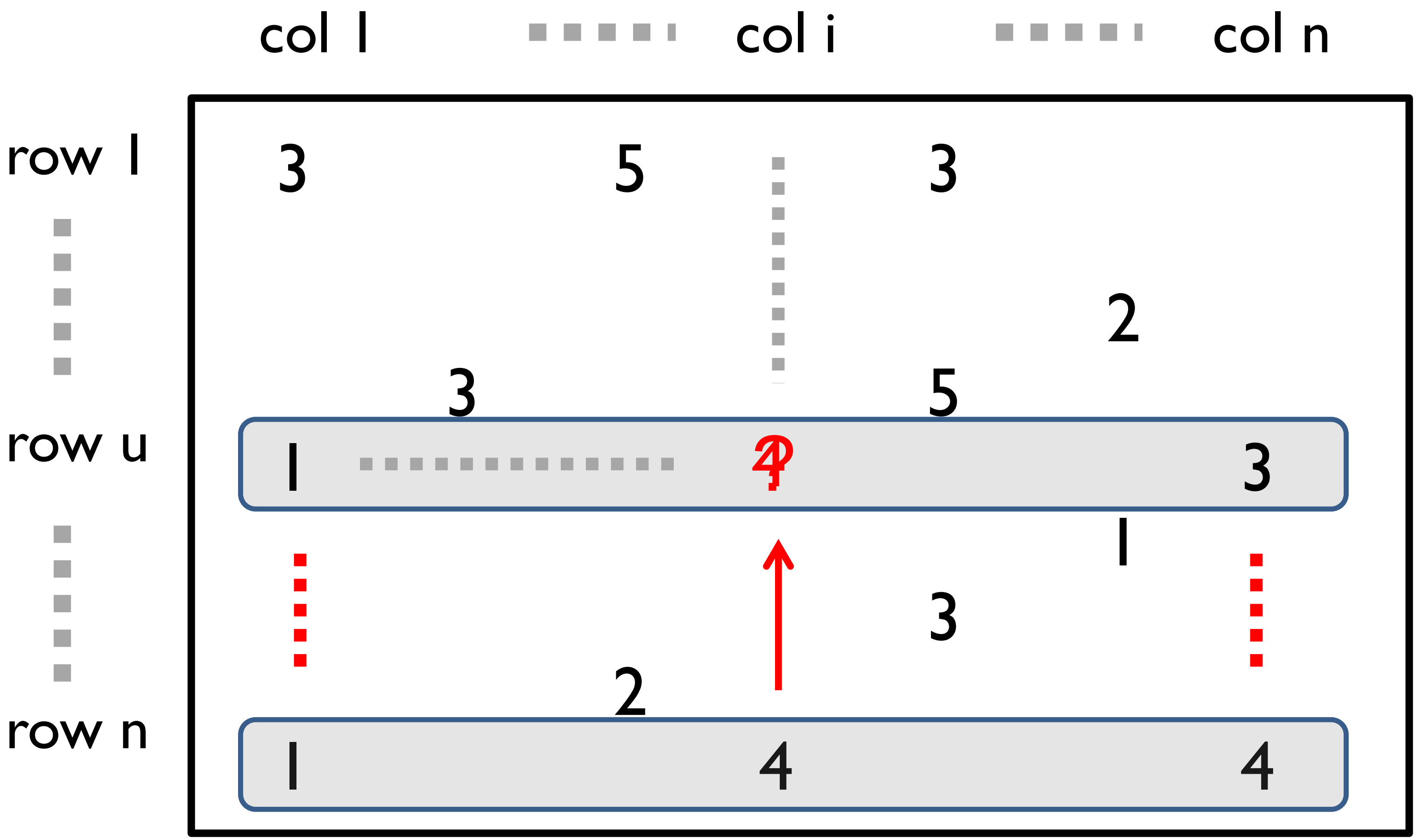
$$\text{MSE} = O\big(d^2 (pn)^{-2/5}\big) \to 0.$$

If $p = \omega(d^5 n^{-1} \ln^5(n))$, then with high probability,

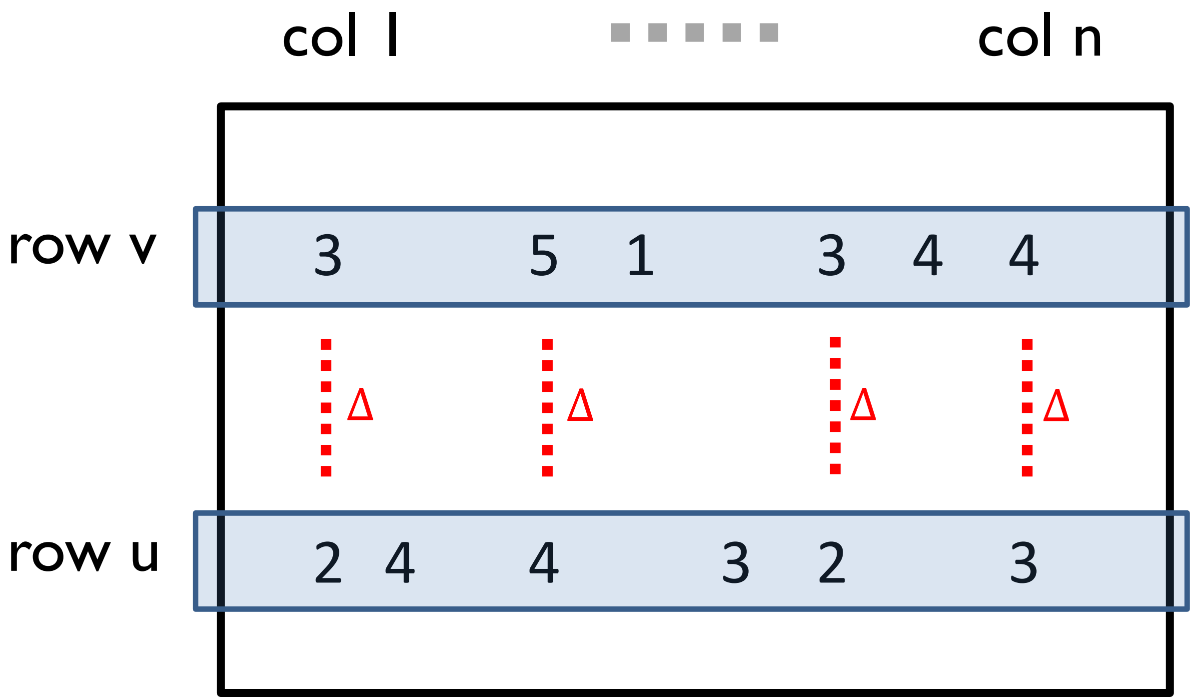
$$\max_{ij} \Big(\hat{F}_{ij} - f\big(\alpha_i, \alpha_j\big)\Big)^2 = O\big(d^2 (pn)^{-1/5}\big) \to 0.$$

Collaborative Filtering [Goldberg et. al. 92]

(user-user variant)

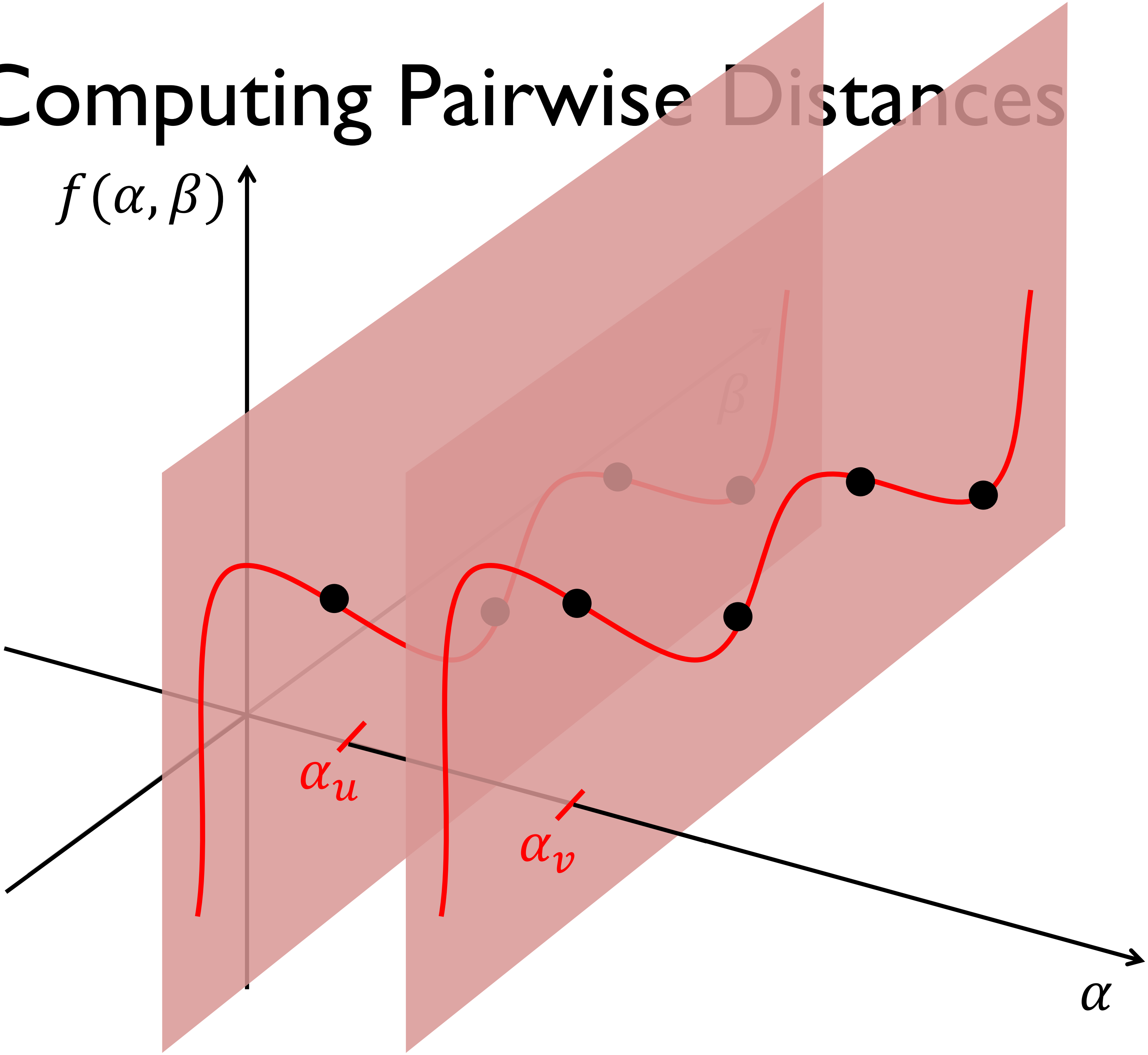


Computing Pairwise Distances



"estimated distances are finite sample estimates of L_2 distance between users' rating functions over movie space" [Lee-Li-Shah-Song NIPS 2016]

Computing Pairwise Distances



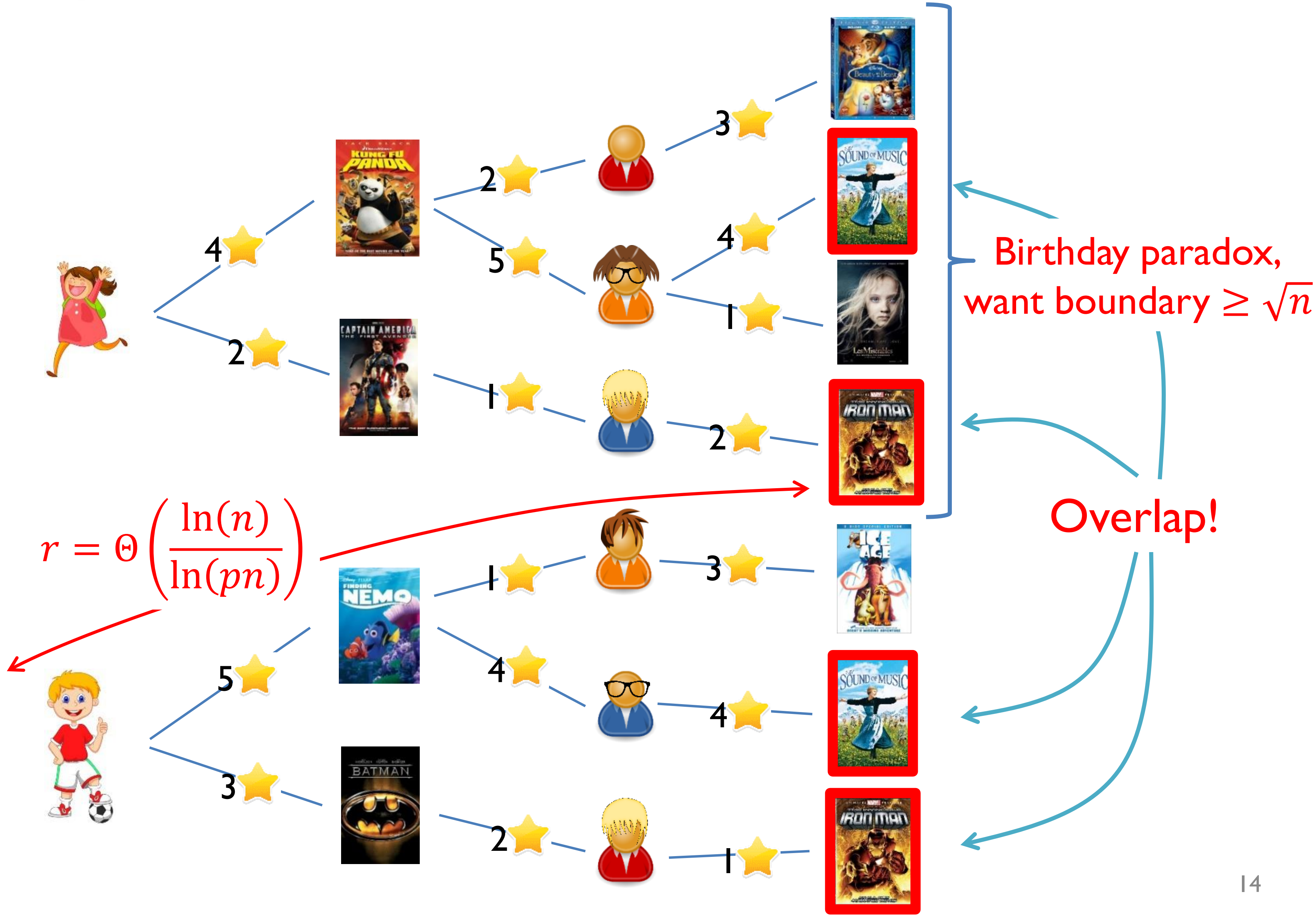
Requires Sample Complexity $\Omega(n^{3/2})$

Computing similarity requires overlap

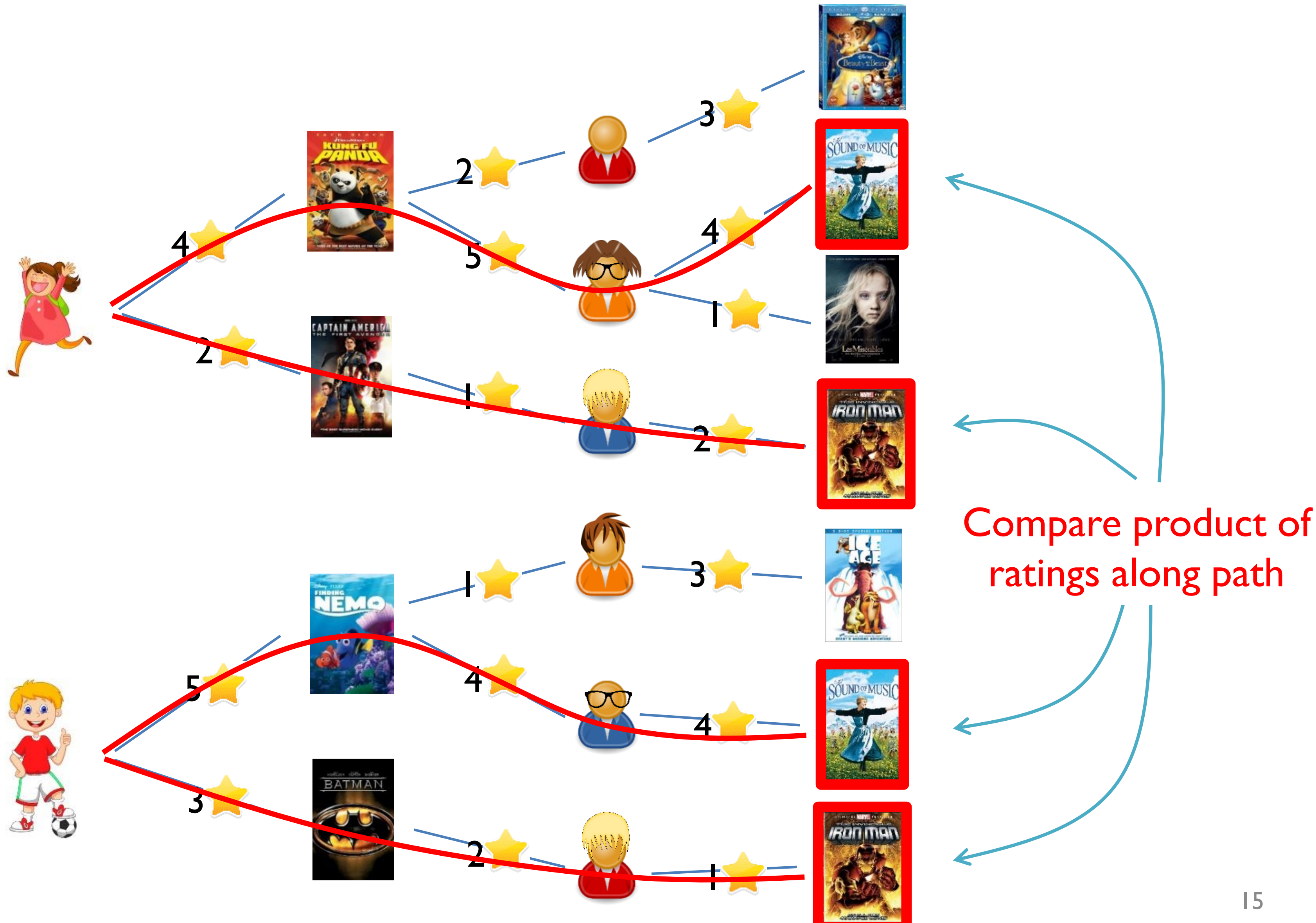
Birthday Paradox requires sample complexity $\Omega(n^{3/2})$

Does not work for sparser datasets!

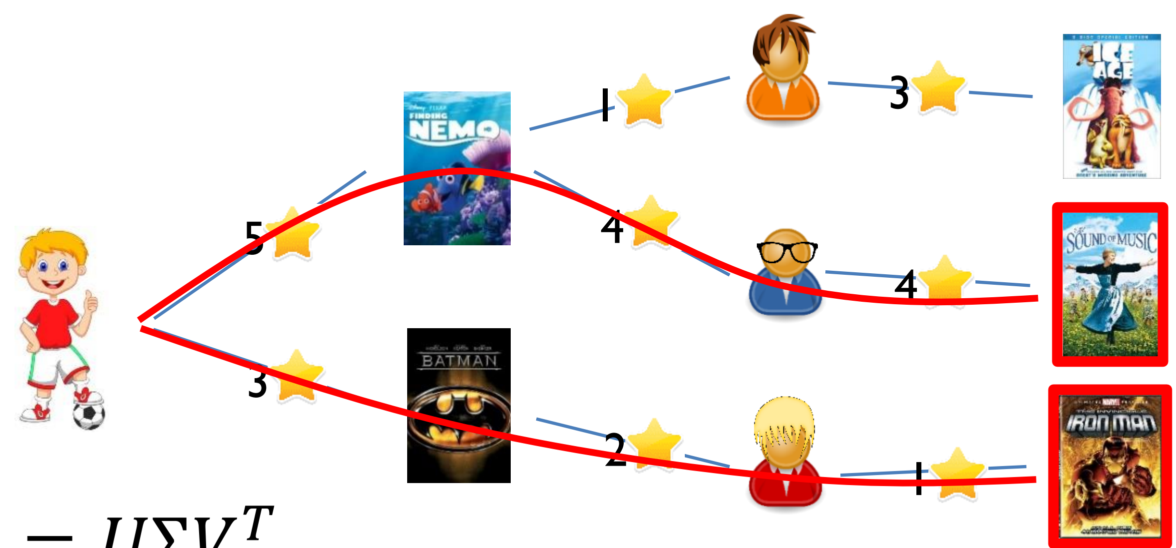
Expanding Neighborhood



Expanding Neighborhood



Expanding Neighborhood



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

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

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

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

Algorithm Summary

Step 1: Estimate distances by comparing product of weights along path to shared vertices in radius r neighborhoods, for $r = \Theta\left(\frac{\ln(n)}{\ln(pn)}\right)$

Step 2: Predict by averaging close neighbors

$$\hat{F}_{ui} = \frac{1}{Z} \sum_{(v,j)} y_{vj} \, \mathbb{I}(uv \text{ "close"}, ij \text{ "close"})$$

Finer details …

- Use sample splitting between algorithm steps
- Do not include loops in path (use breadth first tree)
- Reduce computation by using coarse clustering
- Converges for $p = \omega(n^{-1+\epsilon})$ with $\epsilon > 0$ because r constant, but modification needed for smaller p

Questions? Email celee@mit.edu