Pure Exploration Problems

Wouter M. Koolen
CWI and University of Twente

MML, June 20, 2024

Part I

Motivating Problems

I'll show you three motivating applications:

- A/B Testing
- Self-driving
- Solving Games (MCTS)

I'll show you three motivating applications:

- A/B Testing
- Self-driving
- Solving Games (MCTS)

Here is what is common:

- The learner needs to identify some aspect of reality
- The learner can choose which **data** to collect

A/B Testing

Motivating question

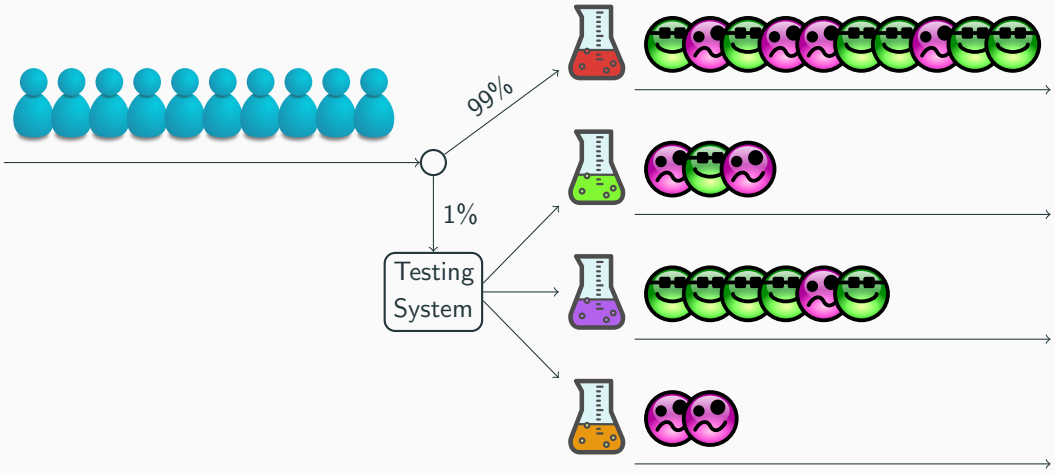
Better to switch from current system  to new version  or  or  ?

Motivating question

Better to switch from current system  to new version  or  or  ?

- A/B testing
- Adaptive clinical trial
- Best arm identification

Let's find out in production!



Best version?

Self-Driving

Motivating question

We are excited about autonomous driving.

Motivating question

We are excited about autonomous driving.

Training AI systems (e.g. deep neural networks) takes a lot of data.

Motivating question

We are excited about autonomous driving.

Training AI systems (e.g. deep neural networks) takes a lot of data.

Yet not all data are equally valuable/useful.

Motivating question

We are excited about autonomous driving.

Training AI systems (e.g. deep neural networks) takes a lot of data.

Yet not all data are equally valuable/useful.

Let's optimise and automate the data collection.

Motivating question

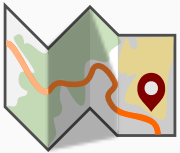
We are excited about autonomous driving.

Training AI systems (e.g. deep neural networks) takes a lot of data.

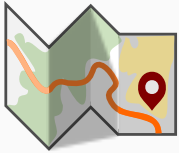
Yet not all data are equally valuable/useful.

Let's optimise and automate the data collection.

Where should I send my prototype for training? to find the best driving strategy?



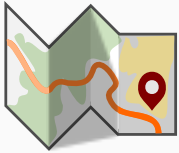
Dragons everywhere



Where should I send my prototype for training to find the best driving strategy?

- AI system is huge parameterised model
- Lots of possible environments to drive in
- Multiple objectives (safety, efficiency, . . .)
- Feedback (crash/intervention) is very one-sided

Dragons everywhere



Where should I send my prototype for training to find the best driving strategy?

- AI system is huge parameterised model
- Lots of possible environments to drive in
- Multiple objectives (safety, efficiency, . . .)
- Feedback (crash/intervention) is very one-sided

Distilled goal:

- Identify Overall safest: fewest crashes in natural environment mix.

Brutal simplification

Close-by parameters in close-by environments result in close-by outcomes

Brutal simplification

Close-by parameters in close-by environments result in close-by outcomes

Often true, but too complicated. Let's discretise!

Brutal simplification

Close-by parameters in close-by environments result in close-by outcomes

Often true, but too complicated. Let's discretise!

Environments

Brutal simplification

Close-by parameters in close-by environments result in close-by outcomes

Often true, but too complicated. Let's discretise!

Environments

Parameters

Brutal simplification

Close-by parameters in close-by environments result in close-by outcomes

Often true, but too complicated. Let's discretise!

Environments

Parameters

Outcomes

World model (Stochastic Bandit)

The world simplifies to vector + table:

World model (Stochastic Bandit)

The world simplifies to vector + table:

Known natural environment mix

$$\mathbb{P}\left(\text{  }\right) = 40\%$$

$$\mathbb{P}\left(\text{  }\right) = 30\%$$

$$\mathbb{P}\left(\text{  }\right) = 15\%$$

$$\mathbb{P}\left(\text{  }\right) = 10\%$$

$$\mathbb{P}\left(\text{  }\right) = 5\%$$

World model (Stochastic Bandit)

The world simplifies to vector + table:

Known natural environment mix

$$\mathbb{P}\left(\text{  }\right) = 40\%$$

$$\mathbb{P}\left(\text{  }\right) = 30\%$$

$$\mathbb{P}\left(\text{  }\right) = 15\%$$

$$\mathbb{P}\left(\text{  }\right) = 10\%$$

$$\mathbb{P}\left(\text{  }\right) = 5\%$$

Unknown crash probabilities

$$\mathbb{P}\left(\text{  }\right) = 0.1\%$$

$$\mathbb{P}\left(\text{  }\right) = 0.3\%$$

$$\mathbb{P}\left(\text{  }\right) = 0.03\%$$

$$\vdots$$

$$\mathbb{P}\left(\text{  }\right) = 2.7\%$$

World model (Stochastic Bandit)

The world simplifies to vector + table:

Known natural environment mix

$\mathbb{P}$() = 40%

$\mathbb{P}$() = 30%

$\mathbb{P}$() = 15%

$\mathbb{P}$() = 10%

$\mathbb{P}$() = 5%

Unknown crash probabilities

$\mathbb{P}$() = 0.1%

$\mathbb{P}$() = 0.3%

$\mathbb{P}$() = 0.03%

$\vdots$

$\mathbb{P}$() = 2.7%

Together these determine the best parameter on average. Say  .

Learning the best Parameter by Driving: Example Interaction

Overall safest?

Learning the best Parameter by Driving: Example Interaction

1   

Overall safest?

Learning the best Parameter by Driving: Example Interaction

1   

2   

Overall safest?

Learning the best Parameter by Driving: Example Interaction

1   

2   

3   

Overall safest?

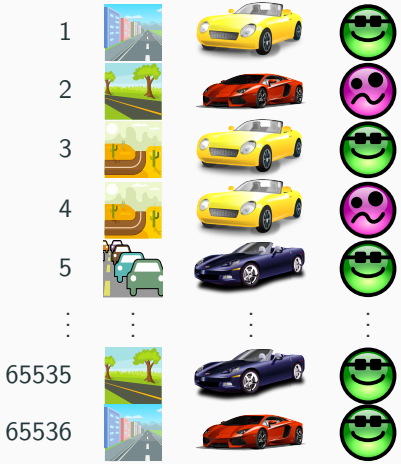
Learning the best Parameter by Driving: Example Interaction

Overall safest?

Learning the best Parameter by Driving: Example Interaction

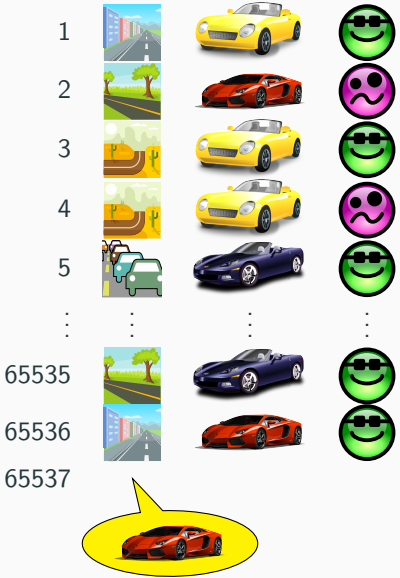
Overall safest?

Learning the best Parameter by Driving: Example Interaction



Overall safest?

Learning the best Parameter by Driving: Example Interaction



Overall safest?

Solving Games (MCTS)

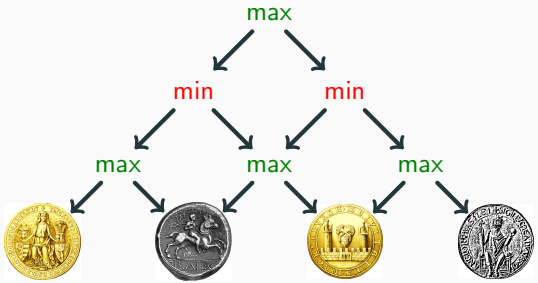
BAI again

Unknown environment

$$\mathbb{P}\left(\text{smile} \mid \text{coin}_1\right) = 1.75\%$$
$$\mathbb{P}\left(\text{smile} \mid \text{coin}_2\right) = 3\%$$
$$\mathbb{P}\left(\text{smile} \mid \text{coin}_3\right) = 2.5\%$$
$$\mathbb{P}\left(\text{smile} \mid \text{coin}_4\right) = 1\%$$

How to find the best arm by sampling arms?

Game DAGs



How to find the best move at the root by sampling leaves?

And Many More

Beyond best arm

Many possible desiderata

- Find the best M arms
- Find all sufficiently good arms
- Arms with combinatorial structure
- Continuously many arms (bandit optimisation)
- Prior knowledge about structure (dosage)
- Personalisation (find a policy: context $\rightarrow$ arm)
- Multi-objective problems
- Risk measures
- Customers return (!) (MDP)

and challenges

- delays
- censoring
- constraints
- non-stationarity
- comparison feedback

Booming Industry within Multi-Armed Bandit / Testing Literature

Explosion of bandit testing papers with analogous problems:

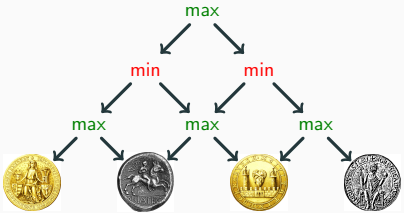
- Top-m 2017; 2017
- Spectral 2021
- Stratified 2021
- Lipschitz 2019
- Linear 2020; 2020
- Threshold 2017
- MaxGap 2019
- Duelling 2021
- Contextual 2020; 2020
- Pareto 2023
- Minimum 2018
- MCTS 2016
- Markov 2019
- Tail-Risk 2021
- MDP 2021

Part II

Theory and Algorithms

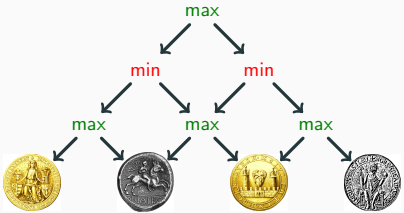
Setup

Setting



K-armed bandit $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$.

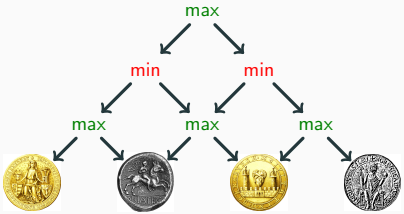
Setting



K-armed bandit $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$.

Each arm k represented by a Bernoulli rate μ_k in $[0, 1]$.

Setting

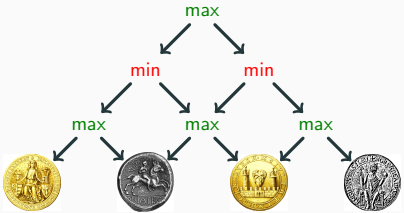


K-armed bandit $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$.

Each arm k represented by a Bernoulli rate μ_k in $[0, 1]$.

Observations from arm k are i.i.d. Bernoulli(μ_k)).

Setting



K-armed bandit $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_K)$.

Each arm k represented by a Bernoulli rate μ_k in $[0, 1]$.

Observations from arm k are i.i.d. Bernoulli(μ_k)).

The **best action at the root** is:

$$i^*(\boldsymbol{\mu}) \;\coloneqq\; \arg\max_i \min_j \max_k \mu_{\mathsf{leaf}(i,j,k)}$$

Protocol

We work in the setting of **fixed confidence** $\delta \in (0, 1)$.

Protocol

For $t = 1, 2, \ldots, \tau$*:*

- *Learner picks an arm* $I_t \in [K]$.
- *Learner sees* $X_t \sim$ *Bernoulli*(μ_{I_t})

Learner recommends child of the root $\hat{\imath}$

Protocol

We work in the setting of **fixed confidence** $\delta \in (0, 1)$.

Protocol

For $t = 1, 2, \ldots, \tau$*:*

- *Learner picks an arm* $I_t \in [K]$.
- *Learner sees* $X_t \sim$ *Bernoulli*(μ_{I_t})

Learner recommends child of the root $\hat{\imath}$

Strategy for Learner specified by

- sampling rule I_t
- stopping rule τ
- recommendation rule $\hat{\imath}$

Objectives

Learner is δ-**correct** if for any bandit instance $\boldsymbol{\mu}$

$$\mathop{\mathbb{P}}_{\boldsymbol{\mu}}\{\tau < \infty \wedge \hat{\imath} \neq i^*(\boldsymbol{\mu})\} \;\leq\; \delta$$

Goal: minimise sample complexity $\mathbb{E}_{\boldsymbol{\mu}}[\tau]$ over all δ-correct strategies.

Sample Complexity Lower Bound

Lower Bounds

Suppose we believe we are in bandit $\boldsymbol{\mu}$.

But we are required to be δ correct in all bandits.

When can we stop and recommend answer $i \equiv i^*(\boldsymbol{\mu})$?

Lower Bounds

Suppose we believe we are in bandit $\boldsymbol{\mu}$.

But we are required to be δ correct in all bandits.

When can we stop and recommend answer $i \equiv i^*(\boldsymbol{\mu})$?

Hypothesis test! Composite null vs point alternative:

$$\mathcal{H}_0 \;=\; \neg i \;:=\; \left\{\boldsymbol{\lambda} \in [0,1]^K \middle| i^*(\boldsymbol{\lambda}) \neq i\right\} \qquad \text{vs} \qquad \mathcal{H}_1 \;=\; \{\boldsymbol{\mu}\}$$

Passive Warm-up

We denote the relative entropy aka **Kullback Leibler divergence** from Bernoulli x to Bernoulli y by $\mathsf{kl}(x, y) = x \ln \frac{x}{y} + (1 - x) \ln \frac{1-x}{1-y}$.

Passive Warm-up

We denote the relative entropy aka **Kullback Leibler divergence** from Bernoulli x to Bernoulli y by $\mathsf{kl}(x, y) = x \ln \frac{x}{y} + (1 - x) \ln \frac{1-x}{1-y}$.

Fix $\boldsymbol{\mu}$ and $\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})$

Fact

Fix the number n_k of samples form each arm k. If the algorithm is δ-correct at time $\tau = \sum_{k=1}^{K} n_k$ for both $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$, then

$$\sum_{k=1}^{K} N_k \, \mathsf{kl}(\mu_k, \lambda_k) \;\geq\; \mathsf{kl}(\delta, 1 - \delta)$$

Passive Warm-up

We denote the relative entropy aka **Kullback Leibler divergence** from Bernoulli x to Bernoulli y by $\mathrm{kl}(x, y) = x \ln \frac{x}{y} + (1 - x) \ln \frac{1-x}{1-y}$.

Fix $\boldsymbol{\mu}$ and $\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})$

Fact

Fix the number n_k of samples form each arm k. If the algorithm is δ-correct at time $\tau = \sum_{k=1}^{K} n_k$ for both $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$, then

$$\sum_{k=1}^{K} N_k \, \mathrm{kl}(\mu_k, \lambda_k) \;\geq\; \mathrm{kl}(\delta, 1 - \delta)$$

Proof by data compression, tensorisation of KL, δ-PAC.

$H \coloneqq (I_1, X_1, \ldots, I_\tau, X_\tau, \hat{\imath})$

Passive Warm-up

We denote the relative entropy aka **Kullback Leibler divergence** from Bernoulli x to Bernoulli y by $\mathrm{kl}(x, y) = x \ln \frac{x}{y} + (1-x) \ln \frac{1-x}{1-y}$.

Fix $\boldsymbol{\mu}$ and $\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})$

Fact

Fix the number n_k of samples form each arm k. If the algorithm is δ-correct at time $\tau = \sum_{k=1}^{K} n_k$ for both $\boldsymbol{\mu}$, $\boldsymbol{\lambda}$, then

$$\sum_{k=1}^{K} N_k \, \mathrm{kl}(\mu_k, \lambda_k) \;\geq\; \mathrm{kl}(\delta, 1-\delta)$$

Proof by data compression, tensorisation of KL, δ-PAC.

$$H \coloneqq (I_1, X_1, \ldots, I_\tau, X_\tau, \hat{\imath}) \qquad \mathrm{KL}\left(\underset{\boldsymbol{\mu}}{\overset{\mathrm{alg}}{\mathbb{P}}}(H) \middle\| \underset{\boldsymbol{\lambda}}{\overset{\mathrm{alg}}{\mathbb{P}}}(H)\right) \;\geq\; \mathrm{kl}\left(\underset{\boldsymbol{\mu}}{\overset{\mathrm{alg}}{\mathbb{P}}}\{\hat{i} \neq i^*(\boldsymbol{\mu})\}, \underset{\boldsymbol{\lambda}}{\overset{\mathrm{alg}}{\mathbb{P}}}\{\hat{i} \neq i^*(\boldsymbol{\mu})\},\right)$$

Active Story

If we follow the sampling and stopping rules, we get on average $\mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]$ samples from arm k at time τ.

Active Story

If we follow the sampling and stopping rules, we get on average $\mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]$ samples from arm k at time τ. With that,

$$\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]\, \mathsf{kl}(\mu_k, \lambda_k) \;\geq\; \mathsf{kl}(\delta, 1-\delta)$$

Active Story

If we follow the sampling and stopping rules, we get on average $\mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]$ samples from arm k at time τ. With that,

$$\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]\, \mathsf{kl}(\mu_k, \lambda_k) \;\geq\; \mathsf{kl}(\delta, 1-\delta)$$

Since this has to hold for all $\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})$, we find

$$\inf_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]\, \mathsf{kl}(\mu_k, \lambda_k) \;\geq\; \mathsf{kl}(\delta, 1-\delta)$$

Active Story

If we follow the sampling and stopping rules, we get on average $\mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]$ samples from arm k at time τ. With that,

$$\sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]\, \mathrm{kl}(\mu_k, \lambda_k) \;\geq\; \mathrm{kl}(\delta, 1-\delta)$$

Since this has to hold for all $\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})$, we find

$$\inf_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{\mu}}[N_k(\tau)]\, \mathrm{kl}(\mu_k, \lambda_k) \;\geq\; \mathrm{kl}(\delta, 1-\delta)$$

Even for the best algorithm, we find

$$\max_{w \in \triangle_K} \inf_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \mathbb{E}_{\boldsymbol{\mu}}[\tau] \sum_{k=1}^{K} w_k\, \mathrm{kl}(\mu_k, \lambda_k) \;\geq\; \mathrm{kl}(\delta, 1-\delta)$$

Lower Bound

Recall that to give answer i we need to reject the composite hypothesis

$$\neg i \;\coloneqq\; \left\{ \boldsymbol{\lambda} \in [0,1]^K \;\middle|\; i^*(\boldsymbol{\lambda}) \neq i \right\}.$$

Lower Bound

Recall that to give answer i we need to reject the composite hypothesis

$$\neg i \;:=\; \left\{\boldsymbol{\lambda} \in [0,1]^K \;\middle|\; i^*(\boldsymbol{\lambda}) \neq i\right\}.$$

Theorem (Garivier and Kaufmann 2016)

Fix a δ-correct strategy. Then for every bandit model $\boldsymbol{\mu}$

$$\mathop{\mathbb{E}}_{\boldsymbol{\mu}}[\tau] \;\geq\; T^*(\boldsymbol{\mu}) \ln \frac{1}{\delta}$$

where the characteristic time $T^*(\boldsymbol{\mu})$ is given by

$$\frac{1}{T^*(\boldsymbol{\mu})} \;=\; \max_{\boldsymbol{w} \in \triangle_K} \min_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \sum_{k=1}^{K} w_k \,\mathrm{kl}(\mu_k, \lambda_k).$$

Example

Unknown environment

$$\mathbb{P}\left(\text{☺}\,|\,\text{🪙}\right) = 1.75\%$$
$$\mathbb{P}\left(\text{☺}\,|\,\text{🪙}\right) = 3\%$$
$$\mathbb{P}\left(\text{☺}\,|\,\text{🪙}\right) = 2.5\%$$
$$\mathbb{P}\left(\text{☺}\,|\,\text{🪙}\right) = 1\%$$

Algorithms

Algorithm

Lower bound gives us more than just sample complexity.

Algorithm

Lower bound gives us more than just sample complexity. We also get the **oracle weight** map

$$\boldsymbol{w}^*(\boldsymbol{\mu}) \;:=\; \max_{w \in \triangle_K} \inf_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \sum_{k=1}^{K} w_k \,\mathsf{kl}(\mu_k, \lambda_k)$$

What if **plug-in** empirical mean $\hat{\boldsymbol{\mu}}(t-1)$ and play arm $I_t \sim \boldsymbol{w}^*\big(\hat{\boldsymbol{\mu}}(t-1)\big)$.

Algorithm

Lower bound gives us more than just sample complexity. We also get the **oracle weight** map

$$\boldsymbol{w}^*(\boldsymbol{\mu}) \;:=\; \max_{w \in \triangle_K} \inf_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \sum_{k=1}^{K} w_k \,\mathsf{kl}(\mu_k, \lambda_k)$$

What if **plug-in** empirical mean $\hat{\boldsymbol{\mu}}(t-1)$ and play arm $I_t \sim \boldsymbol{w}^*\big(\hat{\boldsymbol{\mu}}(t-1)\big)$.

Theorem (Degenne and Koolen, 2019)

Take set-valued interpretation of arg max *defining* $\boldsymbol{w}^*$. *Then* $\boldsymbol{\mu} \mapsto \boldsymbol{w}^*(\boldsymbol{\mu})$ *is upper-hemicontinuous and convex-valued. Suitable tracking ensures that as* $\hat{\boldsymbol{\mu}}(t) \to \boldsymbol{\mu}$, *any sequence of choices* $\boldsymbol{w}_t \in \boldsymbol{w}^*(\hat{\boldsymbol{\mu}}(t-1))$ *has*

$$\min_{\boldsymbol{w} \in \boldsymbol{w}^*(\boldsymbol{\mu})} \|\boldsymbol{w}_t - \boldsymbol{w}\|_\infty \to 0 \quad \text{as } t \to \infty.$$

Chernoff Stopping

How to stop as soon as possible?

Chernoff Stopping

How to stop as soon as possible?

Need confidence region optimised for problem.

Chernoff Stopping

How to stop as soon as possible?

Need confidence region optimised for problem.

Say answer $\hat{\imath}_t$ is correct for $\hat{\boldsymbol{\mu}}(t)$, and we want to conclude that it is correct for $\boldsymbol{\mu}$.

Chernoff Stopping

How to stop as soon as possible?

Need confidence region optimised for problem.

Say answer $\hat{\imath}_t$ is correct for $\hat{\boldsymbol{\mu}}(t)$, and we want to conclude that it is correct for $\boldsymbol{\mu}$.

We need to reject the possibility that any other answer is correct, i.e. the composite hypothesis

$$\neg\hat{\imath}_t \;:=\; \left\{\boldsymbol{\lambda} \;\middle|\; i^*(\lambda_k) \neq \hat{\imath}_t\right\}.$$

Chernoff Stopping

How to stop as soon as possible?

Need confidence region optimised for problem.

Say answer $\hat{\imath}_t$ is correct for $\hat{\boldsymbol{\mu}}(t)$, and we want to conclude that it is correct for $\boldsymbol{\mu}$.

We need to reject the possibility that any other answer is correct, i.e. the composite hypothesis

$$\neg\hat{\imath}_t \;:=\; \left\{\boldsymbol{\lambda} \;\middle|\; i^*(\lambda_k) \neq \hat{\imath}_t\right\}.$$

How to measure the evidence against $\neg\hat{\imath}_t$?

Generalised Likelihood Ratio Test (GLRT) Statistic

$$\Lambda_t \;:=\; \ln \frac{\max_{\boldsymbol{\lambda}\notin\neg\hat{\imath}_t} p(data|\boldsymbol{\lambda})}{\max_{\boldsymbol{\lambda}\in\neg\hat{\imath}_t} p(data|\boldsymbol{\lambda})} \;=\; \min_{\boldsymbol{\lambda}\in\neg\hat{\imath}_t} \sum_{k=1}^{K} N_k(t)\,\mathrm{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right)$$

On working with GLRT

One can show that

Theorem (Kaufmann and Koolen, 2021)

For every bandit $\boldsymbol{\mu}$

$$\mathbb{P}_{\boldsymbol{\mu}}\Big(\exists t : \underbrace{\Lambda_t \geq \ln \frac{1}{\delta} + O(\ln \ln t)}_{\text{GLRT is big}} \text{ and } \underbrace{\hat{\imath}_t \neq i^*(\mu)}_{\text{mistake}} \Big) \leq \delta.$$

Cool, so we can stop when $\Lambda_t \geq \ln \frac{1}{\delta} + O(\ln \ln t)$ and safely output $\hat{\imath}_t$

Obtaining a sampling rule from GLRT

So we can stop when we see a big value in the GLRT

$$\Lambda_t \;=\; \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \sum_{k=1}^{K} N_k(t)\, \mathsf{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right)$$

Obtaining a sampling rule from GLRT

So we can stop when we see a big value in the GLRT

$$\Lambda_t \;=\; \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \; \sum_{k=1}^{K} N_k(t)\, \mathrm{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right)$$

To stop early, the sampling rule should drive Λ_t up.

Obtaining a sampling rule from GLRT

So we can stop when we see a big value in the GLRT

$$\Lambda_t \;=\; \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \sum_{k=1}^{K} N_k(t)\, \mathsf{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right)$$

To stop early, the sampling rule should drive Λ_t up.

The GLRT/round is maximised by sampling with proportions

$$\boldsymbol{w}_t \;\coloneqq\; \operatorname*{arg\,max}_{\boldsymbol{w} \in \triangle_K} \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \sum_{k=1}^{K} w_k\, \mathsf{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right)$$

Again empirical plug-in of oracle weights.

Track and Stop Framework

Definition (Track-and-Stop)

- Stop and output $\hat{\imath}_t$ when $\Lambda_t = \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \sum_{k=1}^{K} N_k(t) \, \mathrm{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right) \geq \ln \frac{1}{\delta} + O(\ln \ln t)$
- Sample with proportions $\boldsymbol{w}_t = \arg\max_{\boldsymbol{w} \in \triangle_K} \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \sum_{k=1}^{K} w_k \, \mathrm{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right)$

Track and Stop Framework

Definition (Track-and-Stop)

- Stop and output $\hat{\imath}_t$ when $\Lambda_t \;=\; \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \sum_{k=1}^{K} N_k(t)\, \mathrm{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right) \;\geq\; \ln \frac{1}{\delta} + O(\ln \ln t)$
- Sample with proportions $\boldsymbol{w}_t \;=\; \arg\max_{\boldsymbol{w} \in \triangle_K} \min_{\boldsymbol{\lambda} \in \neg \hat{\imath}_t} \sum_{k=1}^{K} w_k\, \mathrm{kl}\left(\hat{\mu}_{t,k}, \lambda_k\right)$

Theorem (Garivier and Kaufmann, 2016)

The expected stopping time on any bandit $\boldsymbol{\mu}$ is

$$\mathop{\mathbb{E}}_{\boldsymbol{\mu}}[\tau] \;\leq\; \frac{\ln \frac{1}{\delta}}{\max_{\boldsymbol{w} \in \triangle_K} \; \min_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \; \sum_{k=1}^{K} w_k\, \mathrm{kl}\left(\mu_k, \lambda_k\right)} + \textit{small}$$

and this matches lower bounds.

Upshot of Track-and-Stop

The cost for learning the correct answer is

$$\frac{\ln \frac{1}{\delta}}{\max_{\boldsymbol{w} \in \triangle_K} \ \min_{\boldsymbol{\lambda} \in \neg i^*(\boldsymbol{\mu})} \ \sum_{k=1}^{K} w_k \, \mathrm{kl}\left(\mu_k, \lambda_k\right)} + \text{small}$$

Good:

- reliable
- practically efficient
- optimal (stop search for improvements)

Bad:

- Requires solving $\arg\max_{\boldsymbol{w}} \min_{\boldsymbol{\lambda}} \cdots$
- I hid some details

Discussion

Comparison to Racing

Racing samples uniformly, and eliminates arms sequentially until one is left (Even-Dar, Mannor, and Mansour, 2006).

Can be generalised to MCTS (Teraoka, Hatano, and Takimoto, 2014).

Penultimate arm always sampled equally often as winner.

This by itself can result in a factor 2 sub-optimality in sample complexity.

In addition, how are we comparing arms?

How should one compare two arms?

Let's compare arm A with arm B, using samples, without knowing their qualities μ_A or μ_B.

How should one compare two arms?

Let's compare arm A with arm B, using samples, without knowing their qualities μ_A or μ_B.

Fact

If $|\mu_A - \hat{\mu}_A| \leq \epsilon_A$ *and* $|\mu_B - \hat{\mu}_B| \leq \epsilon_B$ *then*

$$\left|(\mu_A - \mu_B) - (\hat{\mu}_A - \hat{\mu}_B)\right| \leq \epsilon_A + \epsilon_B$$

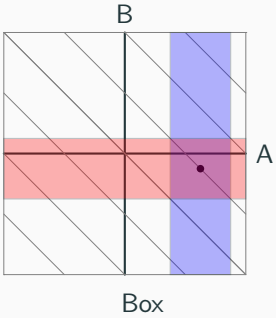
How should one compare two arms?

Let's compare arm A with arm B, using samples, without knowing their qualities μ_A or μ_B.

Fact

If $|\mu_A - \hat{\mu}_A| \leq \epsilon_A$ *and* $|\mu_B - \hat{\mu}_B| \leq \epsilon_B$ *then*

$$\left|(\mu_A - \mu_B) - (\hat{\mu}_A - \hat{\mu}_B)\right| \;\leq\; \epsilon_A + \epsilon_B$$



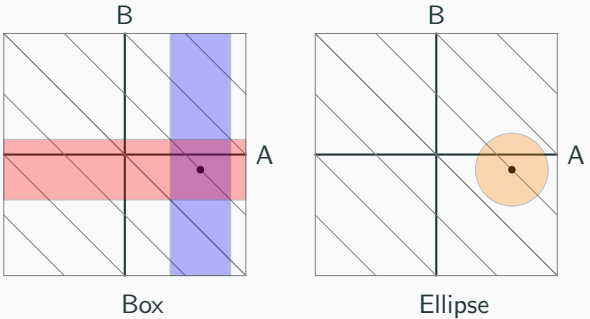
How should one compare two arms?

Let's compare arm A with arm B, using samples, without knowing their qualities μ_A or μ_B.

Fact

If $|\mu_A - \hat{\mu}_A| \leq \epsilon_A$ *and* $|\mu_B - \hat{\mu}_B| \leq \epsilon_B$ *then*

$$\left|(\mu_A - \mu_B) - (\hat{\mu}_A - \hat{\mu}_B)\right| \;\leq\; \epsilon_A + \epsilon_B$$



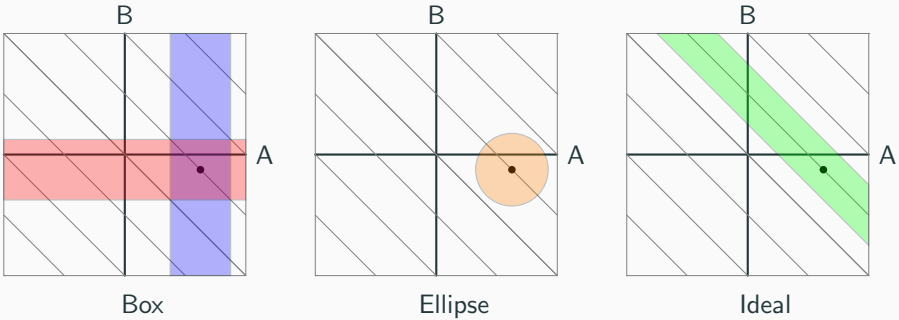
How should one compare two arms?

Let's compare arm A with arm B, using samples, without knowing their qualities μ_A or μ_B.

Fact

If $|\mu_A - \hat{\mu}_A| \leq \epsilon_A$ *and* $|\mu_B - \hat{\mu}_B| \leq \epsilon_B$ *then*

$$\left|(\mu_A - \mu_B) - (\hat{\mu}_A - \hat{\mu}_B)\right| \;\leq\; \epsilon_A + \epsilon_B$$



On $\epsilon > 0$

So far, we discussed active (ϵ, δ)-PAC learning for $\epsilon = 0$.

What about $\epsilon > 0$?

Lower bounds can be extended (Degenne and Koolen, 2019).

Yet algorithms more tricky: oracle weights $w^*(\boldsymbol{\mu})$ now a **discontinuous** function of bandit instance $\boldsymbol{\mu}$.

Simply ignoring that does not work. Terrible WSP.

Sticky-Track-and-Stop algorithm a "theoretical" fix.

Asymptotic optimality not quite so practical here. Also, only works for **finitely many** answers.

Computation

In many cases the oracle weight map can be computed efficiently.

In some cases it cannot (Al Marjani and Proutiere, 2021; Ruitong, Ajallooeian, Szepesvári, and Müller, 2017).

In between are "theoretical" positive case. For example in Pareto Front Identification the run-time is $O(\text{arms}^{\text{dimension}})$ (Crepon, Garivier, and Koolen, 2024).

Somehow every case is different.

Every classical CS problem becomes a pure exploration problem with bandit access to the inputs.

Conclusion

Conclusion

We saw

- How to decide which samples to collect
- Intuition
- Theory and algorithm design

We saw

- The specific question matters for the test!
- Lots to discuss and discover

Let's talk!

References i

- Agrawal, S., W. M. Koolen, and S. Juneja (Dec. 2021). **"Optimal Best-Arm Identification Methods for Tail-Risk Measures".** In: *Advances in Neural Information Processing Systems (NeurIPS) 34*.
- Al Marjani, A. and A. Proutiere (2021). **"Adaptive sampling for best policy identification in markov decision processes".** In: *International Conference on Machine Learning*. PMLR.
- Chen, J., X. Chen, Q. Zhang, and Y. Zhou (2017). **"Adaptive multiple-arm identification".** In: *International Conference on Machine Learning*. PMLR.
- Chen, L., J. Li, and M. Qiao (2017). **"Nearly instance optimal sample complexity bounds for top-k arm selection".** In: *Artificial Intelligence and Statistics*. PMLR.
- Crepon, É., A. Garivier, and W. M. Koolen (Mar. 2024). **"Sequential Learning of the Pareto Front for Multi-objective Bandits".** To appear in AISTATS.
- Degenne, R. and W. M. Koolen (Dec. 2019). **"Pure Exploration with Multiple Correct Answers".** In: *Advances in Neural Information Processing Systems (NeurIPS) 32*.

References ii

- Degenne, R., W. M. Koolen, and P. Ménard (Dec. 2019). **"Non-Asymptotic Pure Exploration by Solving Games".** In: *Advances in Neural Information Processing Systems (NeurIPS) 32*.
- Degenne, R., P. Ménard, X. Shang, and M. Valko (2020). **"Gamification of pure exploration for linear bandits".** In: *International Conference on Machine Learning*. PMLR.
- Even-Dar, E., S. Mannor, and Y. Mansour (2006). **"Action Elimination and Stopping Conditions for the Multi-Armed Bandit and Reinforcement Learning Problems".** In: *Journal of Machine Learning Research* 7.
- Garivier, A. and E. Kaufmann (2016). **"Optimal Best arm Identification with Fixed Confidence".** In: *Proceedings of the 29th Conference On Learning Theory (COLT)*.
- Garivier, A., E. Kaufmann, and W. M. Koolen (June 2016). **"Maximin Action Identification: A New Bandit Framework for Games".** In: *Proceedings of the 29th Annual Conference on Learning Theory (COLT)*.
- Garivier, A., P. Ménard, and L. Rossi (2017). **"Thresholding bandit for dose-ranging: The impact of monotonicity".** In: *arXiv preprint arXiv:1711.04454*.

References iii

- Haddenhorst, B., V. Bengs, and E. Hüllermeier (2021). **"Identification of the generalized Condorcet winner in multi-dueling bandits".** In: *Advances in Neural Information Processing Systems* 34.
- Hao, B., T. Lattimore, and C. Szepesvari (2020). **"Adaptive exploration in linear contextual bandit".** In: *International Conference on Artificial Intelligence and Statistics*. PMLR.
- Jedra, Y. and A. Proutiere (2020). **"Optimal best-arm identification in linear bandits".** In: *Advances in Neural Information Processing Systems* 33.
- Katariya, S., A. Tripathy, and R. Nowak (2019). **"Maxgap bandit: Adaptive algorithms for approximate ranking".** In: *Advances in Neural Information Processing Systems* 32.
- Kaufmann, E. and W. M. Koolen (Nov. 2021). **"Mixture Martingales Revisited with Applications to Sequential Tests and Confidence Intervals".** In: *Journal of Machine Learning Research* 22.246.
- Kaufmann, E., W. M. Koolen, and A. Garivier (Dec. 2018). **"Sequential Test for the Lowest Mean: From Thompson to Murphy Sampling".** In: *Advances in Neural Information Processing Systems (NeurIPS) 31*.

References iv

- Kocák, T. and A. Garivier (2021). **"Epsilon Best Arm Identification in Spectral Bandits."**. In: *IJCAI*.
- Kone, C., E. Kaufmann, and L. Richert (2023). **"Adaptive Algorithms for Relaxed Pareto Set Identification"**. In: *arXiv preprint arXiv:2307.00424*.
- Moulos, V. (2019). **"Optimal best Markovian arm identification with fixed confidence"**. In: *Advances in Neural Information Processing Systems* 32.
- Ruitong, H., M. M. Ajallooeian, C. Szepesvári, and M. Müller (2017). **"Structured Best Arm Identification with Fixed Confidence"**. In: *Algorithmic Learning Theorem (ALT)*.
- Russac, Y., C. Katsimerou, D. Bohle, O. Cappé, A. Garivier, and W. M. Koolen (Dec. 2021). **"A/B/n Testing with Control in the Presence of Subpopulations"**. In: *Advances in Neural Information Processing Systems (NeurIPS) 34*.
- Teraoka, K., K. Hatano, and E. Takimoto (2014). **"Efficient Sampling Method for Monte Carlo Tree Search Problem"**. In: *IEICE Transactions* 97-D.3.

References v

 Tirinzoni, A., M. Pirotta, M. Restelli, and A. Lazaric (2020). **"An asymptotically optimal primal-dual incremental algorithm for contextual linear bandits".** In: *Advances in Neural Information Processing Systems* 33.