Do you Prefer Learning with Preferences:

Foundations of Human Aligned Prediction Models with Relative Feedback

Aadirupa Saha (Apple)
Aditya Gopalan (Indian Institute of Science)

NeurIPS 2023, New Orleans

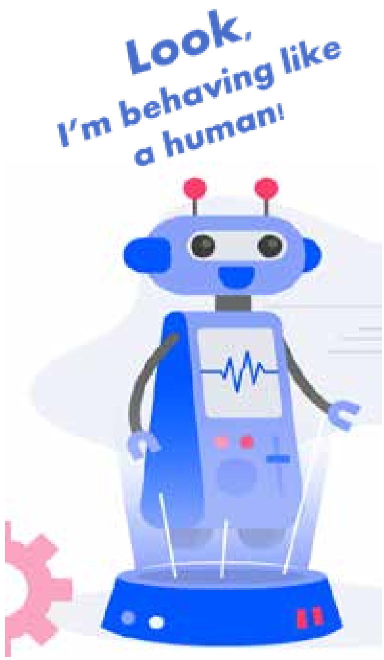
Part – I (Motivation)

AI and implications

- "The field of AI is often thought of as having four distinct approaches, which can be described as thinking humanly, thinking rationally, acting humanly, and acting rationally.“

  - "Artificial Intelligence: A Modern Approach" (S. Russell and P. Norvig)

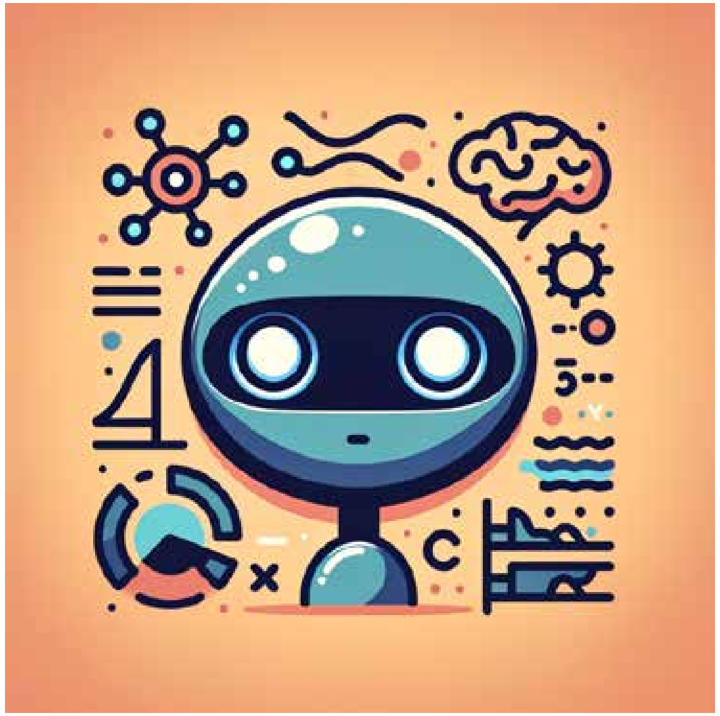
- Implication: If machines must behave like humans and need to learn to do so, then one must grapple with learning from human feedback



AI agent

$x \in \mathcal{X}$

Input



$y \in \mathcal{Y}$

Output

Agent:

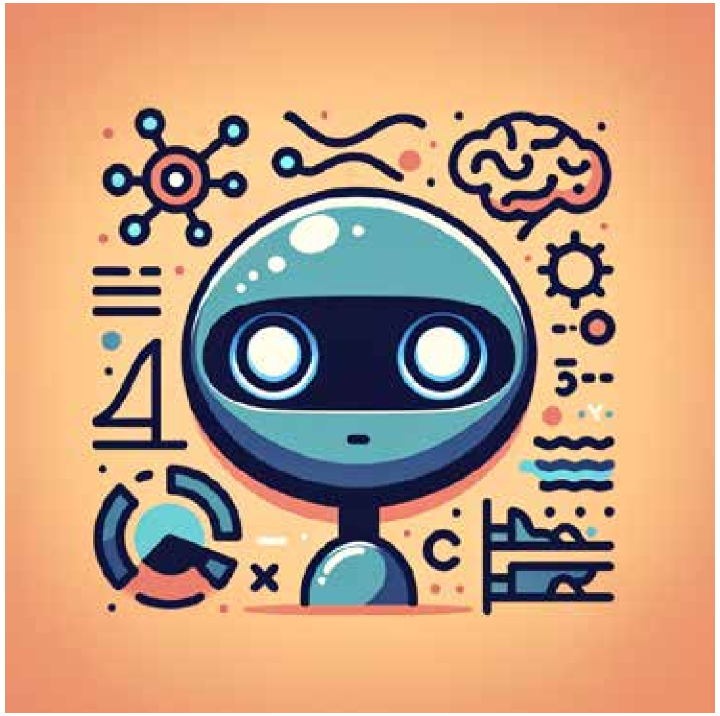
Performs a useful task by mapping input → output

"Artificial Intelligence: A Modern Approach" (S. Russell and P. Norvig), Sec. 2.1

AI agent

$x \in \mathcal{X}$

Input

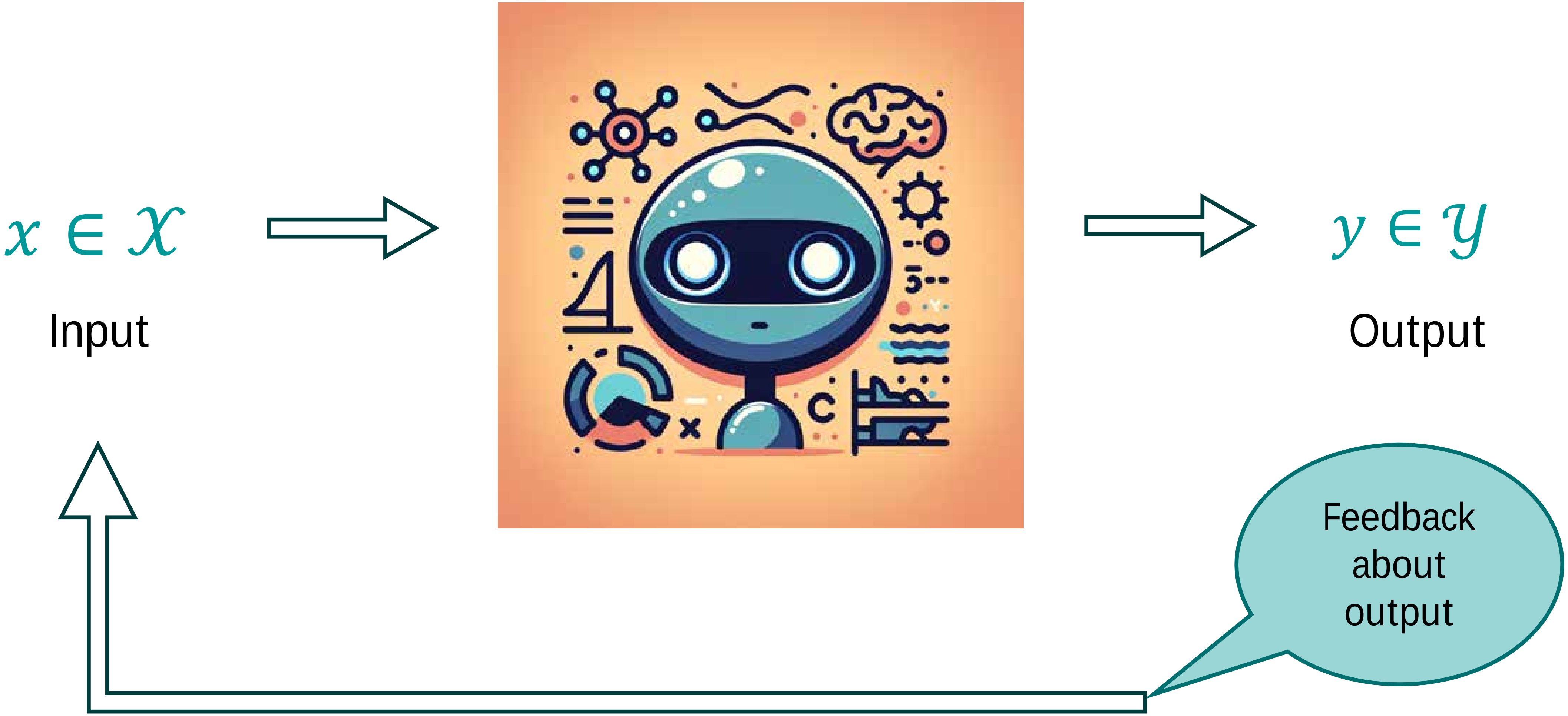


$y \in \mathcal{Y}$

Output

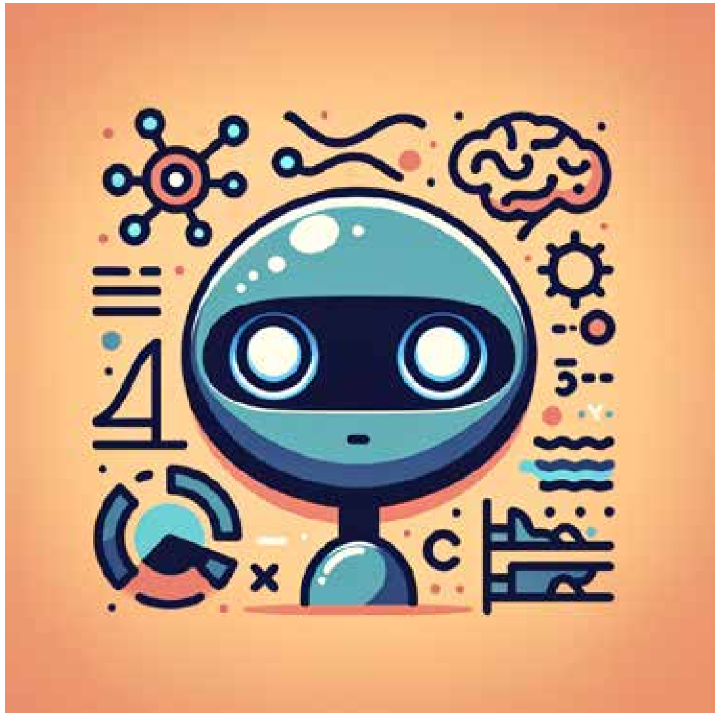
Output space is large and complex

AI agent



$x \in \mathcal{X}$
Input
$y \in \mathcal{Y}$
Output
Feedback about output

Task: Image recognition

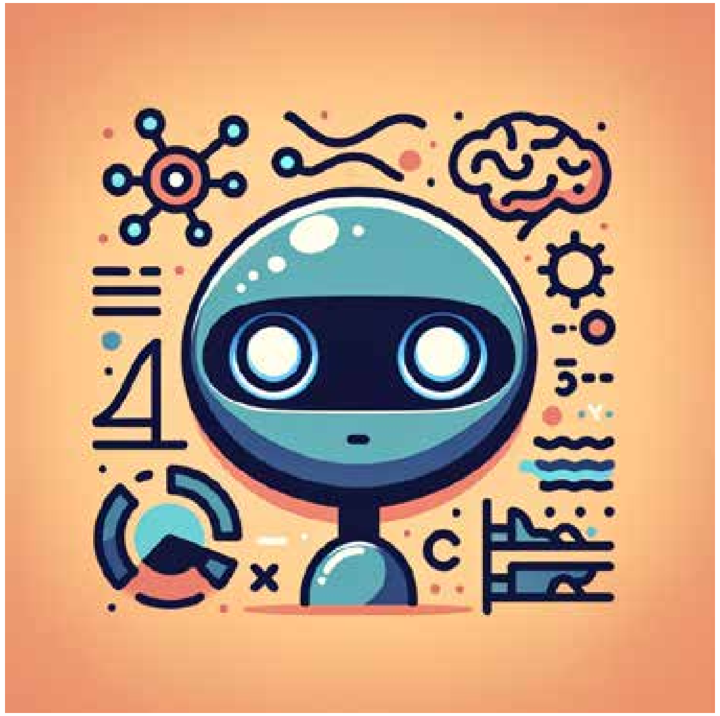


“Cat”

AI Agent

Task: Medical diagnosis

Symptom 1
Symptom 2
Symptom 3
Body temperature
Blood pressure
Blood sugar level
SpO2 level

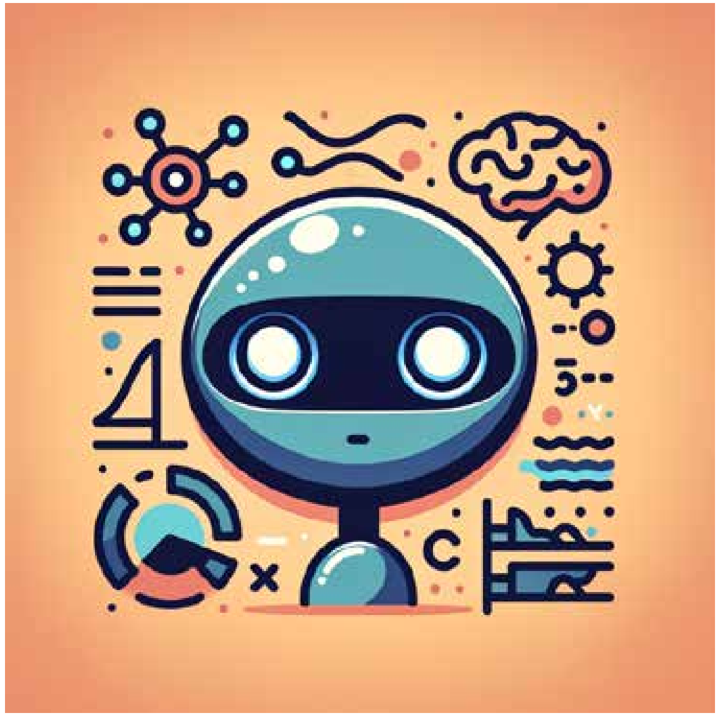


AI Agent

Diagnosis:
Type-2
Diabetes

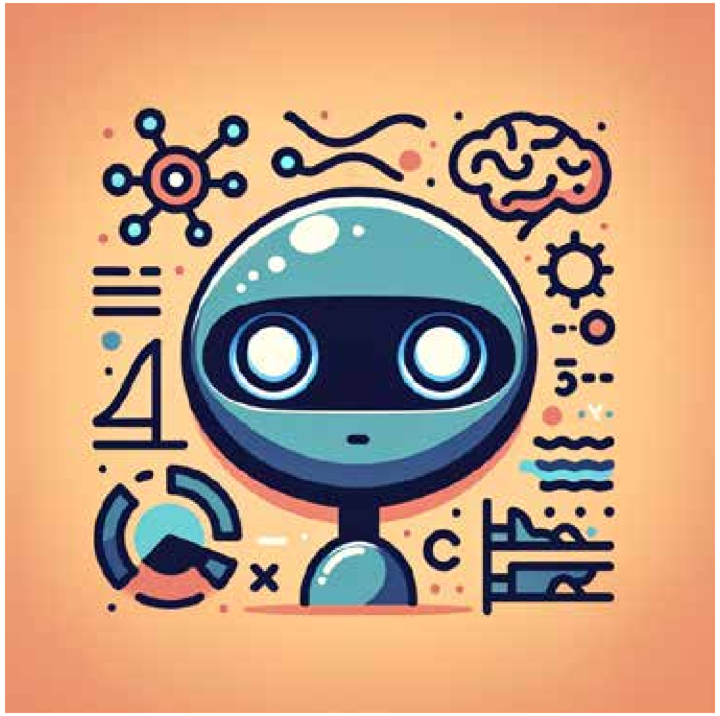
Task: (Personalized) Content recommendation

User features
User history
Product catalogue



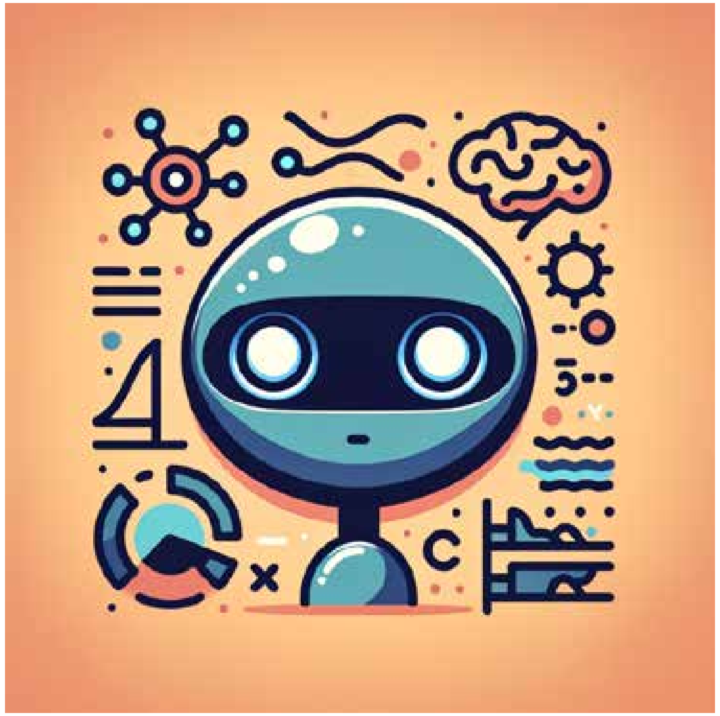
Task: (Personalized) Content recommendation

User features
User history
Product catalogue



Task: Game playing

White to move

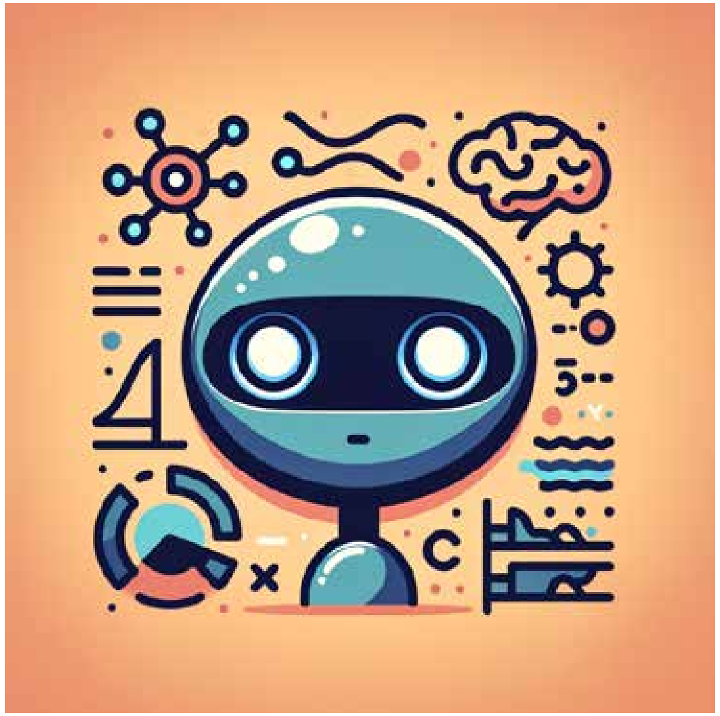


AI Agent

Best move: Nc3
Eval: +2.3

Task: Question-answering (chatbot)

“Where is NeurIPS 2023 being held?”



AI Agent

“Thank you for the question! I can certainly help answer it.

NeurIPS 2023 is going to be held in New Orleans. New Orleans is a city in ...”

Task: Controlling a physical system

Car position
Orientation
Velocity
Traffic status

Human feedback to train the agent

$x \in \mathcal{X}$

Input

$y \in \mathcal{Y}$

Output

Eg Feedback by **Demonstration** (chatbot)

Prompt:
Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.

Response:
Hydrogen comes first as element number one.
Helium is second for balloons to have fun!
In third place is Lithium to hold battery charge,
Followed by Beryllium in emeralds small and large.
Boron's number five to help us keep things clean.
Carbon's next at six, and it's found in every gene.
Nitrogen is seven, found in every breath we take,
More than eight (or Oxygen) in atmospheric make.
Number nine is Fluorine, helping polish up our teeth.
Neon gives us glowing signs to read and stand beneath.

Source: LLAMA 2 Tech Report (2023)

Eg Feedback by **Demonstration** (chatbot)

Prompt:
Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.

Response:
Hydrogen comes first as element number one.
Helium is second for balloons to have fun!
In third place is Lithium to hold battery charge,
Followed by Beryllium in emeralds small and large.
Boron's number five to help us keep things clean.
Carbon's next at six, and it's found in every gene.
Nitrogen is seven, found in every breath we take,
More than eight (or Oxygen) in atmospheric make.
Number nine is Fluorine, helping polish up our teeth.
Neon gives us glowing signs to read and stand beneath.

- Pro: Very informative
- Con: Costly, time-consuming

Source: LLAMA 2 Tech Report (2023)

Eg Feedback by **Demonstration** (robotics)

Robot state (position, velocity, etc)

Arm swing actuation

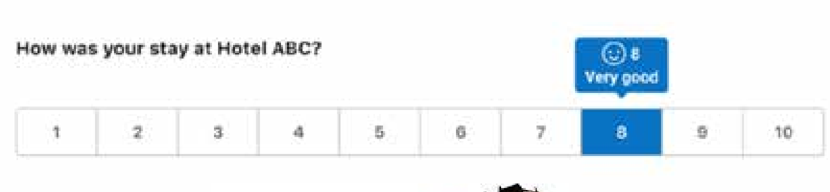
Human demonstration

- Pro: Very informative
- Con: Costly, time-consuming

Eg Feedback by **Numerical scoring** (rec systems)

User preferences, context

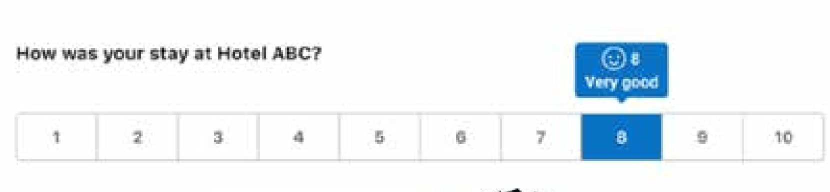
Recommendation: Stay at Hotel ABC



Eg Feedback by **Numerical scoring** (rec systems)

User preferences,
context

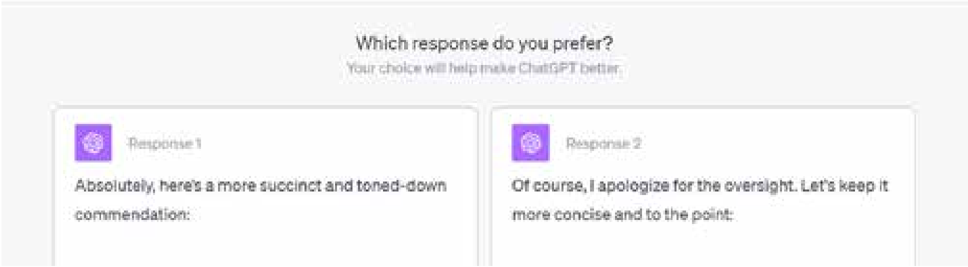
Recommendation:
Stay at Hotel ABC



- **Pro:** Informative
- **Con:** Somewhat costly, burdensome

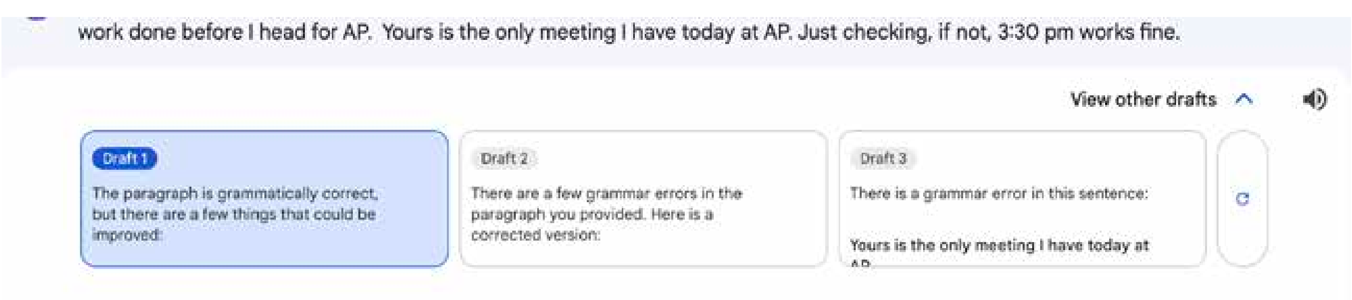
Eg Feedback by **Comparison** (LLM fine tuning)

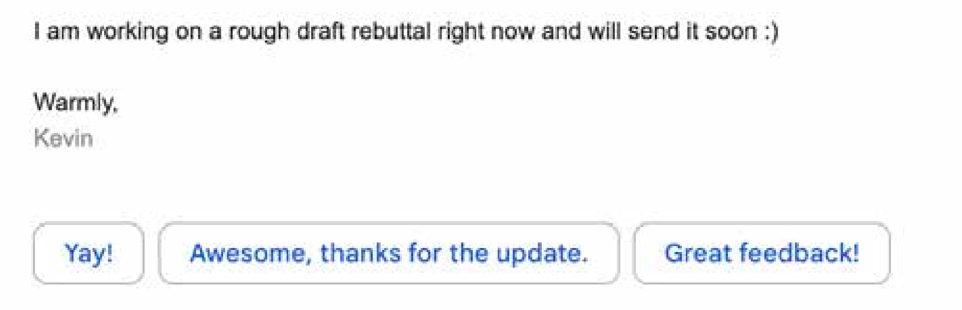
Prompt:
Please generate
a description of
...

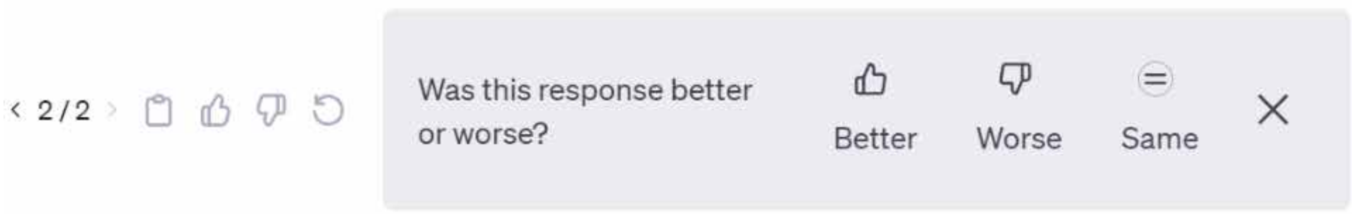


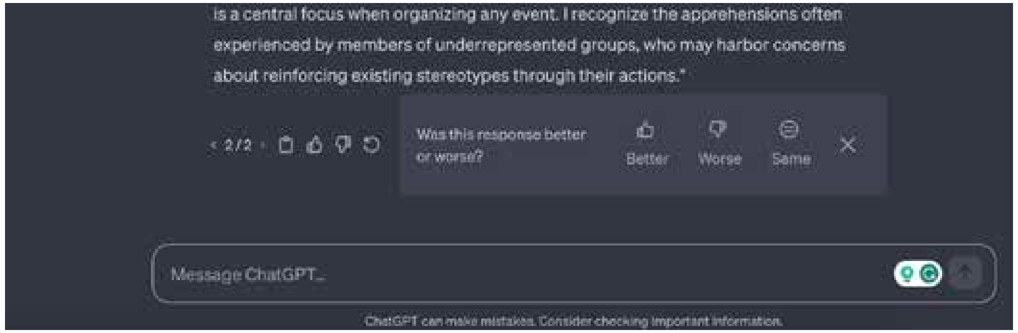
- **Pro:** Easy, Quick, Lightweight
- **Con:** Coarse, Not too informative

Modern LLMs ask for preference feedback









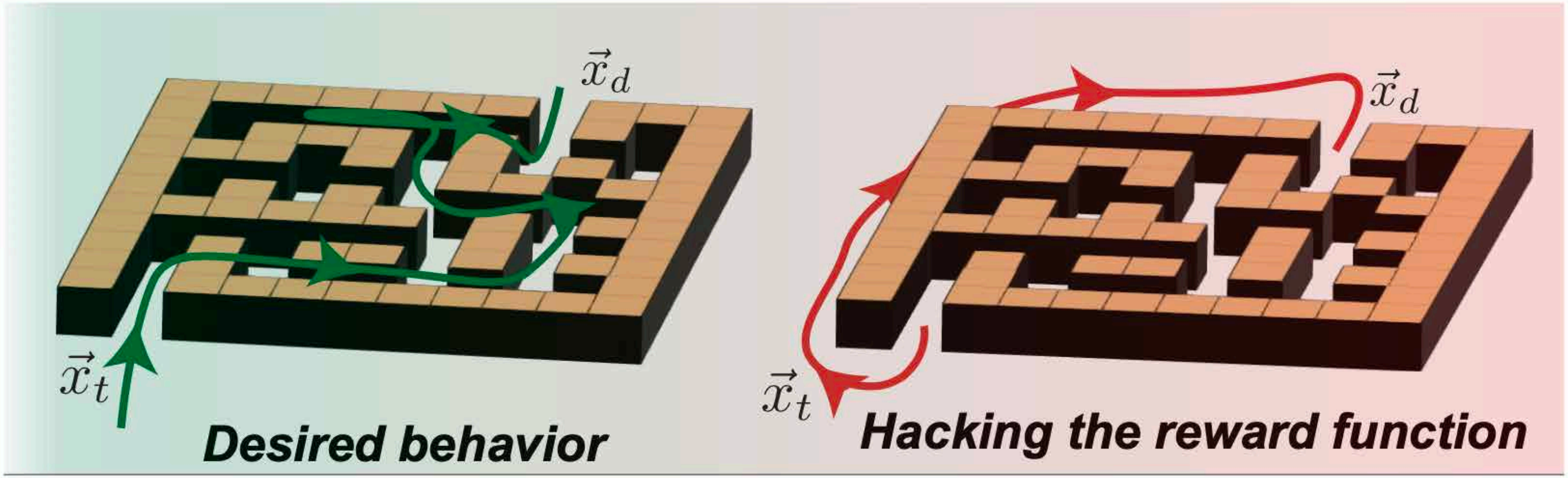
The case for fine-tuning with preferences

- Often one need enormous #preference feedback, which could be hard to obtain too.

- A warm start (with reward / loss based supervised learning) helps to reduce the sample complexity with preference feedback

- **Emotions and Feeling are often hard to quantify in numbers:** Toxicity, friendliness (tone of writing), individuals writing style, etc.

Reward design, misspecification & hacking



$$r(s_t, a_t) = -\|\vec{x}_t - \vec{x}_d\|^2$$

(Reward is a form of "*Minimize distance to goal*")

https://bair.berkeley.edu/blog/2022/04/29/reward-reports/

Reward design, misspecification & hacking

- Paperclip fallacy (Bostrom’03)

- Goodhart's Law: "*When a measure becomes a target, it ceases to be a good measure.*“

- Eliciting preferences over trajectories is arguably more natural than cooking up a (potentially misspecified and hackable) reward function

Reward design, misspecification & hacking

- Paperclip fallacy (Bostrom’03)

- Goodhart's Law: "*When a measure becomes* … *be a good measure.*“

Demo: RL with preferences

- Eliciting preferences over trajectories is arguably more natural than cooking up a (potentially misspecified and hackable) reward function

Tournaments = “Nature” providing preferences

Elo rating

- The Elo rating of a player is an estimate of the parameter of a preference model (Bradley-Terry-Luce), computed using pairwise “preferences” (win/loss/draw)

Tournaments = “Nature” providing preferences

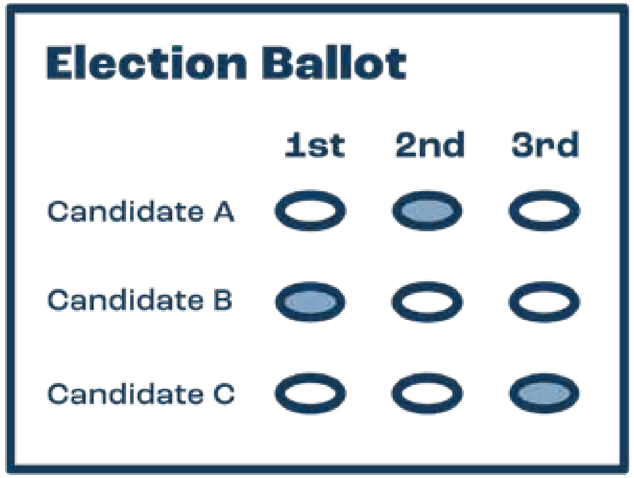
Demo: ELO ratings from chess games

Elo rating

- The Elo rating of a player is an estimate of the parameter of a preference model (Bradley-Terry-Luce), computed using pairwise “preferences” (win/loss/draw)

Voting = Entire populations providing preferences

A universally agreed-upon method to collect feedback about candidates



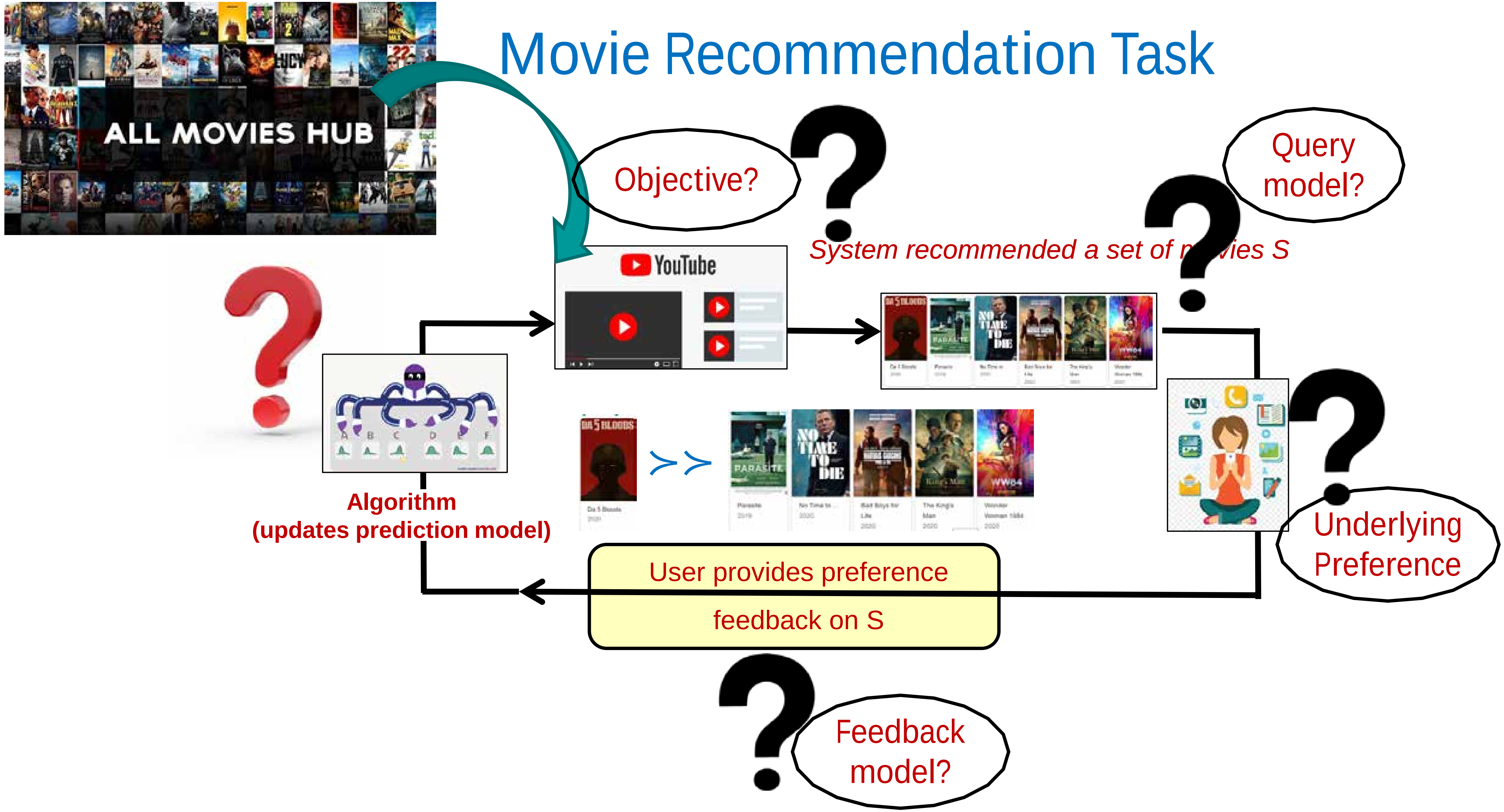
Part – II

Inferences with **Preference Learning** [Technicalities]

Outline

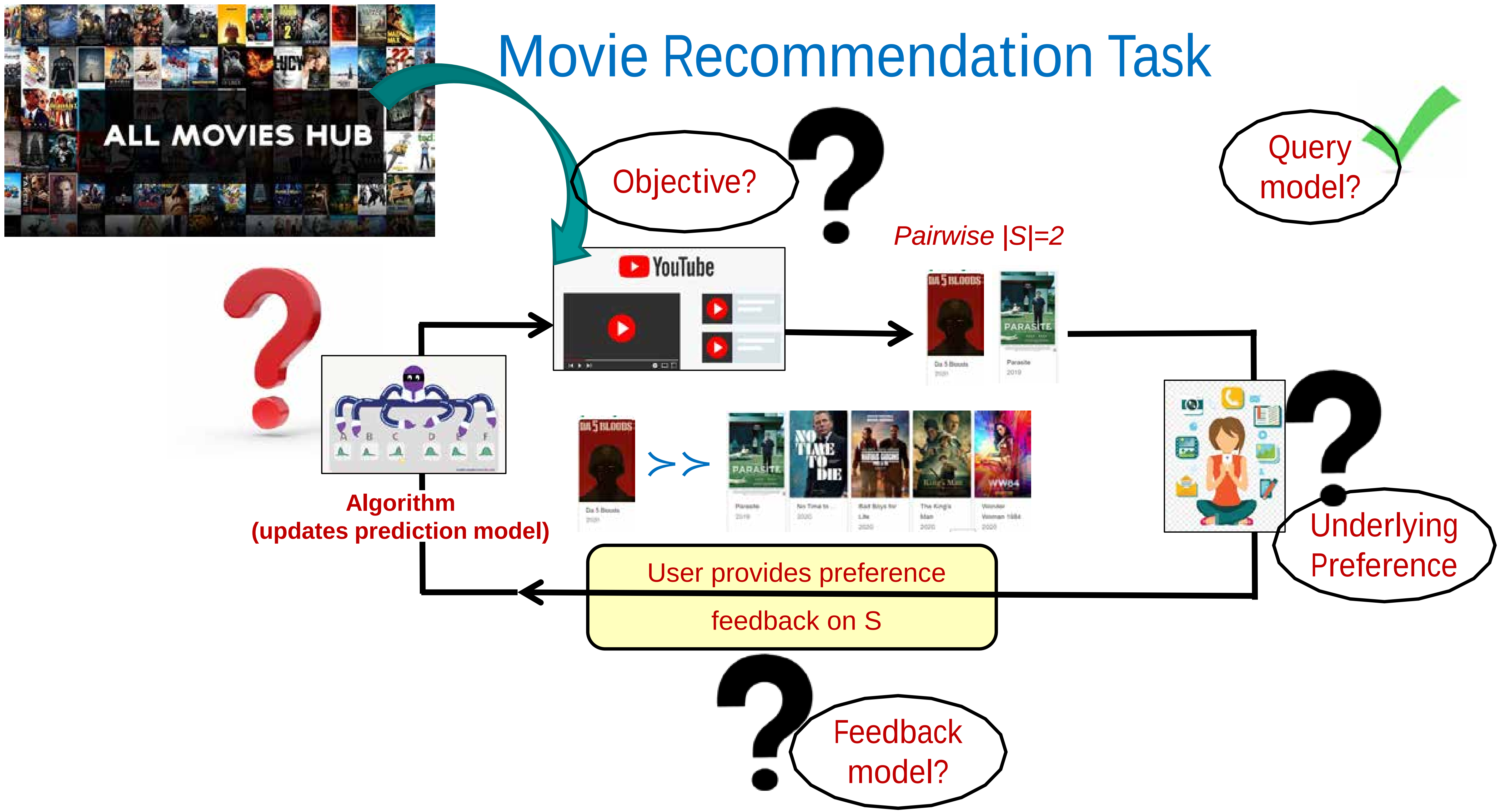
- Motivation: **Learning from Preference**
- Preference Models: **Representation of Preferences**
- Inference from Preferences: **PAC Objectives**
- Handling **Large** Decision Spaces
- **Advanced topics** in Preference Learning
- **PbRL as RLHF:** Preference based Reinforcement Learning
- Open Problems & Beyond

Let’s understand through a case study



Movie Recommendation Task
ALL MOVIES HUB
Objective?
Query model?
YouTube
System recommended a set of movies S
Algorithm
(updates prediction model)
>>
Underlying Preference
User provides preference feedback on S
Feedback model?

Movie Recommendation Task



Movie Recommendation Task

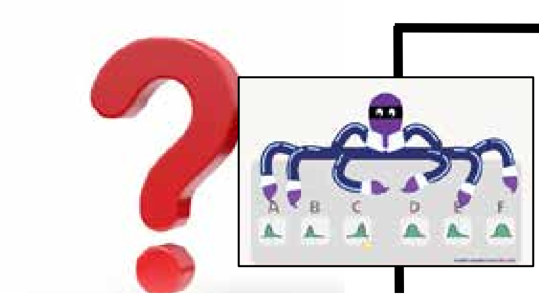
20 Movies

Objective?

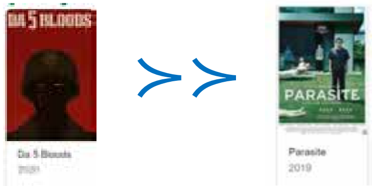
"Best" Movie

Pairwise |S|=2

Query model?



Algorithm
(updates prediction model)



Underlying Preference

Sample (Query) Complexity ?

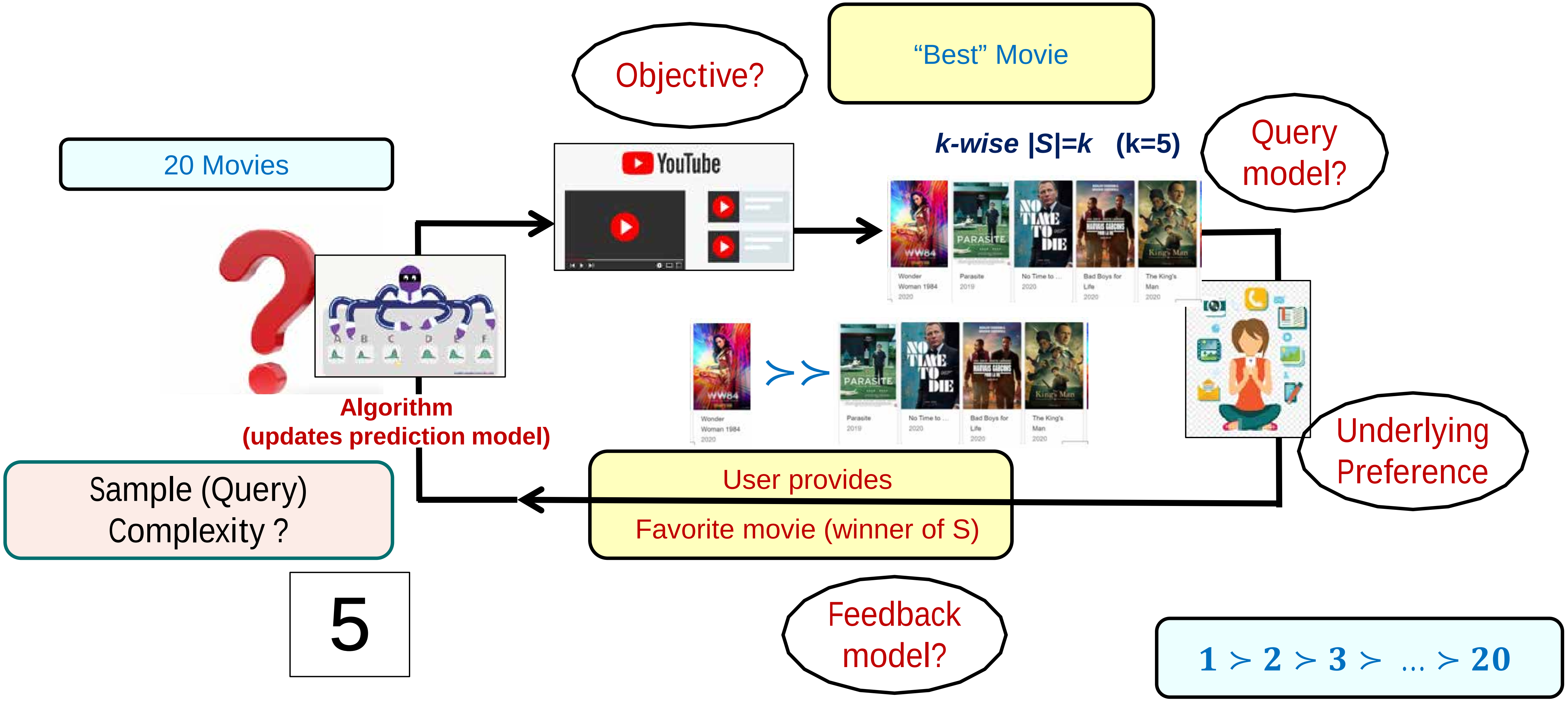
User provides pairwise preference

19

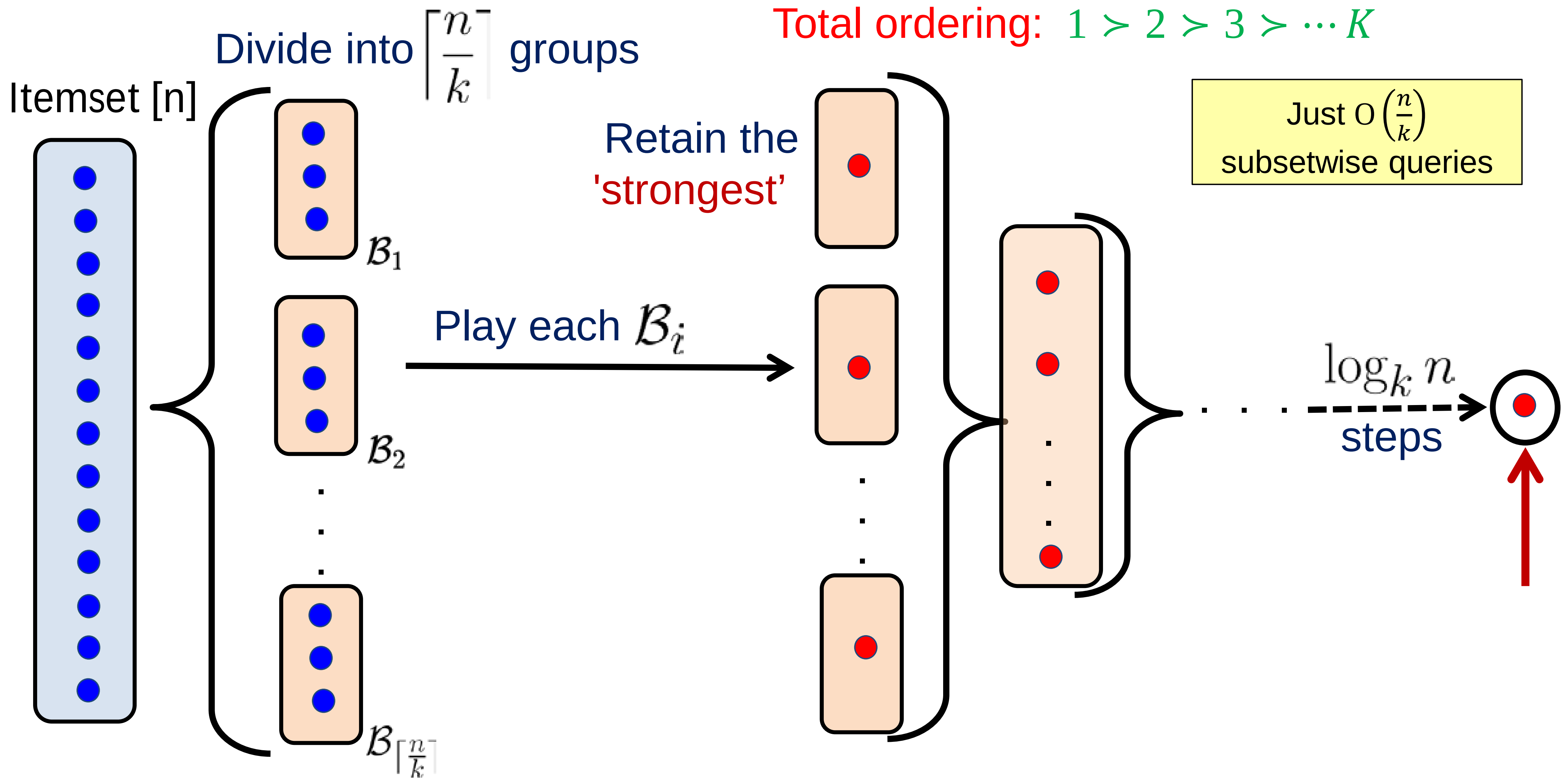
Feedback model?

$1 \succ 2 \succ 3 \succ \ldots \succ 20$

Movie Recommendation Task



“k-Subsetwise” queries lead to faster learning rates!



---- Tradeoffs ---

Assuming n movies

Query model	Feedback model	Objective	Sample-Complexity
2 (pairwise)	winner	winner	n-1
k (k-wise)	winner	winner	$\Theta\left(\frac{n}{k}\right)$
2 (pairwise)	winner	full ranking	$\Theta(n \log n)$

Jamieson, Nowak. Active Ranking using Pairwise Comparisons. NeurIPS 11

---- Tradeoffs ---

Assuming n movies

Query model	Feedback model	Objective	Sample-Complexity
2 (pairwise)	winner	winner	n-1
k (k-wise)	winner	winner	$\Theta\left(\frac{n}{k}\right)$
2 (pairwise)	winner	full ranking	$\Theta(n \log n)$
k (k-wise)	winner	full ranking	$\Theta(n \log n)$
k (k-wise)	full-k-rank	full ranking	$\Theta\left(\frac{n \log n}{k-1}\right)$
k (k-wise)	top-m rank	winner	?
k (k-wise)	top-k rank	full ranking	?

Corman et al. Introduction to Algorithms. 1989

Summary

Sample Efficient algorithm for different noiseless preference feedback.

Towards realistic settings...

Movie Recommendation Task

n could be 20 million !!

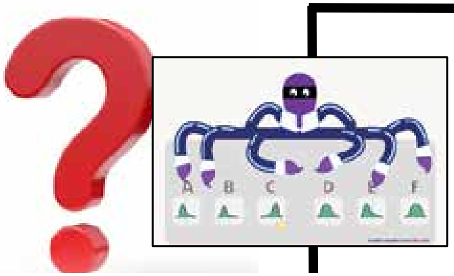
n Movies

Objective?

"Best" Movie

Query model?

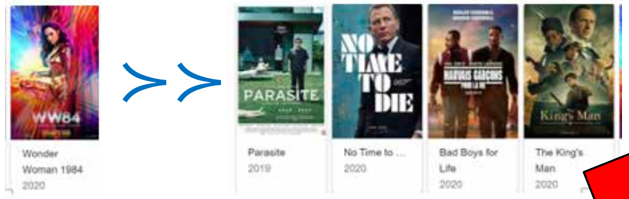
k-wise* |*S*|=*k **(k=5)**



Algorithm
(updates prediction model)

Sample (Query) Complexity ?

5



User provides

Favorite movie (winner of S)

Representation of Preferences !!

Underlying Preference

Feedback model?

$1 \succ 2 \succ 3 \succ \ldots \succ n$

Mathematical Representation of Preferences

Ranking Representation with 2-D Preference Matrix

Humans  are noisy, dynamic, impatient!

$1 \succ 2 \succ 3 \succ \ldots \succ 20$

1(i beats j)

Choice-j

Choice-i

	1	2	3	...	20
1	0.5	1	1	...	1
2	0	0.5	1	...	1
3	0	0	0.5	...	1
⋮	⋮	⋮	⋮	⋮	⋮
20	0	0	0	...	0.5

Simple Representation: 2-D Preference Matrix

~~1 > 2 > 3 > … > 20~~

- Noisy human feedback
- Changes over time
- Aggregated across users!

Prob(i beats j) → P(i,j)

Choice-2 (columns); Choice-1 (rows)

	1	2	3	…	20
1	0.5	0.53	0.54	…	0.6
2	0.47	0.5	0.53	…	0.61
3	0.46	0.47	0.5	…	0.57
4	:	:	:	:	:
20	0.4	0.39	0.43	…	0.5

Preference Modeling Challenges:

Subset-wise preference matrix

	1	2	3	...	#out-comes
S_1	0.13	0.01	0.05	…	0.22
S_2	0.27	0.12	0.03	…	0.19
S_3	0.04	0.11	0.05	…	0.23
...	…	…	…	…	…
$S_{\binom{n}{k}}$	0.23	0.19	0.03	0.19	0.24

1. Choice modeling

   Probabilistic modeling of feedback $\boldsymbol{a}$ in set $\boldsymbol{S} := \boldsymbol{P}(\boldsymbol{a}|\boldsymbol{S})$

2. Combinatorial structure:

   Number of parameters: $\binom{n}{k}$ or n^k --- Combinatorially large!!

3. How to express *relative* utilities of arms within subsets?

Discrete Random Utility based Choice Models

Modelling stochastic preferences of an individual or group of items in a given context (subset)

Probability i wins in S

1 2 3 4 ... n

Random Utility Scores: X_1 X_2 X_3 X_4 ... X_n

$$X_i = \theta_i + \xi_i, \qquad s.t.\ \xi_i \overset{iid}{\sim} D,\ \theta_i > 0\ \forall i \in [n]$$

Possible choices of ξ_i:

Gaussian(0,1)
Exponential(1)
Weibull(1,1)
Uniform(-1,1)
Gamma(1,2)
Gumbel(0,1)....

$P(i|S) = \Pr(X_i > X_j\ \forall j \in [n])$ for any subset $S \subseteq [n], S \ni i$

- Plackett-Luce choice model: $P(i|S) = \dfrac{e^{\theta_i}}{\sum_{j \in S} e^{\theta_j}}$
- Probit: Gaussian noise
- Other Choice models: Mallows, Nested GEV etc.

Azari et al., **Random utility theory for social choice**. Neural Information Processing Systems (NeurIPS), 2012.

Let us work with PL model:

Modelling stochastic preferences of an individual or group of items in a given context (subset)

- Plackett-Luce choice model:

Parameters: $\boldsymbol{\theta} = (\theta_1 \geq \theta_2 \geq \ldots \geq \theta_n),\ \theta_i > 0\ \forall i \in [n]$

$$Pr(i|S) = \frac{\theta_i}{\sum_{j \in S} \theta_j}$$ for any subset $S \subseteq [n],\ S \ni i$

Parameter Reduction!!

n^k à n

Just n parameters!

Azari et al., **Random utility theory for social choice**. Neural Information Processing Systems (NeurIPS), 2012.

PL Model: Winner & Top-Rank Feedback

Modelling stochastic preferences of an individual or group of items in a given context (subset)

- Plackett-Luce choice model:

Parameters: $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n),\ \theta_i > 0\ \forall i \in [n]$

$$Pr(i|S) = \frac{\theta_i}{\sum_{j \in S} \theta_j}$$ for any subset $S \subseteq [n], S \ni i$

Parameter Reduction!!

n^k à n

Just n parameters!

- Type of PL feedback: General Top-m Ranking: $(\sigma_1, \sigma_2, \ldots, \sigma_m) \in \Sigma_S^m$

$$Pr(\boldsymbol{\sigma} = \sigma|S) = \prod_{i=1}^{m} \frac{\theta_{\sigma^{-1}(i)}}{\sum_{j \in S \setminus \sigma^{-1}(1:i-1)} \theta_j}$$

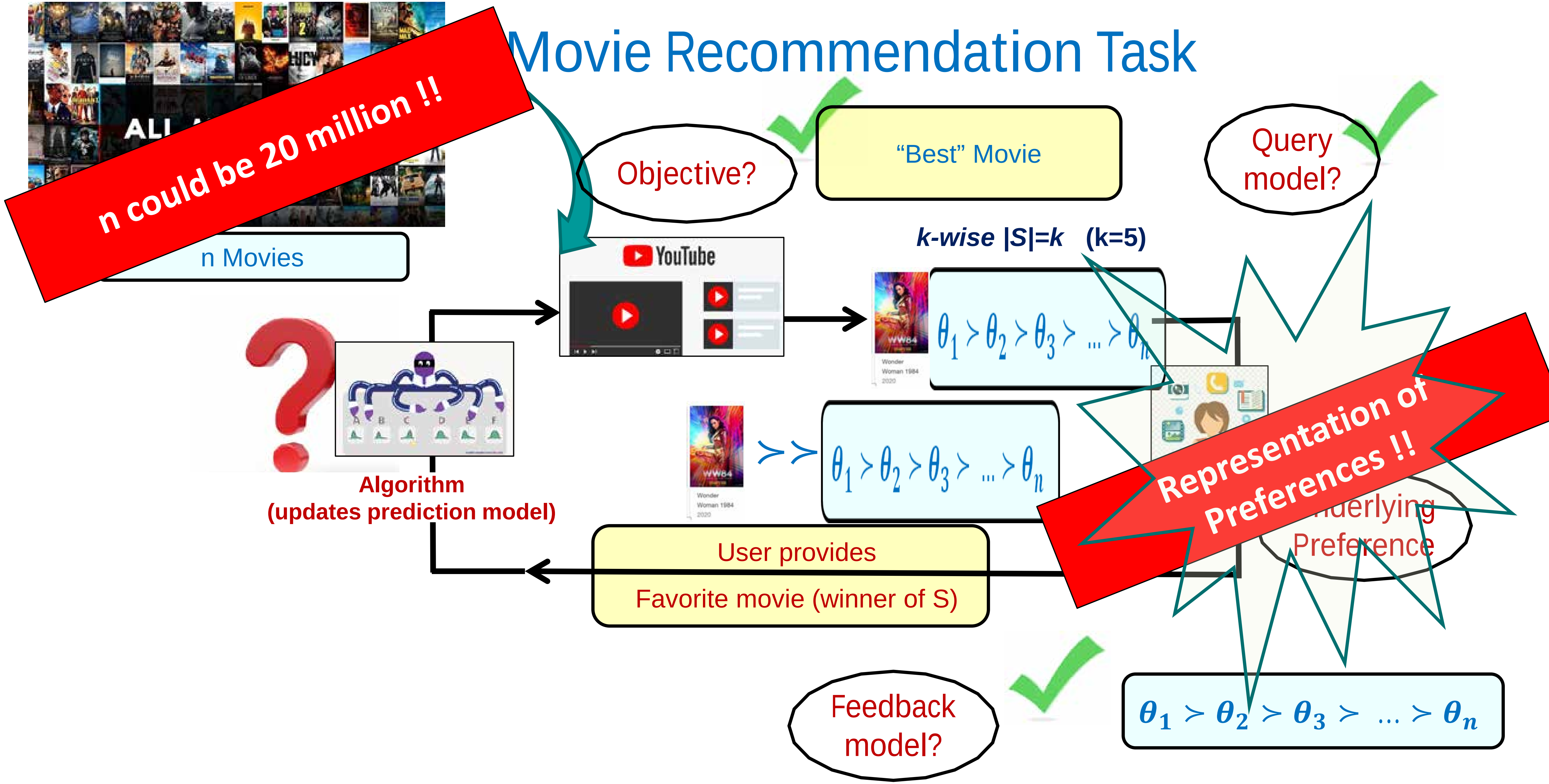
Example: For subset $S = \{a, b, c, d\}$ (k=4)

-- Top-m ranking feedback (m=2): $b \succ a$

-- Full ranking feedback (m=4): $b \succ a \succ c \succ d$

Azari et al., **Random utility theory for social choice**. Neural Information Processing Systems (NeurIPS), 2012.

Movie Recommendation Task



Problem: Find Rank-1 (Winner) item with k-wise PL Feedback

Suppose Parameters: $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_n),\ \theta_i > 0\ \forall i \in [n]$

Objective: (ϵ, δ)-PAC Best Item

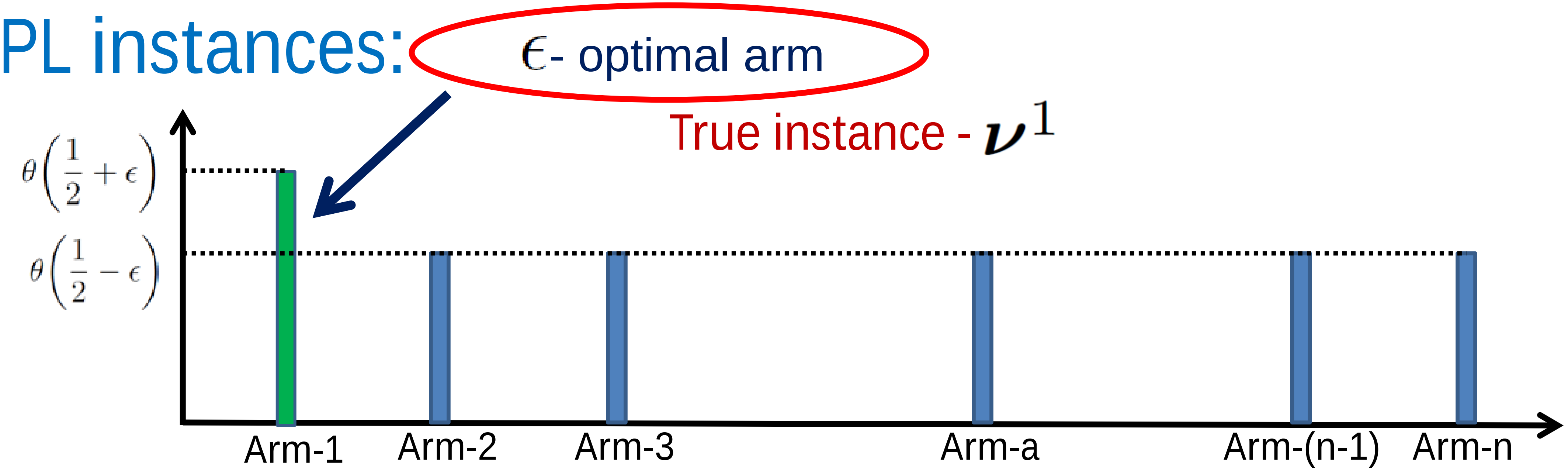
Ideally $\epsilon = 0, \delta = 1$

Output item ***i*** such that: $Pr(\theta_1 - \theta_i > \epsilon) < \delta$

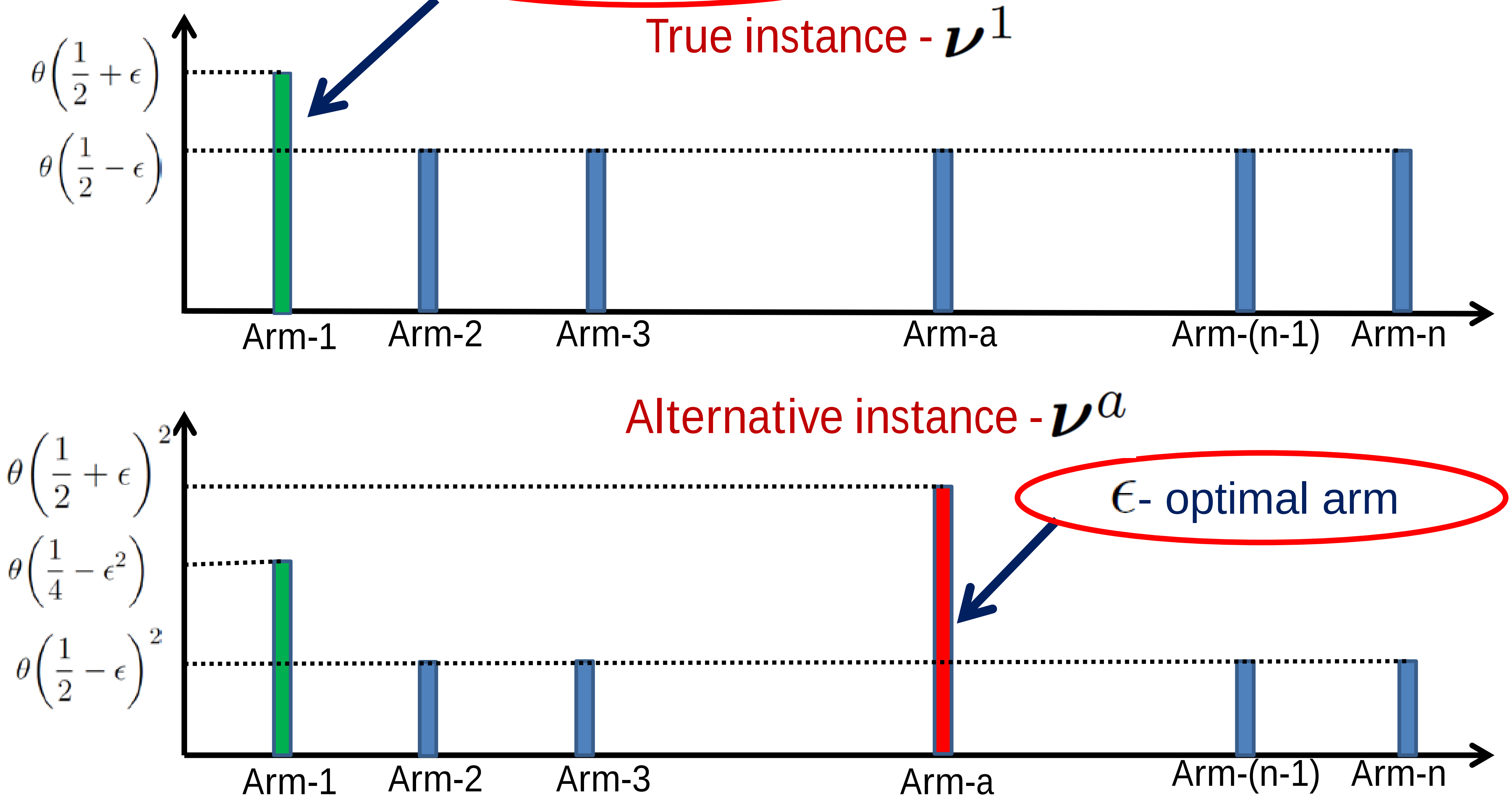
with minimum possible #samples (rounds)

PL model Sample Complexity Lower Bound Analysis

PL instances:



PL instances:



Fundamental Inequality (Kaufmann et al. 2016):

- Consider two MAB instances on n arms: ν and ν'. Arm set: $\mathcal{A} = [n]$
- ν_i reward distribution of arm i for ν (similarly ν_i' for ν')
- $N_i(\tau)$ number of plays of arm i during any finite stopping time τ

$$\sum_{i \in \mathcal{A}} \mathbf{E}_\nu[N_i(\tau)] KL(\nu_i, \nu_i') \geq \sup_{\mathcal{E} \in \mathcal{F}_\tau} kl(Pr_\nu(\mathcal{E}), Pr_{\nu'}(\mathcal{E}))$$

where $kl(x, y) := x \log(\frac{x}{y}) + (1 - x) \log(\frac{1-x}{1-y})$

$\mathcal{E}$: Any event under sigma-algebra of the algorithm's trajectory

Kaufmann et al., On the complexity of best-arm identification in multi-armed bandit models. JMLR 2016.

Lower Bound Analysis:

(Kaufmann et al. 2016)

$$\sum_{i \in \mathcal{A}} \mathbf{E}_\nu[N_i(\tau)] KL(\nu_i, \nu_i') \geq \sup_{\mathcal{E} \in \mathcal{F}_\tau} kl(Pr_\nu(\mathcal{E}), Pr_{\nu'}(\mathcal{E}))$$

- Arm set : $\mathcal{A} = \{S = (S(1), \ldots S(k)) \subseteq [n] \mid S(i) < S(j), \forall i < j\}$
- $\mathcal{E}_0$: Event that Algorithm (A) returns item-1
- $Pr_{\boldsymbol{\nu}^1}(\mathcal{E}_0) > 1 - \delta$, and $Pr_{\boldsymbol{\nu}^a}(\mathcal{E}_0) < \delta$
- LHS: $kl(Pr_{\boldsymbol{\nu}^1}(\mathcal{E}_0), Pr_{\boldsymbol{\nu}^a}(\mathcal{E}_0)) \geq kl(1 - \delta, \delta) \geq \ln \frac{1}{2.4\delta}$,
- RHS: $KL(\boldsymbol{\nu}_S^1, \boldsymbol{\nu}_S^a) \leq \frac{m}{k} 256\epsilon^2$
- Result follows further using: $\tau_A = \sum_{S \in \mathcal{A}} [N_S(\tau_A)]$

Result Overview: (ϵ, δ)-Sample Complexity

1. Sample Complexity Lower Bound:

For any $\epsilon \in (0, \frac{1}{\sqrt{8}}]$ and $\delta \in (0, 1]$ and any (ϵ, δ)-PAC algorithm A, there exist an instance of the PL model where A requires a sample complexity of at least

$$\Omega\left(\frac{n}{m\epsilon^2}\, log \frac{1}{\delta}\right) \text{ rounds}$$

Essentially 'independent' of k !

Reduces with m.

Saha, A. Gopalan. From PAC to Instance-Optimal Sample Complexity in the Plackett-Luce Model. ICML, 2020.

---- Tradeoffs ---

Assuming n movies

Query model	Feedback model	Objective	Sample-Complexity
k (k-wise)	top-k rank	full ranking	?

Intuition

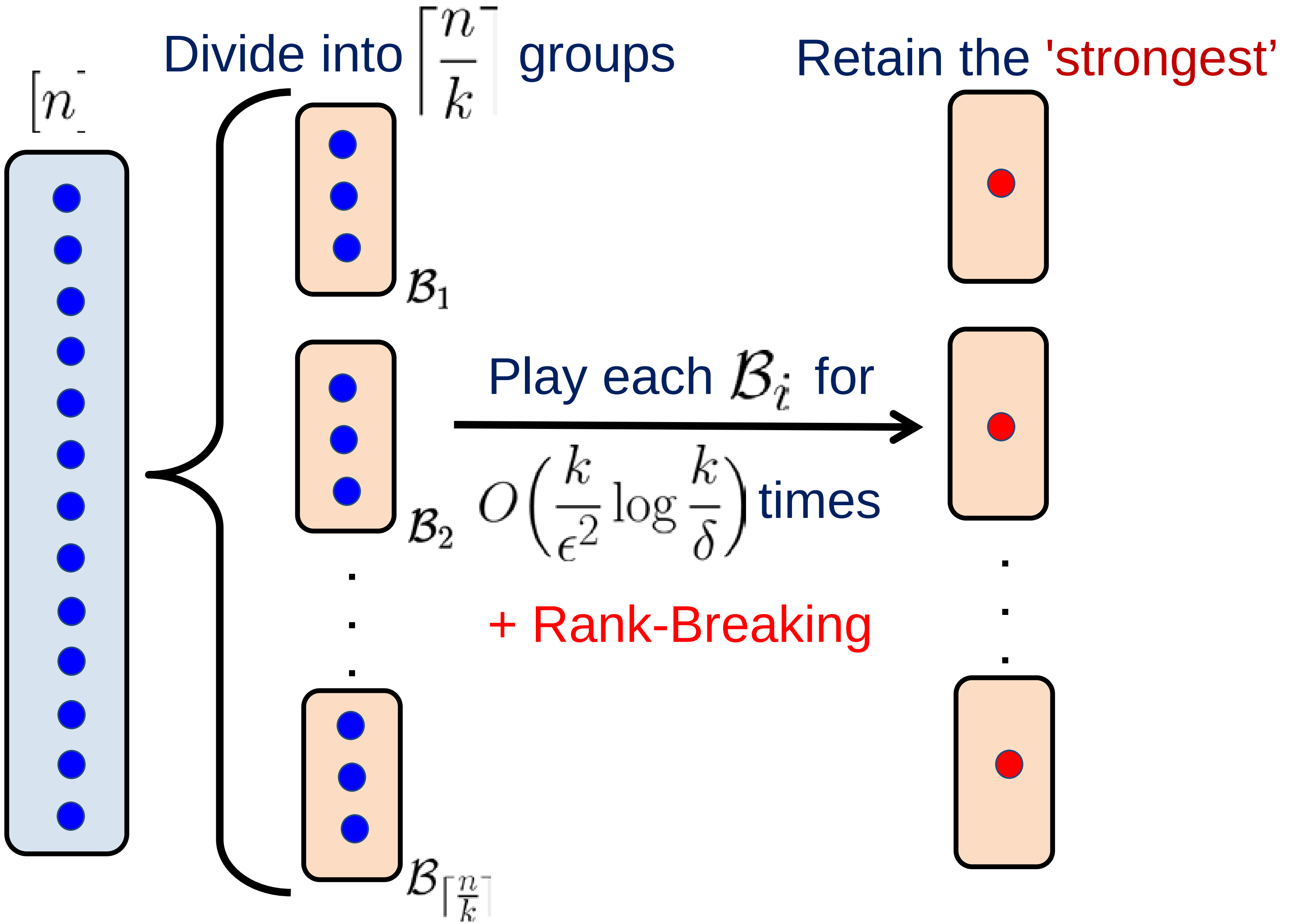
Larger subsets can *cover* more items

but

It is also *harder* for the best item to stand out

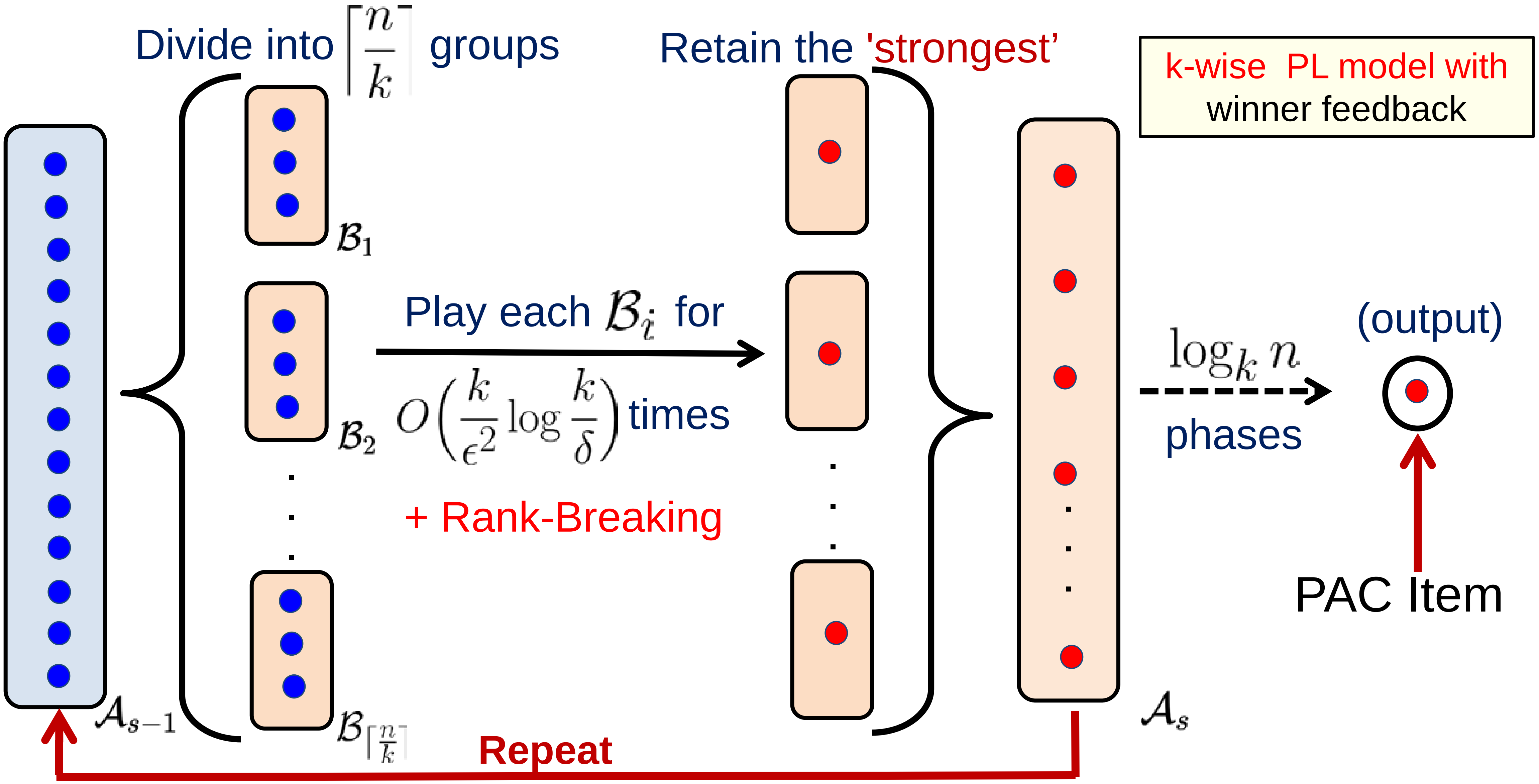
But, Algorithm?

Proposed Algorithm-1: Divide and Battle (DnB)



k-wise PL model with winner feedback

Proposed Algorithm-1: Divide and Battle (DnB)



A Key Concept: Rank Breaking (RB)

Idea of extracting pairwise preferences from subset-wise feedback

Example: Consider a subset $S = \{a, b, c, d\}$ of size (k = 4)

- Ø Upon top-m ranking feedback (m=2): $b \succ a \succ \{c, d\}$

Rank-Breaking à $(b, a \succ c), (b, a \succ d)$ and $(b \succ a)$

- Ø Upon full ranking feedback (m=4): $b \succ a \succ c \succ d$

Rank-Breaking à $\{(b \succ a), (b \succ c), (b \succ d), (a \succ c), (a \succ d), (c \succ d)\}$

'Strongest' à Winner of maximum no. of Pairwise Duels

Hajek, Oh, Xu. **Minimax-optimal Inference from Partial Rankings**. NeurIPS, 2014.

Key Lemma (Deviations of pairwise win-probability estimates for PL model):

Assume,

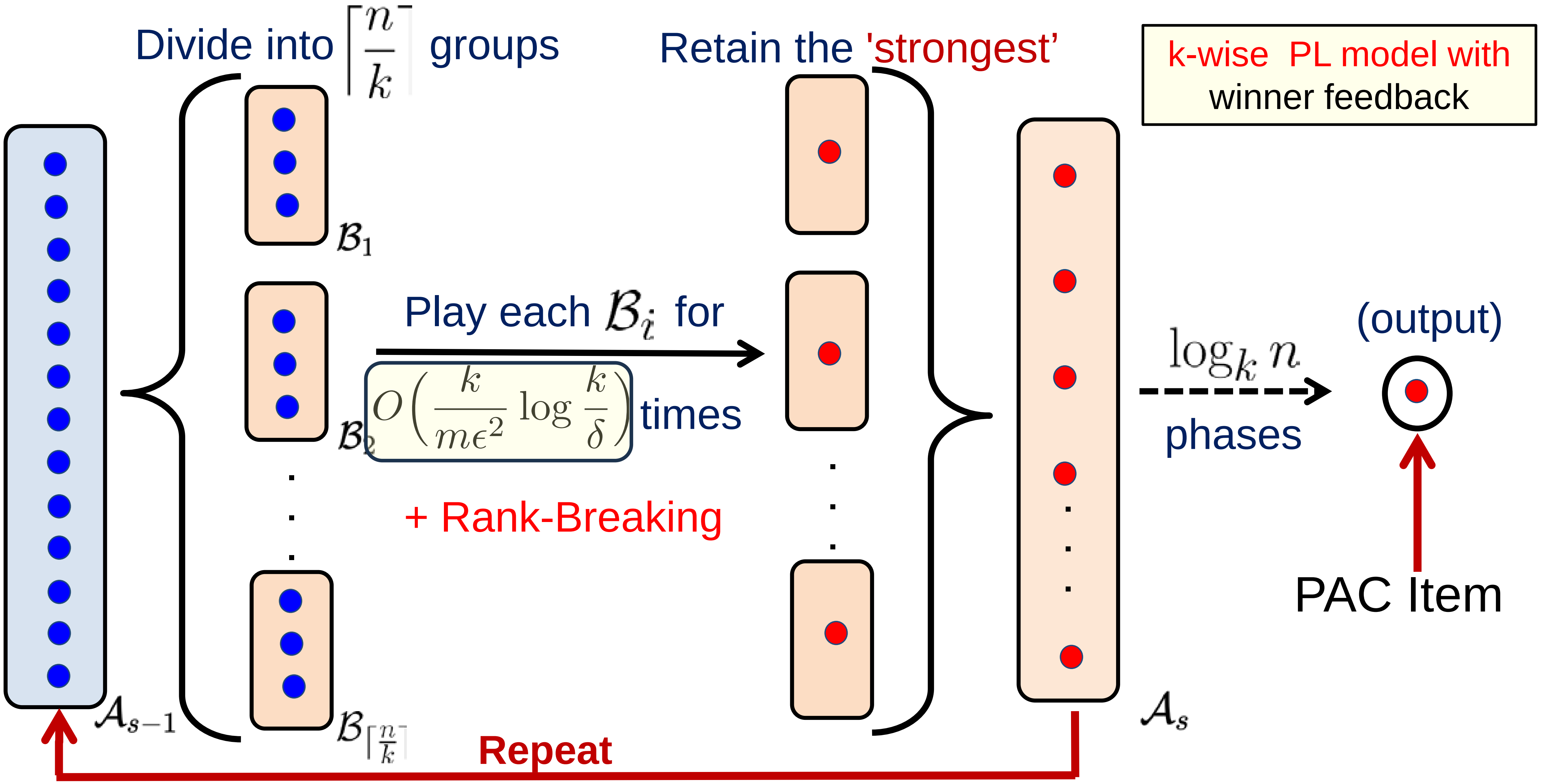
- $S_1, \ldots, S_T$ *be a sequence of (possibly random) subsets*
- S_t *depends only on* $S_1, \ldots, S_{t-1}$
- i_t *is distributed as the Plackett-Luce winner of the subset* S_t

Then: $\hat{p}_{ij}$ p_{ij}

$$Pr\left(\frac{n_i(T)}{n_{ij}(T)} - \frac{\theta_i}{\theta_i + \theta_j} \geq \eta, n_{ij}(T) \geq v\right) \vee Pr\left(\frac{n_i(T)}{n_{ij}(T)} - \frac{\theta_i}{\theta_i + \theta_j} \leq -\eta, n_{ij}(T) \geq v\right) \leq e^{-2v\eta^2}$$

where $n_i(T) = \sum_{t=1}^{T} \mathbf{1}(i_t = i)$ and $n_{ij}(T) = \sum_{t=1}^{T} \mathbf{1}(\{i_t \in \{i,j\}\})$

Proposed Algorithm-1: Divide and Battle (DnB)



Summary of Results on (ϵ, δ)-Sample Complexity in PL

1. Sample Complexity Lower Bound:

For any $\epsilon \in (0, \frac{1}{\sqrt{8}}]$ and $\delta \in (0, 1]$ and any (ϵ, δ)-PAC algorithm A, there exist an instance of the PL model where A requires a sample complexity of at least

$$\Omega\left(\frac{n}{m\,\epsilon^2}\; log \frac{1}{\delta}\right)$$ subsetwise queries.

Essentially 'independent' of k
(no improvement with subsetsize!!)

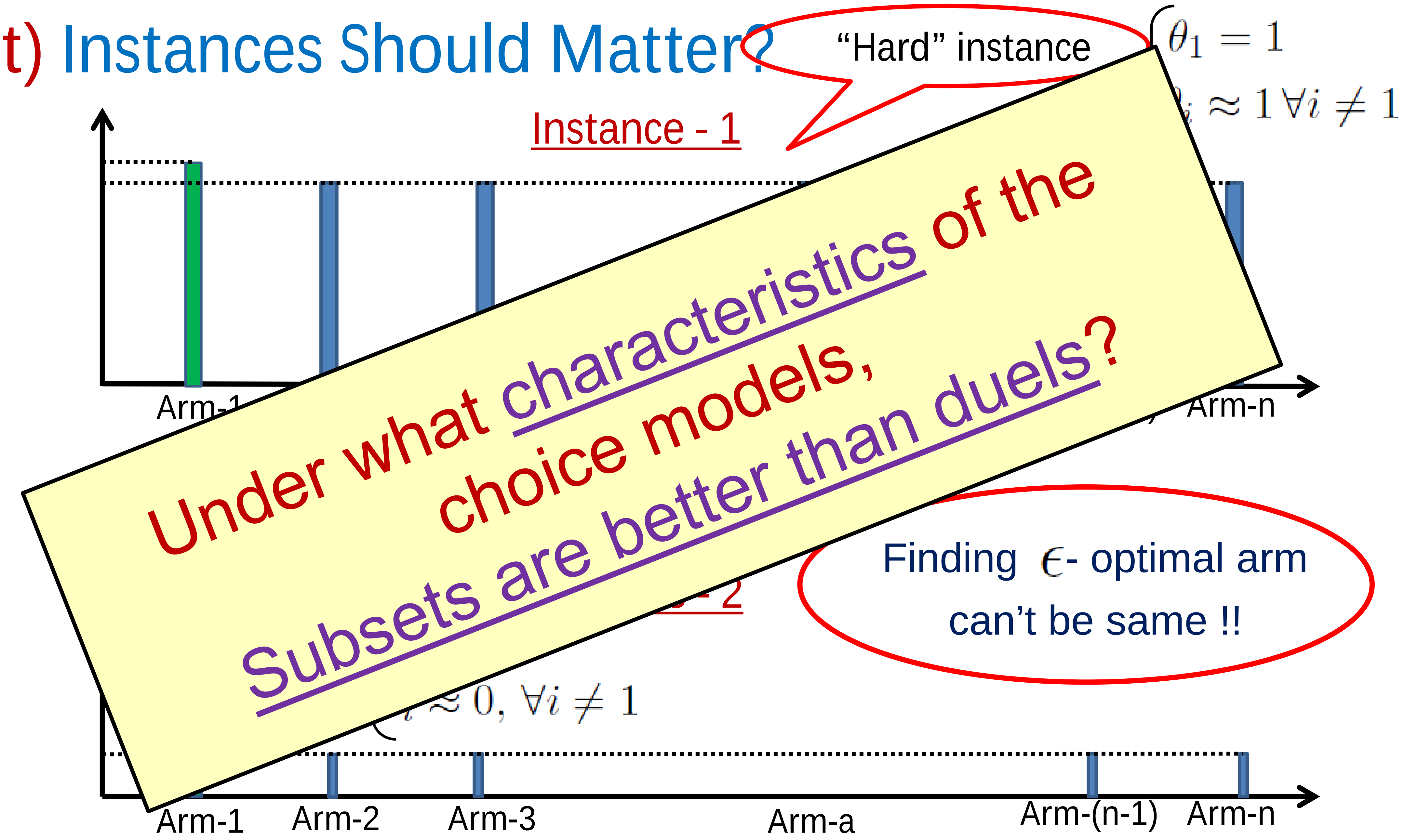
But improves with m
(length of rank-ordered feedback)

2. DnB algorithm takes: $O\left(\frac{n}{m\,\epsilon^2}\; log \frac{k}{\delta}\right)$ rounds.

-- Algorithm: *Divide & Battle (sequential Pairwise-RB with Elimination)*

Aadirupa Saha, Aditya Gopalan. **Best-item Learning in Random Utility Models with Subset Choices**. AISTATS, 2020

(But) Instances Should Matter?



Pure-Exploration: Instance optimal Best-Item

Instant Dependent-Sample Complexity

$\Delta_i = \theta_1 - \theta_i$ for any $i \in [n]$ (Gaps)

"Hard" instance

"Easy" instance

Lower Bound: $\Omega\left(\frac{1}{m}\sum_{i=2}^{n}\frac{\theta_i\theta_1}{\Delta_i^2}\ln\left(\frac{1}{\delta}\right) + \frac{n}{k}\ln\frac{1}{\delta}\right)$

We achieved: $O\left(\frac{\Theta_{[k]}}{k}\sum_{i=2}^{n}\max\left(1, \frac{1}{m\Delta_i^2}\right)\ln\frac{k}{\delta}\left(\ln\frac{1}{\Delta_i}\right)\right)$

$$\Theta_{[k]} = \max_{S\subseteq[n] \mid |S|=k}\sum_{i\in S}\theta_i \begin{cases} O(k) & \text{---- for "Hard" instances} \\ O(1) & \text{---- for "Easy" instances} \end{cases}$$

Saha, A. Gopalan. From PAC to Instance-Optimal Sample Complexity in the Plackett-Luce Model. ICML, 2020.

Summary

Sample Efficient algorithm for PL model with m-rank-ordered feedback.

Learning the entire Ranking ?
(PL model)

Problem Setting: (ϵ, δ)-PAC-Ranking

True-ranking: $\sigma^* \leftarrow \text{argsort}(\theta_1, \theta_2, \ldots \theta_n)$

Objective: Predict a full Ranking ($\boldsymbol{\sigma}$):

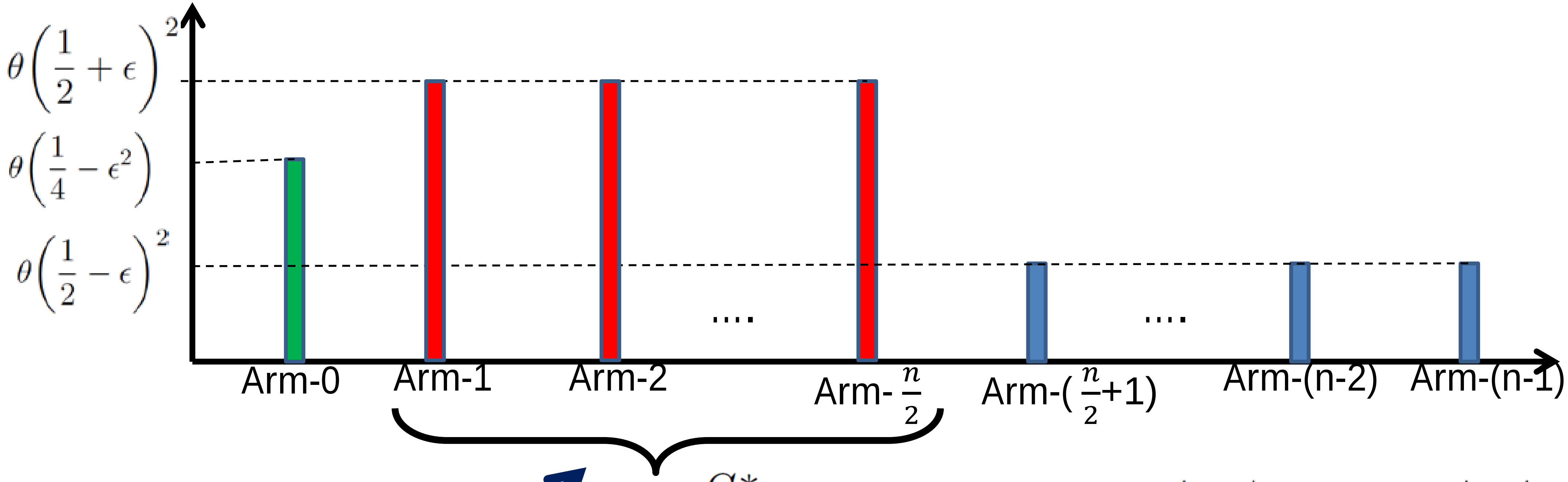
$$Pr\left(\forall i, j \in [n] \mid \theta_i > \theta_j + \epsilon, \text{ then } \sigma(i) < \sigma(j)\right) > (1 - \delta)$$

with minimum possible #samples (rounds)

A. Lower Bound

PL instances:

True instance - ν_{S^*}



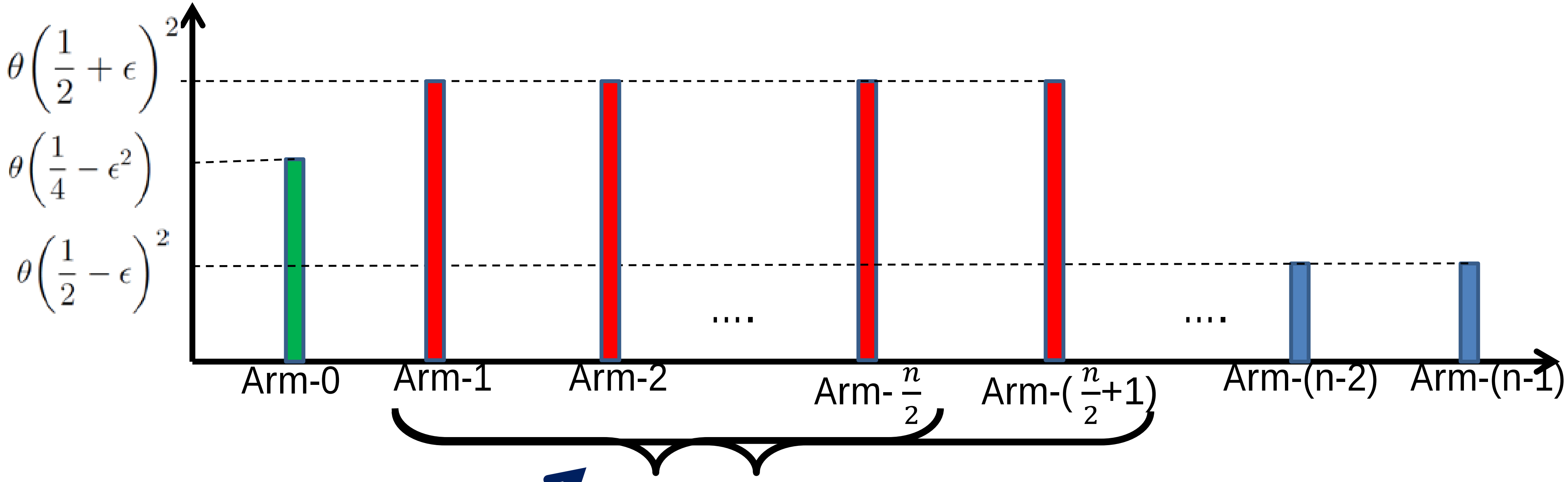
ϵ- best optimal arms

such that $|S^*| = q = \lfloor \frac{n}{2} \rfloor$

$$Pr_{S^*}\Big(\boldsymbol{\sigma}_{\mathcal{A}}(1:q+1) = S^* \cup \{0\}\Big) > 1-\delta$$

PL instances:

Alternative instance - $\nu_{\tilde{S}^*}$



ϵ- best optimal arms

$\tilde{S}^*$ such that $\tilde{S}^* = S^* \cup \{i\}$

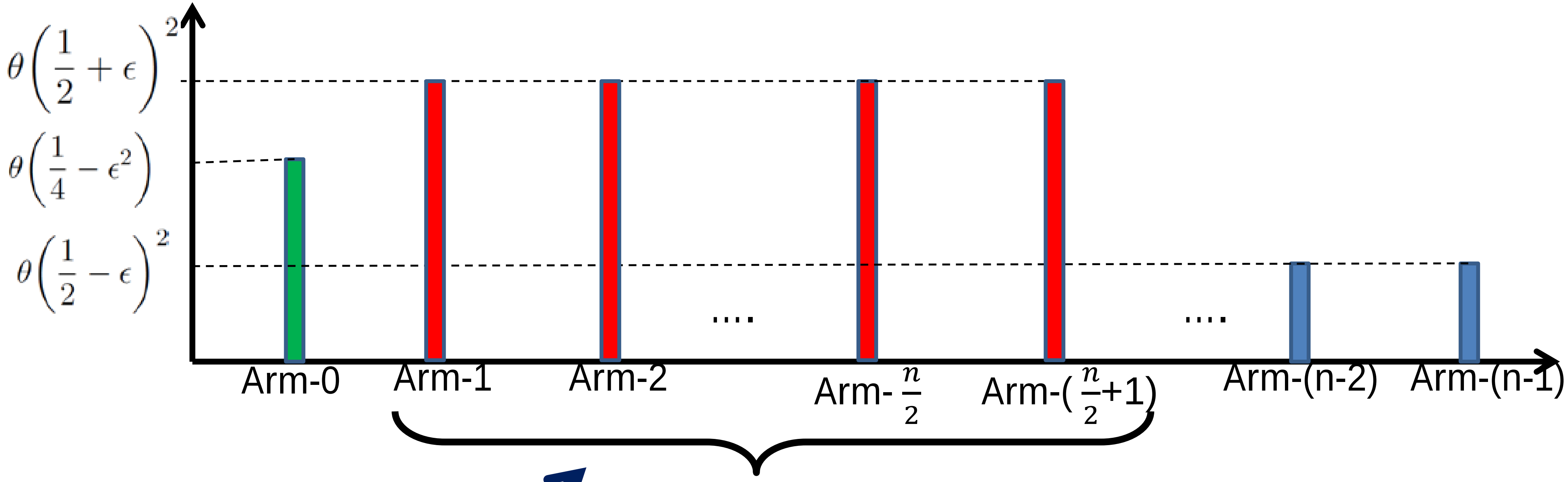
for any $i \in [n-1] \setminus S^*$

$$Pr_{\tilde{S}^*}\Big(\boldsymbol{\sigma}_{\mathcal{A}}(1:q+1) = S^* \cup \{0\}\Big) < Pr_{\tilde{S}^*}\Big(\boldsymbol{\sigma}_{\mathcal{A}}(1:q+1) \neq \tilde{S}^*\Big) < \delta$$

PL instances:

Alternative instance - $\nu_{\tilde{S}^*}$

'Label Invariance'!



$\tilde{S}^*$ such that $\tilde{S}^* = S^* \cup \{i\}$ for any $i \in [n-1] \setminus S^*$

ϵ- best optimal arms

$$Pr_{\tilde{S}^*}\Big(\boldsymbol{\sigma}_{\mathcal{A}}(1:q+1) = S^* \cup \{0\}\Big) < Pr_{\tilde{S}^*}\Big(\boldsymbol{\sigma}_{\mathcal{A}}(1:q+1) \neq \tilde{S}^*\Big) < \frac{\delta}{q}$$

Result Overview: (ϵ, δ)-Sample Complexity

1. Sample Complexity Lower Bound:

For any $\epsilon \in (0, \frac{1}{32}]$ and $\delta \in (0, 1]$ and any (ϵ, δ)-PAC algorithm A satisfying label invariance, there exist an instance of the PL model where A requires a sample complexity of at least $\Omega\left(\frac{n}{m\epsilon^2} \ln \frac{n}{4\delta}\right)$ rounds.

2. Existing results: $O\left(\frac{n}{m\epsilon^2} \ln \frac{n}{\delta}\right)$ rounds.

-- Algorithm-1: *Beat-the-Pivot*
-- Algorithm-2: *Score-and-Rank*

Again 'independent' of k !

A. Saha, A. Gopalan. Active Ranking with Subset-wise Preferences. Artificial Intelligence and Statistics (AISTATS), 2019.

Algorithm + Guarantees

Algorithm : Beat-the-Pivot (BP)

Correctness and Sample Complexity guarantee

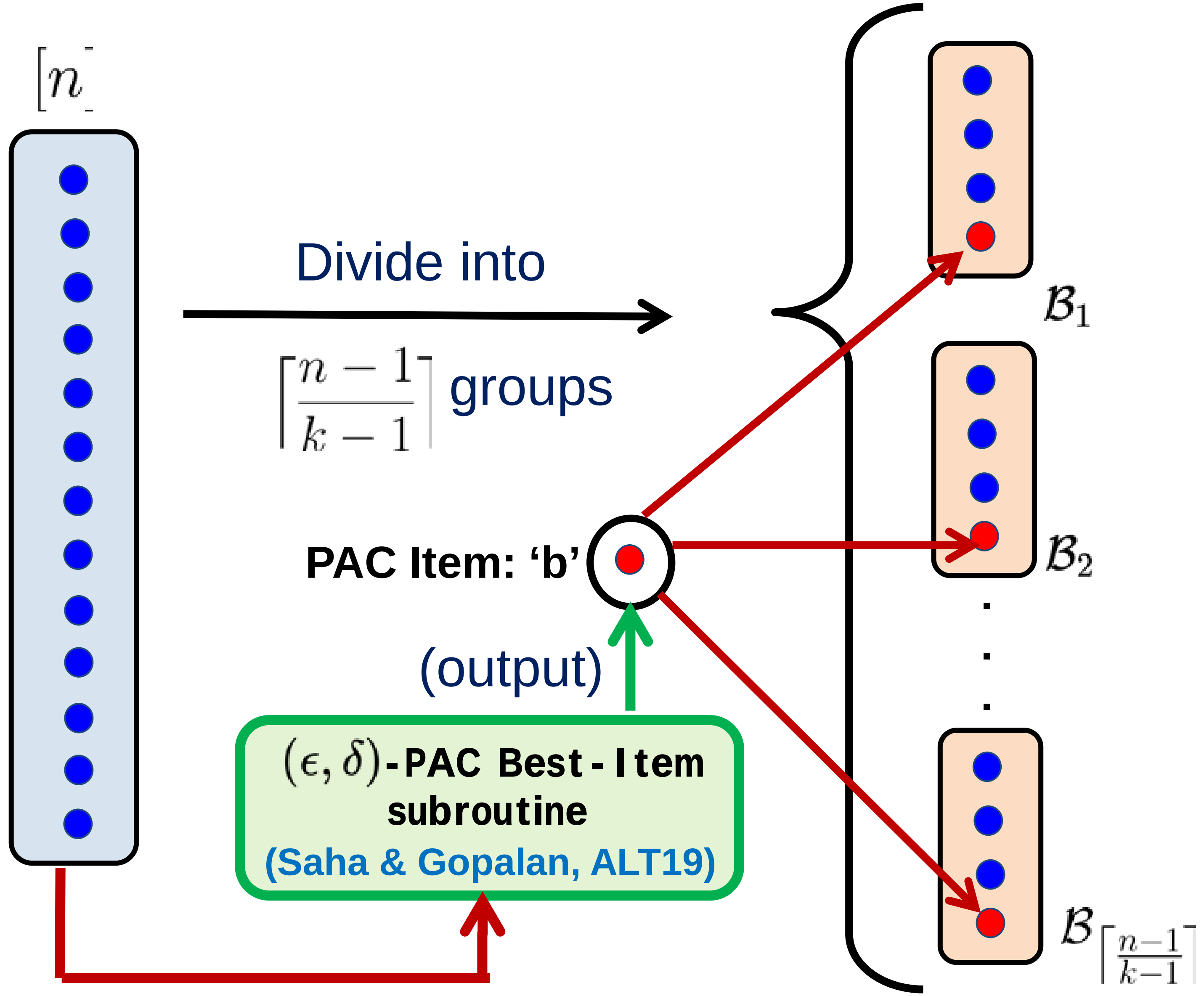
Theorem: Beat the pivot finds an (ϵ, δ)-PAC Optimal Ranking with sample complexity : $O\left(\frac{n}{m\epsilon^2} \ln \frac{n}{\delta}\right)$

Proof?

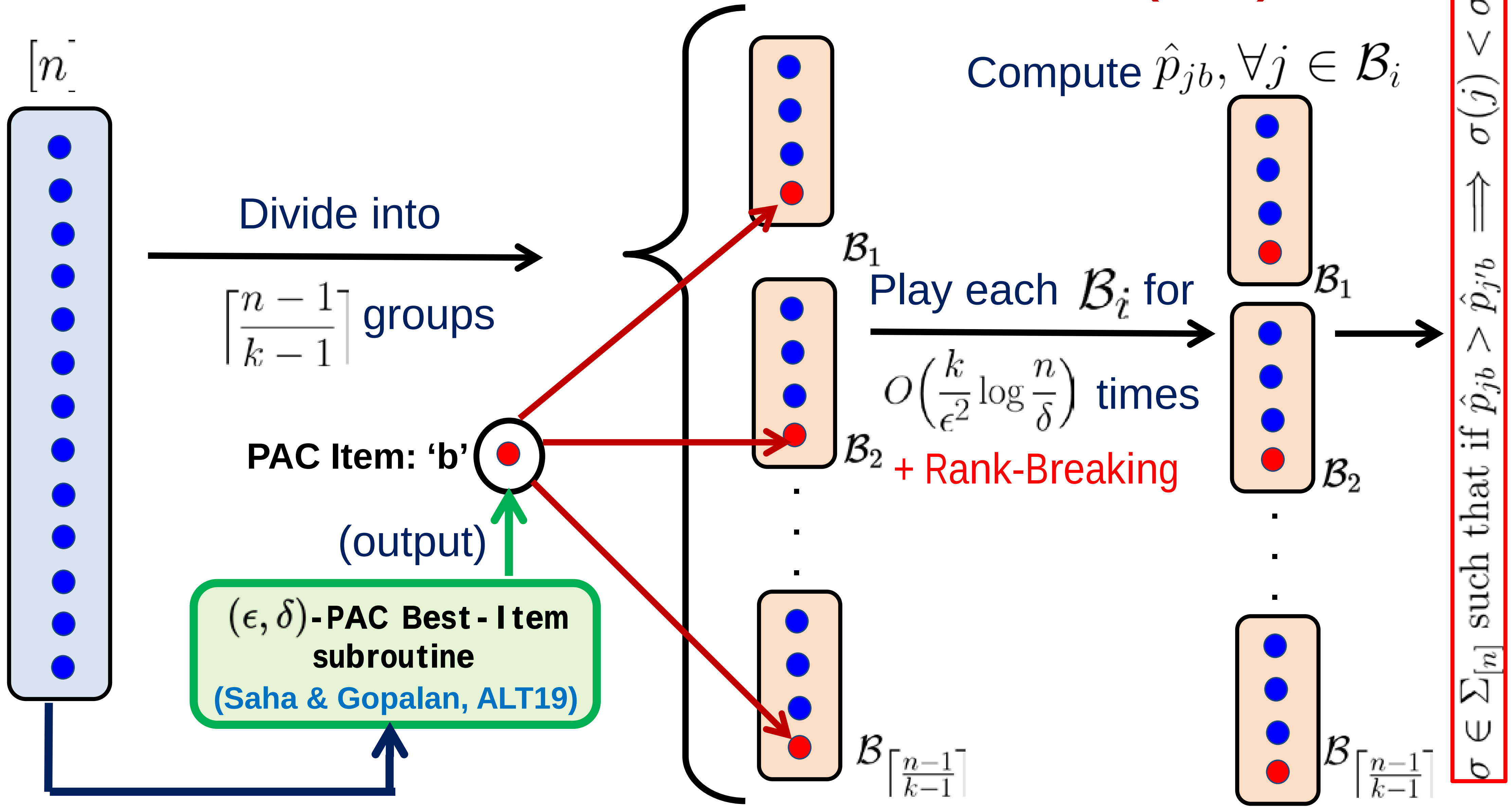
Main Idea: If $\theta_1 > \theta_2 > \ldots > \theta_n$, then for any $b \in [n], p_{1b} > p_{2b} > \ldots > p_{nb}$

Can we estimate $p_{jb}, \; \forall j \in [n]$ with high confidence?

Proposed Algorithm: Beat-the-Pivot (BP)



Proposed Algorithm: Beat-the-Pivot (BP)



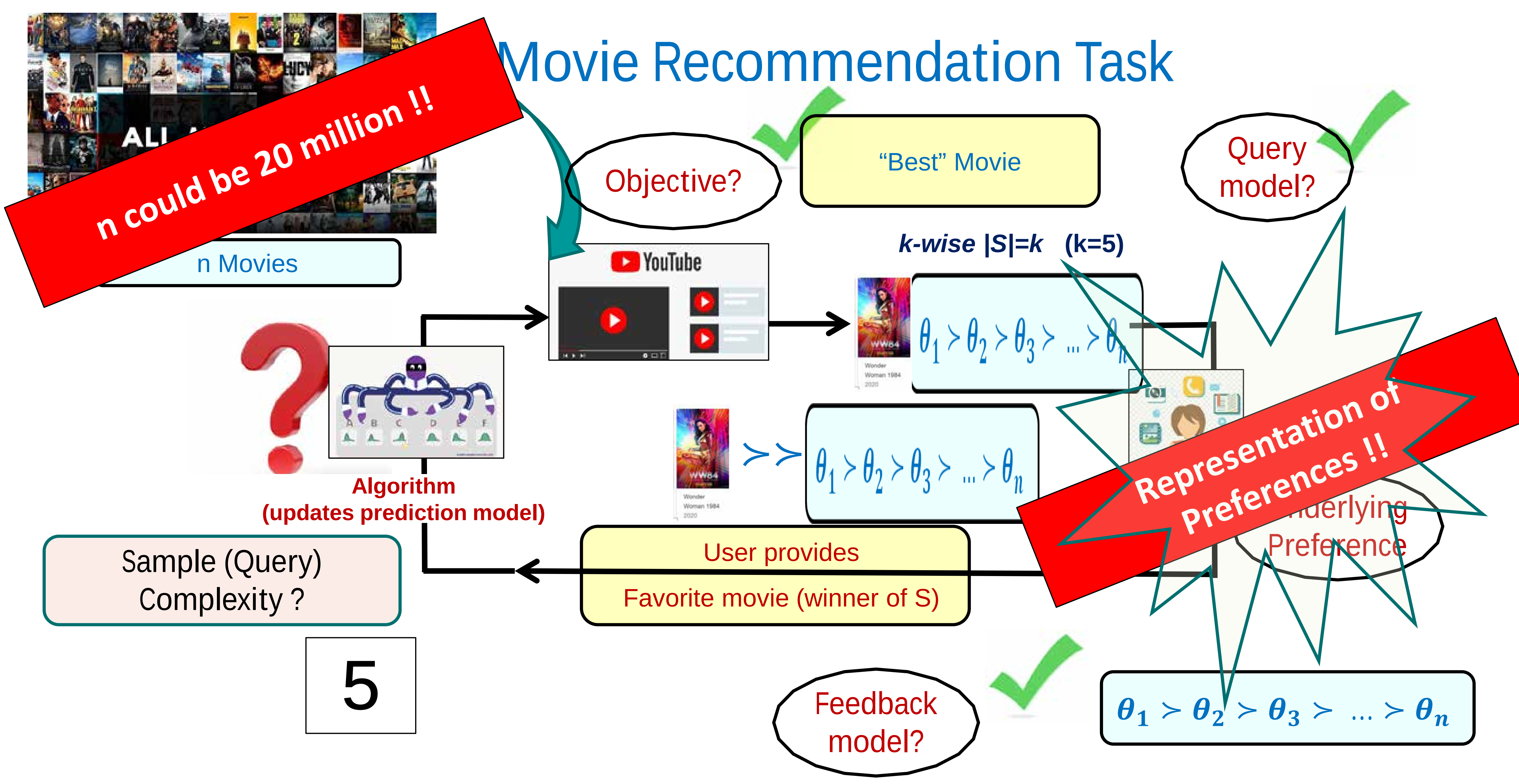
Summary

Sample Efficient “ranking algorithm” for PL model with m-rank-ordered feedback.

Outline

- Motivation: **Learning from Preference**
- Preference Models: **Representation of Preferences**
- Inference from Preferences: **PAC Objectives**
- Handling **Large** Decision Spaces
- **Advanced topics** in Preference Learning
- **PbRL as RLHF:** Preference based Reinforcement Learning
- Open Problems & Beyond

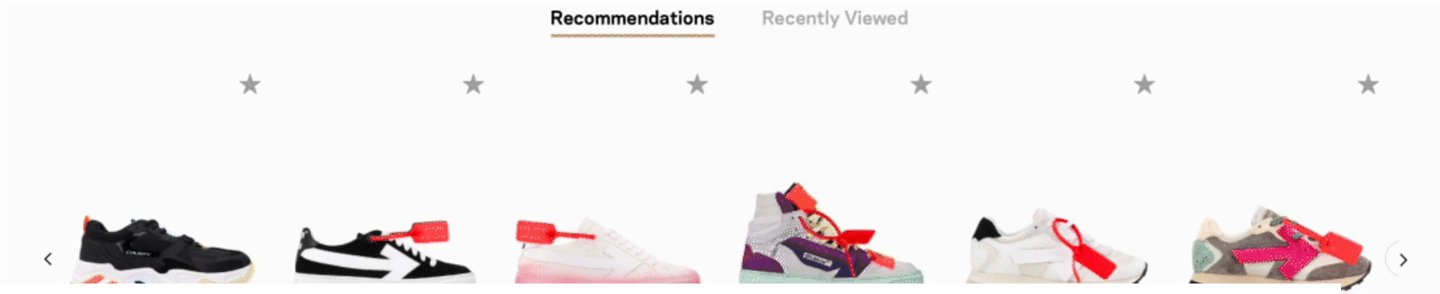
Towards more realistic settings...



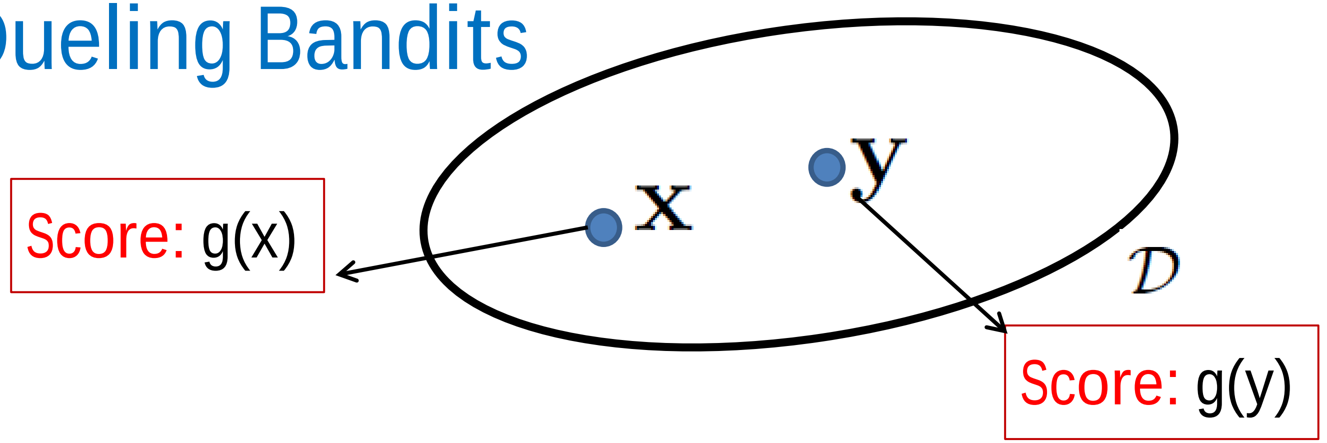
Movie Recommendation Task
n could be 20 million !!
n Movies
Objective?
"Best" Movie
Query model?
k-wise |S|=k (k=5)
$\theta_1 > \theta_2 > \theta_3 > \ldots > \theta_n$
$\succ\succ$
$\theta_1 > \theta_2 > \theta_3 > \ldots > \theta_n$
Algorithm
(updates prediction model)
Representation of Preferences !!
Underlying Preference
Sample (Query) Complexity ?
User provides
Favorite movie (winner of S)
5
Feedback model?
$\theta_1 \succ \theta_2 \succ \theta_3 \succ \ldots \succ \theta_n$

$$\#Items(n) \rightarrow \infty\,?$$

From a Practical Standpoint - Need to Exploit Item Similarities



Structured (Continuous) Dueling Bandits



Problem Setup:

Decision Space: $\mathcal{D} \subseteq \mathbb{R}^d$

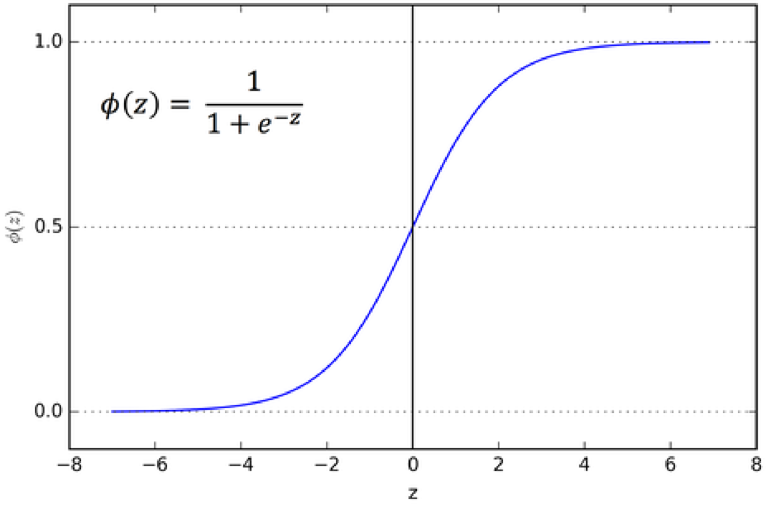
convex

$g : \mathcal{D} \mapsto \mathbb{R}$ reward / utility function

Obj: Find $\mathbf{x}^* := \arg\max_{\mathbf{x} \in \mathcal{D}} g(\mathbf{x})$

$$\Pr(x \succ y) = link(g(x) - g(y))$$

✓ $$Pr(\mathbf{x} \succ \mathbf{y}) = \frac{1}{1+\exp\left(-\left(g(\mathbf{x})-g(\mathbf{y})\right)\right)}$$



sigmoid

Kumagui. Continuous Dueling Bandits, NeurIPS, 2018

Yue & Joachims. Interactively Optimizing Information Retrieval Systems as a Dueling Bandits Problem. ICML 2009

Different Objectives:

Obj-I: Cumulative Regret: Loss of the average quality of the arm-pair in T rounds

$$R_T = \sum_{t=1}^{T} g(\mathbf{x}_*) - \frac{g(\mathbf{x}_t)+g(\mathbf{y}_t)}{2}$$

$g(x) = \mathbf{x}^\top \mathbf{w}^*, \forall \mathbf{x} \in \mathcal{D}$, where $\mathbf{w}^* \in \mathbb{R}^d$ is fixed (unknown)

Different Objectives:

Obj-I: Cumulative Regret: Loss of the average quality of the arm-pair in T rounds

$$R_T = \sum_{t=1}^{T} g(\mathbf{x}_*) - \frac{g(\mathbf{x}_t)+g(\mathbf{y}_t)}{2}$$

Obj-II: Simple Regret (PAC objective): Given $\epsilon, \delta \in (0,1)$, in minimum T, find a decision point $\mathbf{x}_T \in \mathcal{D}$ such that:

$$Pr\big(g(\mathbf{x}^*) - g(\mathbf{x}_T) > \epsilon\big) < \delta$$

with minimum possible #samples (rounds)

Obj-III: Weak Regret: Cumulative loss of *only the best arms* in T rounds

$$R_T = \sum_{t=1}^{T} g(\mathbf{x}_*) - \max\big(g(\mathbf{x}_t), g(\mathbf{y}_t)\big)$$

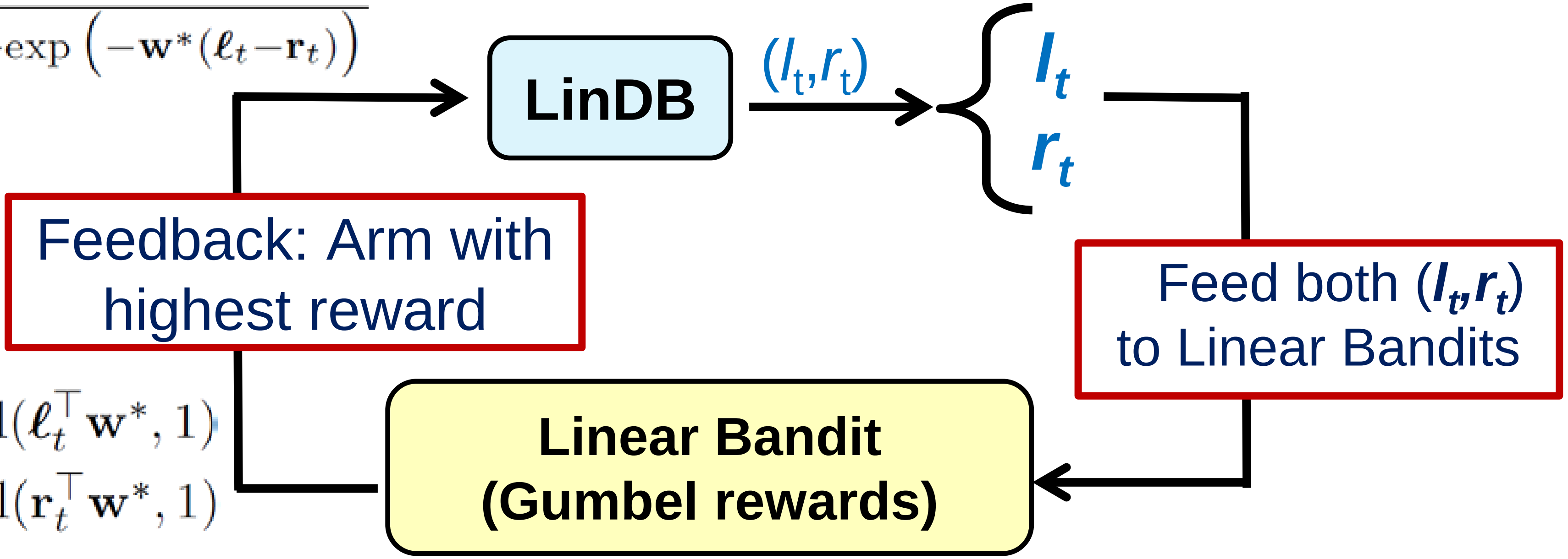
$g(x) = \mathbf{x}^\top \mathbf{w}^*, \forall \mathbf{x} \in \mathcal{D}$, where $\mathbf{w}^* \in \mathbb{R}^d$ is fixed (unknown)

Lower Bound?

(assume pairwise)

Lower Bound: Reducing ‘Gumbel linear-Bandits’ to LinDB

$$Pr(\boldsymbol{\ell}_t \succ \mathbf{r}_t) = Pr\big(X(\boldsymbol{\ell}_t) > X(\mathbf{r}_t)\big)$$
$$= \frac{1}{1+\exp\big(-\mathbf{w}^*(\boldsymbol{\ell}_t - \mathbf{r}_t)\big)}$$



$$X(\boldsymbol{\ell}_t) \sim \text{Gumbel}(\boldsymbol{\ell}_t^\top \mathbf{w}^*, 1)$$
$$X(\mathbf{r}_t) \sim \text{Gumbel}(\mathbf{r}_t^\top \mathbf{w}^*, 1)$$

$$2R_T^{(\text{LinDB})} = \sum_{t=1}^{T} \left((\mathbf{x}^*\mathbf{w}^* - \boldsymbol{\ell}_t^\top \mathbf{w}^*) + (\mathbf{x}^*\mathbf{w}^* - \mathbf{r}_t^\top \mathbf{w}^*) \right) = R_{2T}^{(\text{lin-Bandit})} = \Omega(\sqrt{dT})$$

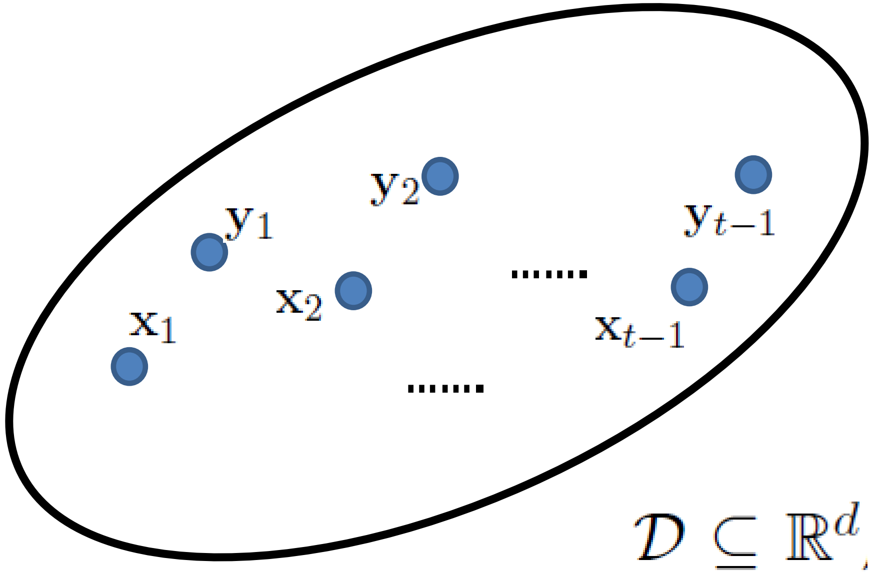
Algorithm?

Dueling Linear-Bandits (Cumulative-Regret-minimization)

Proposed algorithm (at any round t):

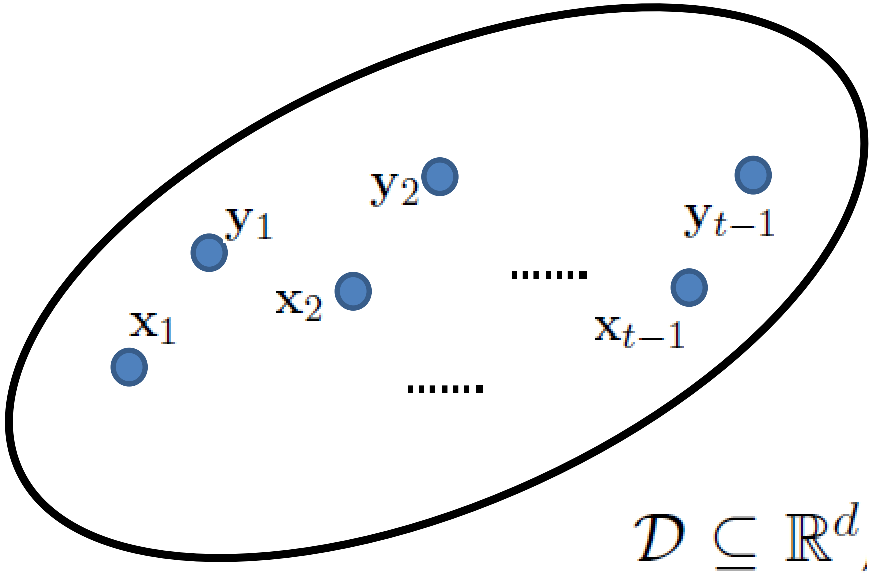
Step 1 (Parameter estimation):

Step 2 (“Most uncertain” arm-pair selection):



Dueling Linear-Bandits (Cumulative-Regret-minimization)

Proposed algorithm (at any round t):



Step 1 (Parameter estimation):

$$\hat{\mathbf{w}} \leftarrow \text{MLE}\left(\{(\mathbf{x}_\tau, \mathbf{y}_\tau, \mathbf{1}(\mathbf{x}_\tau \succ \mathbf{y}_\tau))\}_{\tau=1}^{t-1}, \mathbf{w}\right)$$

Step 2 ("Most uncertain" arm-pair selection):

$$(\mathbf{x}_t, \mathbf{y}_t) \leftarrow \arg\max_{\mathbf{x},\mathbf{y}\in\mathcal{C}_t} \|(\mathbf{x}-\mathbf{y})\|_{V_t^{-1}}$$

Potential good arms

Least observed arm-pair

$$\mathcal{C}_t := \left\{\mathbf{x} \in \mathcal{D} \mid \left((\mathbf{x}-\mathbf{y})^\top \hat{\mathbf{w}}_t + \alpha\|(\mathbf{x}-\mathbf{y})\|_{V_t^{-1}} > 0,\ \forall \mathbf{y} \in \mathcal{D}\right)\right\}$$

$$V_t = \sum_{\tau=1}^{t-1}(\mathbf{x}_t - \mathbf{y}_t)(\mathbf{x}_t - \mathbf{y}_t)^\top + \beta\mathbf{I}_{d\times d}$$

Near-Optimal regret guarantee

$$R_T^{(LinDB)} = O\left(\frac{d}{\kappa}\sqrt{T}\log T\right)$$

Saha et al, Optimal Algorithms for Stochastic Contextual Preference Bandits, NeurIPS, 2021

Summary

(Near) Optimal algorithm for PL Model with Large decision spaces

Outline

- Motivation: **Learning from Preference**
- Preference Models: **Representation of Preferences**
- Inference from Preferences: **PAC Objectives**
- Handling **Large** Decision Spaces
- **Advanced topics** in Preference Learning
- **PbRL as RLHF:** Preference based Reinforcement Learning
- Open Problems & Beyond

Advanced Topics
in Preference Learning

Non-Stationary (Time Varying) Preferences

Non-Stationary Preferences:

		2	3	4	5
		.17	0.50	0.54	0.57
2	0.83	0.5	0.93	0.98	0.91
3	0.50	0.07	0.5	0.60	0.53
4	0.44	0.02	0.40	0.5	0.59
5	0.43	0.09	0.47	0.41	0.5

~~P~~ (t = 1)

P_1

	1			4	5
1	0.5	0.17	0.50	0.54	0.57
2	0.83	0.5	0.93	0.98	0.91
3	0.50	0.07	0.5	0.60	0.53
4	0.44	0.02	0.40	0.5	0.59
5	0.43	0.09	0.47	0.41	0.5

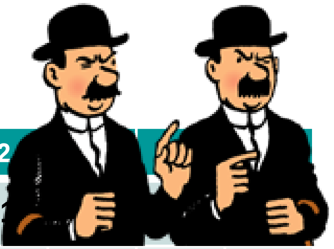
~~P~~ (t = 2)

P_2

	1	2	3		
1	0.5	0.17	0.50		
2	0.83	0.5	0.93	0.98	0.91
3	0.50	0.07	0.5	0.60	0.53
4	0.44	0.02	0.40	0.5	0.59
5	0.43	0.09	0.47	0.41	0.5

~~P~~ (t = 3)

P_3



	1	2			
1	0.5	0.			
2	0.83	0.5	0.93	0.98	0.91
3	0.50	0.07	0.5	0.60	0.53
4	0.44	0.02	0.40	0.5	0.59
5	0.43	0.09	0.47	0.41	0.5

~~P~~ (t = 4) ….

P_4

Makes no sense!

Regret := $max_{i^* \in [K]} \sum_{t=1}^{T} \frac{\{[P_t(i^*, x_t) - \frac{1}{2}] + [P_t(i^*, y_t) - \frac{1}{2}]\}}{2}$ s

Gajane et al. A Relative Exponential Weighing Algorithm for Adversarial Utility-based Dueling Bandits. ICML 2015

Borda Regret objective:

Preference matrix at time t

	1	2	3	4	5
1	0.5	0.53	0.54	0.56	0.6
2	0.47	0.5	0.53	0.58	0.61
3	0.46	0.47	0.5	0.54	0.57
4	0.44	0.42	0.46	0.5	0.51
5	0.4	0.39	0.43	0.49	0.5

P_t

Borda-score of Item-i: $b_t(i) := \frac{1}{K-1}\sum_{j \neq i} P_t(i,j)$

Another Regret definition:

Borda Regret $R_T := \sum_{t=1}^{T} b_t(i^*) - \frac{1}{2}(b_t(x_t) + b_t(y_t))$

where, cumulative Borda-winner: $i^* := \arg\max_{i \in [K]} \sum_{t=1}^{T} b_t(i)$

Jamieson et al. Sparse Dueling Bandit. AIStats 2015

Saha et al. Adversarial Dueling Bandit. AIStats 2015

Dynamic Regret for Time Varying Preferences

One possible solution: Dynamic Regret

P_1 (t = 1)

	1	2	3	4	5
1	0.5	0.79	0.98	0.16	0.06
2	0.21	0.5	0.43	0.08	0.27
3	0.02	0.57	0.5	0.14	0.07
4	0.84	0.92	0.86	0.5	0.51
5	0.94	0.73	0.93	0.49	0.5

P_2 (t = 2)

	1	2	3	4	5
1	0.5	0.13	0.94	0.16	0.06
2	0.87	0.5	0.43	0.08	0.71
3	0.06	0.57	0.5	0.14	0.07
4	0.84	0.92	0.86	0.5	0.51
5	0.94	0.29	0.93	0.49	0.5

P_3 (t = 3)

	1	2	3	4	5
1	0.5	0.83	0.94	0.16	0.06
2	0.17	0.5	0.43	0.18	0.71
3	0.06	0.57	0.5	0.14	0.07
4	0.84	0.82	0.86	0.5	0.51
5	0.94	0.29	0.93	0.49	0.5

P_4 (t = 4) ....

	1	2	3	4	5
1	0.5	0.83	0.94	0.16	0.06
2	0.17	0.5	0.43	0.18	0.71
3	0.06	0.57	0.5	0.14	0.07
4	0.84	0.82	0.86	0.5	0.51
5	0.94	0.29	0.93	0.49	0.5

1. **Switching-Variation** $S := \sum_{t=2}^{T} \mathbf{1}[\mathbf{P}_t \neq \mathbf{P}_{t-1}] + 1$
2. **Continuous-Variation:** $V_T := \sum_{t=2}^{T} \max_{(a,b)\in[K]\times[K]} |P_t(a,b) - P_{t-1}(a,b)| + 1$

for ANY sequence of arms $(i_1^*, i_2^*, \ldots i_T^*)$

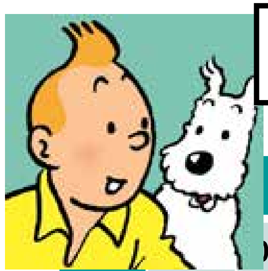
$$O(\sqrt{SKT}\log KT) \longleftarrow \sum_{t=1}^{T} \frac{\left\{\left[P(i_t^*, x_t) - \frac{1}{2}\right] + \left[P(i_t^*, y_t) - \frac{1}{2}\right]\right\}}{2} \longrightarrow O(V_T^{\frac{1}{3}} K^{\frac{1}{3}} T^{\frac{2}{3}} \log KT)$$

(Dynamic) Borda regret: $\sum_{t=1}^{T} \frac{\{[b_t(i_t^*) - b_t(x_t)] + [[b_t(i_t^*) - b_t(y_t)]]\}}{2}$

Gupta et al. Non-Stationary Dueling Bandits. ICML 2022

Personalized Prediction
with User Preferences

Contextual Duels!



c_1

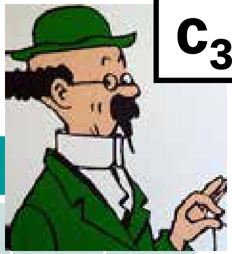
	1	2	3	4	5
1	[illegible]	.17	0.50	0.54	0.57
2	0.83	0.5	0.93	0.98	0.91
3	0.50	0.07	0.5	0.60	0.53
4	0.44	0.02	0.40	0.5	0.59
5	0.43	0.09	0.47	0.41	0.5

$f_1 = f(c_1) = \mathbf{P_1}$ $(t = 1)$

c_2

	1	2	3	4	5
1	[illegible]	0.53	0.54	0.56	0.6
2	0.47	0.5	0.93	0.98	0.91
3	0.46	0.07	0.5	0.54	0.57
4	0.44	0.02	0.46	0.5	0.51
5	0.4	0.09	0.43	0.49	0.5

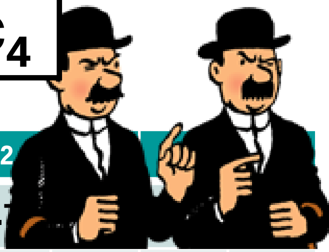
$f_2 = f(c_2) = \mathbf{P_2}$ $(t = 2)$



c_3

	1	2	3	4	5
1	0.5	0.83	0.94	[illegible]	[illegible]
2	0.17	0.5	0.43	0.18	0.71
3	0.06	0.57	0.5	0.14	0.07
4	0.84	0.02	0.86	0.5	0.51
5	0.94	0.29	0.93	0.49	0.5

$f_3 = f(c_3) = \mathbf{P_3}$ $(t = 3)$



c_4

	1	2	3	4	5
1	0.5	0.7	[illegible]	[illegible]	[illegible]
2	0.21	0.5	0.43	0.08	0.27
3	0.02	0.57	0.5	0.14	0.07
4	0.84	0.92	0.86	0.5	0.51
5	0.84	0.73	0.93	0.49	0.5

$f_4 = f(c_4) = \mathbf{P_4}$ $(t = 4)$ ..

π: Context $\mapsto$ Arm

f: Context $\mapsto$ Preference matix

$f \in F$ (known function class)

"Contextual" Regret:

Policy Regret

$$\max_{\{\pi^* \in \Pi\}} \sum_{t=1}^{T} E_{(x_t, y_t) \sim \boldsymbol{p_t}} \left[\frac{\{[f_t(\pi^*(c_t), x_t)] + [f_t(\pi^*(c_t), y_t)]\} - 1}{2} \right]$$

Strongest opponent

Best Response Regret

$$\le \sum_{t=1}^{T} \max_{\boldsymbol{i_t^*} \in [\boldsymbol{K}]} E_{(x_t, y_t) \sim \boldsymbol{p_t}} \left[\frac{\{[f_t(i_t^*, x_t)] + [f_t(i_t^*, y_t)]\} - 1}{2} \right]$$

where $\boldsymbol{p_t} \in \Delta_{K \times K}$

Dudik et al. (2015)

- Suboptimal
- (or) Runtime in-efficient
- But not BOTH

Contextual Dueling: Main Algorithm and Regret

Algorithm MinMaxDB

input: Arm set: $[K]$, parameters $\gamma > 0$.
An instance of SqrReg for function class $\mathcal{F}$
for $t = 1, 2, \ldots, T$ **do**
Receive context c_t
Estimate f: $\hat{f}_t \leftarrow \mathrm{SqrReg}(\{c_\tau, (x_\tau, y_\tau), o_\tau\}_{\tau=1}^{t-1})$
Find $p_t \in \Delta_K$ such that

$$\forall\, i \in [K]: \sum_{b \in [K]} \hat{f}_t(i,b) p_t(b) + \frac{3}{32\, \gamma p_t(i)} \le \frac{K}{\gamma}$$

Sample $(x_t, y_t) \overset{iid}{\sim} \mathbf{p}_t$, play the duel (x_t, y_t) and receive feedback o_t.
Update SqrReg with example $\{c_t, (x_t, y_t), o_t\}$
end for

Regression Oracle

$Reg_{sq}(T) = \sum_{\tau=1}^{T} \left(f(c_\tau)[x_\tau, y_\tau] - \hat{f}_t(c_\tau)[x_\tau, y_\tau] \right)^2$

← from MinMax analysis

Regression Oracle's Regret (≈O(log|F|))

- Optimal and Efficient: $O\left(\sqrt{KT\, Reg_{sq}(T)}\right)$

Saha, Krishnamurthy. Efficient and Optimal Algorithms for Contextual Dueling Bandits under Realizability. ALT'22

Outline

- Motivation: **Learning from Preference**
- Preference Models: **Representation of Preferences**
- Inference from Preferences: **PAC Objectives**
- Handling **Large** Decision Spaces
- **Advanced topics** in Preference Learning
- **PbRL as RLHF:** Preference based Reinforcement Learning
- Open Problems & Beyond

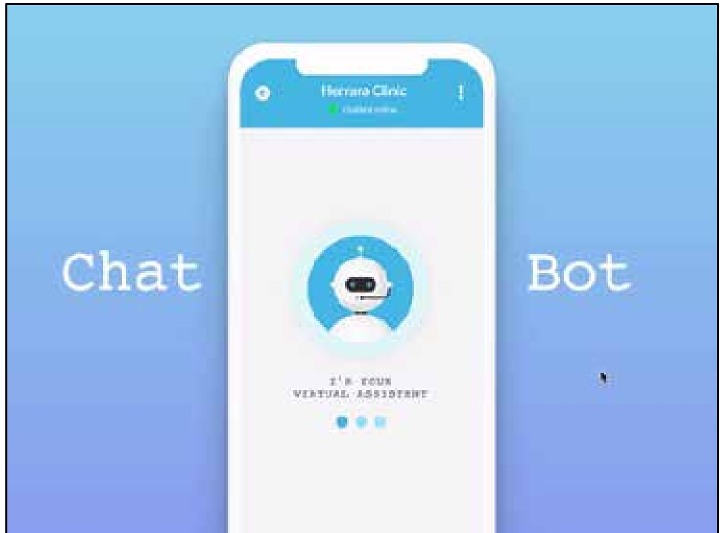
AI Alignment/RLHF
(with Preferences!)

Trajectory Preferences: Long term (complex) predictions

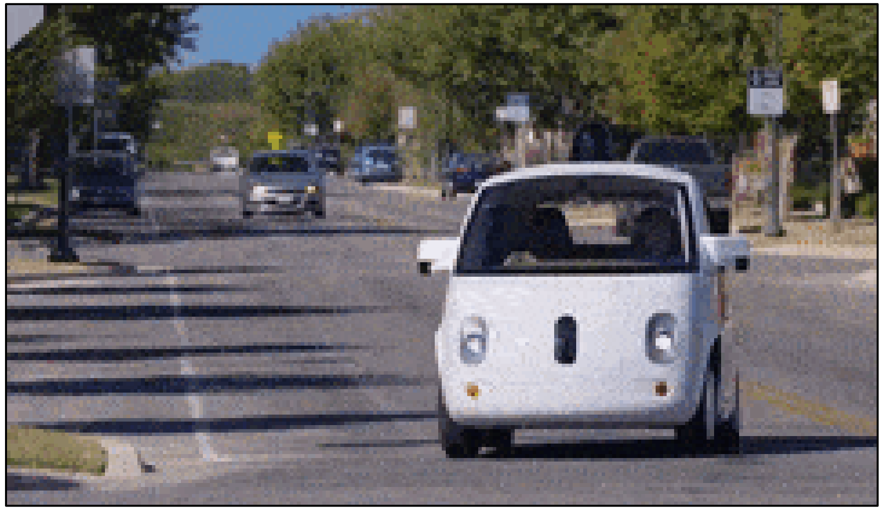
Multiplayer games

Personalized healthcare

ChatBot Conversations



Self-driving cars



Aligning language models with Preference Feedback

Training language models to follow instructions with human feedback

Long Ouyang* Jeff Wu* Xu Jiang* Diogo Almeida* Carroll L. Wainwright*

Pamela Mishkin* Chong Zhang Sandhini Agarwal Katarina Slama Alex Ray

John Schulman Jacob Hilton Fraser Kelton Luke Miller Maddie Simens

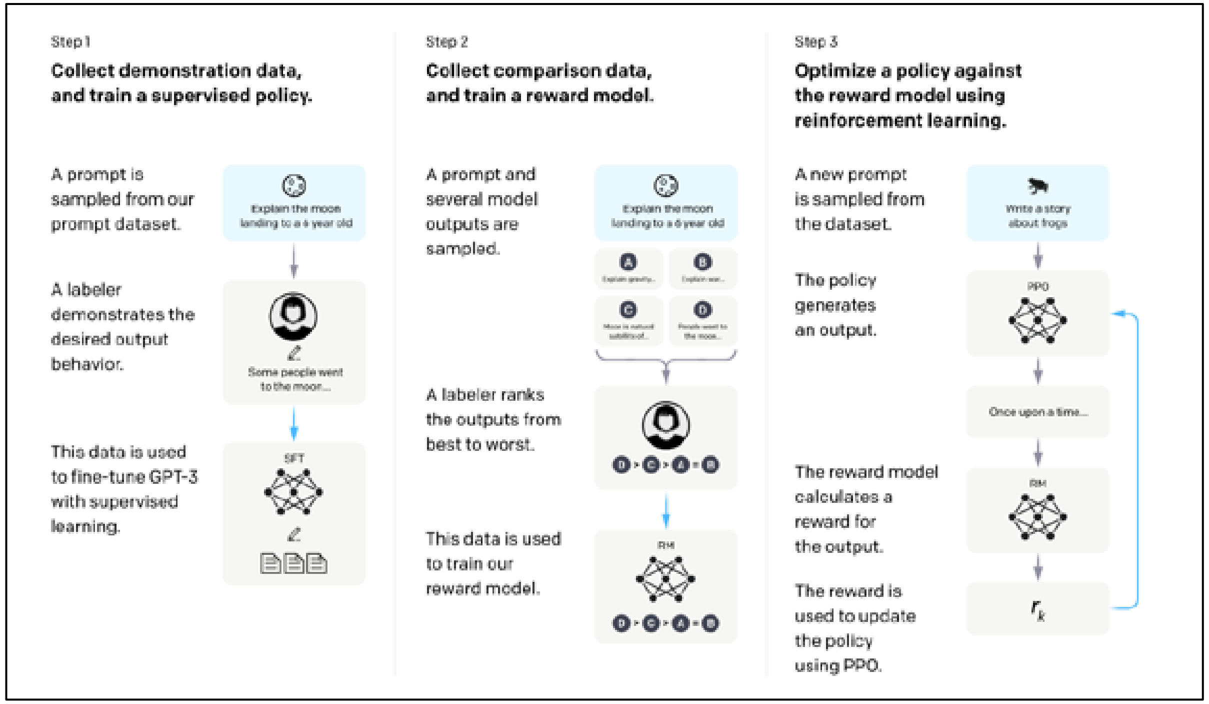
Amanda Askell† Peter Welinder Paul Christiano*†

Jan Leike* Ryan Lowe*

OpenAI

Abstract

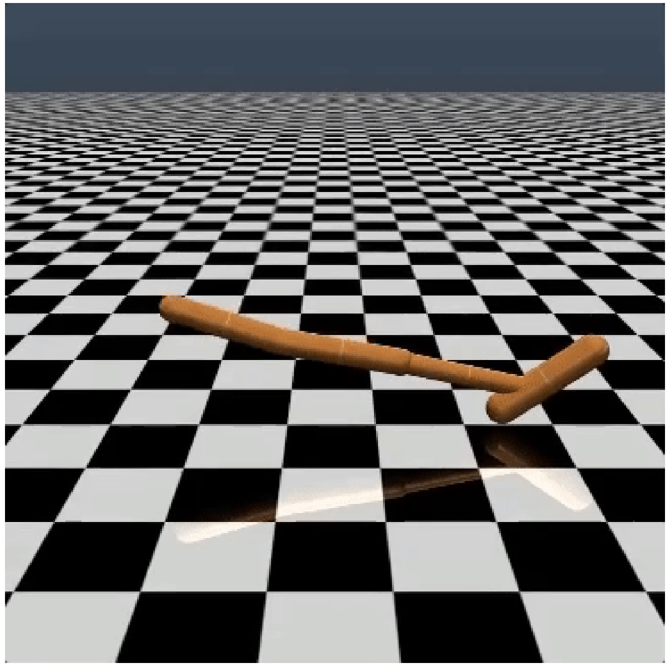
Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models *InstructGPT*. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT still makes simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.



https://openai.com/research/instruction-following

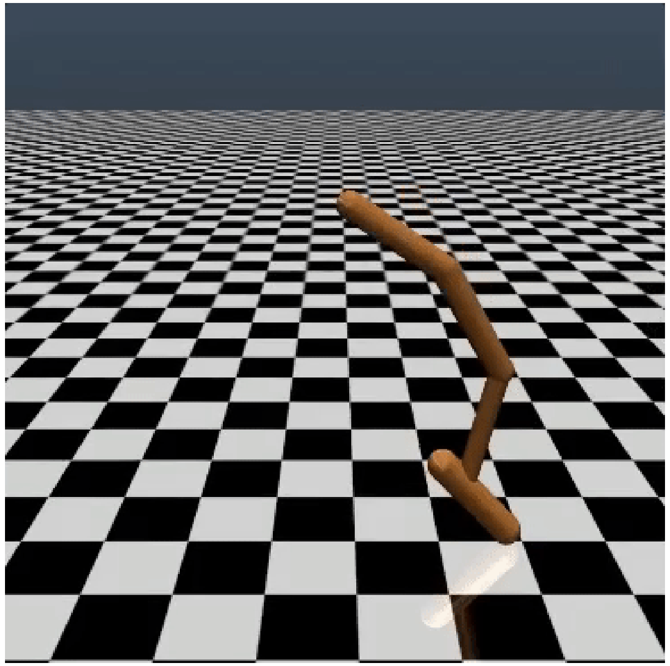
https://openai.com/research/learning-from-human-preferences

Remedy: Preferences for Reward Shaping!



A

≪

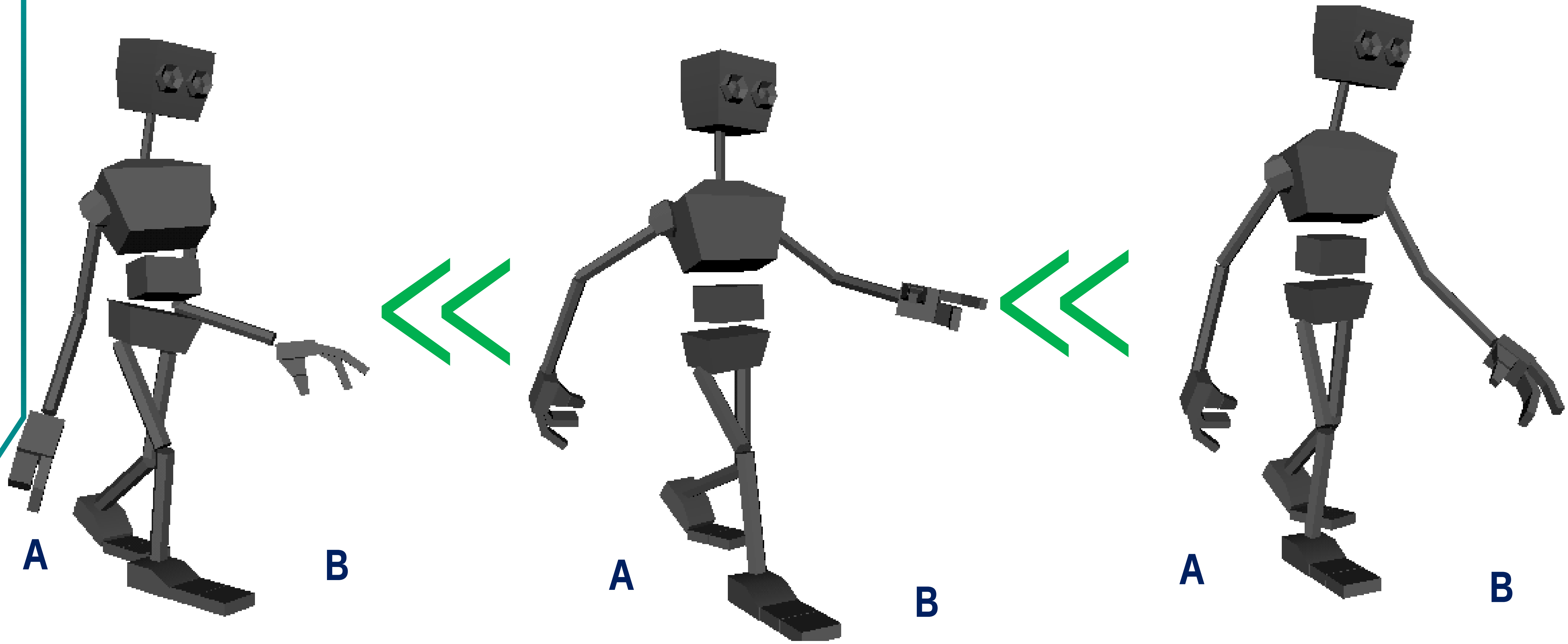


B

Picture courtesy: https://openai.com/research/learning-from-human-preferences

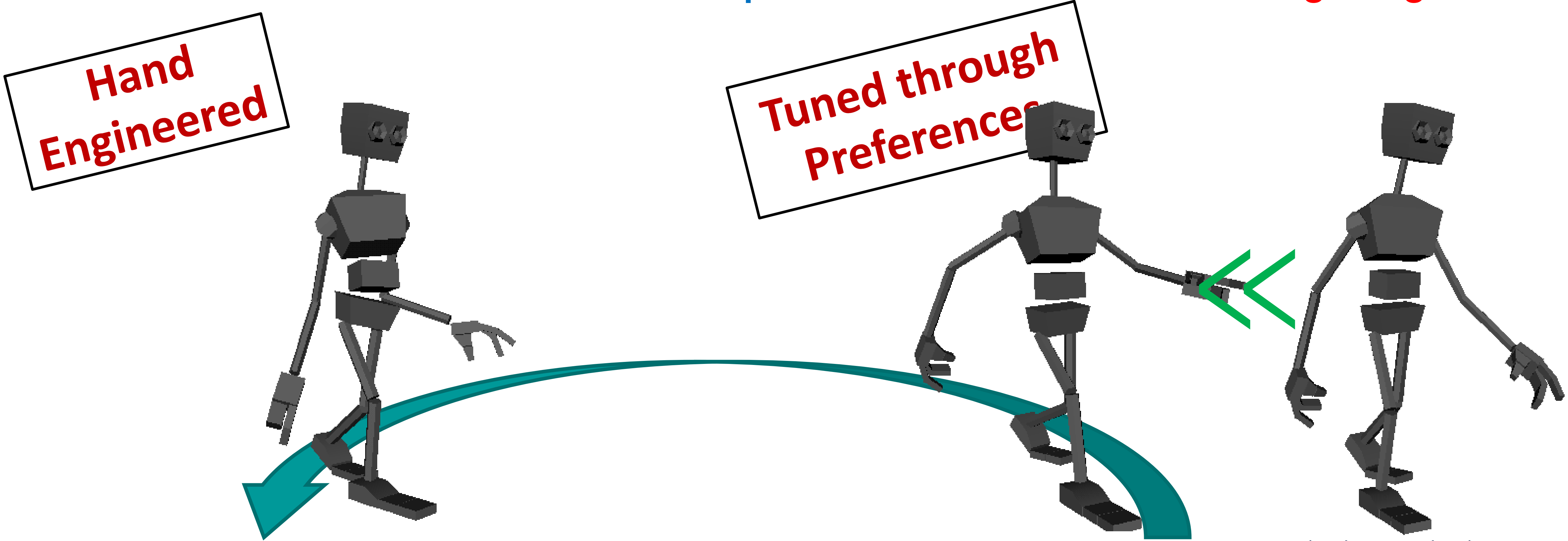
Christiano et al., Deep reinforcement learning from human preferences. NeurIPS, 2017.

Remedy: Preferences for Reward Shaping!



Picture courtesy: https://cs.brown.edu/people/rpatel59/research-code.html

PbRL & the "Difficult" problem of Reward Designing



r(s,a) = how to assign?
$\forall\, a \in A, s \in S$

Preference(τ_1 -vs- τ_2) $\propto r(\tau_1) - r(\tau_2)$

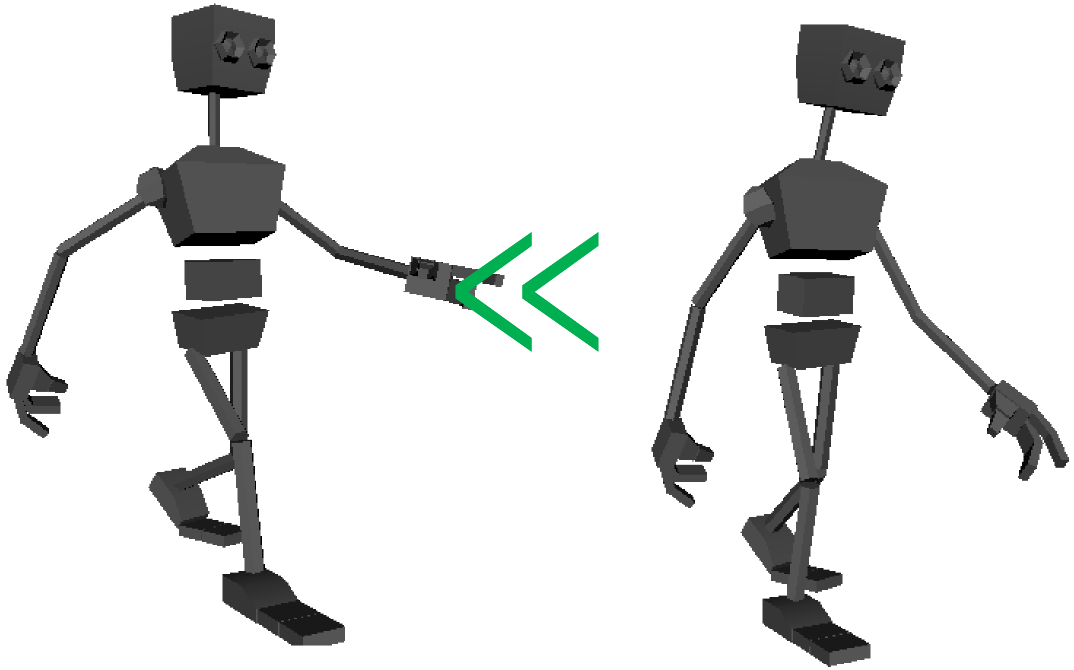
for all trajectory pairs τ_1, τ_2

```
def reward_fn(a, ob):
    backroll = -ob[7]
    height = ob[0]
    vel_act = a[0] * ob[8] + a[1] * ob[9] + a[2] * ob[10]
    backslide = -ob[5]
    return backroll * (1.0 + .3 * height + .1 * vel_act + .05 * backslide)
```

- ü No painful reward encoding
- ü Sample efficient
- ü Safe and Fair
- ü Captures human "enjoyment"

https://openai.com/blog/deep-reinforcement-learning-from-human-preferences/

Reinforcement Learning with State-based Preferences



Preference(τ_1 -vs- τ_2) $\propto score(\tau_1) - score(\tau_2)$

for all trajectory pairs τ_1, τ_2

- ü No painful reward engineering!
- ü Sample efficient
- ü Safe and Fair
- ü Captures human "enjoyment"

Trajectory $\tau := (s_1, a_1, \cdots, s_H, a_H)$

Linear Score func $s(\tau) := \langle \phi(\tau), \mathbf{w}^* \rangle$

where Trajectory Feature

$$\phi(\tau) = \sum_{h=1}^{H} \phi(s_h, a_h), \text{ where } \phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$$

Modeling trajectory preference:

$$\mathbb{P}(\tau_1 \succ \tau_2) = \sigma(\langle \phi(\tau_1) - \phi(\tau_2), \mathbf{w}^* \rangle) = \frac{\exp(\phi(\tau_1)^\top \mathbf{w}^*)}{\exp(\phi(\tau_1)^\top \mathbf{w}^*) + \exp(\phi(\tau_2)^\top \mathbf{w}^*)}$$

Saha*, Pacchiano*, Lee. Dueling RL: RL with Trajectory Preferences. AIStats, 2023.

Our Result: Dueling RL

Preference based RL framework (finite horizon): $(\mathbb{P}, \mathcal{S}, \mathcal{A}, H, \rho)$

Dynamics $s_{t+1} \sim p(\cdot \mid s_t, a_t)$

Algorithm 1 LPbRL: Regret minimization (Known Model)

input: Regularization parameter λ, Learning rate $\eta_t > 0$, exploration length $t_0 > 0$
Define $\alpha_{d,T}(\delta) = 20BW\sqrt{d\log(T(1+2T)/\delta)}$ and $\gamma_t(\delta) = 2\kappa\beta_t(\delta) + \alpha_{d,T}(\delta)$.
Initialize $\overline{\mathbf{V}}_t = \kappa\lambda\mathbb{I}_d$
for $t = 1, 2, \ldots T$ **do**
  Compute $\mathbf{w}_t^L$ (using MLE on history)
  Set $\Pi_t = \{\pi^1 | (\phi(\pi^1) - \phi(\pi))^\top \mathbf{w}_t^L +$

$$\gamma_t(\delta)\|\phi(\pi^1) - \phi(\pi)\|_{\overline{\mathbf{V}}_t^{-1}} \geq 0 \ \forall \pi\}$$

  Compute

$$(\pi_t^1, \pi_t^2) = \arg\max_{\pi^1, \pi^2 \in \Pi_t} \|\phi(\pi^1) - \phi(\pi^2)\|_{\overline{\mathbf{V}}_t^{-1}}.$$

  Sample $\tau_t^1 \sim \pi_t^1$ and $\tau_t^2 \sim \pi_t^2$.
  Play the duel (τ_t^1, τ_t^2) and receive $o_t = \mathbb{1}(\tau_t^1 \text{ beats } \tau_t^2)$
  Update

$$\overline{\mathbf{V}}_{t+1} = \overline{\mathbf{V}}_t + (\phi(\pi_t^1) - \phi(\pi_t^2))(\phi(\pi_t^1) - \phi(\pi_t^2))^\top$$

end for

Regret :

$$= \sum_{t=1}^{T} \frac{2s(\pi^*) - (s(\pi_t^1) + s(\pi_t^2))}{2}$$

$\pi : \mathcal{S} \mapsto \mathcal{A}$ [[Policy: States $\mapsto$ Actions]]

Our Results

$$\tilde{O}\left(SHd\log(T/\delta)\sqrt{T}\right)$$

- Known Model:

$$\tilde{O}((\sqrt{d} + H^2 + |S|)\sqrt{dT} + \sqrt{|S||A|TH})$$

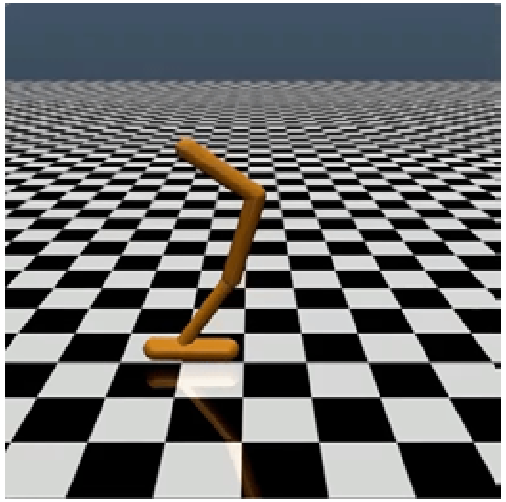
- Unknown Model:

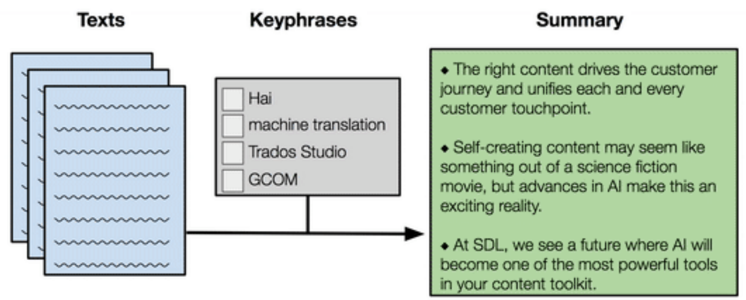
Saha*, Pacchiano*, Lee. Dueling RL: RL with Trajectory Preferences. AIStats, 2023.

PbRL literature: Very few works!

- Busa-Fekete. ML 2014
- Christiano et al. NeurIPS 2017
- Wirth et al. JMLR 2017

Predominantly applied (Deepmind, OpenAI, …)





- Sui et al. UAI 2017
- Xu et al. NeurIPS 2020
- Saha et al. AIStats 2023

Unsatisfactory theoretical developments (but restrictive assumptions / guarantees)

Outline

- Motivation: **Learning from Preference**
- Preference Models: **Representation of Preferences**
- Inference from Preferences: **PAC Objectives**
- Handling **Large** Decision Spaces
- **Advanced topics** in Preference Learning
- **PbRL as RLHF:** Preference based Reinforcement Learning
- Open Problems & Beyond

Emerging Directions

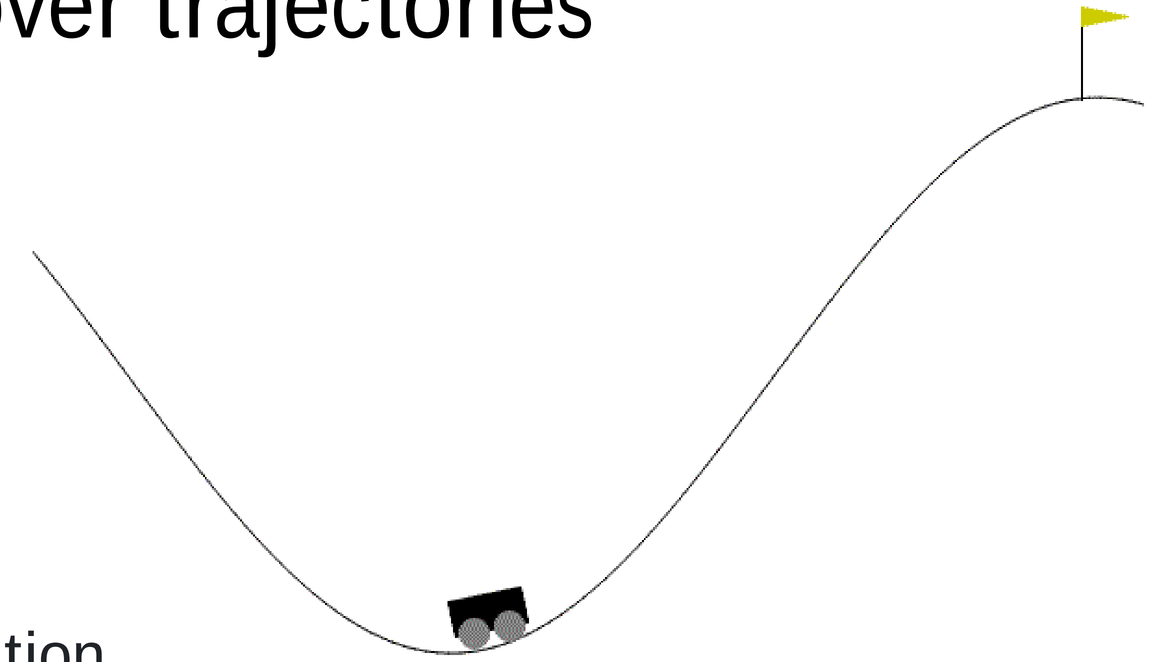
- Preferences for Alignment of LLMs?
- Automating Training with Preferences?
- PbRL is subcase of RLHF, what other implicit human feedback?
- Is sigmoid enough for modeling preferences?
- How much we lose by using preferences instead of rewards? (not quantifed study)
- User adaptation modeling for preferences?

Part – III (Demos)

Demo 1: ELO ratings of chess players

Demo 2: RL with preferences over trajectories

Environment: Mountain car
State: (position, velocity) $\in \mathbb{R}^2$
Action: force $\in$ {Left, Right, None}
Transitions: Gravity
Trajectory for preference elicitation:
$(s_1, a_1, \ldots s_n, a_n)$
Trajectory features: min position, max position, average speed



Credits:

- APReL: A Library for Active Preference-based Reward Learning Algorithms, Erdem Bıyık, Aditi Talati, Dorsa Sadigh
- https://github.com/Stanford-ILIAD/APReL

Acknowledgments

भारतीय विज्ञान संस्थान

INDIAN INSTITUTE OF SCIENCE

A CSR Initiative by

kotak

Kotak Mahindra Bank

Kotak IISc AI–ML Centre

Part – IV (Panel)