Online Learning and Decision Making

James Grant (he/him)

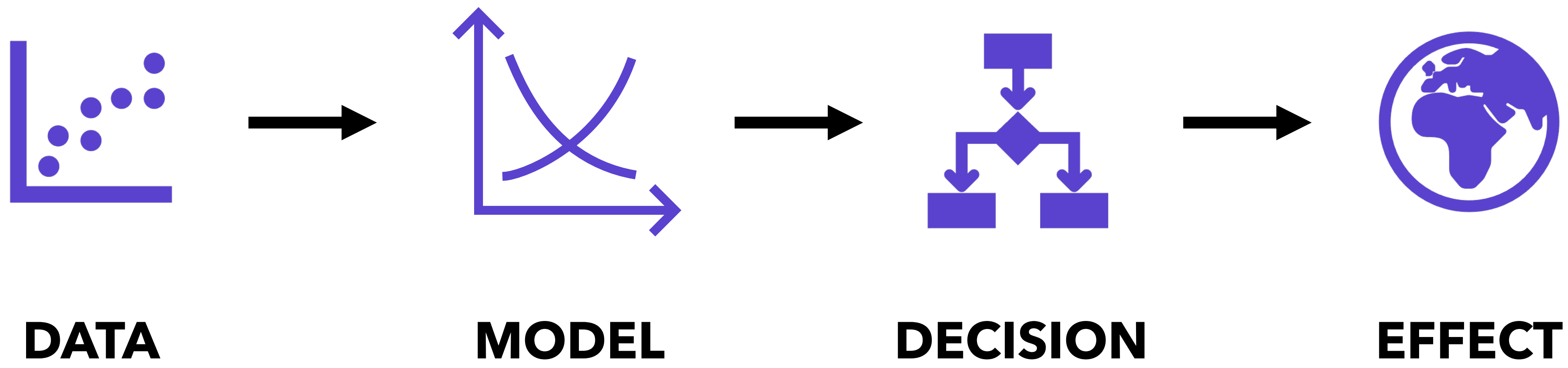
Lancaster University

j.grant@lancaster.ac.uk

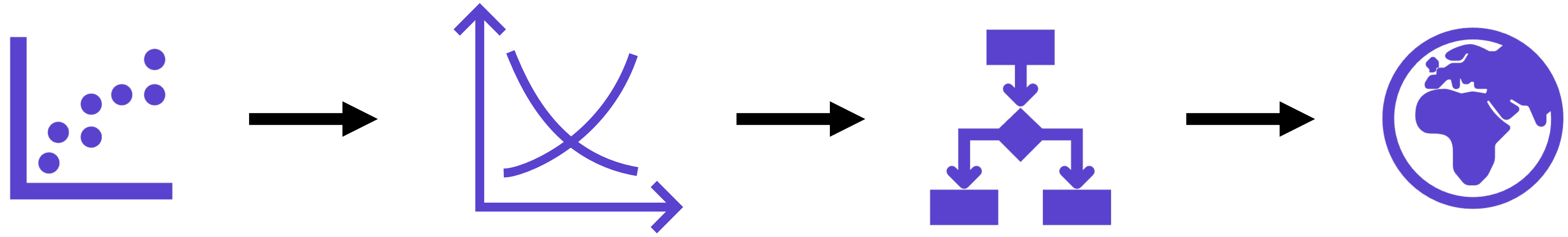
@james_a_grant

Peak Ensemble – Wednesday 7th July

Using data to make decisions



Example: what to send customers?



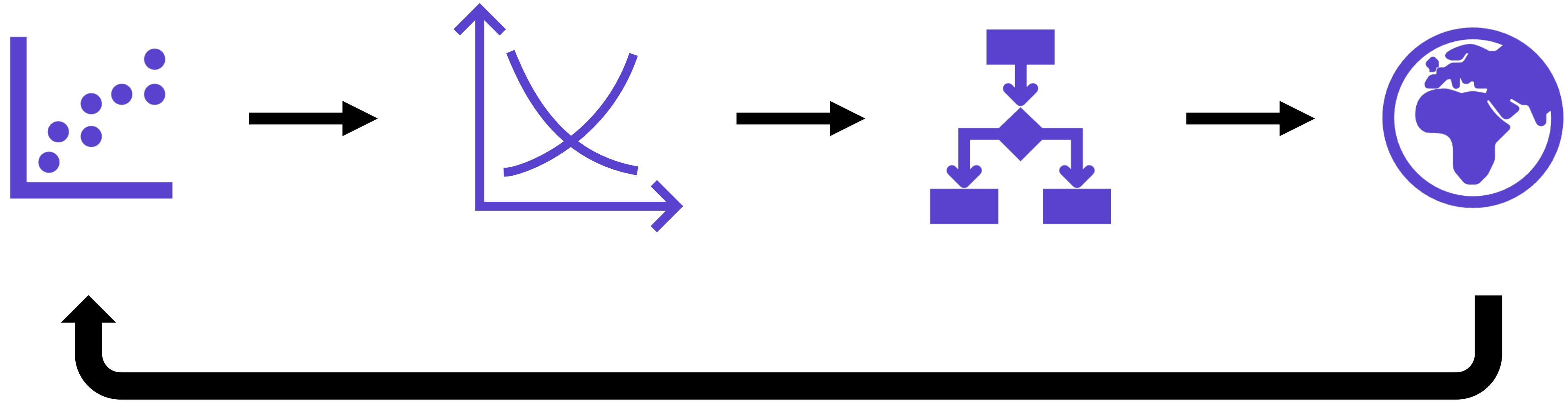
Test some different messaging strategies

Model customer response to strategies

Optimise for return, satisfaction etc.

Hope that's a good choice?

We can observe effects and iterate



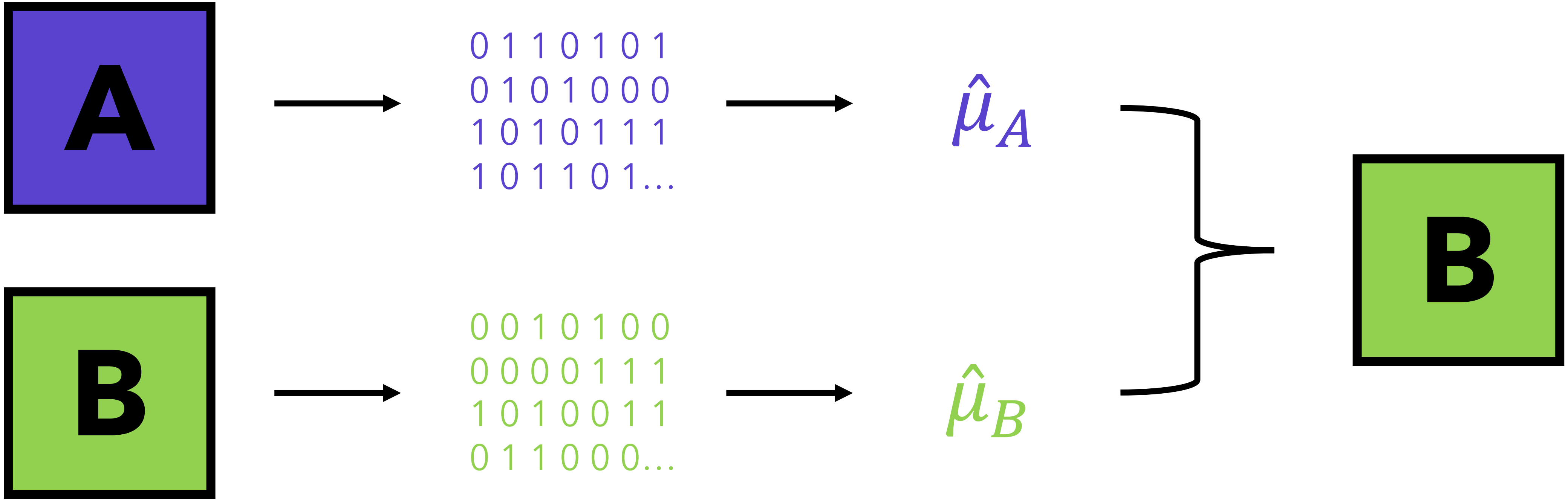
Unlike in many classical applications, we often have capacity to revise an initial decision (many times).

Today's central message:

When the potential to make decisions repeatedly arises, we **can** and **ought** to do better than collecting data once, fitting a model once, and hoping for the best.

1. "But can't we just do A/B testing?"

We certainly can, but it's not necessarily optimal

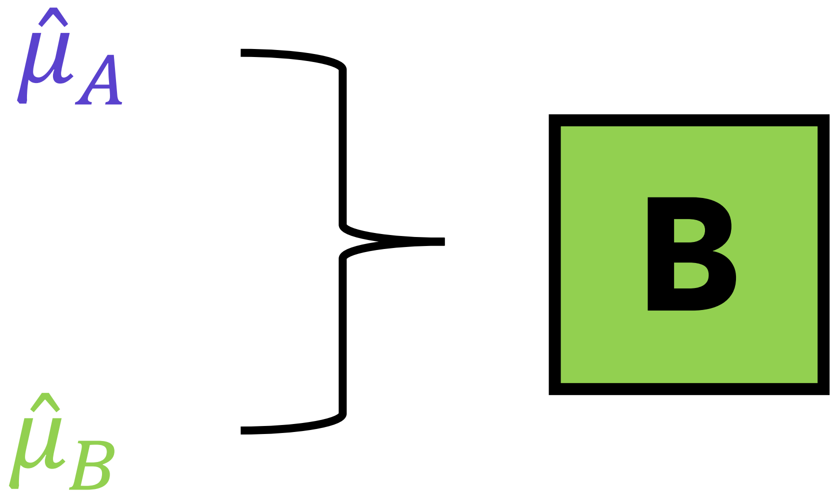


1. "But can't we just do A/B testing?"

We certainly can, but it's not necessarily optimal

When we make the comparison between $\hat{\mu}_A$ and $\hat{\mu}_B$, it's possible to make a mistake, i.e.

$$P(\hat{\mu}_B > \hat{\mu}_A \mid \mu_A \geq \mu_B) > 0$$



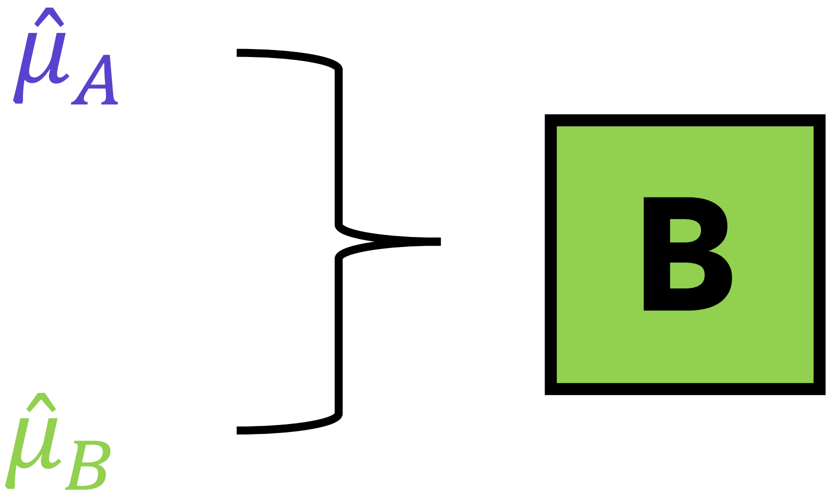
1. "But can't we just do A/B testing?"

We certainly can, but it's not necessarily optimal

When we make the comparison between $\hat{\mu}_A$ and $\hat{\mu}_B$, it's possible to make a mistake, i.e.

$$P(\hat{\mu}_B > \hat{\mu}_A \mid \mu_A \geq \mu_B) \propto 1/n$$

The chance is small, but if we use this forever more, we may lose out in the long run.



2. "So what is the right approach?"

An adaptive balance between data collection and aiming for the best outcome.

Multi-armed Bandit
Consider making decisions between A and B at times $t = 1, 2, \ldots$

If $D_t = A$ successful with probability μ_A
If $D_t = B$ successful with probability $\mu_B < \mu_A$ (but we don't know that!)

Whenever $D_t = B$, a loss is (effectively) incurred. We want to minimise expected number of times the suboptimal action is used.

2. "So what is the right approach?"

An adaptive balance between data collection and aiming for the best outcome.

Whenever $D_t = B$, a loss is (effectively) incurred. We want to minimise number of times B is used.

If A/B testing:

- We use A and B n times, and then commit to best.
- It's possible to commit to the wrong one, and incur error for $T - n$ subsequent decisions.
- Very bad if $T \gg n$!

2. “So what is the right approach?”

An adaptive balance between data collection and aiming for the best outcome.

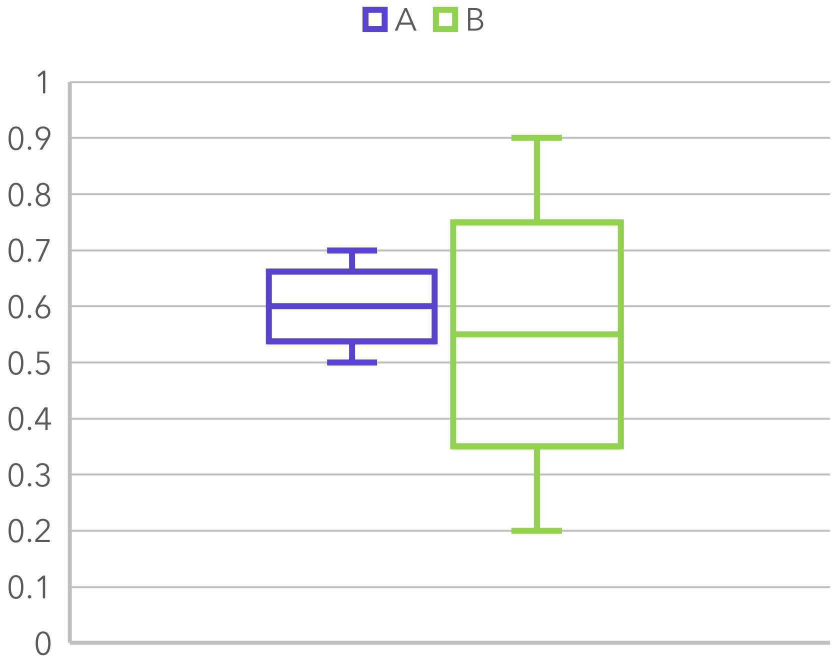
Whenever $D_t = B$, a loss is (effectively) incurred. We want to minimise number of times B is used.

If A/B testing + Greedy Follow-up

- We use A and B n times, and then use whatever has highest mean, with continued monitoring.
- If B looks best after n samples, we need $\hat{\mu}_B$ to fall below $\hat{\mu}_A$ at some point – may not happen if A underestimated initially!

3. "How do we strike the balance?"

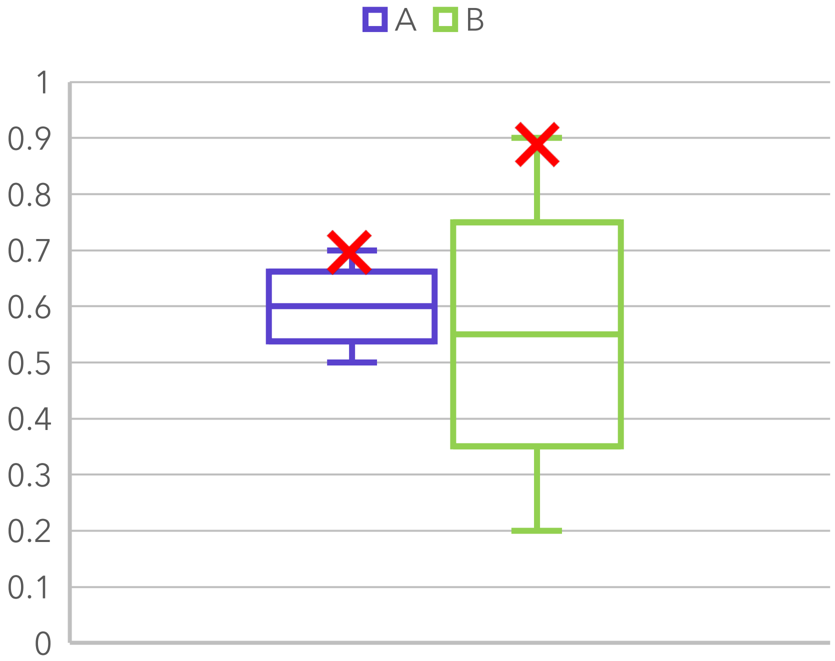
Two successful methodologies: 1) Optimism



When we make decision D_t, we consider an optimistic estimate of expected reward (**upper confidence bound**)

3. "How do we strike the balance?"

Two successful methodologies: 1) Optimism



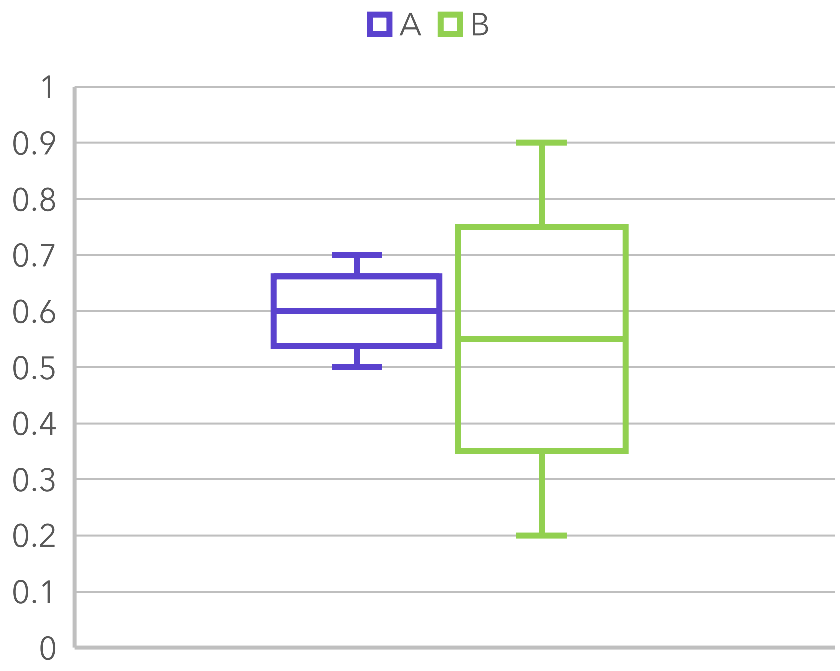
When we make decision D_t, we consider an optimistic estimate of expected reward (**upper confidence bound**)

B is under-explored, so here we choose it, despite it having a lower mean estimate.

Iterating this process ensures we both **explore** and **exploit**

3. "How do we strike the balance?"

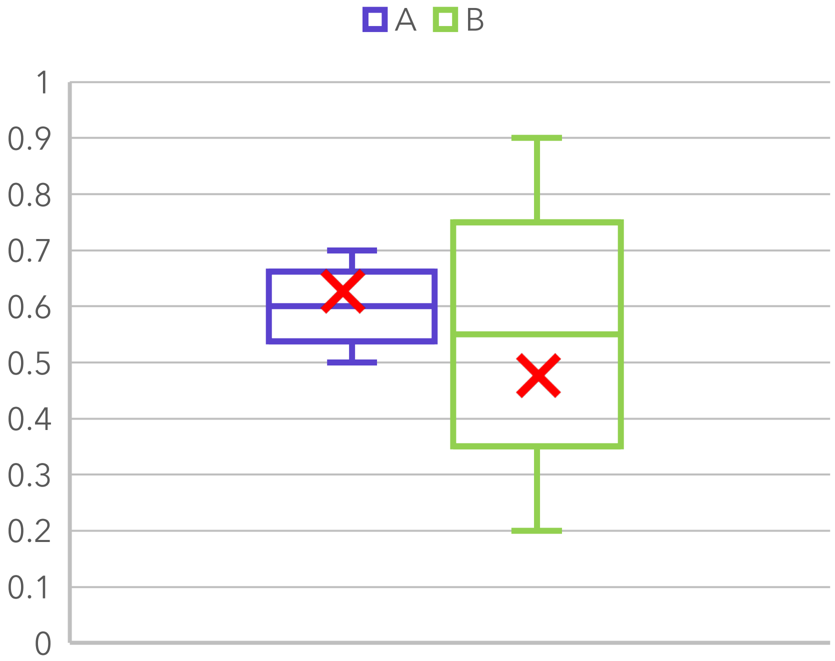
Two successful methodologies: 2) Randomisation



When we make decision D_t, we consider an random sample from posterior distribution on each mean (**Thompson Sampling**)

3. "How do we strike the balance?"

Two successful methodologies: 2) Randomisation

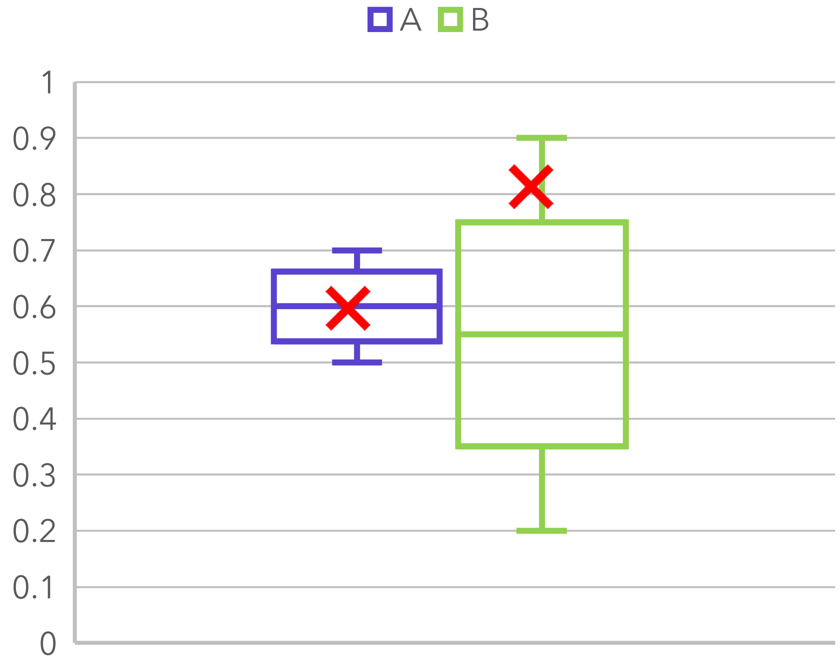


When we make decision D_t, we consider an random sample from posterior distribution on each mean (**Thompson Sampling**)

A has the higher sample in this instance, so we choose it.

3. "How do we strike the balance?"

Two successful methodologies: 2) Randomisation



When we make decision D_t, we consider an random sample from posterior distribution on each mean (**Thompson Sampling**)

B has the higher sample in this instance, so we choose it.

Iterating this process ensures we both **explore** and **exploit**

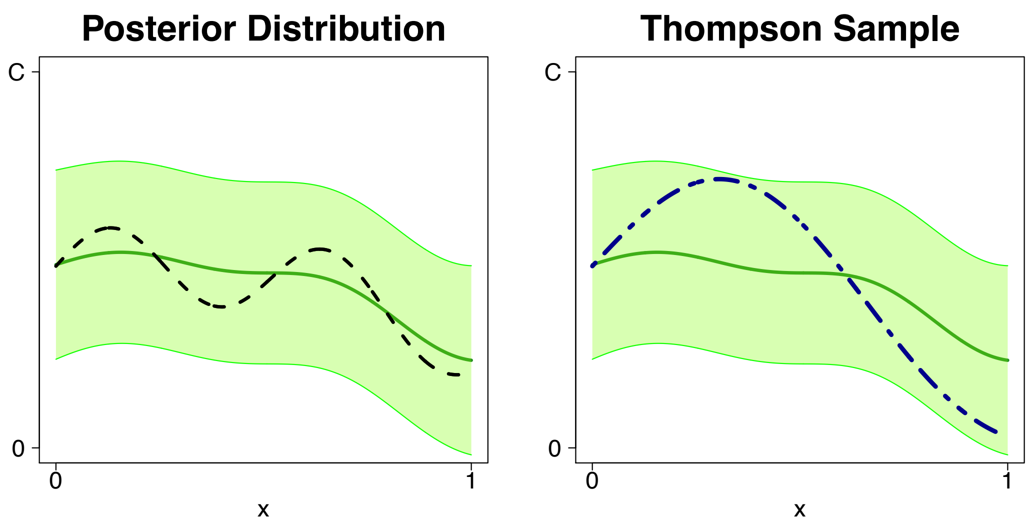
4. "Real life is more complicated."

These methods have found successful application in a much broader range of problems.

1. **Continuous Decision Space**

Binary or even discrete set of options is often unrealistic – e.g. pricing, parameter tuning.

Real optimisation problem is of an unknown function $f(d)$.



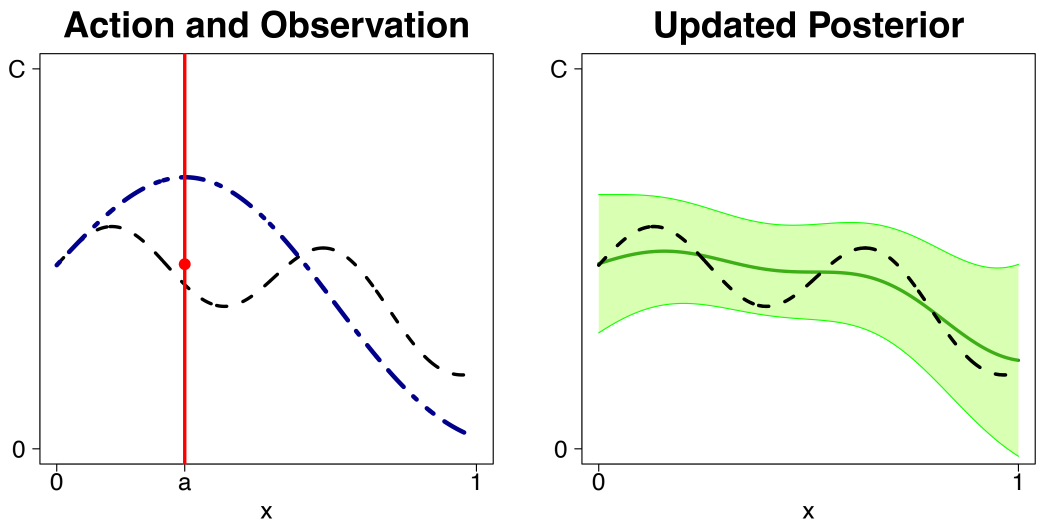
4. “Real life is more complicated.”

These methods have found successful application in a much broader range of problems.

1. **Continuous Decision Space**

Binary or even discrete set of options is often unrealistic – e.g. pricing, parameter tuning.

Real optimisation problem is of an unknown function $f(d)$.



4. "Real life is more complicated."

These methods have found successful application in a much broader range of problems.

2. **Combinatorial Decisions**

Some decisions involve multiple components – e.g. managing stock levels, portfolio optimisation, slate recommendations

Plug optimism or randomisation in to a combinatorial optimisation problem

Rather than maximise at parameter estimates as best guess

$$\max_{\boldsymbol{D}} f(\boldsymbol{D}, \hat{\mu}, \hat{\theta}, \hat{\lambda}, \dots)$$

Substitute optimistic/randomised parameters to same problem

$$\max_{\boldsymbol{D}} f(\boldsymbol{D}, \tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\lambda}}, \dots)$$

4. “Real life is more complicated.”

These methods have found successful application in a much broader range of problems.

3. **Non-stationary reward functions**

Value of an option can often change through time – different customers on different days, seasonality, diminishing interest in repeated actions.

Suitably modified optimism and randomisation continue to be successful.

5. “How do I get started?”

There is a large literature around a few central ideas

What are you optimising over?

- Discrete set of options – multi-armed bandit
- Combinations of components – combinatorial bandit
- Continuous set – continuum-armed bandit, X-armed bandit, Bayesian optimisation

Stationary in Time?

- Yes – great, use the above
- No, due to exogenous variables – contextual bandit
- No, due to unpredictable variation – non-stationary bandit, restless bandit

Other considerations?

- Immediate feedback or not? Parallelisation? Distribution of rewards? – bespoke extensions
- State effects - Reinforcement Learning

5. “How do I get started?”

There is less open-source code than in some areas

Simplicity vs Need to Interface

- The complex aspect tends to be interfacing live inference with decision-making.
- The actual rules are often not complex
- Bayesian Optimisation and RL methods tend to be more complex, and have some associated code, e.g. BOTorch in Python.

Theoretical Guarantees

- A LOT of the literature deals in regret guarantees
- Important academic work, and useful as reassurance of efficacy – but complicated
- Don’t be put off!

Today's central message:

When the potential to make decisions repeatedly arises, we **can** and **ought** to do better than collecting data once, fitting a model once, and hoping for the best.

Today’s central message:

Optimistic and **randomised** techniques, such as upper confidence bounds and Thompson Sampling allow an appropriate, optimal balance between **exploration** (data collection) and **exploitation** (optimal decisions) to be struck.

Thank you for listening!

James Grant (he/him)

Lancaster University

j.grant@lancaster.ac.uk

@james_a_grant