

Tutorial on A/B testing (and bandit best arm identification)

Wouter M. Koolen

CWI and University of Twente

Booking, June 5, 2024

Welcome! A bit about me



Senior Researcher, Machine Learning group, Centrum Wiskunde & Informatica

Professor of Mathematical Machine Learning, University of Twente

Welcome! A bit about me



Senior Researcher, Machine Learning group, Centrum Wiskunde & Informatica

Professor of Mathematical Machine Learning, University of Twente

Lecturer *Machine Learning Theory* for MasterMath

Lecturer *Graphical Models and Causality* at UT

Welcome! A bit about me



Senior Researcher, Machine Learning group, Centrum Wiskunde & Informatica

Professor of Mathematical Machine Learning, University of Twente

Lecturer *Machine Learning Theory* for MasterMath

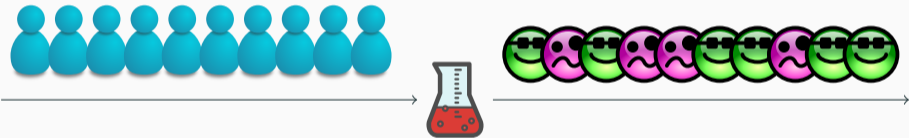
Lecturer *Graphical Models and Causality* at UT

Keywords: Machine Learning, Online Learning, Statistics, Game theory, Optimisation



Part I

How can A/B tests be used for iterative product refinement?

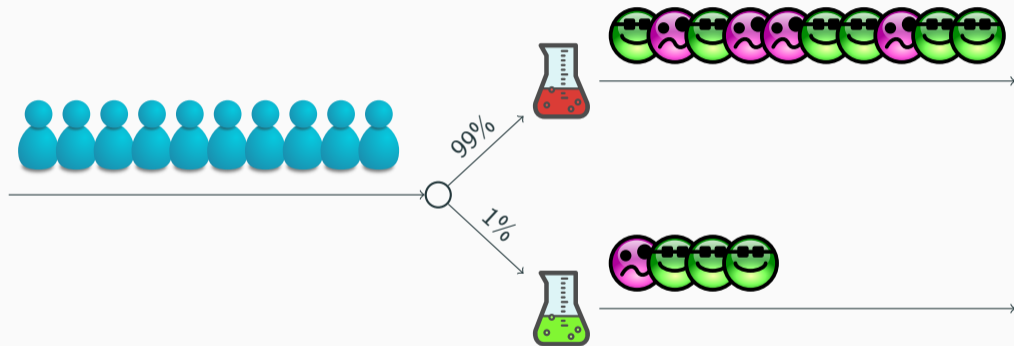
Model of the Status Quo



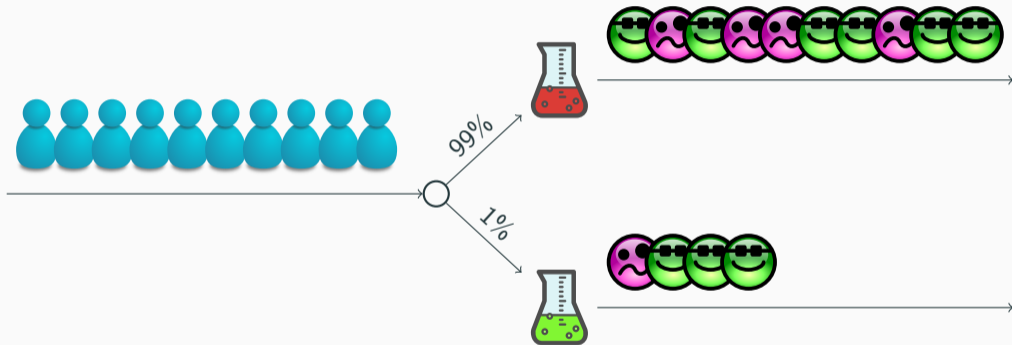
A question arises

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

Let's find out (in production)!

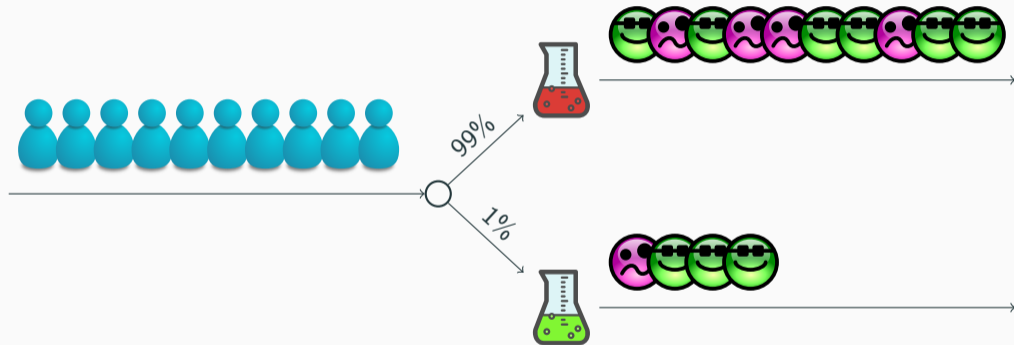


Let's find out (in production)!



Sufficient interactions pass ...

Let's find out (in production)!

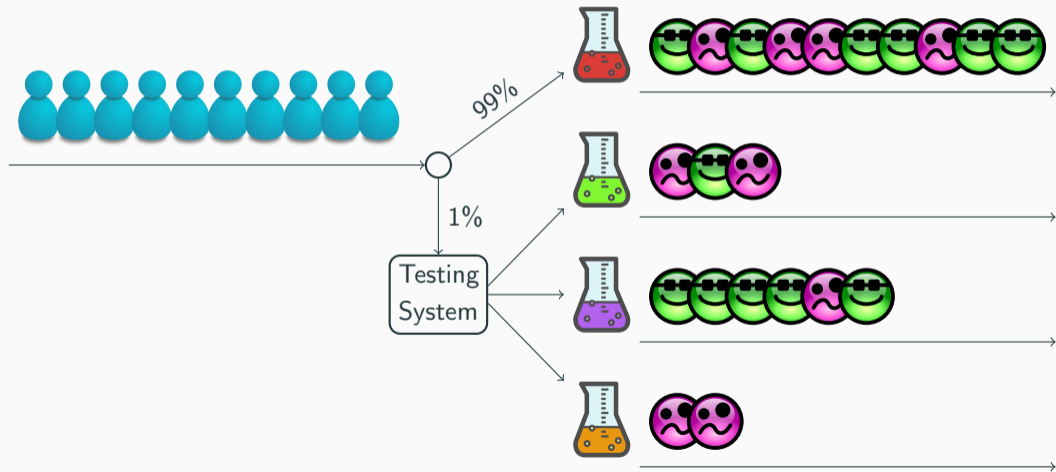


Sufficient interactions pass Decide to switch or not.

More questions emerge (sneak peek)

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

Let's find out in production (sneak peek)



Testing One Version

Setup

Current baseline performance is γ .

Setup

Current baseline performance is γ .

If we expose a stream of users to our new design, we generate a stream of rewards X_1, X_2, \dots

Setup

Current baseline performance is γ .

If we expose a stream of users to our new design, we generate a stream of rewards X_1, X_2, \dots

Rewards modelled as independent and identically distributed $X \sim \mathbb{P}$.

Setup

Current baseline performance is γ .

If we expose a stream of users to our new design, we generate a stream of rewards X_1, X_2, \dots

Rewards modelled as independent and identically distributed $X \sim \mathbb{P}$.

We want to know if $\mathbb{E}[X] > \gamma$.

Setup

Current baseline performance is γ .

If we expose a stream of users to our new design, we generate a stream of rewards X_1, X_2, \dots

Rewards modelled as independent and identically distributed $X \sim \mathbb{P}$.

We want to know if $\mathbb{E}[X] > \gamma$.

Say we take T samples and compute the empirical mean $\hat{\mu}_T = \frac{1}{T} \sum_{t=1}^T X_t$.

Roughly,

- if $\hat{\mu}_T \gg \gamma$ adopt new version
- if $\hat{\mu}_T \ll \gamma$ discard new version
- if $\hat{\mu}_T \approx \gamma$ vacillate?

Workhorse: Concentration Inequality

Consider T samples X_1, X_2, \dots, X_T drawn i.i.d. from a probability distribution with mean $\mu = \mathbb{E}[X]$. Let $\hat{\mu}_T = \frac{1}{T} \sum_{t=1}^T X_t$ be the empirical mean.

Workhorse: Concentration Inequality

Consider T samples X_1, X_2, \dots, X_T drawn i.i.d. from a probability distribution with mean $\mu = \mathbb{E}[X]$. Let $\hat{\mu}_T = \frac{1}{T} \sum_{t=1}^T X_t$ be the empirical mean.

Concentration Inequality

Relates **sample size** T , **deviation** ϵ and **confidence** δ ensuring

$$\mathbb{P}\{\hat{\mu}_T - \mu \geq \epsilon\} \leq \delta.$$

Workhorse: Concentration Inequality

Consider T samples X_1, X_2, \dots, X_T drawn i.i.d. from a probability distribution with mean $\mu = \mathbb{E}[X]$. Let $\hat{\mu}_T = \frac{1}{T} \sum_{t=1}^T X_t$ be the empirical mean.

Concentration Inequality

Relates **sample size** T , **deviation** ϵ and **confidence** δ ensuring

$$\mathbb{P}\{\hat{\mu}_T - \mu \geq \epsilon\} \leq \delta.$$

Hoeffding (bounded samples) or Chernoff (sub-Gaussian samples) give

Confidence width	For fixed T, δ , get $\epsilon = \sqrt{\frac{\ln \frac{1}{\delta}}{T}}$
Exponential error decay	For fixed ϵ, T , get $\delta = e^{-T\epsilon^2}$
Sample complexity	For fixed ϵ, δ , get $T = \frac{\ln \frac{1}{\delta}}{\epsilon^2}$.

Making it precise

Current baseline performance is γ (known).

The new version has quality $\mu = \mathbb{E}[X]$ (unknown).

Let's fix $\delta, \epsilon \in (0, 1)$, take $T = \frac{\ln \delta}{\epsilon^2}$ samples, and accept the new version if $\hat{\mu}_T \geq \gamma$.

Making it precise

Current baseline performance is γ (known).

The new version has quality $\mu = \mathbb{E}[X]$ (unknown).

Let's fix $\delta, \epsilon \in (0, 1)$, take $T = \frac{\ln \delta}{\epsilon^2}$ samples, and accept the new version if $\hat{\mu}_T \geq \gamma$.

Then

- If $\mu \geq \gamma + \epsilon$, we accept with probability $\geq 1 - \delta$.
- If $\mu \leq \gamma - \epsilon$, we accept with probability $\leq \delta$.
- If $\mu \in \gamma \pm \epsilon$ either can happen.

About setting the confidence δ

Suppose we run an algorithm that makes a **mistake** with probability $\leq \delta$.

If the algorithm is correct, our gain is G . But upon a mistake, the cost is C .

About setting the confidence δ

Suppose we run an algorithm that makes a **mistake** with probability $\leq \delta$.

If the algorithm is correct, our gain is G . But upon a mistake, the cost is C .

Then our **expected profit** is (at least)

$$(1 - \delta)G - \delta C$$

About setting the confidence δ

Suppose we run an algorithm that makes a **mistake** with probability $\leq \delta$.

If the algorithm is correct, our gain is G . But upon a mistake, the cost is C .

Then our **expected profit** is (at least)

$$(1 - \delta)G - \delta C$$

This is positive if

$$\delta \leq \frac{G}{G + C}$$

About setting the confidence δ

Suppose we run an algorithm that makes a **mistake** with probability $\leq \delta$.

If the algorithm is correct, our gain is G . But upon a mistake, the cost is C .

Then our **expected profit** is (at least)

$$(1 - \delta)G - \delta C$$

This is positive if

$$\delta \leq \frac{G}{G + C}$$

This is at least 95% of the possible gain G if

$$\delta \leq 0.05 \frac{G}{G + C}$$

Summary so far

We can test one version with error guarantees.

Fact

Detecting any difference (“effect”) of size ϵ at confidence δ takes $T = \frac{\ln \frac{1}{\delta}}{\epsilon^2}$ samples.

Pro:

- Simple
- Practical
- Reliable
- Known, fixed cost

Con:

- Need to have an idea about effect size ϵ
- Not adaptive: sample size is T even if $\mu \gg \gamma + \epsilon$
- Only one version under test

Spicing It Up

Anytime Confidence Intervals

Upgrade from a fixed sample size T to an assessment at every sample $t = 1, 2, 3, \dots$

Fix confidence $\delta \in (0, 1)$. We saw that for every fixed sample size T :

$$\mathbb{P} \left\{ \hat{\mu}_T - \mu \geq \sqrt{\frac{\ln \frac{1}{\delta}}{T}} \right\} \leq \delta.$$

It is not true that

$$\mathbb{P} \left\{ \exists t : \hat{\mu}_t - \mu \geq \sqrt{\frac{\ln \frac{1}{\delta}}{t}} \right\} \leq \delta.$$

Anytime Confidence Intervals

Upgrade from a fixed sample size T to an assessment at every sample $t = 1, 2, 3, \dots$

Fix confidence $\delta \in (0, 1)$. We saw that for every fixed sample size T :

$$\mathbb{P} \left\{ \hat{\mu}_T - \mu \geq \sqrt{\frac{\ln \frac{1}{\delta}}{T}} \right\} \leq \delta.$$

It is not true that

$$\mathbb{P} \left\{ \exists t : \hat{\mu}_t - \mu \geq \sqrt{\frac{\ln \frac{1}{\delta}}{t}} \right\} \leq \delta.$$

But almost!

Prototypical Anytime Concentration Inequality

$$\mathbb{P} \left\{ \exists t : \hat{\mu}_t - \mu \geq \sqrt{\frac{\ln \frac{1}{\delta} + O(\ln \ln t)}{t}} \right\} \leq \delta.$$

Anytime Upshot

Fact

Detecting an effect of size ϵ at confidence δ takes, on average, samples

$$\frac{\ln \frac{1}{\delta}}{\epsilon^2} + \textit{small}$$

Taking Variance into Account

Suppose rewards are Bernoulli μ . Then the variance is $\text{Var}(\mu) = \mu(1 - \mu)$.

Maximal around $\mu \approx \frac{1}{2}$, tending to zero for μ near $\{0, 1\}$. Conversion / click-through rates(!)

Taking Variance into Account

Suppose rewards are Bernoulli μ . Then the **variance** is $\text{Var}(\mu) = \mu(1 - \mu)$.

Maximal around $\mu \approx \frac{1}{2}$, tending to zero for μ near $\{0, 1\}$. Conversion / click-through rates(!)

Our concentration inequality

$$\mathbb{P} \left\{ \hat{\mu}_T - \mu \geq \sqrt{\frac{\ln \frac{1}{\delta}}{T}} \right\} \leq \delta$$

has a beautiful information-theoretic generalisation in terms of **Kullback-Leibler divergence**:

KL concentration inequality (single parameter exponential family)

$$\mathbb{P} \left\{ T \text{KL}(\hat{\mu}_T, \mu) \geq \ln \frac{1}{\delta} \right\} \leq \delta.$$

Closely related to **(Empirical) Bernstein**.

Variance Upshot

For close-by arguments $\mu \approx \gamma$,

$$\text{KL}(\mu, \gamma) \approx \frac{(\mu - \gamma)^2}{2 \text{Var}(\mu)}$$

Fact

Testing how μ relates to γ at confidence δ takes, on average, samples

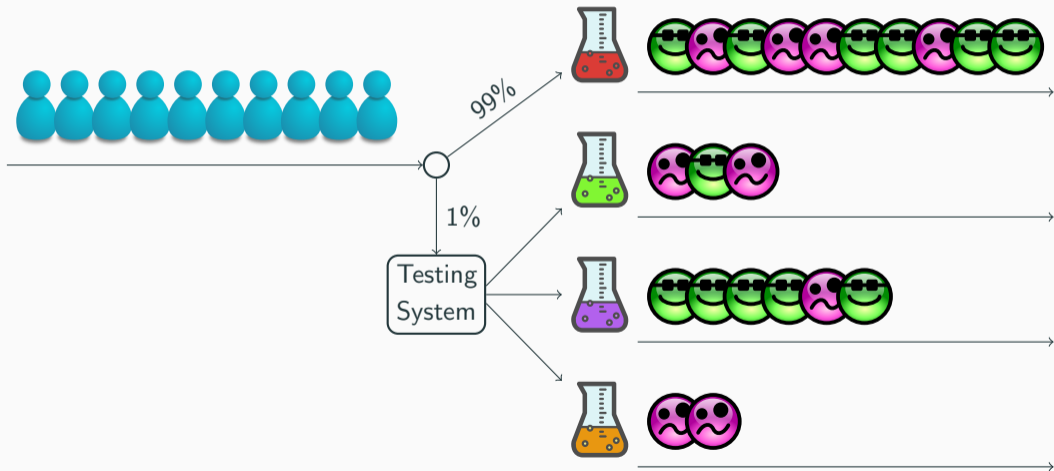
$$\frac{\ln \frac{1}{\delta}}{\text{KL}(\mu, \gamma)} + \text{small} \approx \frac{2 \text{Var}(\mu) \ln \frac{1}{\delta}}{(\mu - \gamma)^2} + \text{small}$$

Testing Multiple Versions

More questions emerge

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

Let's find out (in production)!



What are we trying to solve?

- Want to spend our samples discriminating near-optimal versions
- Need to compare estimated qualities **with each other**
- We may not know the baseline quality

What are we trying to solve?

- Want to spend our samples discriminating near-optimal versions
- Need to compare estimated qualities **with each other**
- We may not know the baseline quality

Versions customarily called **arms**.

Racing

For epochs $j = 1, 2, \dots$

- Estimate each arm up to precision $\epsilon = 2^{-j}$ by sampling it $\frac{\ln \frac{1}{\delta}}{2^{-2j}}$ times (in batch).
- Discard all provably sub-optimal arms.
- Stop when single arm remains.

Racing

For epochs $j = 1, 2, \dots$

- Estimate each arm up to precision $\epsilon = 2^{-j}$ by sampling it $\frac{\ln \frac{1}{\delta}}{2^{-2j}}$ times (in batch).
- Discard all provably sub-optimal arms.
- Stop when single arm remains.

Theorem (Even-Dar, Mannor, and Mansour, 2006)

Arm a of sub-optimality $\Delta_a = \mu^ - \mu_a$ is eliminated after it has been sampled*

$$\text{constant} \cdot \frac{\ln \frac{1}{\delta}}{\Delta_a^2} \text{ times}$$

Racing

For epochs $j = 1, 2, \dots$

- Estimate each arm up to precision $\epsilon = 2^{-j}$ by sampling it $\frac{\ln \frac{1}{\delta}}{2^{-2j}}$ times (in batch).
- Discard all provably sub-optimal arms.
- Stop when single arm remains.

Theorem (Even-Dar, Mannor, and Mansour, 2006)

Arm a of sub-optimality $\Delta_a = \mu^ - \mu_a$ is eliminated after it has been sampled*

$$\text{constant} \cdot \frac{\ln \frac{1}{\delta}}{\Delta_a^2} \text{ times}$$

Good:

- simple
- reliable

Bad:

- conservative: many union bounds,
- comparisons using per-arm estimates

Comparing two arms

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

Comparing 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

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

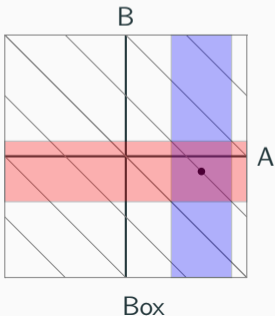
Comparing two arms

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

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



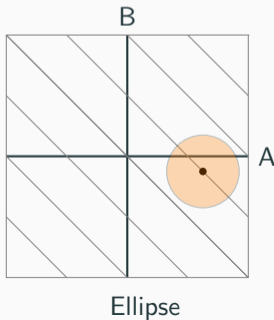
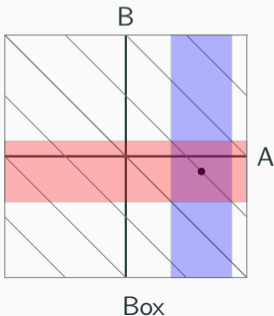
Comparing two arms

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

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



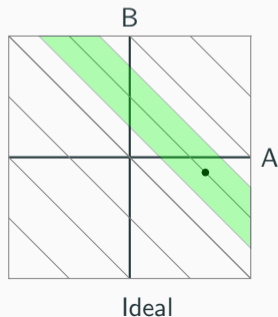
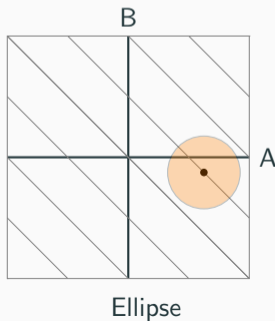
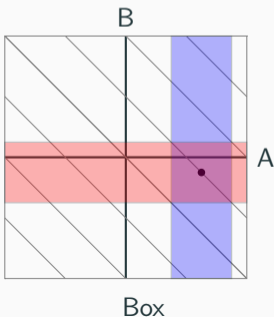
Comparing two arms

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

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



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 arm \hat{i}_t is best for $\hat{\mu}_t$, and we want to conclude that it is best for μ .

Chernoff Stopping

How to stop as soon as possible?

Need **confidence region** optimised for problem.

Say arm \hat{i}_t is best for $\hat{\mu}_t$, and we want to conclude that it is best for μ .

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

$$\neg \hat{i}_t := \left\{ \lambda \mid \arg \max_k \lambda_k \neq \hat{i}_t \right\}.$$

Chernoff Stopping

How to stop as soon as possible?

Need **confidence region** optimised for problem.

Say arm \hat{i}_t is best for $\hat{\mu}_t$, and we want to conclude that it is best for μ .

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

$$\neg \hat{i}_t := \left\{ \lambda \mid \arg \max_k \lambda_k \neq \hat{i}_t \right\}.$$

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

Generalised Likelihood Ratio Test (GLRT) Statistic

$$\Lambda_t := \ln \frac{\max_{\lambda \notin \neg \hat{i}_t} p(\text{data} \mid \lambda)}{\max_{\lambda \in \neg \hat{i}_t} p(\text{data} \mid \lambda)} = \min_{\lambda \in \neg \hat{i}_t} \sum_{a=1}^K N_{t,a} \text{KL}(\hat{\mu}_{t,a}, \lambda_a)$$

On working with GLRT

One can show that

Theorem (Kaufmann and Koolen, 2021)

For every bandit μ

$$\mathbb{P}_{\mu} \left(\underbrace{\exists t : \Lambda_t \geq \ln \frac{1}{\delta} + O(\ln \ln t)}_{\text{GLRT is big}} \text{ and } \underbrace{\hat{i}_t \neq \arg \max_k \mu_k}_{\text{mistake}} \right) \leq \delta.$$

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

Obtaining a sampling rule from GLRT

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

$$\Lambda_t = \min_{\lambda \in \hat{\mathcal{I}}_t} \sum_{a=1}^K N_{t,a} \text{KL}(\hat{\mu}_{t,a}, \lambda_a)$$

Obtaining a sampling rule from GLRT

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

$$\Lambda_t = \min_{\lambda \in \hat{\mathcal{I}}_t} \sum_{a=1}^K N_{t,a} \text{KL}(\hat{\mu}_{t,a}, \lambda_a)$$

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

$$\Lambda_t = \min_{\lambda \in \hat{\mathcal{I}}_t} \sum_{a=1}^K N_{t,a} \text{KL}(\hat{\mu}_{t,a}, \lambda_a)$$

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

The GLRT/round is maximised by sampling with proportions

$$\mathbf{w}_t := \arg \max_{\mathbf{w} \in \Delta_K} \min_{\lambda \in \hat{\mathcal{I}}_t} \sum_{a=1}^K w_a \text{KL}(\hat{\mu}_{t,a}, \lambda_a)$$

Track and Stop Framework

Definition (Track-and-Stop)

- Stop and output \hat{i}_t when $\Lambda_t = \min_{\lambda \in \neg \hat{i}_t} \sum_{a=1}^K N_{t,a} \text{KL}(\hat{\mu}_{t,a}, \lambda_a) \geq \ln \frac{1}{\delta} + O(\ln \ln t)$
- Sample with proportions $w_t = \arg \max_{w \in \Delta_K} \min_{\lambda \in \neg \hat{i}_t} \sum_{a=1}^K w_a \text{KL}(\hat{\mu}_{t,a}, \lambda_a)$

Track and Stop Framework

Definition (Track-and-Stop)

- Stop and output \hat{i}_t when $\Lambda_t = \min_{\lambda \in \neg \hat{i}_t} \sum_{a=1}^K N_{t,a} \text{KL}(\hat{\mu}_{t,a}, \lambda_a) \geq \ln \frac{1}{\delta} + O(\ln \ln t)$
- Sample with proportions $w_t = \arg \max_{w \in \Delta_K} \min_{\lambda \in \neg \hat{i}_t} \sum_{a=1}^K w_a \text{KL}(\hat{\mu}_{t,a}, \lambda_a)$

Theorem (Garivier and Kaufmann, 2016)

The expected stopping time on any bandit μ is

$$\mathbb{E}_{\mu}[\tau] \leq \frac{\ln \frac{1}{\delta}}{\max_{w \in \Delta_K} \min_{\lambda \in \neg \hat{i}_t} \sum_{a=1}^K w_a \text{KL}(\mu_a, \lambda_a)} + \text{small}$$

and this *matches* lower bounds.

Upshot of Track-and-Stop

The cost for learning the best arm is

$$\frac{\ln \frac{1}{\delta}}{\max_{\mathbf{w} \in \Delta_K} \min_{\lambda \in \hat{\nu}_t} \sum_{a=1}^K w_a \text{KL}(\mu_a, \lambda_a)} + \text{small} \ll \sum_a \frac{\ln \frac{1}{\delta}}{\Delta_a^2}$$

Good:

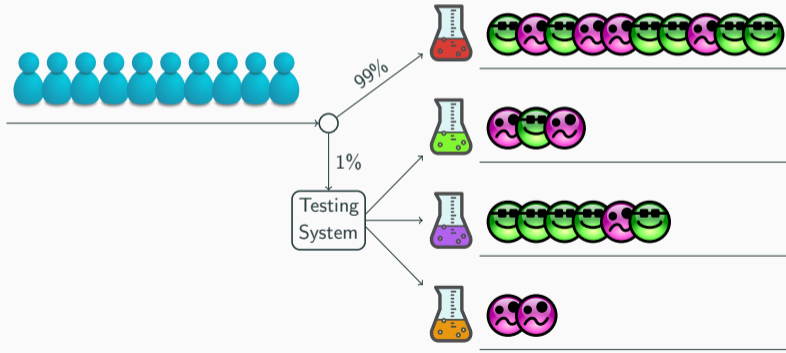
- reliable
- practically efficient
- optimal (stop search for improvements)

Bad:

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

Conclusion of Part I

We saw how to organise sampling for testing several versions.



Part II

How can A/B tests be tuned to aid higher-level decision making?

Question

At some cost in samples, we can find the arm of highest mean.

Question

At some cost in samples, we can find the arm of highest mean.

What are we going to **do** with that output?

Question

At some cost in samples, we can find the arm of highest mean.

What are we going to **do** with that output?

Did we **want/need** to find **that** arm?

Maybe not quite best arm

Many **alternative 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:

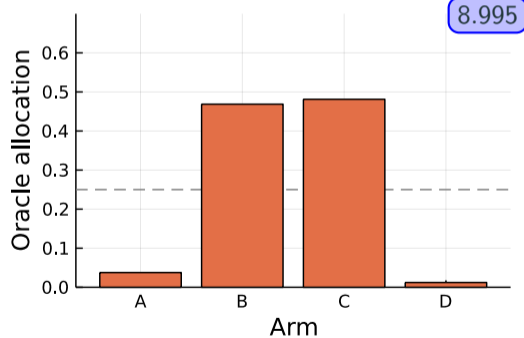
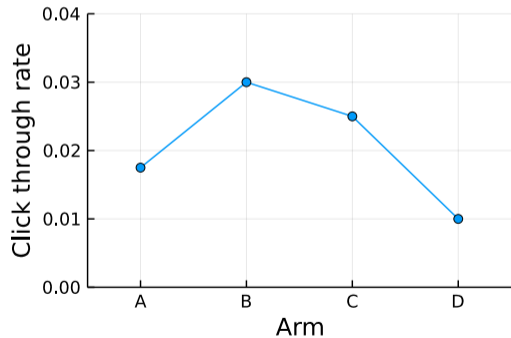
- Top- m 2017; 2017
- Spectral 2021
- Stratified 2021
- Lipschitz 2019
- Linear 2020; 2020
- Threshold 2017
- MaxGap 2019
- Duelling 2021
- Contextual 2020; 2020
- Pareto 2023
- Minimum 2018
- MCTS 2016
- Markov 2019
- Tail-Risk 2021
- MDP 2021

Spectrum

Best Arm Identification (BAI)

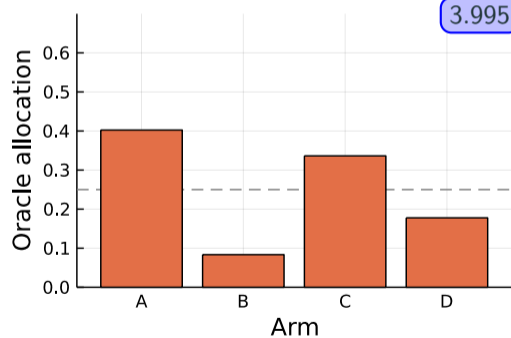
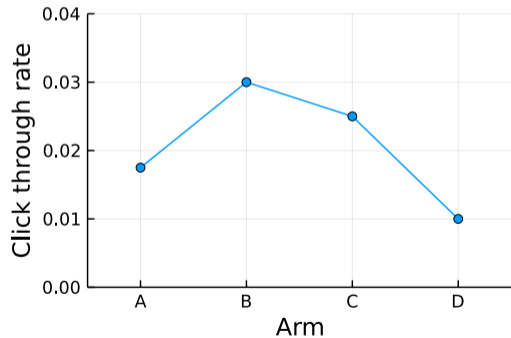
$$\arg \max_{a \in \mathcal{A}} \mu_a$$

where $\mathcal{A} = \{A, B, C, D\}$



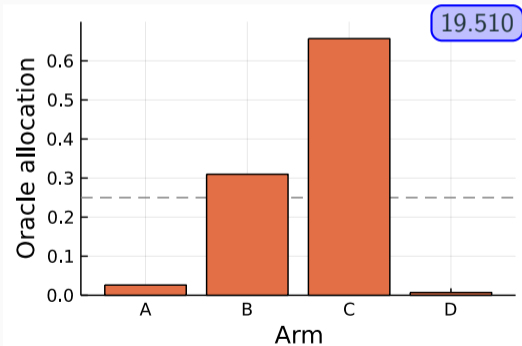
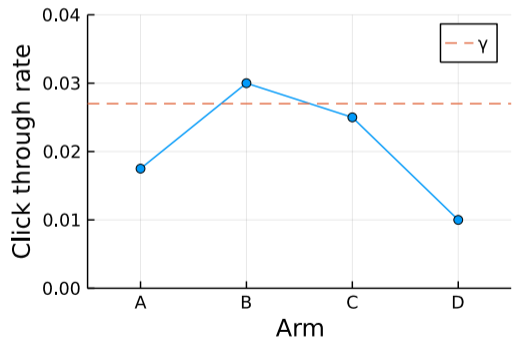
All-Better-than-the-Control (ABC)

$$\{a \in \{B, C, D\} \mid \mu_a \geq \mu_A\}$$



All-Better-than-Threshold

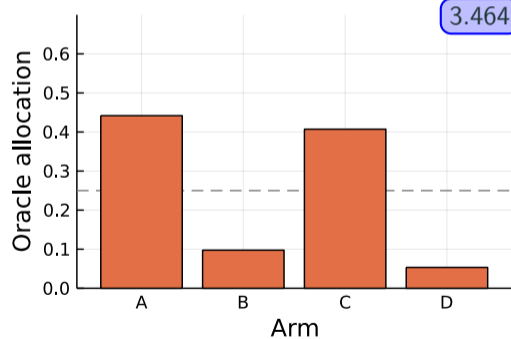
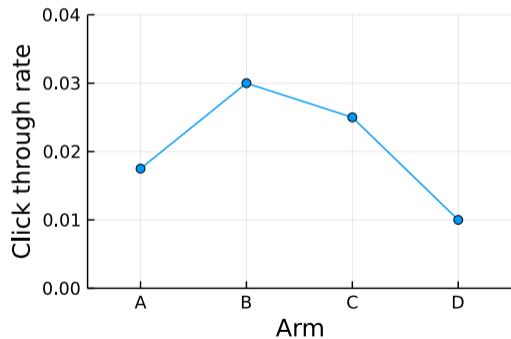
$$\{a \in \mathcal{A} \mid \mu_a \geq \gamma\}$$



Top-2

$$\{a \in \mathcal{A} \mid \mu_a \geq \mu_{(2)}\}$$

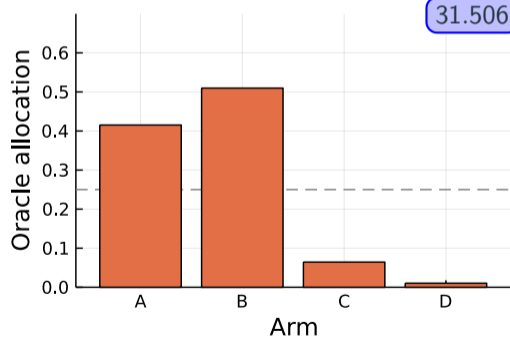
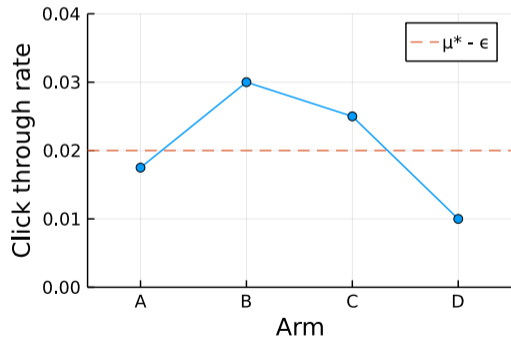
where $\mu_{(1)} \geq \mu_{(2)} \geq \dots$



Near-optimal arms

$$\{a \in \mathcal{A} \mid \mu_a \geq \mu^* - \epsilon\}$$

where $\mu^* = \max_{a \in \mathcal{A}} \mu_a$



Case: Pareto Frontier

Starting Point



Almost all **optimisation** is **multi-objective** when you think about it.

- Vacation : sunny and tasty
- Drug trial : efficacy and toxicity
- Product dev: cost and sustainability
- ...

Starting Point

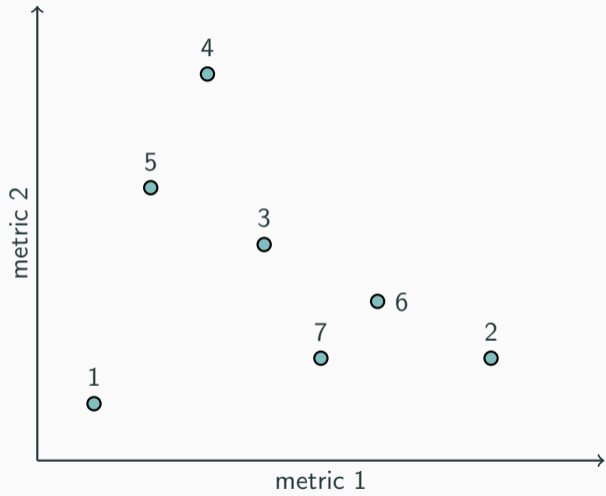


Almost all **optimisation** is **multi-objective** when you think about it.

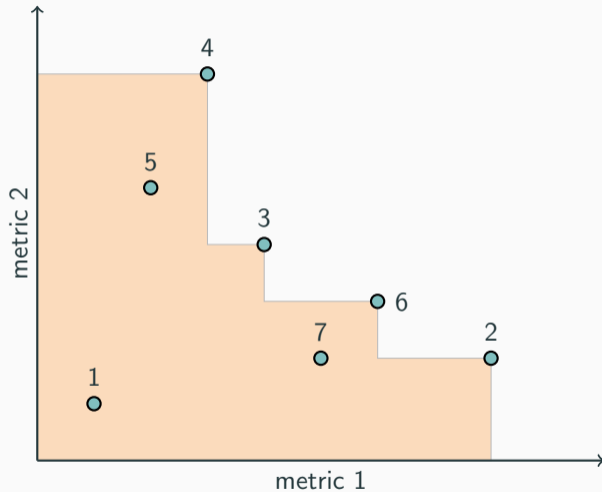
- Vacation : sunny and tasty
- Drug trial : efficacy and toxicity
- Product dev: cost and sustainability
- ...

Today: *not* in the mood to scalarise

Pareto Front



Pareto Front



Pareto front is {4, 3, 6, 2}.

Setting

K -armed multi-objective bandit $\vec{\mu} = (\mu_1, \dots, \mu_K)$.

Setting

K -armed multi-objective bandit $\vec{\mu} = (\mu_1, \dots, \mu_K)$.

Each arm k represented by a mean vector μ_k in \mathbb{R}^d .

Setting

K -armed multi-objective bandit $\vec{\mu} = (\mu_1, \dots, \mu_K)$.

Each arm k represented by a mean vector μ_k in \mathbb{R}^d .

Observations from arm k are i.i.d. multivariate Gaussian $\mathcal{N}(\mu_k, I)$.

Setting

K -armed multi-objective bandit $\vec{\mu} = (\mu_1, \dots, \mu_K)$.

Each arm k represented by a mean vector μ_k in \mathbb{R}^d .

Observations from arm k are i.i.d. multivariate Gaussian $\mathcal{N}(\mu_k, I)$.

We say arm k **dominates** arm i , denoted $\mu_k \succeq \mu_i$, if $\mu_k^j \geq \mu_i^j$ in every dimension $j = 1, \dots, d$.

Setting

K -armed multi-objective bandit $\vec{\mu} = (\mu_1, \dots, \mu_K)$.

Each arm k represented by a mean vector μ_k in \mathbb{R}^d .

Observations from arm k are i.i.d. multivariate Gaussian $\mathcal{N}(\mu_k, I)$.

We say arm k **dominates** arm i , denoted $\mu_k \succeq \mu_i$, if $\mu_k^j \geq \mu_i^j$ in every dimension $j = 1, \dots, d$.

The **Pareto front** is the set of non-dominated arms:

$$S^*(\vec{\mu}) := \{k \mid \forall i \neq k : \mu_i \not\succeq \mu_k\}$$

Protocol

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

Protocol

For $t = 1, 2, \dots, \tau$:

- Learner picks an arm $I_t \in [K]$.
- Learner sees $X_t \sim \mathcal{N}(\mu_{I_t}, I)$

Learner recommends Pareto front $\hat{S} \subseteq [K]$

Objectives



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

$$\mathbb{P}_{\vec{\mu}} \left\{ \tau < \infty \wedge \hat{S} \neq S^*(\vec{\mu}) \right\} \leq \delta$$

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

Background Theory: Lower Bound

Define the *alternatives* to $\vec{\mu}$ by

$$\text{Alt}(\vec{\mu}) := \{\vec{\lambda} \in \mathbb{R}^{K \times d} \mid S^*(\vec{\lambda}) \neq S^*(\vec{\mu})\}.$$

NB recall S^* is **Pareto front**

Background Theory: Lower Bound

Define the *alternatives* to $\vec{\mu}$ by

$$\text{Alt}(\vec{\mu}) := \{ \vec{\lambda} \in \mathbb{R}^{K \times d} \mid S^*(\vec{\lambda}) \neq S^*(\vec{\mu}) \}.$$

NB recall S^* is **Pareto front**

Theorem (Garivier and Kaufmann 2016)

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

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

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

$$\frac{1}{T^*(\vec{\mu})} = \max_{w \in \Delta_K} \min_{\vec{\lambda} \in \text{Alt}(\vec{\mu})} \frac{1}{2} \sum_{k=1}^K w_k \|\mu_k - \lambda_k\|^2.$$

Background Theory II: Algorithm



Idea is consider the oracle weight map

$$w^*(\vec{\mu}) := \arg \max_{w \in \Delta_K} \min_{\vec{\lambda} \in \text{Alt}(\vec{\mu})} \frac{1}{2} \sum_{k=1}^K w_k \|\mu_k - \lambda_k\|^2$$

and track the plug-in estimate: sample arm $I_t \sim w^*(\hat{\mu}(t-1))$.

Background Theory II: Algorithm



Idea is consider the oracle weight map

$$\mathbf{w}^*(\vec{\mu}) := \arg \max_{\mathbf{w} \in \Delta_K} \min_{\vec{\lambda} \in \text{Alt}(\vec{\mu})} \frac{1}{2} \sum_{k=1}^K w_k \|\mu_k - \lambda_k\|^2$$

and track the plug-in estimate: sample arm $I_t \sim \mathbf{w}^*(\hat{\vec{\mu}}(t-1))$.

Theorem (Degenne and Koolen, 2019)

Take set-valued interpretation of $\arg \max$ defining \mathbf{w}^* . Then $\vec{\mu} \mapsto \mathbf{w}^*(\vec{\mu})$ is upper-hemicontinuous and convex-valued. Suitable tracking ensures that as $\hat{\vec{\mu}}(t) \rightarrow \vec{\mu}$, any choice $\mathbf{w}_t \in \mathbf{w}^*(\hat{\vec{\mu}}(t-1))$ have

$$\min_{\mathbf{w} \in \mathbf{w}^*(\vec{\mu})} \|\mathbf{w}_t - \mathbf{w}\|_\infty \rightarrow 0$$

Track-and-Stop is asymptotically optimal: $\limsup_{\delta \rightarrow 0} \frac{\mathbb{E} \bar{\mu}[\tau]}{\ln \frac{1}{\delta}} = T^*(\vec{\mu})$.

Contribution

Kone, Kaufmann, and Richert (2023) consider identifying the Pareto Front among K arms in d dimensions.

- Asymptotically optimal algorithm for Pareto Front Identification.
- Computations in exponential $O(d^K)$ time per round.

Our Contribution

- Computations in polynomial $O(K^d)$ time per round.

What do we need to calculate

Degenne, Koolen, and Ménard (2019): sufficient to implement best-response oracle (= gradient)

$$\vec{\mu}, \mathbf{w} \mapsto \min_{\vec{\lambda} \in \text{Alt}(\vec{\mu})} \frac{1}{2} \sum_{k=1}^K w_k \|\mu_k - \lambda_k\|^2$$

What do we need to calculate

Degenne, Koolen, and Ménard (2019): sufficient to implement best-response oracle (= gradient)

$$\vec{\mu}, \mathbf{w} \mapsto \min_{\vec{\lambda} \in \text{Alt}(\vec{\mu})} \frac{1}{2} \sum_{k=1}^K w_k \|\mu_k - \lambda_k\|^2$$

Objective is convex, but domain $\text{Alt}(\vec{\mu})$ is **not**.

What do we need to calculate

Degenne, Koolen, and Ménard (2019): sufficient to implement best-response oracle (= gradient)

$$\vec{\mu}, \mathbf{w} \mapsto \min_{\vec{\lambda} \in \text{Alt}(\vec{\mu})} \frac{1}{2} \sum_{k=1}^K w_k \|\mu_k - \lambda_k\|^2$$

Objective is convex, but domain $\text{Alt}(\vec{\mu})$ is **not**.

Optimal transport problem

Being in the Alternative

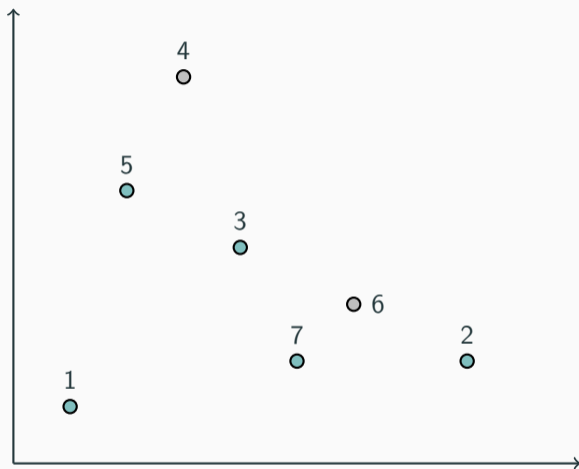
Recall

$$\vec{\lambda} \in \text{Alt}(\vec{\mu}) \quad \text{i.e.} \quad S^*(\vec{\lambda}) \neq S^*(\vec{\mu})$$

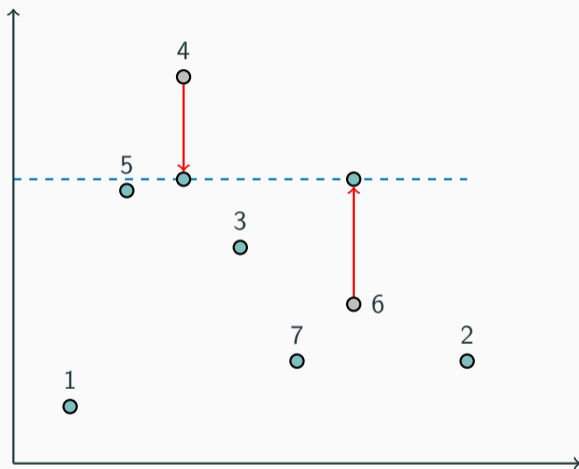
Having a different Pareto front means either

- An arm on the front in $\vec{\mu}$ is off the front in $\vec{\lambda}$, or
- An arm off the front in $\vec{\mu}$ is on the front in $\vec{\lambda}$.

Taking arm 4 off the Pareto Front

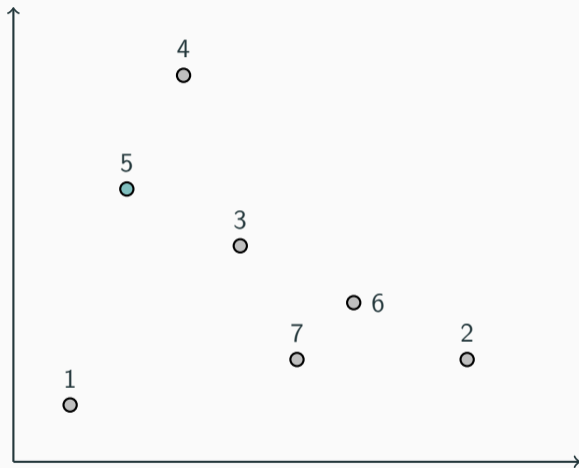


Taking arm 4 off the Pareto Front

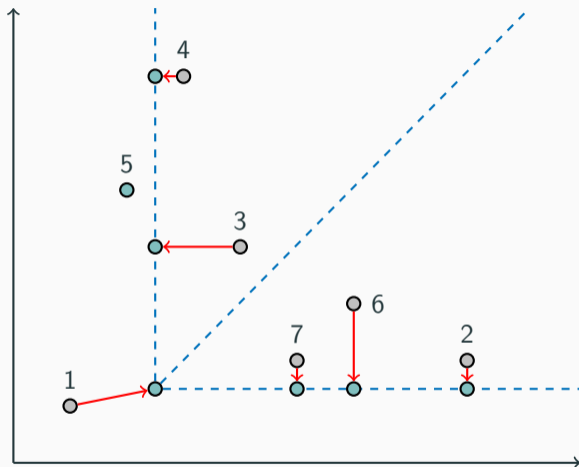


Example: we dominate arm 4 using arm 6 by moving each to the **weighted mid-point** in non-dominated coordinates.

Putting arm 1 on the Pareto Front



Putting arm 1 on the Pareto Front



Example: we make point 1 dominant by moving it north-east, and then moving all dominators
out of the way.

The heart of the insight

The cost for moving point 1 onto the front is:

$$\min_{\lambda_1} \frac{w_1}{2} \|\mu_1 - \lambda_1\|^2 + \sum_{k \in S^*(\bar{\mu})} \frac{w_k}{2} \min_{j \in [d]} (\mu_k^j - \lambda_1^j)_+^2$$

The heart of the insight

The cost for moving point 1 onto the front is:

$$\min_{\lambda_1} \frac{w_1}{2} \|\mu_1 - \lambda_1\|^2 + \sum_{k \in S^*(\vec{\mu})} \frac{w_k}{2} \min_{j \in [d]} (\mu_k^j - \lambda_1^j)_+^2$$

and that is

$$\min_{\phi: S^*(\vec{\mu}) \rightarrow [d]} \min_{\lambda_1} \underbrace{\frac{w_1}{2} \|\mu_1 - \lambda_1\|^2 + \sum_{k \in S^*(\vec{\mu})} \frac{w_k}{2} (\mu_k^{\phi(k)} - \lambda_1^{\phi(k)})_+^2}_{\text{separable convex problem}}$$

The heart of the insight

The cost for moving point 1 onto the front is:

$$\min_{\lambda_1} \frac{w_1}{2} \|\mu_1 - \lambda_1\|^2 + \sum_{k \in S^*(\vec{\mu})} \frac{w_k}{2} \min_{j \in [d]} (\mu_k^j - \lambda_1^j)_+^2$$

and that is

$$\min_{\phi: S^*(\vec{\mu}) \rightarrow [d]} \min_{\lambda_1} \underbrace{\frac{w_1}{2} \|\mu_1 - \lambda_1\|^2 + \sum_{k \in S^*(\vec{\mu})} \frac{w_k}{2} (\mu_k^{\phi(k)} - \lambda_1^{\phi(k)})_+^2}_{\text{separable convex problem}}$$

Not **all** $\phi: S^*(\vec{\mu}) \rightarrow [d]$ need to be attempted.

Only $\binom{K+d-1}{d-1}$ due to geometry of \mathbb{R}^d .

Conclusion of Part II

With that, everything slots in place and we obtain an algorithm for Pareto Front Identification with

- asymptotically optimal sample complexity
- polynomial time cost per round

Now interested in going beyond

- Gaussian
- $\epsilon = 0$
- independence

Conclusion

Conclusion of Tutorial

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.
-  Degenne, R. and W. M. Koolen (Dec. 2019). **“Pure Exploration with Multiple Correct Answers”**. In: *Advances in Neural Information Processing Systems (NeurIPS) 32*.
-  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*.





References ii

-  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*.
-  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.

References iii

-  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.
-  Kocák, T. and A. Garivier (2021). **“Epsilon Best Arm Identification in Spectral Bandits.”**. In: *IJCAI*.

References iv

-  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.
-  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.
-  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.