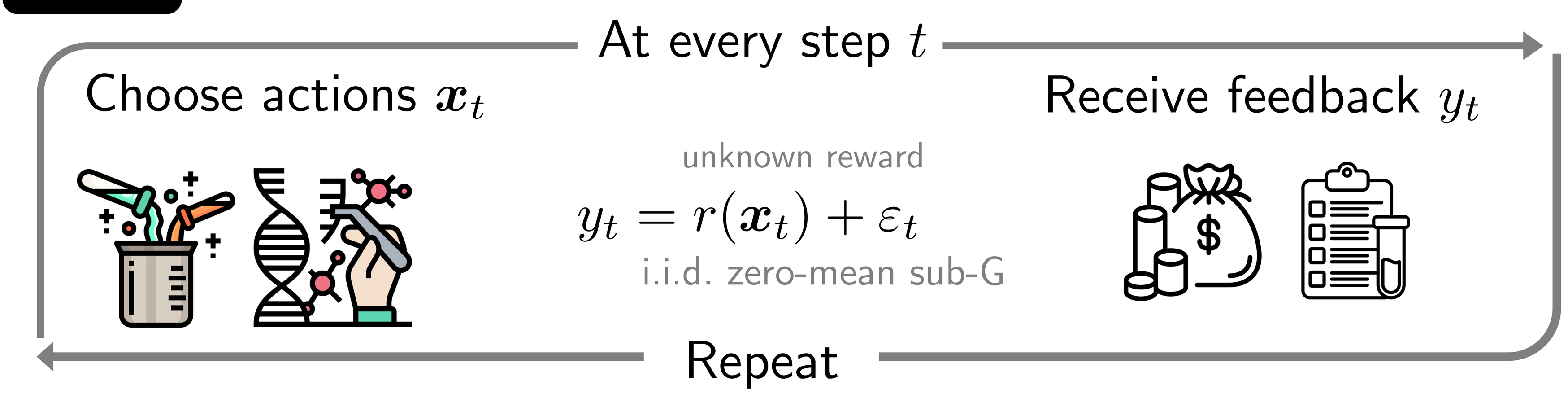
Anytime Model Selection in Linear Bandits

Parnian Kassraie, Nicolas Emmenegger, Andreas Krause, Aldo Pacchiano

ETH zürich

Intro



- The statistical modeling of the reward function plays a crucial role in efficiency of bandit algorithms -- they maintain an estimate of the target function, and use it to choose the next action.
- It is not known a priori which model is going to yield the most sample efficient algorithm, and we can only select the right model as we gather empirical evidence.
- Online Model Selection is not fun and games. $\boldsymbol{x}_t \in \mathcal{X} \subset \mathbb{R}^{d_0}$

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

Reward maximization → not so diverse sample

History dependence → non-i.i.d sample

Can we perform adaptive model selection, while simultaneously optimizing for a reward? Can we be sample-efficient & anytime?

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

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

$$M \gg T$$

+ typical regularity assumptions

- Online Model Selection problem

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

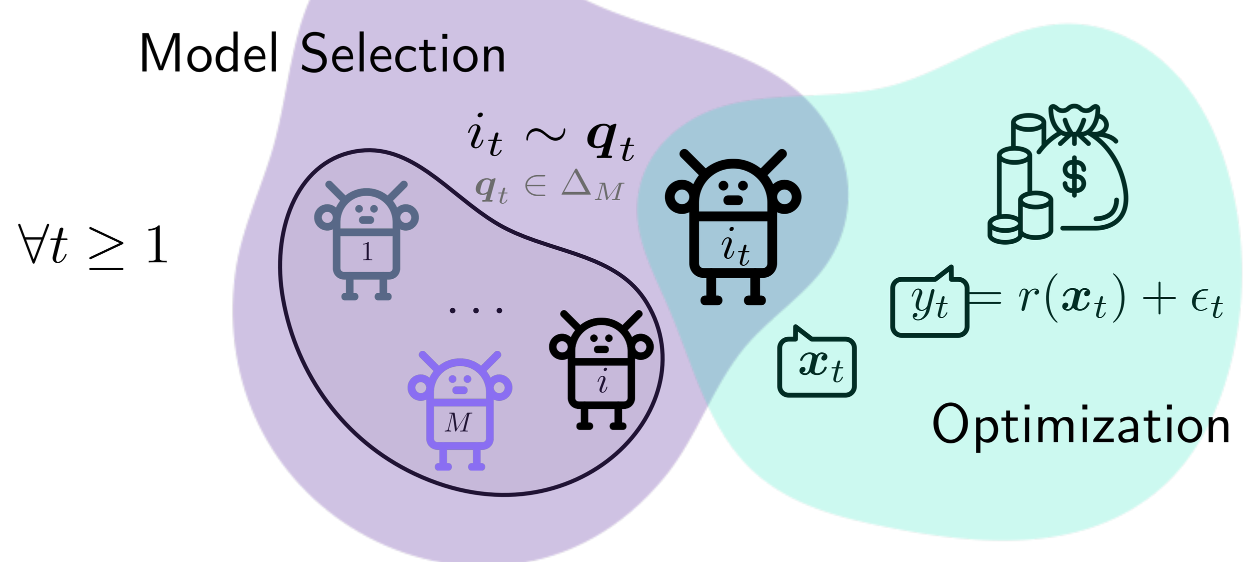
$$R(T) = \sum_{t=1}^{T} r(\boldsymbol{x}^\star) - r(\boldsymbol{x}_t)$$

– Sublinear in T

– $\log M$

Approach

- Probabilistic Aggregation: Instantiate M algorithm each using a different $\boldsymbol{\phi}_j$ to model the reward. Randomly iterate over them.



- With probability $q_{t,j}$ choose agent j and let them choose an action according to their action selection policy $p_{t,j} \in \mathcal{M}(\mathcal{X})$

Main Results

- This gives ALEXP: **A**nytime **Ex**ponential weighting algorithm with **L**asso reward estimates

log M regret	MS guarantee	adaptive & anytime
✓	✓	✓

Algorithm 1 ALEXP

Inputs: γ_t, η_t, λ_t for $t \geq 1$

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

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

  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 $\hat{r}_{t,j}$

  Update selection distribution $q_{t+1,j}$

end for

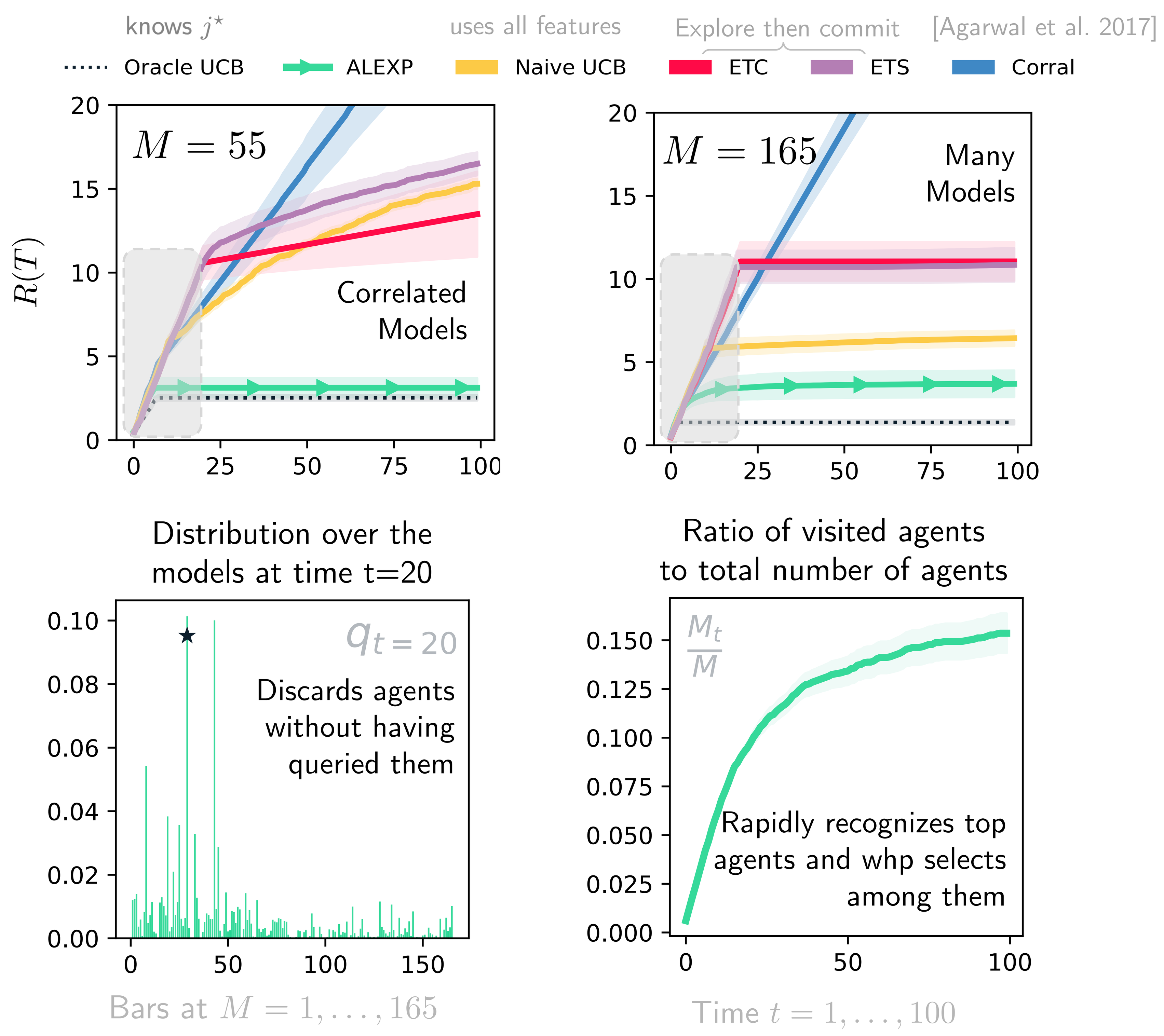
Theorem (Regret - Informal)

For appropriate choices of parameters, [prescribed in the paper]

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

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

Empirical Insights



Ingredient I: Exponential Weights Updates

Increase $q_{t,j}$ if the the agent *seems* to be lucrative

Estimate of the reward obtained by agent j so far

$$q_{t,j} = \frac{\exp\left(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,j}\right)}{\sum_{i=1}^{M} \exp\left(\eta_t \sum_{s=1}^{t-1} \hat{r}_{s,i}\right)} \qquad \hat{r}_{t,j} = \mathbb{E}_{\boldsymbol{x}\sim p_{t,j}} \hat{\boldsymbol{\theta}}_t^\top \boldsymbol{\phi}(\boldsymbol{x})$$

sensitivity of updates

- This technique is known to yield $\log M$ regret in full-info setting, when all $r_{t,j}$ are known. But now, the regret will depend on the bias and variance of $\hat{\boldsymbol{\theta}}_t$
- Typical online regression oracles are $\sqrt{M}$ → poly$M$ regret

Ingredient II: Sparse Online Regression Oracle

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 Lasso Conf Seq)

If for all $t \geq 1$

Bias and variance are both $\log M$

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

cost of going 'time uniform'

then,

Restricted Eigenvalue property

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

image source: flaticon