# Model Selection for Sequential Inference and Decision-making

Parnian Kassraie, ETH Zurich

ETH zürich | Learning & Adaptive Systems

# Sequential Decision-Making & Bandits: Problem

At every step $t$

Choose actions $\boldsymbol{x}_t$

unknown reward

$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

obsv.noise

Receive feedback $y_t$

Repeat

image source: flaticon

# Sequential Decision-Making & Bandits: Problem

At every step $t$

Choose actions $\boldsymbol{x}_t$

unknown reward

$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

obsv.noise

Receive feedback $y_t$


Repeat

Goal: Choose actions that give a high reward

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

image source: flaticon

# Sequential Decision-Making & Bandits: Problem

At every step $t$

Choose actions $\boldsymbol{x}_t$

unknown reward

$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

obsv.noise

Receive feedback $y_t$

Repeat

Goal: Choose actions that give a high reward

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

Motivation: maximize r using the fewest queries

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

To take actions at every step:

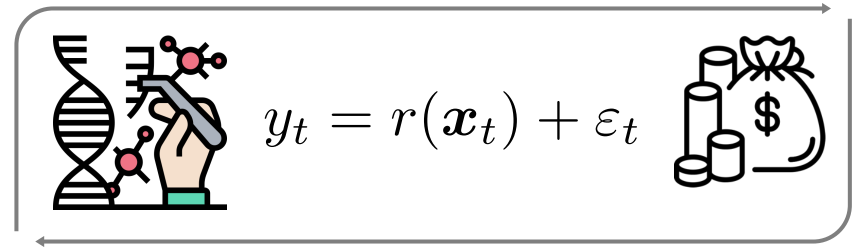
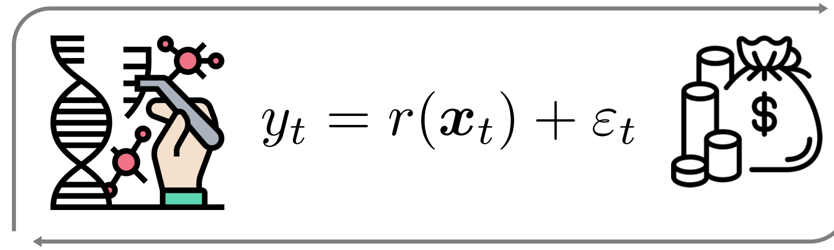



image source: flaticon

# Sequential Decision-Making & Bandits: Solutions




To take actions at every step:

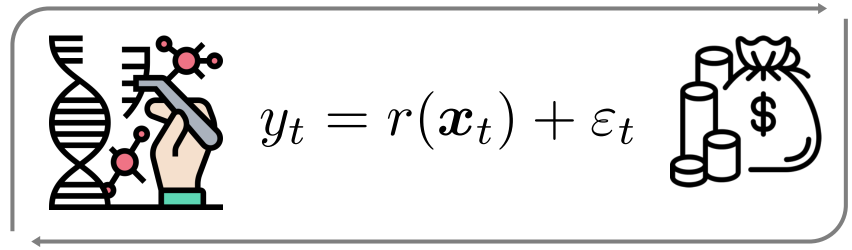
- Estimate the reward function

based on:

Statistical model for the reward e.g. r is a linear function

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

# Sequential Decision-Making & Bandits: Solutions




To take actions at every step:

- Estimate the reward function

  based on:
  - Statistical model for the reward e.g. r is a linear function
  - history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$

- Use reward estimate to choose the next action

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$

To take actions at every step:

- Estimate the reward function

based on: Statistical model for the reward e.g. r is a linear function

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

- Use reward estimate to choose the next action

(better) estimate r

explore

maximize r

exploit

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$

To take actions at every step:

- Estimate the reward function

based on:
Statistical model for the reward e.g. r is a linear function
history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$

- Use reward estimate to choose the next action

Many principles: optimism, expected improvement, entropy search

(better) estimate r
explore

maximize r
exploit

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$

To take actions at every step:

- Estimate the reward function

based on:
Statistical model for the reward e.g. r is a linear function
history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$

- Use reward estimate to choose the next action

Many principles: optimism, expected improvement, entropy search

(better) estimate r
explore

maximize r
exploit

Heavily rely on the choice of model → Model selection is key!

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$

To take actions at every step:

- Estimate the reward function

based on:

Statistical model for the reward e.g. r is a linear function

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

- Use reward estimate to choose the next action

(better) estimate r

explore

maximize r

exploit

Model selection in this setting is not fun and games...

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$



To take actions at every step:

- Estimate the reward function

based on:

Statistical model for the reward e.g. r is a linear function

history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$ samples are non-i.i.d

- Use reward estimate to choose the next action

(better) estimate r
explore

maximize r
exploit

Model selection in this setting is not fun and games...

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

To take actions at every step:

- Estimate the reward function

based on: Statistical model for the reward e.g. r is a linear function

history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$ samples are non-i.i.d

- Use reward estimate to choose the next action

(better) estimate r
explore

maximize r
exploit

samples are not so diverse

Model selection in this setting is not fun and games...

image source: flaticon

# Sequential Decision-Making & Bandits: Solutions

$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

To take actions at every step:

- Estimate the reward function

based on:

Statistical model for the reward e.g. r is a linear function

history $H_{t-1} = \{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_{t-1}, y_{t-1})\}$ samples are non-i.i.d

- Use reward estimate to choose the next action

(better) estimate r
explore

maximize r
exploit

samples are not so diverse

Model selection in this setting is not fun and games...

Open problem: when is (efficient) online model selection possible?

[Agarwal et al. 2017]

image source: flaticon

## Online Model Selection problem

Find $j^\star$ while maximizing for the unknown $r$

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

– Sublinear in $T$

– $\log M$

image source: flaticon

## Online Model Selection problem

Find $j^\star$ … n $r$

…linear in $T$

$t=1$ … g $M$

**Why do we need to select?**
**Why not just try out everything?**

## Online Model Selection problem

Find $j^\star$ ... n $r$

... linear in $T$

$t=1$ ... g $M$

Why do we need to select?
Why not just try out everything?

Instatiate M algorithms each using a different model

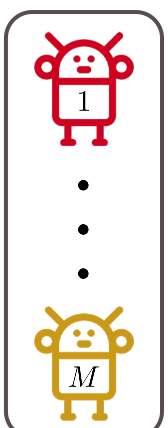



image source: flaticon

## Online Model Selection problem

Find $j^\star$ ... n $r$

... linear in $T$

$t=1$ ... g $M$

**Why do we need to select?**
**Why not just try out everything?**

Instatiate M algorithms each using a different model

Run all algorithms in parallel (or search through them, adversarial-bandit-style)

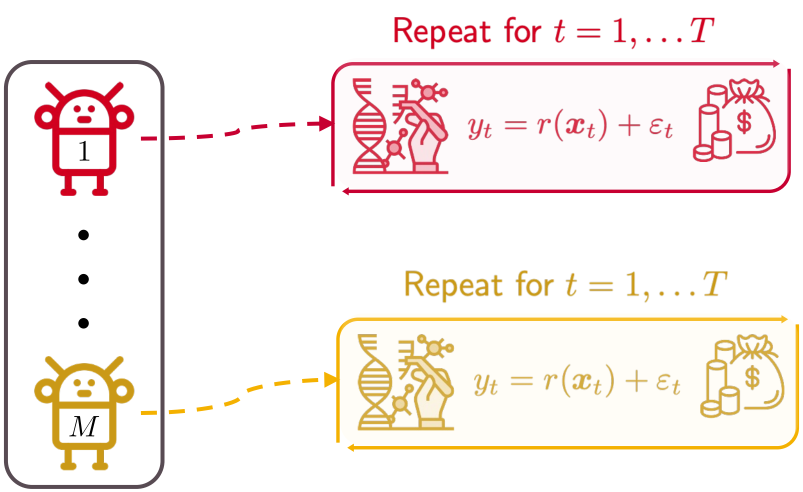



image source: flaticon

Online Model Selection problem

Find $j^\star$ … n $r$

…linear in $T$

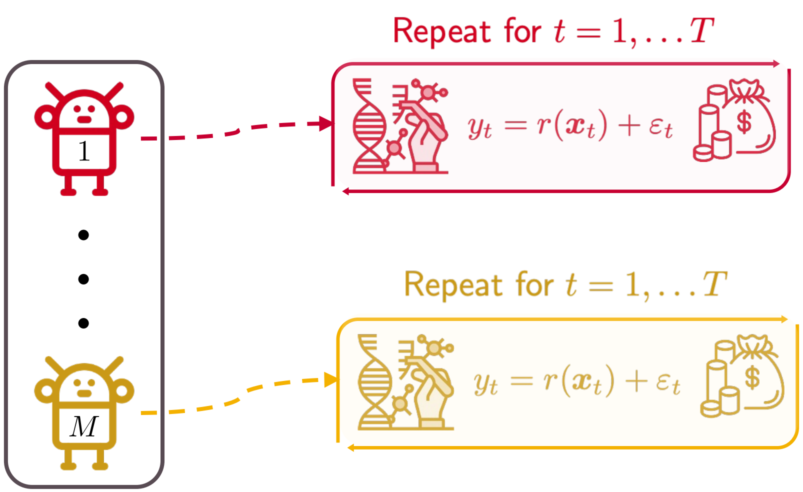
…g $M$

$t=1$

Why do we need to select?
Why not just try out everything?

Instatiate M algorithms each using a different model

Run all algorithms in parallel (or search through them, adversarial-bandit-style)




Statistically expensive
$\leftarrow\rightarrow$ High regret

$$\mathrm{poly}(M)$$

image source: flaticon

# Problem Setting in this Talk

# Problem Setting in this Talk

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$
$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

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

# Problem Setting in this Talk

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$
$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

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

The reward is linearly parametrized by an unknown feature map

Model Class $\{\boldsymbol{\phi}_j : \mathbb{R}^{d_0} \rightarrow \mathbb{R}^d,\ j = 1, \dots, M\}$ $\quad M \gg T$

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

+ typical bdd assump. $\|r\|_\infty \leq B$

# Problem Setting in this Talk

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$
$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

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

The reward is linearly parametrized by an unknown feature map

Model Class $\{\boldsymbol{\phi}_j : \mathbb{R}^{d_0} \to \mathbb{R}^d,\, j = 1, \dots, M\}$ $M \gg T$

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

+ typical bdd assump. $\|r\|_\infty \leq B$

Online Model Selection problem:

Find $j^\star$ while maximizing for the unknown $r$

# Problem Setting in this Talk

$$\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$$
$$y_t = r(\boldsymbol{x}_t) + \varepsilon_t$$

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

The reward is linearly parametrized by an unknown feature map

Model Class $\{\boldsymbol{\phi}_j : \mathbb{R}^{d_0} \to \mathbb{R}^d,\, j = 1, \dots, M\}$ $M \gg T$

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

+ typical bdd assump. $\|r\|_\infty \leq B$

Online Model Selection problem:

Find $j^\star$ while maximizing for the unknown $r$

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

– Sublinear in $T$

– $\log M$

# Warm-up Solution: Explore then Commit

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or “diverse” distribution

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or "diverse" distribution

Use Group Lasso for implicit model selection

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M) \in \mathbb{R}^{dM}$$

$$\hat{\boldsymbol{\theta}} = \arg\min \frac{1}{T_0} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{\theta}_j\|_2$$

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\boldsymbol{\phi}_1(\boldsymbol{x}), \ldots, \boldsymbol{\phi}_M(\boldsymbol{x}))$$

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or "diverse" distribution

Use Group Lasso for implicit model selection

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M) \in \mathbb{R}^{dM}$$

$$\hat{\boldsymbol{\theta}} = \arg\min \frac{1}{T_0} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{\theta}_j\|_2$$

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\boldsymbol{\phi}_1(\boldsymbol{x}), \ldots, \boldsymbol{\phi}_M(\boldsymbol{x}))$$

For the remaining steps, always do

$$\boldsymbol{x}_t = \arg\max \hat{\boldsymbol{\theta}}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or "diverse" distribution

Use Group Lasso for implicit model selection

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M) \in \mathbb{R}^{dM}$$

$$\hat{\boldsymbol{\theta}} = \arg\min \frac{1}{T_0} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{\theta}_j\|_2$$

$$\boldsymbol{\phi}(\boldsymbol{x}) = (\boldsymbol{\phi}_1(\boldsymbol{x}), \dots, \boldsymbol{\phi}_M(\boldsymbol{x}))$$

For the remaining steps, always do

$$\boldsymbol{x}_t = \arg\max \hat{\boldsymbol{\theta}}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Under good choice of $T_0$ and $\lambda$ satisfies,

$$R(T) = \mathcal{O}(\sqrt[3]{T^2 \log M}) \quad \text{w.h.p.}$$

(matches lower bound in certain action domains)

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or "diverse" distribution

Use the Lasso for implicit model selection

$$\hat{\boldsymbol{\theta}} = \arg\min \frac{1}{T_0} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{\theta}_j\|_2$$

For the remaining steps, always do

$$\boldsymbol{x}_t = \arg\max \hat{\boldsymbol{\theta}}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Under good choice of $T_0$ and $\lambda$ satisfies,

$$R(T) = \mathcal{O}(\sqrt[3]{T^2 \log M}) \quad \text{w.h.p.}$$

(matches lower bound in certain action domains)

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or "diverse" distribution

Incur high regret of $2BT_0$

Use the Lasso for implicit model selection

$$\hat{\boldsymbol{\theta}} = \arg\min \frac{1}{T_0} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{\theta}_j\|_2$$

For the remaining steps, always do

$$\boldsymbol{x}_t = \arg\max \hat{\boldsymbol{\theta}}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Under good choice of $T_0$ and $\lambda$ satisfies,

$$R(T) = \mathcal{O}(\sqrt[3]{T^2 \log M}) \quad \text{w.h.p.}$$

(matches lower bound in certain action domains)

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or "diverse" distribution

Incur high regret of $2BT_0$

Use the Lasso for implicit model selection

$$\hat{\boldsymbol{\theta}} = \arg\min \frac{1}{T_0} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{\theta}_j\|_2$$

Relies on Lasso variable selection

For the remaining steps, always do

$$\boldsymbol{x}_t = \arg\max \hat{\boldsymbol{\theta}}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Under good choice of $T_0$ and $\lambda$ satisfies,

$$R(T) = \mathcal{O}(\sqrt[3]{T^2 \log M}) \quad \text{w.h.p.}$$

(matches lower bound in certain action domains)

# Warm-up Solution: Explore then Commit

For $T_0$ steps, take i.i.d. samples following a uniform, or "diverse" distribution

Incur high regret of $2BT_0$

Use the Lasso for implicit model selection

$$\hat{\boldsymbol{\theta}} = \arg\min \frac{1}{T_0} \|\boldsymbol{y} - \Phi\boldsymbol{\theta}\|_2^2 + \lambda \sum_{j=1}^{M} \|\boldsymbol{\theta}_j\|_2$$

Relies on Lasso variable selection

For the remaining steps, always do

$$\boldsymbol{x}_t = \arg\max \hat{\boldsymbol{\theta}}^\top \boldsymbol{\phi}(\boldsymbol{x})$$

Is not any-time: only works if horizon $T$ is known in advance (doubling trick aside)

Under good choice of $T_0$ and $\lambda$ satisfies,

$$R(T) = \mathcal{O}(\sqrt[3]{T^2 \log M}) \quad \text{w.h.p.}$$

(matches lower bound in certain action domains)

# Online Model Selection

# Online Model Selection

Instead of commiting to a single model,

Randomly iterate over the models and at each step choose one

# Online Model Selection

Instead of commiting to a single model,

Randomly iterate over the models and at each step choose one

Instatiate M "agents"

- Agent j only uses $\phi_j$ to model the reward
- Has action selection strategy $p_{t,j} \in \mathcal{M}(\mathcal{X})$ which is updated at every step e.g. UCB [for those who know]

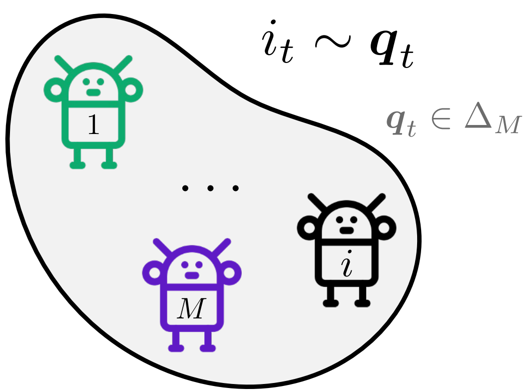



image source: flaticon

# Online Model Selection

Instead of commiting to a single model,

Randomly iterate over the models and at each step choose one

Instatiate M "agents/algorithms"

Agent j only uses $\phi_j$ to model the reward

Has action selection strategy $p_{t,j} \in \mathcal{M}(\mathcal{X})$ which is updated at every step e.g. UCB

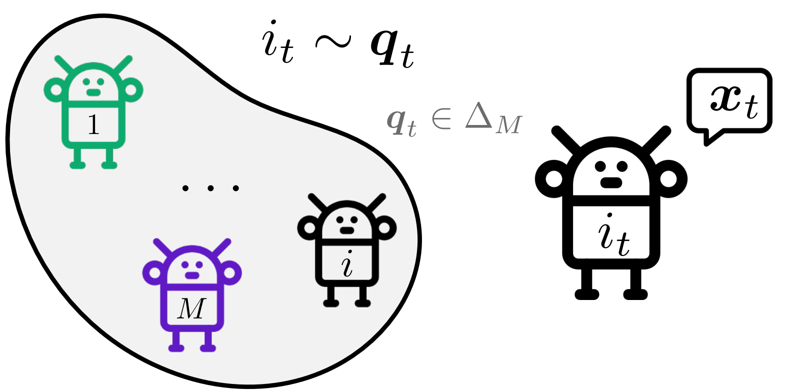



image source: flaticon

# Online Model Selection

Instead of commiting to a single model,

Randomly iterate over the models and at each step choose one

Instatiate M "agents"

- Agent j only uses $\phi_j$ to model the reward
- Has action selection strategy $p_{t,j} \in \mathcal{M}(\mathcal{X})$ which is updated at every step e.g. UCB

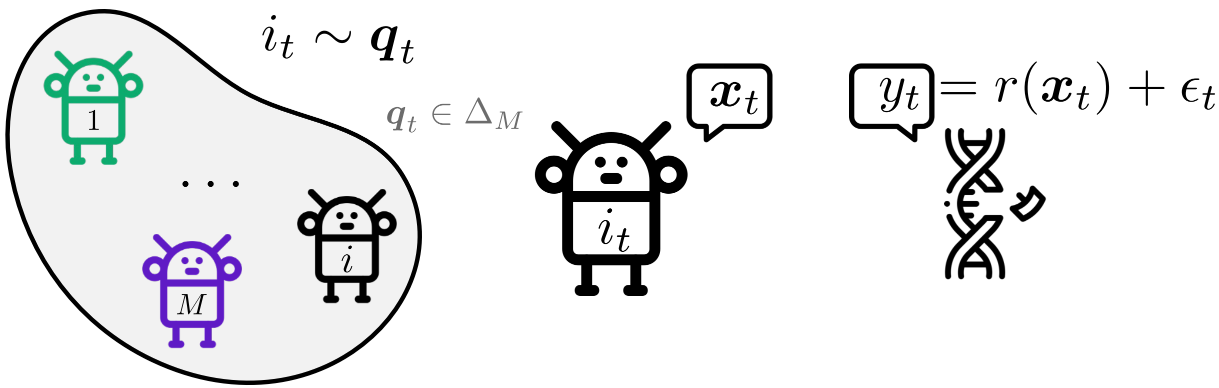



image source: flaticon

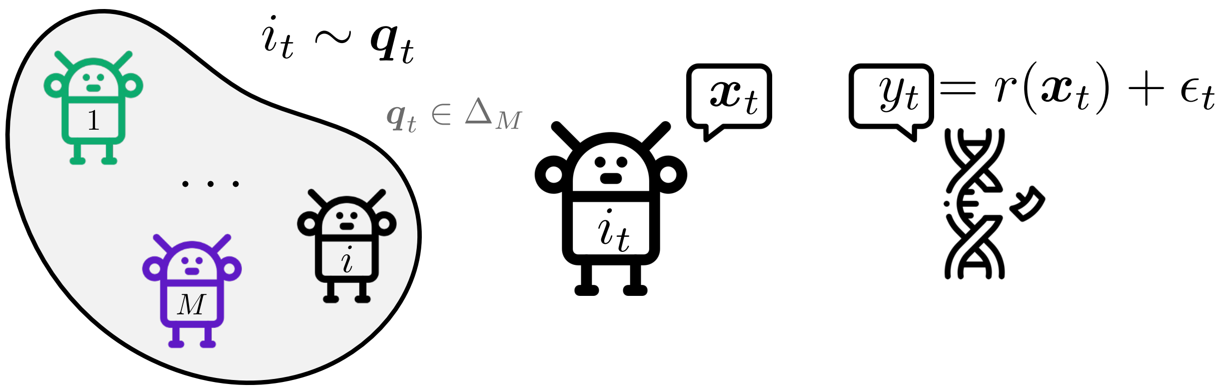
# Online Model Selection

Instead of commiting to a single model,

Randomly iterate over the models and at each step choose one

Instatiate M "agents"

- Agent j only uses $\phi_j$ to model the reward
- Has action selection strategy $p_{t,j} \in \mathcal{M}(\mathcal{X})$ which is updated at every step e.g. UCB




Update $\boldsymbol{q}_t$

Update all agents

image source: flaticon

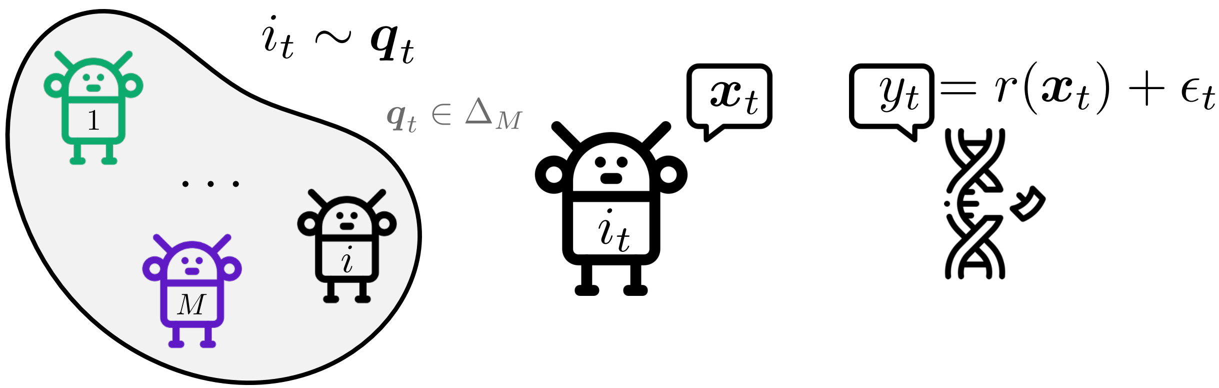
# Online Model Selection

Instead of commiting to a single model,

Randomly iterate over the models and at each step choose one

Instatiate M "agents"

- Agent j only uses $\phi_j$ to model the reward
- Has action selection strategy $p_{t,j} \in \mathcal{M}(\mathcal{X})$ which is updated at every step e.g. UCB




Update $\boldsymbol{q}_t$

Update all agents

Requires having observed the reward for the choice of each agent

Reward not observed? Hallucinate it.

image source: flaticon

# How to hallucinate rewards



[KEKP 2023]

# How to hallucinate rewards

Turn group lasso into a sparse online regression oracle

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

# How to hallucinate rewards

Turn group lasso into a sparse online regression oracle

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

**Theorem (Anytime Lasso Conf Seq)**

*For appropriate choice of* $(\lambda_t)_{t\geq 1}$,

$$\mathbb{P}\left(\forall t \geq 1 :\ \left\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\right\|_2 \lessapprox \sqrt{\frac{\log(M/\delta)}{t}}\right) \geq 1 - \delta$$

# How to hallucinate rewards

Turn group lasso into a sparse online regression oracle

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

**Theorem (Anytime Lasso Conf Seq)**

*For appropriate choice of* $(\lambda_t)_{t\geq 1}$,

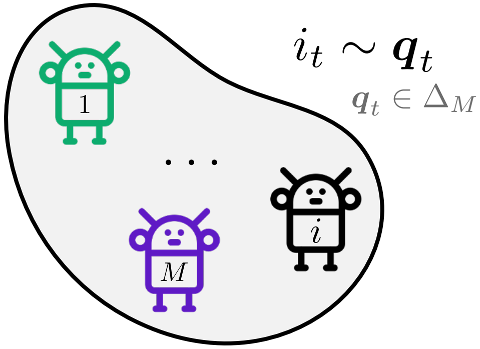
$$\mathbb{P}\left(\forall t \geq 1 : \left\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_t\right\|_2 \lessapprox \sqrt{\frac{\log(M/\delta)}{t}}\right) \geq 1 - \delta$$

Hallucinate the reward of agent j as

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

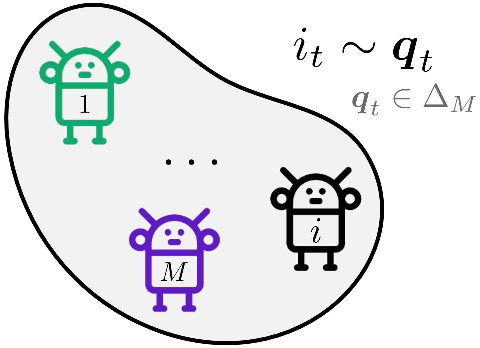
$p_{t,j} \in \mathcal{M}(\mathcal{X})$ action selection strategy

# How to iterate over agents




increase $q_{t,j}$ if $\hat{r}_{t,j}$ was high

# How to iterate over agents



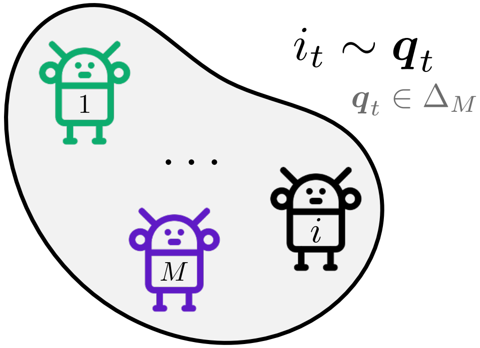

increase $q_{t,j}$ if $\hat{r}_{t,j}$ was high

## Exponential Weighting

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

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

# How to iterate over agents




increase $q_{t,j}$ if $\hat{r}_{t,j}$ was high

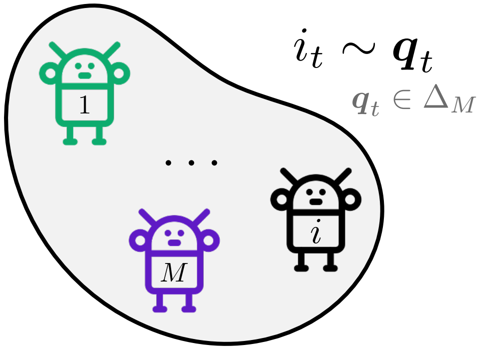
Exponential Weighting

Estimate of the reward obtained by agent j so far

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

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

# How to iterate over agents




increase $q_{t,j}$ if $\hat{r}_{t,j}$ was high

## Exponential Weighting

Estimate of the 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

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

# Putting it all together

Find $j^\star$ while maximizing for the unknown $r$

**A**nytime **Exp**onential weighting algorithm with **L**asso reward estimates

# Putting it all together

Find $j^\star$ while maximizing for the unknown $r$

**A**nytime **Exp**onential weighting algorithm with **L**asso reward estimates

**Algorithm 1** ALEXP

Inputs: $\gamma_t$, $\eta_t$, $\lambda_t$ 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 $j = 1, \dots, M$.

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

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

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

Find $j^\star$ while maximizing for the unknown $r$

**A**nytime **Exp**onential weighting algorithm with **L**asso reward estimates

**Algorithm 1** ALEXP

Inputs: $\gamma_t$, $\eta_t$, $\lambda_t$ for $t \geq 1$

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

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

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 $j = 1, \dots, M$.

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

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

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

Find $j^\star$ while maximizing for the unknown $r$

**A**nytime **Exp**onential weighting algorithm with **L**asso reward estimates

**Algorithm 1** ALEXP

Inputs: $\gamma_t$, $\eta_t$, $\lambda_t$ for $t \geq 1$
**for** $t \geq 1$ **do**
Draw $\boldsymbol{x}_t \sim (1-\gamma_t)\sum_{j=1}^{M} q_{t,j} p_{t,j} + \gamma_t \mathrm{Unif}(\mathcal{X})$
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 $j = 1, \dots, M$.
Calculate $\hat{\boldsymbol{\theta}}_t \leftarrow \mathrm{Lasso}(H_t, \lambda_t)$ and estimate

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

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

Find $j^\star$ while maximizing for the unknown $r$

**A**nytime **Exp**onential weighting algorithm with **L**asso reward estimates

**Algorithm 1** ALEXP

Inputs: $\gamma_t,\ \eta_t,\ \lambda_t$ for $t \geq 1$

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

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

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 $j = 1, \dots, M$.

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

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

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

Find $j^\star$ while maximizing for the unknown $r$

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

**Algorithm 1** ALEXP

Inputs: $\gamma_t$, $\eta_t$, $\lambda_t$ for $t \geq 1$

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

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

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 $j = 1, \dots, M$.

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

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

**Theorem (Online Model Selection)**

*For appropriate choices of parameters, ALEXP with a UCB oracle agent satisfies*

$$R(T) = \mathcal{O}\left(\sqrt{T \log^3 M} + T^{3/4}\sqrt{\log M}\right)$$

*w.h.p. simultaneously for all $T \geq 1$.*

[Open problem of Agarwal et al. 2017 in the Linear case]

# Putting it all together: ALExp

Find $j^\star$ while maximizing for the unknown $r$

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

**Algorithm 1** ALEXP

Inputs: $\gamma_t, \eta_t, \lambda_t$ for $t \geq 1$

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

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

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 $j = 1, \dots, M$.

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

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

**Theorem (Online Model Selection)**

*For appropriate choices of parameters, ALEXP with a UCB oracle agent satisfies*

$$R(T) = \mathcal{O}\left(\sqrt{T \log^3 M} + T^{3/4}\sqrt{\log M}\right)$$

*w.h.p. simultaneously for all* $T \geq 1$.

[Open problem of Agarwal et al. 2017 in the Linear case]

Probably not tight? Lower bounds not clear.

# A classic interpretation of ALExp [for bandit enthusiasts]

# A classic interpretation of ALExp [for bandit enthusiasts]

... is almost an Exp4 algorithm.

each expert is adaptive

as oppose to static experts with pre-set sequence of actions/advices

regression oracle is Lasso

as oppose to Importance Weighted Estimator or OLS

# A classic interpretation of ALExp [for bandit enthusiasts]

... is almost an Exp4 algorithm.

each expert is adaptive

as oppose to static experts with pre-set sequence of actions/advices

regression oracle is Lasso

as oppose to Importance Weighted Estimator or OLS

In this context, regret w.r.t. to agent j roughly is..

$$R(T,j) \leq \frac{\log M}{\eta} + \eta \sum_{t=1}^{T} \mathrm{Var}^2(\hat{\boldsymbol{\theta}}_t) + \sum_{t=1}^{T} \mathrm{Bias}(\hat{\boldsymbol{\theta}}_t)$$

# A classic interpretation of ALExp [for bandit enthusiasts]

... is almost an Exp4 algorithm.

each expert is adaptive

as oppose to static experts with pre-set sequence of actions/advices

regression oracle is Lasso

as oppose to Importance Weighted or OLS estimators

In this context, regret w.r.t. to agent j roughly is..

$$R(T, j) \leq \frac{\log M}{\eta} + \eta \sum_{t=1}^{T} \mathrm{Var}^2(\hat{\boldsymbol{\theta}}_t) + \sum_{t=1}^{T} \mathrm{Bias}(\hat{\boldsymbol{\theta}}_t)$$

$\hat{\boldsymbol{\theta}}_t \in \mathbb{R}^{dM}$ { OLS/IW: $\mathrm{Var}^2$ term $M$, $\mathrm{Bias}$ term $0$

# A classic interpretation of ALExp [for bandit enthusiasts]

… is almost an Exp4 algorithm.

| each expert is adaptive | regression oracle is Lasso |
| --- | --- |
| as oppose to static experts with pre-set sequence of actions/advices | as oppose to Importance Weighted Estimator or OLS |

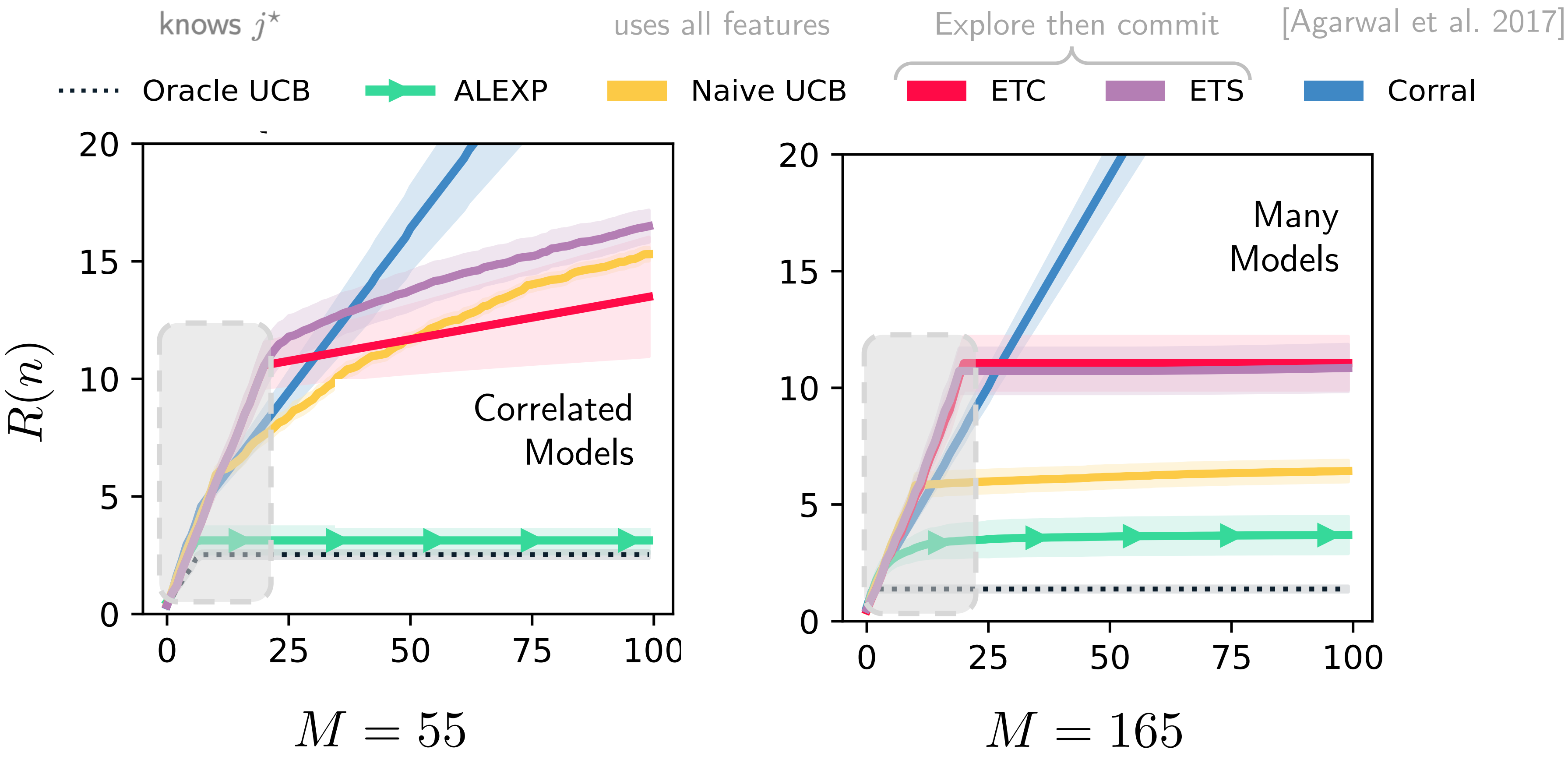
In this context, regret w.r.t. to agent j roughly is..

$$R(T, j) \leq \frac{\log M}{\eta} + \eta \sum_{t=1}^{T} \mathrm{Var}^2(\hat{\boldsymbol{\theta}}_t) + \sum_{t=1}^{T} \mathrm{Bias}(\hat{\boldsymbol{\theta}}_t)$$

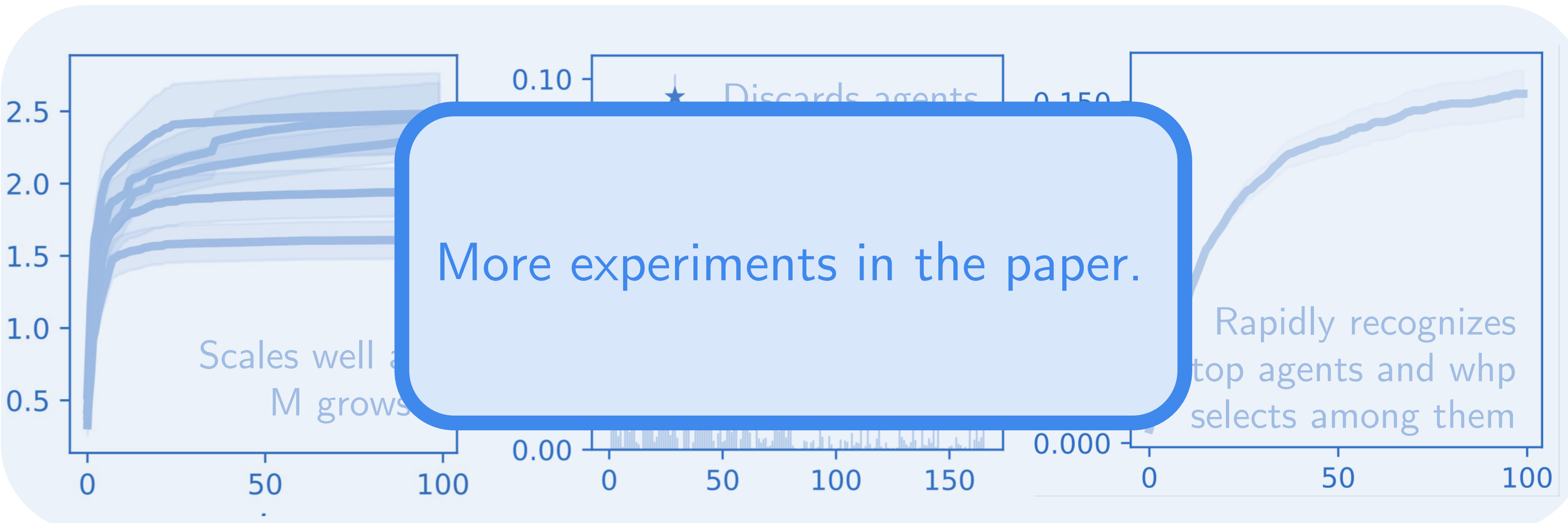
| $\hat{\boldsymbol{\theta}}_t \in \mathbb{R}^{dM}$ | $\mathrm{Var}^2(\hat{\boldsymbol{\theta}}_t)$ | $\mathrm{Bias}(\hat{\boldsymbol{\theta}}_t)$ |
| --- | --- | --- |
| OLS/IW | $M$ | $0$ |
| Lasso | $\log M$ | $\log M$ |

# Model Selection for Optimistic algorithms

data generation & baselines
described in the paper.



If I am running out of time:




More experiments in the paper.

If not...

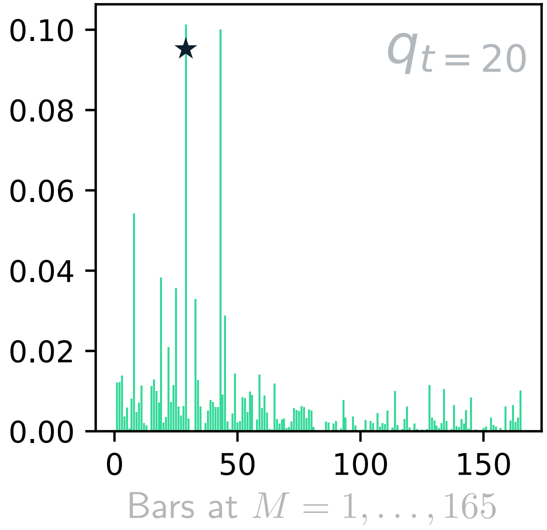
# Model Selection Dynamics of ALExp

Let's see how things evolve turing training...

# Model Selection Dynamics of ALExp

Let's see how things evolve turing training...

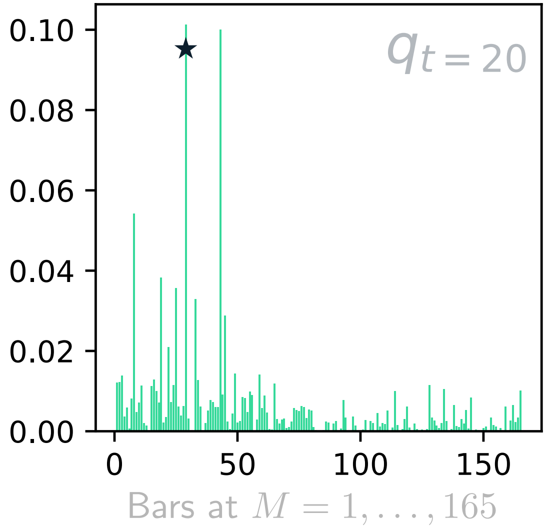
Distribution over the models
at time t=20



# Model Selection Dynamics of ALExp

Let's see how things evolve turing training...

Distribution over the models
at time t=20

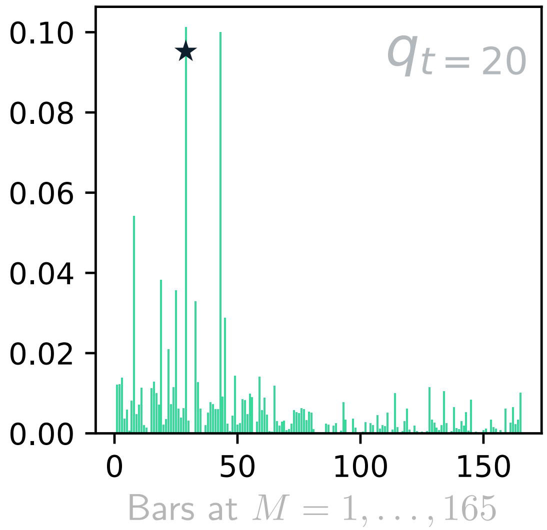



Discards agents without
having queried them

# Model Selection Dynamics of ALExp

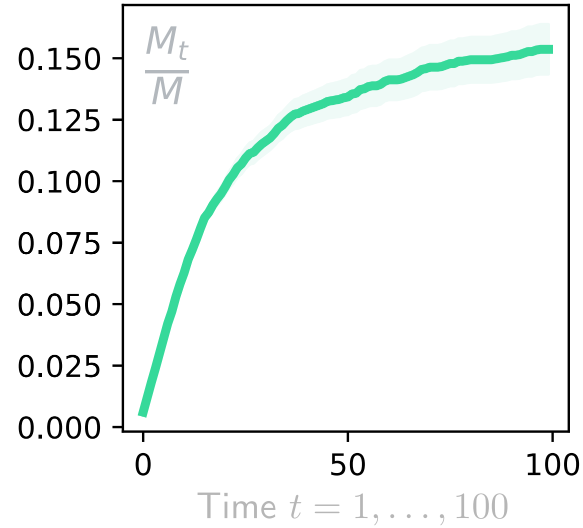
Let's see how things evolve turing training...

Distribution over the models
at time t=20




Discards agents without
having queried them

Number of visited agents
Total number of agents




Rapidly recognizes top agents
and whp selects among them

# What's left open?

1. Is exploration necessary for model selection?

$$\gamma_t \sim t^{-1/4}$$

**Algorithm 1** ALEXP

Inputs: $\gamma_t$, $\eta_t$, $\lambda_t$ 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 $j = 1, \dots, M$.
  Calculate $\hat{\boldsymbol{\theta}}_t \leftarrow \text{Lasso}(H_t, \lambda_t)$ and estimate rewards
  Update selection distribution

connected to lowerbounds on min-eigenvals of covariance matrix

some new results: *pure* exploration is not necessary.

# What's left open?

1. Is exploration necessary for model selection?

2. For what other model classes (efficient) model selection is possible?

Linear ✔

$$\{\boldsymbol{\phi}_j : \mathbb{R}^{d_0} \to \mathbb{R}^d,\ j = 1, \dots, M\}$$

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

# What's left open?

1. Is exploration necessary for model selection?

2. For what other model classes (efficient) model selection is possible?

Linear ✔

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

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

Blackbox Class of size M?

$\mathrm{Poly}(M)$ lower bound?

Infinite class with bounded eluder dimension?

$\log \tilde{d}$ upper bound?

**PK**, Nicolas Emmenegger, AK, and Aldo Pacchiano. "Anytime Model Selection in Linear Bandits." NeurIPS, 2023.

# Thank you!

# 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 $t \geq 1$*

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]

Variance & bias are both $\log M$

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)

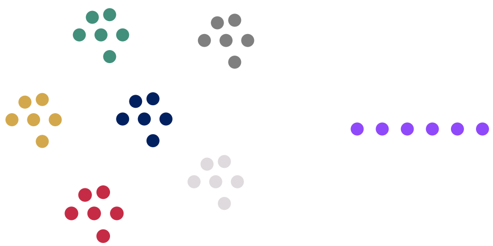
Apply a "stitched" time uniform boundary [Howard et al. '21]



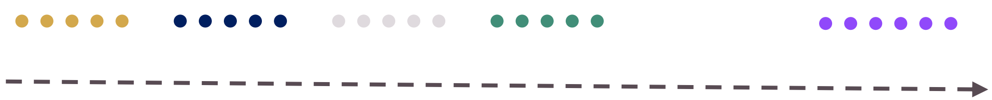
Thanks to Aaditya & SAVI friends!

# We consider 3 scenarios of increasing diffculty

1. Offline Data from similar tasks is available [KRK 2022]



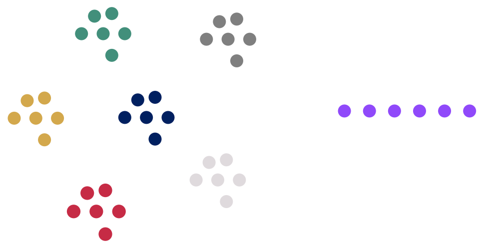
2. Online data from similar tasks can be available [SKRK 2023]



3. No data from similar tasks is available [KPEK 2023]

# Meta-Model Selection: Offline

When offline data from similar tasks is available,



$$y_{s,i} = r_s(\boldsymbol{x}_{s,i}) + \varepsilon_{s,i} \qquad i = 1, \dots, n \text{ and } s = 1, \dots, m$$

$$r_s(\cdot) = \sum_{j=1}^{M} \langle \boldsymbol{\theta}_s^{(j)}, \boldsymbol{\phi}_j(\cdot) \rangle \qquad J \text{ is shared}$$

Classical feature selection with Lasso

$$\hat{\boldsymbol{\theta}}^{(1)}, \dots, \hat{\boldsymbol{\theta}}^{(M)} = \arg\min \frac{1}{mn} \|\boldsymbol{y} - \sum_{j=1}^{M} \Phi_j \boldsymbol{\theta}^{(j)}\|_2^2 + \lambda \sum_{j=1}^{M} ||\boldsymbol{\theta}^{(j)}||_2$$

$$\hat{J} = \{j \in [M] \text{ s.t. } \hat{\boldsymbol{\theta}}^{(j)} > \omega\}$$

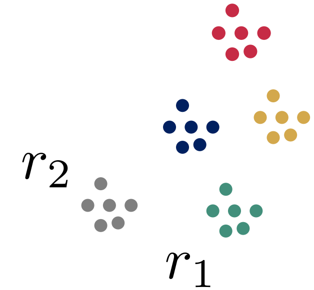
Solving the online optimization problem using the learned model

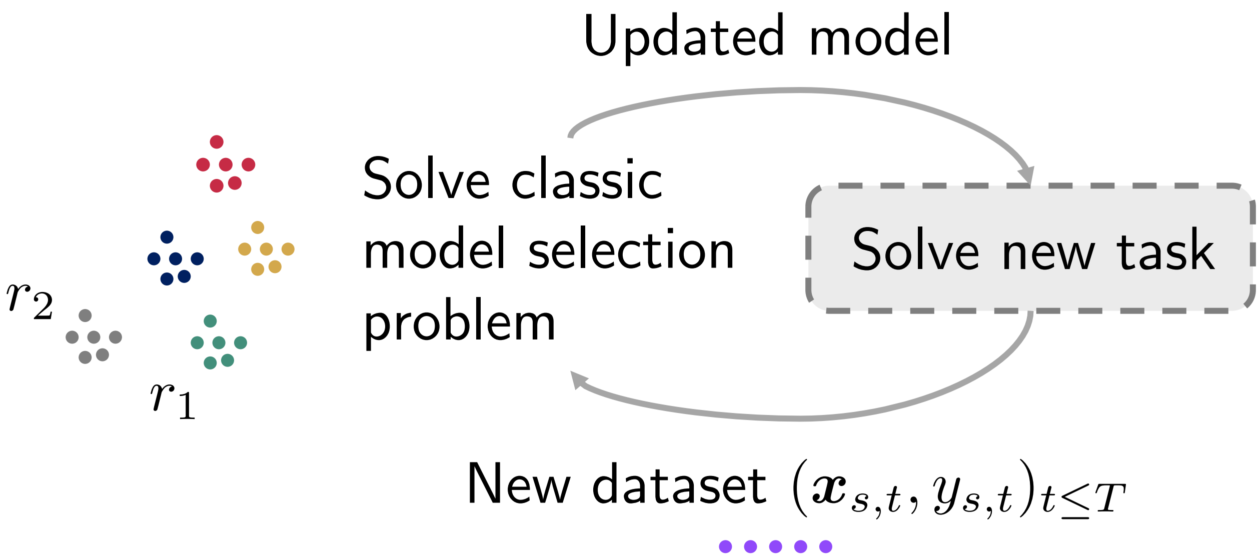
# Meta Model Selection: Lifelong

$\forall s \geq 1 : r_s \in \mathcal{H}$

Suppose the bandit task is of repetitive nature,

Optimizing for different molecular properties
Recommending products to different costumers



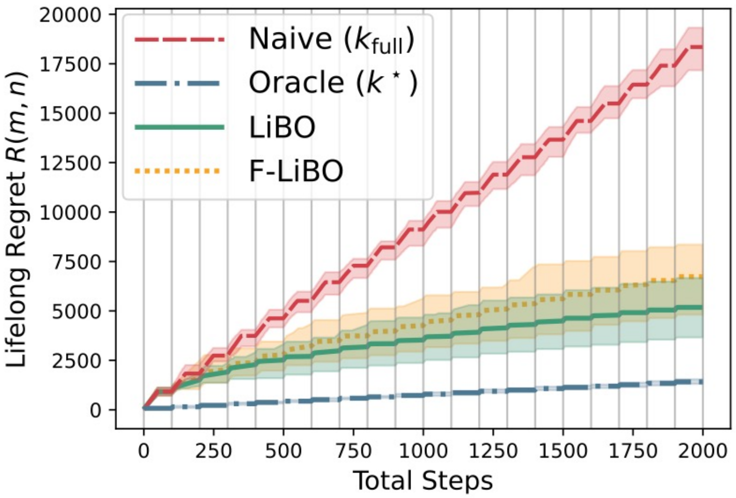



**Theorem (Lifelong Model Selection)**

*Under mild assumptions on the meta-data, and for an appropriate choice of $\lambda$, w.h.p.*

- *$\hat{J}$ is a consistent estimator of $J$,*
- *The optimization algorithm which uses $\hat{J}$ achieves oracle performance $R^\star(T, m)$,*

*as $m$ grows.*




the regret converges at a $\mathcal{O}(\log M/\sqrt{m})$ rate

$$R(T, m) = \sum_{s=1}^{m} \sum_{t=1}^{T} r_s(\boldsymbol{x}_s^\star) - r_s(\boldsymbol{x}_{s,t})$$