Statistical Methods for Large-Scale Personalization

> Design algorithms that effectively make use of collaborative effects across users to jointly learn optimal recommendation policies

Soumyabrata Pal

Arun Suggala

Karthikeyan Shanmugam

Prateek Jain

AISTATS' 23, ICLR' 23, NeurIPS' 23

Outline

01 Need for collaboration in Bandits

- ➢ Multi-User Multi-Armed Bandit Problem
- ➢ Solve optimization problems jointly

02 Collaborative Multi-Armed Bandits

- ➢ Greedy Algorithm
- ➢ LATTICE (Latent Bandits via Matrix Completion)

03 Simplified Practical Algorithm

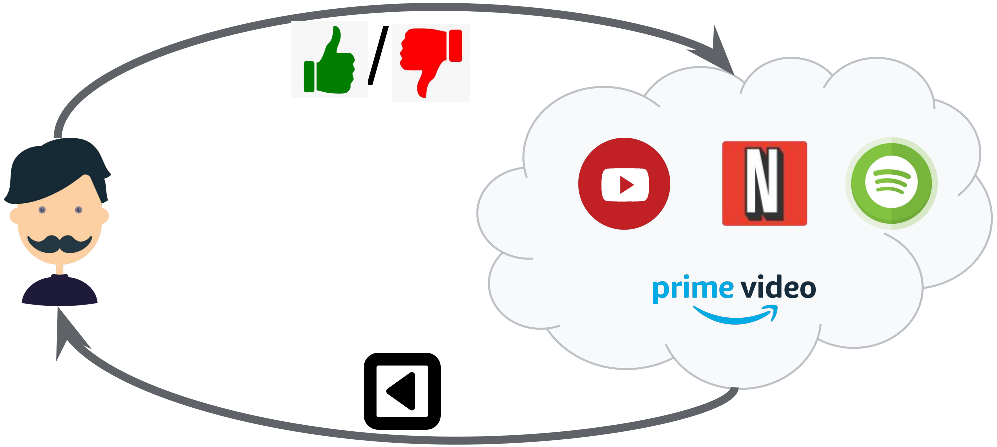
- ➢ Maternal Healthcare (ARMAAN)
- ➢ Great Empirical Promise

04 Conclusion

Bandit Optimization

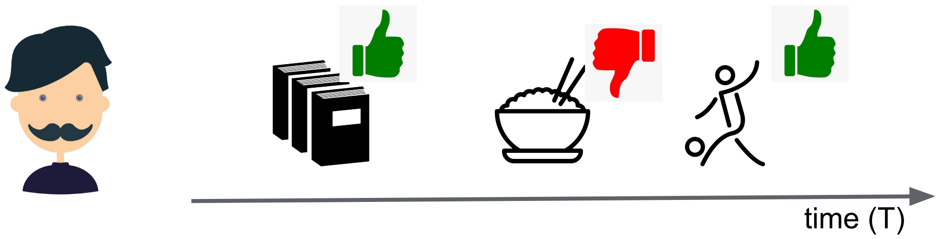
Decision making under uncertainty

- YouTube/Netflix : which video or movie to recommend?
- AI for Social Good: Healthcare, wildlife poaching



Multi-armed Bandit Optimization

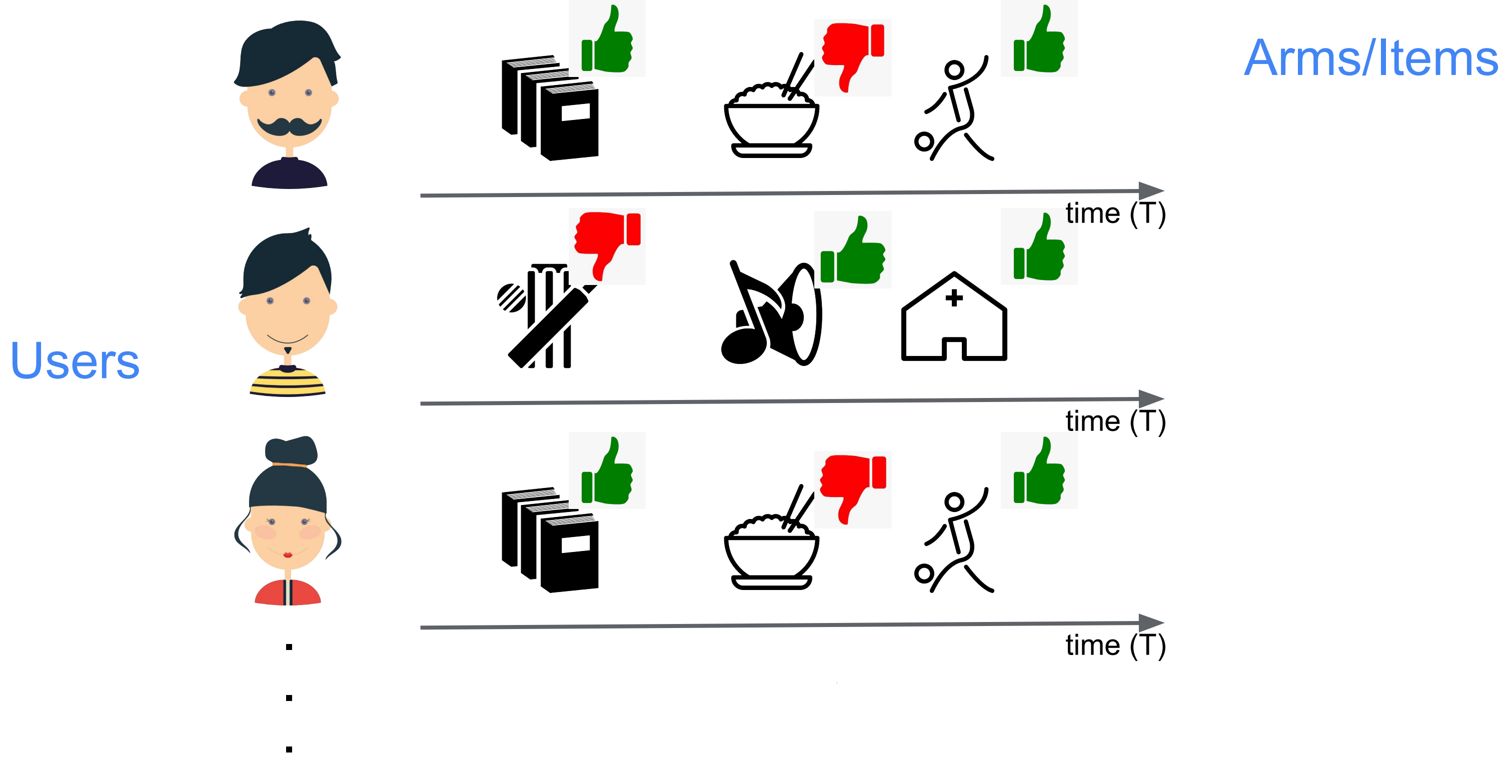
Items (Arms in Bandit literature)



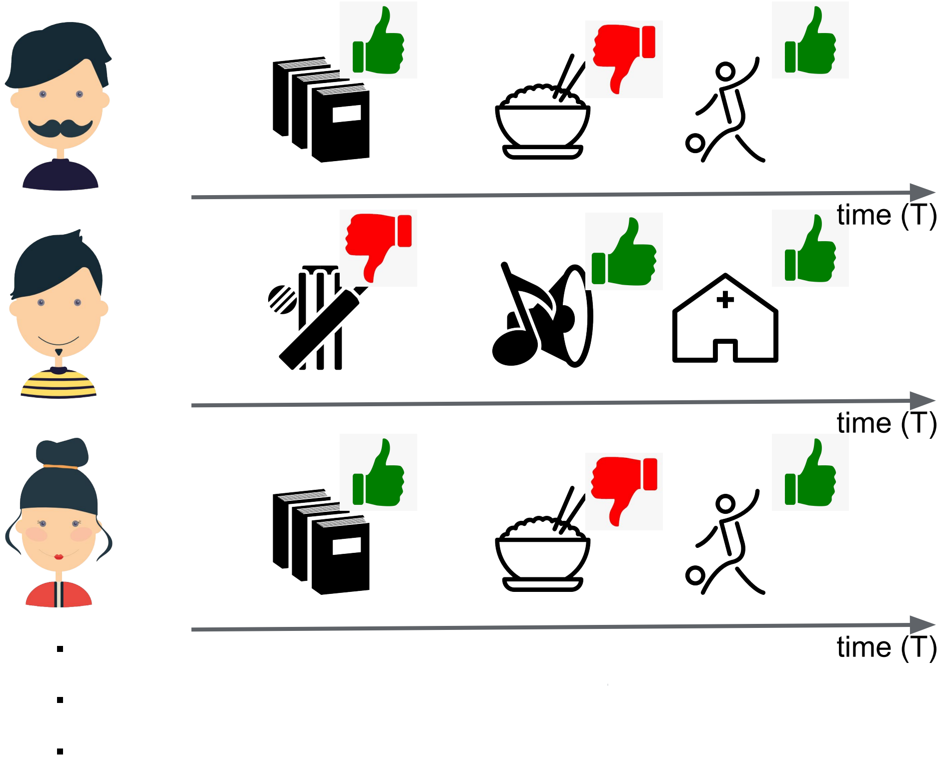
$f(\text{items})$ exploration

to learn user preferences well

Need for Collaboration in Multi-User MAB



Need for Collaboration in Multi-User MAB

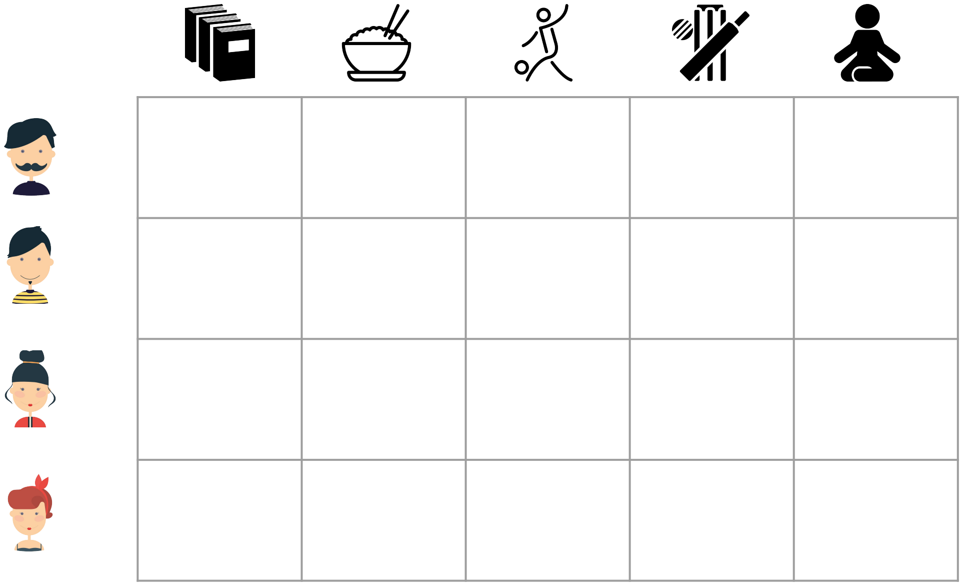


Infeasible with millions of items, users

$f(\text{items} \times \text{users})$

exploration to learn users preferences well **(separate optimization)**

Our Model (Cluster Structure)



True Reward Matrix (Unknown)

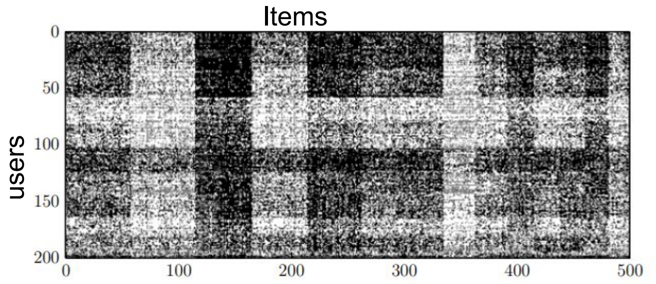
Our Model (Cluster Structure)

	0.9	0.8	0.7	0.6	0.5	Cluster 1
	0.9	0.8	0.7	0.6	0.5	
	0.1	0.2	0.4	0.6	0.9	Cluster 2
	0.1	0.2	0.4	0.6	0.9	

True Reward Matrix (Unknown)

Collaborative Bandits with Latent Clusters

	📚	🍚	⚽	🏏	🧘
👨	0.9	0.8	0.7	0.6	0.5
👦	0.9	0.8	0.7	0.6	0.5
👩	0.1	0.2	0.4	0.6	0.9
👧	0.1	0.4	0.8	0.4	0.1

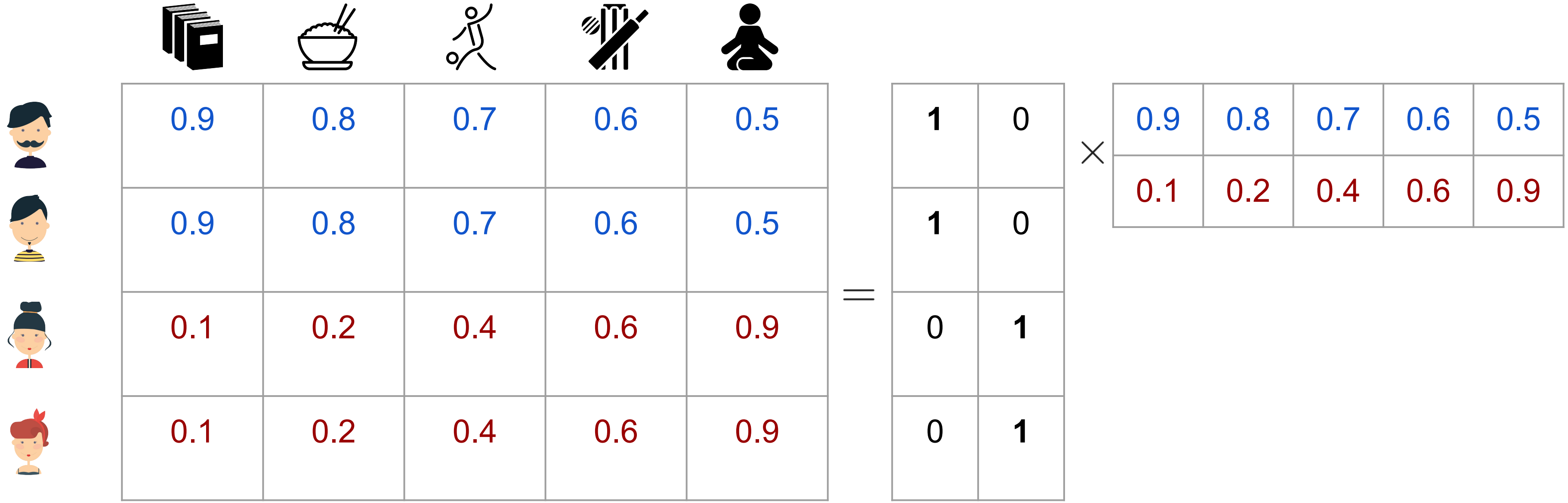


Movielens10M dataset: clusters among top users

Source: Bresler et al.

Assumption: Users can be grouped into **small number** of latent clusters

Our Model (Cluster Structure)



$$\begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.5 \\ 0.9 & 0.8 & 0.7 & 0.6 & 0.5 \\ 0.1 & 0.2 & 0.4 & 0.6 & 0.9 \\ 0.1 & 0.2 & 0.4 & 0.6 & 0.9 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & 0 \\ \mathbf{1} & 0 \\ 0 & \mathbf{1} \\ 0 & \mathbf{1} \end{bmatrix} \times \begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.5 \\ 0.1 & 0.2 & 0.4 & 0.6 & 0.9 \end{bmatrix}$$

True Reward Matrix (Unknown)

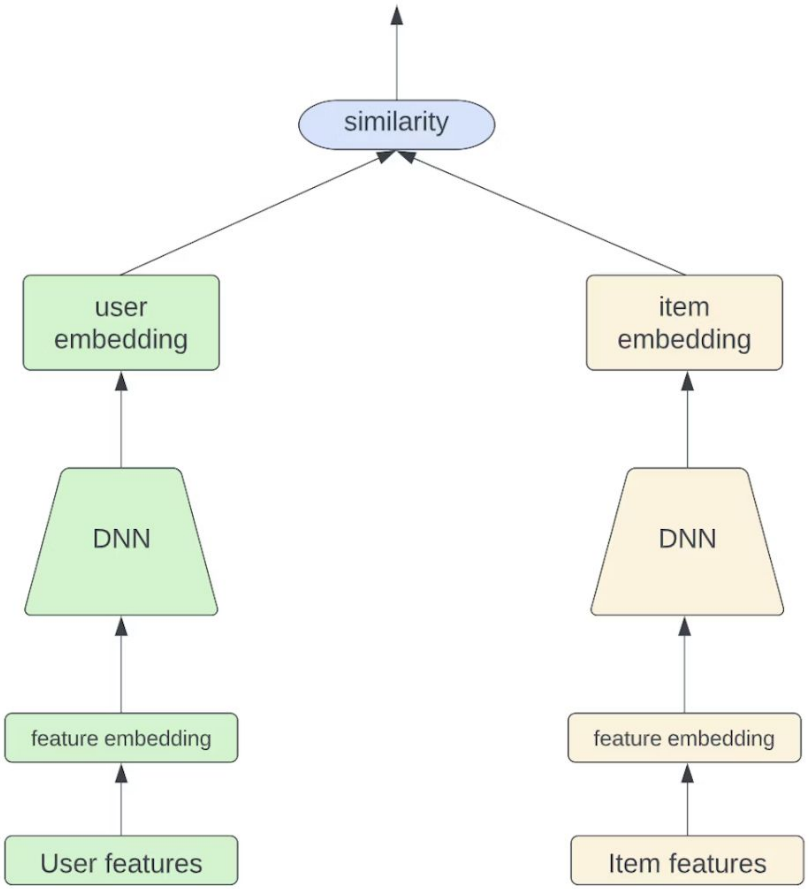
Learnable Parameters: #Users+ #Items#clusters

Rank < Clusters

Prior Work on Collaborative Bandits with Cluster Structure

	Setting	**Optimal Cost?**
Maillard & Mannor [2014]	Known cluster or known reward setting	✓
Gentile et al [2014, 2017]	Unknown clusters, rewards	✗
This Work (LATTICE)	Unknown clusters, rewards	✓

Enhancing popular Two-tower model



- Missing User/Item Features (Healthcare)
- Features corrupted/misleading/non-informative

Outline

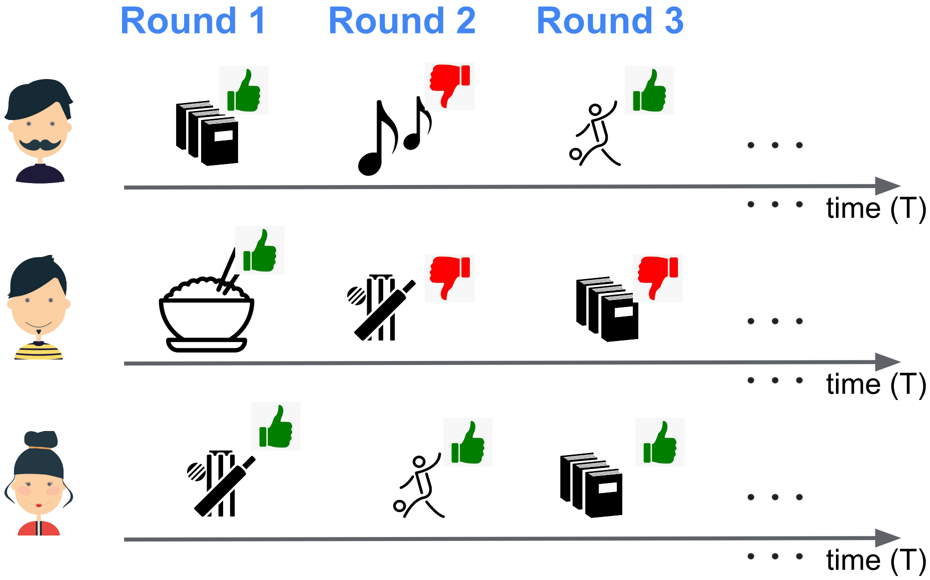
01 Need for collaboration in Bandits

02 Collaborative Multi-Armed Bandits

03 Parameter-free Practical Algorithm

04 Conclusion

Collaborative Bandits (M users, N items, T rounds)



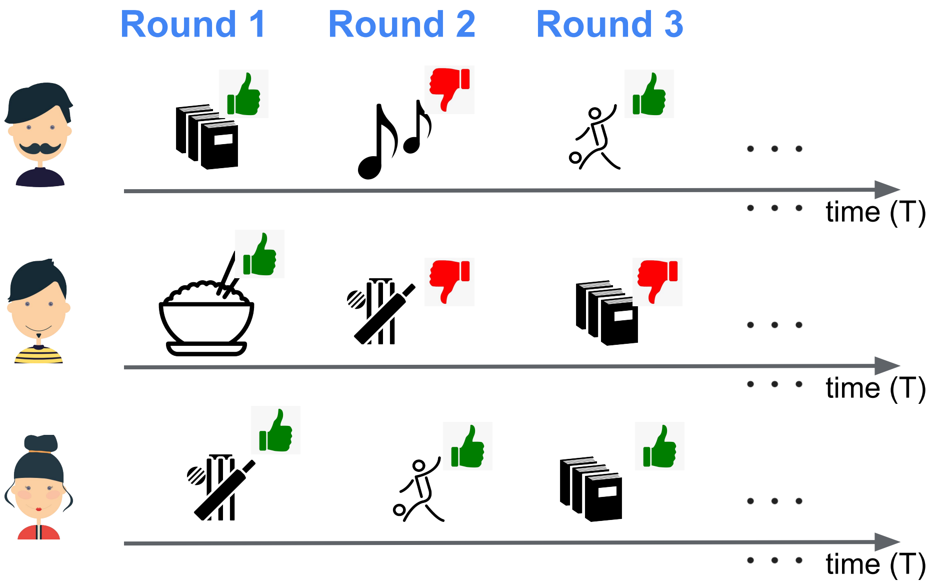
Corresponding entry of unknown true reward matrix

$$\mathbf{R}_{u\rho_u(t)} = \mathbf{P}_{u\rho_u(t)} + \text{Sub-Gaussian Noise}$$

Reward at round t for user u

$$\text{Regret} \triangleq \mathsf{M}^{-1} \sum_{u \in [\mathsf{M}]} \left(\mathsf{T} \max_{j \in [\mathsf{N}]} \mathbf{P}_{uj} - \mathbb{E} \sum_{t \in [\mathsf{T}]} \mathbf{P}_{u\rho_u(t)} \right)$$

Collaborative Bandits (M users, N items, T rounds)

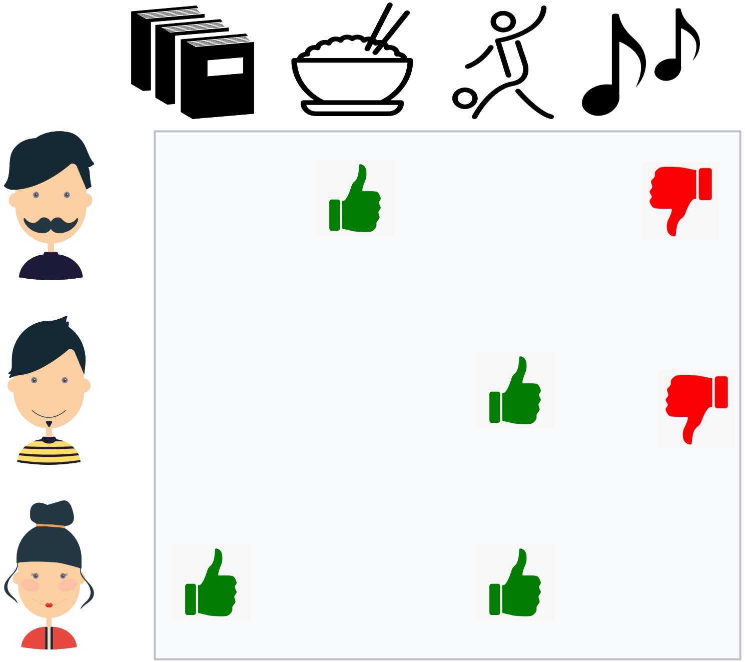


Reward accrued whenever user u is recommended item j **Observed** = *Entry of True Reward Matrix (user u and item j)* **Unknown** + *Sub-Gaussian Noise*

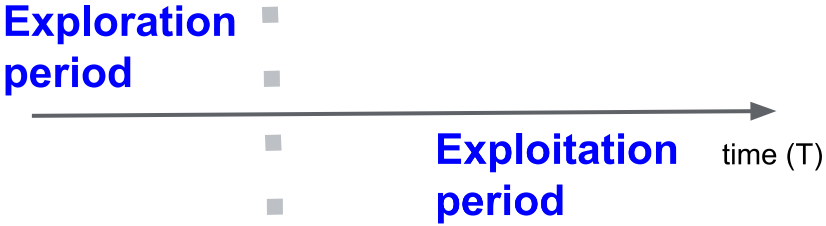
$$\text{Regret} \triangleq \mathsf{M}^{-1} \sum_{u \in [\mathsf{M}]} \Big(\mathsf{T} * \text{True reward of most rewarding item for user u}$$

$$- \mathbb{E} \sum_{t \in [\mathsf{T}]} \text{True reward of item recommended to user u at round t} \Big)$$

Greedy Algorithm (Exploration)



Exploration period
(Gather data randomly)

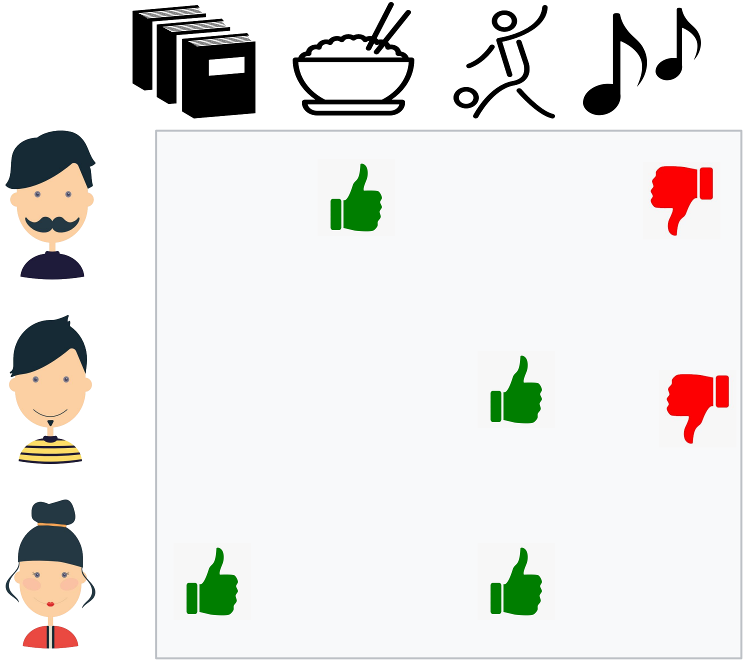


Exploration period

Exploitation period

time (T)

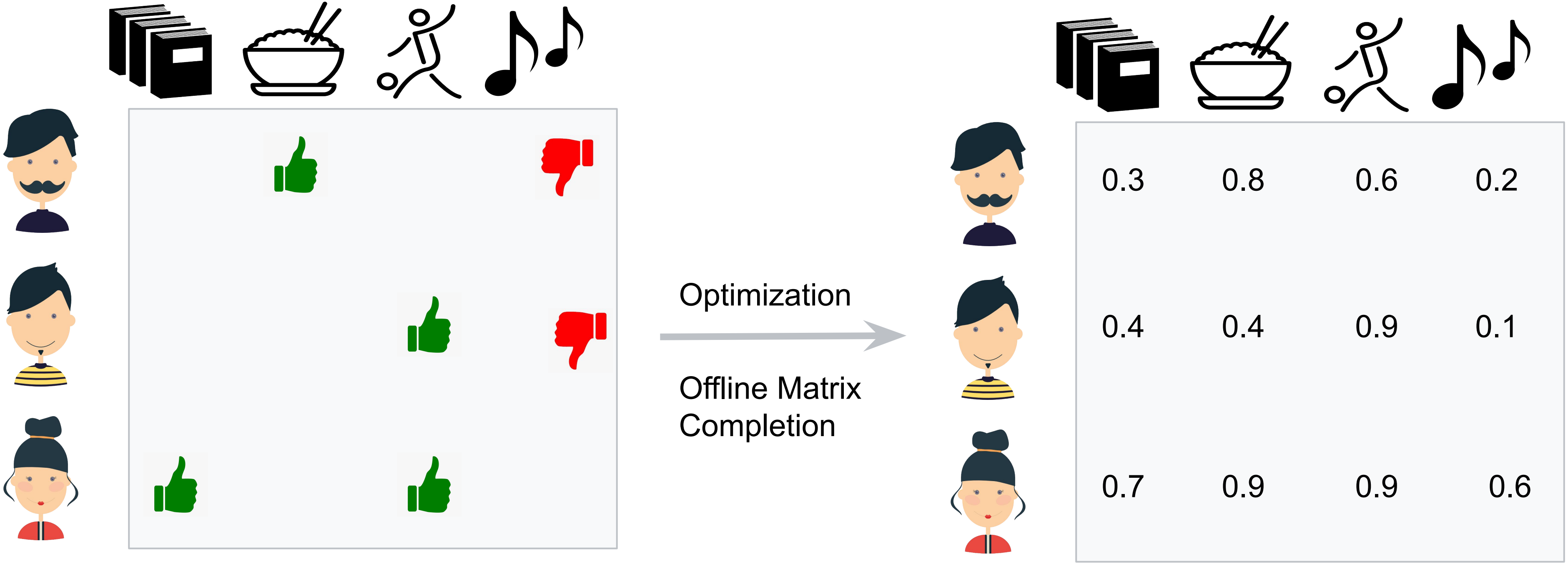
Greedy Algorithm



Low Rank Expected Reward Matrix

Can we estimate the entire matrix from a few observations?

Greedy Algorithm (Exploration)



	👍		👎
		👍	👎
👍		👍	

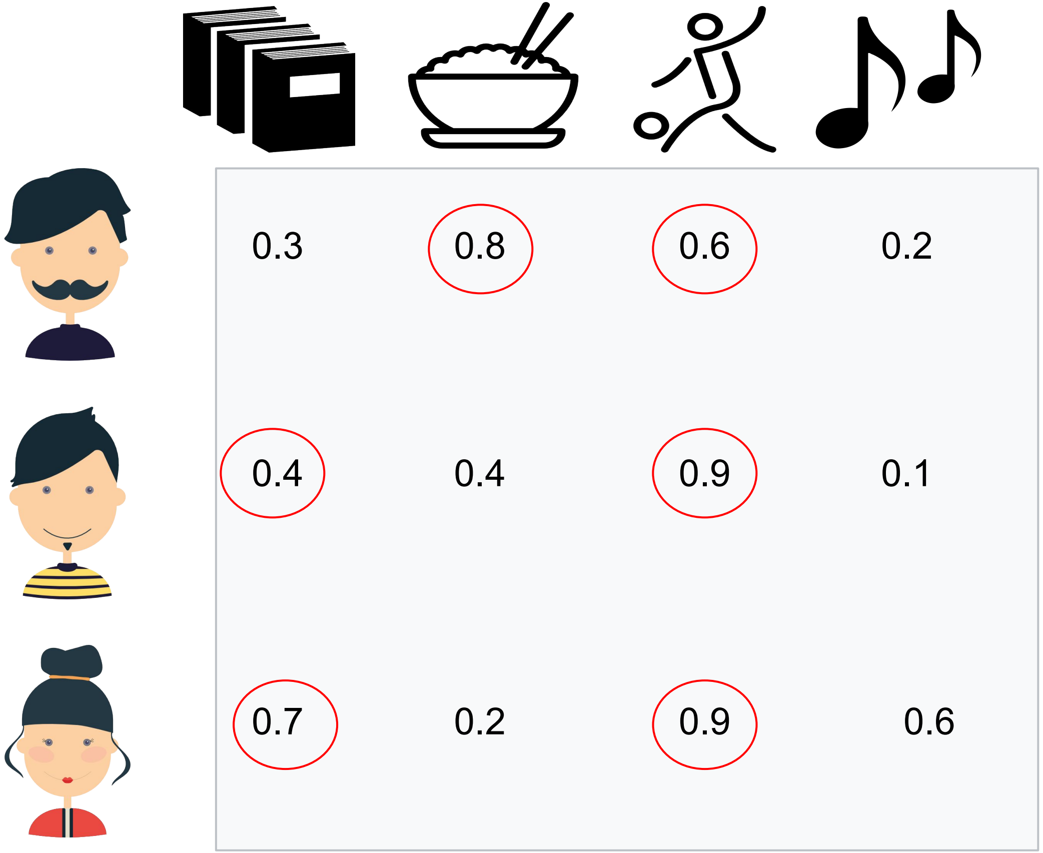
Optimization

Offline Matrix Completion

0.3	0.8	0.6	0.2
0.4	0.4	0.9	0.1
0.7	0.9	0.9	0.6

Low Rank Expected Reward Matrix

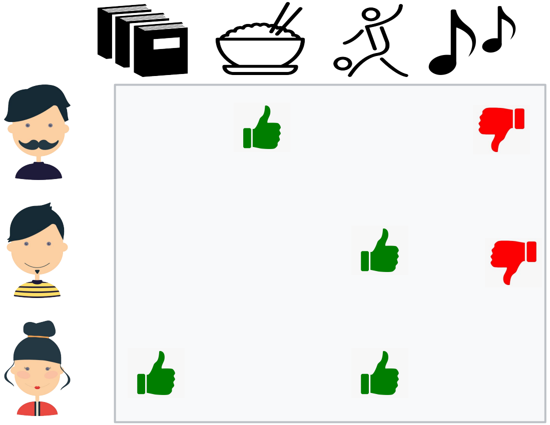
Greedy Algorithm (Exploitation)



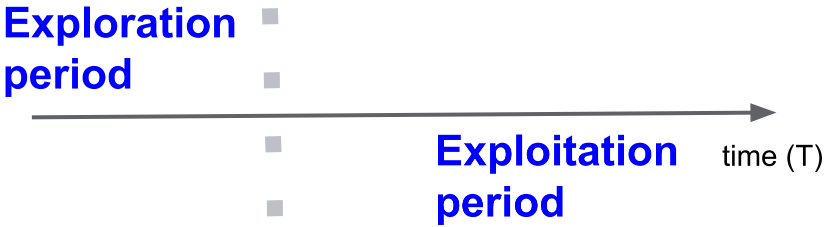
📚	🍚	⚽	🎵
0.3	[✓ 0.8]	[✓ 0.6]	0.2
[✓ 0.4]	0.4	[✓ 0.9]	0.1
[✓ 0.7]	0.2	[✓ 0.9]	0.6

Recommend items with highest estimated probability of being liked

Greedy Algorithm



Exploration

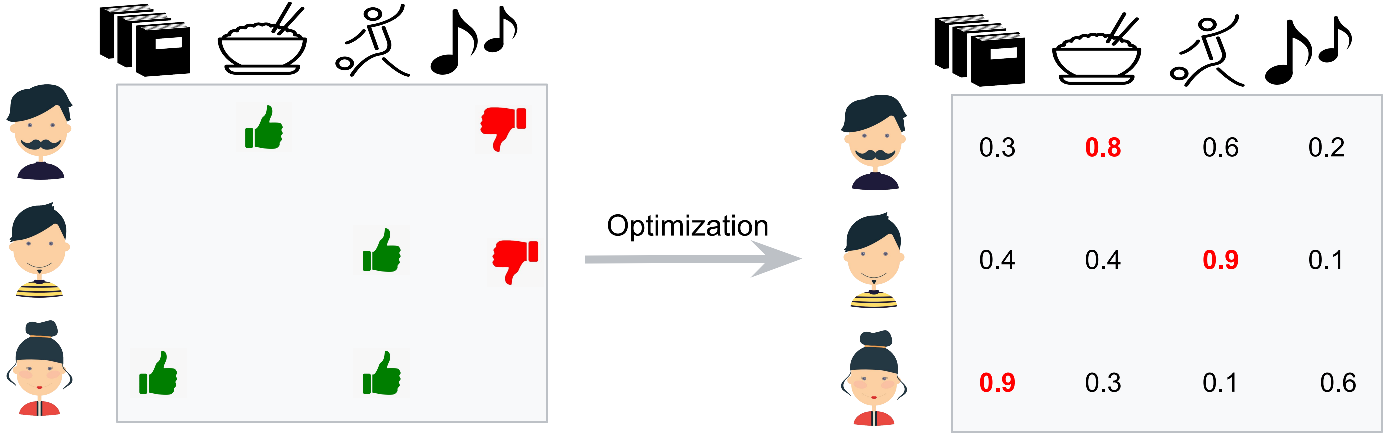


Exploration period

Exploitation period

time (T)

Greedy Algorithm



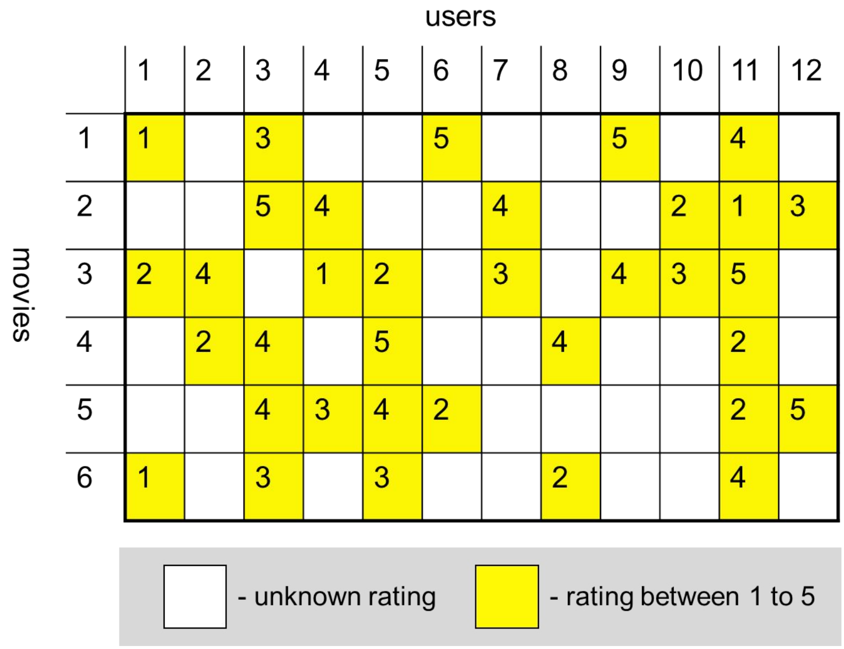
	👍		👎
		👍	👎
👍		👍	

0.3	**0.8**	0.6	0.2
0.4	0.4	**0.9**	0.1
0.9	0.3	0.1	0.6

Exploration

Matrix Completion

Low-rank Matrix Completion



users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

□ - unknown rating 🟨 - rating between 1 to 5

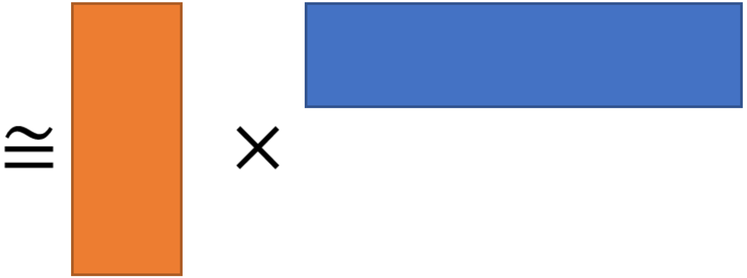
- Task: Complete ratings matrix
- Applications: recommendation systems, PCA with missing entries

Low-rank

users

movies	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3			5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

□ - unknown rating 🟨 - rating between 1 to 5



$$M \cong U \times V^T$$

- M: characterized by U, V
- No. of variables:
  - U: $m \times k = mk$
  - V: $n \times k = nk$
- DoF: $mk + nk$

Low-rank Matrix Completion

- $$\min_X \ Error_\Omega(X) = \sum_{(i,j)\in\Omega} \left(X_{ij} - M_{ij}\right)^2$$
  $$s.t \qquad \mathbf{rank}(X) \leq k$$
- Ω: set of known entries
- Problem is NP-hard in general
  - Two approaches:
    - Relax rank function to its convex surrogate (Trace-norm based method)
    - Use alternating minimization

Existing method: Trace-norm minimization

$$\min_{X} \sum_{(i,j)\in\Omega} \left(X_{ij} - M_{ij}\right)^2$$

$$s.t. \quad \mathrm{rank}(X) \leq k$$

$$s.t. \quad ||X||_* \leq \lambda(k)$$

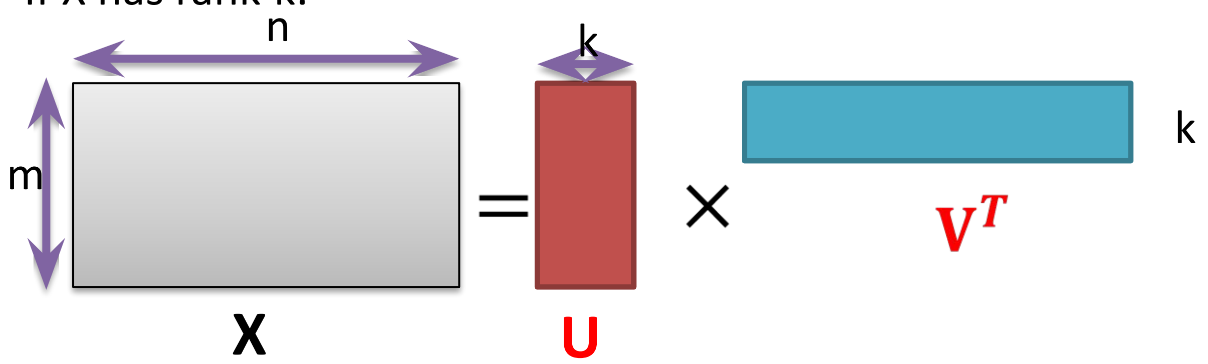
-
  - $||X||_*$: sum of singular values
  - Candes and Recht prove that above problem solves matrix completion (under assumptions on Ω and M)
  - However, convex optimization methods for this problem don’t scale well

Alternating Minimization

$$\min_{X} \; Error_{\Omega}(X) = \sum_{(i,j)\in\Omega} \left(X_{ij} - M_{ij}\right)^2$$

$$s.t \qquad \mathbf{rank}(X) \leq k$$

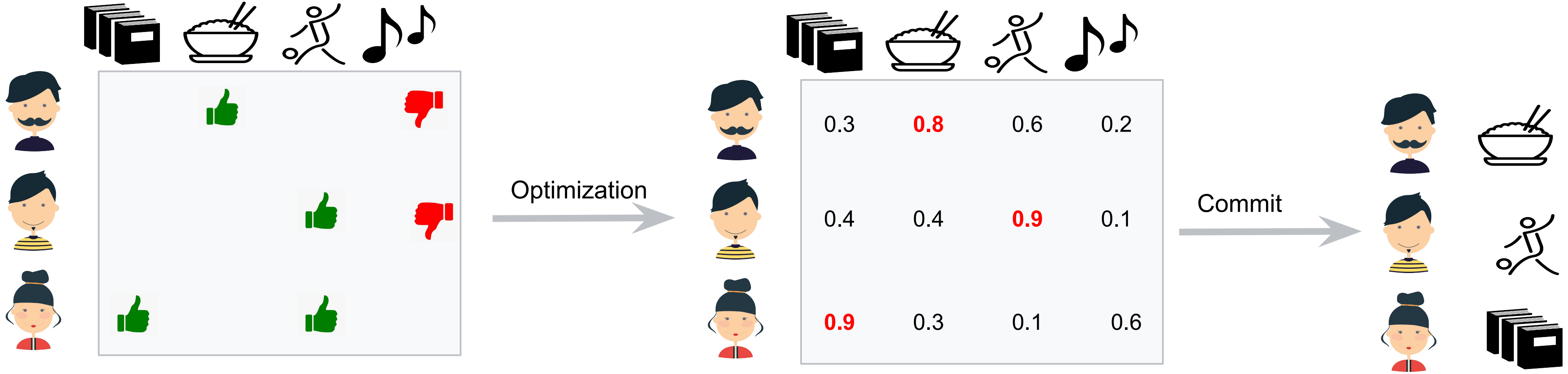
- If X has rank-k:



$$V^{t+1} = \min_{V} Error_{\Omega}(U^{t}, V)$$

$$U^{t+1} = \min_{U} Error_{\Omega}(U, V^{t+1})$$

Greedy Algorithm



Optimization

0.3	**0.8**	0.6	0.2
0.4	0.4	**0.9**	0.1
0.9	0.3	0.1	0.6

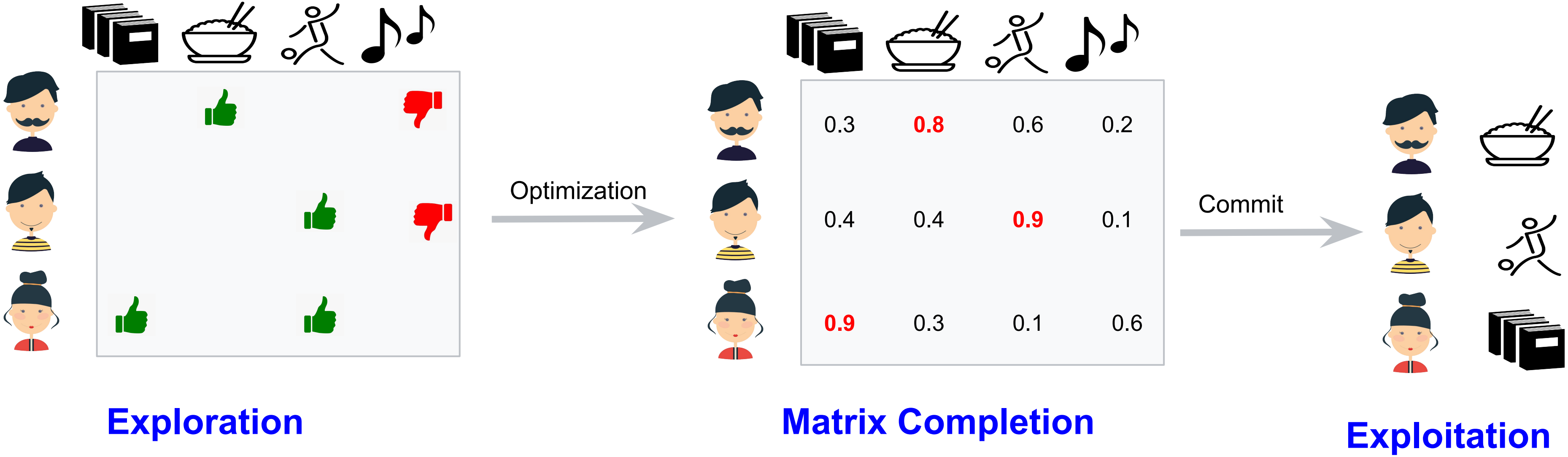
Commit

Exploration

Matrix Completion

Exploitation

Greedy Algorithm (Regret Guarantees)



0.3	**0.8**	0.6	0.2
0.4	0.4	**0.9**	0.1
0.9	0.3	0.1	0.6

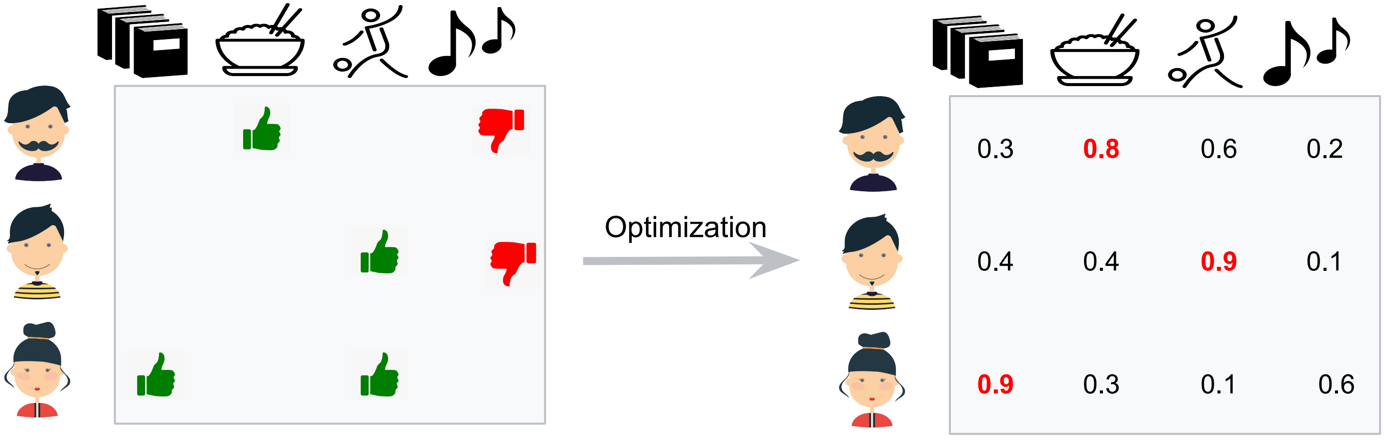
How much exploration data should we use? Better exploration → better estimate but longer cold-start

Regret of our **Greedy** algorithm:

$$\left(1+\frac{\text{items}}{\text{users}}\right)^{1/3}\times \text{rounds}^{2/3}$$

Sub-optimal in rounds

Challenges in Analysis (Greedy Algorithm)



	👍		👎
		👍	👎
👍		👍	

0.3	**0.8**	0.6	0.2
0.4	0.4	**0.9**	0.1
0.9	0.3	0.1	0.6

Partial Noisy Observations

Entire Estimated Matrix

Limitations of Offline Matrix Completion	Challenge	Solution
Square Matrices	Incoherence, Condition number	Random partition

Main Result for LATTICE

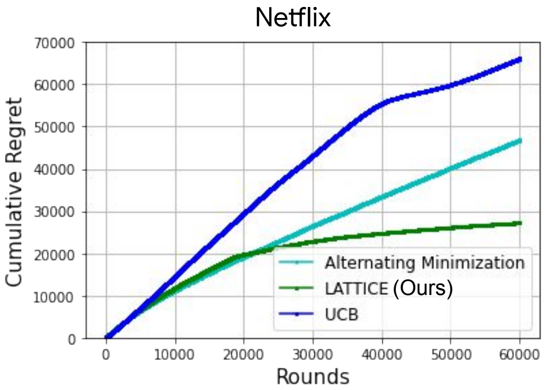
Regret of our collaborative algorithm (LATTICE):

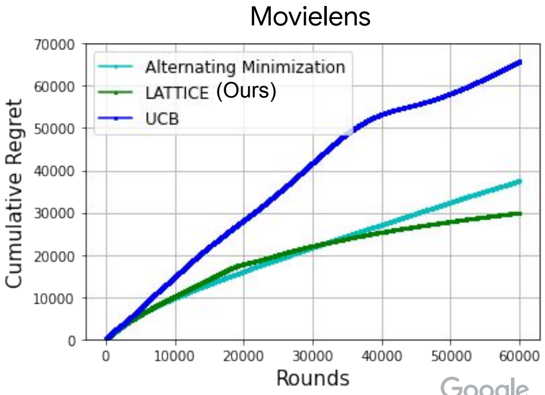
$$\sqrt{\left(1+\frac{\text{items}}{\text{users}}\right)\times \text{rounds}}$$

Optimal in items, users, rounds!

Regret of non-collaborative algorithms (UCB):

$$\sqrt{\text{items}\times\text{rounds}}$$


Netflix


Movielens

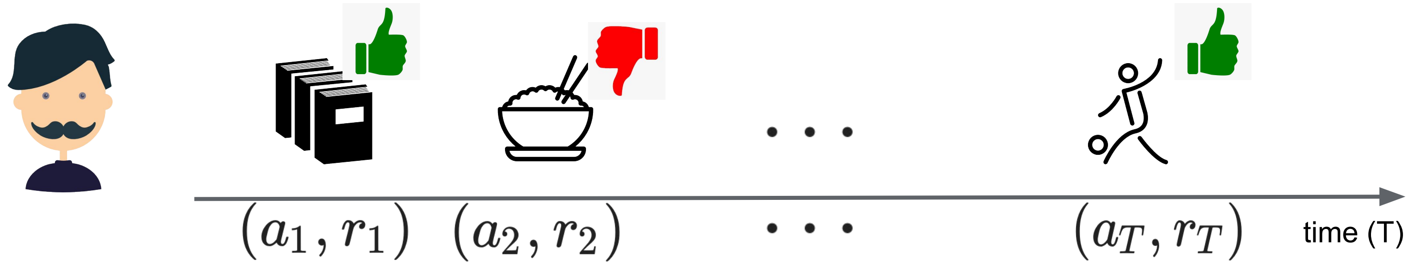
Remarks on Main Result

- Regret is minimax optimal in terms of $\text{items}, \text{users}, T$
- Dependence of regret on number of clusters (C) is C^3
  - Lower bound has $C^{1/2}$ dependence

- Our results extend to
  - Relaxed definitions of cluster. For u_1, u_2 in same cluster,

$$|\mu_{u_1,a} - \mu_{u_2,a}| \leq \epsilon, \text{ for all } a$$

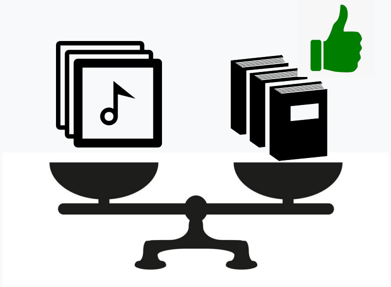
Review: Multi-Armed Bandits (MABs)



$$\text{Regret}_T = T \max_a \mu_a - \sum_{t=1}^{T} \mathbb{E}[r_t]$$

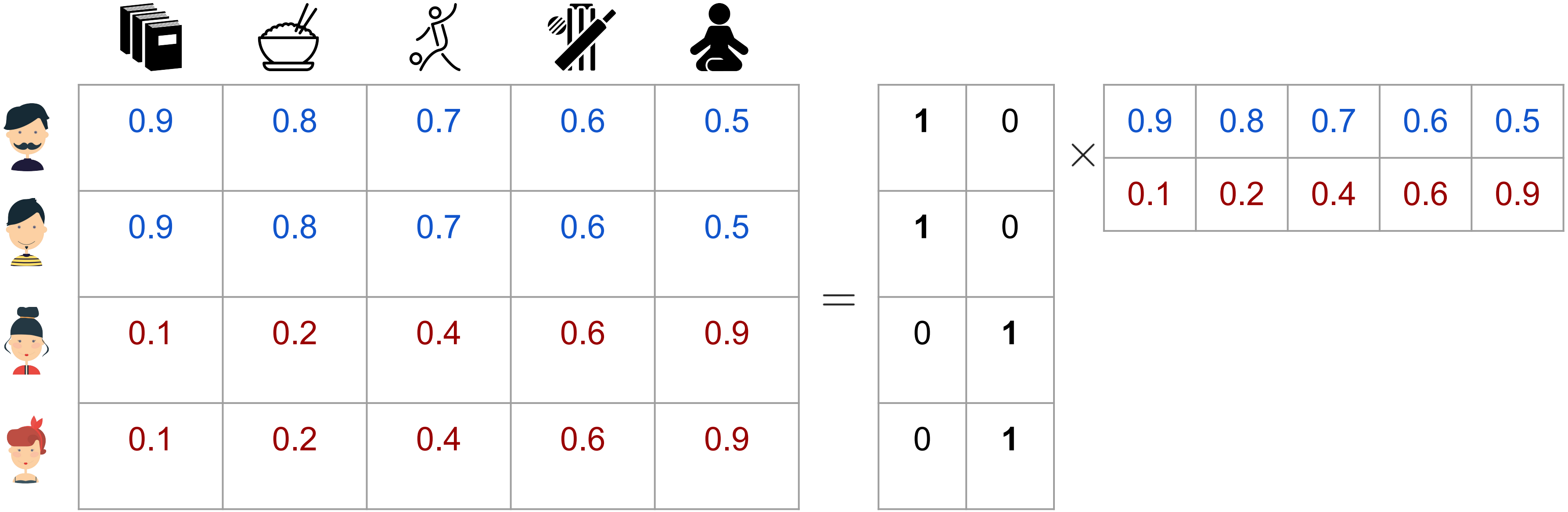
"Distance" of our recommendations from the optimal item

$$r_t = \mu_{a_t} + \epsilon_t$$



Exploration - Exploitation

Algorithmic Idea 1: Reduction to Offline Low-Rank Matrix Completion



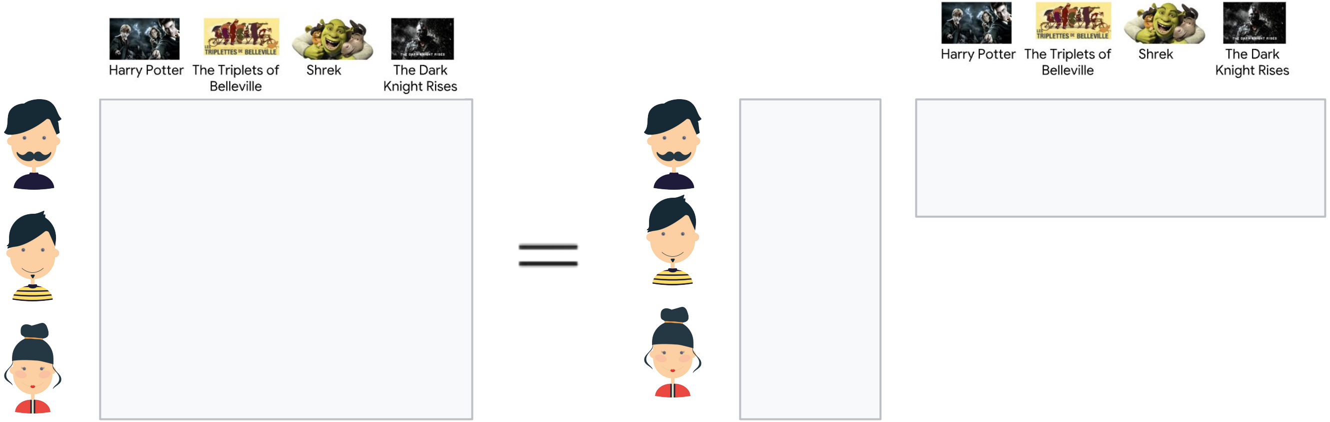
$$\begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.5 \\ 0.9 & 0.8 & 0.7 & 0.6 & 0.5 \\ 0.1 & 0.2 & 0.4 & 0.6 & 0.9 \\ 0.1 & 0.2 & 0.4 & 0.6 & 0.9 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & 0 \\ \mathbf{1} & 0 \\ 0 & \mathbf{1} \\ 0 & \mathbf{1} \end{bmatrix} \times \begin{bmatrix} 0.9 & 0.8 & 0.7 & 0.6 & 0.5 \\ 0.1 & 0.2 & 0.4 & 0.6 & 0.9 \end{bmatrix}$$

User x Item reward matrix is low-rank

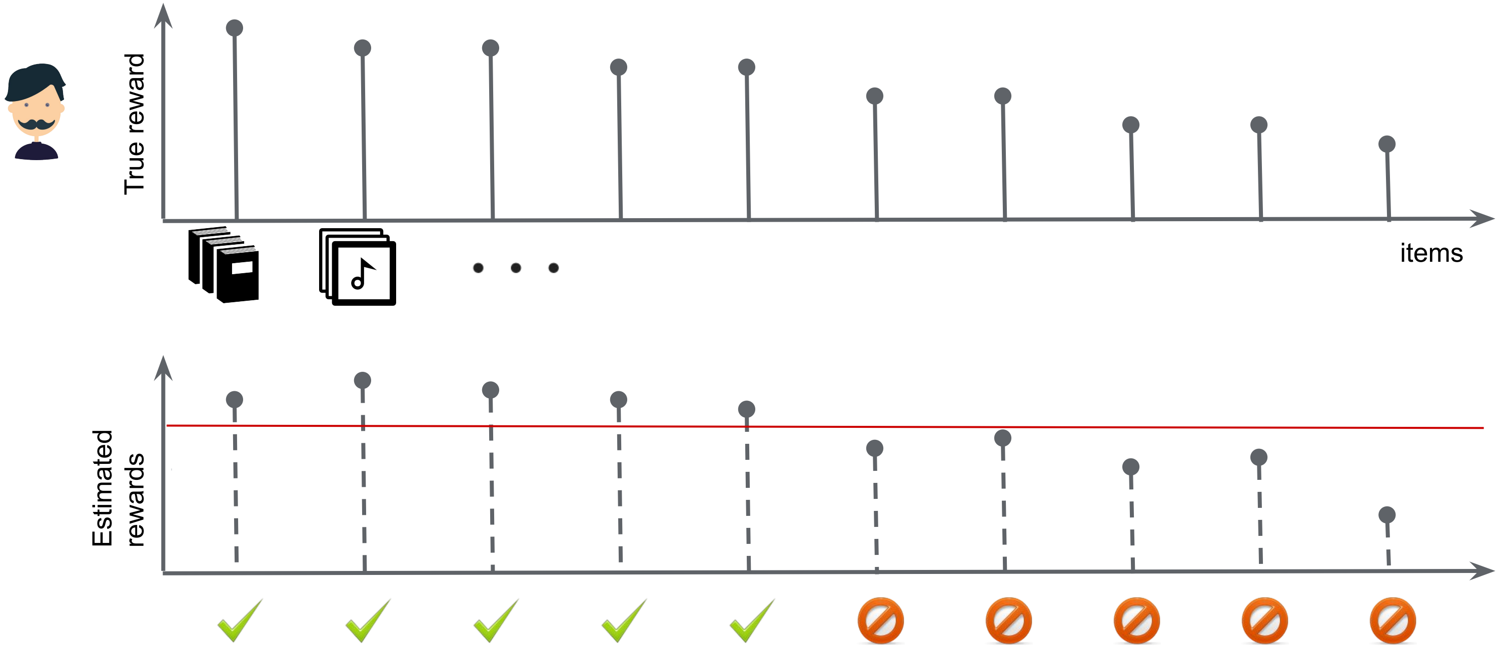
Challenges

Offline matrix completion difficult to extend

- **Entry-wise guarantees** are necessary
- Existing results are with **bernoulli sampling** and for **square matrices**



Review: Phased Elimination for MABs

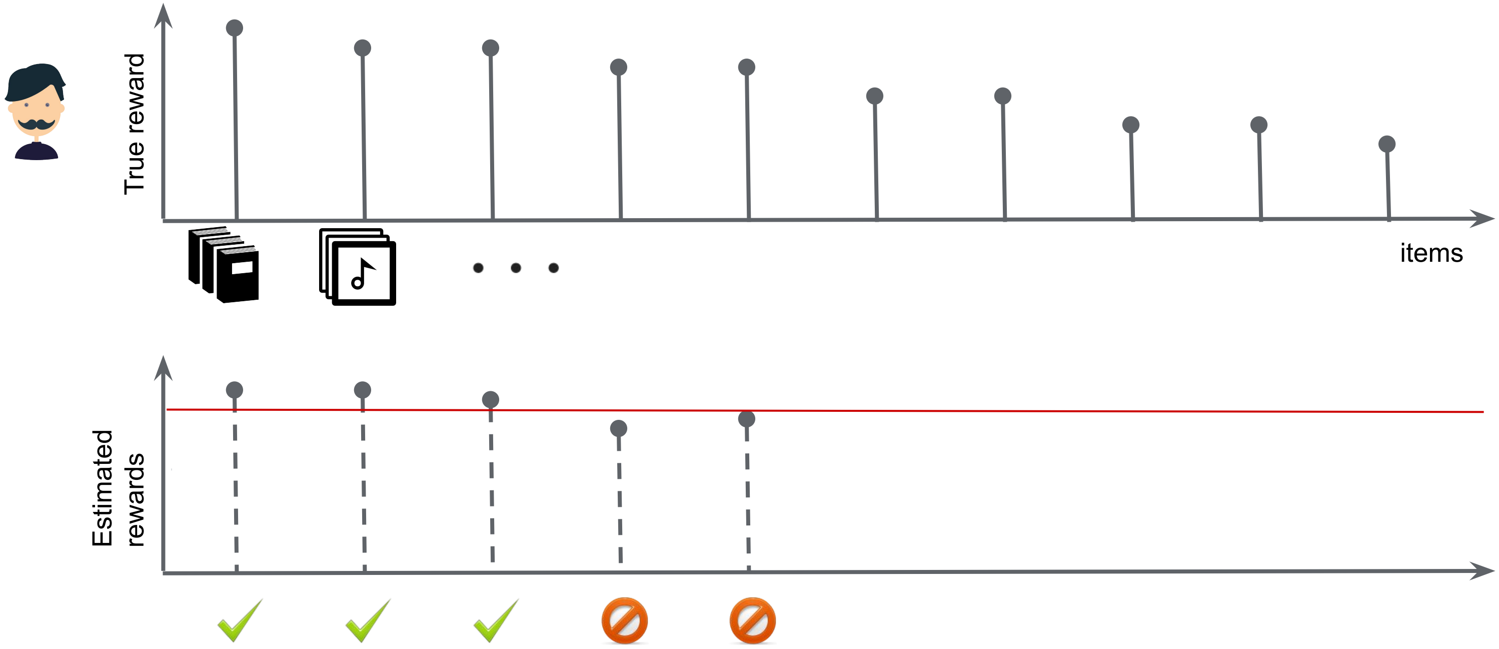


Phase 1: Estimate rewards of each item to precision ½ by repeated sampling

Eliminate items with rewards farther than 1 away from best estimated item

Auer, Peter, and Ronald Ortner. "UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem." *Periodica Mathematica Hungarica* 61, no. 1-2 (2010): 55-65.

Review: Phased Elimination for MABs

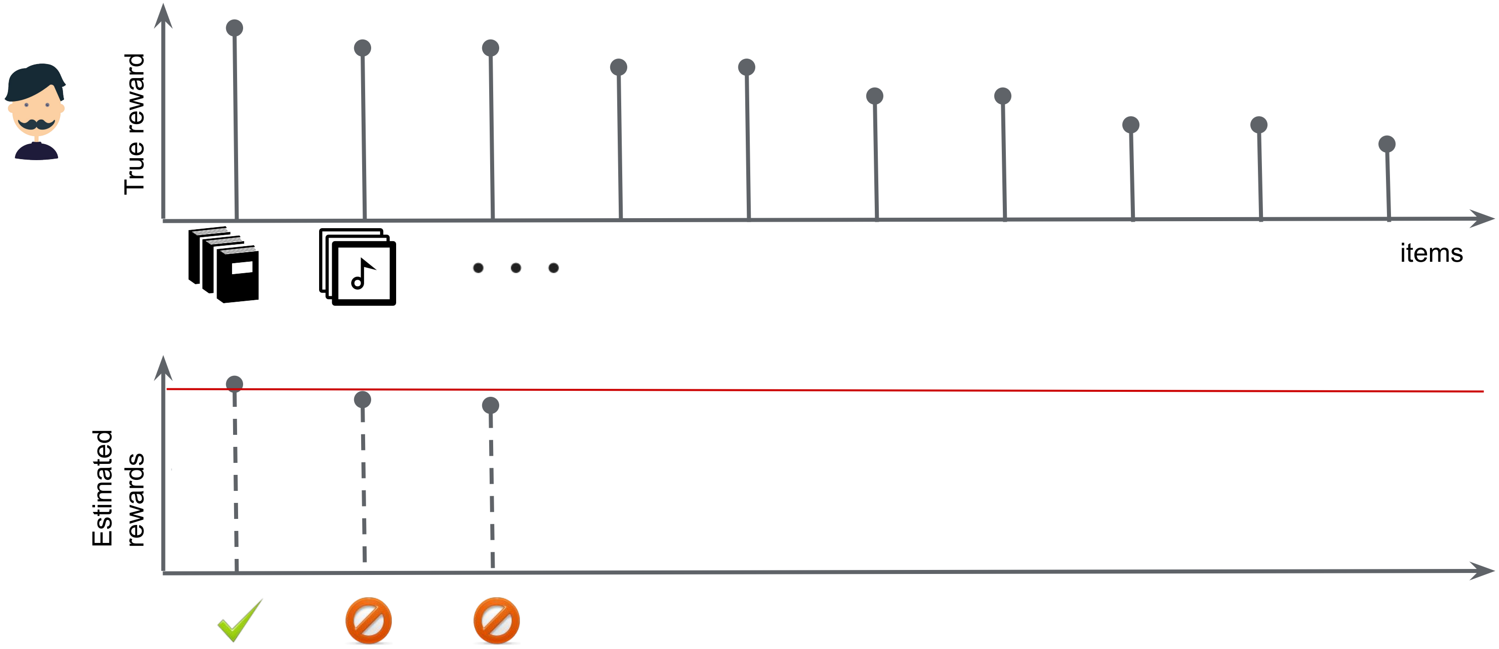


Phase 2: Estimate rewards of each item to precision ¼ by repeated sampling

Eliminate items with rewards farther than 1/2 away from best estimated item

Auer, Peter, and Ronald Ortner. "UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem." *Periodica Mathematica Hungarica* 61, no. 1-2 (2010): 55-65.

Review: Phased Elimination for MABs



Phase 3: Estimate rewards of each item to precision ⅛ by repeated sampling

Eliminate items with rewards farther than 1/4 away from best estimated item

Auer, Peter, and Ronald Ortner. "UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem." *Periodica Mathematica Hungarica* 61, no. 1-2 (2010): 55-65.

Review: Phased Elimination for MABs

These rates are optimal (modulo log factors)

Phased elimination has worst case regret of $\tilde{O}(\sqrt{\text{items} \times T})$

- Average of n independent Gaussian random variables is within $\tilde{O}(1/\sqrt{n})$ of true mean

- Suppose item 1 is optimal. Let $\Delta_i = \mu_1 - \mu_i$.
- WLOG assume $\Delta_i > \sqrt{\text{items}/T}$. Then item i is pulled at most $1 + \tilde{O}(1/\Delta_i^2)$ times
  - If $\Delta_i < \sqrt{\text{items}/T}$, rewards of items 1, i are close. So we won't incur much regret

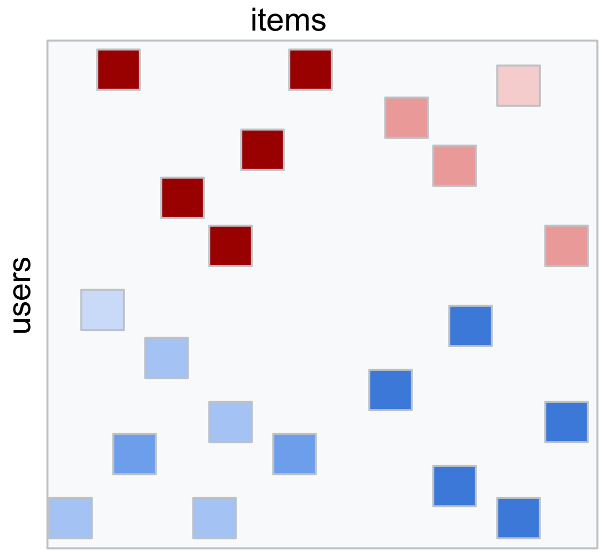
- Regret of algorithm is $\sum_{i>2} \Delta_i + \tilde{O}(1/\Delta_i) = \tilde{O}(\sqrt{\text{items} \times T})$

LATTICE: Phased Elimination + User Clustering

Phase 1

- Place all users in a single group, and keep all items active

- Sample item randomly to get $\Delta = 1/2$ accurate estimate of reward matrix

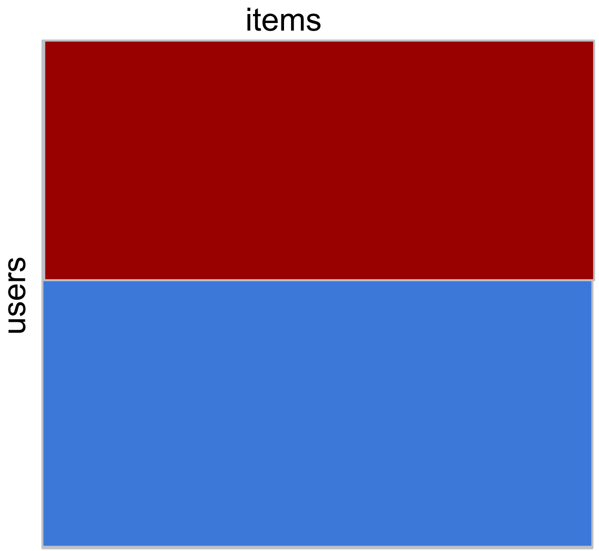


LATTICE: Phased Elimination + User Clustering

Phase 1

- Place all users in a single **group**, and keep all items active

- Sample items randomly to get $\Delta = 1/2$ accurate estimates of reward matrix
- Perform matrix completion with partial noisy observations
- Group/Cluster users with estimated embeddings (from Matrix Completion)

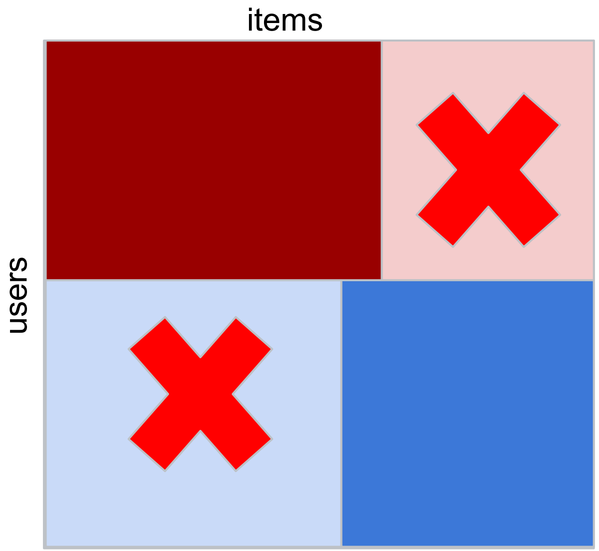


LATTICE: Phased Elimination + User Clustering

Phase 1

- Place all users in a single **group**, and keep all items active

- For each user, it’s good items are less than 2Δ away from empirical best item
- **Joint good items** - Union of good items for users in each group
- Discard all items not in joint good items for each group/cluster

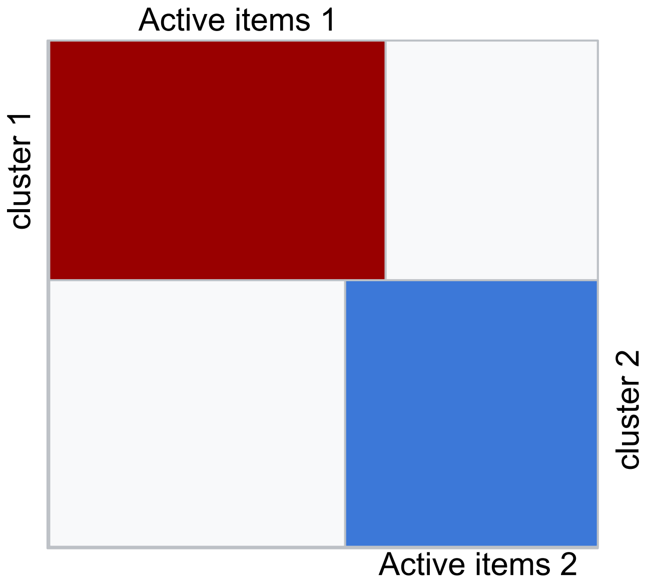


LATTICE: Phased Elimination + User Clustering

Phase 1

- Place all users in a single group, and keep all items active

- For each user, it's good items are less than 2Δ away from empirical best item
- **Joint good items** - Union of good items for users in each group
- Discard all items not in joint good items for each group/cluster



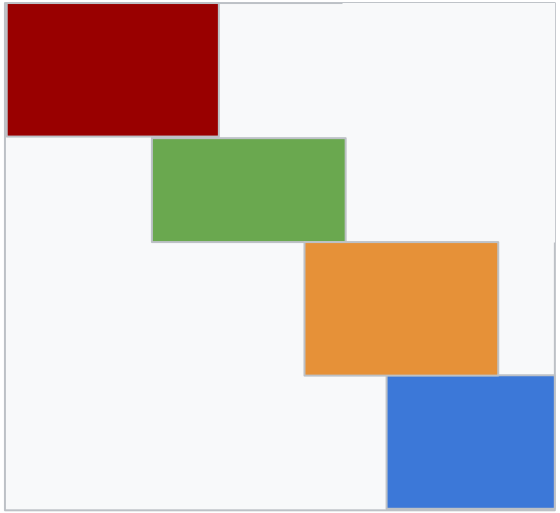
Key Intermediate Results

- For each user, their corresponding true best item always survives (Easy)
- Each constructed group is a union of true clusters (Hard)
- For each user, the joint active items are still good (Hard)
- Each sub-matrix satisfies incoherence, condition number conditions necessary for matrix completion oracle (Hard)

LATTICE: Phased Elimination + User Clustering

Phases > 1

- Repeat the previous procedure on each submatrix
- Improve the quality of reward estimates exponentially
- Why Phased Elimination? **Uniformly sample** in carefully constructed sub-matrices
  - Difficult to generalize optimal algorithms in single user MAB



Key Step in Proof

Phase i requires at most $\tilde{O}((\text{items} + \text{users})/\Delta_i^2)$ samples

- relies on matrix-completion guarantees
- Without collaboration the sample complexity is $\tilde{O}(\text{items} \times \text{users}/\Delta_i^2)$

Sum of regret across phases

$$\sum_{i>2} \Delta_i + \tilde{O}((\text{items} + \text{users})/\Delta_i) = \tilde{O}(\sqrt{(\text{items} + \text{users}) \times T})$$

Idea: Why Phased Elimination ?

- Confidence Bound based algorithm or Thomson Sampling will not work
- Our algorithm (LATTICE): Phased elimination framework
  - Carefully constructed sub-matrices of reward matrix in each phase where we collect data randomly
  - Run low-rank matrix completion algorithm on data collected in each phase to get a good estimate of the reward sub-matrices
  - Discard sub-optimal arms after each phase jointly for groups of users

Idea: Why Phased User Clustering

- Cluster First - Greedy algorithm
- Group (**coarsely cluster**) users in each phase based on the estimated rewards
  - Users with similar estimated rewards are placed in the same group
- Collaborate across users in the group to get better reward estimates in the next phase
  - Users in different clusters but in same group **are also similar (to some extent)**

Outline

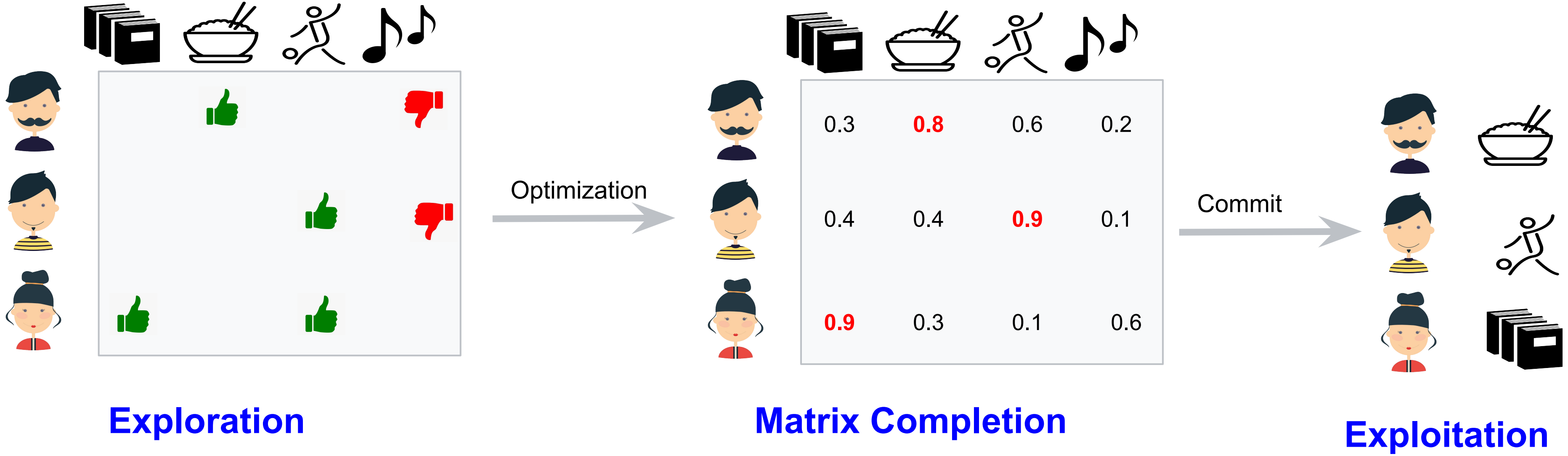
01 Need for collaboration in Bandits

02 Collaborative Multi-Armed Bandits

03 Simplified Practical Algorithm

04 Conclusion

Greedy Algorithm (Regret Guarantees)



Optimization

Commit

0.3	**0.8**	0.6	0.2
0.4	0.4	**0.9**	0.1
0.9	0.3	0.1	0.6

Exploration

Matrix Completion

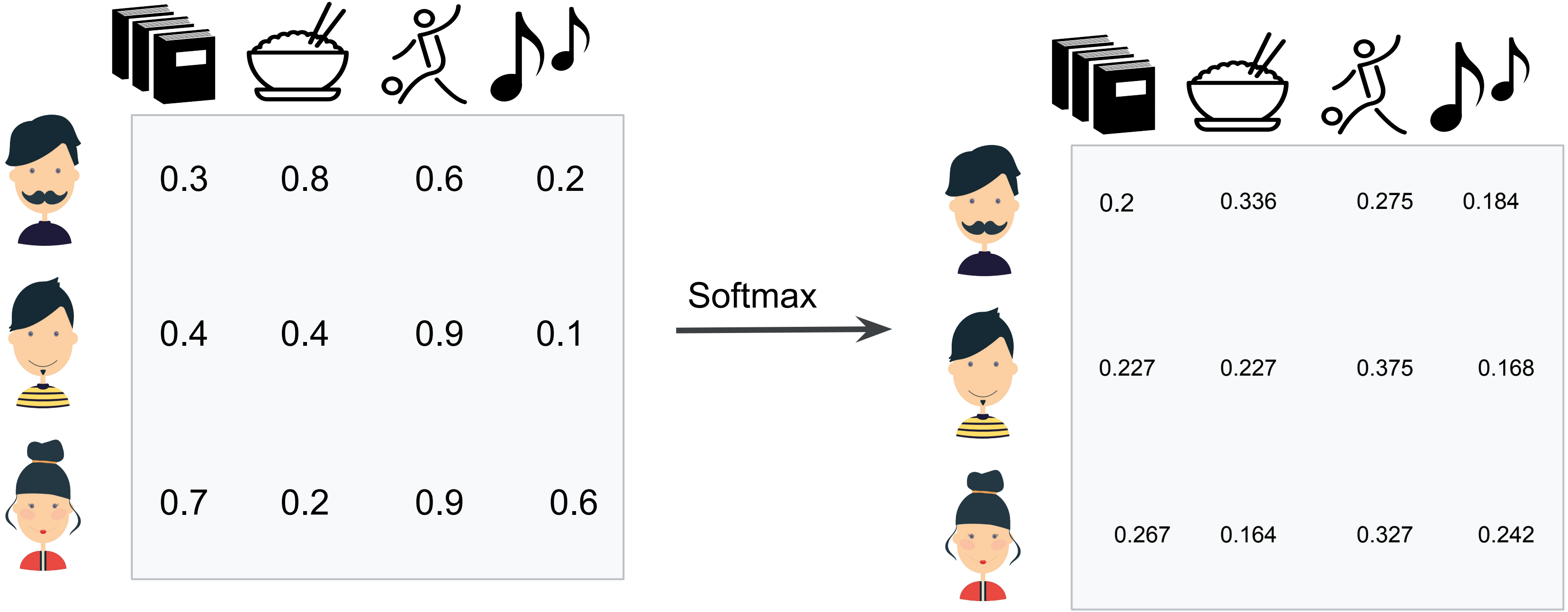
Exploitation

Regret of our **Greedy** algorithm:

$$\left(1+\frac{\text{items}}{\text{users}}\right)^{1/3}\times \text{rounds}^{2/3}$$

Sub-optimal in rounds

Fix - Soft Commit instead of Hard Commit



	📚	🍚	⚽	🎵
👨	0.3	0.8	0.6	0.2
👦	0.4	0.4	0.9	0.1
👩	0.7	0.2	0.9	0.6

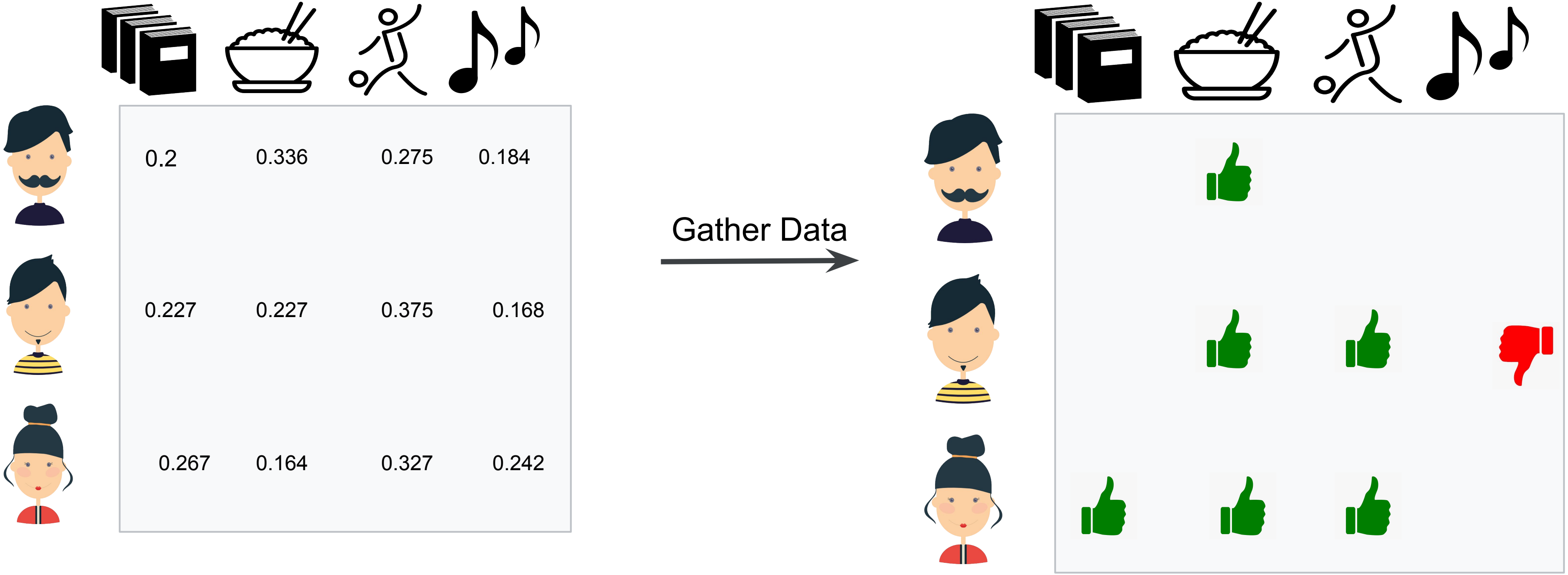
Softmax →

	📚	🍚	⚽	🎵
👨	0.2	0.336	0.275	0.184
👦	0.227	0.227	0.375	0.168
👩	0.267	0.164	0.327	0.242

Softmax

$$\sigma(\mathbf{z})_i = \frac{e^{\beta z_i}}{\sum_{j=1}^{K} e^{\beta z_j}}$$

Gather new data via soft commit

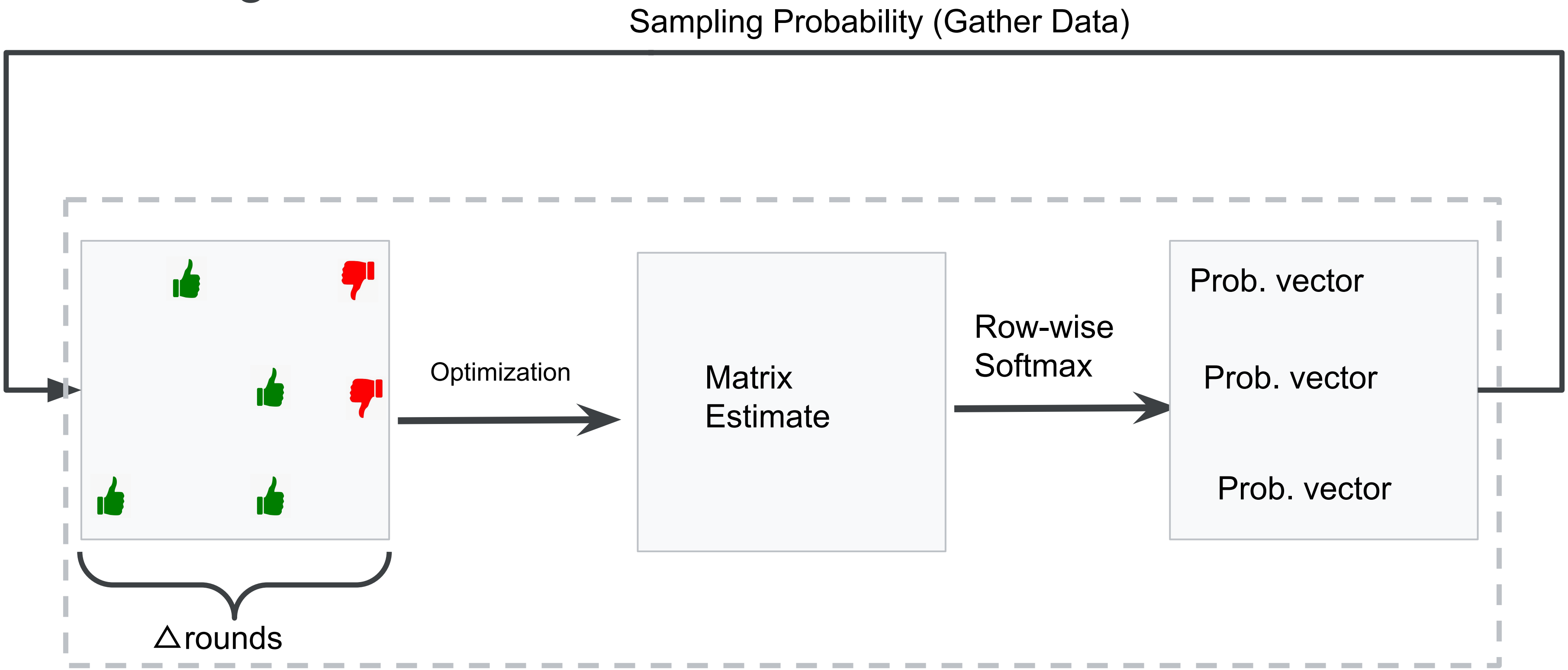


	📚	🍚	⚽	🎵
	0.2	0.336	0.275	0.184
	0.227	0.227	0.375	0.168
	0.267	0.164	0.327	0.242

	📚	🍚	⚽	🎵
		👍		
		👍	👍	👎
	👍	👍	👍	

Intuitively, more likes !!

Iterative Algorithm

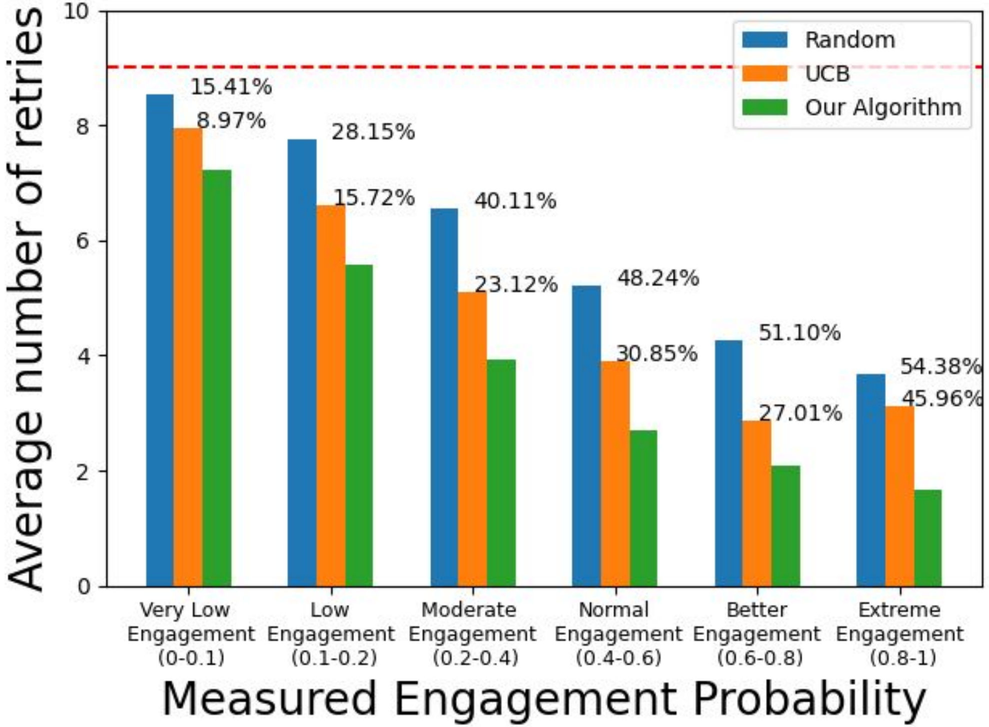


- Iterative algorithm where we update sampling probabilities for each user based on estimate
- Gather more data respecting constraints and current sampling probabilities
- Update estimate of reward matrix

Application to Maternal Health Program - Kilikari

- Kilkari (AI4SG) – Collaboration with MASSI
  - Improve the wellbeing of expecting mothers
  - 3 million mothers. Figure out optimal time-slot to call each individual
  - This is a mobile maternal health program run by ARMAAN in collaboration with Ministry of Health (GoI).

Offline Results on the Kilikari Dataset



Across different engagement slices, we check how many call retries before engagement happens

Given same resources,
We can onboard **56%** more users vs
a policy that plays best bandit algorithm per user (when we don’t collaborate)

Outline

01 Need for collaboration in Bandits

02 Collaborative Multi-Armed Bandits

03 Simplified Practical Algorithm

04 Conclusion

Extension to Reinforcement Learning

- Each user now is an MDP with temporal dependence.
- Actions have long term consequences
- Assumption: low rank reward
- Challenge: some states might be unreachable with any policy
- Our contribution: Collaborative exploration for RL

Main Result:
Number of sample trajectories to obtain optimal policy for each user:

O(users + states*actions)

Naive Algorithm:

O(users*states*actions)

[Agarwal, Jain, Kowshik, Nagaraj,Netrapalli, ICML 2023]

Takeaways

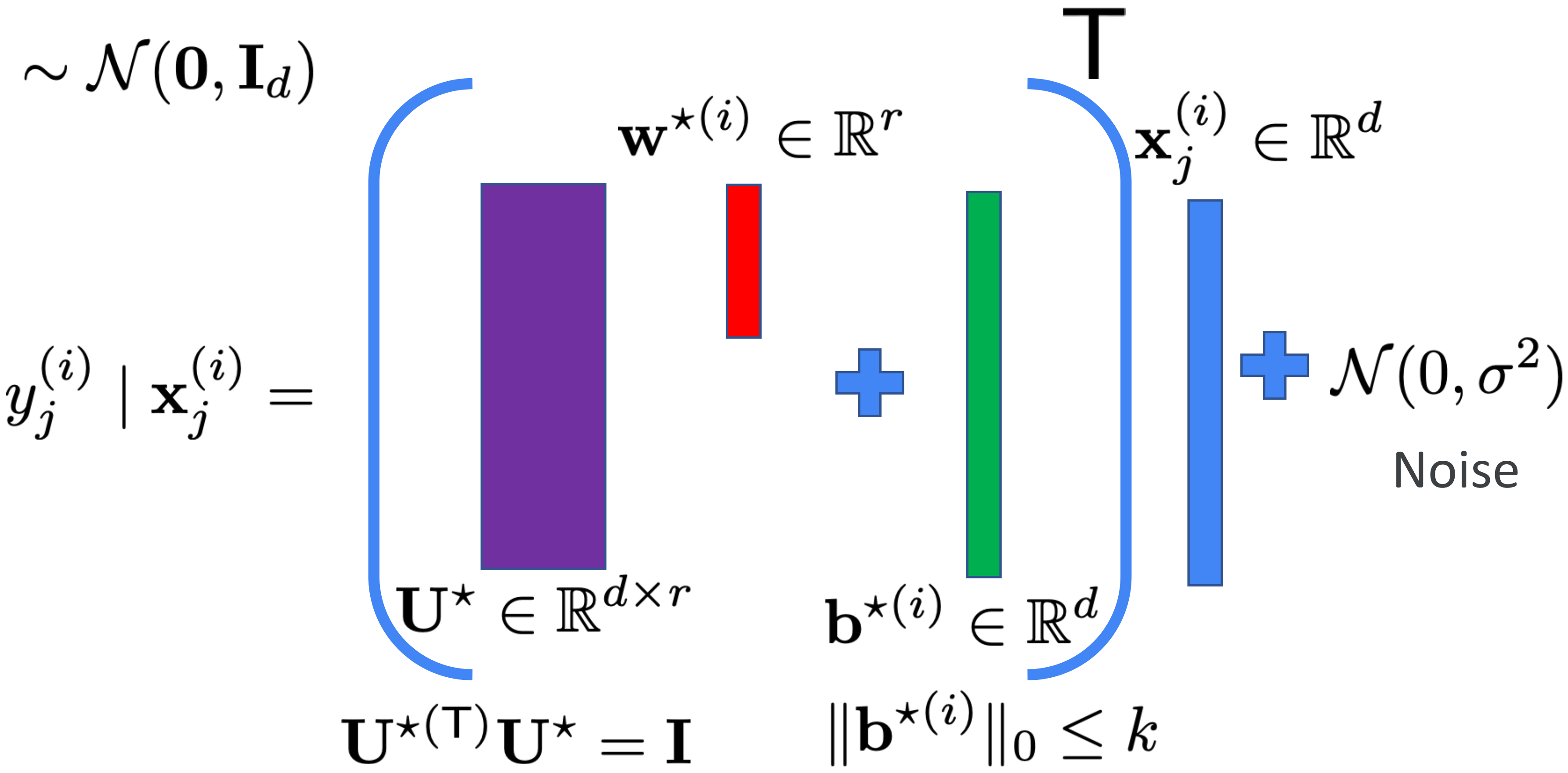
- Modern recommender systems have millions of users, items
    - Non-collaborative algorithms suffer from a large cost
- Collaboration improves the regret of bandit algorithms
- Greedy algorithm - suboptimal in rounds
- LATTICE - Phased Elimination Algorithm for collaborative bandits with latent clusters
    - gets minimax optimal regret
    - relies on phased-elimination of arms and phased-clustering of users
- Greedy with Soft Commit - shows great empirical promise in healthcare data

Other Results

- Optimal Regret for rank-1 setting (ICLR 23)
- Near-Optimal Regret for rank >1 setting with hott items assumption (In submission)
- Optima Regret with blocking constraint (NeurIPS 23)

Model for Offline Personalization/Meta-Learning

$\mathbf{x}_j^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$



$y_j^{(i)} \mid \mathbf{x}_j^{(i)} =$

T

$\mathbf{w}^{\star(i)} \in \mathbb{R}^r$

$\mathbf{x}_j^{(i)} \in \mathbb{R}^d$

$+ \mathcal{N}(0, \sigma^2)$

Noise

$\mathbf{U}^\star \in \mathbb{R}^{d \times r}$

$\mathbf{b}^{\star(i)} \in \mathbb{R}^d$

$\mathbf{U}^{\star(\mathsf{T})}\mathbf{U}^\star = \mathbf{I}$

$\|\mathbf{b}^{\star(i)}\|_0 \leq k$

Broad Future Work

Personalization

- ➢ Contextual models for multi-user setting (Observable context)
- ➢ Incorporate Side Information (Graphs, Offline data) - Truly Hybrid Models
- ➢ Incorporate Robustness, Privacy, Unlearning
- ➢ Broadly, improve our understanding of scalable algorithms for non-convex optimization

Broad Future Work

Personalization

- Contextual models for multi-user setting (Observable context)
- Incorporate Side Information (Graphs, Offline data) - Truly Hybrid Models
- Incorporate Robustness, Privacy, Unlearning
- Broadly, improve our understanding of scalable algorithms for non-convex optimization

Parameter Efficient Fine-Tuning/ Meta Learning

- Extend scalability of LoRA and QLoRA via incorporating constraints across tasks.
- Sub-channel quantization/Sparse dictionary learning to improve memory footprint.
- Even initializing slightly modified Transformer Blocks are a challenge.
- Generalize within and outside task distributions

Broad Future Work

Personalization

- Contextual models for multi-user setting (Observable context)
- Incorporate Side Information (Graphs, Offline data) - Truly Hybrid Models
- Incorporate Robustness, Privacy, Unlearning
- Broadly, improve our understanding of scalable algorithms for non-convex optimization

Parameter Efficient Fine-Tuning/ Meta Learning

- Extend scalability of LoRA and QLoRA via incorporating constraints across tasks.
- Sub-channel quantization/Sparse dictionary learning to improve memory footprint.
- Even initializing slightly modified Transformer Blocks are a challenge.
- Generalize within and outside task distributions

Multi-task Learning

- Mixture Models when task labels are lost/expensive
- Non-convex constraints on components of mixture models