Contextual Bandits in Recommendation

James McInerney

IVADO Recommender Systems Workshop
August 22, 2019

NETFLIX RESEARCH

jmcinerney@netflix.com | @mcinerneyj

About Me

Senior Research Scientist at Netflix, California

Previously:

Senior Research Scientist at Spotify, New York

Postdoc at Columbia & Princeton University

Background in probabilistic machine learning, causality, recommender systems, & spatiotemporal modeling.

Jargon in Bandits

Bandit / Reinforcement Learning Term	Machine Learning Term
action	recommendation
arm	item
reward / payoff	relevance, target, output
context	features
policy	distribution over actions
exploit	perform the optimal action
explore	perform an action to learn more
inverse propensity scoring	importance sample reweighting
dueling bandits	comparison between bandits

Part I

- Motivation
- Cold start
- Predictive Uncertainty
- Explore-Exploit

Part II

- Bandits
- Contextual Bandits
- On-Policy Learning & Evaluation

Part III

- Off-Policy Learning & Evaluation
- Feedback Loops
- Slate Recommendation

Why Recommend?

Why Recommend?

Why Recommend?

crucial to enjoying items from a large and growing catalogue

Why Recommend?

crucial to enjoying items from a large and growing catalogue

reduction in churn —> saves Netflix $1 billion / year [Uribe & Hunt, 2015]

What is a Recommendation?

Working definition for this talk:

“A decision made by an interface that exposes user attention to an item.”

What is a Recommendation?

Working definition for this talk:

“A decision made by an interface that exposes user attention to an item.”

decision ≡ action

What is a Recommendation?

Working definition for this talk:

"A decision made by an interface that exposes user attention to an item."

decision ≡ action

—> corollary: recommendation influences what is consumed and enjoyed

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)}$$

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

R : relevance

Collaborative Filtering

<u>Goal of user relevance model</u>: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

R : relevance

relevance ≡ reward
e.g. click / no click (binary)
or length of stream (non-negative real)

$$\boxed{\hat{r}(A, X)} \approx R$$

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

R : relevance

A : recommendation

relevance ≡ reward
e.g. click / no click (binary)
or length of stream (non-negative real)

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

R : relevance ← relevance ≡ reward e.g. click / no click (binary) or length of stream (non-negative real)

A : recommendation ← 1 of K items

Collaborative Filtering

<u>Goal of user relevance model</u>: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

R : relevance ← relevance ≡ reward
e.g. click / no click (binary)
or length of stream (non-negative real)

A : recommendation ← 1 of K items

X : context

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

R : relevance — relevance ≡ reward
e.g. click / no click (binary)
or length of stream (non-negative real)

A : recommendation — 1 of K items

X : context — anything observed about the user, items, or background information

Collaborative Filtering

<u>Goal of user relevance model</u>: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

R : relevance

relevance ≡ reward
e.g. click / no click (binary)
or length of stream (non-negative real)

A : recommendation

1 of K items

X : context

anything observed about the user, items, or background information

$\hat{r}$: user relevance model

Collaborative Filtering

Goal of user relevance model: to predict what a user will like based on past interactions between all users and items.

e.g., matrix factorization, factorization machines, deep learning, word2vec.

user relevance model

$$\boxed{\hat{r}(A, X)} \approx R$$

R : relevance ← relevance ≡ reward e.g. click / no click (binary) or length of stream (non-negative real)

A : recommendation ← 1 of K items

X : context ← anything observed about the user, items, or background information

$\hat{r}$: user relevance model

how to use predicted relevance to decide which item to recommend?

Choose Actions from a Policy

Policy π

Choose Actions from a Policy

Policy $\pi : \{1, \ldots, K\} \to \mathbb{R}_{\geq 0}$ such that $\sum_{k=1}^{K} \pi(a_k) = 1$

Choose Actions from a Policy

Policy $\pi : \{1, \ldots, K\} \to \mathbb{R}_{\geq 0}$ such that $\sum_{k=1}^{K} \pi(a_k) = 1$

$A \sim \pi$  K-sided die

Choose Actions from a Policy

Policy $\pi : \{1, \ldots, K\} \to \mathbb{R}_{\geq 0}$ such that $\sum_{k=1}^{K} \pi(a_k) = 1$

$A \sim \pi$  K-sided die

User Model $\neq$ Policy

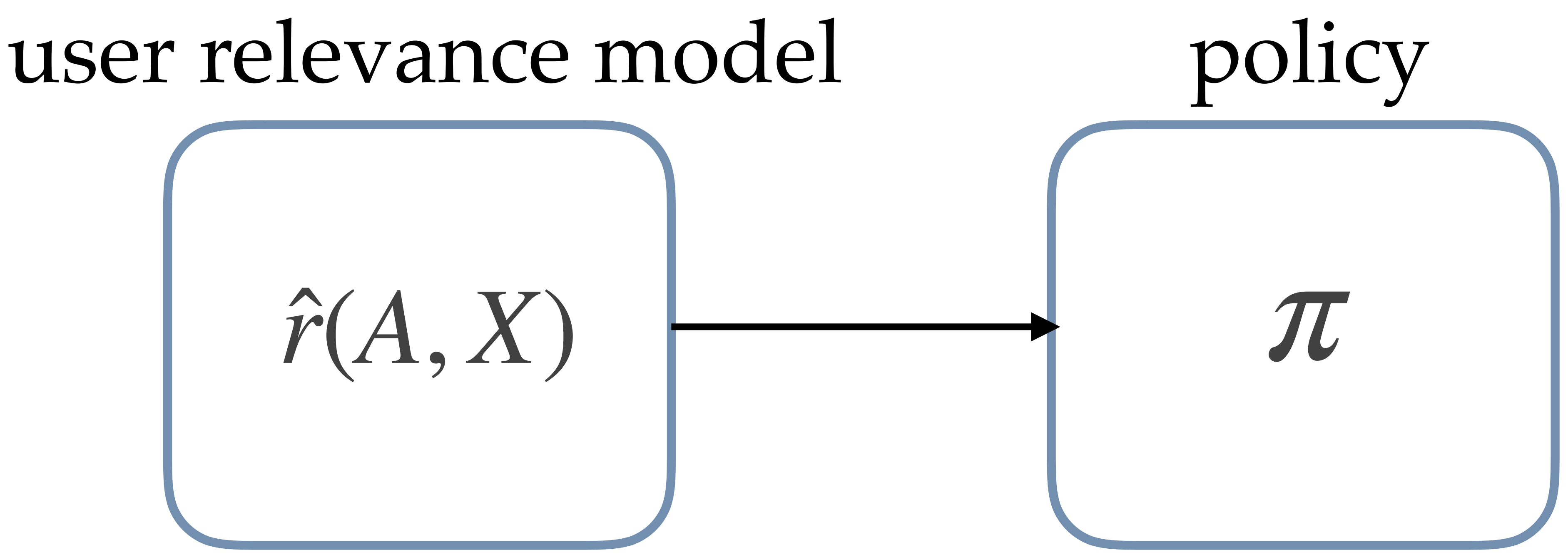
Choose Actions from a Policy

Policy $\pi : \{1, \ldots, K\} \to \mathbb{R}_{\geq 0}$ such that $\sum_{k=1}^{K} \pi(a_k) = 1$

$A \sim \pi$  K-sided die

User Model $\neq$ Policy

Policy Derived from User Relevance Model



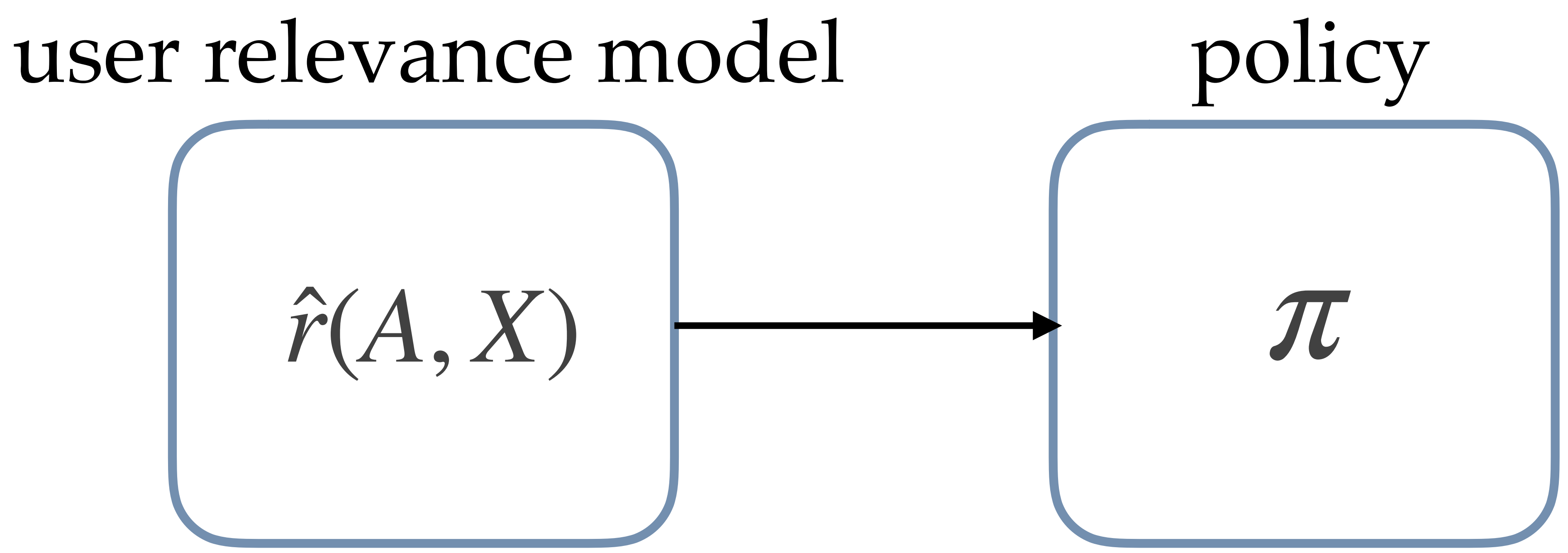
Choose Actions from a Policy

Policy $\pi : \{1, \ldots, K\} \to \mathbb{R}_{\geq 0}$ such that $\sum_{k=1}^{K} \pi(a_k) = 1$

$A \sim \pi$  K-sided die

User Model $\neq$ Policy

Policy Derived from User Relevance Model



Policy Model

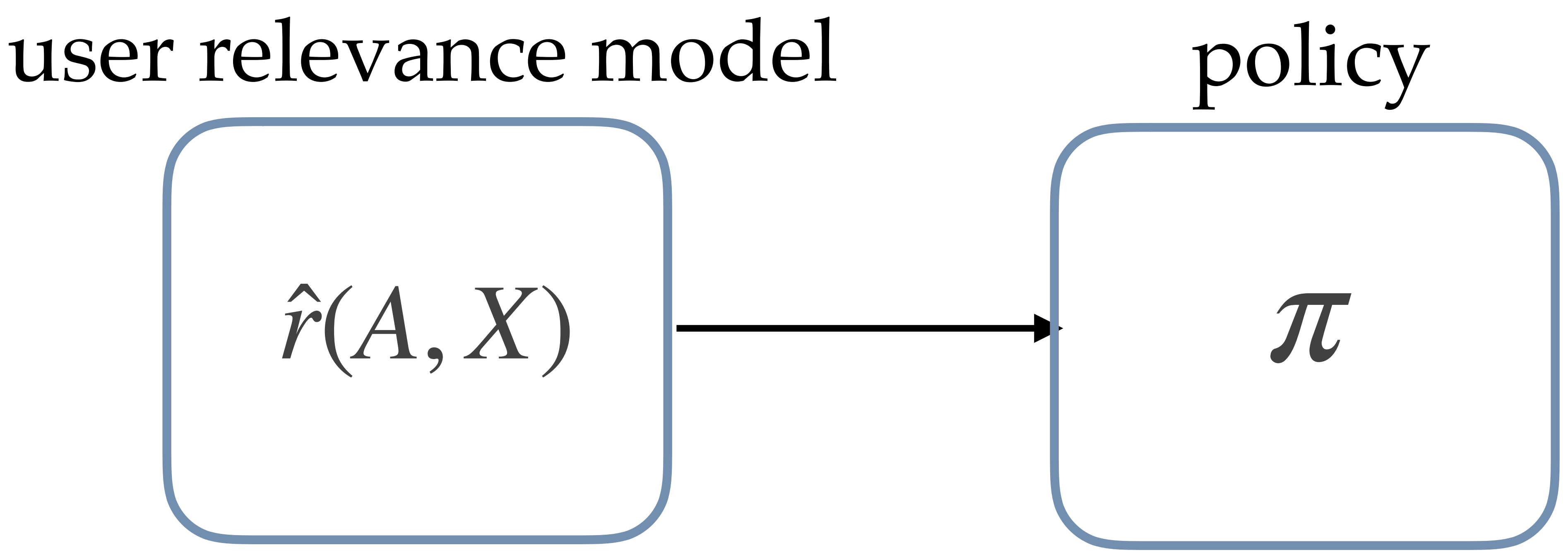
$\pi \to p_\phi(A \mid X)$ e.g. multiclass classification

Originated in learning to rank, now in recsys.

A Very Simple Policy

A Very Simple Policy

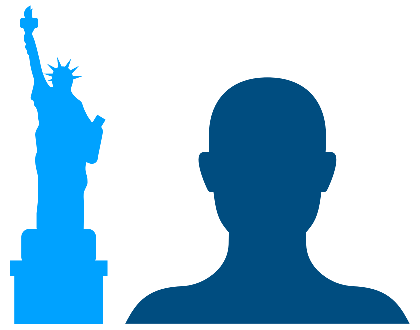
Policy Derived from User Relevance Model



A Very Simple Policy

Policy Derived from User Relevance Model

user relevance model		policy
$\hat{r}(A, X)$	$\longrightarrow$	π



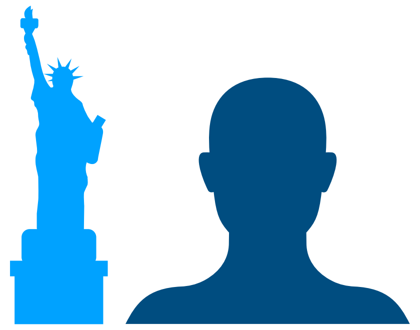
0.3

0.05

A Very Simple Policy

Policy Derived from User Relevance Model

user relevance model	policy
$\hat{r}(A, X)$ $\longrightarrow$	π



0.3

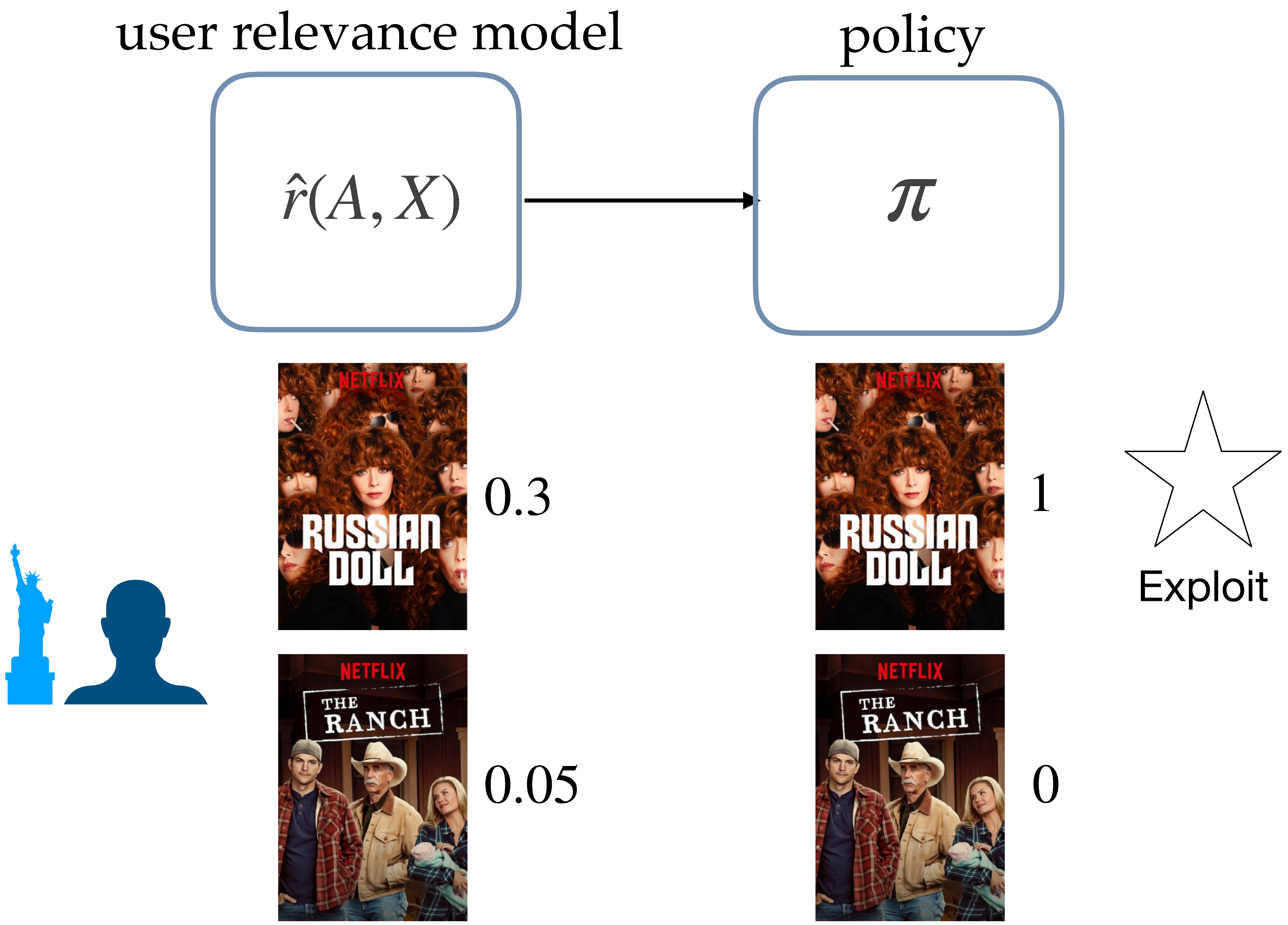
1

0.05

0

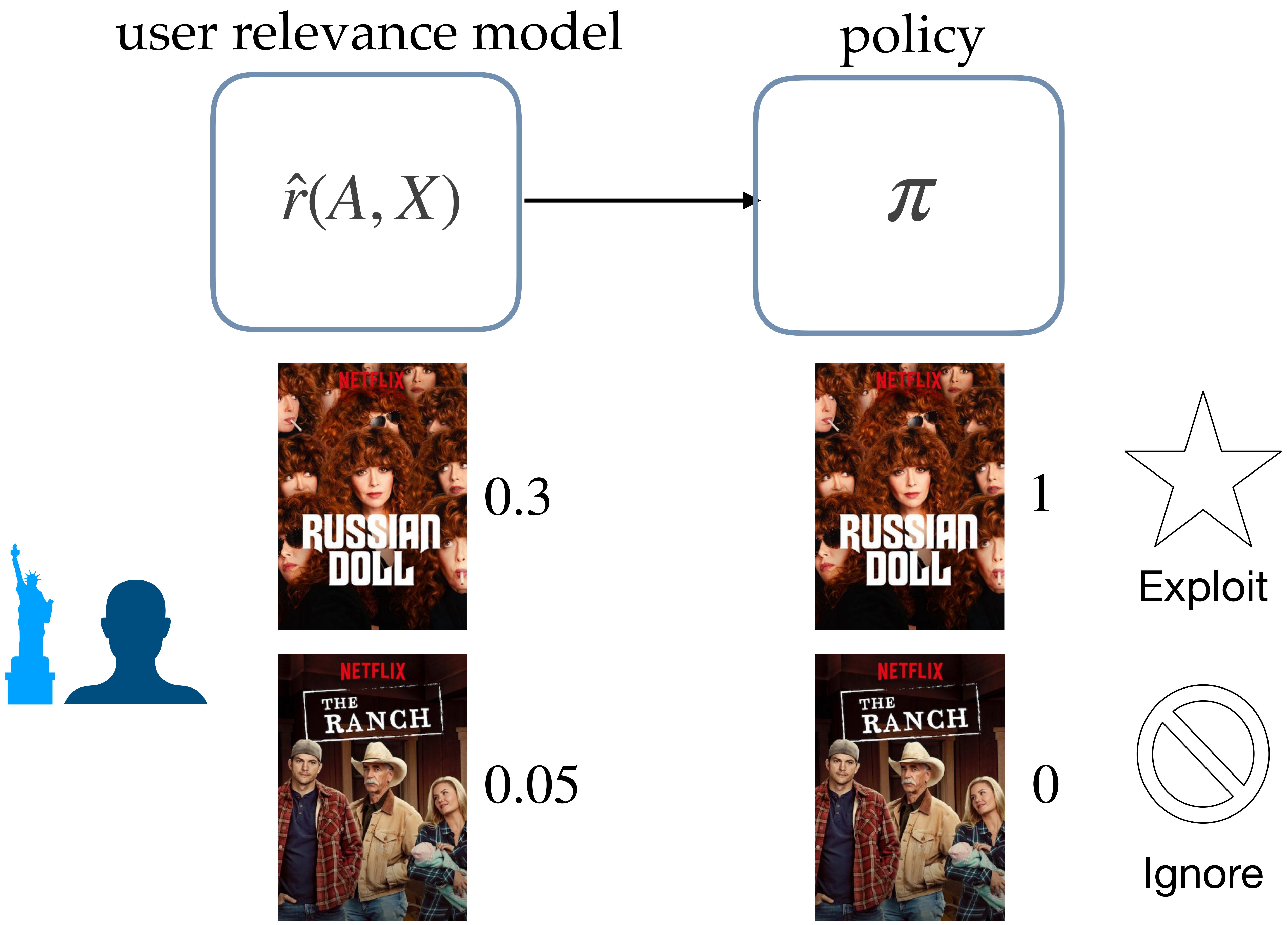
A Very Simple Policy

Policy Derived from User Relevance Model



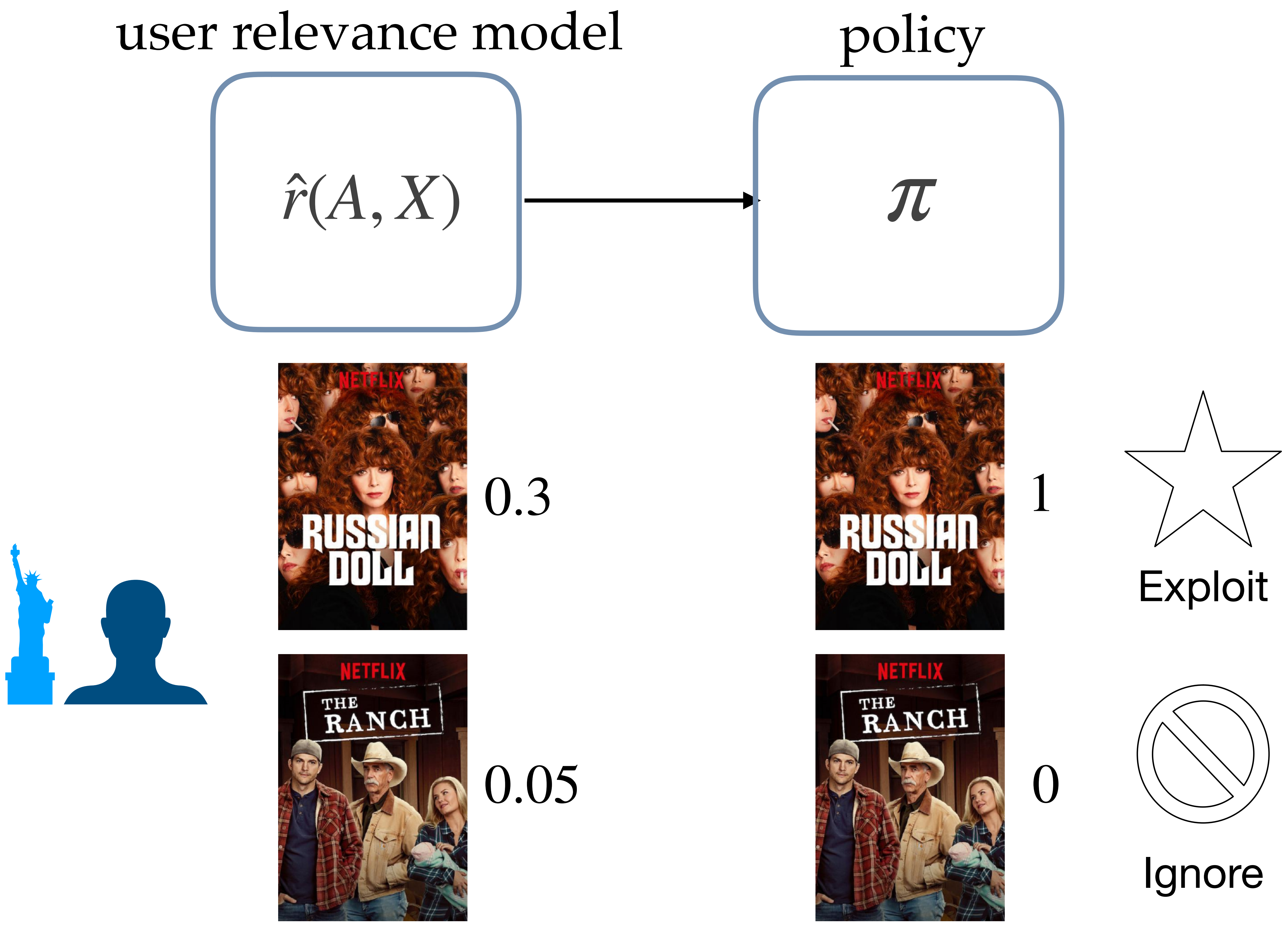
A Very Simple Policy

Policy Derived from User Relevance Model



A Very Simple Policy

Policy Derived from User Relevance Model



In general: policies can be deterministic or stochastic.

Collaborative Filtering

out of matrix prediction

X : context

A : recommendation

		
	0.1	0.3
	0.3	0.05

Collaborative Filtering

out of matrix prediction

X : context

A : recommendation

		
	0.1	0.3
	0.3	0.05

Service launches in the UK….

Collaborative Filtering

out of matrix prediction

X : context

A : recommendation

			
	0.1	0.3	?
	0.3	0.05	?

Collaborative Filtering

out of matrix prediction

X : context

A : recommendation

<table>
<tr><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>0.1</td><td>0.3</td><td>?</td></tr>
<tr><td></td><td>0.3</td><td>0.05</td><td>?</td></tr>
</table>

New original title released….

Collaborative Filtering

out of matrix prediction

X : context

A : recommendation

			
	0.1	0.3	?
	0.3	0.05	?
	?	?	?

Collaborative Filtering

out of matrix prediction

X : context

A : recommendation

			
	0.1	0.3	?
	0.3	0.05	?
	?	?	?

cold start

non-stationarity

Perpetual Coldness

Large context, large action space, growing item set, growing user base, changing culture.

Predictive Uncertainty

A useful principle for understanding cold start.

Predictive Uncertainty

A useful principle for understanding cold start.

$$\text{dataset } \mathcal{D} := \{(x_n, a_n, r_n)_{n=1}^{N}\}$$

Predictive Uncertainty

A useful principle for understanding cold start.

$$\text{dataset } \mathscr{D} := \{(x_n, a_n, r_n)_{n=1}^{N}\}$$

$$\text{predictive distribution } p(R \mid \mathscr{D}, A, X)$$

Predictive Uncertainty

A useful principle for understanding cold start.

$$\text{dataset } \mathscr{D} := \{(x_n, a_n, r_n)_{n=1}^N\}$$

$$\text{predictive distribution } p(R \mid \mathscr{D}, A, X) = \int p(R \mid A, X, \theta) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

Predictive Uncertainty

A useful principle for understanding cold start.

$$\text{dataset } \mathscr{D} := \{(x_n, a_n, r_n)_{n=1}^{N}\}$$

$$\text{predictive distribution } p(R \mid \mathscr{D}, A, X) = \int p(R \mid A, X, \theta) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

$$\text{where } p(\theta \mid \mathscr{D}) \propto p(\mathscr{D} \mid \theta) p(\theta)$$

Predictive Uncertainty

A useful principle for understanding cold start.

$$\text{dataset } \mathcal{D} := \{(x_n, a_n, r_n)_{n=1}^{N}\}$$

$$\text{predictive distribution } p(R \mid \mathcal{D}, A, X) = \int \underbrace{p(R \mid A, X, \theta)}_{\text{intrinsic uncertainty}} \overbrace{p(\theta \mid \mathcal{D})}^{\substack{\text{parameter uncertainty} \\ \text{based on data}}} \mathrm{d}\theta$$

$$\text{where } p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta) p(\theta)$$

Where does the uncertainty come from?

— intrinsic uncertainty: how deterministic is behavior?

— data uncertainty: how much data do we have? how noisy is the data?

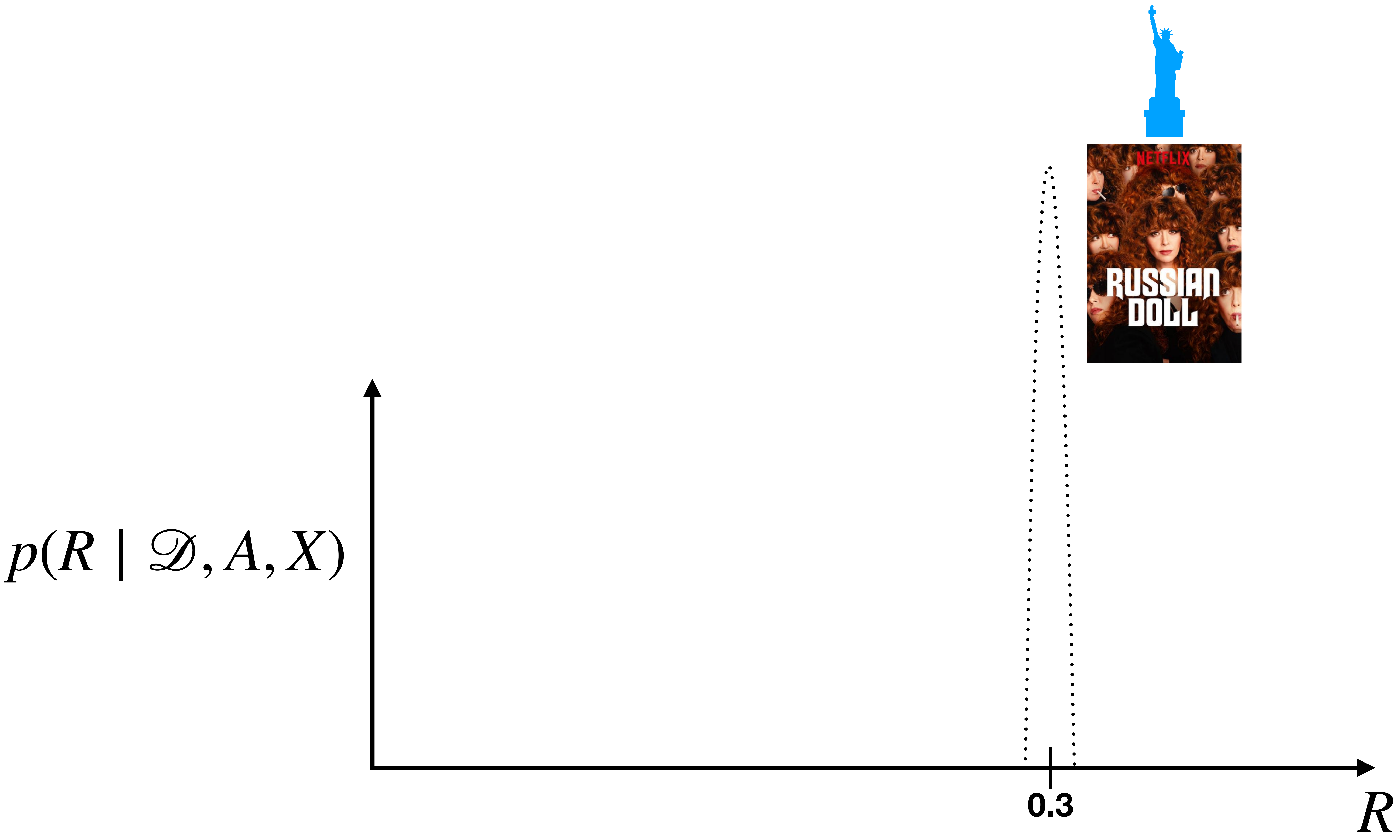
— (ignored here: model mismatch)

Predictive Uncertainty

A useful principle for understanding cold start.

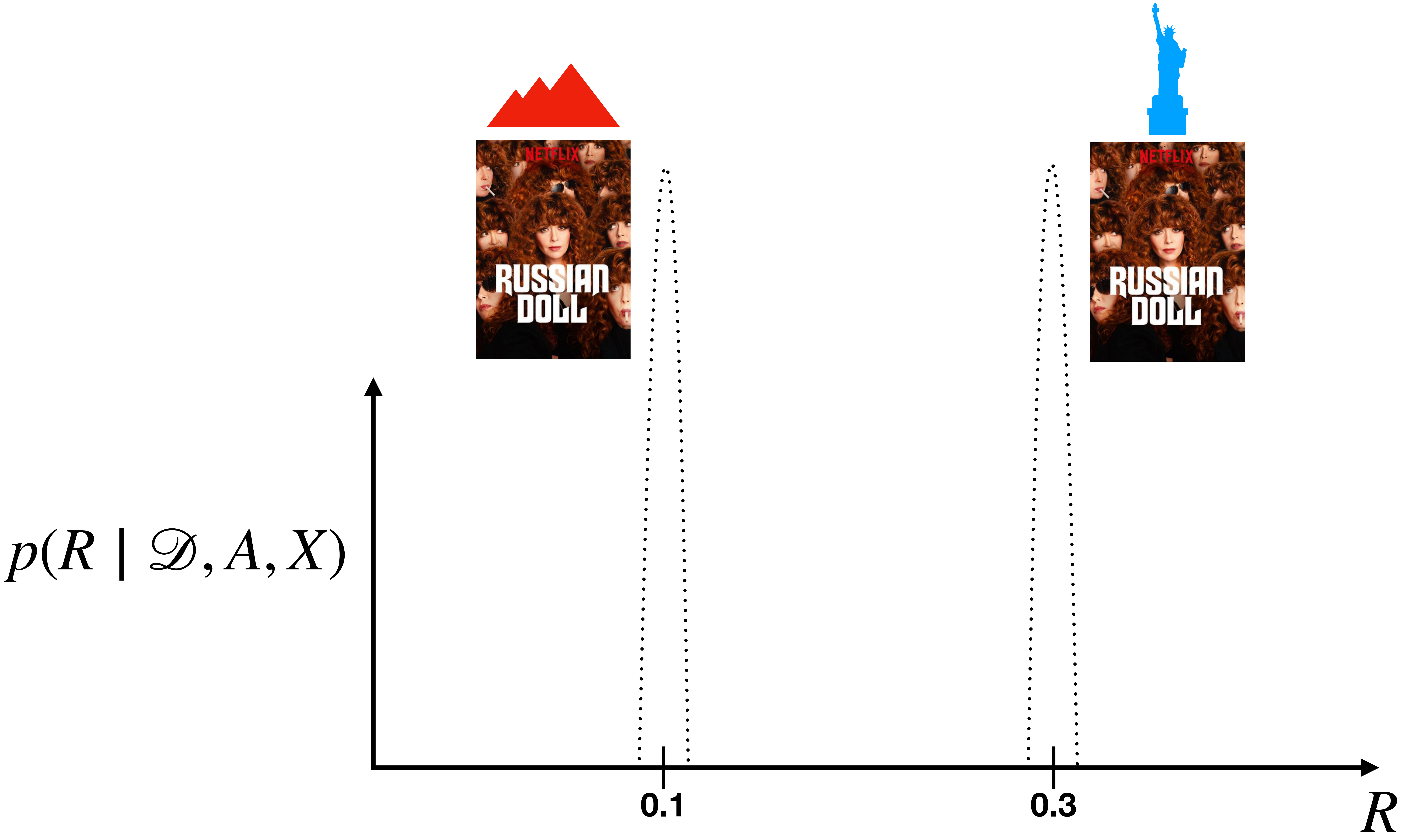
Predictive Uncertainty

A useful principle for understanding cold start.



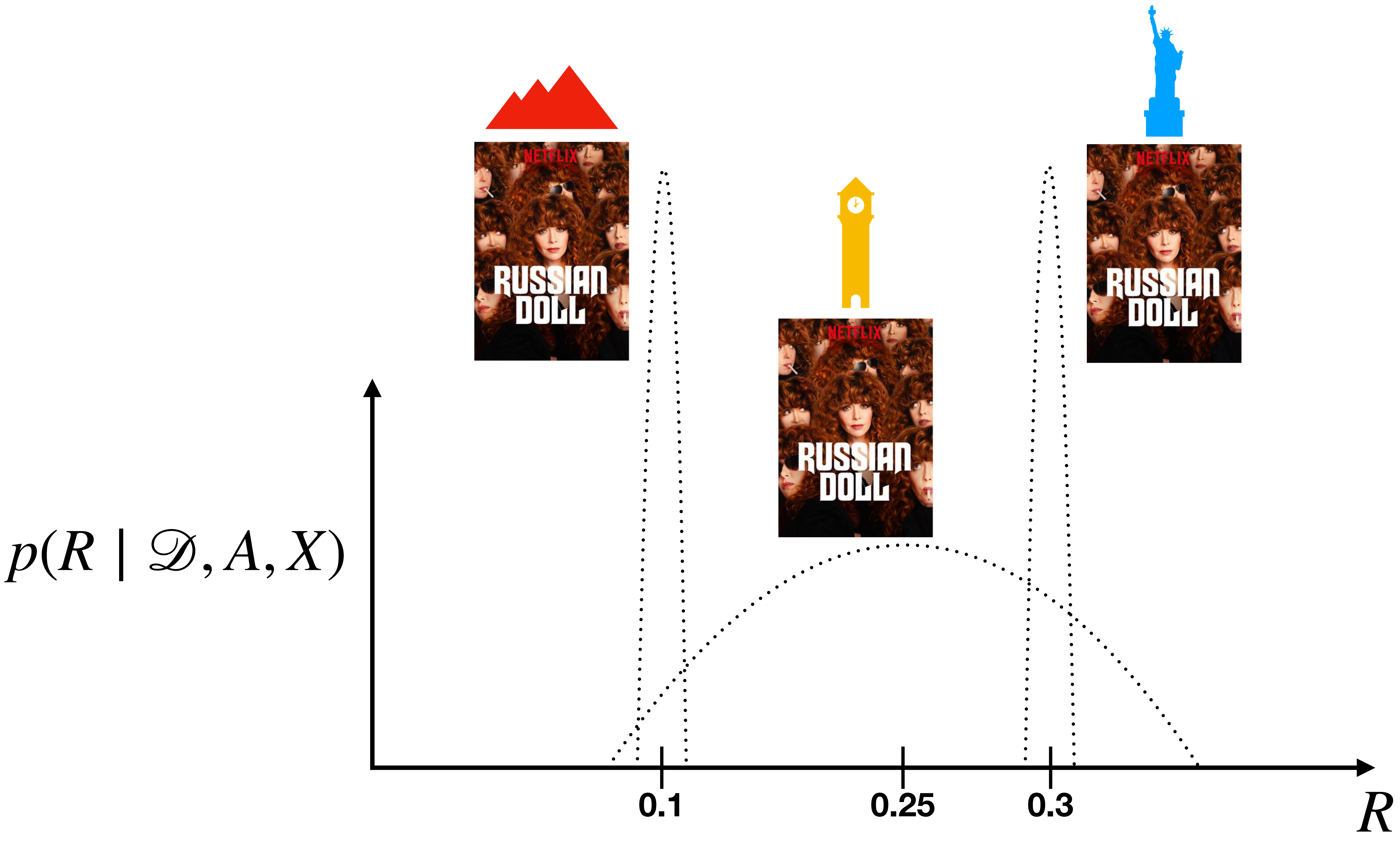
Predictive Uncertainty

A useful principle for understanding cold start.



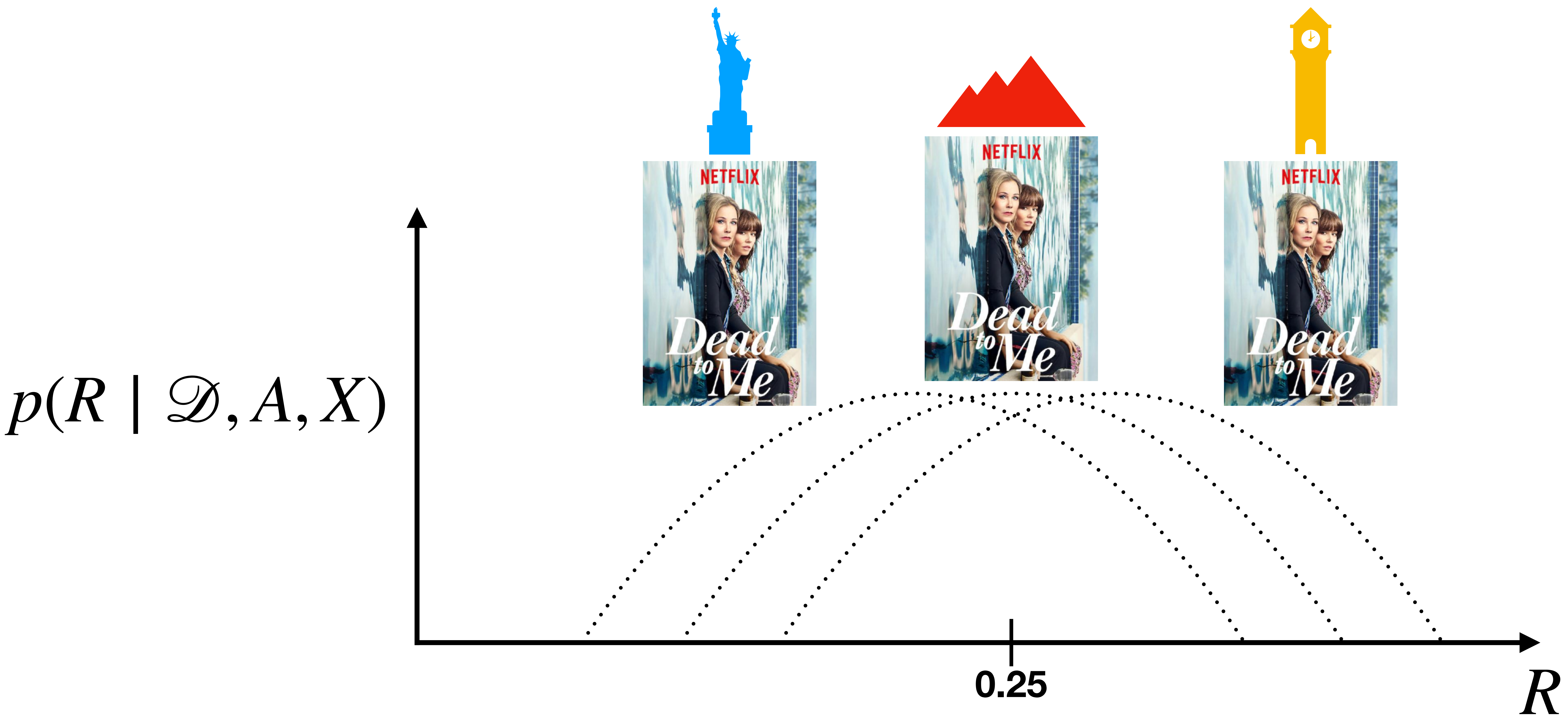
Predictive Uncertainty

A useful principle for understanding cold start.

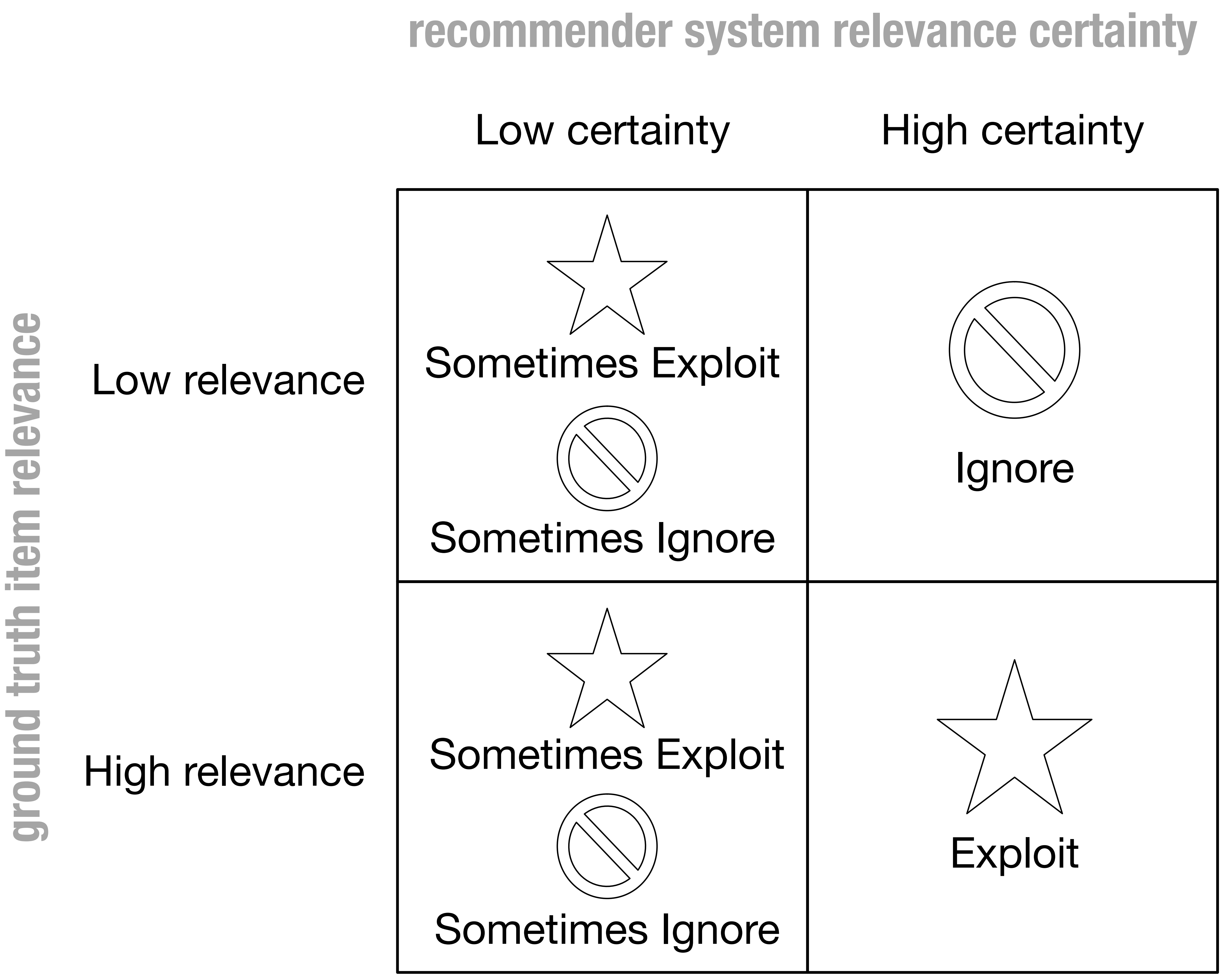


Predictive Uncertainty

A useful principle for understanding cold start.



Being Restricted to *Exploit* and *Ignore* Can Lead to Bad Decisions Under Uncertainty



Multi-armed bandits [Robbins, 1952]

1. enter the world with zero knowledge

repeat N times

2. pick $a_k \sim \pi$
3. observe a payoff r_n

Multi-armed bandits [Robbins, 1952]

repeat N times

1. enter the world with zero knowledge
2. pick $a_k \sim \pi$
3. observe a payoff r_n

$$\text{regret} = \max_a \sum_{n=1}^{N} (\text{reward}(a) - r_n)$$

Multi-armed bandits [Robbins, 1952]

1. enter the world with zero knowledge

repeat N times

2. pick $a_k \sim \pi$
3. observe a payoff r_n

$$\text{regret} = \max_a \sum_{n=1}^{N} (\text{reward}(a) - r_n)$$

how to choose a policy to minimize regret?

Contextual bandits [Abe et al. 2003]

repeat N times

1. enter the world with zero knowledge
2. observe a context x_n
3. pick $a_k \sim \pi(A \mid x_n)$
4. observe a payoff r_n

$$\text{regret} = \max_{\pi'} \sum_{n=1}^{N} (\text{reward}(\pi'(x_n)) - r_n)$$

Contextual bandits [Abe et al. 2003]

repeat N times

1. enter the world with zero knowledge
2. observe a context x_n
3. pick $a_k \sim \pi(A \mid x_n)$
4. observe a payoff r_n

$$\text{regret} = \max_{\pi'} \sum_{n=1}^{N} (\text{reward}(\pi'(x_n)) - r_n)$$

Bandit

1. enter the world with zero knowledge
2. pick $a_k \sim \pi(A)$
3. observe a payoff r_n

$$\text{regret} = \max_{a} \sum_{n=1}^{N} (\text{reward}(a) - r_n)$$

Beyond Pure Exploration

Beyond Pure Exploration

- Enter exploration-exploitation

Beyond Pure Exploration

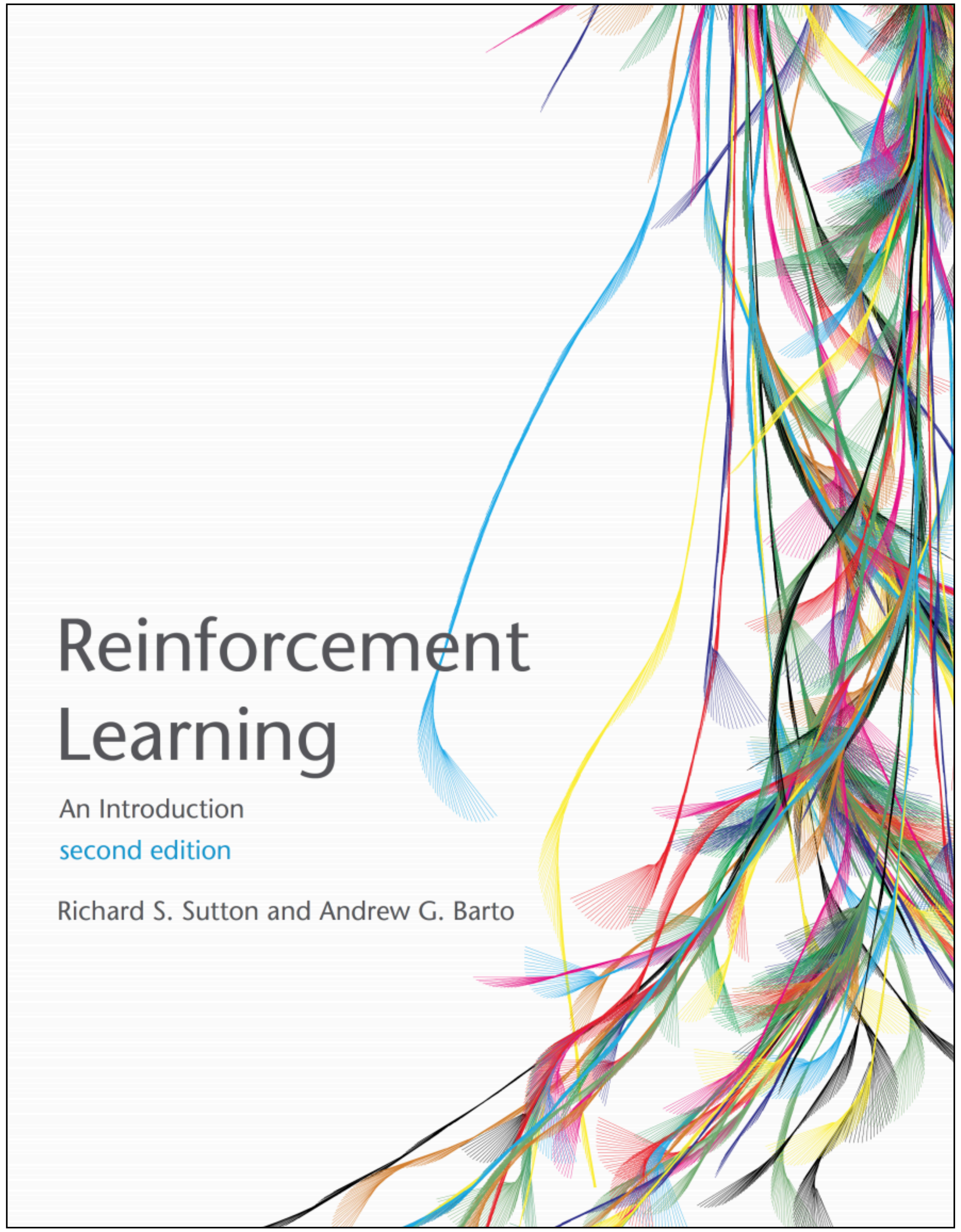
- Enter exploration-exploitation

recommender system relevance certainty

ground truth item relevance

	Low certainty	High certainty
Low relevance	Explore	Ignore
High relevance	Explore	Exploit

Sutton & Barto, 2018 (first edition 1998)



http://incompleteideas.net/book/the-book-2nd.html

How to Balance Exploration with Exploitation?

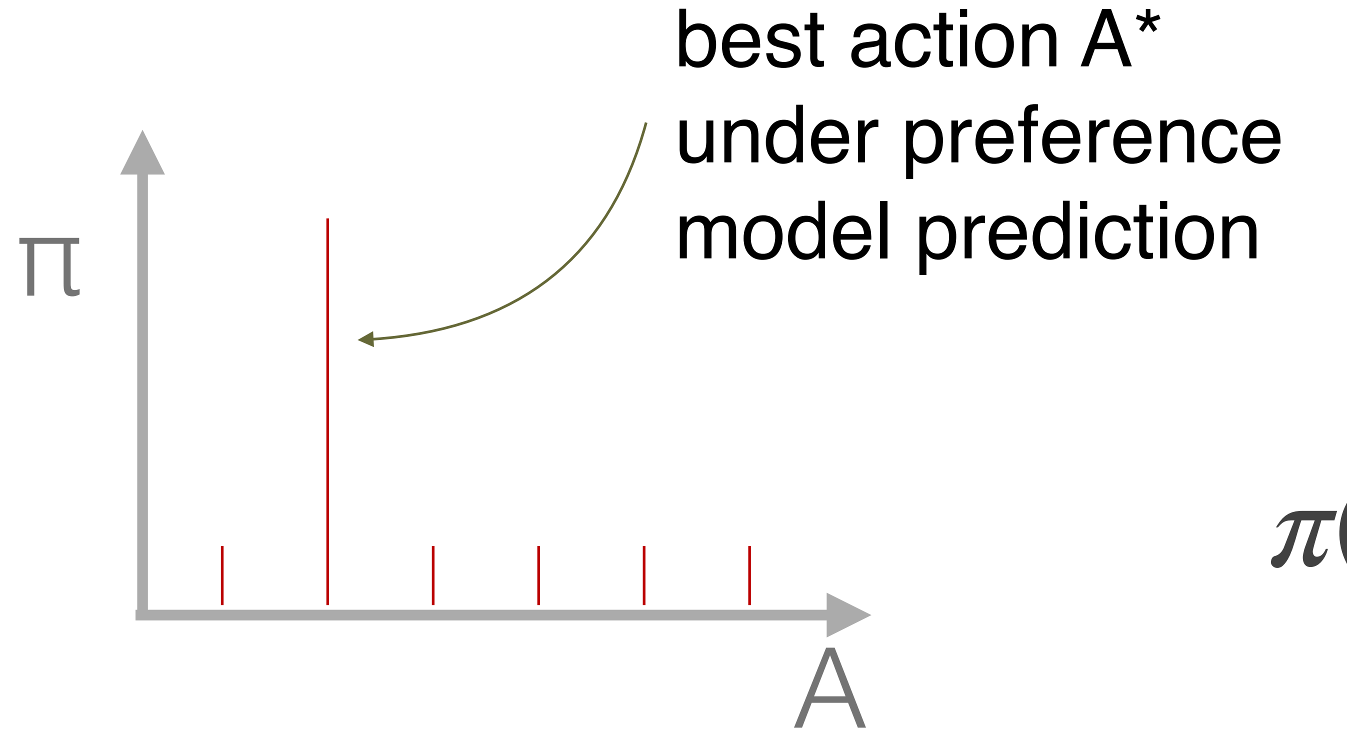
How to Balance Exploration with Exploitation?

- Simplest method is epsilon-greedy

How to Balance Exploration with Exploitation?

— Simplest method is epsilon-greedy

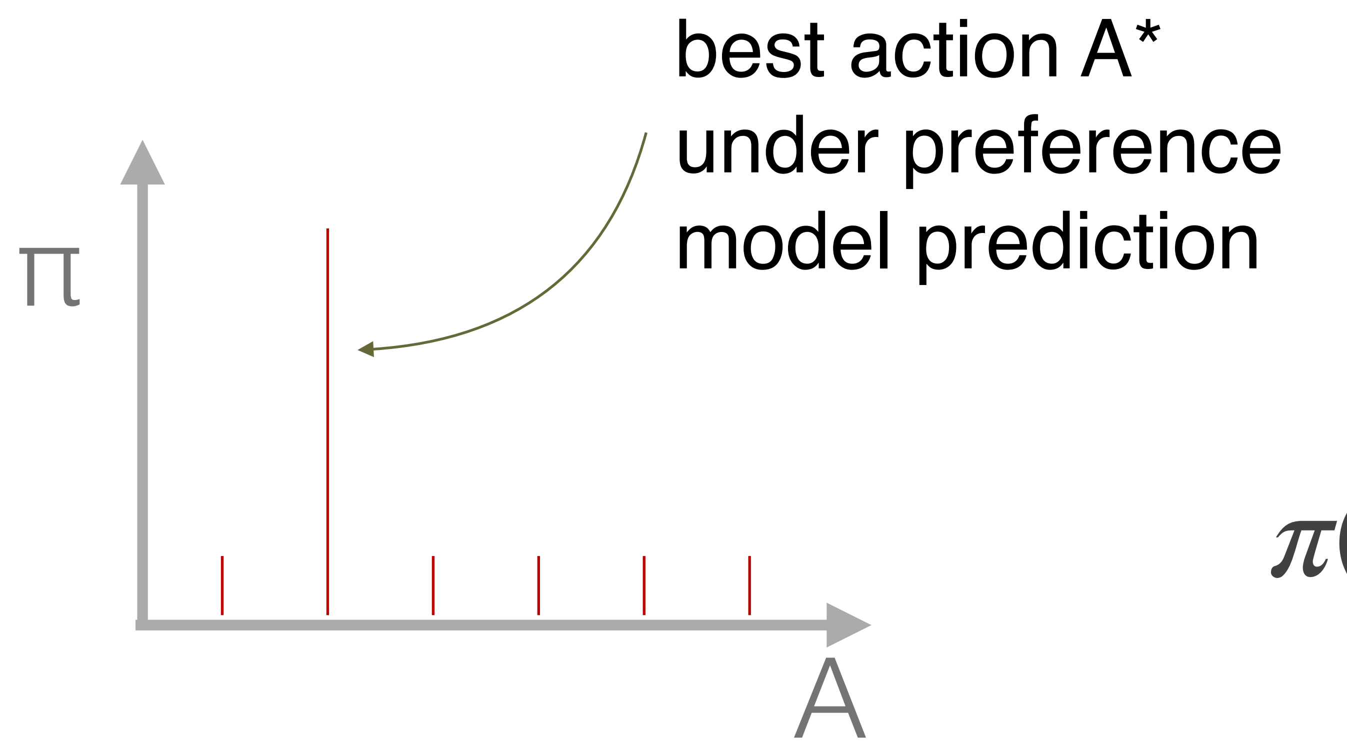
ε-greedy



$$\pi(A \mid X) = \begin{cases} (1-\epsilon) + \frac{\epsilon}{|\mathscr{A}|} & \text{when } A = A^* \\ \frac{\epsilon}{|\mathscr{A}|} & \text{otherwise} \end{cases}$$

How to Balance Exploration with Exploitation?

— Simplest method is epsilon-greedy



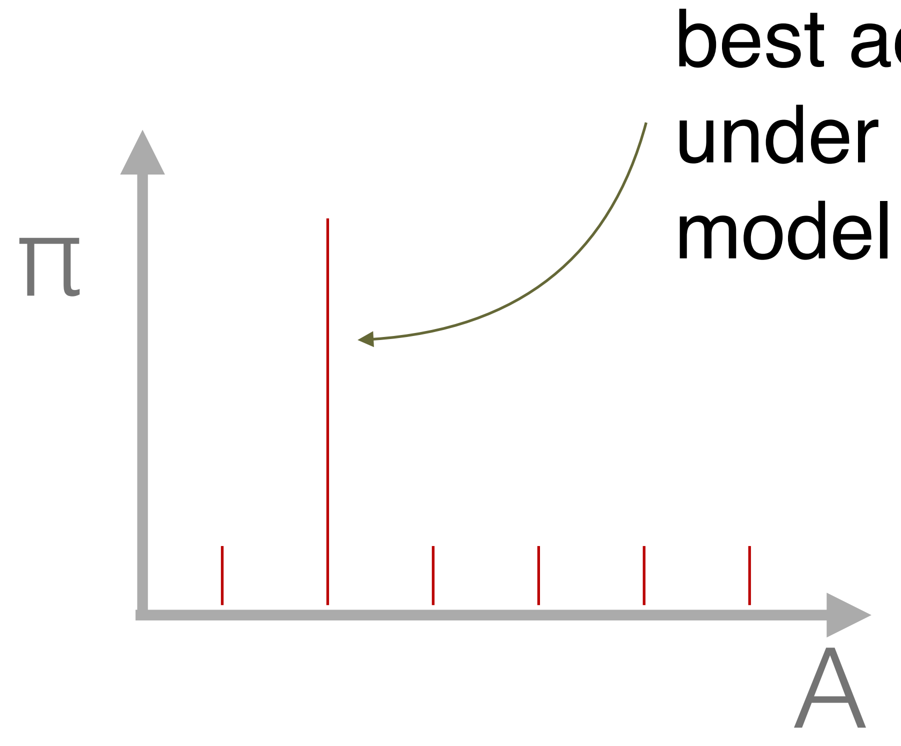
ε-greedy

exploration rate (can also decay over time)

$$\pi(A \mid X) = \begin{cases} (1-\epsilon) + \frac{\epsilon}{|\mathcal{A}|} & \text{when } A = A^* \\ \frac{\epsilon}{|\mathcal{A}|} & \text{otherwise} \end{cases}$$

How to Balance Exploration with Exploitation?

— Simplest method is epsilon-greedy



ε-greedy

exploration rate (can also decay over time)

$$\pi(A \mid X) = \begin{cases} (1-\epsilon) + \frac{\epsilon}{|\mathscr{A}|} & \text{when } A = A^* \\ \frac{\epsilon}{|\mathscr{A}|} & \text{otherwise} \end{cases}$$

treats all sub-optimal arms the same

regret linear in N

How to Balance Exploration with Exploitation?

- Thompson sampling [Thompson, 1933]

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

$$\text{predictive mean } \mathbb{E}(R \mid \mathscr{D}, A, X) = \int \mathbb{E}(R \mid A, X, \theta) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

$$\text{predictive mean } \mathbb{E}(R \mid \mathscr{D}, A, X) = \int \mathbb{E}(R \mid A, X, \theta) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

$$\pi(A \mid X) = \int \mathbb{I}(\mathbb{E}[R \mid A, X, \theta] = \max_{A'} \mathbb{E}[R \mid A', X, \theta]) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

$$\text{predictive mean } \mathbb{E}(R \mid \mathscr{D}, A, X) = \int \mathbb{E}(R \mid A, X, \theta) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

$$\pi(A \mid X) = \int \mathbb{I}(\underbrace{\mathbb{E}[R \mid A, X, \theta]}_{\text{avg reward with action A}} = \max_{A'} \mathbb{E}[R \mid A', X, \theta]) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

$$\text{predictive mean } \mathbb{E}(R \mid \mathcal{D}, A, X) = \int \mathbb{E}(R \mid A, X, \theta) p(\theta \mid \mathcal{D}) \mathrm{d}\theta$$

$$\pi(A \mid X) = \int \mathbb{I}\Big(\underbrace{\mathbb{E}[R \mid A, X, \theta]}_{\text{avg reward with action A}} = \underbrace{\max_{A'} \mathbb{E}[R \mid A', X, \theta]}_{\text{avg reward with optimal action}}\Big) p(\theta \mid \mathcal{D}) \mathrm{d}\theta$$

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

$$\text{predictive mean } \mathbb{E}(R \mid \mathcal{D}, A, X) = \int \mathbb{E}(R \mid A, X, \theta) p(\theta \mid \mathcal{D}) \mathrm{d}\theta$$

$$\pi(A \mid X) = \int \underbrace{\mathbb{I}}_{\text{Iverson bracket}}(\underbrace{\mathbb{E}[R \mid A, X, \theta]}_{\text{avg reward with action A}} = \underbrace{\max_{A'} \mathbb{E}[R \mid A', X, \theta]}_{\text{avg reward with optimal action}}) p(\theta \mid \mathcal{D}) \mathrm{d}\theta$$

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

$$\text{predictive mean } \mathbb{E}(R \mid \mathscr{D}, A, X) = \int \mathbb{E}(R \mid A, X, \theta) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

$$\pi(A \mid X) = \int \mathbb{I}(\mathbb{E}[R \mid A, X, \theta] = \max_{A'} \mathbb{E}[R \mid A', X, \theta]) p(\theta \mid \mathscr{D}) \mathrm{d}\theta$$

avg reward with action A

avg reward with optimal action

simulate parameters from posterior distribution

Iverson bracket

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Regret depends on the sum of observed rewards —> optimize average reward

$$\text{predictive mean } \mathbb{E}(R \mid \mathcal{D}, A, X) = \int \mathbb{E}(R \mid A, X, \theta) p(\theta \mid \mathcal{D}) \mathrm{d}\theta$$

$$\pi(A \mid X) = \int \underbrace{\mathbb{I}}_{\text{Iverson bracket}}(\underbrace{\mathbb{E}[R \mid A, X, \theta]}_{\text{avg reward with action A}} = \underbrace{\max_{A'} \mathbb{E}[R \mid A', X, \theta]}_{\text{avg reward with optimal action}}) \underbrace{p(\theta \mid \mathcal{D}) \mathrm{d}\theta}_{\text{simulate parameters from posterior distribution}}$$

how to implement in practice?

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Monte Carlo approximation:

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Monte Carlo approximation:

$$\pi(A \mid X) \approx \mathbb{I}(\mathbb{E}[R \mid A, X, \theta'] = \max_{A'} \mathbb{E}[R \mid A', X, \theta']) \quad \text{where } \theta' \sim p(\theta \mid \mathscr{D})$$

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Monte Carlo approximation:

$$\pi(A \mid X) \approx \mathbb{I}(\mathbb{E}[R \mid A, X, \theta'] = \max_{A'} \mathbb{E}[R \mid A', X, \theta']) \quad \text{where } \theta' \sim p(\theta \mid \mathcal{D})$$

Algorithm:

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

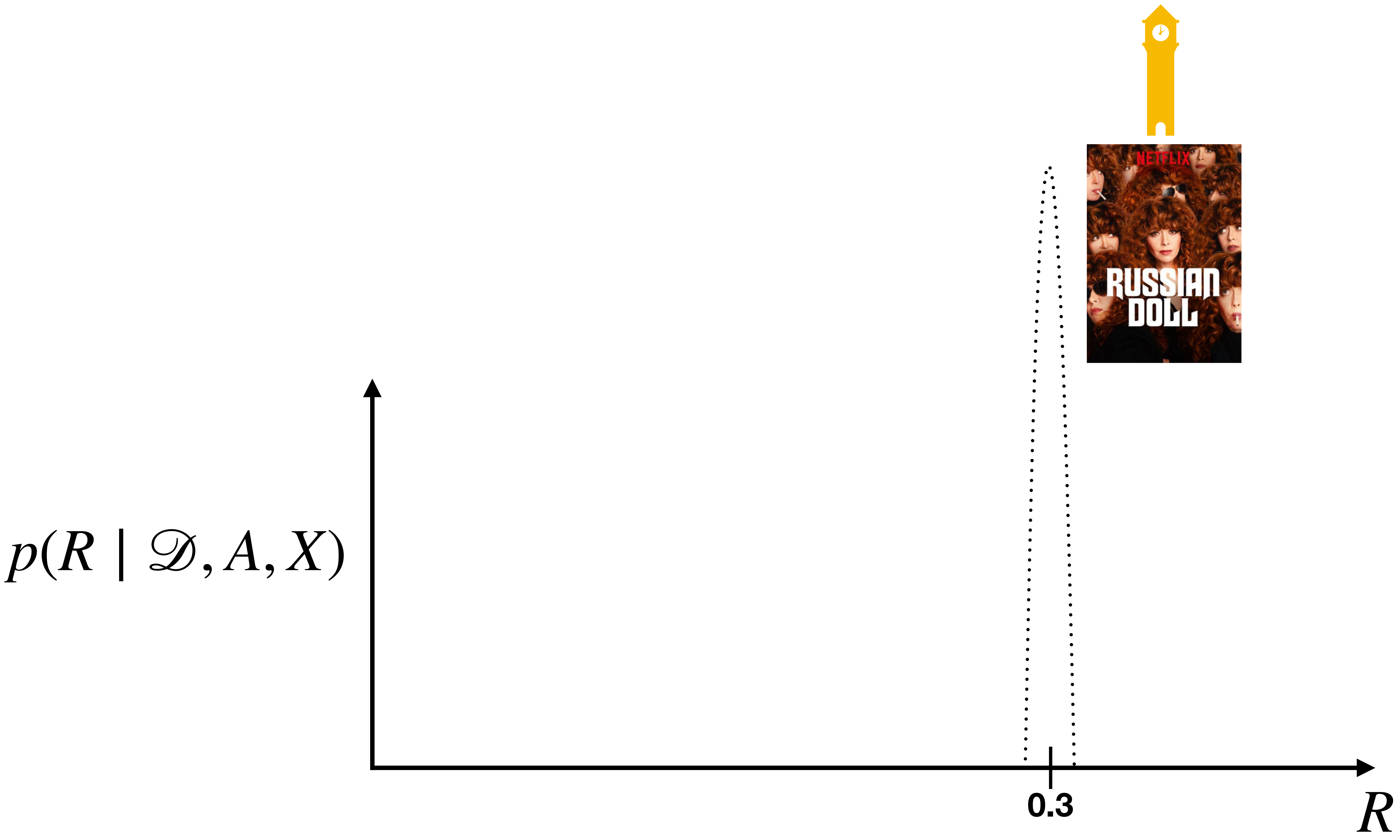
Monte Carlo approximation:

$$\pi(A \mid X) \approx \mathbb{I}(\mathbb{E}[R \mid A, X, \theta'] = \max_{A'} \mathbb{E}[R \mid A', X, \theta']) \quad \text{where } \theta' \sim p(\theta \mid \mathscr{D})$$

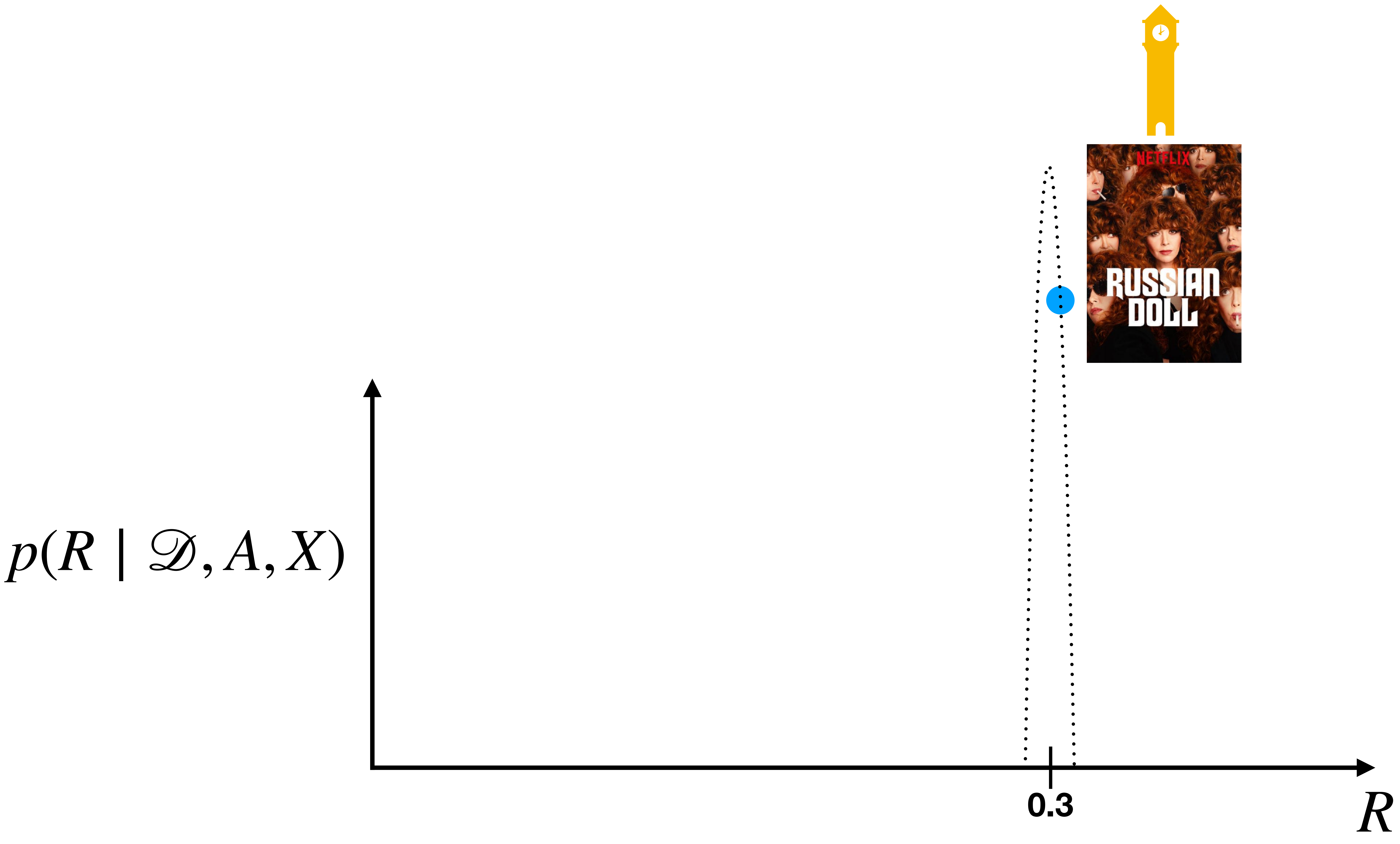
Algorithm:

1. pick a model $p(R \mid A, X, \theta)$ and $p(\theta)$
2. initialize $q_1(\theta) = p(\theta)$
3. for n = 1 … N:
    4. observe x_n
    5. sample $\theta' \sim q_n(\theta)$
    6. pick $A' = \arg\max_A \mathbb{E}[R \mid A, X, \theta']$
    7. observe r_n
    8. update $q_{n+1}(\theta) \propto q_n(\theta) p(r_n \mid a_n, x_n, \theta)$

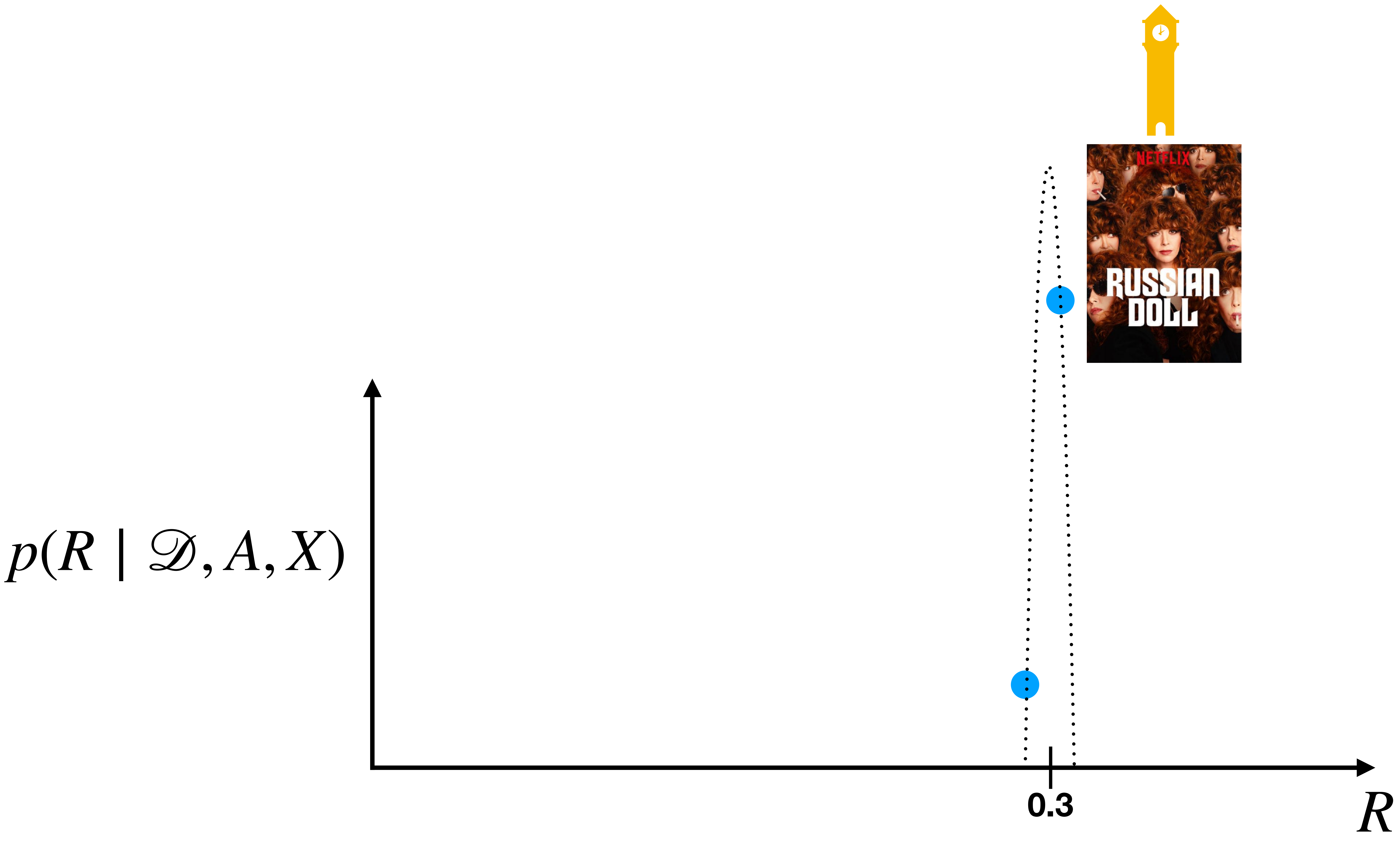
Why Does Thompson Sampling Work?



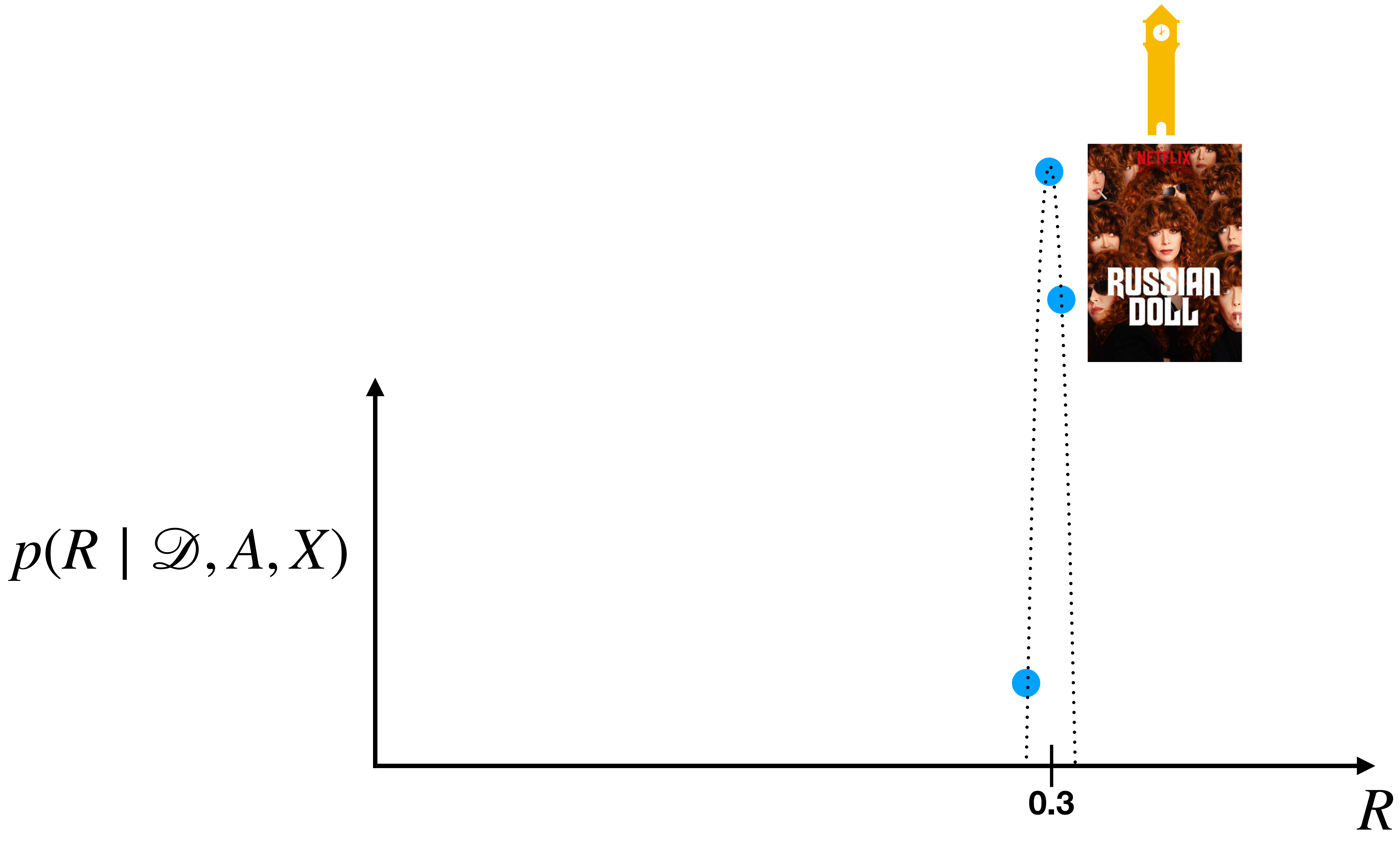
Why Does Thompson Sampling Work?



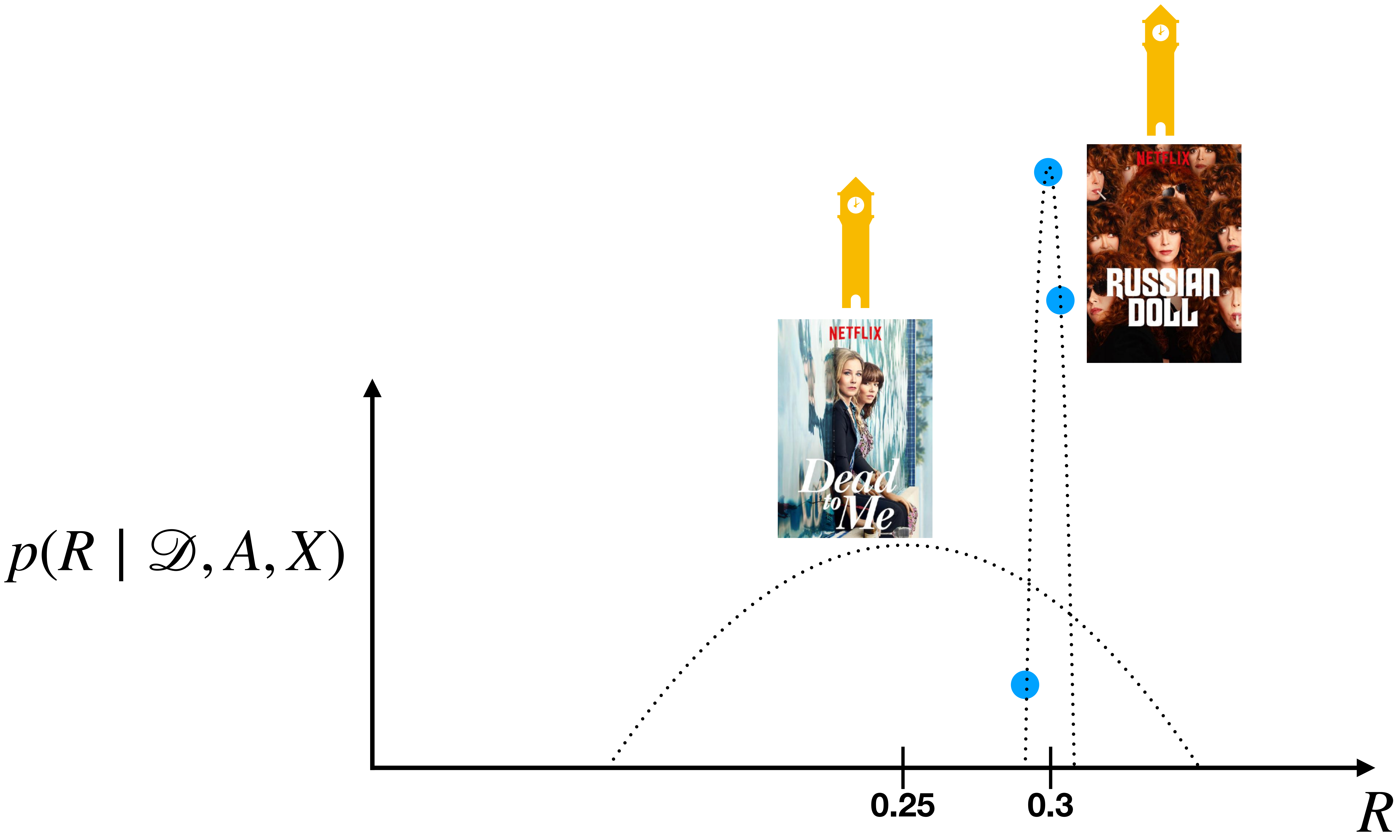
Why Does Thompson Sampling Work?



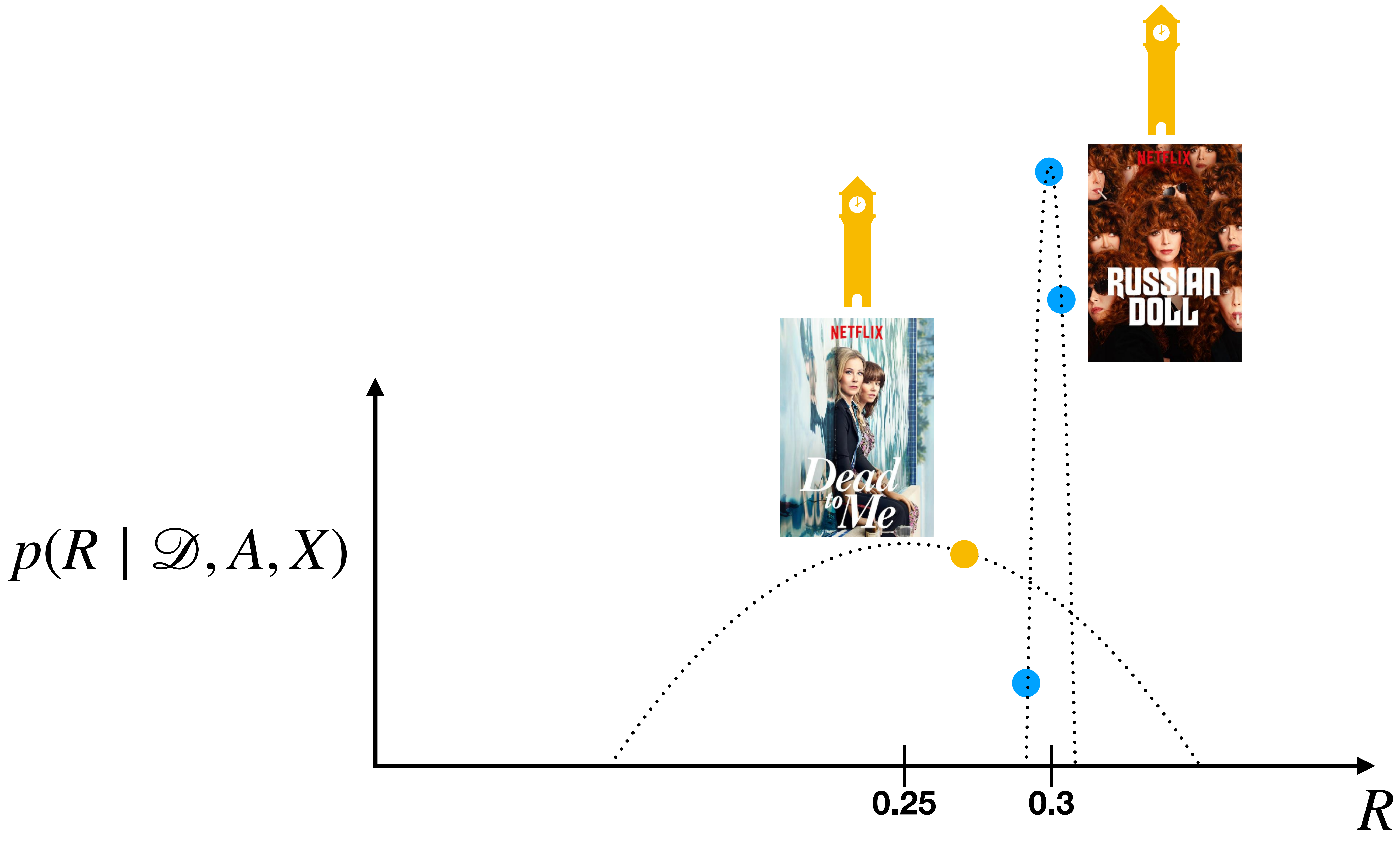
Why Does Thompson Sampling Work?



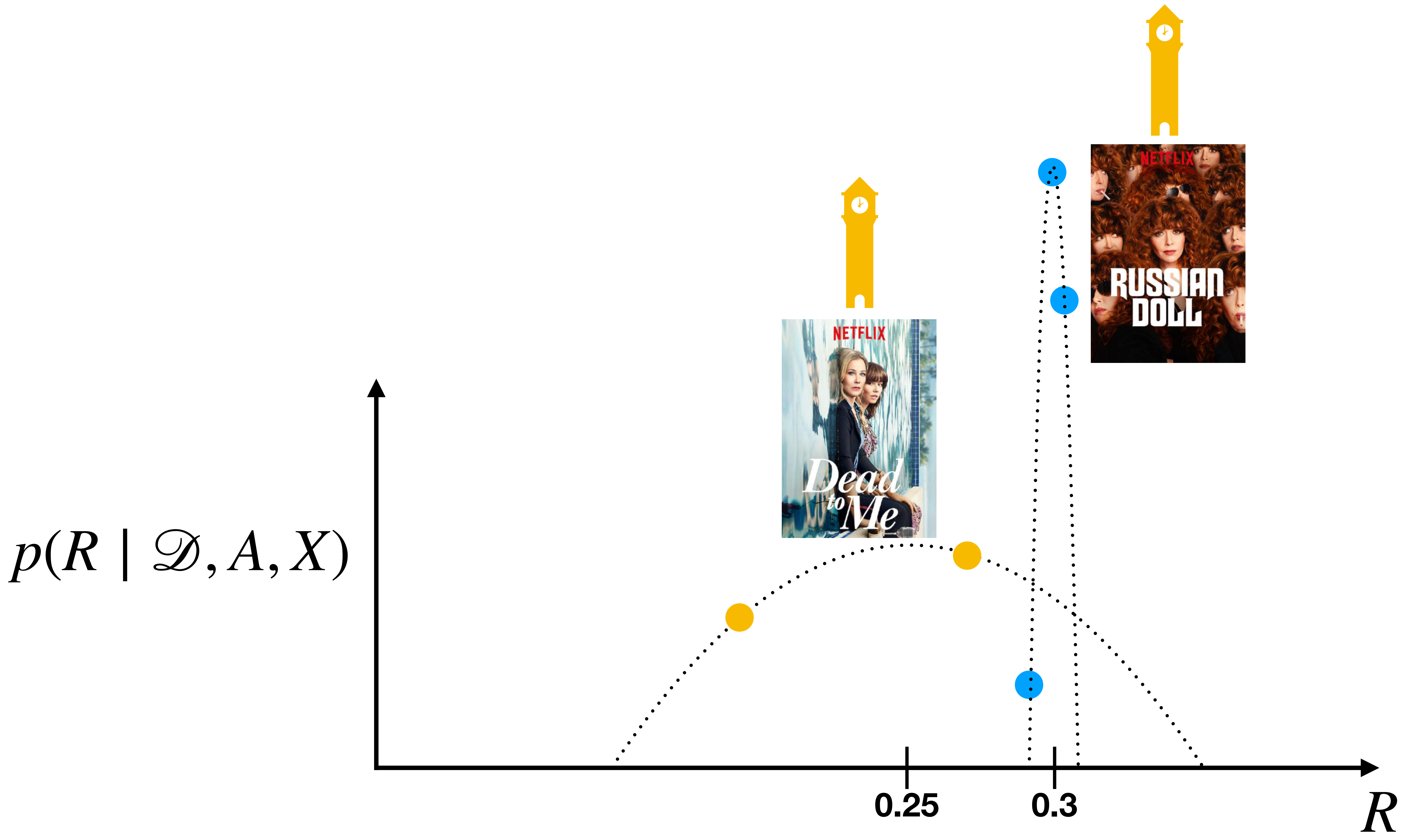
Why Does Thompson Sampling Work?



Why Does Thompson Sampling Work?

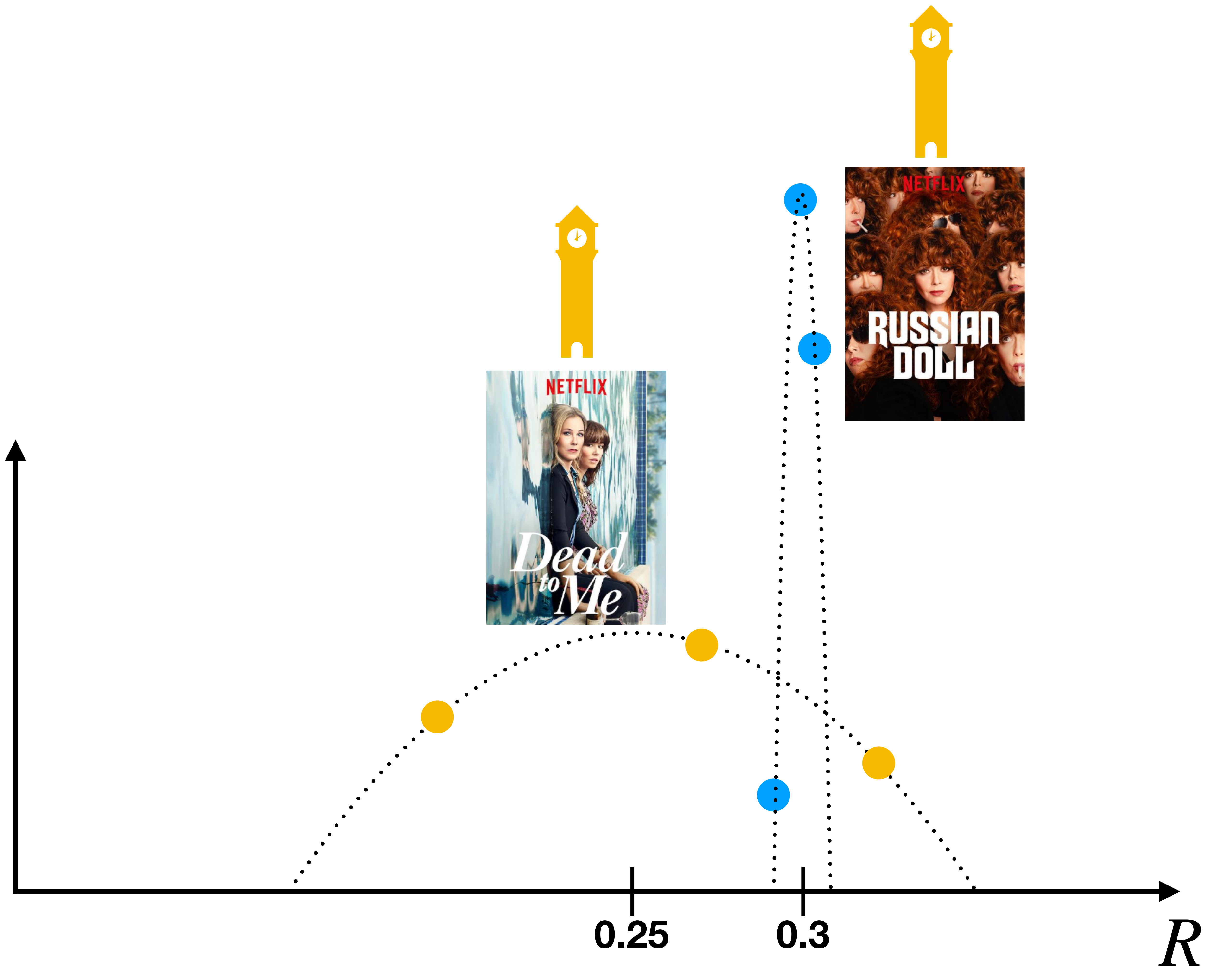


Why Does Thompson Sampling Work?



Why Does Thompson Sampling Work?

$p(R \mid \mathscr{D}, A, X)$



How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Beta-Bernoulli Example:

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Beta-Bernoulli Example:

Algorithm:

1. initialize $S_{1,i,k} = \alpha, F_{1,i,k} = \beta \quad \text{for } i = 1,\ldots,D \quad \text{for } k = 1,\ldots,K$
2. for n = 1 … N:
   3. observe x_n
   4. sample $\theta_k' \sim \mathrm{Beta}(S_{n,x_n,k}, F_{n,x_n,k}) \quad \text{for } k = 1,\ldots,K$
   5. pick $A' = \arg\max_k \theta_k'$
   6. observe r_n
   7. update $S_{n+1,i,A'} \leftarrow S_{n+1,i,A'} + r_n, \quad F_{n+1,i,A'} \leftarrow F_{n+1,i,A'} + (1 - r_n)$

How to Balance Exploration with Exploitation?

— Thompson sampling [Thompson, 1933]

Beta-Bernoulli Example:

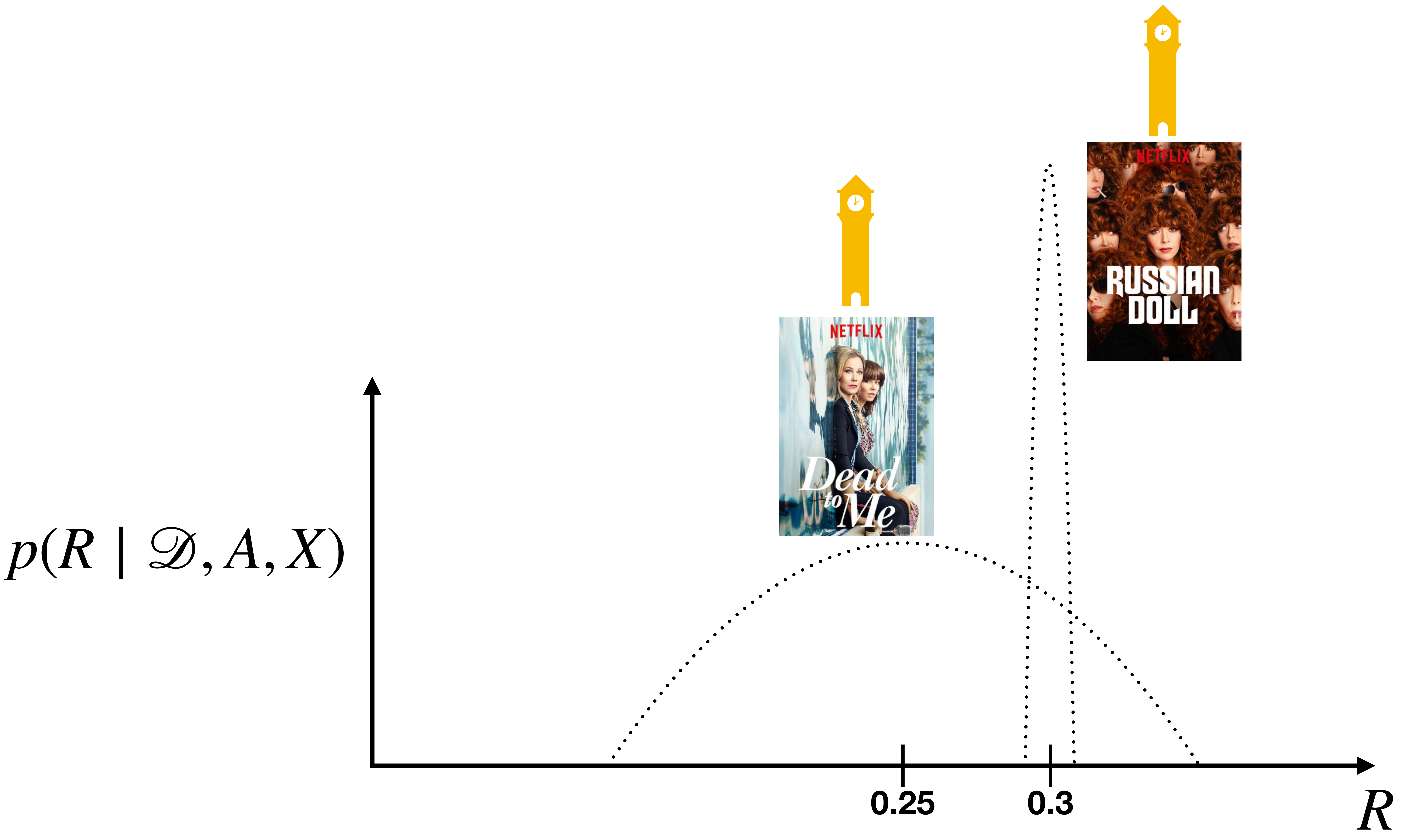
Algorithm:

1. initialize $S_{1,i,k} = \alpha, F_{1,i,k} = \beta \quad \text{for } i = 1,\ldots,D \quad \text{for } k = 1,\ldots,K$
2. for n = 1 … N:
   3. observe x_n
   4. sample $\theta'_k \sim \text{Beta}(S_{n,x_n,k}, F_{n,x_n,k}) \quad \text{for } k = 1,\ldots,K$
   5. pick $A' = \arg\max_k \theta'_k$
   6. observe r_n
   7. update $S_{n+1,i,A'} \leftarrow S_{n+1,i,A'} + r_n, \quad F_{n+1,i,A'} \leftarrow F_{n+1,i,A'} + (1 - r_n)$

$$\text{regret} \leq \text{O}(\sqrt{KN \log N})$$ [Agrawal & Goyal, 2013]

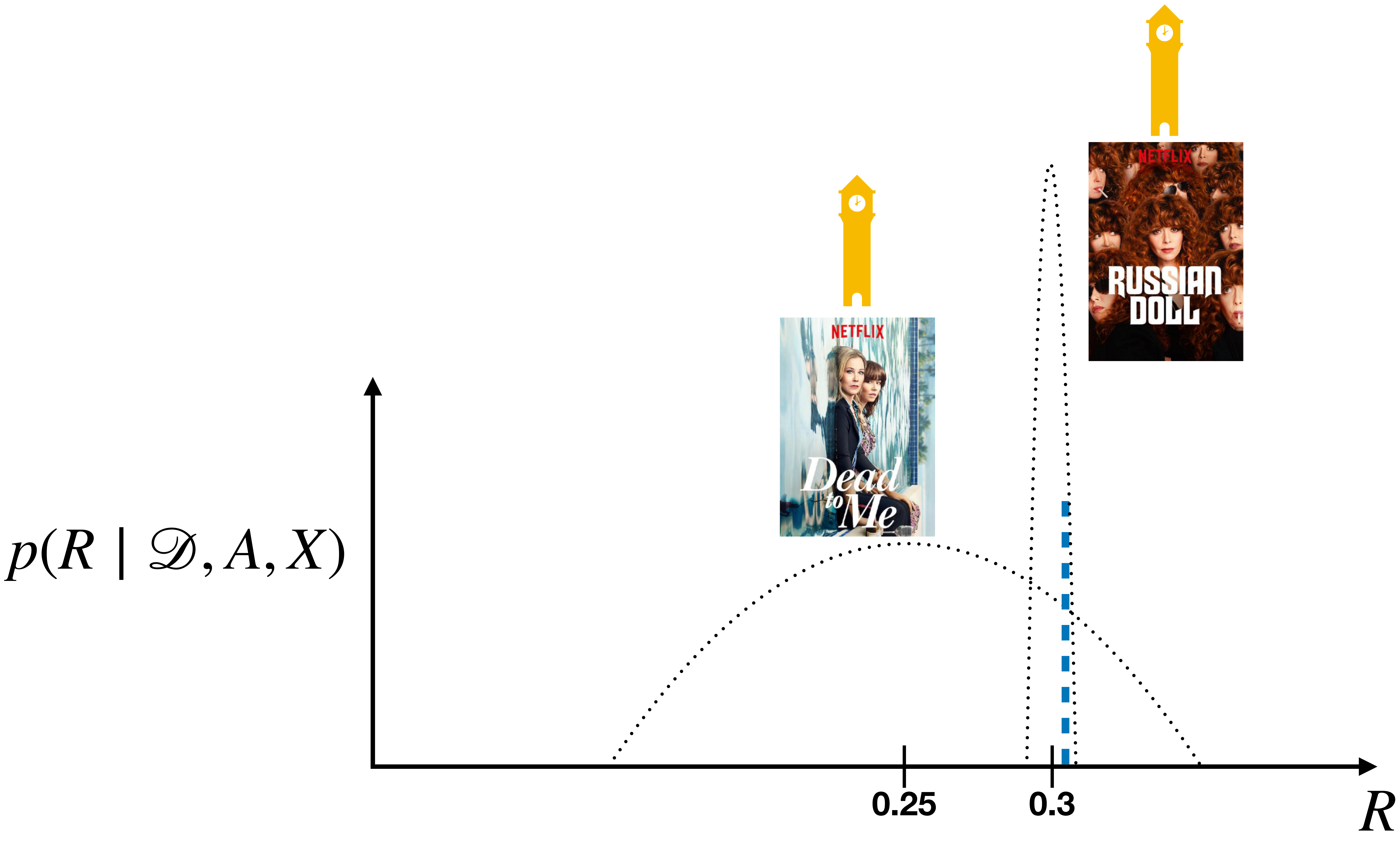
How to Balance Exploration with Exploitation?

— Upper confidence bound



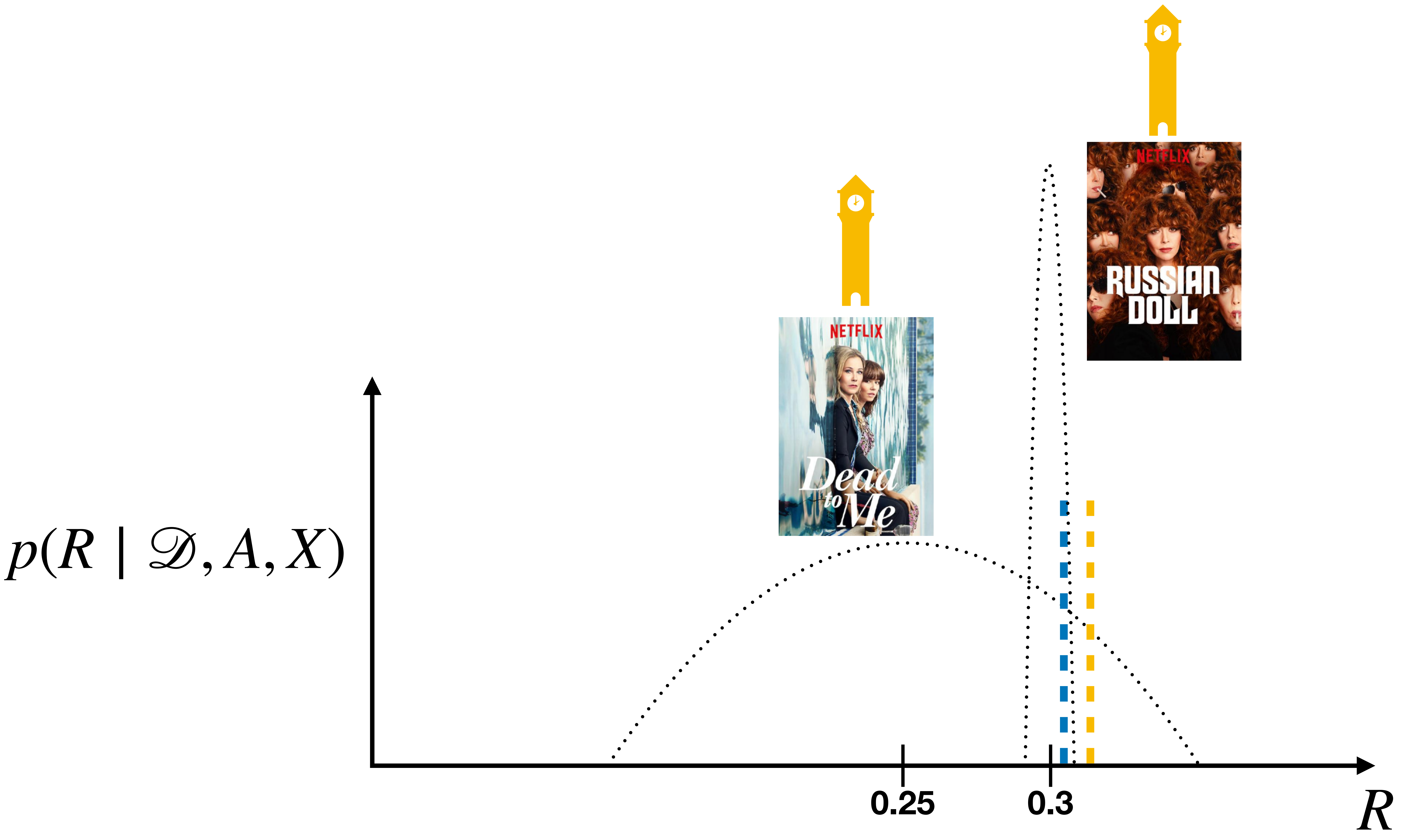
How to Balance Exploration with Exploitation?

— Upper confidence bound



How to Balance Exploration with Exploitation?

— Upper confidence bound



How to Balance Exploration with Exploitation?

- Upper confidence bound [Auer et al. 2002]

How to Balance Exploration with Exploitation?

— Upper confidence bound
[Auer et al. 2002]

$$A' = \arg\max_{A} \mathbb{E}[R \mid A, X] + c\sqrt{\frac{\log N}{N_A}}$$

How to Balance Exploration with Exploitation?

— Upper confidence bound
[Auer et al. 2002]

total number of trials

$$A' = \arg\max_A \mathbb{E}[R \mid A, X] + c\sqrt{\frac{\log N}{N_A}}$$

How to Balance Exploration with Exploitation?

— Upper confidence bound
[Auer et al. 2002]

total number of trials

$$A' = \arg\max_A \mathbb{E}[R \mid A, X] + c\sqrt{\frac{\log N}{N_A}}$$

number of times A
has been tried before

How to Balance Exploration with Exploitation?

— Upper confidence bound [Auer et al. 2002]

$$A' = \arg\max_{A} \mathbb{E}[R \mid A, X] + \overset{\text{confidence score}}{c}\sqrt{\frac{\overbrace{\log N}^{\text{total number of trials}}}{\underbrace{N_A}_{\text{number of times A has been tried before}}}}$$

How to Balance Exploration with Exploitation?

— Upper confidence bound [Auer et al. 2002]

confidence score

total number of trials

$$A' = \arg\max_A \mathbb{E}[R \mid A, X] + c\sqrt{\frac{\log N}{N_A}}$$

number of times A has been tried before

score = $0.3 + 0.05$ optimism bonus

How to Balance Exploration with Exploitation?

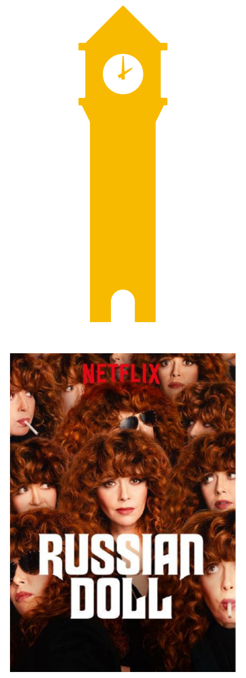
— Upper confidence bound [Auer et al. 2002]

confidence score

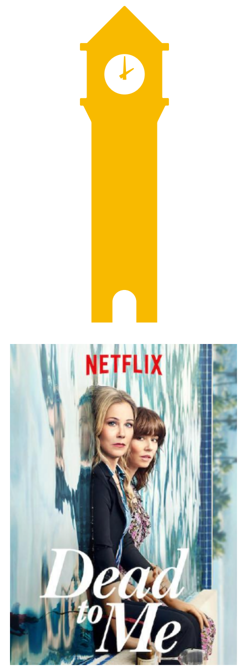
total number of trials

$$A' = \arg\max_A \mathbb{E}[R \mid A, X] + c\sqrt{\frac{\log N}{N_A}}$$

number of times A has been tried before



score = $0.3 + 0.05$ optimism bonus



score = $0.25 + 0.2$ optimism bonus

How to Balance Exploration with Exploitation?

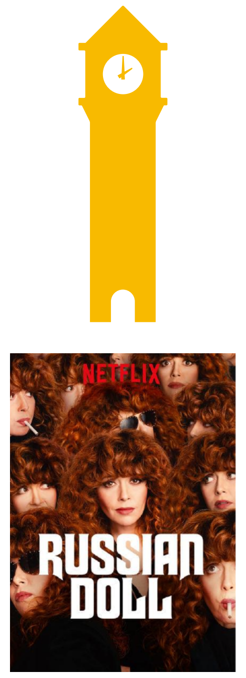
— Upper confidence bound
[Auer et al. 2002]

confidence score

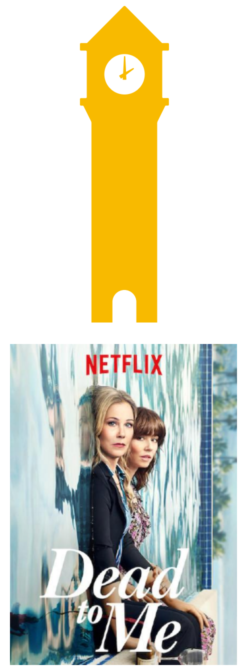
total number of trials

$$A' = \arg\max_{A} \mathbb{E}[R \mid A, X] + c\sqrt{\frac{\log N}{N_A}}$$

number of times A has been tried before



score = $0.3 + 0.05$ optimism bonus



score = $0.25 + 0.2$ optimism bonus

deterministic!

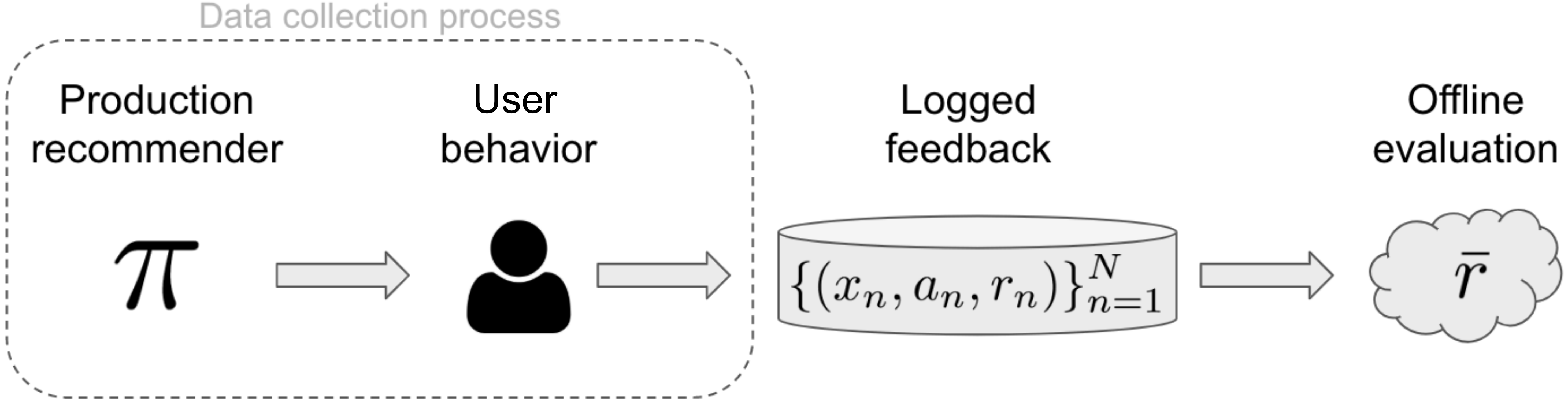
LinUCB for News Recommendation

[Li et al. 2012]

$$\mathbb{E}[R \mid X, A = k, \theta] = X_k^\top \theta_k$$

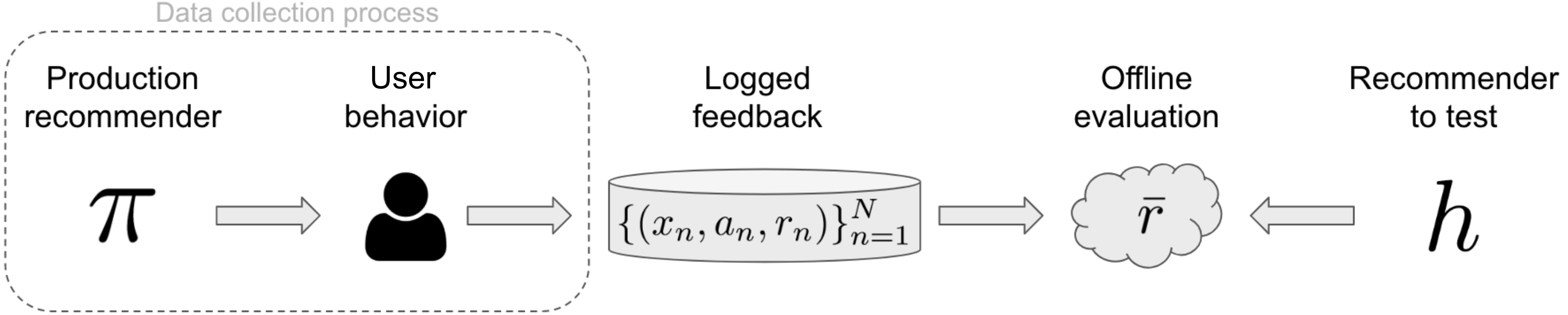
15 minute break

Evaluation Depends on the Method of Data Collection



- N is total number of impressions
- x_n is context of impression n (e.g. user vector, user demographics, time, content)
- a_n is the recommendation that production recommender π made for impression n
- r_n is reward of impression n after performing action
- $\bar{r}$ is average reward of recommender (e.g. stream rate, listening time)

Evaluation Depends on the Method of Data Collection



- N is total number of impressions
- x_n is context of impression n (e.g. user vector, user demographics, time, content)
- a_n is the recommendation that production recommender π made for impression n
- r_n is reward of impression n after performing action
- $\bar{r}$ is average reward of recommender (e.g. stream rate, listening time)

On-policy

— *evaluate* policy using data collected by the same policy

— *update* policy using data collected by the same policy

Off-policy

— *evaluate* policy using data collected by a different policy

— *learn* policy using data collected by a different policy

On-policy

— *evaluate* policy using data collected by the same policy

— *update* policy using data collected by the same policy

— vanilla bandits, simpler approach
— need to be able to tune the policy by interleaving recommendation with policy updating
— need to be able to evaluate possibly bad policies

Off-policy

— *evaluate* policy using data collected by a different policy

— *learn* policy using data collected by a different policy

On-policy

- *evaluate* policy using data collected by the same policy
- *update* policy using data collected by the same policy

- vanilla bandits, simpler approach
- need to be able to tune the policy by interleaving recommendation with policy updating
- need to be able to evaluate possibly bad policies

Off-policy

- *evaluate* policy using data collected by a different policy
- *learn* policy using data collected by a different policy

- common in contextual bandits
- much wider setting, but need methods to deal with policy mismatch

On-policy

"on-policy evaluation"

— *evaluate* policy using data collected by the same policy

— *update* policy using data collected by the same policy

"on-policy learning"

— vanilla bandits, simpler approach
— need to be able to tune the policy by interleaving recommendation with policy updating
— need to be able to evaluate possibly bad policies

Off-policy

"off-policy evaluation"

— *evaluate* policy using data collected by a different policy

— *learn* policy using data collected by a different policy

"off-policy learning"

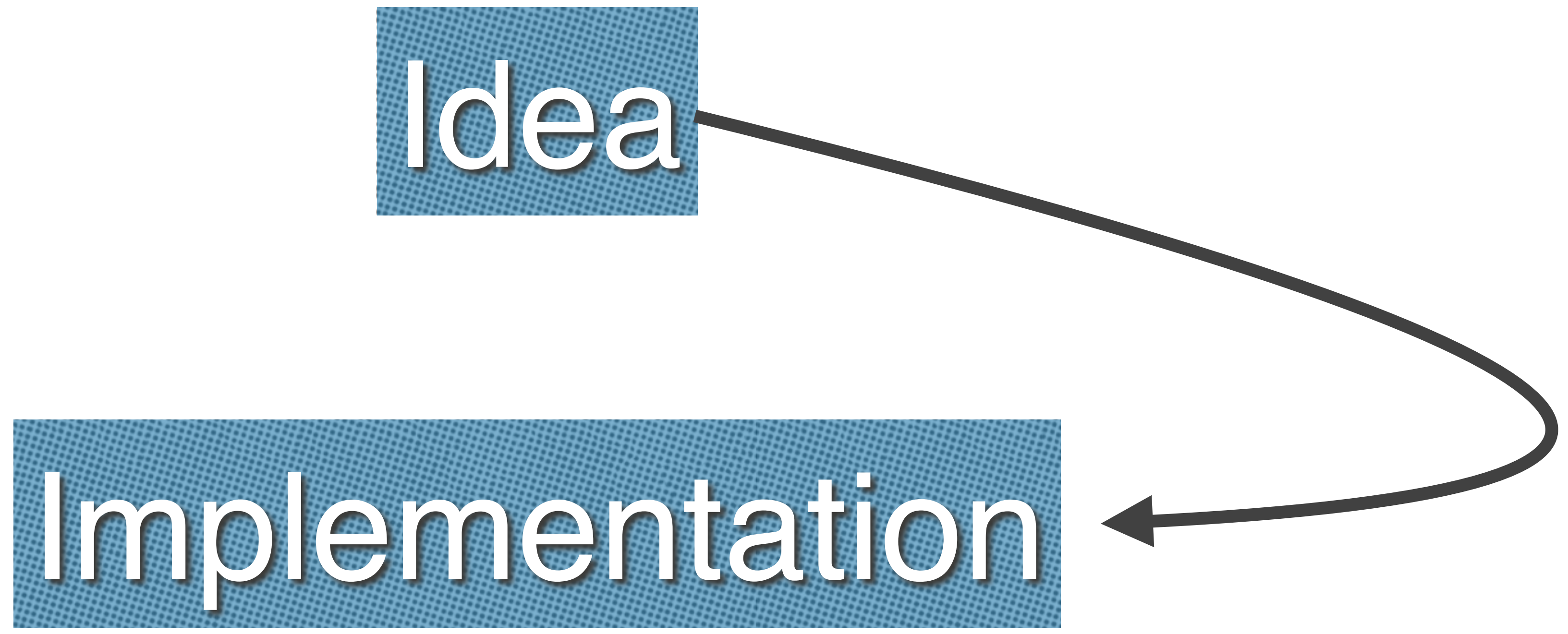
— common in contextual bandits

— much wider setting, but need methods to deal with policy mismatch

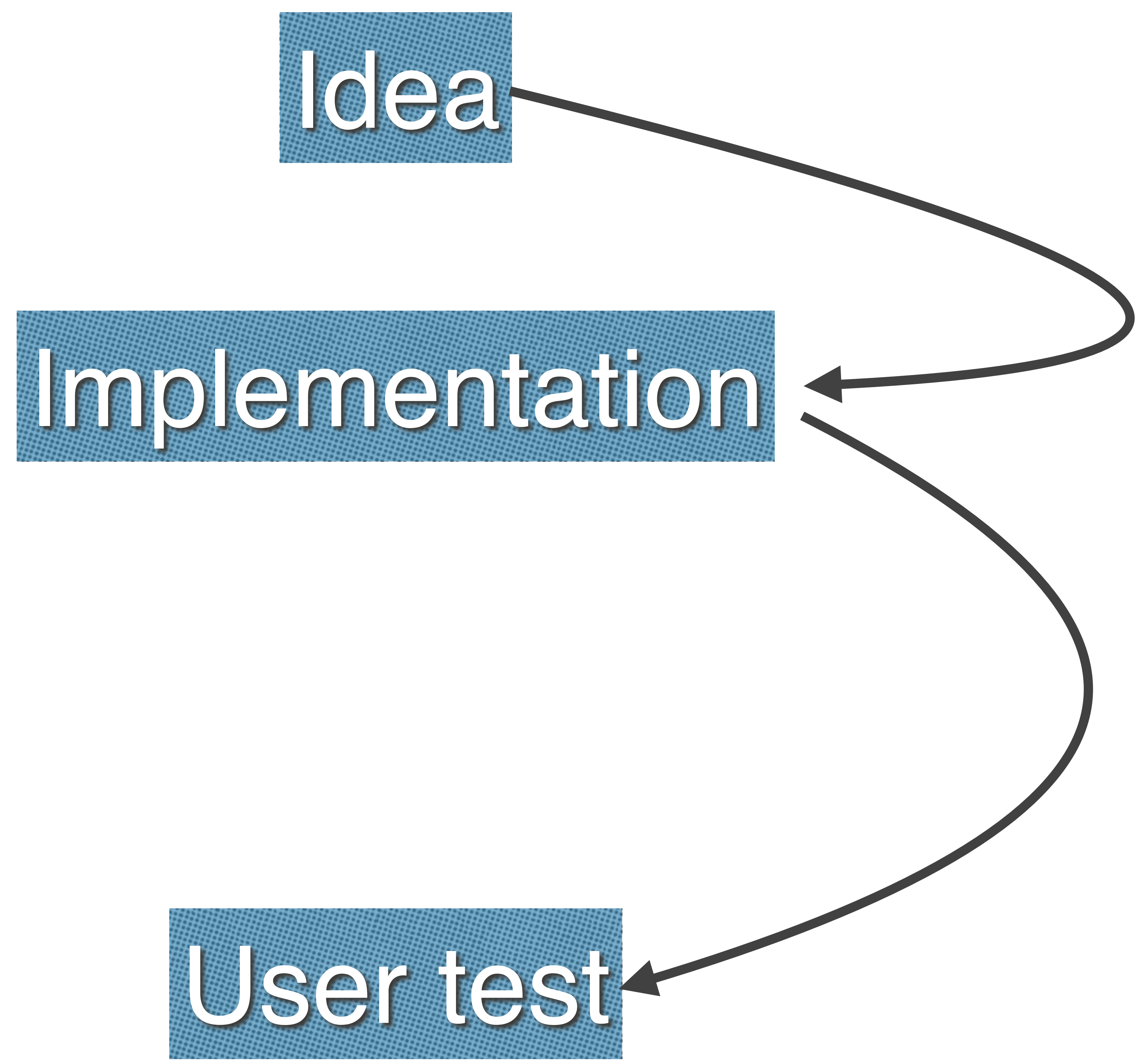
Offline Evaluation is Crucial for Innovation

Idea

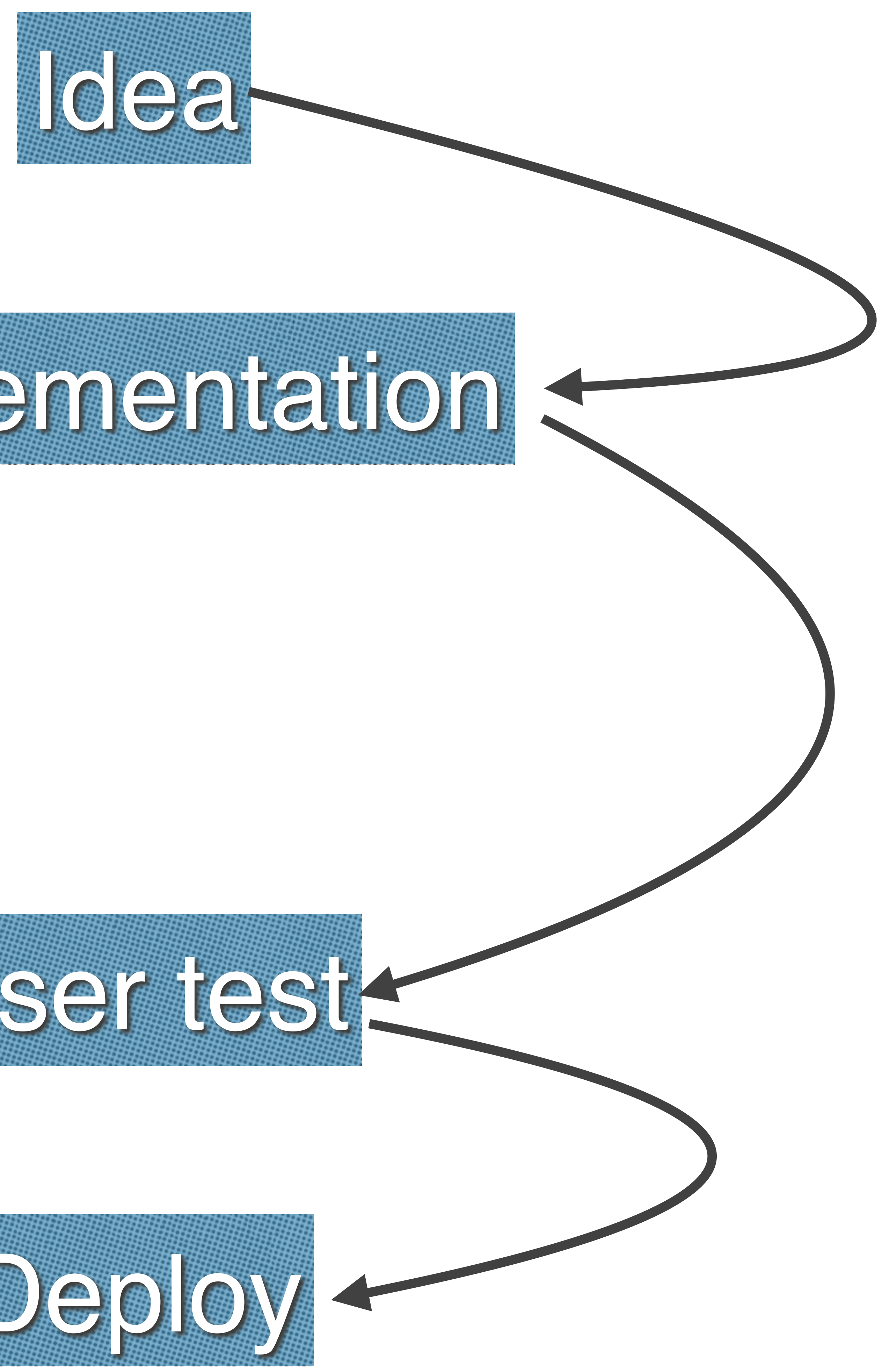
Offline Evaluation is Crucial for Innovation



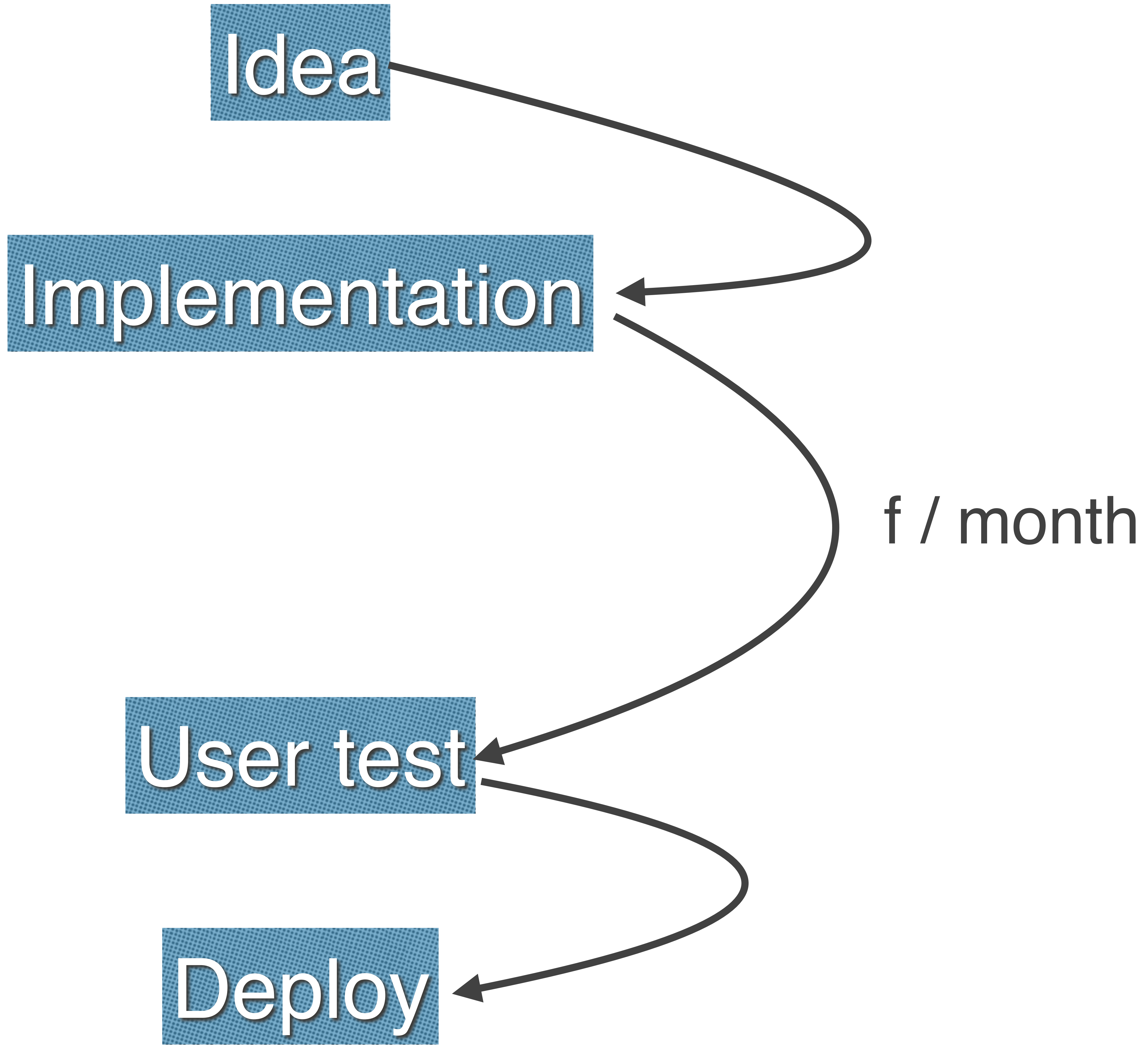
Offline Evaluation is Crucial for Innovation



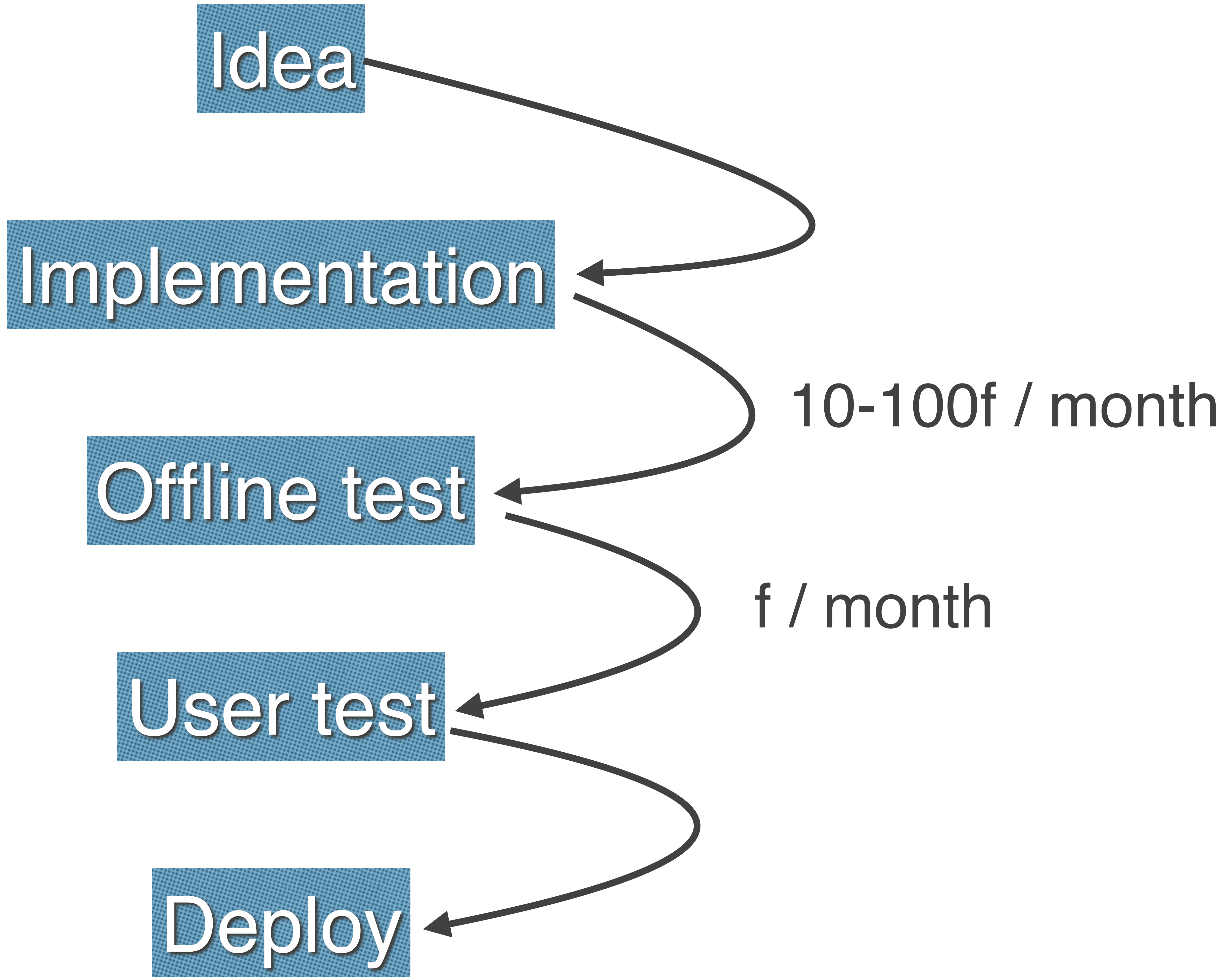
Offline Evaluation is Crucial for Innovation



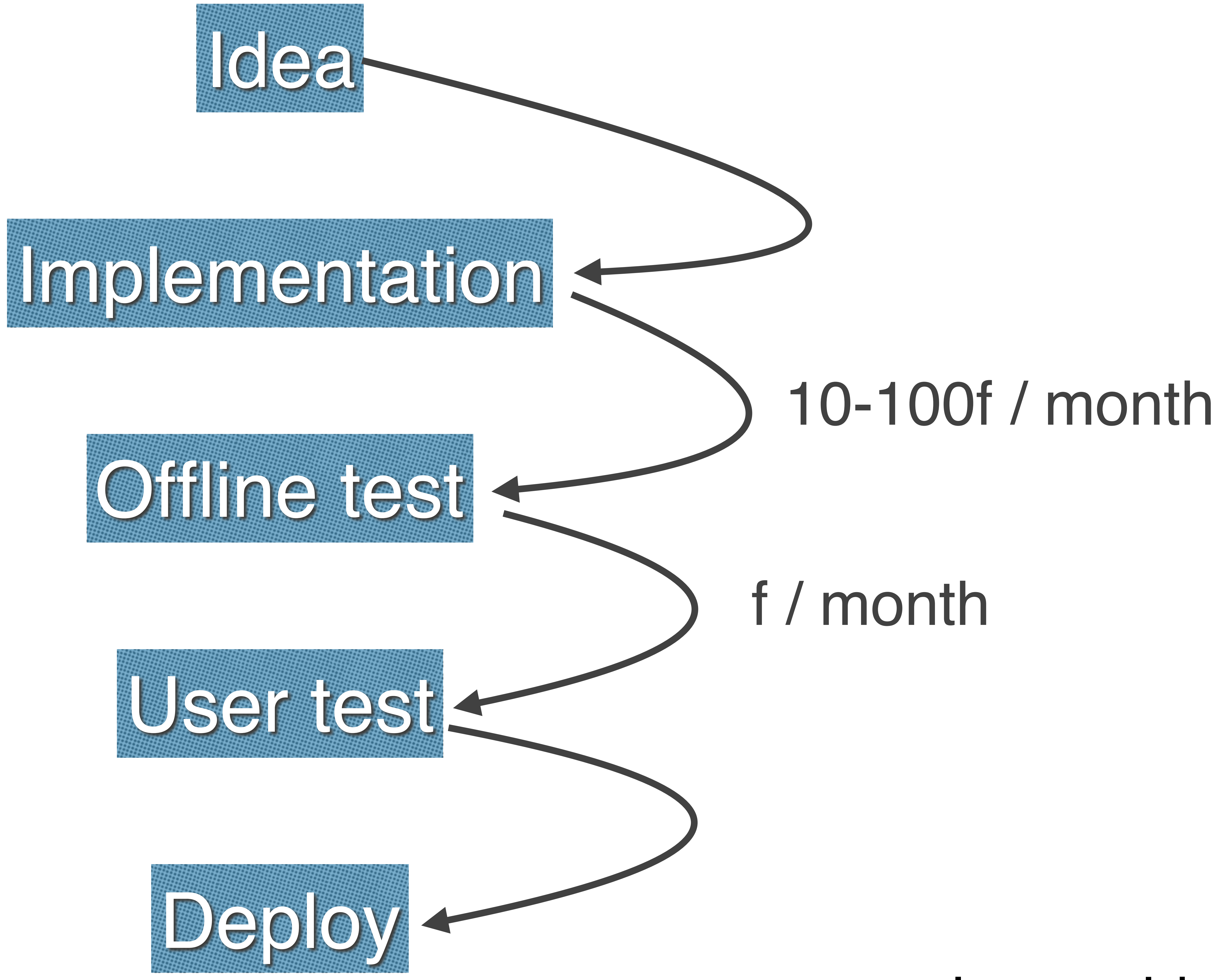
Offline Evaluation is Crucial for Innovation



Offline Evaluation is Crucial for Innovation

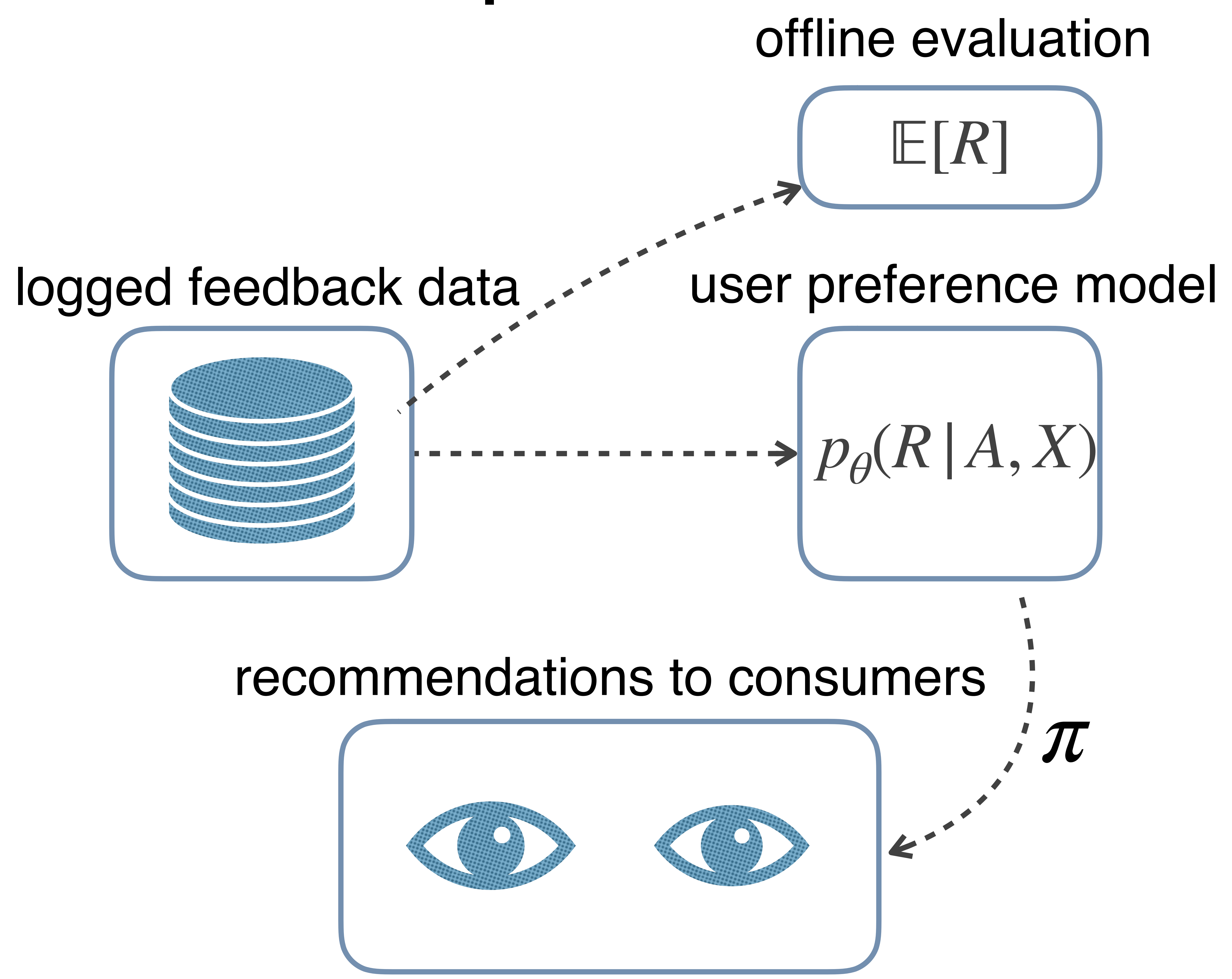


Offline Evaluation is Crucial for Innovation

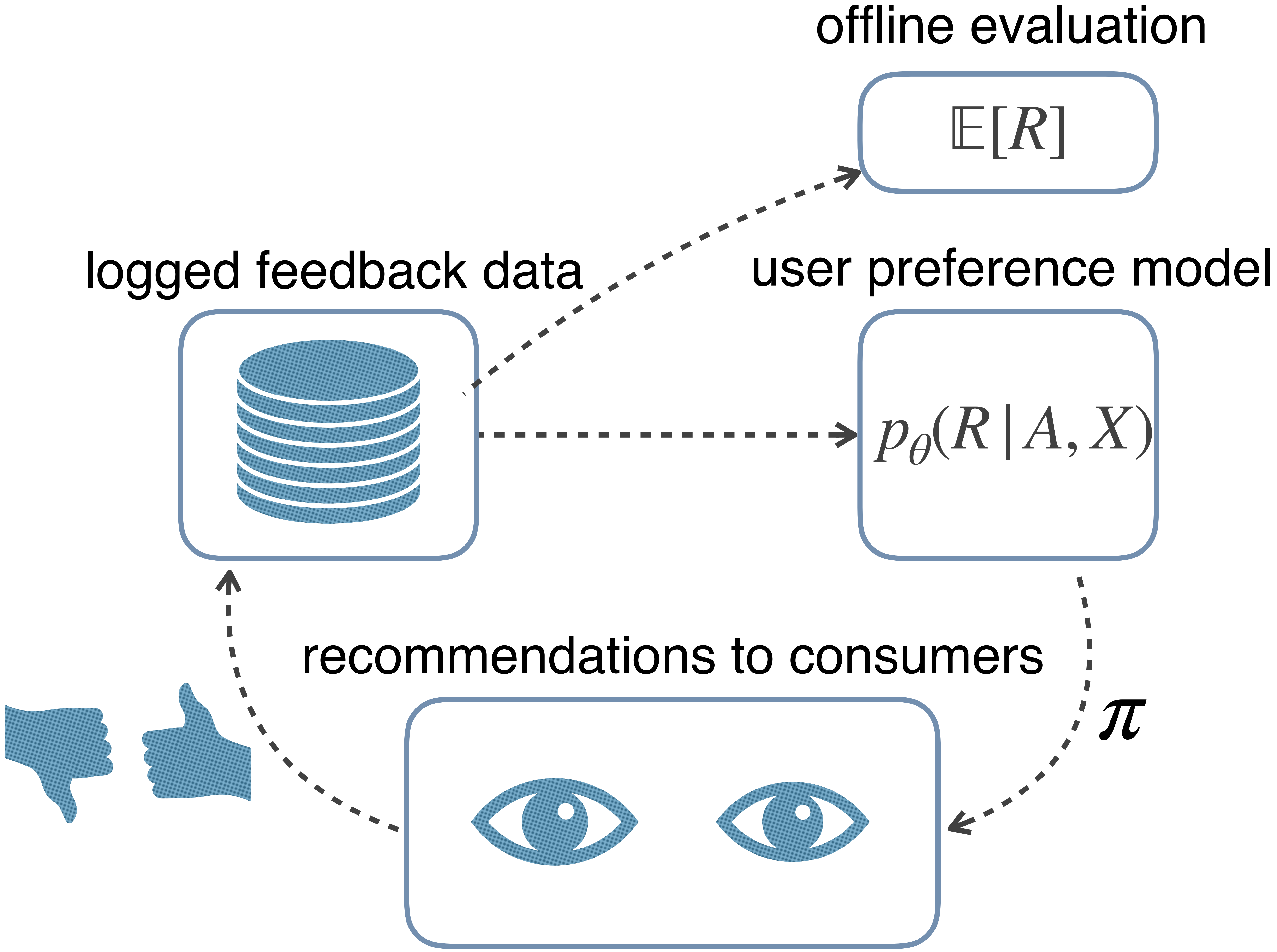


can also avoid subjecting users to bad ideas!

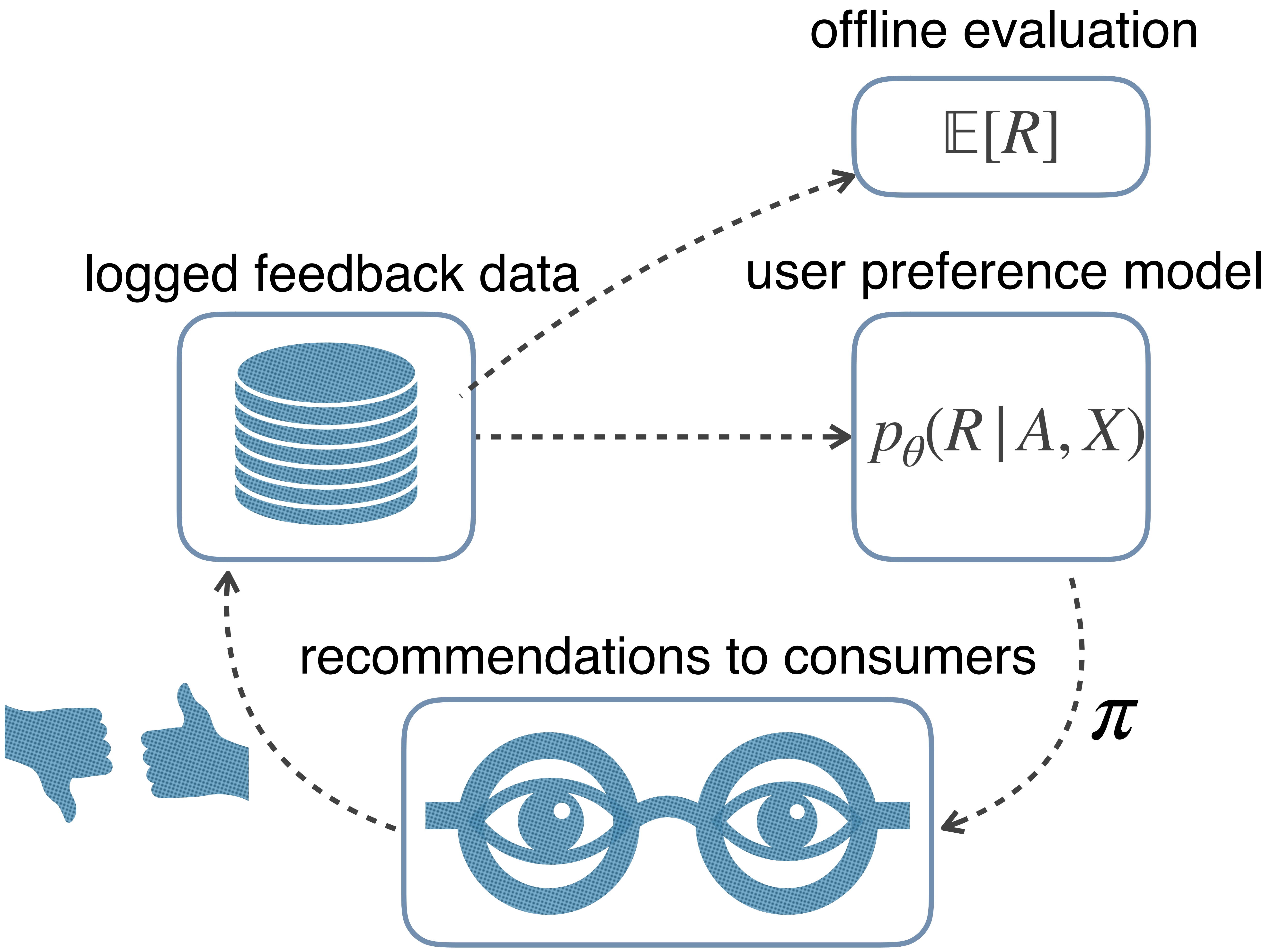
The Recommender Interaction Loop



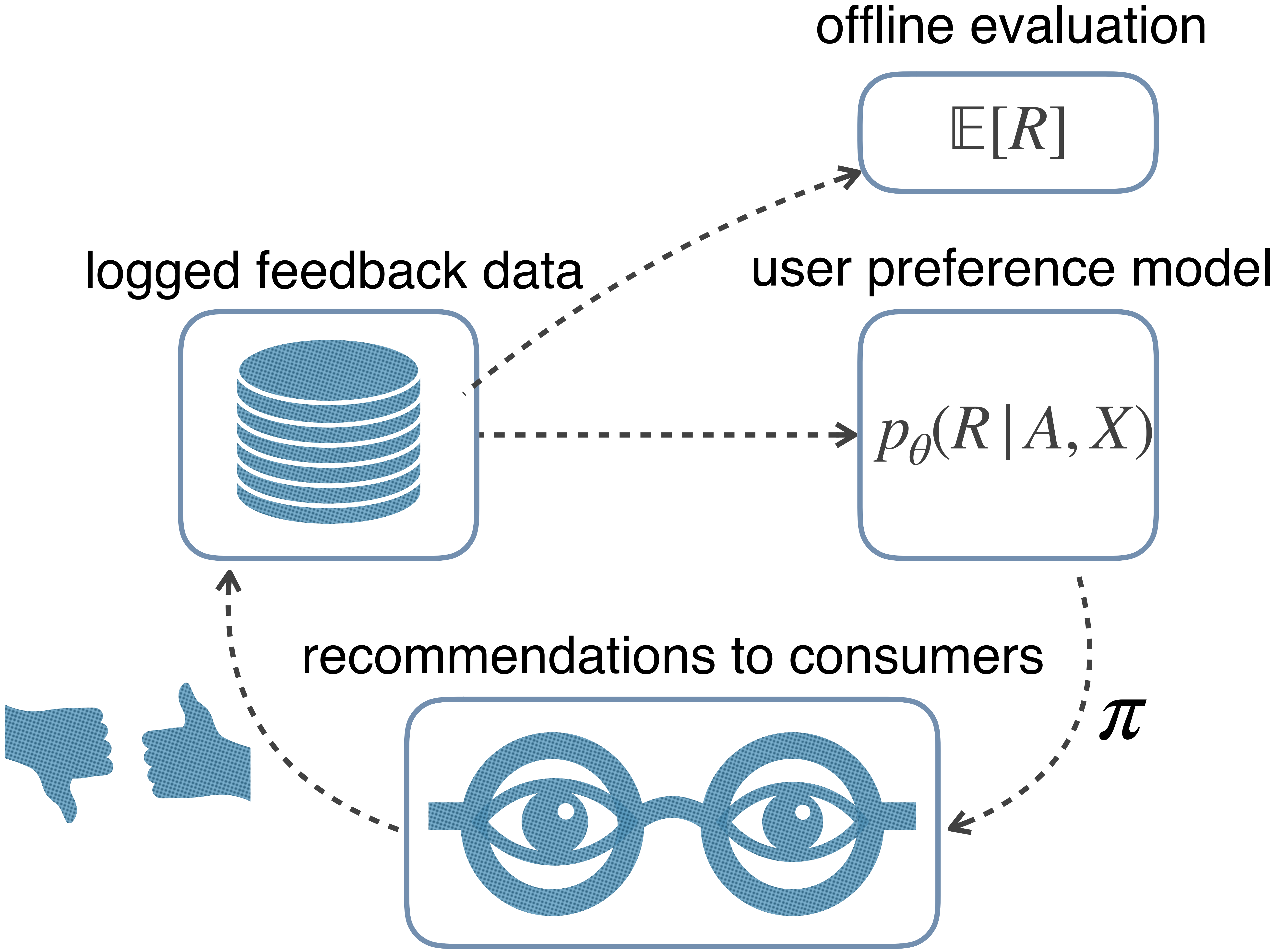
The Recommender Interaction Loop



The Recommender Interaction Loop



The Recommender Interaction Loop

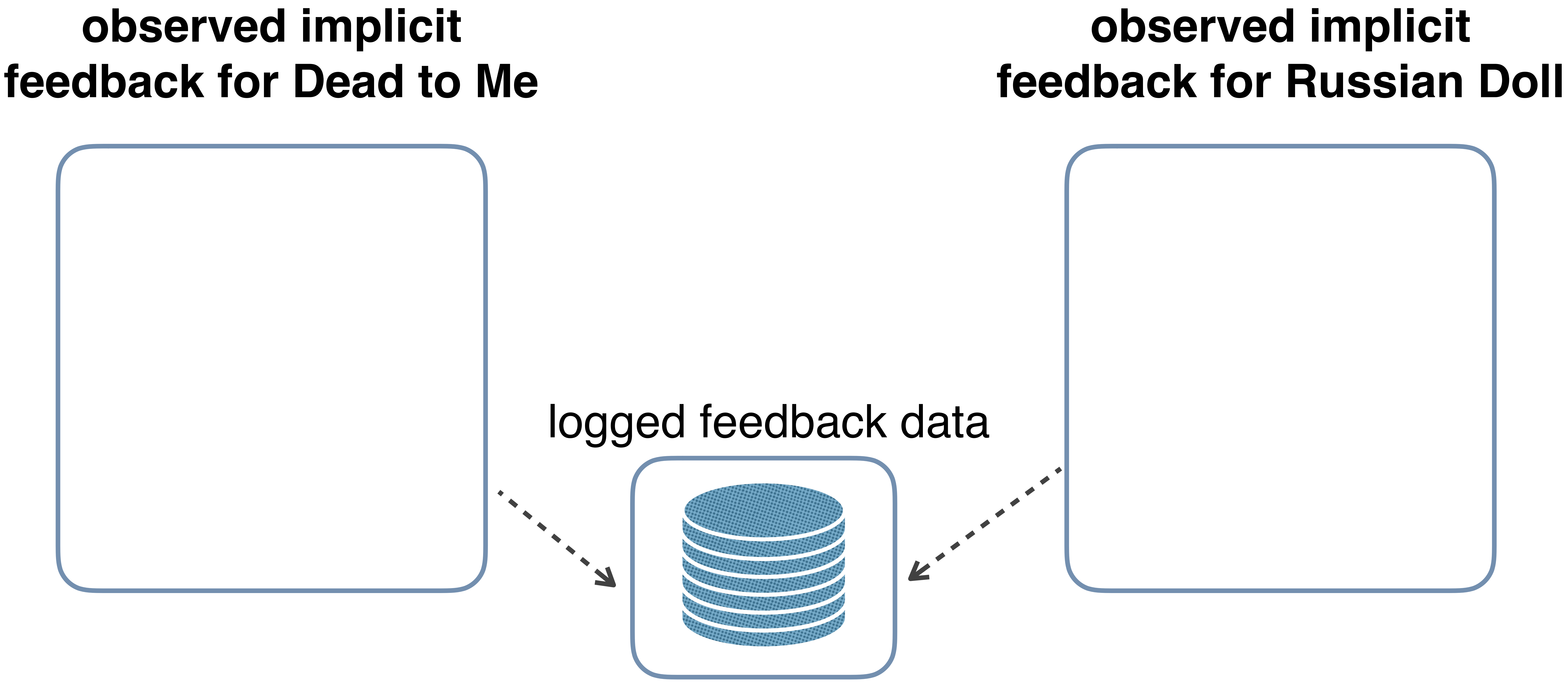


"How Algorithmic Confounding in Recommendation Systems Increases Homogeneity and Decreases Utility" (Chaney et al. 2017)

"Modeling User Exposure in Recommendation" (Liang et al. 2016)

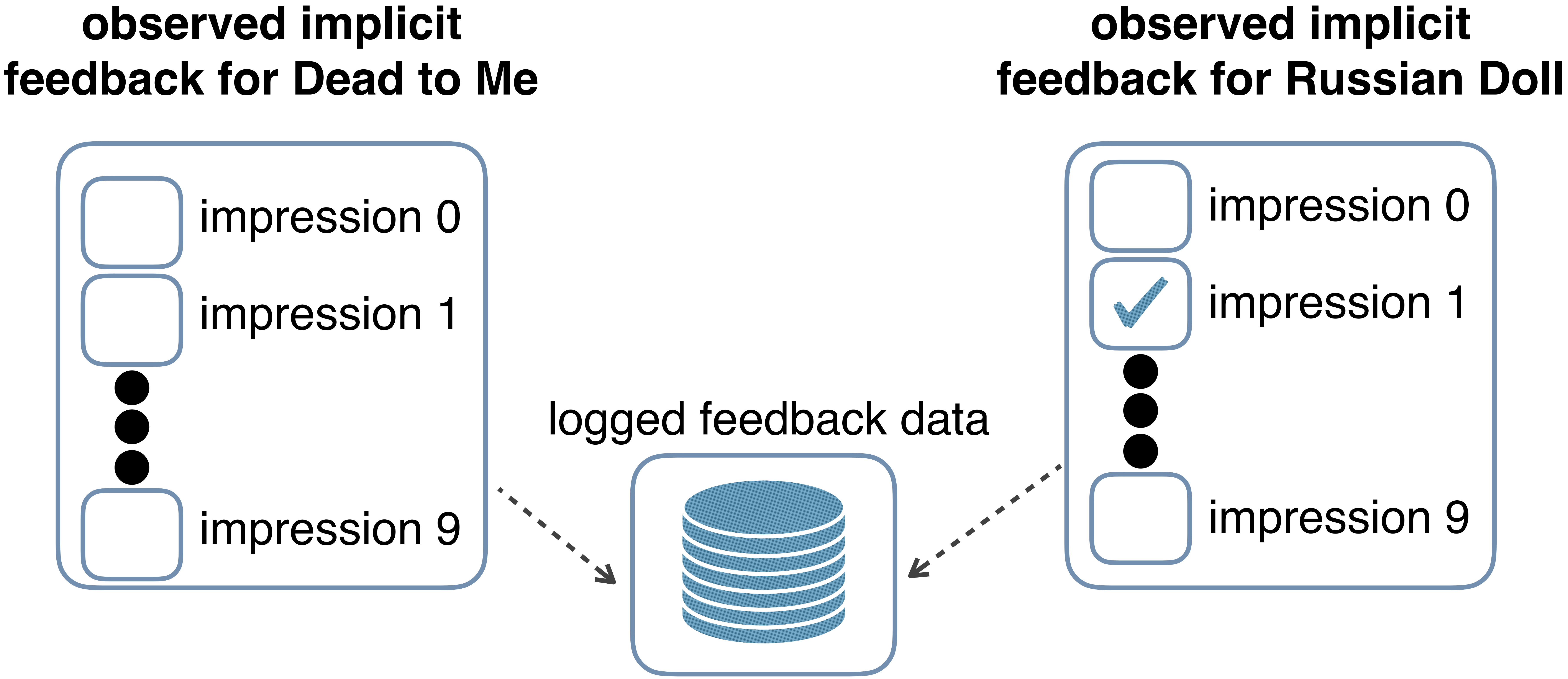
A Simple Example

- e.g. two items, A and B, with the same probability of reward = 0.1



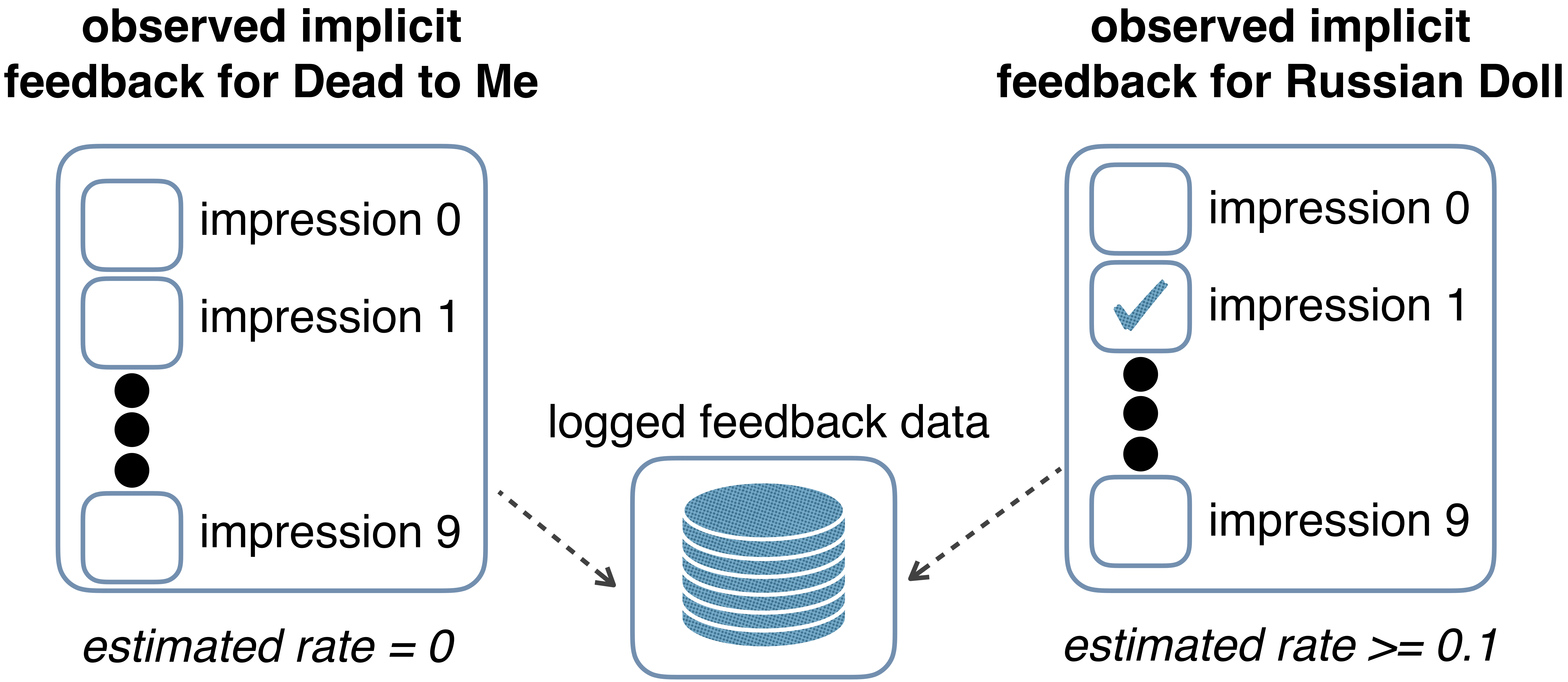
A Simple Example

- e.g. two items, A and B, with the same probability of reward = 0.1



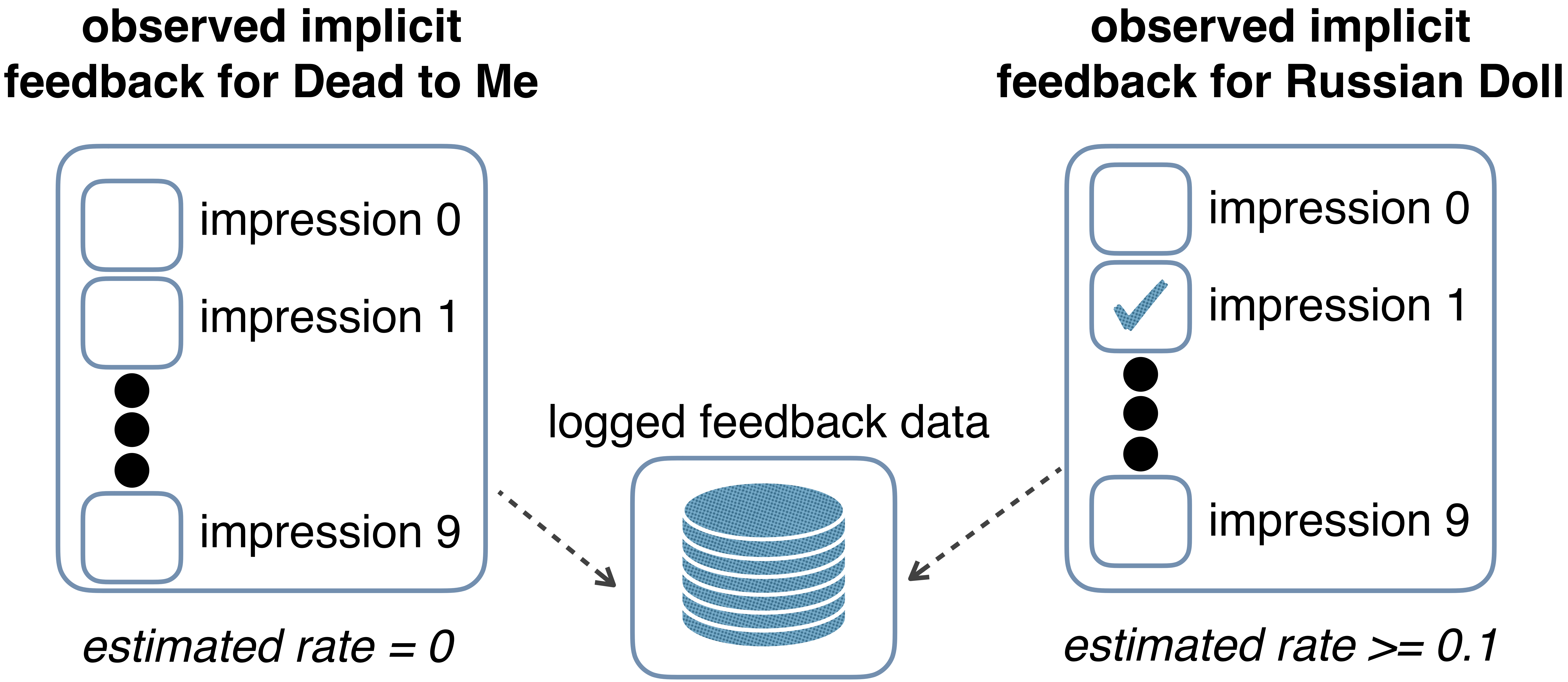
A Simple Example

- e.g. two items, A and B, with the same probability of reward = 0.1



A Simple Example

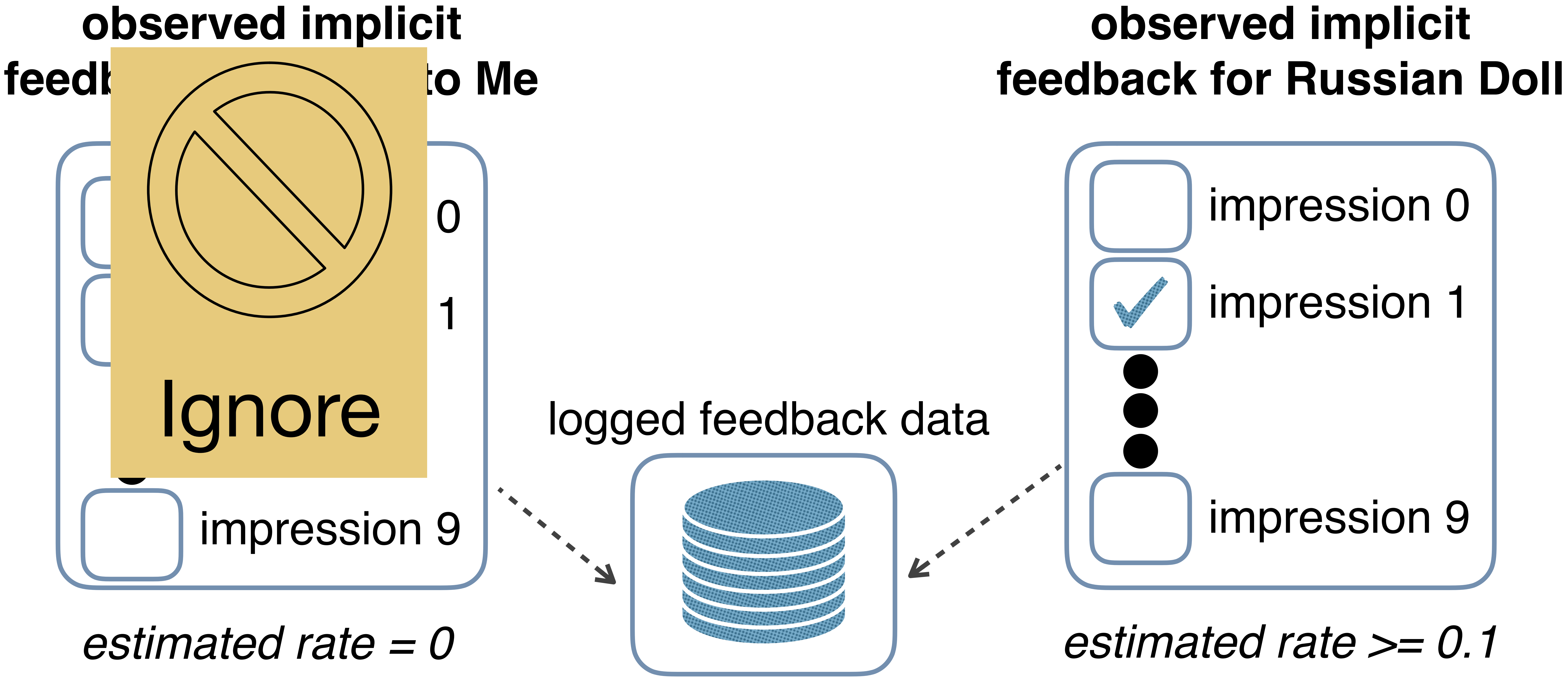
- e.g. two items, A and B, with the same probability of reward = 0.1



- the estimated relevance will be identical only 31.3% of the time

A Simple Example

- e.g. two items, A and B, with the same probability of reward = 0.1



- the estimated relevance will be identical only 31.3% of the time

Randomized Controlled Trials

Charles Sanders Peirce

"*At the beginning […] the pack was well shuffled, and, the operator and subject having taken their places, the operator was governed by the color of the successive cards in choosing whether he should first diminish the weight and then increase it, or vice versa.*"

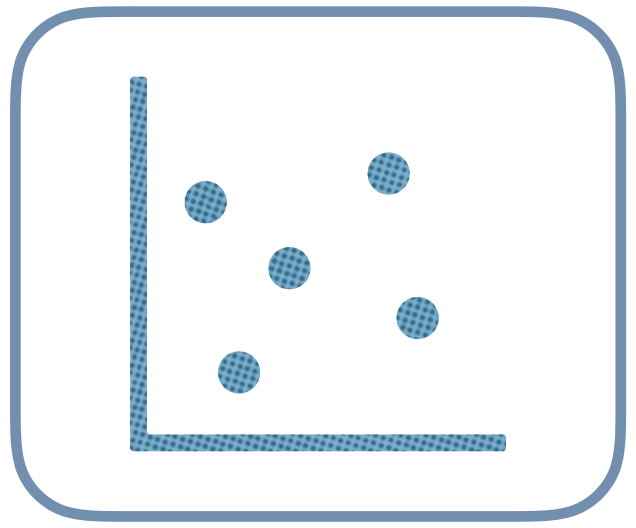
On Small Differences in Sensation,
C. S. Peirce & J. Jastrow (1885)

Randomized Controlled Trials

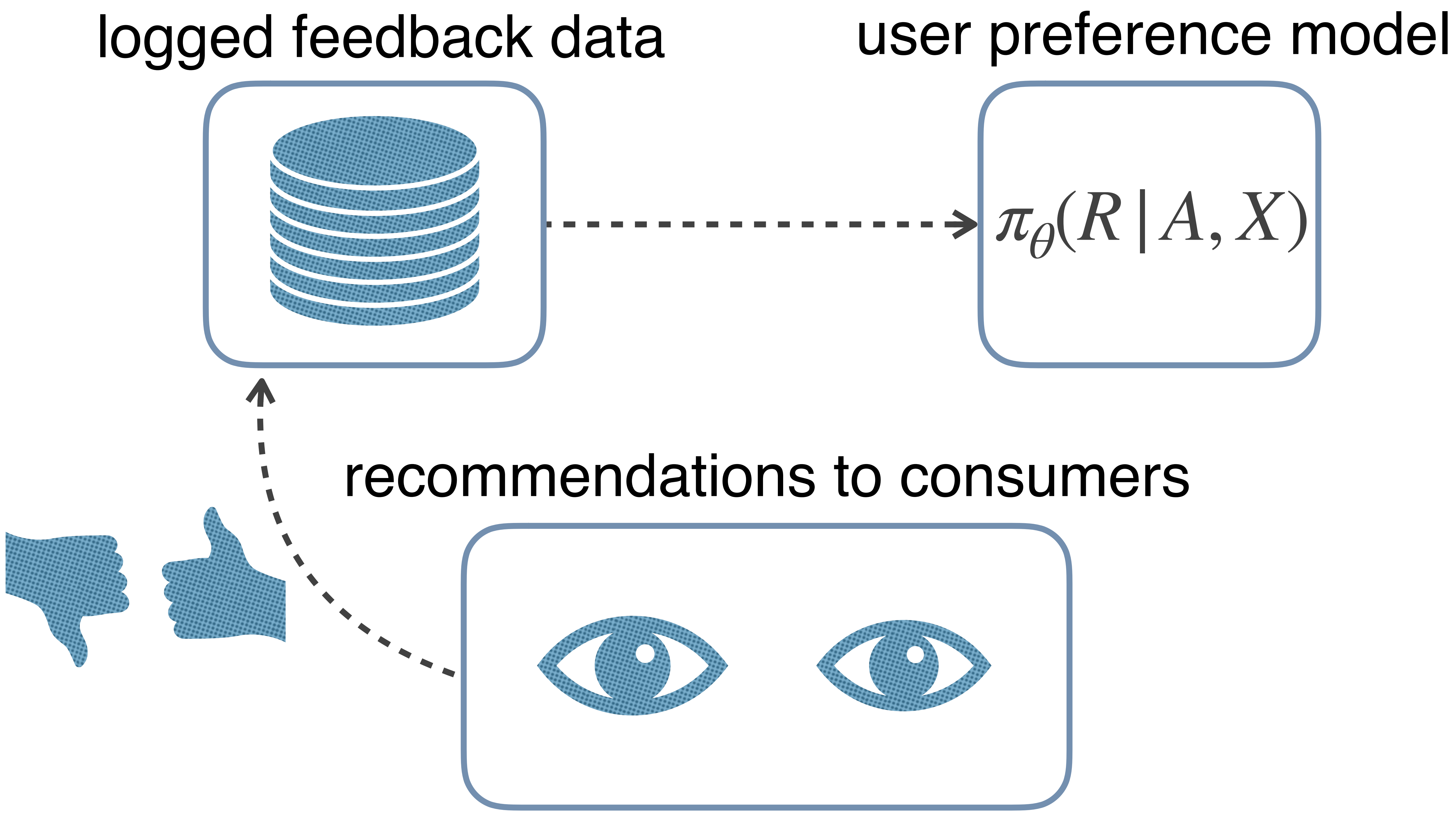
Charles Sanders Peirce

"At the beginning […] the pack was well shuffled, and, the operator and subject having taken their places, the operator was governed by the color of the successive cards in choosing whether he should first diminish the weight and then increase it, or vice versa."

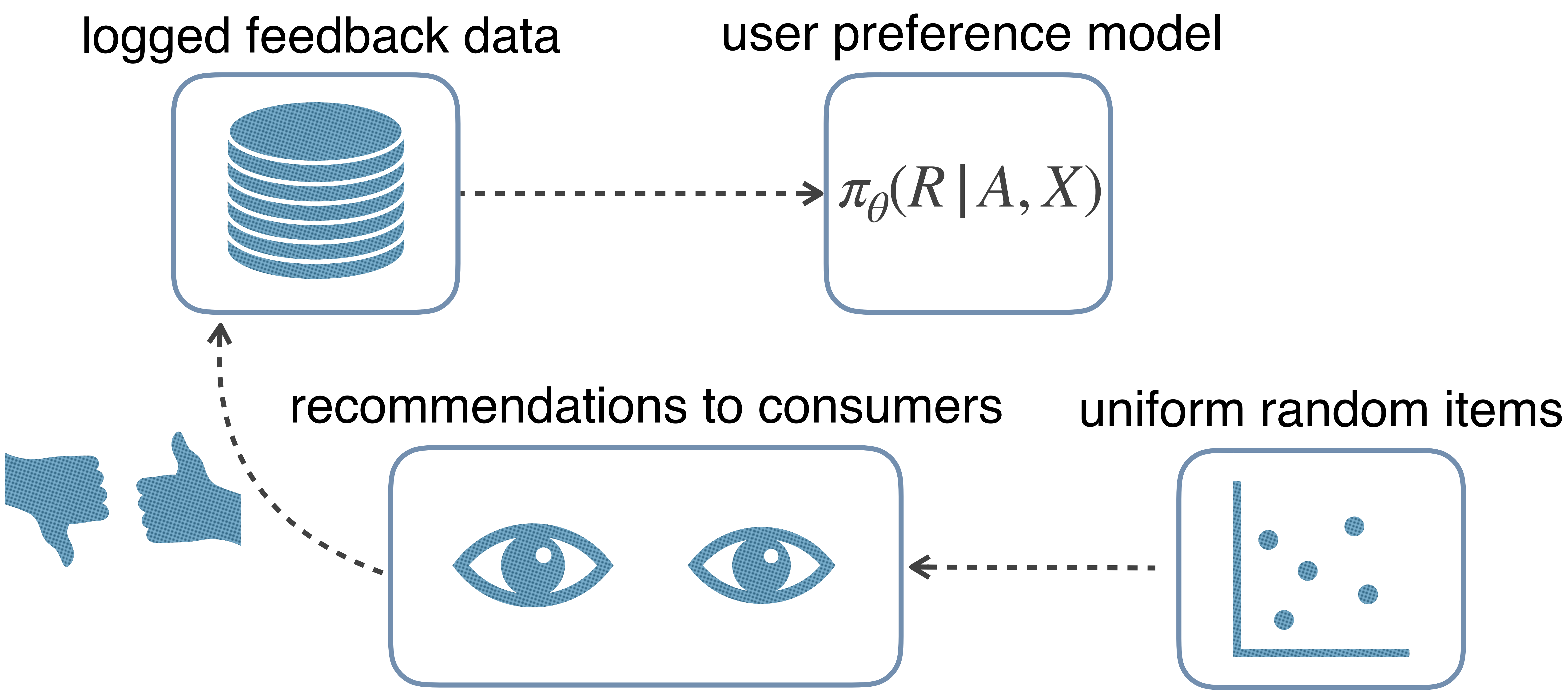
On Small Differences in Sensation,
C. S. Peirce & J. Jastrow (1885)

In recommendation:  uniform random items

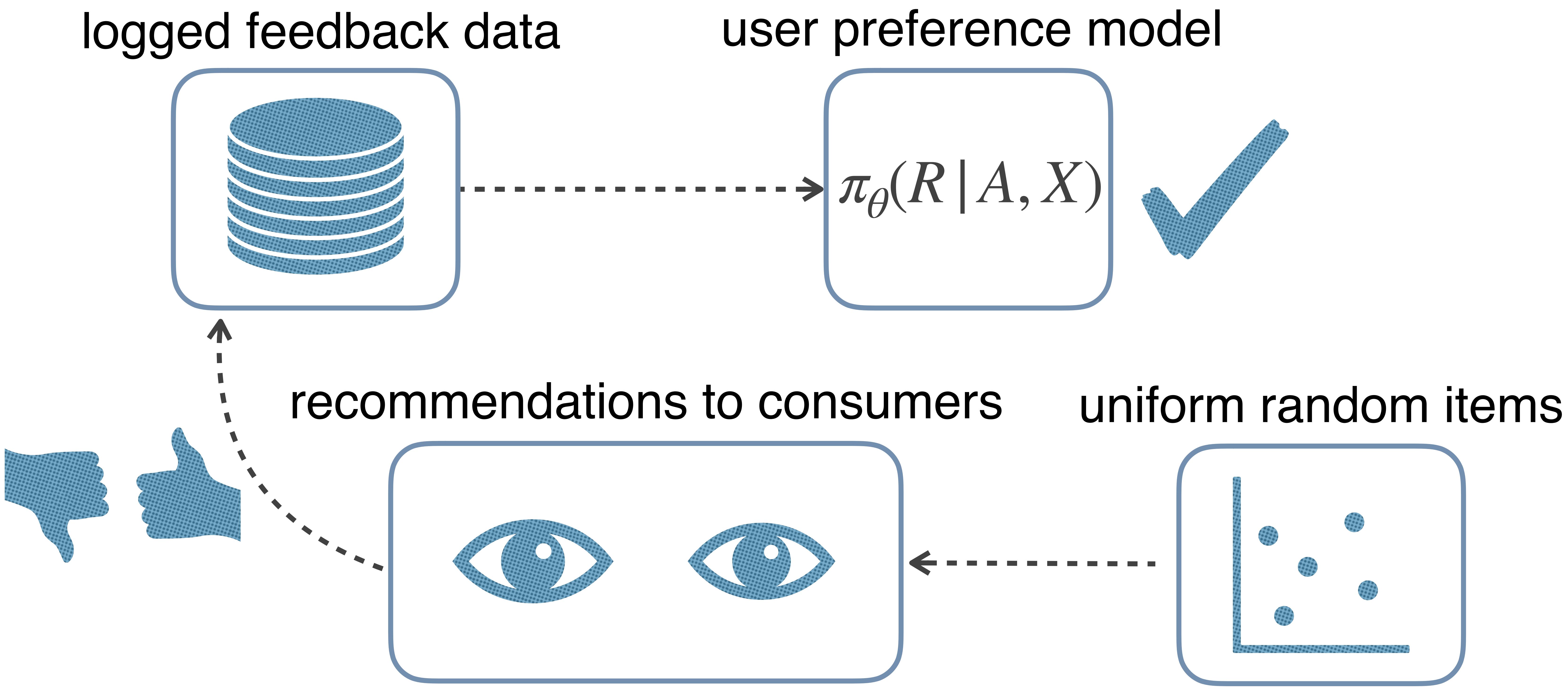
Randomized Controlled Trials



Randomized Controlled Trials



Randomized Controlled Trials



A Small Number of Producers Dominate Consumption in Culture

producer popularity

leads to leads to

exposure to new consumers

A Small Number of Producers Dominate Consumption in Culture

e.g. actors, musicians, authors

producer popularity

leads to

leads to

exposure to new consumers

A Small Number of Producers Dominate Consumption in Culture

e.g. actors, musicians, authors

producer popularity

leads to

leads to

exposure to new consumers

e.g. word of mouth, media coverage

A Small Number of Producers Dominate Consumption in Culture

e.g. actors, musicians, authors

producer popularity

leads to

leads to

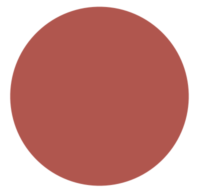
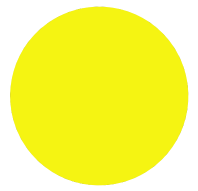
exposure to new consumers

e.g. word of mouth, media coverage

— Matthew effect / Pareto principle [Juran, 1937]

Simple Off-Policy Example

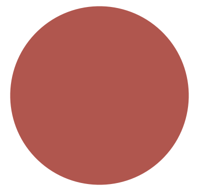
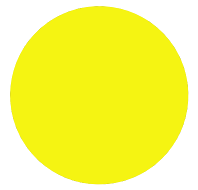
Simple Example

actions:  or 

reward:  or 

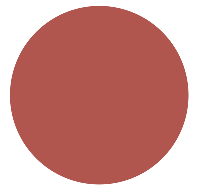
Simple Off-Policy Example

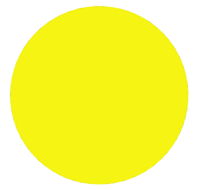
Simple Example

actions:  or 

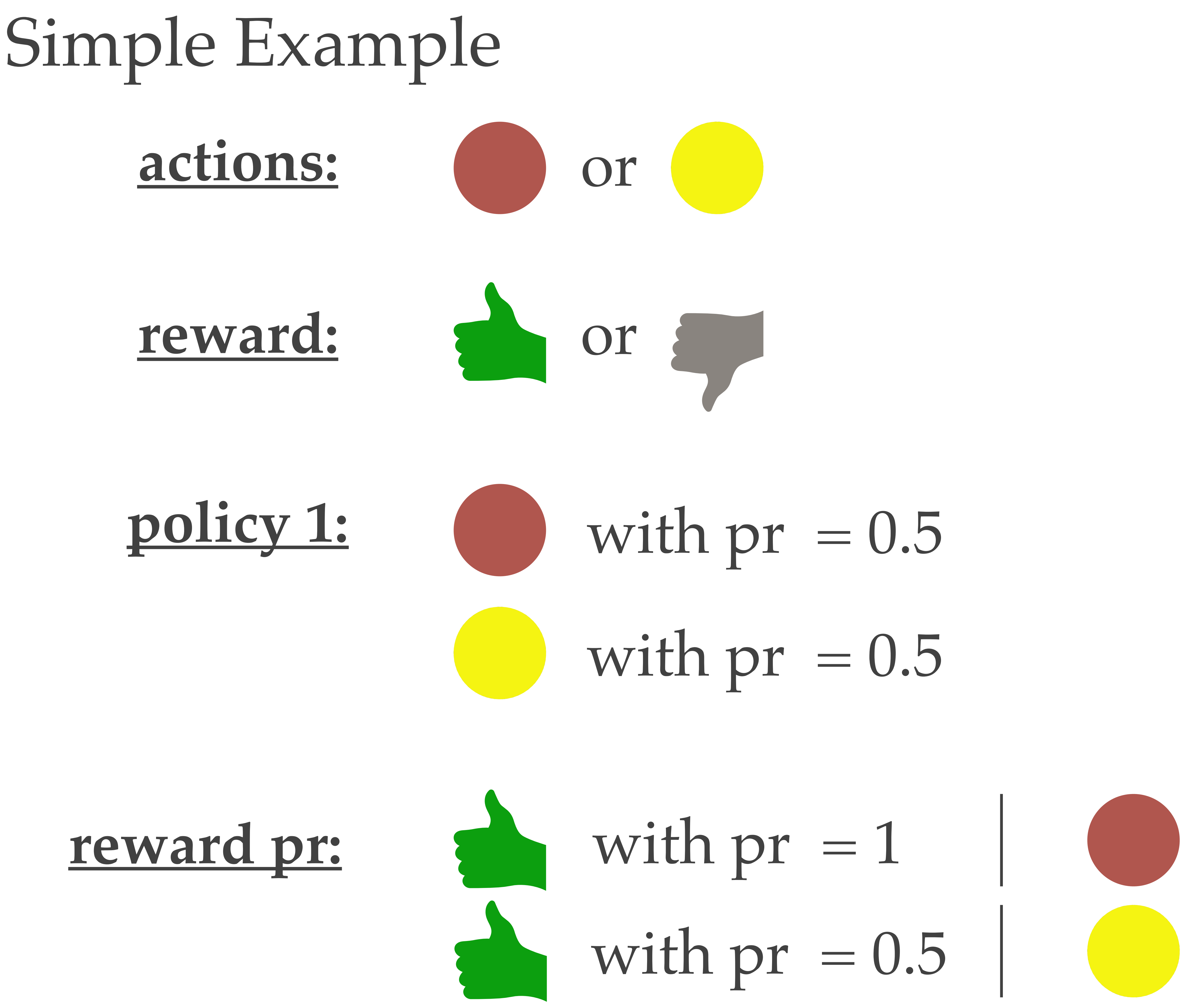
reward:  or 

policy 1:

 with pr $= 0.5$

 with pr $= 0.5$

Simple Off-Policy Example



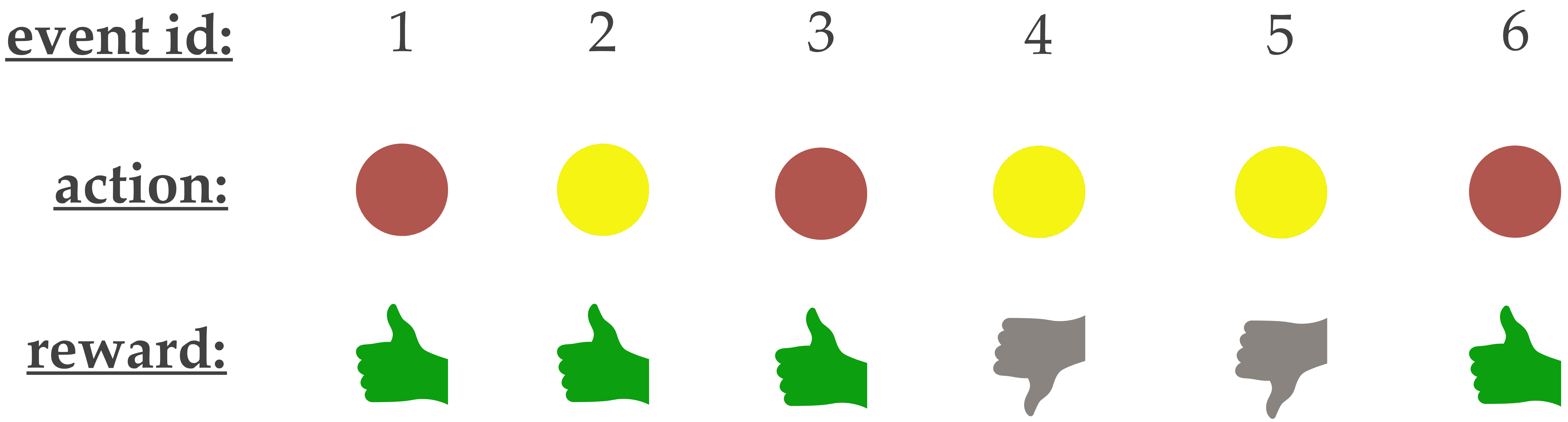
Simple Off-Policy Example

collect data with policy 1 π

event id:	1	2	3	4	5	6

Simple Off-Policy Example

collect data with policy 1 π



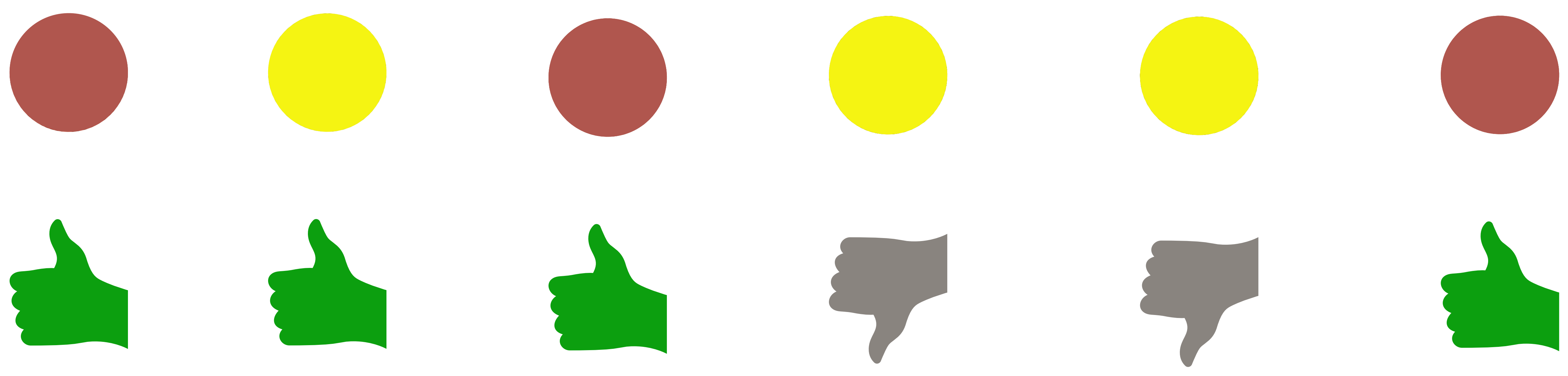
Simple Off-Policy Example

collect data with policy 1 π

event id: 1 2 3 4 5 6

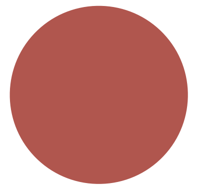
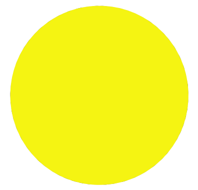
action:

reward:

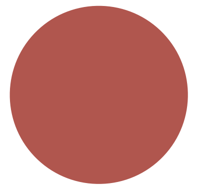
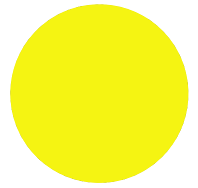


evaluate average reward for policy 1 $\bar{r} = \frac{1}{6} \sum_{n=1}^{6} r_n = \frac{2}{3}$

Simple Off-Policy Example

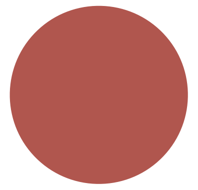
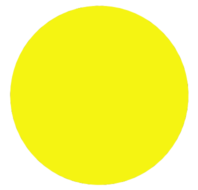
evaluate policy 2:  with pr $= 1$  with pr $= 0$ h

Simple Off-Policy Example

evaluate policy 2:  with pr $= 1$  with pr $= 0$ $\quad h$

step 1: *collect data with policy 1* $\quad \pi$

Simple Off-Policy Example

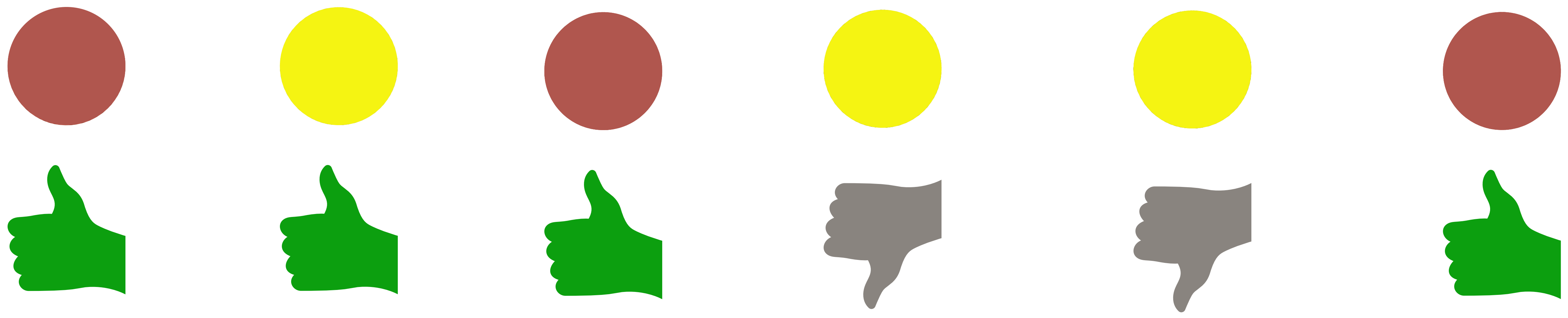
evaluate policy 2:  with pr $= 1$  with pr $= 0$ h

step 1: *collect data with policy 1* π

event id: 1 2 3 4 5 6

action:

reward:



Simple Off-Policy Example

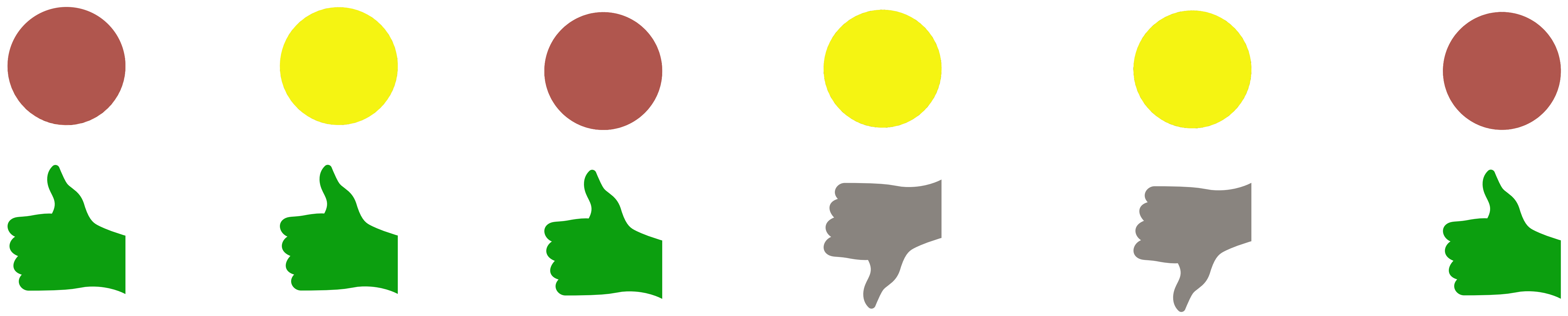
evaluate policy 2: (red) with pr $= 1$ (yellow) with pr $= 0$ h

step 1: *collect data with policy 1* π

event id: 1 2 3 4 5 6

action:

reward:

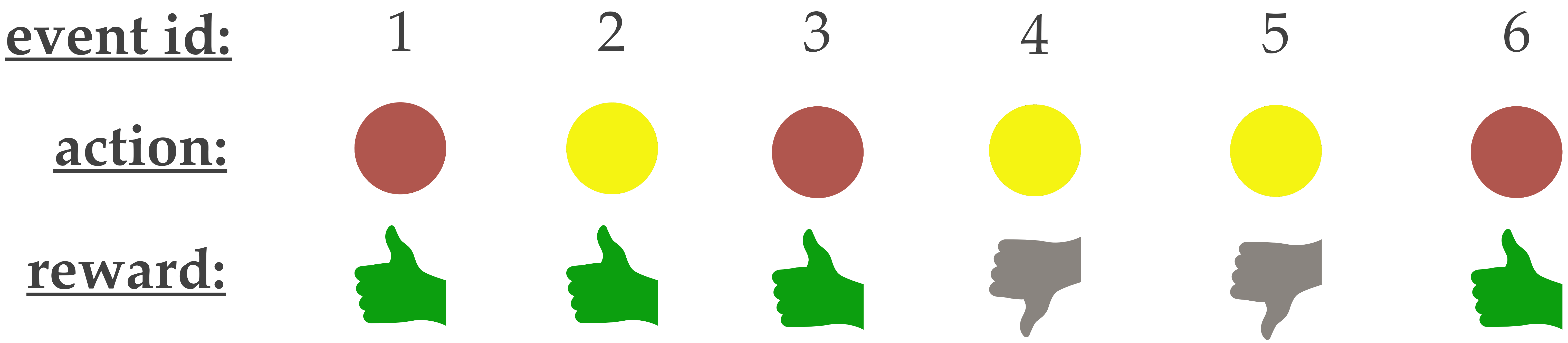


step 2: *"hallucinate" what policy 2 would do on the same data*

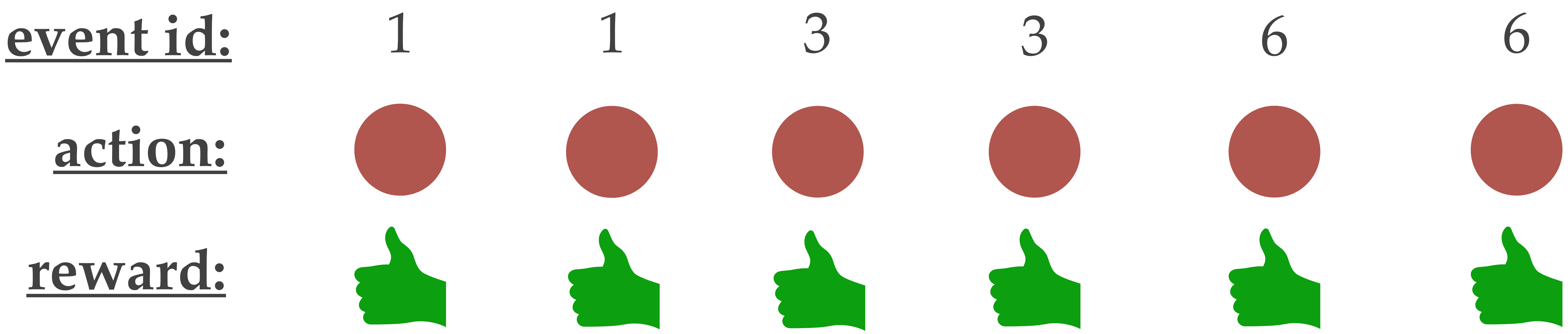
Simple Off-Policy Example

evaluate policy 2: with pr $= 1$ with pr $= 0$ h

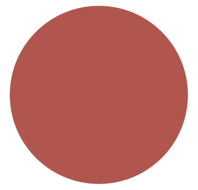
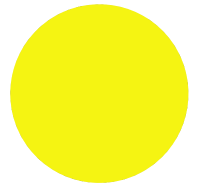
step 1: *collect data with policy 1* π



step 2: *"hallucinate" what policy 2 would do on the same data*



Simple Off-Policy Example

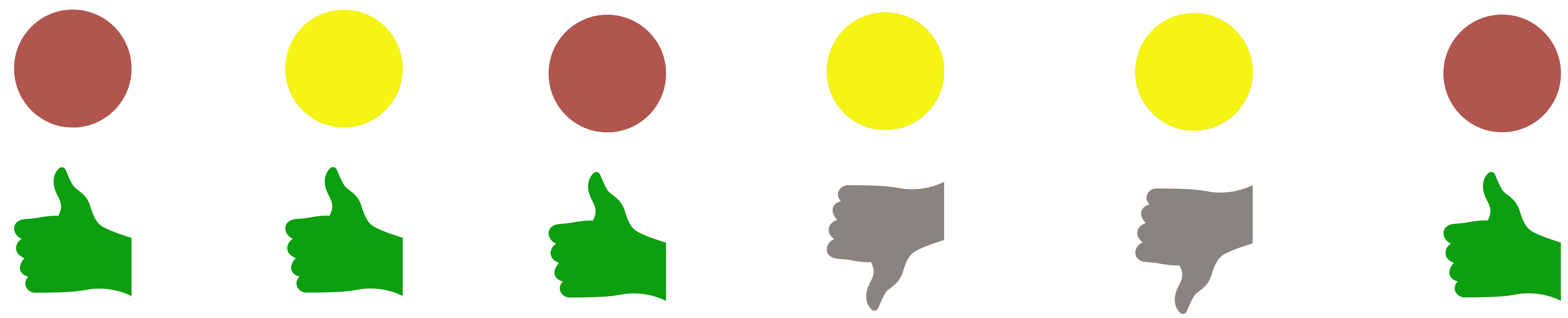
evaluate policy 2:  with pr $= 1$  with pr $= 0$ h

step 1: *collect data with policy 1* π

event id: 1 2 3 4 5 6

action:

reward:

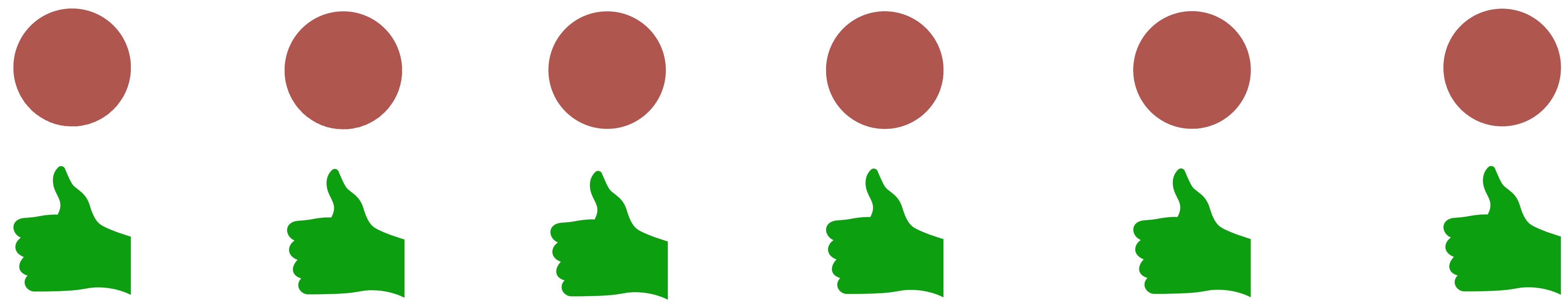


step 2: *"hallucinate" what policy 2 would do on the same data*

event id: 1 1 3 3 6 6

action:

reward:



step 3: *evaluate* $\bar{r} = 1$

Inverse Propensity Scoring for Off-Policy Evaluation

How many times should we hallucinate each event?

$$\frac{1}{N}\sum_{n=1}^{N} w_n r_n \approx \mathbb{E}_h[R]$$ what should w_n be?

Inverse Propensity Scoring for Off-Policy Evaluation

How many times should we hallucinate each event?

$$\frac{1}{N}\sum_{n=1}^{N} w_n r_n \approx \mathbb{E}_h[R]$$ what should w_n be?

Use importance sample reweighting:

$$w_n = \frac{h(a_n)}{\pi(a_n)}$$

Inverse Propensity Scoring for Off-Policy Evaluation

How many times should we hallucinate each event?

$$\frac{1}{N}\sum_{n=1}^{N} w_n r_n \approx \mathbb{E}_h[R]$$ what should w_n be?

Use importance sample reweighting:

$$w_n = \frac{h(a_n)}{\pi(a_n)}$$

Technique is called inverse propensity scoring (IPS) and $\pi(a_n)$ is the propensity score for action a_n .

Off-Policy Learning with IPS

$$\checkmark = \mathbb{E}_{X,A\sim\mathrm{Uniform}(\mathscr{A}),R}[\log p_\theta(R \mid A, X)]$$

random item recommended

set of all items

model parameters

context

Off-Policy Learning with IPS

"choose a model and train it on data how you like"

$$\checkmark = \mathbb{E}_{X,A\sim\text{Uniform}(\mathscr{A}),R}[\log p_\theta(R \mid A, X)]$$

random item recommended

set of all items

model parameters

context

Off-Policy Learning with IPS

"train on the right data"

$$\checkmark = \mathbb{E}_{X,A\sim \mathrm{Uniform}(\mathscr{A}),R}[\log p_\theta(R \mid A, X)]$$

random item recommended

set of all items

model parameters

context

Off-Policy Learning with IPS

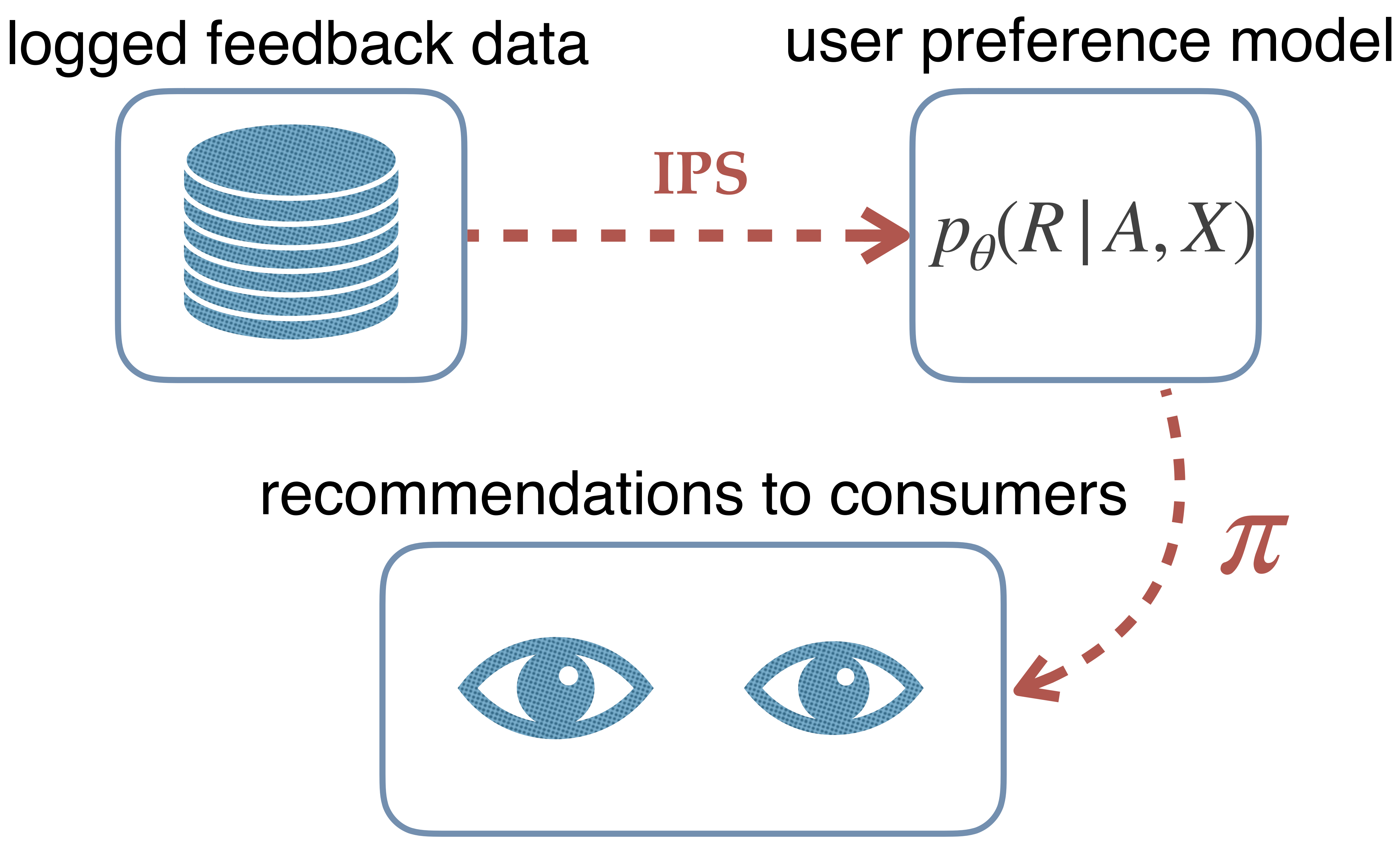
$$\checkmark = \mathbb{E}_{X,A\sim\mathrm{Uniform}(\mathscr{A}),R}[\log p_\theta(R \mid A, X)]$$

Off-Policy Learning with IPS

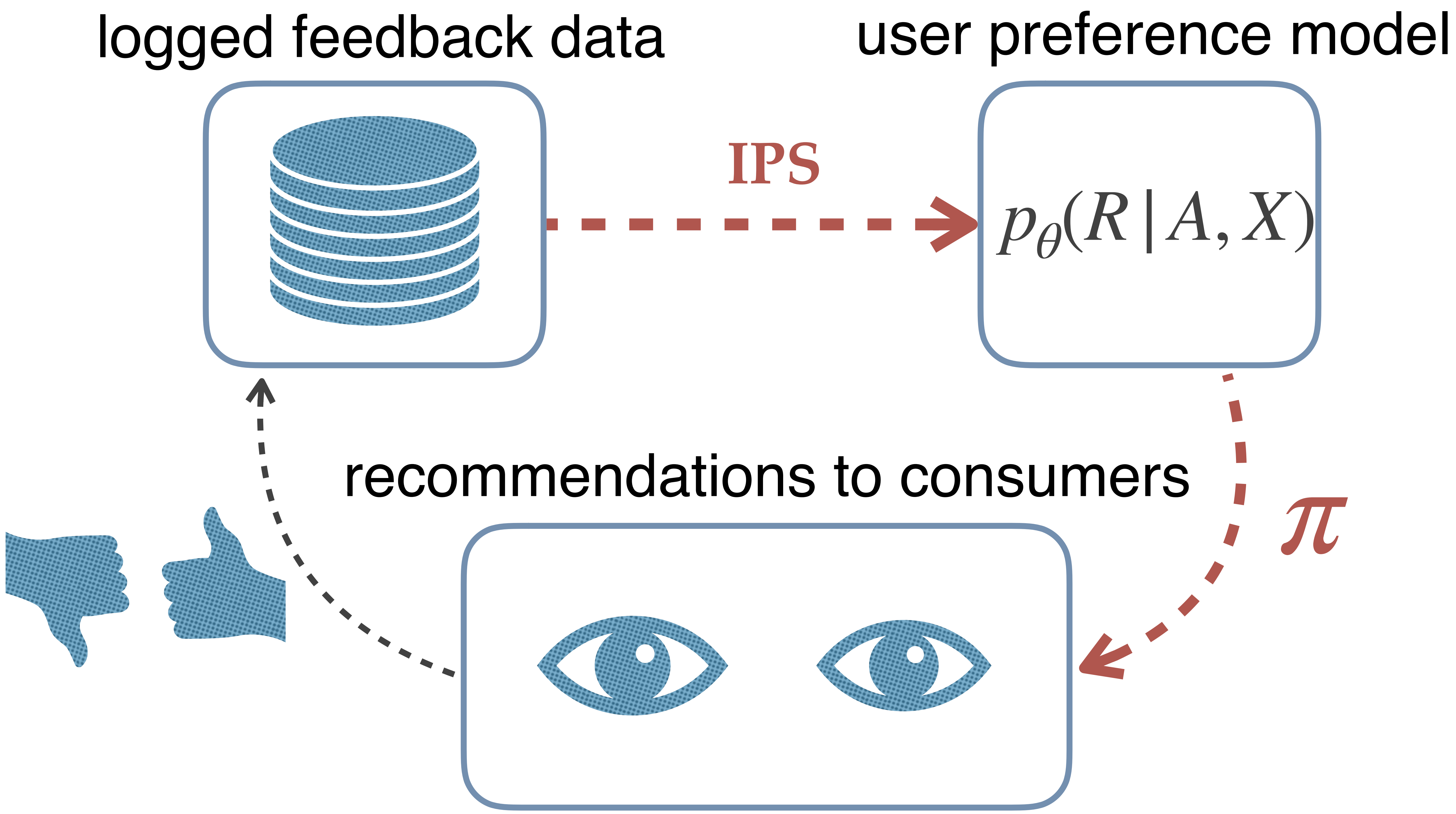
$$
\begin{aligned}
\checkmark &= \mathbb{E}_{X,A\sim\mathrm{Uniform}(\mathscr{A}),R}[\log p_\theta(R\,|\,A,X)] \\
&\approx \frac{1}{N}\sum_{n=1}^{N}\frac{\log p_\theta(r_n\,|\,a_n,x_n)}{|\mathscr{A}|\,\pi(a_n\,|\,x_n)} \\
&\quad \text{for } x_n, a_n, r_n \text{ collected with } \pi
\end{aligned}
$$

- enables counterfactual evaluation and model training, usually used with variance reduction techniques [Joachims & Swaminathan, 2016]

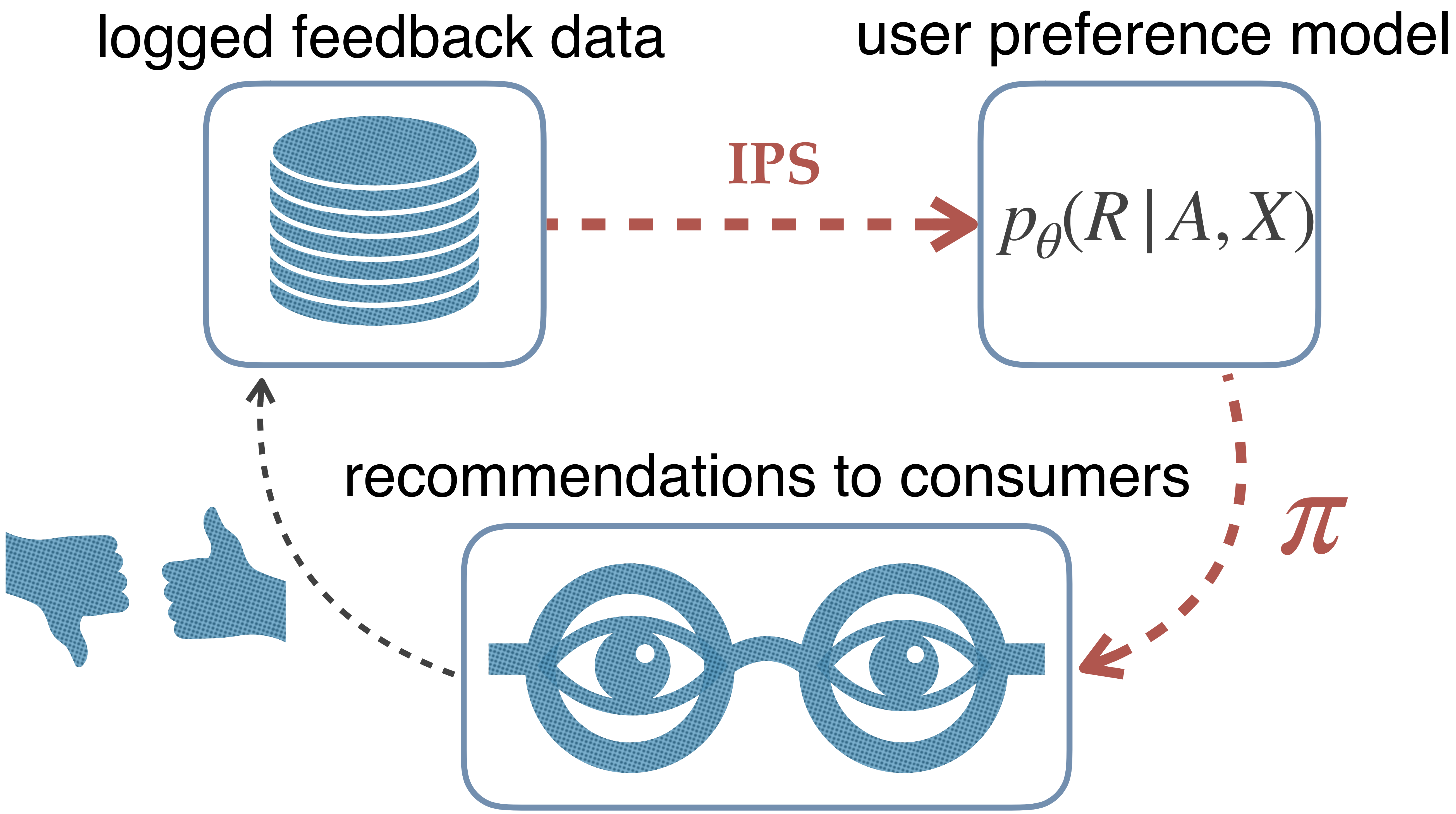
Modified Recommendation Pipeline



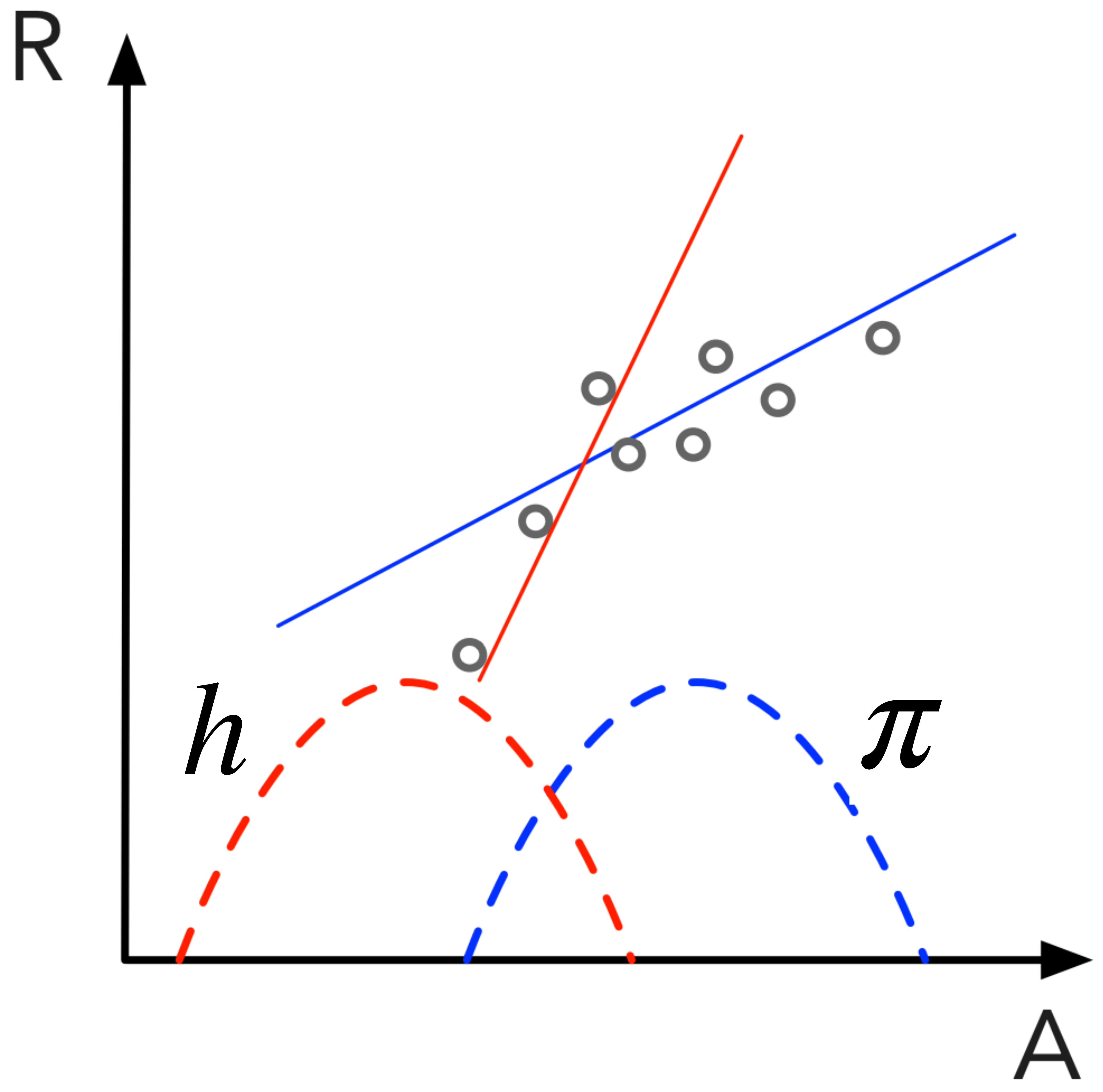
Modified Recommendation Pipeline



Modified Recommendation Pipeline



When Does IPS Help?



Bias-Variance Tradeoff

online value
of policy

offline
estimate

$$\mathbb{E}[(\bar{r} - \hat{r})^2] =$$

Bias-Variance Tradeoff

online value of policy

offline estimate

$$\underbrace{\mathbb{E}[(\bar{r} - \hat{r})^2]}_{\text{mean squared error}} =$$

Bias-Variance Tradeoff

$$\mathbb{E}[(\underbrace{\overset{\text{online value of policy}}{\bar{r}} - \overset{\text{offline estimate}}{\hat{r}})^2]}_{\text{mean squared error}} =$$

— Offline evaluation approaches vary in the way they trade off bias and variance.

Bias-Variance Tradeoff

online value of policy

offline estimate

$$\underbrace{\mathbb{E}[(\bar{r} - \hat{r})^2]}_{\text{mean squared error}} = (\mathbb{E}[\hat{r}] - \bar{r})^2 + \mathbb{E}[\hat{r}^2] - \mathbb{E}[\hat{r}]^2$$

— Offline evaluation approaches vary in the way they trade off bias and variance.

Bias-Variance Tradeoff

online value of policy

offline estimate

$$\underbrace{\mathbb{E}[(\bar{r} - \hat{r})^2]}_{\text{mean squared error}} = \underbrace{(\mathbb{E}[\hat{r}] - \bar{r})^2}_{\text{bias^2}} + \underbrace{\mathbb{E}[\hat{r}^2] - \mathbb{E}[\hat{r}]^2}_{\text{variance}}$$

— Offline evaluation approaches vary in the way they trade off bias and variance.

When Does IPS Work?

IPS requires:

- *absolute continuity*
  - i.e. $\pi(a_n \mid x_n) > 0$ wherever $h(a_n \mid x_n) > 0$
- independent actions (conditional on context)
- independent rewards (conditional on actions, context)

IPS has high variance with extreme weights:

- extreme propensities (e.g. large action space)
- large divergence between h and π

Reducing Variance

— Several methods to reduce IPS variance:
- cap the weights [Bottou et al. 2013]
- normalize the weights [Swaminathan & Joachims, 2015]
- doubly robust method [Dudik et al., 2011]

— [Gilotte, 2018] has a good review of methods.

Direct Method

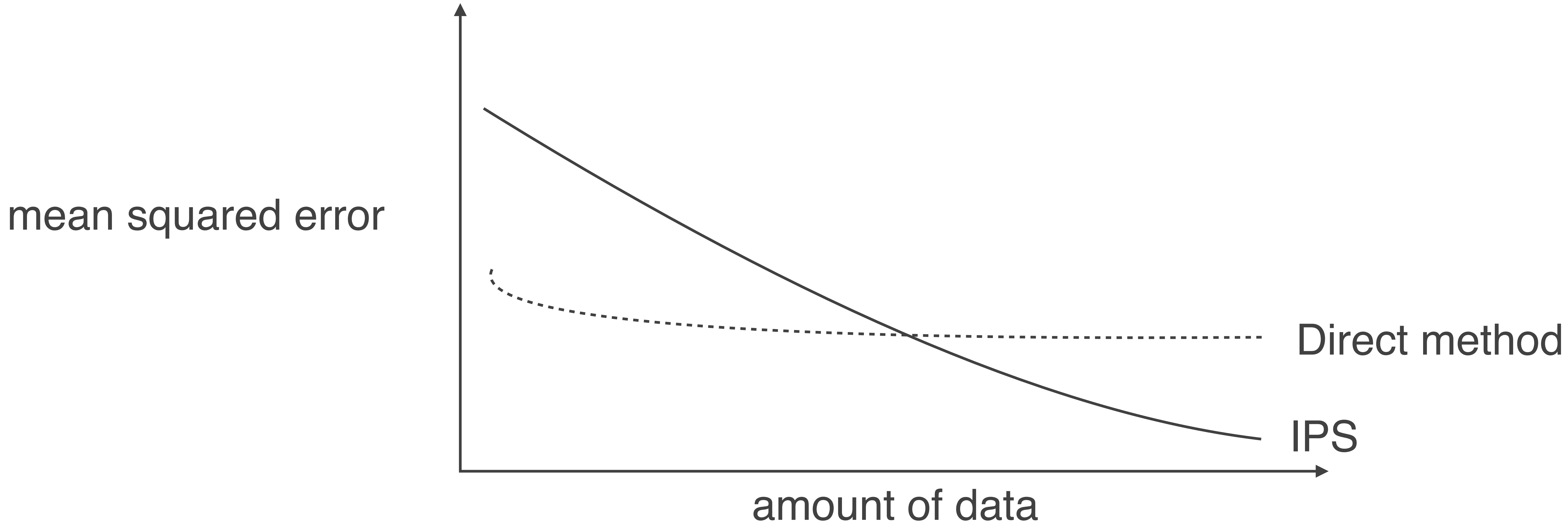
$$\bar{r}(h) = \frac{1}{N} \sum_{n=1}^{N} \sum_{A} h(A \mid x_n) \mathbb{E}[R \mid A, x_n]$$

— Can use the “direct method” to reduce variance.

— Introduces bias from the model assumptions.

— For each task, direct method requires positing a model, fitting parameters (hyperparameters), criticizing fits.

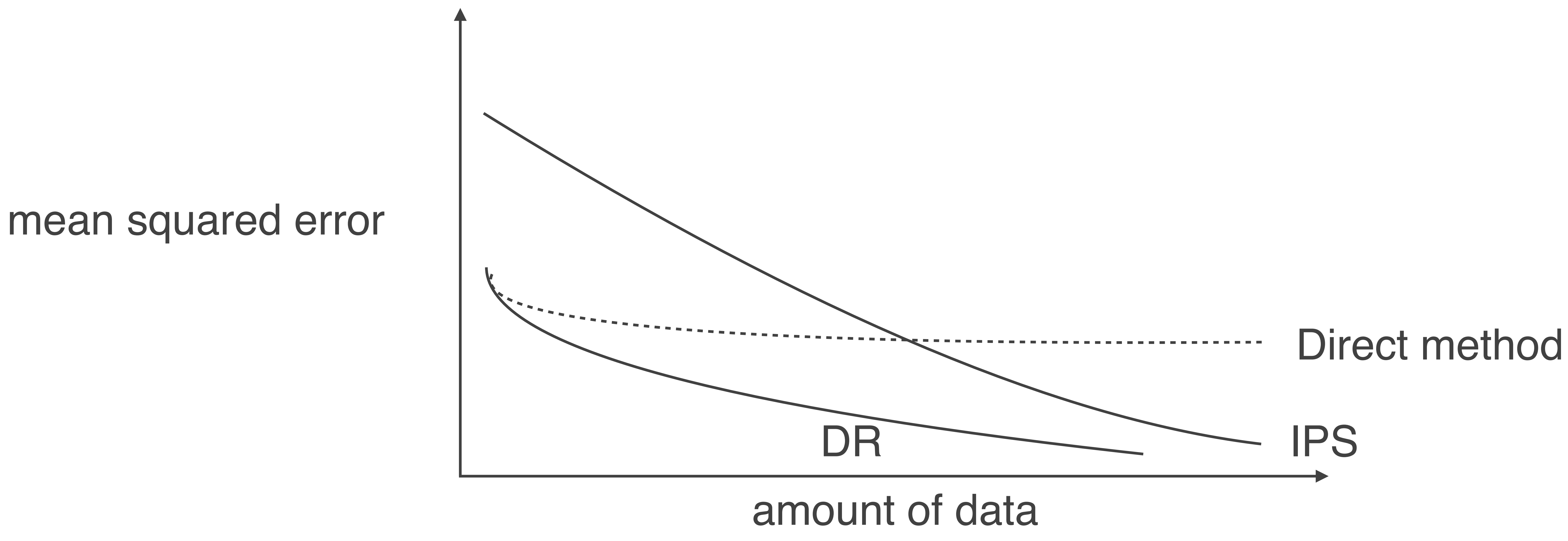
Bias-Variance Tradeoff



IPS: unbiased, high variance

Direct method: biased, low variance

Bias-Variance Tradeoff



IPS: unbiased, high variance

Direct method: biased, low variance

Doubly robust: unbiased, lower variance

Doubly Robust

Combine direct method with IPS:

Doubly Robust

Combine direct method with IPS:

$$\bar{r}(h) = \frac{1}{N}\sum_{n=1}^{N}\frac{h(a_n|x_n)}{\pi(a_n|x_n)}(r_{n,k} - \mathbb{E}[R \mid a_n, x_n]) + \mathbb{E}_h[\mathbb{E}[R \mid A, x_n]]$$

Doubly Robust

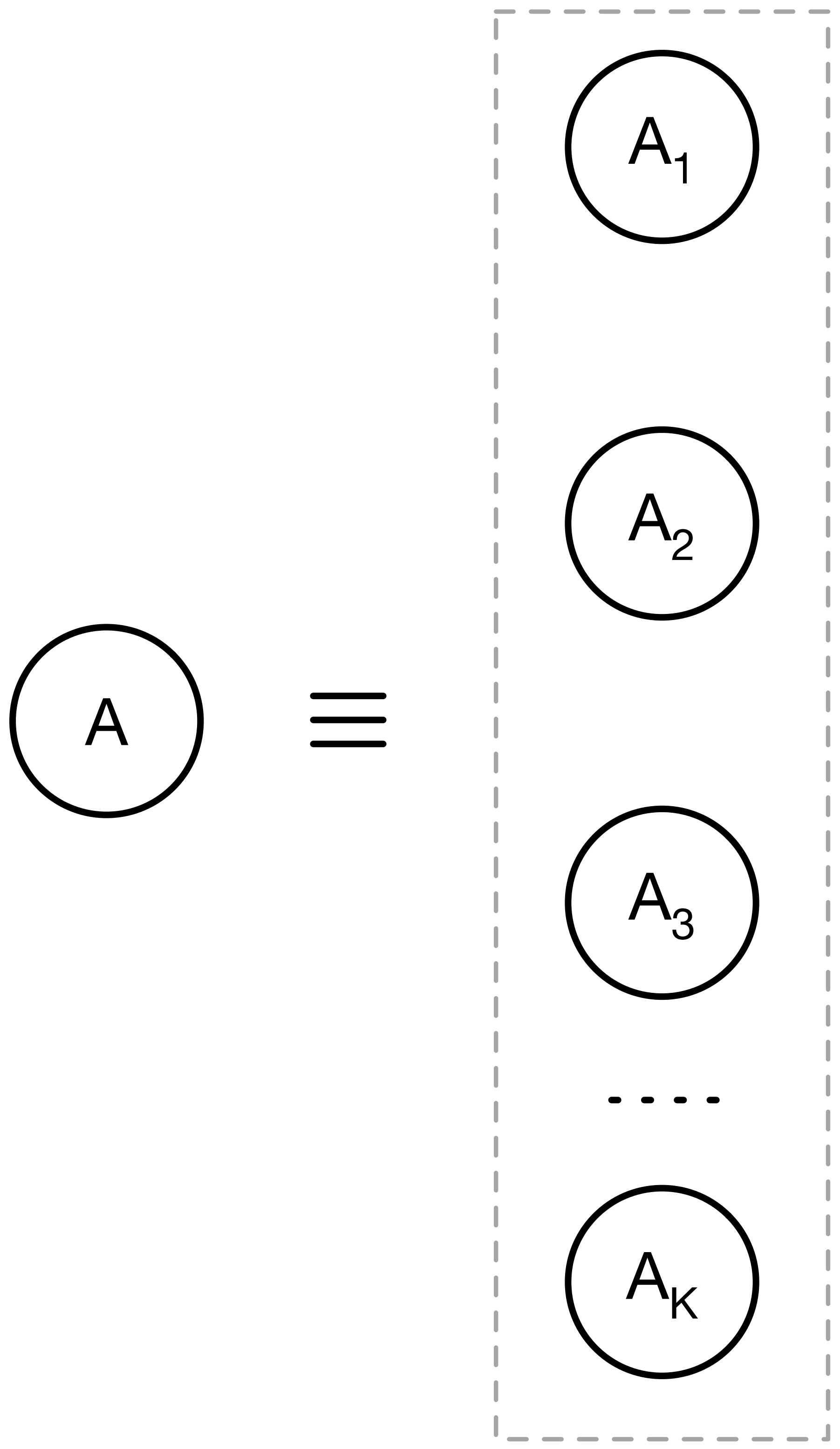
Combine direct method with IPS:

$$\bar{r}(h) = \frac{1}{N} \sum_{n=1}^{N} \frac{h(a_n | x_n)}{\pi(a_n | x_n)} (r_{n,k} - \mathbb{E}[R \mid a_n, x_n]) + \mathbb{E}_h[\mathbb{E}[R \mid A, x_n]]$$

Lower variance than IPS if predicted reward correlated with actual reward.

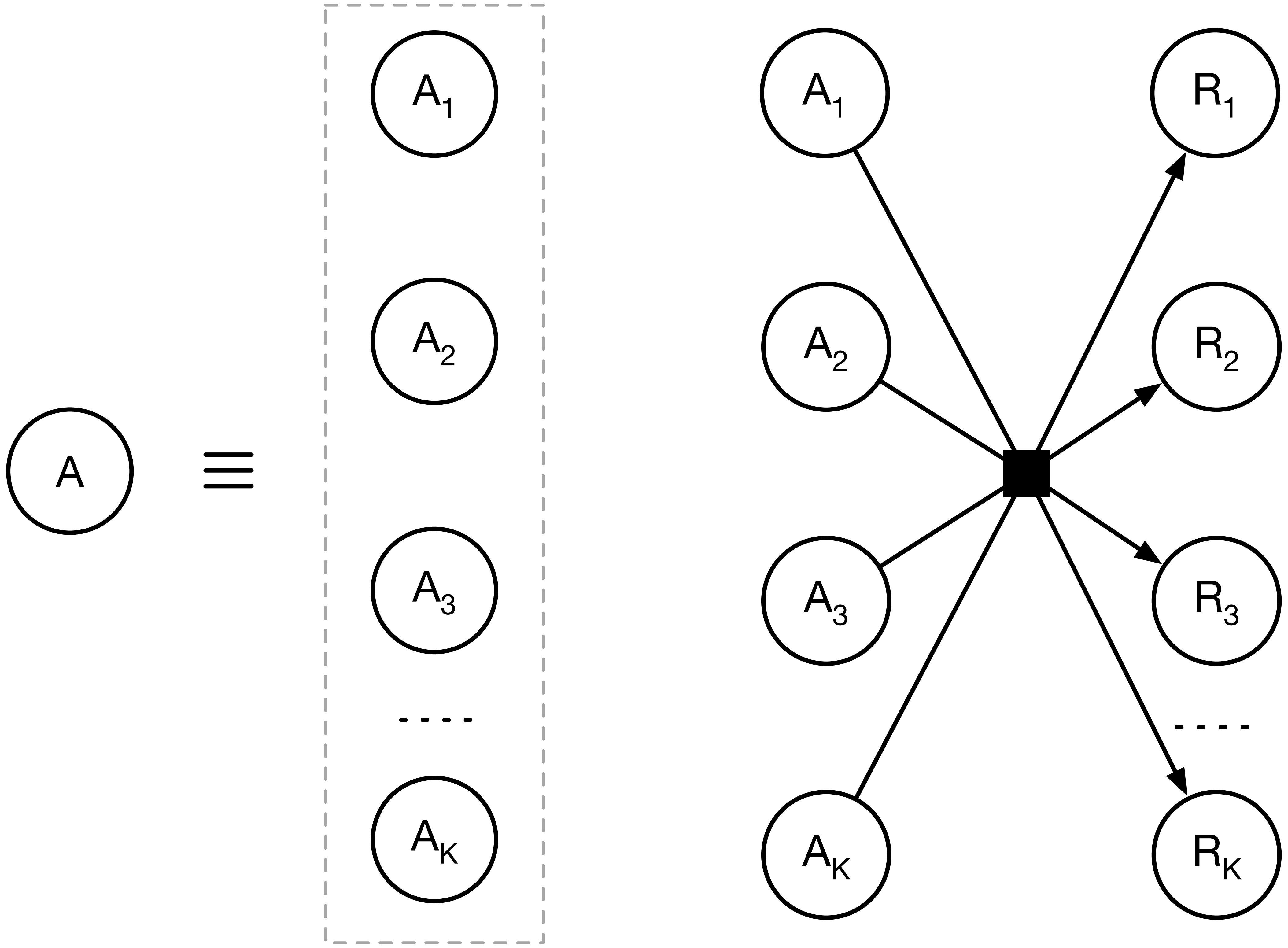
Slate Actions

Assumption: each action consists of K sub-actions, each associated with an observed reward.



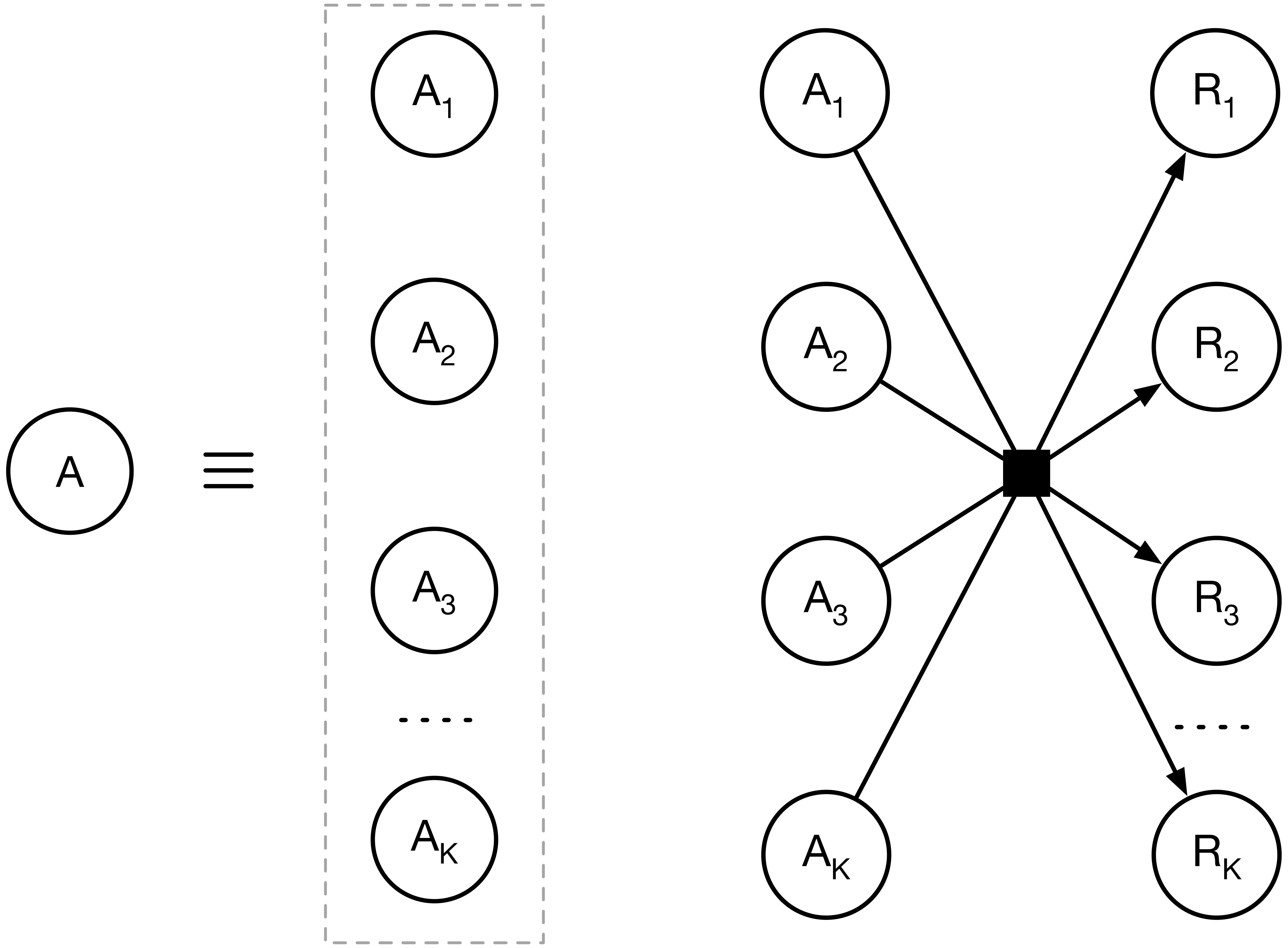
Slate Actions

Assumption: each action consists of K sub-actions, each associated with an observed reward.



Slate Actions

Assumption: each action consists of K sub-actions, each associated with an observed reward.



$$R = \sum_{k=1}^{K} R^{(k)}$$

IPS with Slate Actions

Large action space

Absolute continuity (effectively) violated

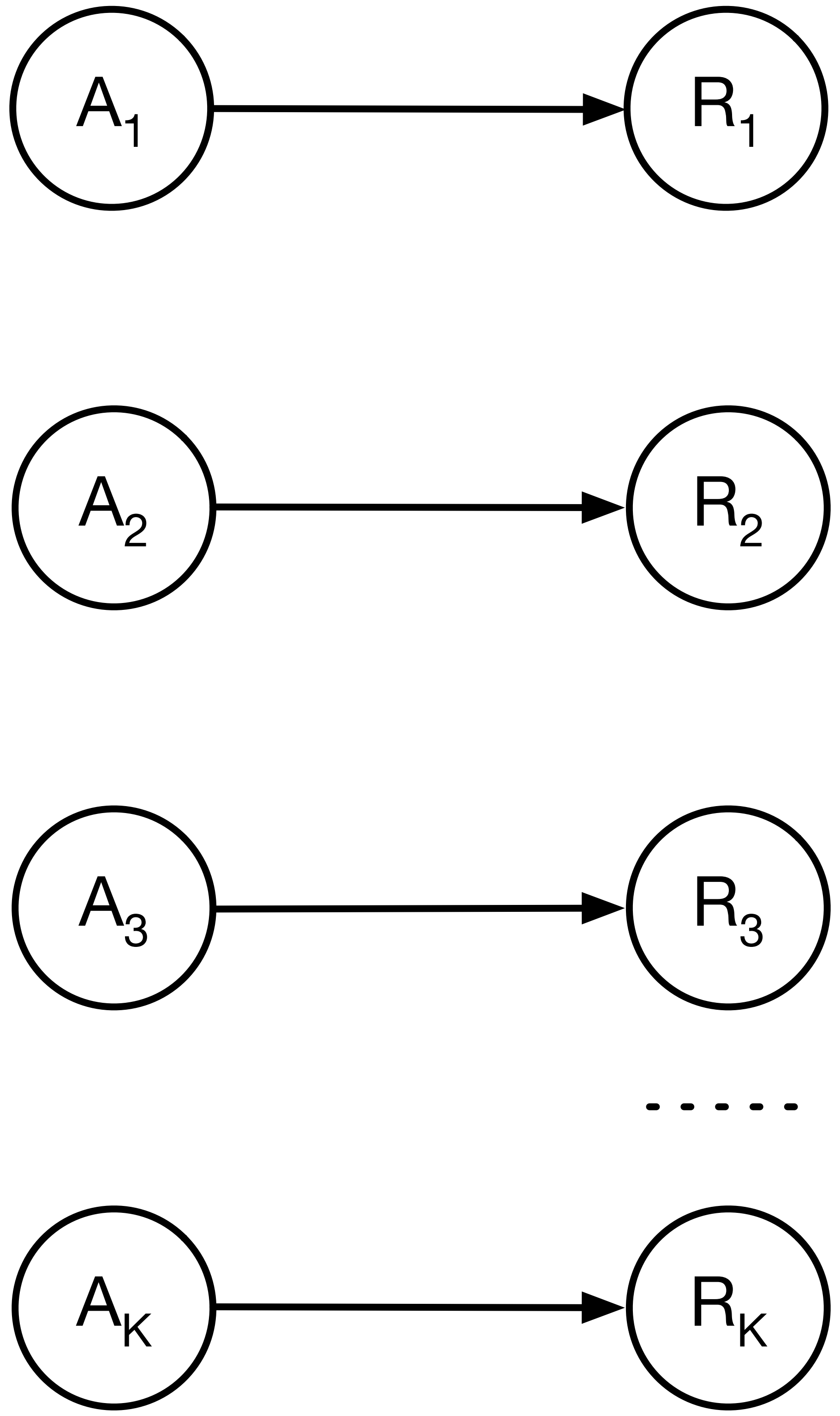
example:

π

A	R
{ a, b, c, e, d }	2
{ b, a, c, d, e }	3
{ b, c, a, d, e }	2
{ b, c, d, a, e }	1
{ b, c, d, e, a }	4

h = { a, b, c, d, e } ?

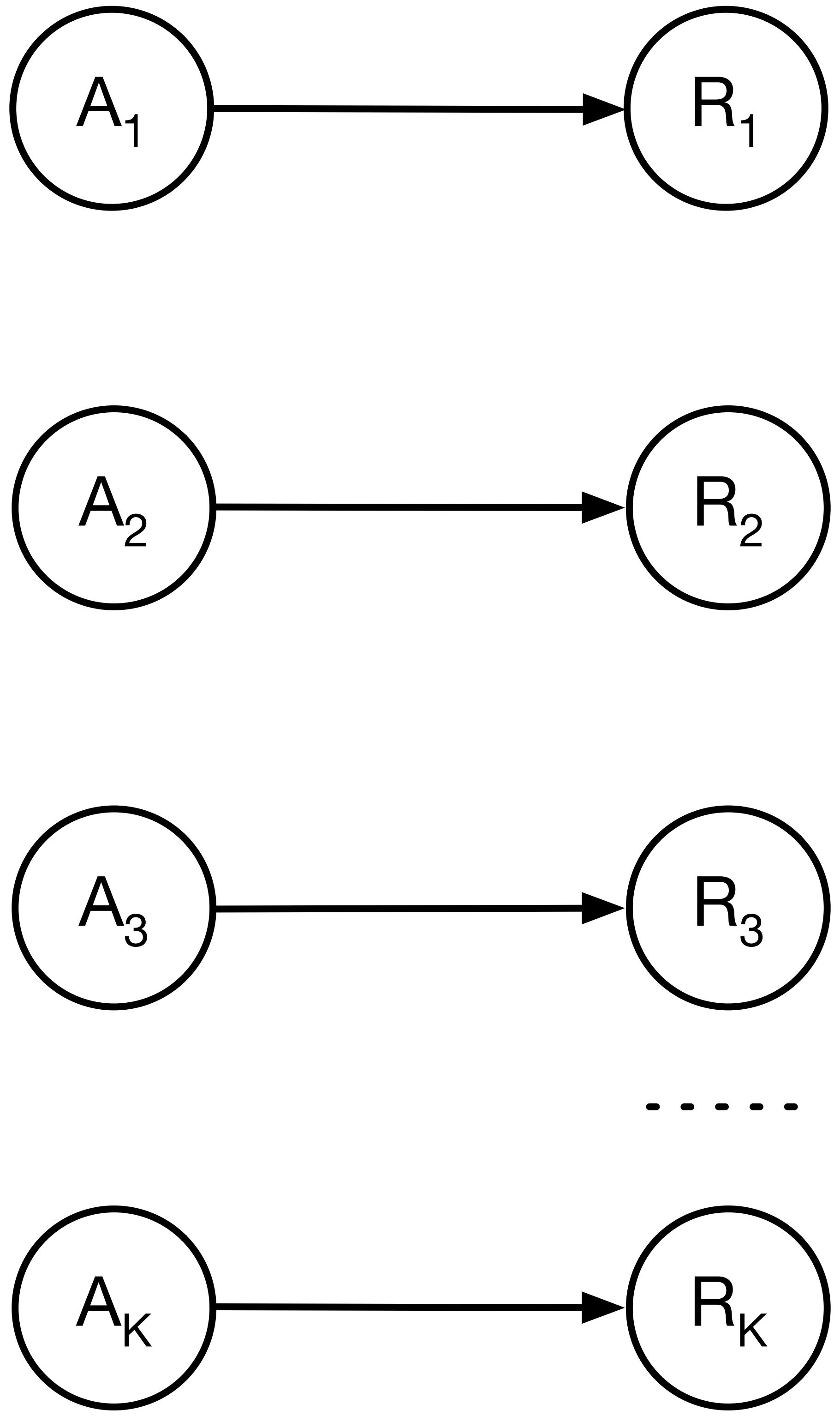
Independent IPS [Li et al. 2018]



Strong independence assumption:

- very convenient form (essentially have NK independent observations)
- much lower variance
- completely ignores reward interactions in the slate

Independent IPS [Li et al. 2018]

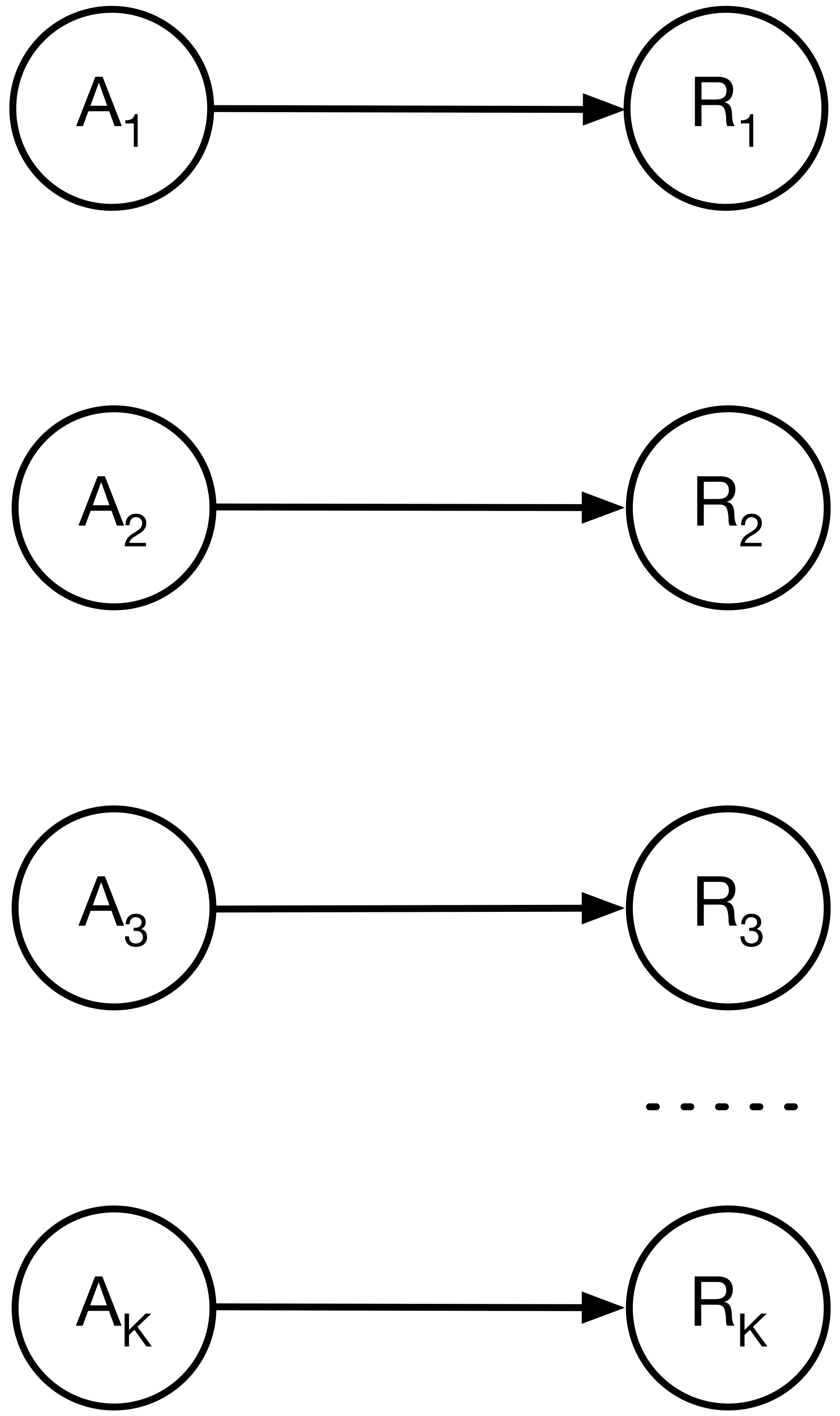


Strong independence assumption:

- very convenient form (essentially have NK independent observations)
- much lower variance
- completely ignores reward interactions in the slate

$$\bar{r}(h) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{h(a_{n,k} \mid x_n, k)}{\pi(a_{n,k} \mid x_n, k)} r_{n,k}$$

Independent IPS [Li et al. 2018]



Strong independence assumption:

— very convenient form (essentially have NK independent observations)

— much lower variance

— completely ignores reward interactions in the slate

$$\bar{r}(h) = \frac{1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} \frac{h(a_{n,k} \mid x_n, k)}{\pi(a_{n,k} \mid x_n, k)} r_{n,k}$$

other methods such as slate bandit [Swaminathan et al. 2017]

Markov Decision Process

- A Markov decision process (MDP) describes how an agent interacts with an environment.
- MDP is defined as:
  - a set of states $\mathscr{S}$
  - a set of actions $\mathscr{A}$
  - a reward function $\mathscr{S} \times \mathscr{A} \to \mathbb{R}$
  - a transition probability function $\mathscr{S} \times \mathscr{A} \times \mathscr{S} \to \mathbb{R}$

Markov Decision Process

- A Markov decision process (MDP) describes how an agent interacts with an environment.
- MDP is defined as:
  - a set of states $\mathcal{S}$
  - a set of actions $\mathcal{A}$
  - a reward function $\mathcal{S} \times \mathcal{A} \to \mathbb{R}$
  - a transition probability function $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$

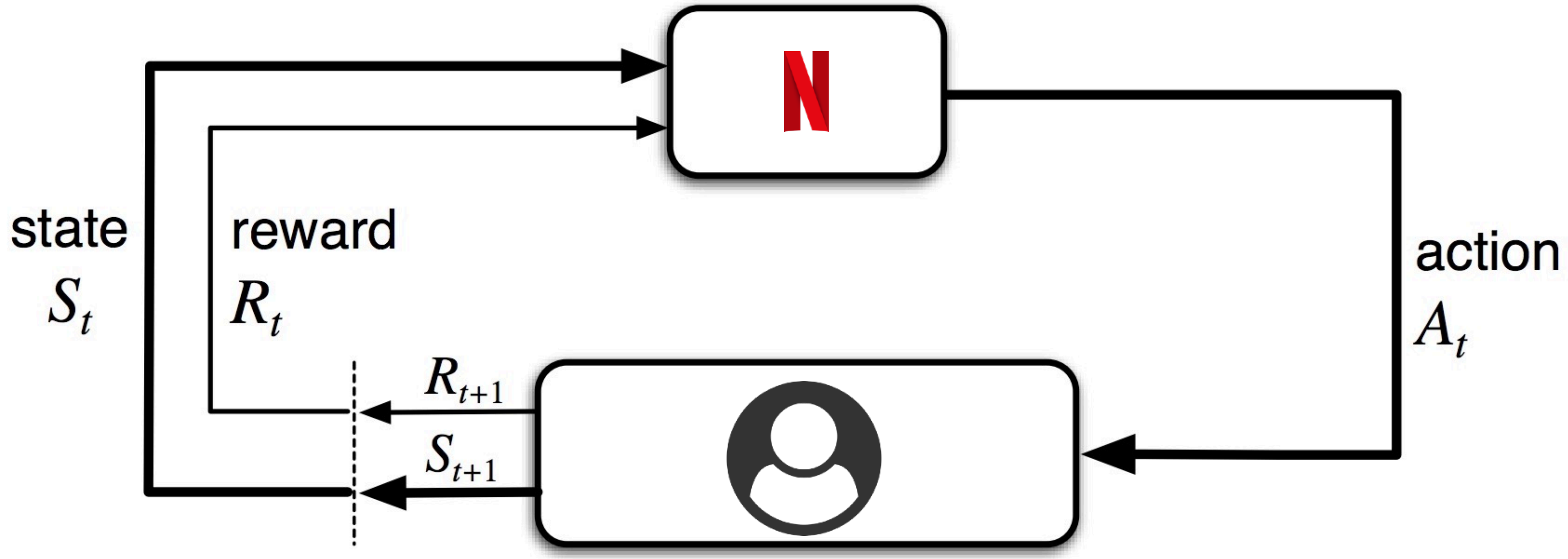


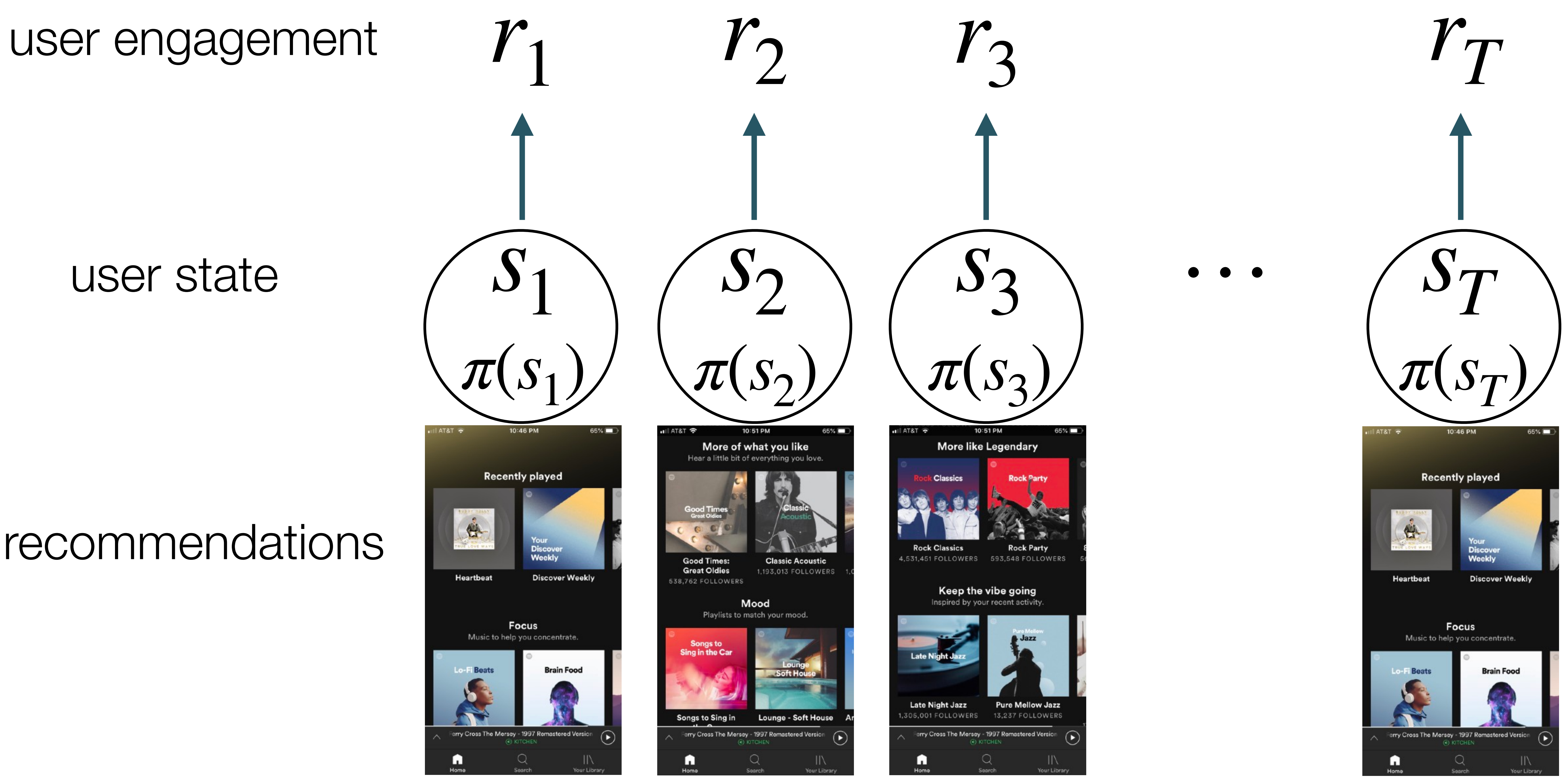
Diagram from RL bible: "Reinforcement Learning: An Introduction" (Sutton & Barto, 2017)

Bandits are a Special Type of Markov Decision Process

assumption is that actions in bandits do not affect future states

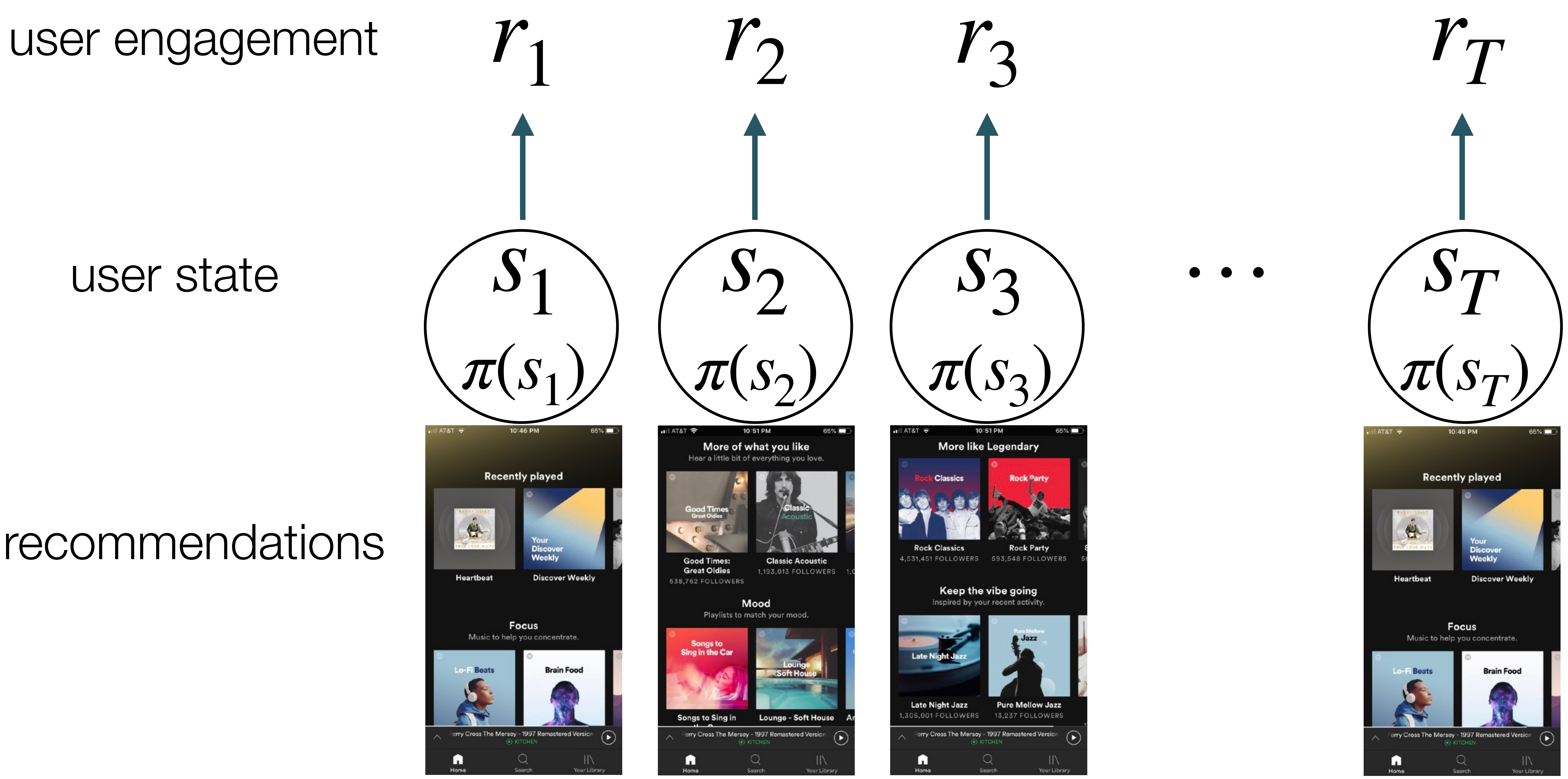
Bandits are a Special Type of Markov Decision Process

assumption is that actions in bandits do not affect future states



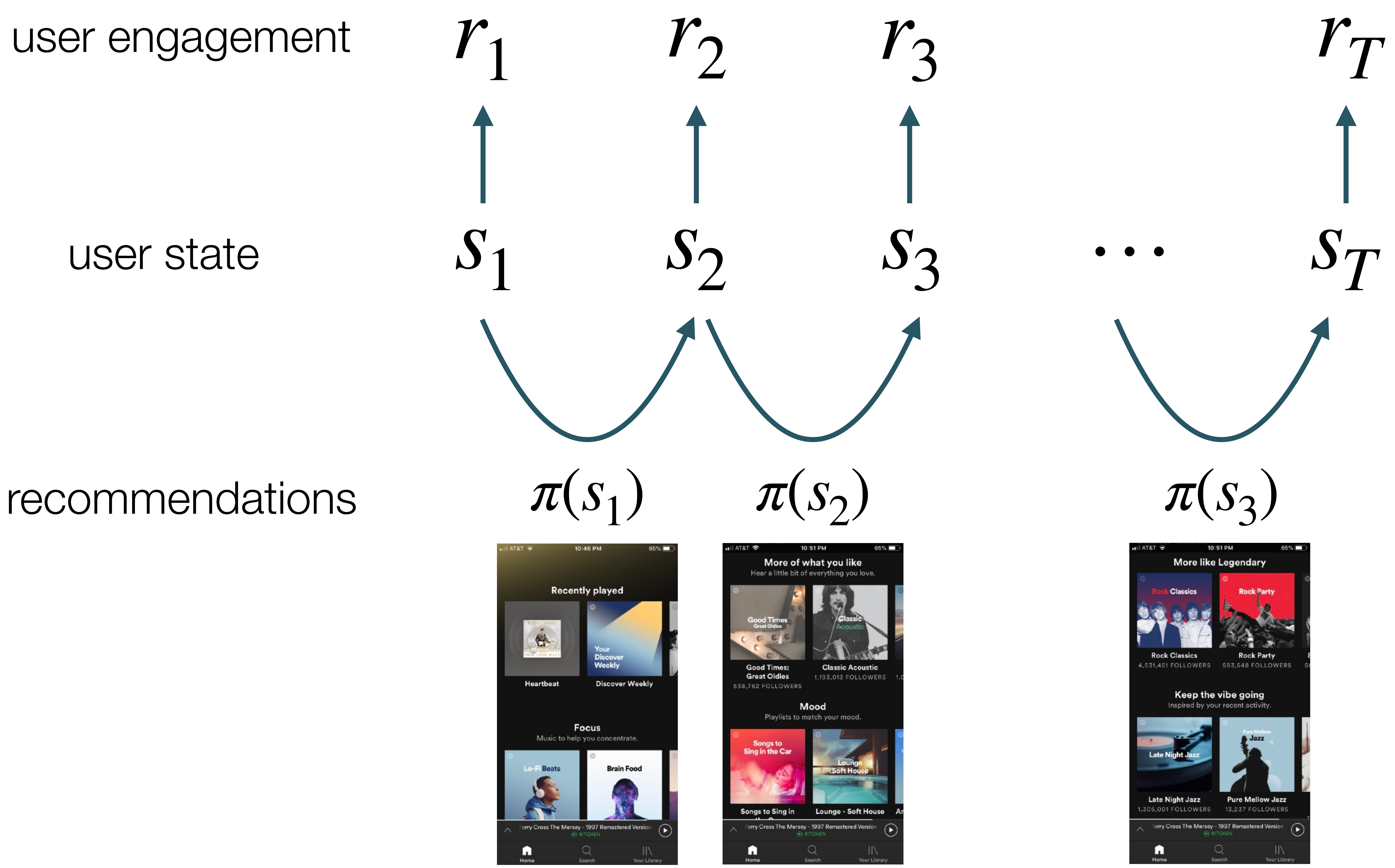
Bandits are a Special Type of Markov Decision Process

assumption is that actions in bandits do not affect future states



Leads to myopic policies

Bandits are a Special Type of Markov Decision Process



Thank You.

Special thanks for feedback (all errors are my own):

- Ashok Chandrashekar
- Arden Dertat
- Maria Dimakopoulou
- Ehtsham Elahi
- Maryam Esmaeili
- Mahdi Kalayeh
- Jingu Kim
- Dawen Liang
- Claudia Roberts
- Ehsan Saberian
- Kedar Sadekar
- Pannaga Shivaswamy
- Harald Steck

James McInerney
jmcinerney@netflix.com