Resource Allocation in Multi-armed Bandits

Kirthevasan Kandasamy

UC Berkeley

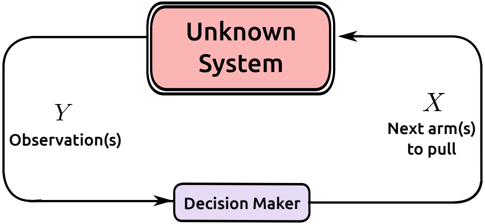
Joint work with: Brijen Thananjeyan, Ion Stoica, Michael I Jordan,
Ken Goldberg, and Joseph E Gonzalez

(More collaborators at the end)

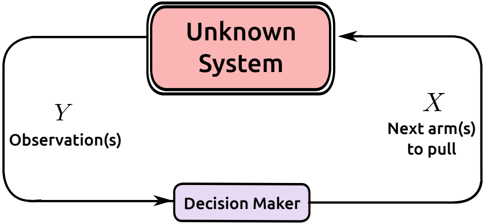
June 9, 2021

Google Research

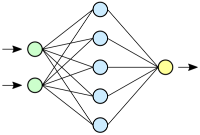
Multi-armed Bandits: optimise an unknown system



Multi-armed Bandits: optimise an unknown system

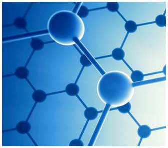


Applications in optimising

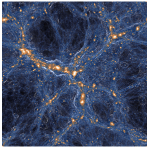


Hyperparameters of ML models

Distributed systems

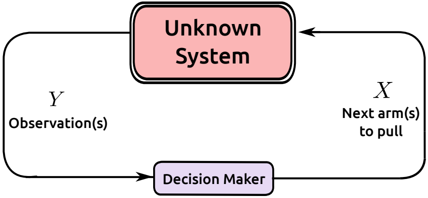


Materials & drugs



Physics simulations

Bandit problems are usually formulated in terms of the number of arm pulls (sample complexity)

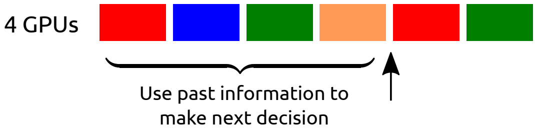


Examples:

- Cumulative regret after T pulls: $R_T = \sum_{t=1}^{T} (\mu^\star - \mu_t)$
- Best arm identification: Minimise number of pulls to identify the optimal arm $\mu^\star$ with probability at least $1 - \delta$.

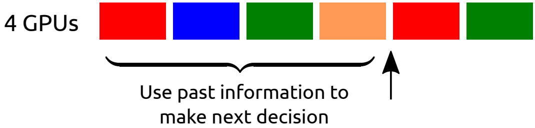
Bandit problems are usually formulated in terms of the number of arm pulls (sample complexity)

Sequential

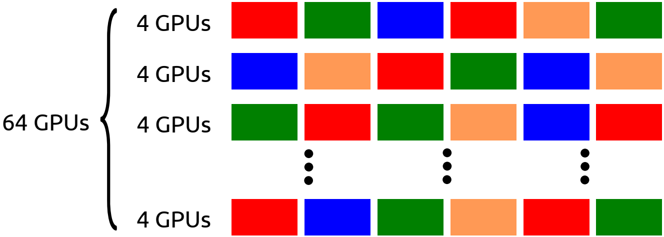


Bandit problems are usually formulated in terms of the number of arm pulls (sample complexity)

Sequential



Batch parallel



Bandit problems are usually formulated in terms of the number of arm pulls (sample complexity)

Usual assumption:

$$\begin{aligned} &\text{number of arm pulls (samples collected)} \\ &\quad \approx \text{ resource consumption} \\ &\quad\quad = \text{ cost, time, ethical concerns etc.} \end{aligned}$$

Bandit problems are usually formulated in terms of the number of arm pulls (sample complexity)

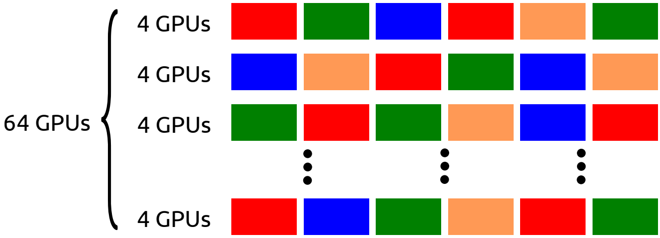
Usual assumption:

$$
\begin{aligned}
&\text{number of arm pulls (samples collected)} \\
&\quad \approx \ \text{resource consumption} \\
&\qquad = \ \text{cost, time, ethical concerns etc.}
\end{aligned}
$$

We usually care about time and cost, and often, the number of samples does translate to these considerations under practical constraints.

- Formulating bandit problems directly in terms of time and resource constraints give rise to new algorithms, theory, and better empirical results.

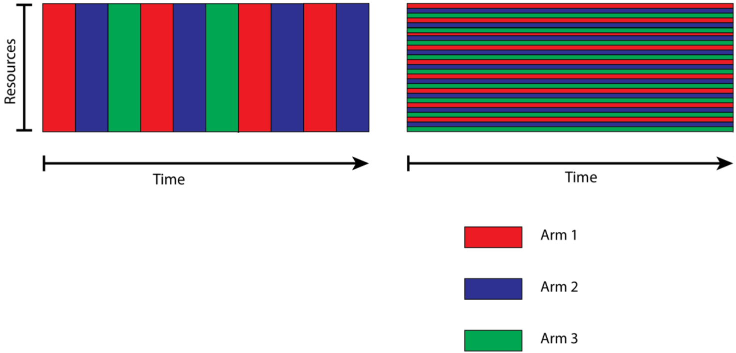
Setting 1: Systems with sublinear scaling



What is the best way to allocate the GPUs to different hyperparameters to identify the optimal hyperparameter in the shortest time?

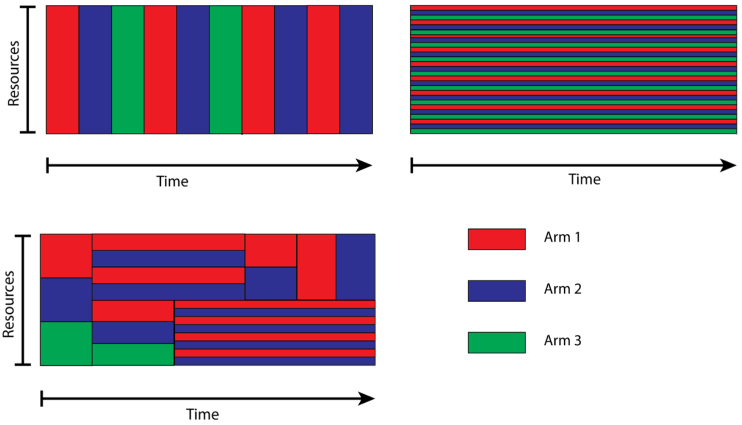
Setting 1: Systems with sublinear scaling

Given a set of resources, what is the best way to allocate them to arm pulls to identify the optimal arm in the shortest possible time?



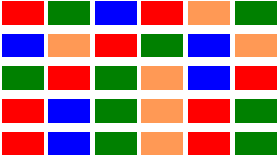
Setting 1: Systems with sublinear scaling

Given a set of resources, what is the best way to allocate them to arm pulls to identify the optimal arm in the shortest possible time?



Setting 2: Elastic resources

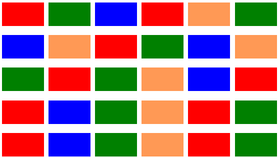
Setting 2: Elastic resources



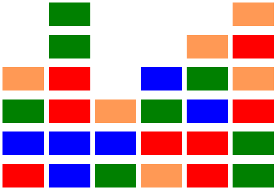
Fixed amount of resources

Setting 2: Elastic resources

Google Cloud aws Azure IBM Cloud

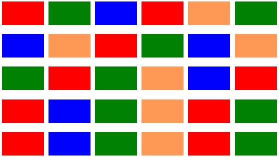


Fixed amount of resources

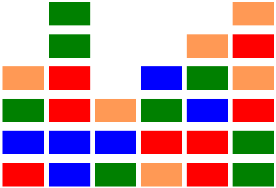


Elastic resources

Setting 2: Elastic resources



Fixed amount of resources



Elastic resources

Other natural applications outside of cloud-based experiments.

Outline

1. Review: Best Arm Identification (BAI)

2. BAI with fixed amount of resources & sublinear scaling
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *Resource Allocation in Multi-armed Bandit Exploration: Overcoming Sublinear Scaling with Adaptive Parallelism*, ICML 2021

3. BAI with elastic resources
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *PAC Best Arm Identification under a Deadline*, Under review

4. Hyperparameter tuning with elastic resources
   - Misra, Liaw, Dunlap, Bhardwaj, Kandasamy, Gonzalez, Stoica, Tumanov, *Rubberband: Cloud-based Hyperparameter Tuning*, EuroSys 2021
   - Dunlap, Misra, Liaw, Kandasamy, Gonzalez, Jordan, Stoica, *Rubberband: Hyperparameter Tuning on the Cloud*, Under review

Outline

1. Review: Best Arm Identification (BAI)

2. BAI with fixed amount of resources & sublinear scaling
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *Resource Allocation in Multi-armed Bandit Exploration: Overcoming Sublinear Scaling with Adaptive Parallelism*, ICML 2021

3. BAI with elastic resources
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *PAC Best Arm Identification under a Deadline*, Under review

4. Hyperparameter tuning with elastic resources
   - Misra, Liaw, Dunlap, Bhardwaj, Kandasamy, Gonzalez, Stoica, Tumanov, *Rubberband: Cloud-based Hyperparameter Tuning*, EuroSys 2021
   - Dunlap, Misra, Liaw, Kandasamy, Gonzalez, Jordan, Stoica, *Rubberband: Hyperparameter Tuning on the Cloud*, Under review

Best Arm Identification

- Given a bandit problem $\nu = (\nu_1, \ldots, \nu_n)$ of n arms.
- When we pull arm i, you observe a *reward* from a distribution ν_i. Arm i has mean $\mathbb{E}_{X \sim \nu_i}[X] = \mu_i$.
  - Each arm is a candidate/configuration to be tested. Pulling an arm corresponds to an experiment.
- Let $\mu_{(1)} > \mu_{(2)} \geq \mu_{(3)} \geq \cdots \geq \mu_{(n)}$. Arm (1) is the *best arm*.

Best Arm Identification

- Given a bandit problem $\nu = (\nu_1, \ldots, \nu_n)$ of n arms.
- When we pull arm i, you observe a *reward* from a distribution ν_i. Arm i has mean $\mathbb{E}_{X \sim \nu_i}[X] = \mu_i$.
  - Each arm is a candidate/configuration to be tested. Pulling an arm corresponds to an experiment.
- Let $\mu_{(1)} > \mu_{(2)} \geq \mu_{(3)} \geq \cdots \geq \mu_{(n)}$. Arm (1) is the *best arm*.
- **BAI:** Given $\delta \in (0, 1)$, identify the best arm with probability at least $1 - \delta$, by pulling arms to collect information about their mean values.
  - Keep the number of arm pulls to a minimum.

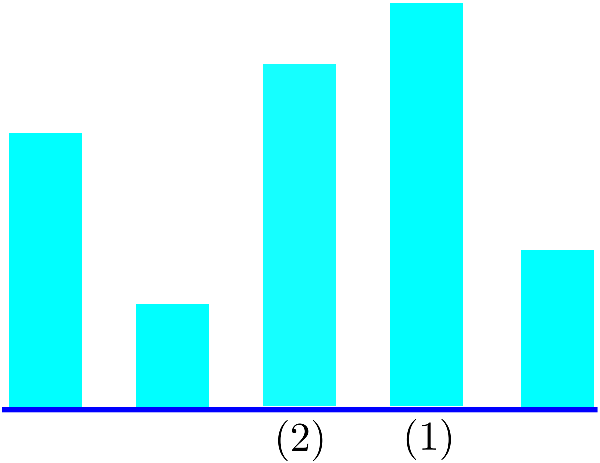
Best Arm Identification

- Given a bandit problem $\nu = (\nu_1, \ldots, \nu_n)$ of n arms.
- When we pull arm i, you observe a *reward* from a distribution ν_i. Arm i has mean $\mathbb{E}_{X \sim \nu_i}[X] = \mu_i$.
  - Each arm is a candidate/configuration to be tested. Pulling an arm corresponds to an experiment.
- Let $\mu_{(1)} > \mu_{(2)} \geq \mu_{(3)} \geq \cdots \geq \mu_{(n)}$. Arm (1) is the *best arm*.
- **BAI:** Given $\delta \in (0, 1)$, identify the best arm with probability at least $1 - \delta$, by pulling arms to collect information about their mean values.
  - Keep the number of arm pulls to a minimum.
- (ϵ, δ)-**PAC BAI:** Given $\epsilon > 0$ and $\delta \in (0, 1)$, identify an ϵ optimal arm with probability at least $1 - \delta$.

$$\text{an } \epsilon\text{-optimal arm } i: \quad \mu_i > \mu_{(1)} - \epsilon$$

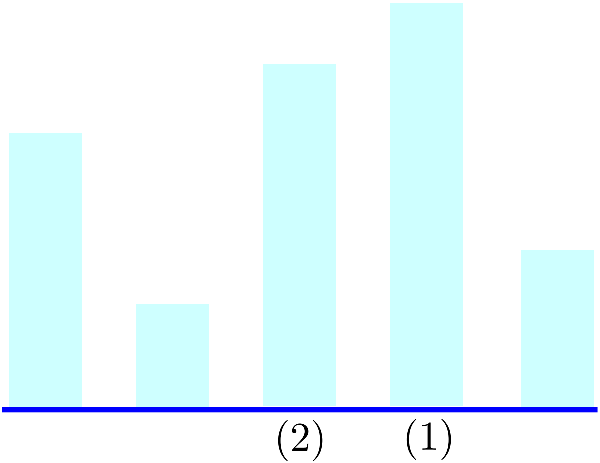
Racing Algorithms for BAI

(Maron & Moore 1993, Kaufmann & Kalyanakrishnan 2013, Jun et al. 2016)



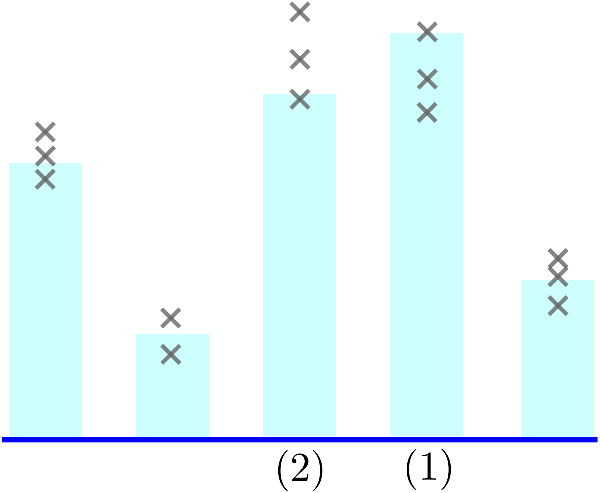
Racing Algorithms for BAI

(Maron & Moore 1993, Kaufmann & Kalyanakrishnan 2013, Jun et al. 2016)



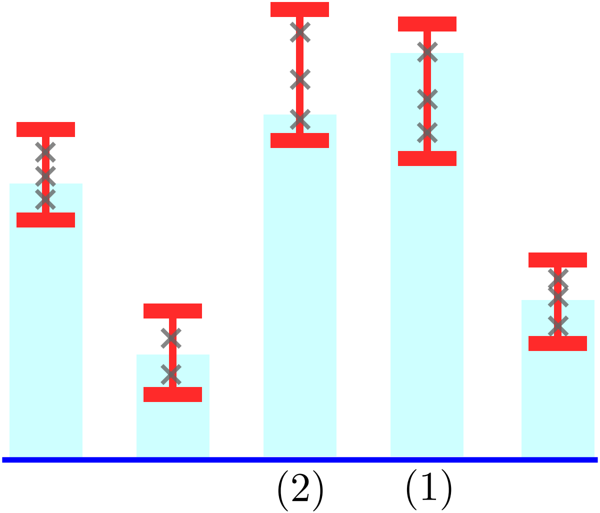
Racing Algorithms for BAI

(Maron & Moore 1993, Kaufmann & Kalyanakrishnan 2013, Jun et al. 2016)



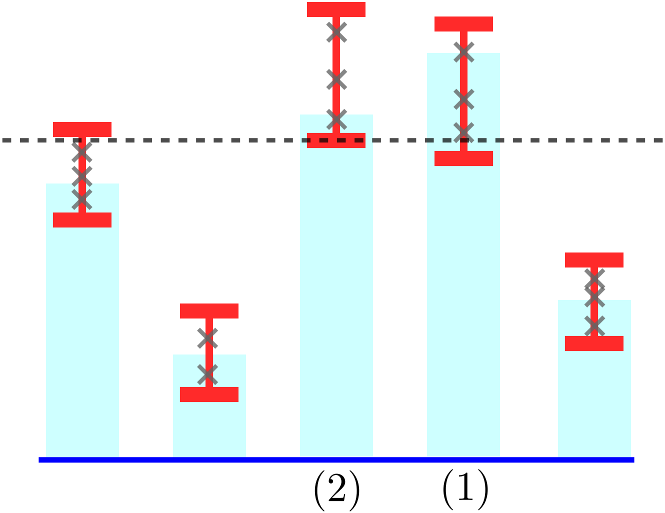
Racing Algorithms for BAI

(Maron & Moore 1993, Kaufmann & Kalyanakrishnan 2013, Jun et al. 2016)



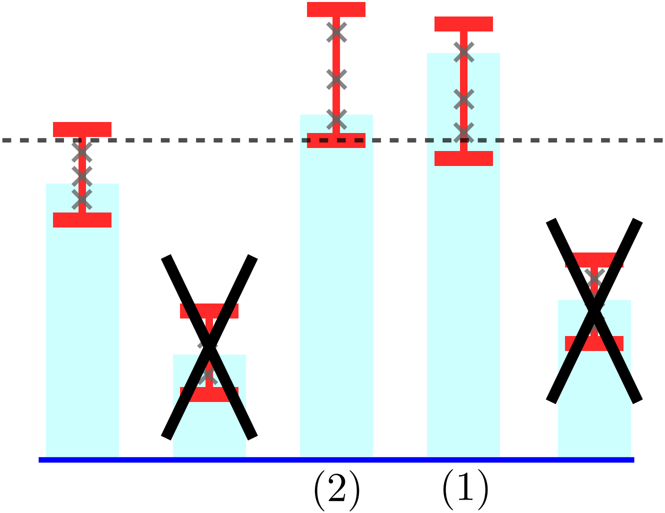
Racing Algorithms for BAI

(Maron & Moore 1993, Kaufmann & Kalyanakrishnan 2013, Jun et al. 2016)



Racing Algorithms for BAI

(Maron & Moore 1993, Kaufmann & Kalyanakrishnan 2013, Jun et al. 2016)



Theoretical Results

Theorem (Informal) *: Number of samples required for BAI,

$$\# \text{ samples} \in \tilde{\Theta}\left(\mathcal{H}(\nu) \cdot \log\left(\frac{1}{\delta}\right)\right)$$

$$\mathcal{H}(\nu) = \sum_{i=1}^{n} \frac{1}{\Delta_i^2}\,, \qquad \text{where} \;\; \Delta_i = \begin{cases} \mu_{(1)} - \mu_{(2)} & \text{if } i = (1), \\ \mu_{(1)} - \mu_i & \text{otherwise.} \end{cases}$$

* *Hiding some details and caveats.*

Theoretical Results

Theorem (Informal) *: Number of samples required for BAI,

$$\#\ \text{samples} \in \tilde{\Theta}\left(\mathcal{H}(\nu) \cdot \log\left(\frac{1}{\delta}\right)\right)$$

$$\mathcal{H}(\nu) = \sum_{i=1}^{n} \frac{1}{\Delta_i^2}\ , \qquad \text{where}\ \ \Delta_i = \begin{cases} \mu_{(1)} - \mu_{(2)} & \text{if } i = (1), \\ \mu_{(1)} - \mu_i & \text{otherwise.} \end{cases}$$

For (ϵ, δ)-PAC BAI,

$$\mathcal{H}(\nu) = \sum_{i=1}^{n} \frac{1}{(\Delta_i + \epsilon)^2}\ ,$$

* *Hiding some details and caveats.*

Outline

1. Review: Best Arm Identification (BAI)

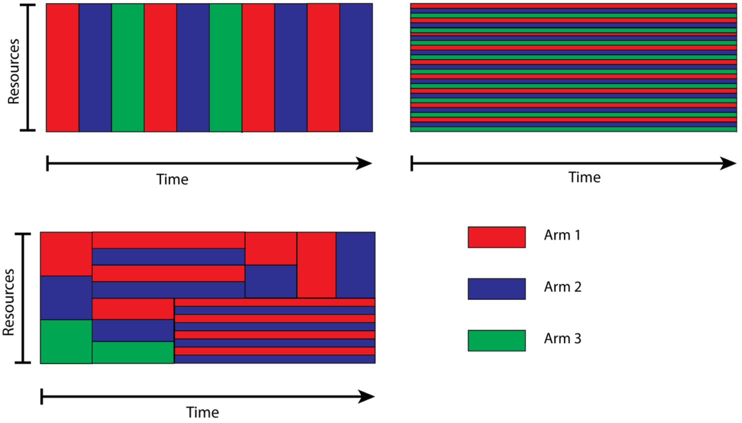
2. BAI with fixed amount of resources & sublinear scaling
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *Resource Allocation in Multi-armed Bandit Exploration: Overcoming Sublinear Scaling with Adaptive Parallelism*, ICML 2021

3. BAI with elastic resources
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *PAC Best Arm Identification under a Deadline*, Under review

4. Hyperparameter tuning with elastic resources
   - Misra, Liaw, Dunlap, Bhardwaj, Kandasamy, Gonzalez, Stoica, Tumanov, *Rubberband: Cloud-based Hyperparameter Tuning*, EuroSys 2021
   - Dunlap, Misra, Liaw, Kandasamy, Gonzalez, Jordan, Stoica, *Rubberband: Hyperparameter Tuning on the Cloud*, Under review

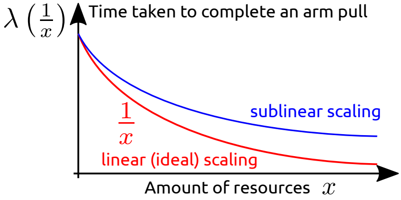
BAI with a fixed amount of resources & sublinear scaling

Given a set of resources, what is the best way to allocate them to arm pulls to identify the best arm in the shortest possible time?



BAI with a fixed amount of resources & sublinear scaling

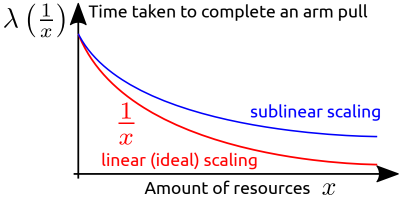
Tradeoffs: Parallelising arm pulls do not scale linearly (due to communication, synchronisation costs etc.)



- More resources per arm pull: arm pull finishes faster, and we can use that information to make better subsequent decisions. However resource usage is inefficient.
- Fewer resources per arm pull: vice versa.

BAI with a fixed amount of resources & sublinear scaling

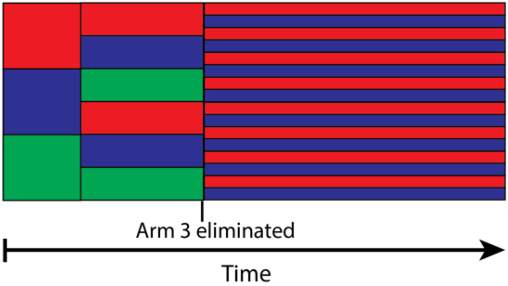
Tradeoffs: Parallelising arm pulls do not scale linearly (due to communication, synchronisation costs etc.)



- More resources per arm pull: arm pull finishes faster, and we can use that information to make better subsequent decisions. However resource usage is inefficient.
- Fewer resources per arm pull: vice versa.

Tradeoff between *information accumulation* and *resource efficiency*.

Adaptive Parallel Racing (APR)



- Racing style: pull arms on each round, and at the end of the round, eliminate arms based on confidence intervals.
- Increase number of pulls for surviving arms in the next round. Rate at which the number of pulls are increased depends on the scaling function λ.

Theoretical Results

Theorem (Informal): Number of samples required for BAI with a fixed amount of resources and sublinear scaling,

$$\#\text{ samples} \in \tilde{\Theta}\left(\mathcal{T} \cdot \log\left(\frac{1}{\delta}\right)\right)$$

$\mathcal{T} = \mathcal{T}_2\left(\{\Delta_i^{-2}\}_{i=2}^{n}\right)$ is given via the following dynamic program,

$$\mathcal{T}_j\left(\{z_i\}_{i=j}^{n}\right) = \min_{k \in \{j,\ldots,n\}} \Big(\lambda(k(z_j - z_{k+1})) + \mathcal{T}_{k+1}\left(\{z_i\}_{i=k+1}^{n}\right)\Big).$$

Theoretical Results

Theorem (Informal): Number of samples required for BAI with a fixed amount of resources and sublinear scaling,

$$\# \text{ samples} \in \tilde{\Theta}\left(\mathcal{T} \cdot \log\left(\frac{1}{\delta}\right)\right)$$

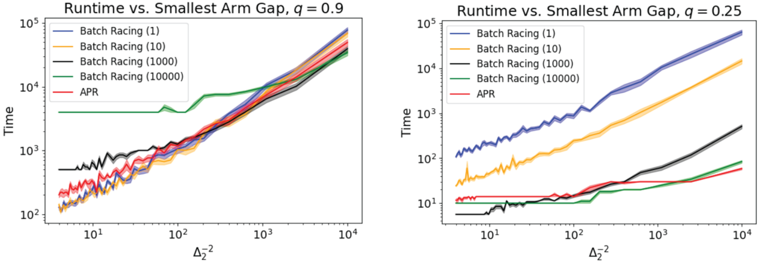
$\mathcal{T} = \mathcal{T}_2\left(\{\Delta_i^{-2}\}_{i=2}^n\right)$ is given via the following dynamic program,

$$\mathcal{T}_j\left(\{z_i\}_{i=j}^n\right) = \min_{k \in \{j,\ldots,n\}} \left(\lambda(k(z_j - z_{k+1})) + \mathcal{T}_{k+1}\left(\{z_i\}_{i=k+1}^n\right)\right).$$

- Upper bound: via our algorithm APR.
- Lower bound: integrates the scaling funtion into hypothesis testing techniques.

Simulations

Used a scaling function of the form, $\lambda\left(\frac{1}{x}\right) \propto \frac{1}{x^q}$, for $q \in (0, 1]$.
When q is large, the scaling is good.



APR is able to match the best fixed amount of parallelization for a problem without knowledge of the gap values.

Outline

1. Review: Best Arm Identification (BAI)

2. BAI with fixed amount of resources & sublinear scaling
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *Resource Allocation in Multi-armed Bandit Exploration: Overcoming Sublinear Scaling with Adaptive Parallelism*, ICML 2021

3. BAI with elastic resources
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *PAC Best Arm Identification under a Deadline*, Under review

4. Hyperparameter tuning with elastic resources
   - Misra, Liaw, Dunlap, Bhardwaj, Kandasamy, Gonzalez, Stoica, Tumanov, *Rubberband: Cloud-based Hyperparameter Tuning*, EuroSys 2021
   - Dunlap, Misra, Liaw, Kandasamy, Gonzalez, Jordan, Stoica, *Rubberband: Hyperparameter Tuning on the Cloud*, Under review

BAI with elastic resources

Current literature on (ϵ, δ)-PAC BAI:

- Sequential setting: pull arms one at a time.
  $\#$ samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu) \cdot \log\left(\frac{1}{\delta}\right)\right)$, where,

$$\mathcal{H}(\nu) = \sum_{i=1}^{n} (\Delta_i + \epsilon)^{-2}, \quad \text{where} \quad \Delta_i = \begin{cases} \mu_{(1)} - \mu_{(2)} & \text{if } i = (1), \\ \mu_{(1)} - \mu_{(i)} & \text{otherwise.} \end{cases}$$

BAI with elastic resources

Current literature on (ϵ, δ)-PAC BAI:

- Sequential setting: pull arms one at a time.
  $\#$ samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu) \cdot \log\left(\frac{1}{\delta}\right)\right)$, where,

$$\mathcal{H}(\nu) = \sum_{i=1}^{n} (\Delta_i + \epsilon)^{-2}, \quad \text{where} \quad \Delta_i = \begin{cases} \mu_{(1)} - \mu_{(2)} & \text{if } i = (1), \\ \mu_{(1)} - \mu_{(i)} & \text{otherwise.} \end{cases}$$

- Passive setting: Pull all arms at one go and identify the best arm. $\#$ samples $\in \tilde{\Theta}\left(n\epsilon^{-2} \log\left(\frac{1}{\delta}\right)\right)$.

BAI with elastic resources

Current literature on (ϵ, δ)-PAC BAI:

- Sequential setting: pull arms one at a time.
  $\#$ samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu) \cdot \log\left(\frac{1}{\delta}\right)\right)$, where,

$$\mathcal{H}(\nu) = \sum_{i=1}^{n} (\Delta_i + \epsilon)^{-2}, \quad \text{where} \quad \Delta_i = \begin{cases} \mu_{(1)} - \mu_{(2)} & \text{if } i = (1), \\ \mu_{(1)} - \mu_{(i)} & \text{otherwise.} \end{cases}$$

- Passive setting: Pull all arms at one go and identify the best arm. $\#$ samples $\in \tilde{\Theta}\left(n\epsilon^{-2} \log\left(\frac{1}{\delta}\right)\right)$.

Sequential algorithms can adapt to problem difficulty, and have better sample complexity.

BAI with elastic resources

Current literature on (ϵ, δ)-PAC BAI:

- Sequential setting: pull arms one at a time.
  \# samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu) \cdot \log\left(\frac{1}{\delta}\right)\right)$, where,

$$\mathcal{H}(\nu) = \sum_{i=1}^{n} (\Delta_i + \epsilon)^{-2}, \quad \text{where} \quad \Delta_i = \begin{cases} \mu_{(1)} - \mu_{(2)} & \text{if } i = (1), \\ \mu_{(1)} - \mu_{(i)} & \text{otherwise.} \end{cases}$$

- Passive setting: Pull all arms at one go and identify the best arm. \# samples $\in \tilde{\Theta}\left(n\epsilon^{-2} \log\left(\frac{1}{\delta}\right)\right)$.

Sequential algorithms can adapt to problem difficulty, and have better sample complexity.

- **Our work:** Given a deadline of T rounds.
  Can execute multiple arm pulls per round.

BAI with elastic resources

Current literature on (ϵ, δ)-PAC BAI:

- Sequential setting: pull arms one at a time.
  \# samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu) \cdot \log\left(\frac{1}{\delta}\right)\right)$, where,

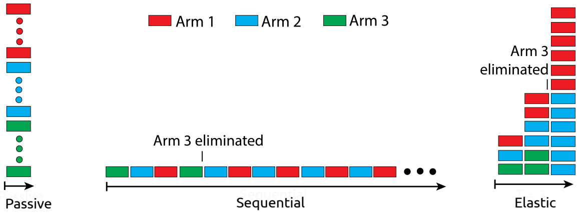
$$\mathcal{H}(\nu) = \sum_{i=1}^{n} (\Delta_i + \epsilon)^{-2}, \quad \text{where} \quad \Delta_i = \begin{cases} \mu_{(1)} - \mu_{(2)} & \text{if } i = (1), \\ \mu_{(1)} - \mu_{(i)} & \text{otherwise.} \end{cases}$$

- Passive setting: Pull all arms at one go and identify the best arm. \# samples $\in \tilde{\Theta}\left(n\epsilon^{-2} \log\left(\frac{1}{\delta}\right)\right)$.

Sequential algorithms can adapt to problem difficulty, and have better sample complexity.

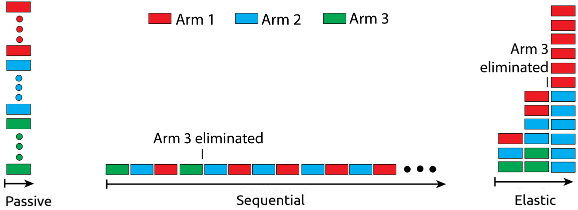
- **Our work:** Given a deadline of T rounds.
  Can execute multiple arm pulls per round.
  (**Prior work:** Agarwal et al 2017, Jin et al 2019)

BAI with elastic resources



- Passive ($T = 1$): requires a lot of samples even on easy problems.
- Sequential ($T = \infty$): small sample complexity as it can adapt to the problem, but can take a long time.

BAI with elastic resources



- Passive ($T = 1$): requires a lot of samples even on easy problems.
- Sequential ($T = \infty$): small sample complexity as it can adapt to the problem, but can take a long time.
- $1 < T < \infty$: "best of both worlds" – short time, intermediate sample complexity.

Applications for elastic bandits

- Cloud-based experiments for hyperparameter tuning, scientific simulations etc.
- Clinical trials: identify the best treatment/vaccine among multiple candidates over a few rounds trials.
  - Reduce number of trials to reduce costs and for ethical reasons.
- Laboratory experiments: high-throughput experimentation platforms can be used to conduct several trials at a time.
  - Reduce number of experiments to reduce the cost of reagents.

Applications for elastic bandits

- Cloud-based experiments for hyperparameter tuning, scientific simulations etc.
- Clinical trials: identify the best treatment/vaccine among multiple candidates over a few rounds trials.
  - Reduce number of trials to reduce costs and for ethical reasons.
- Laboratory experiments: high-throughput experimentation platforms can be used to conduct several trials at a time.
  - Reduce number of experiments to reduce the cost of reagents.

Standard assumption of a single resource or a fixed amount of parallel resources often does not hold true in practice.

Elastic Batch Racing (EBR)

- Racing style: pull arms on each round, and at the end of the round, eliminate arms based on confidence intervals.
- Pull each surviving arm $\mathcal{O}(\epsilon^{-2t/T})$ times on round t.
- If an arm survives until round T, it gets pulled $\mathcal{O}(\epsilon^{-2})$ times.

Theoretical results 1: minimax optimality over subclasses

- Passive setting: $\#\text{samples} \in \tilde{\Theta}\left(n\epsilon^{-2}\log\left(\delta^{-1}\right)\right)$.
- Sequential setting: $\#$ samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu)\cdot\log\left(\delta^{-1}\right)\right)$, where, $\mathcal{H}(\nu) = \sum_{i=1}^{n}\left(\Delta_i + \epsilon\right)^{-2}$.

Theoretical results 1: minimax optimality over subclasses

- Passive setting: #samples $\in \tilde{\Theta}\left(n\epsilon^{-2}\log\left(\delta^{-1}\right)\right)$.
- Sequential setting: # samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu)\cdot\log\left(\delta^{-1}\right)\right)$, where, $\mathcal{H}(\nu) = \sum_{i=1}^{n}\left(\Delta_i + \epsilon\right)^{-2}$.

How well can we adapt to problem difficulty with only limited rounds of adaptivity?

We partition all the problems $\mathcal{P} = \bigcup_{\gamma\in\{1,\ldots,T\}^n}\mathcal{P}_\gamma$ based on their $\{\Delta_i\}_i$ values. Here, $\mathcal{P}_\gamma = \{\nu; \gamma(\nu) = \gamma\}$, where,

$$\gamma_i(\nu) = \begin{cases} 1 & \text{if} \quad \Delta_i \in [\epsilon^{\frac{1}{T}}, 1] \\ k & \text{if} \quad \Delta_i \in [\epsilon^{\frac{k}{T}}, \epsilon^{\frac{k-1}{T}}) \text{ for k } \in \{2,\ldots,T-1\}\,, \\ T & \text{if} \quad \Delta_i \in [0, \epsilon^{\frac{T-1}{T}}). \end{cases}$$

When γ is “small” (e.g. $\gamma = [1,\ldots,1]$), the problems in $\mathcal{P}_\gamma$ are “easy” (i.e Δ_i values are large).

Theoretical results 1: minimax optimality over subclasses

We partition all the problems $\mathcal{P} = \bigcup_{\gamma \in \{1,\dots,T\}^n} \mathcal{P}_\gamma$ based on their $\{\Delta_i\}_i$ values.

Upper bound 1: EBR on problem $\nu \in \mathcal{P}_\gamma$,

$$\# \text{ samples} \in \tilde{\mathcal{O}}\left(\sup_{\nu' \in \mathcal{P}_\gamma} \mathcal{H}(\nu') \cdot \log(\delta^{-1}) \right)$$

Theoretical results 1: minimax optimality over subclasses

We partition all the problems $\mathcal{P} = \bigcup_{\gamma \in \{1,\dots,T\}^n} \mathcal{P}_\gamma$ based on their $\{\Delta_i\}_i$ values.

Upper bound 1: EBR on problem $\nu \in \mathcal{P}_\gamma$,

$$\#\,\text{samples} \in \tilde{\mathcal{O}}\left(\sup_{\nu' \in \mathcal{P}_\gamma} \mathcal{H}(\nu') \cdot \log(\delta^{-1})\right)$$

Lower bound 1:

$$\inf_{A \in (\epsilon,\delta)-\text{PAC}} \sup_{\nu' \in \mathcal{P}_\gamma} \#\,\text{samples} \in \tilde{\Omega}\left(\sup_{\nu' \in \mathcal{P}_\gamma} \mathcal{H}(\nu') \cdot \log(\delta^{-1})\right)$$

Theoretical results 2: relative to sequential setting

- Passive setting: $\#\text{samples} \in \tilde{\Theta}\left(n\epsilon^{-2}\log\left(\delta^{-1}\right)\right)$.
- Sequential setting: $\#$ samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu)\cdot\log\left(\delta^{-1}\right)\right)$, where, $\mathcal{H}(\nu) = \sum_{i=1}^{n}\left(\Delta_i + \epsilon\right)^{-2}$.

Theoretical results 2: relative to sequential setting

- Passive setting: #samples $\in \tilde{\Theta}\left(n\epsilon^{-2}\log\left(\delta^{-1}\right)\right)$.
- Sequential setting: # samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu)\cdot\log\left(\delta^{-1}\right)\right)$, where, $\mathcal{H}(\nu)=\sum_{i=1}^{n}\left(\Delta_i+\epsilon\right)^{-2}$.

Upper bound 2: EBR on any problem $\nu\in\mathcal{P}$,

$$\#\text{ samples } \in \tilde{\mathcal{O}}\left(\epsilon^{-2/T}\cdot\mathcal{H}(\nu)\cdot\log(\delta^{-1})\right)$$

Theoretical results 2: relative to sequential setting

- Passive setting: #samples $\in \tilde{\Theta}\left(n\epsilon^{-2}\log\left(\delta^{-1}\right)\right)$.
- Sequential setting: # samples $\in \tilde{\Theta}\left(\mathcal{H}(\nu)\cdot\log\left(\delta^{-1}\right)\right)$, where, $\mathcal{H}(\nu) = \sum_{i=1}^{n}\left(\Delta_i + \epsilon\right)^{-2}$.

Upper bound 2: EBR on any problem $\nu \in \mathcal{P}$,

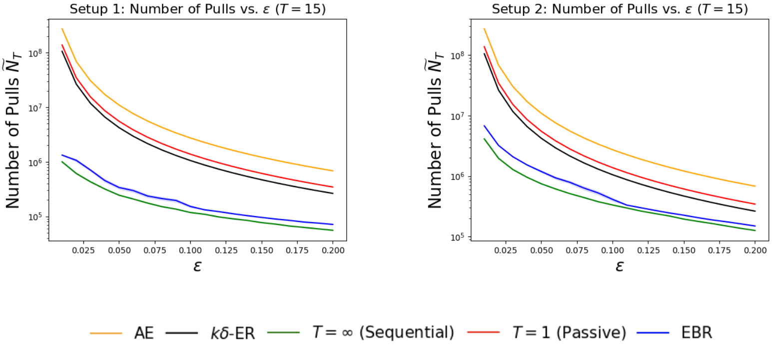
$$\#\text{ samples } \in \tilde{\mathcal{O}}\left(\epsilon^{-2/T}\cdot\mathcal{H}(\nu)\cdot\log(\delta^{-1})\right)$$

Lower bound 2 *:

$$\inf_{A\in(\epsilon,\delta)-\text{PAC}}\ \sup_{\nu\in\mathcal{P}}\ \frac{\#\text{ samples}}{\mathcal{H}(\nu)} \in \tilde{\Omega}\left(\epsilon^{-2/T}\right)$$

* *With some caveats.*

Simulations



Outline

1. Review: Best Arm Identification (BAI)

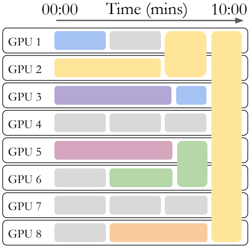
2. BAI with fixed amount of resources & sublinear scaling
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *Resource Allocation in Multi-armed Bandit Exploration: Overcoming Sublinear Scaling with Adaptive Parallelism*, ICML 2021

3. BAI with elastic resources
   - Thananjeyan, Kandasamy, Stoica, Jordan, Goldberg, Gonzalez, *PAC Best Arm Identification under a Deadline*, Under review

4. Hyperparameter tuning with elastic resources
   - Misra, Liaw, Dunlap, Bhardwaj, Kandasamy, Gonzalez, Stoica, Tumanov, *Rubberband: Cloud-based Hyperparameter Tuning*, EuroSys 2021
   - Dunlap, Misra, Liaw, Kandasamy, Gonzalez, Jordan, Stoica, *Rubberband: Hyperparameter Tuning on the Cloud*, Under review

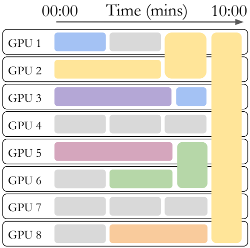
Hyperparameter Tuning with Elastic Resources

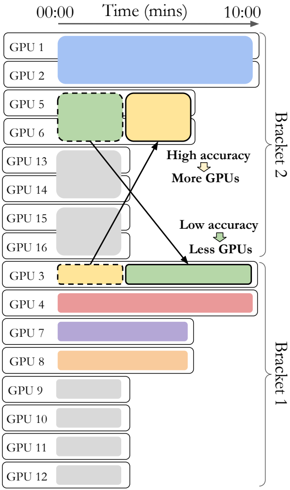
Goal: identify the optimal hyperparameters for a machine learning model under a deadline and resource-time budget.



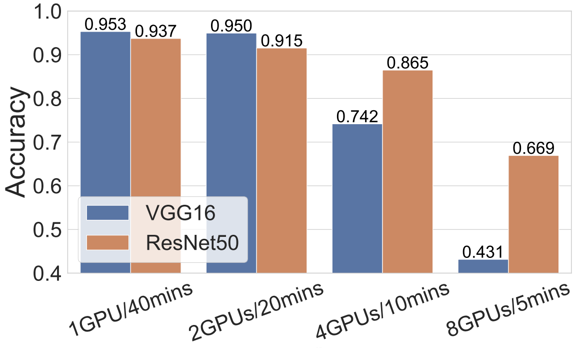
Hyperparameter Tuning with Elastic Resources

Goal: identify the optimal hyperparameters for a machine learning model under a deadline and resource-time budget.





Elasticity helps circumvent undesirable effects of sublinear scaling



Experiments

Image classification

Method	Model	Dataset	Deadline	GPU minutes	Accuracy	Std-Error
Random	VGG16	SVHN	15	4×15	0.19	0.028
ASHA	VGG16	SVHN	15	4×15	0.819	0.053
HyperSched	VGG16	SVHN	15	4×15	0.927	0.021
BOHB	VGG16	SVHN	15	4×15	0.458	0.086
E-Hyperband	VGG16	SVHN	15	4×15	0.921	0.015
E-Grid Search	VGG16	SVHN	15	4×15	0.944	0.010
SEER	VGG16	SVHN	15	4×15	**0.956**	**0.005**
Random	ResNet18	CIFAR10	60	16×60	0.226	0.101
ASHA	ResNet18	CIFAR10	60	16×60	0.896	0.006
HyperSched	ResNet18	CIFAR10	60	16×60	0.932	0.005
BOHB	ResNet18	CIFAR10	60	16×60	0.864	0.000
E-Hyperband	ResNet18	CIFAR10	60	16×60	0.914	0.005
E-Grid Search	ResNet18	CIFAR10	60	16×60	0.904	0.001
SEER	ResNet18	CIFAR10	60	16×60	**0.935**	**0.001**
Random	ResNet50	TinyImageNet	60	16×60	0.091	0.064
ASHA	ResNet50	TinyImageNet	60	16×60	0.212	0.068
HyperSched	ResNet50	TinyImageNet	60	16×60	0.581	0.019
BOHB	ResNet50	TinyImageNet	60	16×60	0.110	0.055
E-Hyperband	ResNet50	TinyImageNet	60	16×60	0.630	0.003
E-Grid Search	ResNet50	TinyImageNet	60	16×60	0.632	0.049
SEER	ResNet50	TinyImageNet	60	16×60	**0.675**	**0.001**

ASHA and Hypersched are implementations of Hyperband on a fixed amount of resources. E-Hyperband and E-Grid search are (naive) elastic variants of Hyperband and grid search.

Experiments

Text classification

Method	M/MM Accuracy	Std-Error
Random	0.651 / 0.657	0.078 / 0.098
ASHA	0.837 / 0.831	0.002 / 0.001
HyperSched	0.834 / 0.837	0.001 / 0.001
BOHB	0.814 / 0.817	0.001 / 0.000
E-Hyperband	0.836 / 0.831	0.002 / 0.001
E-Grid Search	0.833 / 0.815	0.001 / 0.002
SEER	**0.839 / 0.840**	**0.001 / 0.002**

Image segmentation

Method	Mean-IOU	Std-Error
Random	0.413	0.078
ASHA	0.519	0.005
HyperSched	0.524	0.061
BOHB	0.474	0.078
E-Hyperband	0.503	0.003
E-Grid Search	0.524	0.006
SEER	**0.541**	**0.008**

ASHA and Hypersched are implementations of Hyperband on a fixed amount of resources. E-Hyperband and E-Grid search are (naive) elastic variants of Hyperband and grid search.

Summary

- Usually, BAI problems are formulated in terms of the number of pulls (sample complexity).
- Carefully considering resource constraints that arise in practice gives rise to interesting algorithms and theory, and better empirical results.
  - Setting 1: a fixed amount of parallel resoures, but with sublinear scaling.
  - Setting 2: Elastic resources. As we are increasingly running jobs on the cloud, we are no longer constrained by a fixed pool of resources.

Best Arm Identification


Brijen T


Ion S


Joey G


Ken G


Mike J

Elastic Hyperparameter Tuning


Alexey T


Ion S


Joey G


Lisa D


Mike J


Richard L


Romil B


Ujval M

Thank You