TechFest
the & in R&D

Microsoft Research

Learning to Interact

Léon Bottou

Microsoft

User interaction data is useful

Recommendation system

Did users take our suggestion? → Improve recommendation system.

Search and ads

Did users click on our results? → Improve search algorithm.

Did users click on our ads? → Improve ad placement.

User interface

Did users format this text as a list? → Improve auto-format logic.

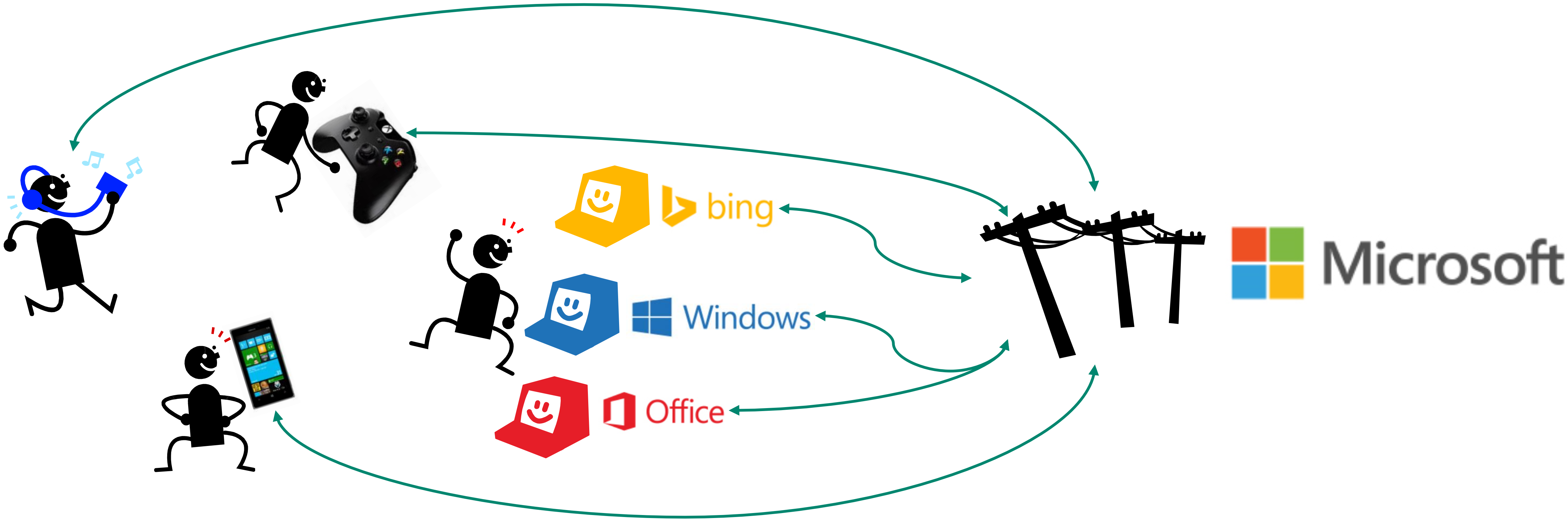
Did users complete the task? → Improve GUI nannies.

Personalization

Did this user change the default font size? → Personalize UI.

Did this user show signs of boredom? → Adjust game difficulty.

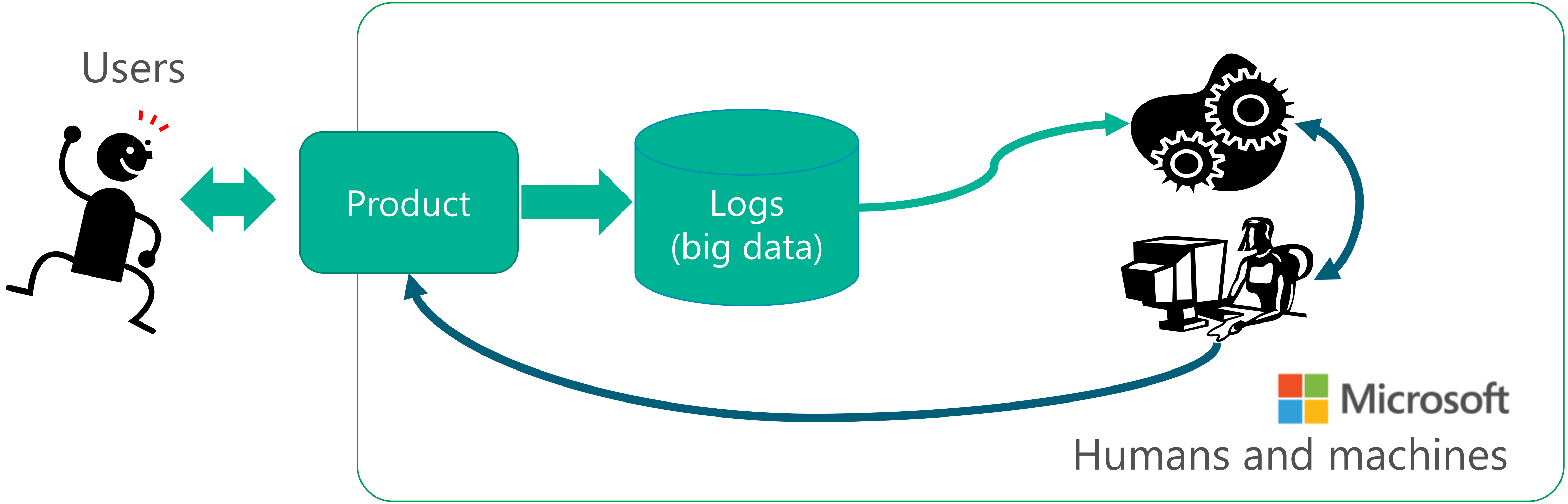
All products are connected



Unprecedented volume of user interaction data

Improve products / Personalize products / Learn from users / Fast

Machine learning to the rescue ?

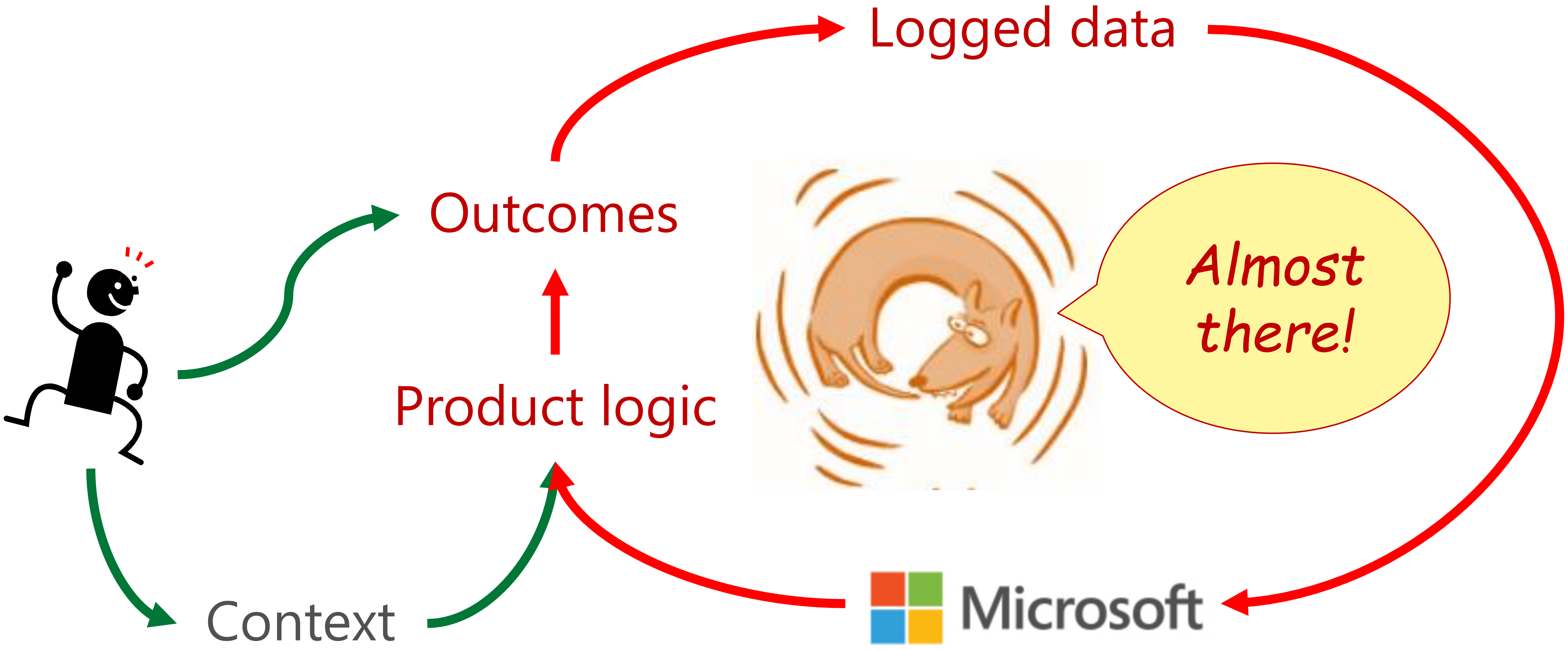


Using log statistics to justify product changes

PMs arguing about coding projects, ML algos updating click prediction models, etc.

The causal loop

Always one step behind … or worse ?



Learning to interact

Counterfactual reasoning





Multi-world testing






Contextual bandits
Explore/Exploit





Offline policy evaluation

And many more people working on connected topics.

Summary

I. Causation and correlation

The nature of the problem.

II. Randomization, counterfactuals, etc.

Elements of the solution

Causation and correlation

Manipulations

Correlations have predictive value

"It is raining" ⇒ *"People probably carry open umbrellas."*

"People carry open umbrellas" ⇒ *"It is probably raining."*

What is the outcome of a manipulation?

Manipulating the system changes the data!

- *"Will it rain if we ban umbrellas?"*
- *"Would it have rained if we had banned umbrellas?"*

Causation

Causal relations let us reason about the outcome of manipulations.

Reichenbach's common cause principle

Why are events A and B correlated?

Example event A : *"Suggestion is highlighted in red."*

Example event B : *"User takes the suggestion."*

Three cases:

- A causes B.
- B causes A.
- A and B have a common causes C.

Hans Reichenbach
1891-1953

What happens to B if we manipulate A?

The answer is different for each case.

Case 1 – A causes B

Then, manipulating A has an effect on B

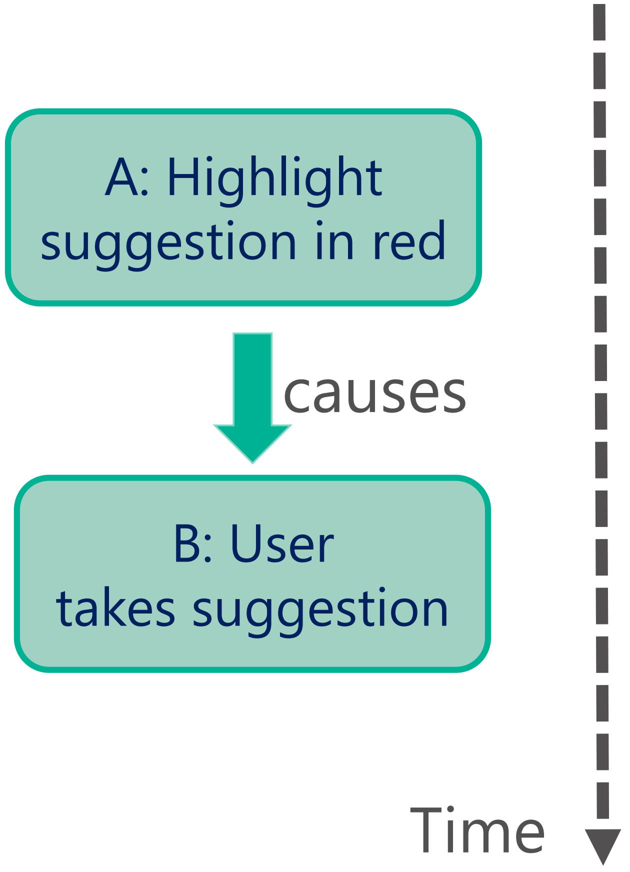
Example

Event A : *"Suggestion is highlighted in red."*

Event B : *"User takes the suggestion."*

Highlighting suggestions in red more often causes users to take the suggestions more often.

Maybe our suggestions were not visible enough…



Case 2 – B causes A

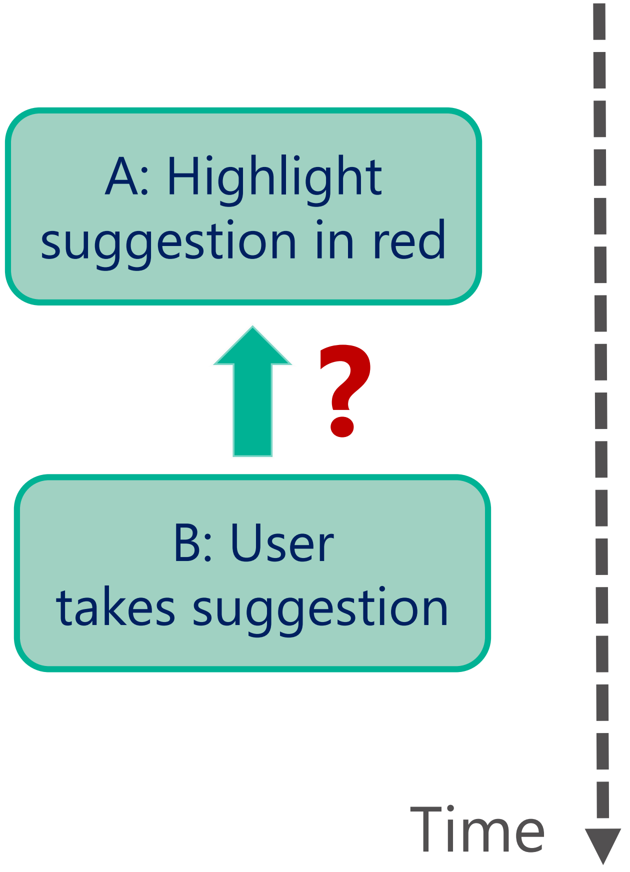
Then, manipulating A has no obvious effect on B

But we cannot go against time…

Event A : *"Suggestion is highlighted in red."*

Event B : *"User takes the suggestion."*

In this case, event B occurs after event A.
Therefore it is unlikely that B causes A!



Case 3 – A and B have common causes

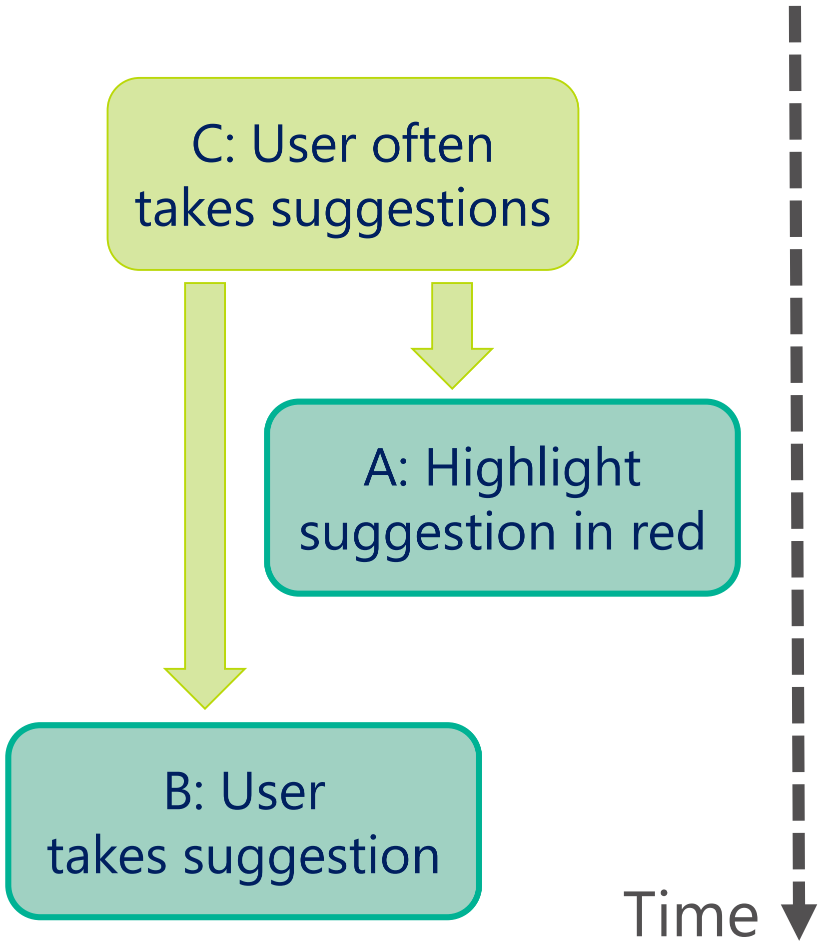
Example

Event C : *"User often takes suggestions."*

Assume that a piece of code (or a bug) favors red highlights when the user has a history of taking our suggestions.

Outcome of manipulating A ?

Will we increase the take rate if we use more red highlights?

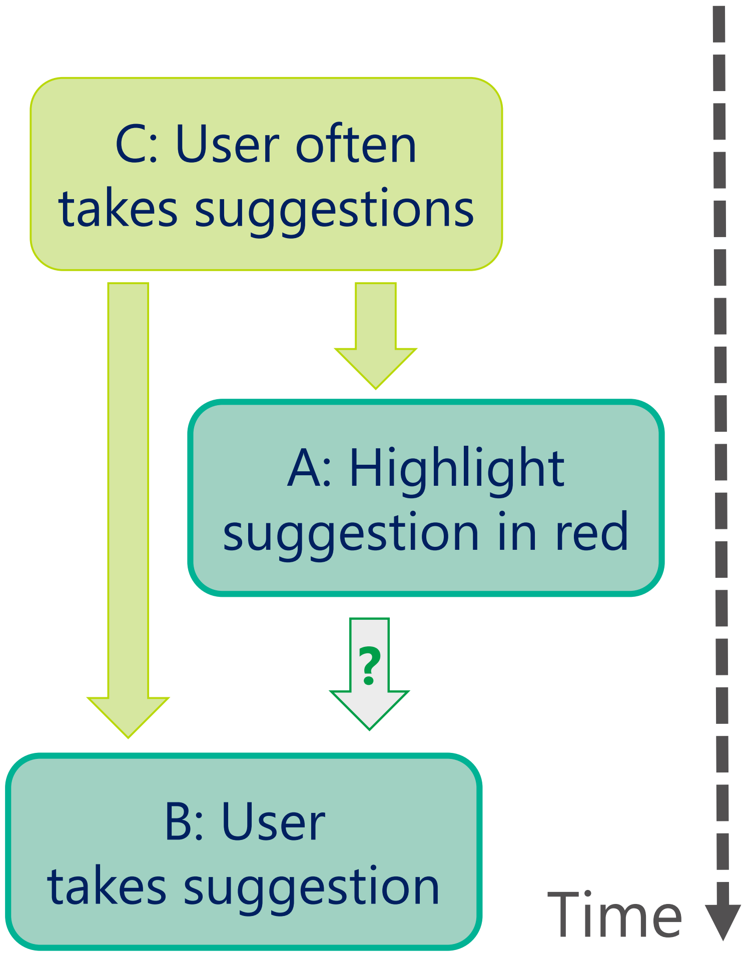


Case 3 – Manipulations

Outcome of manipulating A ?

Does the user take the suggestion
because he likes suggestions,
and also because he likes red?
→ Red highlight increases take rate.

Does the user take the suggestion
because he likes suggestions,
despite the fact that he dislikes red.
→ Red highlight decreases take rate.



An extreme case

	Average take rate	Take rate for C=false	Take rate for C=true
No highlight	24/200 (12%)		
Red highlight	30/200 (15%)		

1

Users take suggestions more often with red highlights

An extreme case

	Average take rate	Take rate for C=false	Take rate for C=true
No highlight	24/200 (12%)	•/182	•/18
Red highlight	30/200 (15%)	•/150	•/50

2 Bug favors red highlights when user is known to like suggestions

1 Users take suggestions more often with red highlights

An extreme case

	Average take rate	Take rate for C=false	Take rate for C=true
No highlight	24/200 (12%)	18/182 (10%)	6/18 (33.3%)
Red highlight	30/200 (15%)	14/150 (9.3%)	16/50 (32%)

2 Bug favors red highlights when user is known to like suggestions

1 Users take suggestions more often with red highlights

3 In fact, the users take the suggestion because they like suggestions, and despite slightly disliking the red highlights!

This effect is named “Simpson’s paradox” (1951).

Simpson's "paradox"

	Average take rate	Take rate for C=false	Take rate for C=true
No highlight	24/200 (12%)	18/182 (10%)	6/18 (33.3%)
Red highlight	30/200 (15%)	14/150 (9.3%)	16/50 (32%)

2 Bug favors red highlights when user is known to like suggestions

1 Users take suggestions more often with red highlights

3 Red highlights are a bad idea for both kinds of users.

→ Using more data won't make the answer correct!

Consequences

In summary

- Complicated conditions create a bias:
  red highlights often shown to good users.
- Unaware of this condition, engineers observe a positive correlation between red highlights and user take rate.
- They manipulate the system to produce more red highlights.
- Red highlights were a bad idea all along.
  The global take rate goes down.
  The positive correlation is still there!

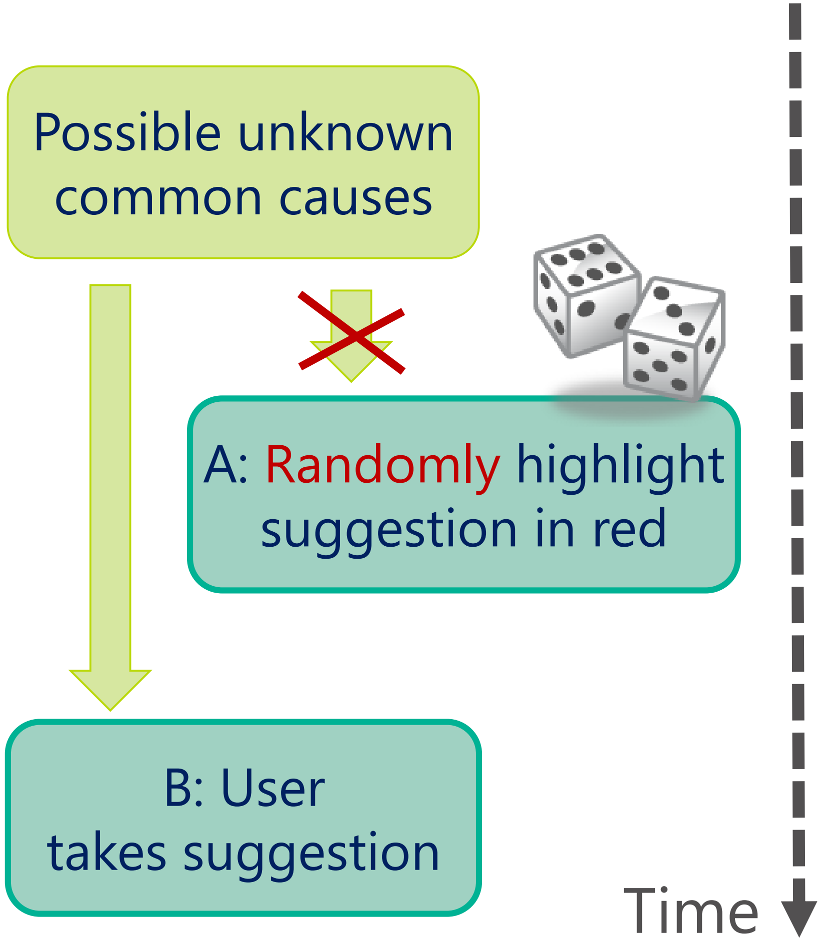
Randomization

Unknown common causes

We can control for the known common causes.
What leads us astray are those we don't know.

Randomly picking event A

The only cause of A is a roll of the dices.
Therefore no event C can be a common cause.



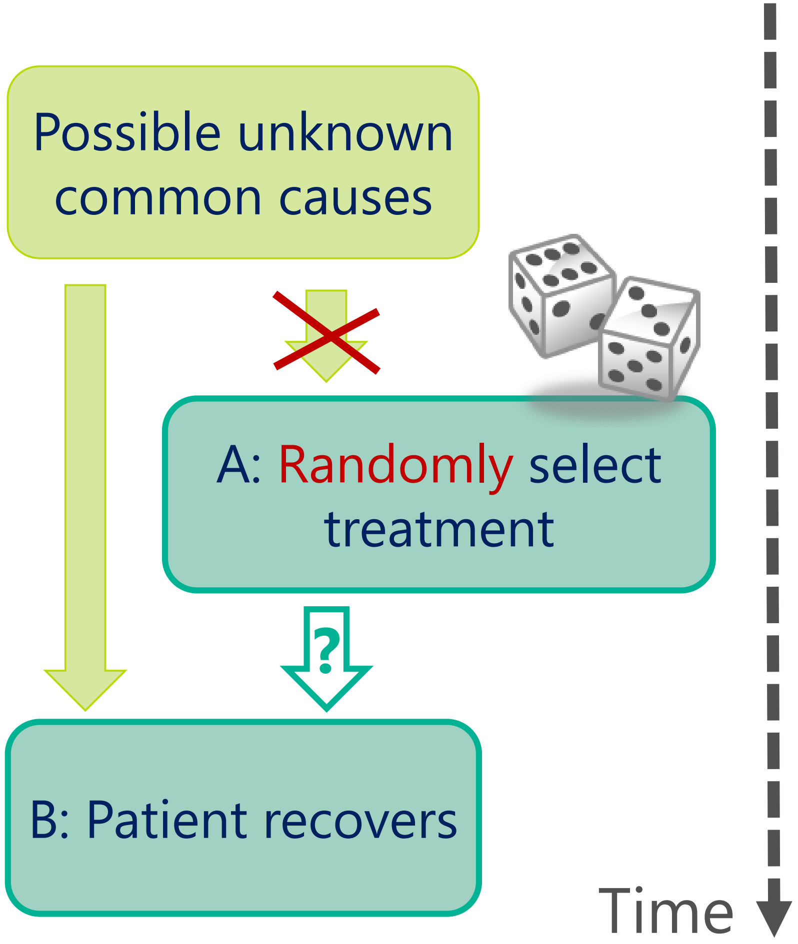
Randomized drug testing

Randomized experiment

Patient randomly receives drug or placebo.
Recovery rate are compared.

If a correlation is observed

- A causes B : possible.
- B causes A : no (direction of time)
- A, B have common causes : no (randomization)



Randomization, counterfactuals, etc.

Asking the correct question

Correlation question

Do we observe a higher suggestion take rate when certain conditions are true?

Manipulation question

Will we observe a higher suggestion take rate if we change the system in a certain manner?

Two kinds of manipulation questions

Hypothetical conditional

*"Will we observe a higher take rate
if we apply this change to the suggestion logic?"*

A/B Testing

Counterfactual conditional

*"Would we have observed a higher take rate
if we had applied this change to the suggestion
logic when the data was collected?"*

"Multi-world" Testing

Both can be answered using randomization.

A/B Testing

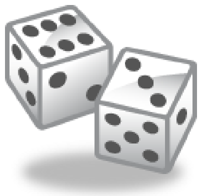
Formulate the question

"Will we observe a higher take rate if we apply this specific change to the suggestion logic?"

Run data collection experiment

Randomly decide which users receive

- normal treatment,
- or modified treatment.



Compare performance metrics.

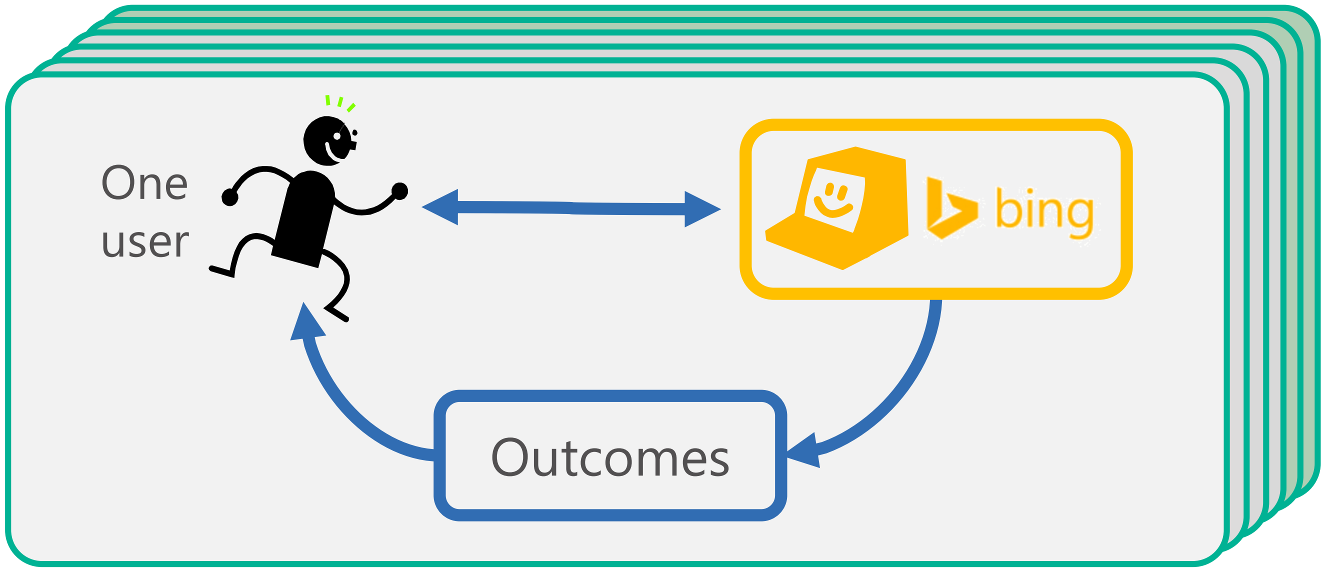
Same as randomized drug testing!

Isolated experimental units

Experiment isolation in search

"The experience of a Bing user does not affect other users."

→ We can safely apply different treatments to different users.

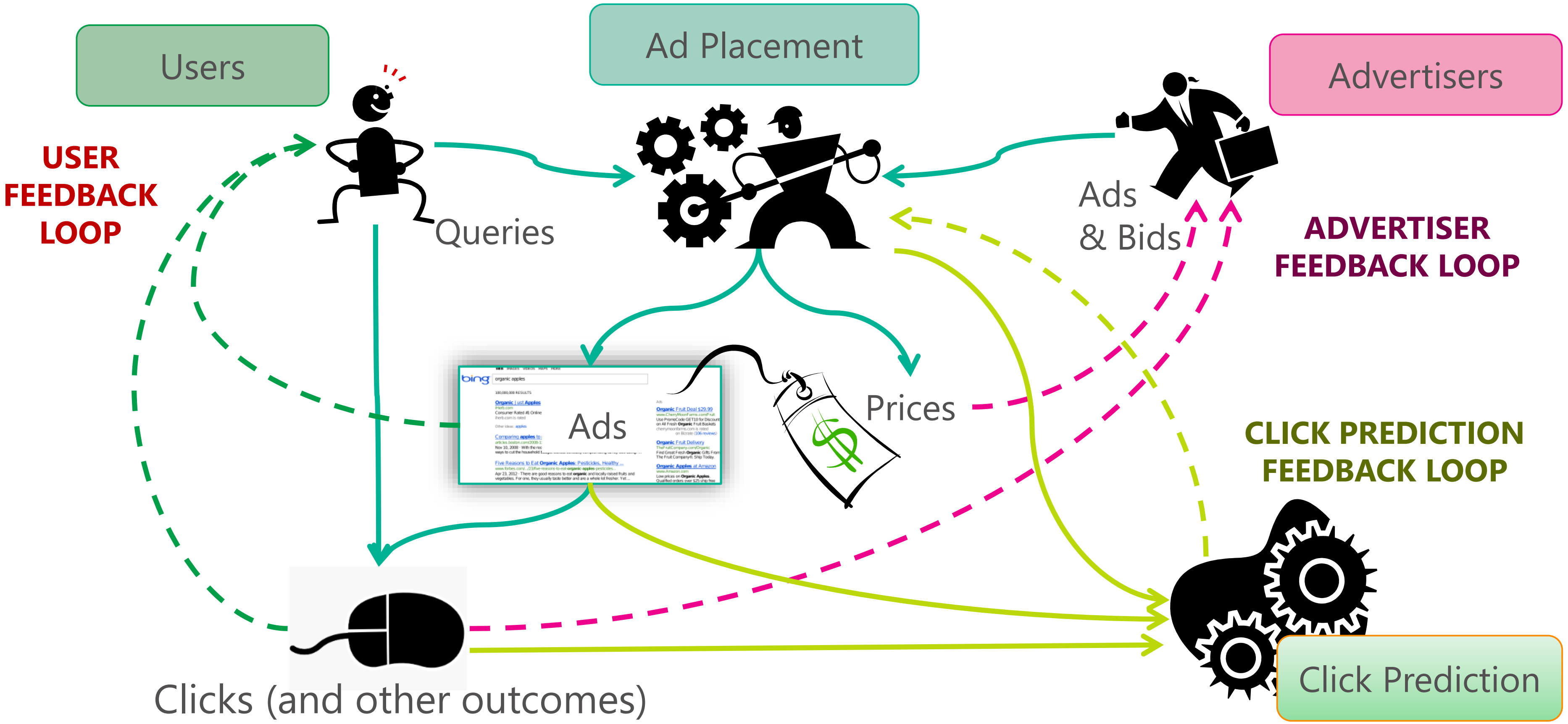


EXP

BingAB

Isolated experimental units

Experiment isolation is more problematic in ads



Experimentation throughput

A/B testing

- Formulate the question:

  "How will the performance metric move
  if we apply some specific change(s) to the product?"
- Code alternative treatments with product-grade quality,
- Allocate traffic for the data collection experiment,
- Collect data long enough to get meaningful results.

Bing Experimentation concludes ~30 experiments per day.

Multi-world testing

Offline counterfactual evaluation

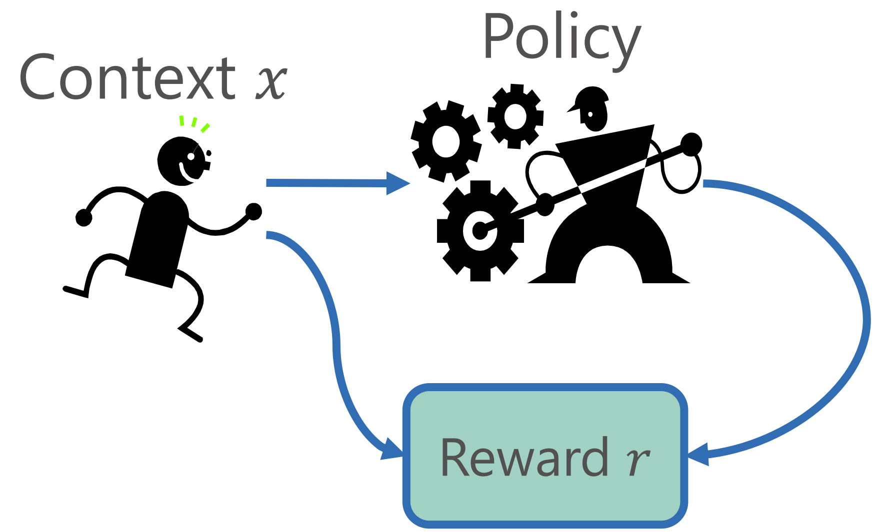
- Collect data during a carefully randomized experiment.
- Formulate the question:

  "How would the performance metric have moved if we had applied a specific change to the product when the data was collected?"
- Answer the question offline, using previously collected data.

> - Pay the data collection price only once.
> - Iterate very quickly because evaluation happens offline.

How does it work?

The case of the "contextual bandits"



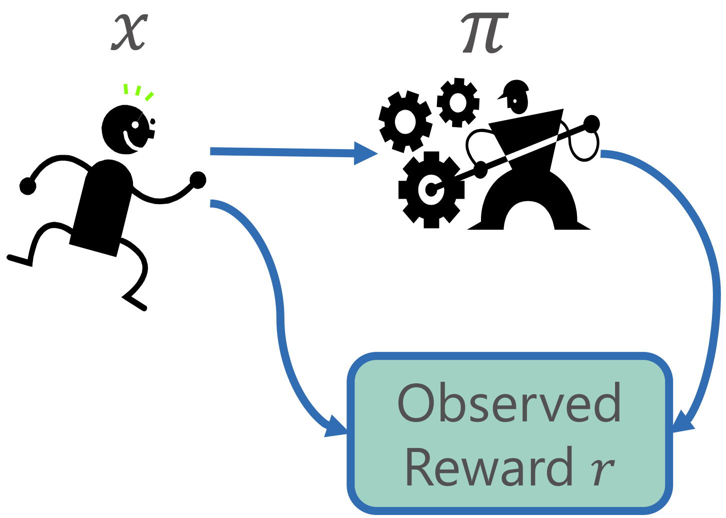
Action	Probability
a_1	10%
a_2	47%
a_3	25%
a_4	18%

- Policy computes the probability of each action.
- One action is randomly selected.
- The reward depends on both context and action.

How does it work?

Importance sampling

"What would have been the average reward if we had used policy π' instead of the data collection policy π?"



Action	π	π'
a_1	10%	3%
a_2	25%	12%
a_3	47%	35%
a_4	18%	50%

Observed reward r

- occurs with p=47% under policy π,
- occurs with p=35% under policy π'.

Estimate the average reward under policy π', by giving weight 35/47 to this reward r.

How does it work?

The general case

- can require a full fledged causal inference machinery,
- can involve multiple feedback loops,
- can involve equilibria analysis,

Fortunately

Importance sampling can go a long way.

Work in progress

A cloud service targeting contextual bandits-style problems.

Success stories

Yahoo (2011) – Personalized news.

AdCenter (2011) – "Metropolis" randomization.

LinkedIn (2013) – LinkedIn ad placement.

Bing (2014) – Optimization of click metrics in Bing Speller.

Li, Chu, Langford, Wang – "Unbiased offline evaluation …", WSDM 2011.
Bottou et al. – "Counterfactual reasoning and learning systems …", JMLR 2013.
Agarwal - Simons Foundation Workshop, Berkeley, 2013.
Li et al. – Submitted, KDD 2014.

Lessons

Design process

Envision the full spectrum of user interaction policies from the start.

Deploy simple policy with randomized exploration.

Use interaction data to tune the policy and learn how to “delight users”.

Reliable and verifiable logs

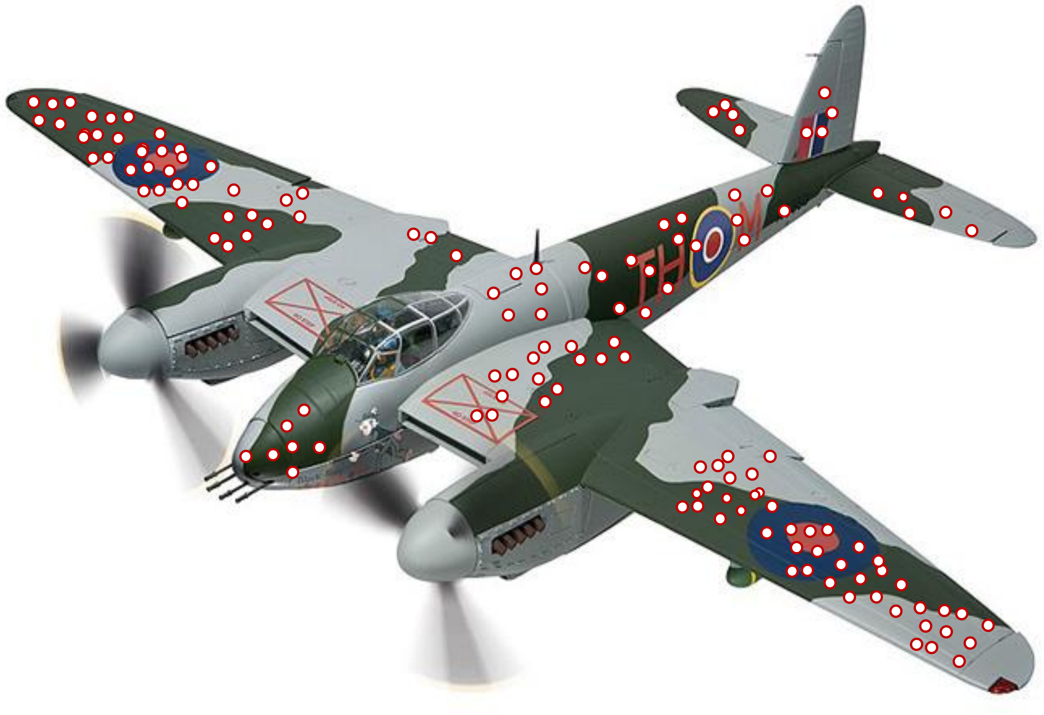
Logging is often under-appreciated.

Correctness and completeness of the logs is critical.

Example: logging outcomes rather than decisions…

A broader perspective

Counterfactual reasoning saves lives!



Conclusions

Summary

The opportunity

Large scale user interaction data offers
extraordinary opportunities to improve our products.

The difficulty

This is more complex than vanilla machine learning
because correlation does not imply causation.

The good news

MSR has world-class expertise and technology in this area.