Anytime Model Selection for Linear Bandits

Parnian Kassraie, Aldo Pacchiano, Nicolas Emmenegger, Andreas Krause

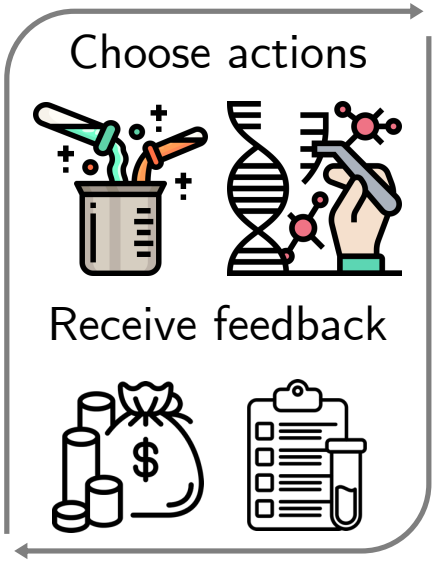



ETH zürich

Online Model Selection

Solving a Bandit/BO problem:

1. Commit to a reward model (a priori)
2. Interact with the environment accordingly



Repeat until happy

(w.r.t. some notion of regret)

image source: flaticon

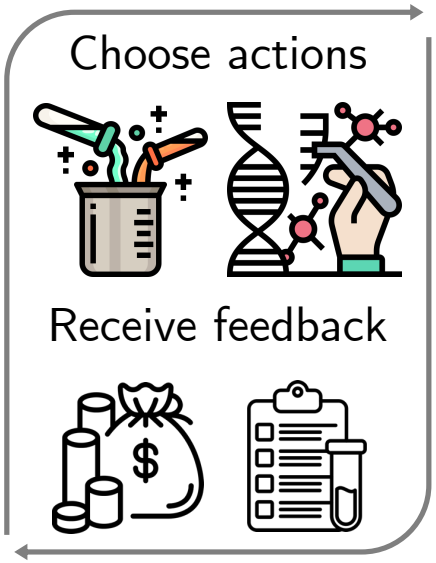
Online Model Selection

Solving a Bandit/BO problem:

1. Commit to a reward model (a priori)
2. Interact with the environment accordingly

There are many ways to model the reward

$$M \gg n$$ horizon/stopping time



Repeat until happy

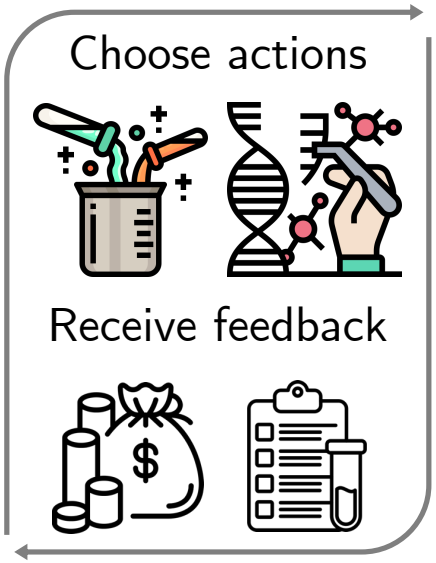
(w.r.t. some notion of regret)

image source: flaticon

Online Model Selection

Solving a Bandit/BO problem:

1. Commit to a reward model (a priori)
2. Interact with the environment accordingly



Repeat until happy

(w.r.t. some notion of regret)

There are many ways to model the reward

$$M \gg n$$ horizon/stopping time

Not known a priori which agent is going to be the best

e.g. in terms of sample efficiency

We can guess based on emprical evidence.

image source: flaticon

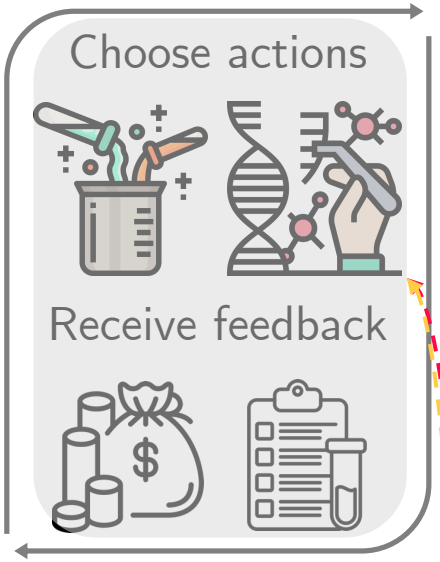
Online Model Selection

Solving a Bandit/BO problem:

1. Commit to a reward model (a priori)
2. Interact with the environment accordingly

There are many ways to model the reward

$$M \gg n$$ horizon/stopping time



Repeat until happy

(w.r.t. some notion of regret)

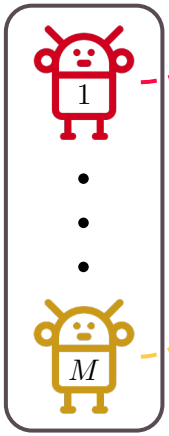
Not known a priori which agent is going to be the best

e.g. in terms of sample efficiency

We can guess based on emprical evidence.

Instatiate M agents each using a different model

Run all agents in parallel



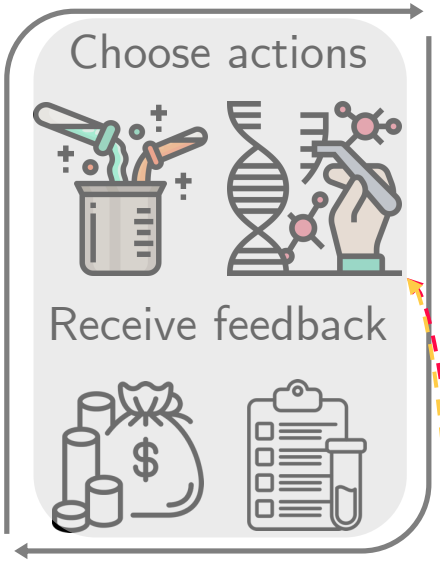
$\forall t \geq 1$

image source: flaticon

Online Model Selection

Solving a Bandit/BO problem:

1. Commit to a reward model (a priori)
2. Interact with the environment accordingly



Repeat until happy

(w.r.t. some notion of regret)

There are many ways to model the reward

$M \gg n$ horizon/stopping time

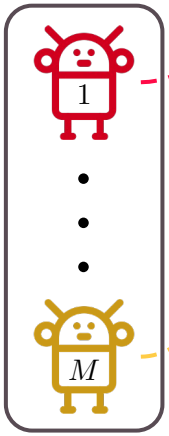
Not known a priori which agent is going to be the best

e.g. in terms of sample efficiency

We can guess based on emprical evidence.

Instatiate M agents each using a different model

Run all agents in parallel



$\forall t \geq 1$

Statistically expensive

poly$M$

image source: flaticon

Online Model Selection

Randomly iterate over the agents and at each step play only one

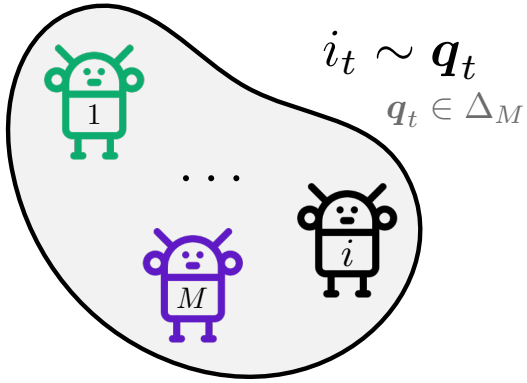


image source: flaticon

Online Model Selection

Randomly iterate over the agents and at each step play only one

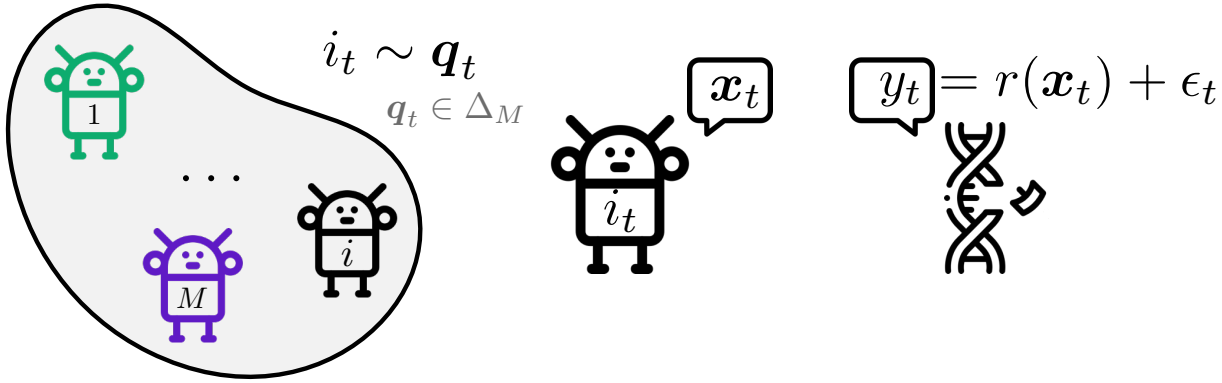


image source: flaticon

Online Model Selection

Randomly iterate over the agents and at each step play only one

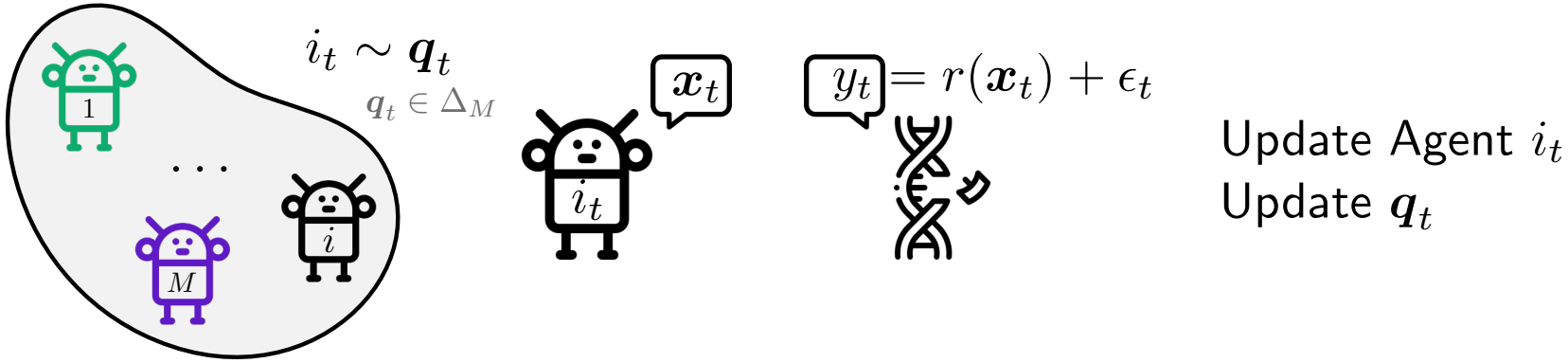
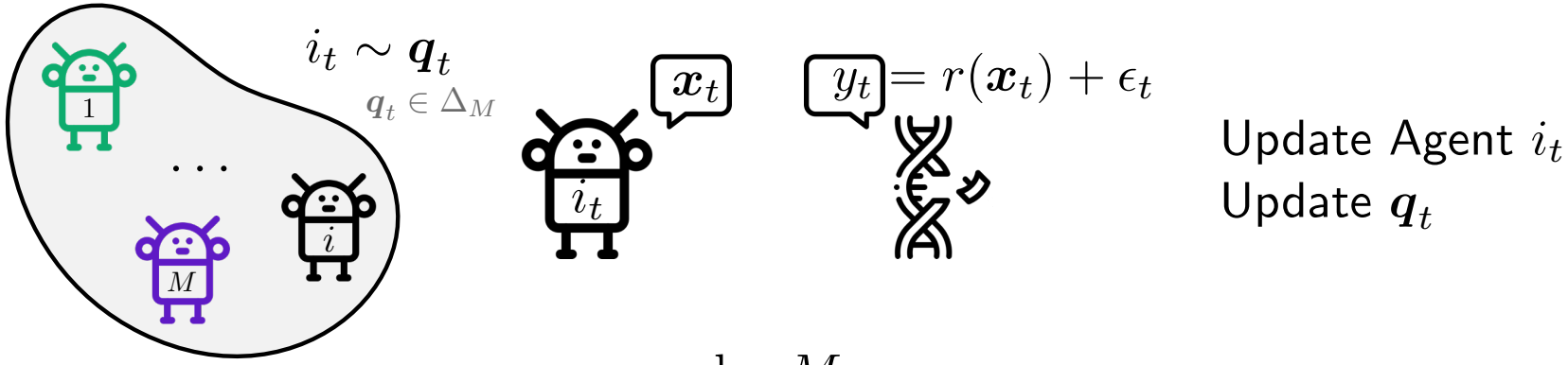


image source: flaticon

Online Model Selection

Randomly iterate over the agents and at each step play only one



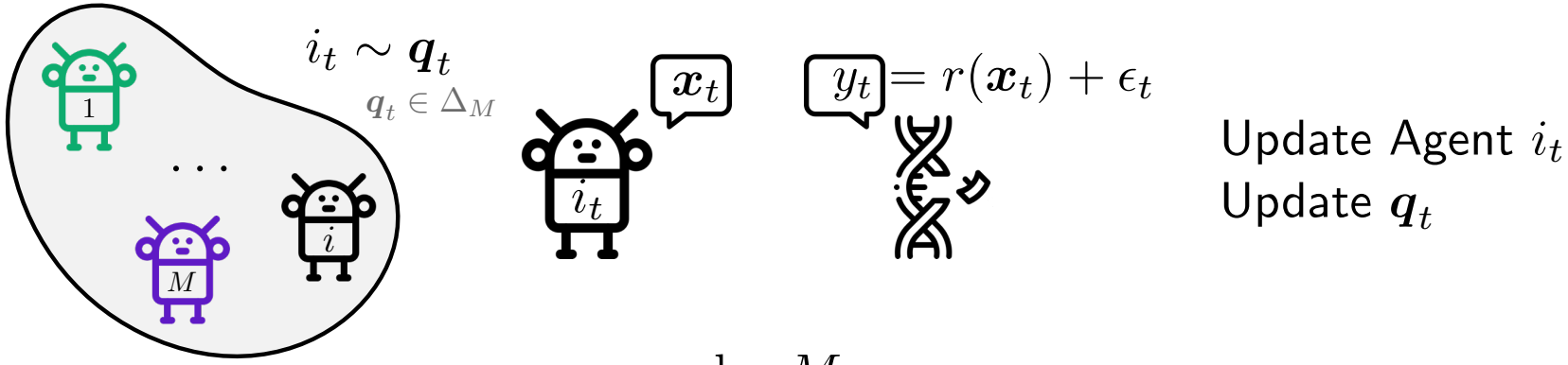
$$\log M$$

This is still poly(M). Is there a more efficient way? Open Problem (Agarwal et a. 2017)

image source: flaticon

Online Model Selection

Randomly iterate over the agents and at each step play only one



$$\log M$$

This is still poly(M). Is there a more efficient way? Open Problem (Agarwal et a. 2017)

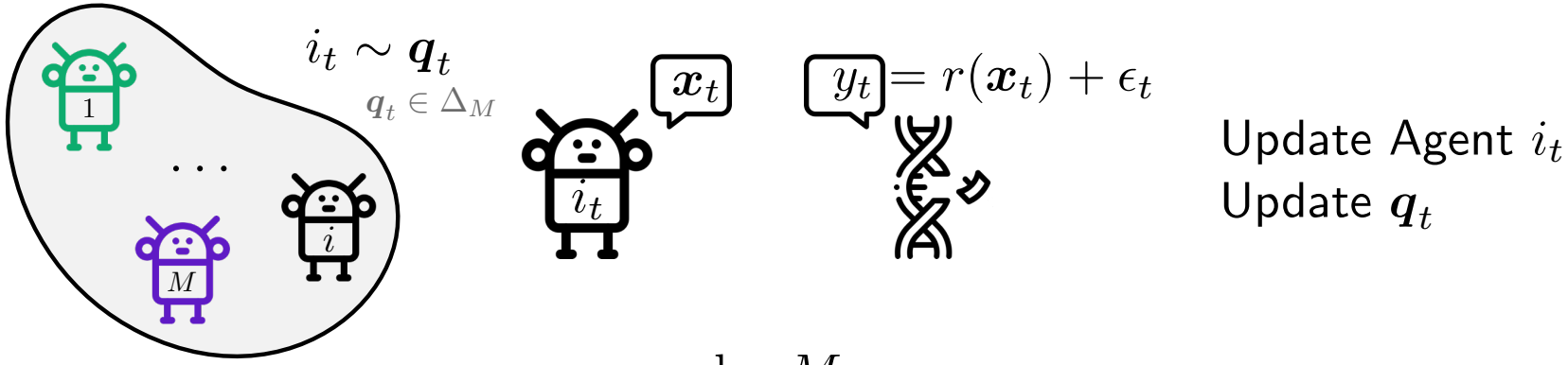
Our answer: If reward is linearly parametrizable, yes!

$$r(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

image source: flaticon

Online Model Selection

Randomly iterate over the agents and at each step play only one



$$\log M$$

This is still poly(M). Is there a more efficient way? Open Problem (Agarwal et a. 2017)

Our answer: If reward is linearly parametrizable, yes!

$$r(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Play one agent, but update all.
Reward not observed? Estimate it.

Choose your update rule and your estimator **very** carefully.

image source: flaticon

Problem Setting

Problem Setting

Environment

i.i.d. zero-mean sub-gaussian noise

$$\forall t \geq 1 \qquad y_t = r(\boldsymbol{x}_t) + \epsilon_t$$

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$

Problem Setting

Environment

i.i.d. zero-mean sub-gaussian noise

$$\forall t \geq 1 \qquad y_t = r(\boldsymbol{x}_t) + \epsilon_t$$

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$

Model Class

$$\{\boldsymbol{\phi}_j : \mathbb{R}^{d_0} \to \mathbb{R}^d, j = 1, \ldots, M\} \qquad M \gg n$$

$$\exists j^\star \in [M] \text{ s.t. } r(\cdot) = \boldsymbol{\theta}_{j^\star}^\top \boldsymbol{\phi}_{j^\star}(\cdot)$$

+ typical bdd assump.

Problem Setting

Environment

i.i.d. zero-mean sub-gaussian noise

$$\forall t \geq 1 \qquad y_t = r(\boldsymbol{x}_t) + \epsilon_t$$

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$

Model Class

$$\{\boldsymbol{\phi}_j : \mathbb{R}^{d_0} \to \mathbb{R}^d, j = 1, \ldots, M\} \qquad M \gg n$$

$$\exists\, j^\star \in [M] \text{ s.t. } r(\cdot) = \boldsymbol{\theta}_{j^\star}^\top \boldsymbol{\phi}_{j^\star}(\cdot)$$

+ typical bdd assump.

Agents

Agent j only uses $\boldsymbol{\phi}_j$ to model the reward

Has action selection policy $p_{t,j} \in \mathcal{M}(\mathcal{X})$

Which depends on the full history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$

Problem Setting

Environment

i.i.d. zero-mean sub-gaussian noise

$$\forall t \geq 1 \qquad y_t = r(\boldsymbol{x}_t) + \epsilon_t$$

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$

Model Class

$$\{\boldsymbol{\phi}_j : \mathbb{R}^{d_0} \to \mathbb{R}^d, j = 1, \ldots, M\} \qquad M \gg n$$

$$\exists\, j^\star \in [M] \text{ s.t. } r(\cdot) = \boldsymbol{\theta}_{j^\star}^\top \boldsymbol{\phi}_{j^\star}(\cdot)$$

+ typical bdd assump.

Agents

Agent j only uses $\boldsymbol{\phi}_j$ to model the reward

Has action selection policy $p_{t,j} \in \mathcal{M}(\mathcal{X})$

Which depends on the full history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$

Goal

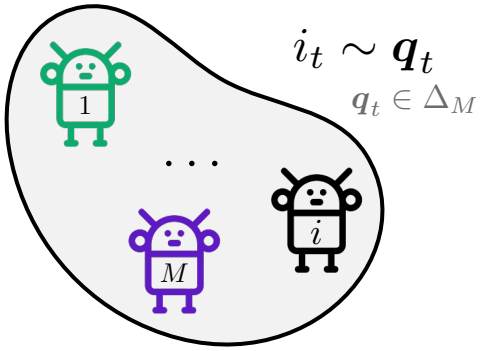
n unknown

$$R(n) = \sum_{t=1}^{n} r(\boldsymbol{x}^*) - r(\boldsymbol{x}_t)$$

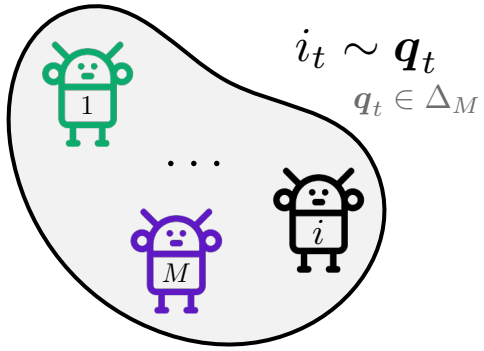
– Sublinear

– $\log M$

How to iterate over agents



How to iterate over agents



Exponential Weighting

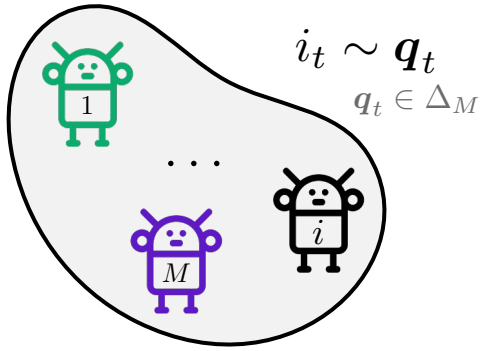
reward obtained by agent j so far

$$q_{t,j} = \frac{\exp(\eta_t \sum_{s=1}^{t-1} r_{s,j})}{\sum_{i=1}^{M} \exp(\eta_t \sum_{s=1}^{t-1} r_{s,i})}$$

sensitivity of updates

Known to yield $\log M$ regret in full-info setting

How to iterate over agents



Exponential Weighting

reward obtained by agent j so far

$$q_{t,j} = \frac{\exp(\eta_t \sum_{s=1}^{t-1} r_{s,j})}{\sum_{i=1}^{M} \exp(\eta_t \sum_{s=1}^{t-1} r_{s,i})}$$

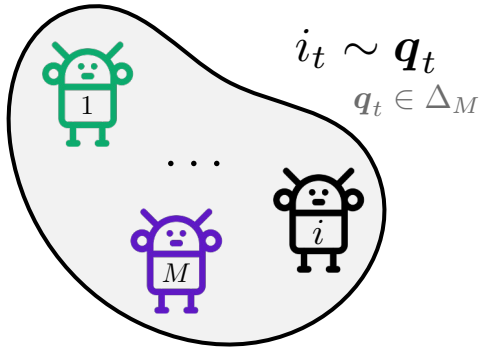
sensitivity of updates

Known to yield $\log M$ regret in full-info setting

$$= \frac{\exp(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,j})}{\sum_{i=1}^{M} \exp(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,i})}$$

$$\hat{r}_{t,j} = \mathbb{E}_{\boldsymbol{x} \sim p_{t,j}} \hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})$$

How to iterate over agents



Exponential Weighting

reward obtained by agent j so far

$$q_{t,j} = \frac{\exp(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,j})}{\sum_{i=1}^{M} \exp(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,i})}$$

sensitivity of updates

Known to yield $\log M$ regret in full-info setting

$$= \frac{\exp(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,j})}{\sum_{i=1}^{M} \exp(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,i})}$$

$$\hat{r}_{t,j} = \mathbb{E}_{\boldsymbol{x} \sim p_{t,j}} \hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Classical choices are unbiased high-variance estimators e.g. importance weighted or OLS estimator

Regret will depend on bias and variance of $\hat{\boldsymbol{\theta}}_t$

Ridge/OLS regression oracles are $\sqrt{M} \to \text{poly}M$ regret

How to hallucinate rewards

$$\hat{r}_{t,j} = \mathbb{E}_{\boldsymbol{x} \sim p_{t,j}} \hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Turn lasso into a sparse online regression oracle

$$\hat{\boldsymbol{\theta}}_t = \arg\min \frac{1}{t} ||\boldsymbol{y}_t - \Phi_t \boldsymbol{\theta}||_2^2 + \lambda_t \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||_2$$

How to hallucinate rewards

$$\hat{r}_{t,j} = \mathbb{E}_{\boldsymbol{x}\sim p_{t,j}} \hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Turn lasso into a sparse online regression oracle

$$\hat{\boldsymbol{\theta}}_t = \arg\min \frac{1}{t} ||\boldsymbol{y}_t - \Phi_t \boldsymbol{\theta}||_2^2 + \lambda_t \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||_2$$

Theorem (Anytime Conf. Seq.)

If for all $t \geq 1$

$$\lambda_t \geq \frac{c_1}{\sqrt{t}} \sqrt{\log(M/\delta) + \sqrt{d\,(\log(M/\delta) + (\log\log d)_+)}}$$

then,

c_1 and c_2 made exact in the paper

$$\mathbb{P}\left(\forall t \geq 1 : \left\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\right\|_2 \leq \frac{c_2 \lambda_t}{\kappa^2(\Phi_t, 2)}\right) \geq 1 - \delta$$

Restricted Eigenvalue property [check paper]

How to hallucinate rewards

$$\hat{r}_{t,j} = \mathbb{E}_{\boldsymbol{x}\sim p_{t,j}} \hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Turn lasso into a sparse online regression oracle

$$\hat{\boldsymbol{\theta}}_t = \arg\min \frac{1}{t} ||\boldsymbol{y}_t - \Phi_t \boldsymbol{\theta}||_2^2 + \lambda_t \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||_2$$

Variance & bias are both $\log M$

Theorem (Anytime Conf. Seq.)

If for all $t \geq 1$

cost of going 'time uniform'

$$\lambda_t \geq \frac{c_1}{\sqrt{t}} \sqrt{\log(M/\delta) + \sqrt{d\,(\log(M/\delta) + (\log\log d)_+)}}$$

then,

c_1 and c_2 made exact in the paper

$$\mathbb{P}\left(\forall t \geq 1 : \left\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\right\|_2 \leq \frac{c_2 \lambda_t}{\kappa^2(\Phi_t, 2)}\right) \geq 1 - \delta$$

Restricted Eigenvalue property [check paper]

How to hallucinate rewards

$$\hat{r}_{t,j} = \mathbb{E}_{\boldsymbol{x}\sim p_{t,j}} \hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Turn lasso into a sparse online regression oracle

$$\hat{\boldsymbol{\theta}}_t = \arg\min \frac{1}{t} ||\boldsymbol{y}_t - \Phi_t \boldsymbol{\theta}||_2^2 + \lambda_t \sum_{j=1}^{M} ||\boldsymbol{\theta}_j||_2$$

Variance & bias are both $\log M$

Theorem (Anytime Conf. Seq.)

If for all $t \geq 1$

cost of going 'time uniform'

$$\lambda_t \geq \frac{c_1}{\sqrt{t}} \sqrt{\log(M/\delta) + \sqrt{d\,(\log(M/\delta) + (\log\log d)_+)}}$$

then, c_1 and c_2 made exact in the paper

$$\mathbb{P}\left(\forall t \geq 1 : \left\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\right\|_2 \leq \frac{c_2 \lambda_t}{\kappa^2(\Phi_t, 2)}\right) \geq 1 - \delta$$

Restricted Eigenvalue property [check paper]

Difference with offline Lasso?

Instead of sub-gaussian concentration,

Design a self-normalized martingale based on $\left|\left|\left(\Phi_t^\top \boldsymbol{\epsilon}_t\right)_j\right|\right|$ — Empirical process error

Apply a "stitched" time uniform boundary [Howard et al. '21]

ETH zürich

Thanks to Aaditya & SAVI friends!

Putting it all together: ALExp

Anytime **Exp**onential weighting algorithm with **L**asso reward estimates

Algorithm 1 ALEXP

Inputs: γ_t, η_t, λ_t for $t \geq 1$

for $t \geq 1$ **do**

Draw $\boldsymbol{x}_t \sim (1-\gamma_t)\sum_{j=1}^{M} q_{t,j} p_{t,j} + \gamma_t \text{Unif}(\mathcal{X})$

Observe $y_t = r(\boldsymbol{x}_t) + \epsilon_t$.

Append history $H_t = H_{t-1} \cup \{(\boldsymbol{x}_t, y_t)\}$.

Update agents $p_{t,j}$ for $j = 1, \ldots, M$.

Calculate $\hat{\boldsymbol{\theta}}_t \leftarrow \text{Lasso}(H_t, \lambda_t)$ and estimate

$$\hat{r}_{t,j} \leftarrow \mathbb{E}_{\boldsymbol{x} \sim p_{t+1,j}}[\hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})]$$

Update selection distribution

$$q_{t+1,j} \leftarrow \frac{\exp(\eta_t \sum_{s=1}^{t} \hat{r}_{s,j})}{\sum_{i=1}^{M} \exp(\eta_t \sum_{s=1}^{t} \hat{r}_{s,i})}$$

Putting it all together: ALExp

Anytime **Exp**onential weighting algorithm with **L**asso reward estimates

Algorithm 1 ALEXP

Inputs: γ_t, η_t, λ_t for $t \geq 1$

for $t \geq 1$ **do**

Draw $\boldsymbol{x}_t \sim (1-\gamma_t)\sum_{j=1}^{M} q_{t,j} p_{t,j} + \gamma_t \text{Unif}(\mathcal{X})$

Observe $y_t = r(\boldsymbol{x}_t) + \epsilon_t$.

Append history $H_t = H_{t-1} \cup \{(\boldsymbol{x}_t, y_t)\}$.

Update agents $p_{t,j}$ for $j = 1, \ldots, M$.

Calculate $\hat{\boldsymbol{\theta}}_t \leftarrow \text{Lasso}(H_t, \lambda_t)$ and estimate

$$\hat{r}_{t,j} \leftarrow \mathbb{E}_{\boldsymbol{x} \sim p_{t+1,j}}[\hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})]$$

Update selection distribution

$$q_{t+1,j} \leftarrow \frac{\exp(\eta_t \sum_{s=1}^{t} \hat{r}_{s,j})}{\sum_{i=1}^{M} \exp(\eta_t \sum_{s=1}^{t} \hat{r}_{s,i})}$$

prescribed in the paper

Theorem (Regret - Informal)

For appropriate choices of $(\gamma_t, \lambda_t, \eta_t)$,

$$R(n) = \mathcal{O}\left(C(M, \delta, d)\left(\sqrt{n}\log M + n^{3/4}\right)\right)$$

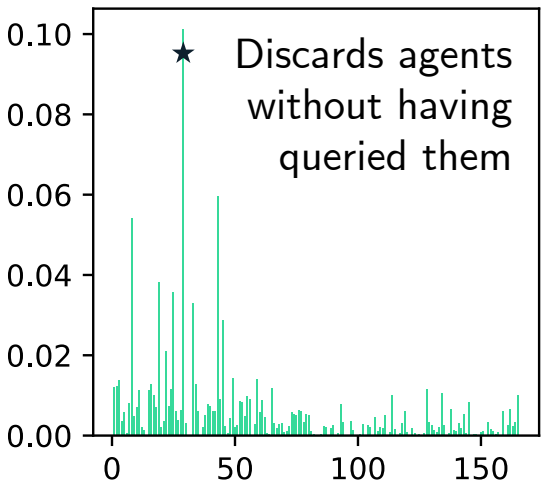
with probability greater than $1-\delta$, *simultaneously for all* $n \geq 1$.

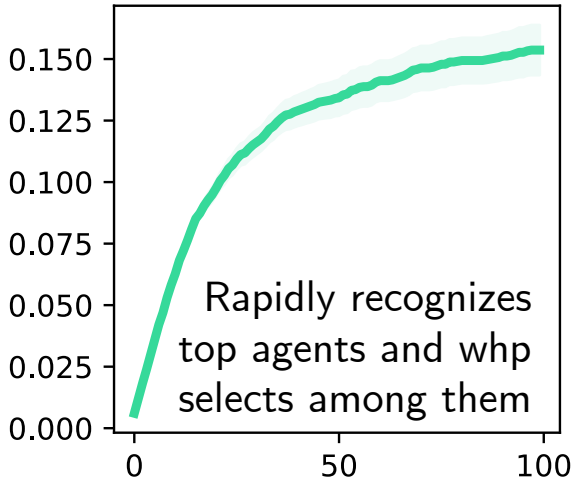
$$C(M, \delta, d) = \mathcal{O}\left(\sqrt{d \log M/\delta + \sqrt{d \log M/\delta}}\right)$$

Synthetic Experiments

data generation & baselines
described in the paper.

How does
ALExp work?

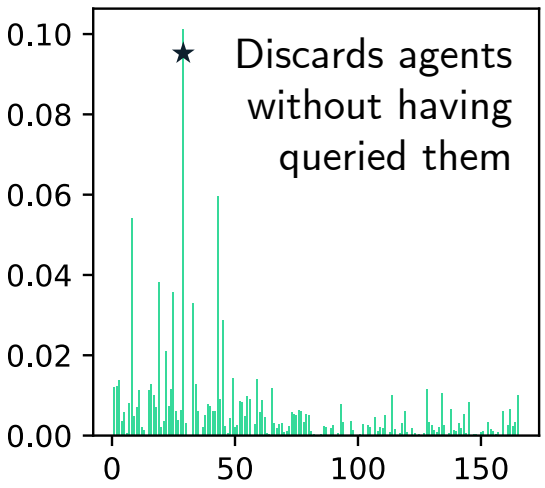


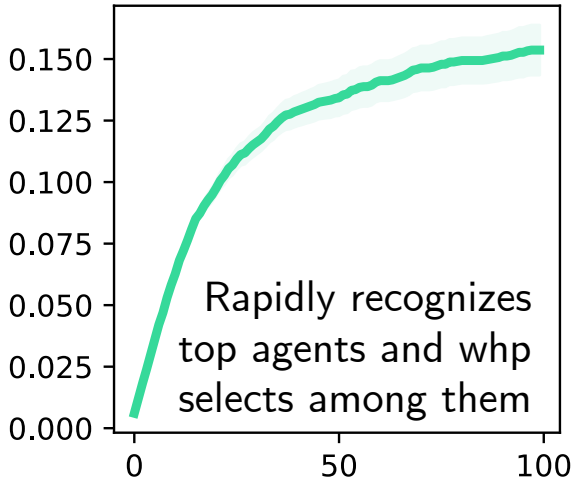


Synthetic Experiments

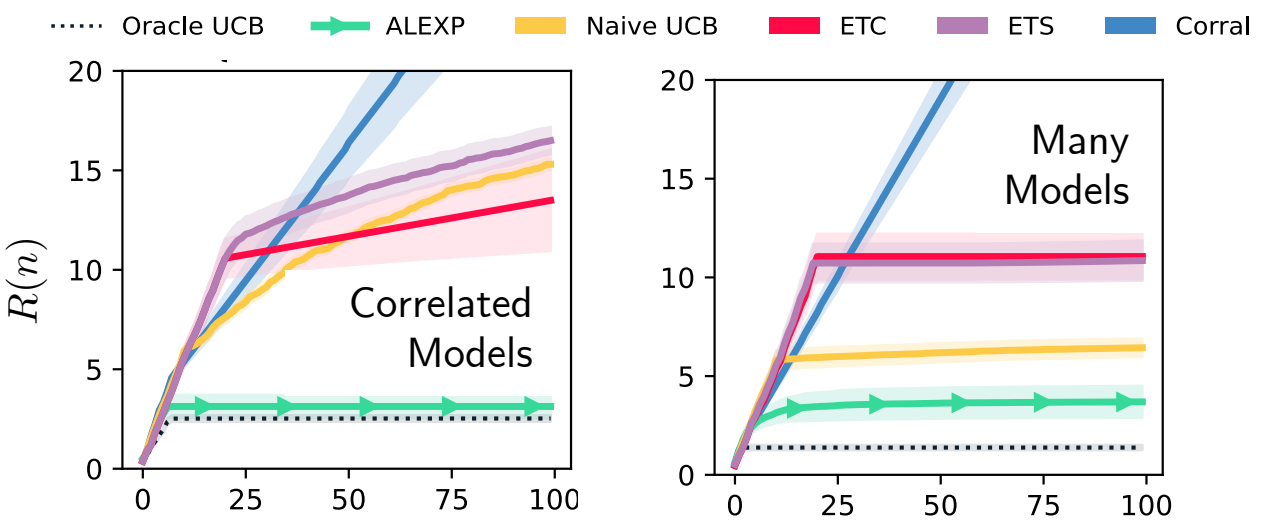
data generation & baselines
described in the paper.

How does
ALExp work?





How well?



Thank you.