A/B Testing of Stateful Algorithms: A Markov Chain Perspective

Ramesh Johari
Joint work with Jose Blanchet, Peter Glynn, Mohammad Rasouli, Linjia Wu
Stanford University
rjohari@stanford.edu

September 21, 2021

Motivation: Testing algorithms

Suppose you are one of these:

You have two algorithms A and B that you want to compare (e.g., matching algorithms).

Each algorithm changes the *state* of the system.

How do you design an experiment (A/B test) and an estimator to compare them?

Naive solution: Randomize over time

Goal: Estimate TE, the treatment effect – the difference in steady-state "reward" of the two algorithms.

Naive solution: Randomize over time

Goal: Estimate TE, the treatment effect – the difference in steady-state "reward" of the two algorithms.

Suppose at each decision epoch, we randomly flip a coin and run either A (heads) or B (tails).

Why is this not a good idea?

Naive solution: Randomize over time

Goal: Estimate TE, the treatment effect – the difference in steady-state "reward" of the two algorithms.

Suppose at each decision epoch, we randomly flip a coin and run either A (heads) or B (tails).

Why is this not a good idea?

Carryover effect ("temporal interference"): Each algorithm's action changes the *state* as seen by the other algorithm.

Therefore experimental units (time steps) *interfere* with each other, introducing *bias*.

One approach: Switchback designs

Many platforms (ridesharing, delivery marketplaces, etc.) use *switchback designs* to run A/B tests of algorithms:

1. Divide time into *fixed length non-overlapping intervals*.
2. In each successive interval, assign one of algorithm A or B.
3. Compute sample average estimate $\widehat{\mathsf{SAE}}_A$ and $\widehat{\mathsf{SAE}}_B$ of reward of A and B respectively.
4. Compute $\widehat{\mathsf{SAE}}_A - \widehat{\mathsf{SAE}}_B$ as *treatment effect estimate* $\widehat{\mathsf{TE}}$.

Note: Doesn't eliminate carryover effect.

Overview of this talk

- Testing two algorithms as a theoretical problem of *testing two Markov chains*
- *Consistent* and *sample efficient* experimental design using *nonparametric MLE*
- Challenges and opportunities
  - Computational complexity
  - The role of modeling
  - Alternative experimental designs
  - Novel applications

Preliminaries: A Markov chain approach

A Markov chain viewpoint

A key innovation in our approach:
We cast the testing problem as one of *estimating the difference in steady state reward between two Markov chains*.

Nonparametric model

- Discrete time $n = 0, 1, 2, \ldots$
- Finite state space S (x, y denote states)

Nonparametric model

- Discrete time $n = 0, 1, 2, \ldots$
- Finite state space S (x, y denote states)
- Two algorithms (actions) 1 and 2 (ℓ denotes algorithm)
- Unknown irreducible transition matrices $\boldsymbol{P}(\ell) = (P(\ell, x, y), x, y \in S)$
- Invariant distributions $\boldsymbol{\pi}(\ell) = (\pi(\ell, x), x \in S)$ (row vector)

Nonparametric model

- Discrete time $n = 0, 1, 2, \ldots$
- Finite state space S (x, y denote states)
- Two algorithms (actions) 1 and 2 (ℓ denotes algorithm)
- Unknown irreducible transition matrices $\boldsymbol{P}(\ell) = (P(\ell, x, y), x, y \in S)$
- Invariant distributions $\boldsymbol{\pi}(\ell) = (\pi(\ell, x), x \in S)$ (row vector)
- Unknown reward distribution $R \sim f(\cdot|\ell, x, y)$ (finite mean and variance)
- $r(\ell, x) = \mathbb{E}[R|\ell, x]$; $\boldsymbol{r}(\ell) = (r(\ell, x), x \in S)$ (column vector)

Nonparametric model

- Discrete time $n = 0, 1, 2, \ldots$
- Finite state space S (x, y denote states)
- Two algorithms (actions) 1 and 2 (ℓ denotes algorithm)
- Unknown irreducible transition matrices $\boldsymbol{P}(\ell) = (P(\ell, x, y), x, y \in S)$
- Invariant distributions $\boldsymbol{\pi}(\ell) = (\pi(\ell, x), x \in S)$ (row vector)
- Unknown reward distribution $R \sim f(\cdot|\ell, x, y)$ (finite mean and variance)
- $r(\ell, x) = \mathbb{E}[R|\ell, x]$; $\boldsymbol{r}(\ell) = (r(\ell, x), x \in S)$ (column vector)

At time n: State X_n, action A_n, reward R_n

The estimation problem

Treatment effect of interest is the *steady state reward difference*:

$$\begin{aligned}\alpha = \alpha(2) - \alpha(1) &= \sum_x \pi(2,x) r(2,x) - \sum_x \pi(1,x) r(1,x) \\ &= \boldsymbol{\pi}(2)\boldsymbol{r}(2) - \boldsymbol{\pi}(1)\boldsymbol{r}(1).\end{aligned}$$

The estimation problem

Treatment effect of interest is the *steady state reward difference*:

$$\begin{aligned}\alpha = \alpha(2) - \alpha(1) &= \sum_x \pi(2,x) r(2,x) - \sum_x \pi(1,x) r(1,x) \\ &= \boldsymbol{\pi}(2)\boldsymbol{r}(2) - \boldsymbol{\pi}(1)\boldsymbol{r}(1).\end{aligned}$$

We get to choose an estimator and a design:

- Estimator: $\alpha = (\alpha_n : n \geq 0)$, $\alpha_n \in \mathbb{R}$
- Design: $A = (A_n : n \geq 0)$, $A_n \in \{1, 2\}$

Estimator and design are adapted to history, and design can be randomized.

Nonparametric maximum likelihood estimation

Nonparametric maximum likelihood (NP MLE)

Definitions:

$$\Gamma_n(\ell, x) := \#\text{ of plays of } \ell \text{ in first } n \text{ steps} = \sum_{j=0}^{n-1} I(X_j = x, A_j = \ell)$$

$$r_n(\ell, x) := \text{SAE of } r(\ell, x) = \frac{\sum_{j=0}^{n-1} I(X_j = x, A_j = \ell) R_{j+1}}{\max\{\Gamma_n(\ell, x), 1\}}$$

$$P_n(\ell, x, y) := \text{SAE of } P(\ell, x, y) = \frac{\sum_{j=0}^{n-1} I(X_j = x, A_j = \ell, X_{j+1} = y)}{\max\{\Gamma_n(\ell, x), 1\}}$$

Let $\boldsymbol{\pi}_n(\ell)$ be invariant distribution of $\boldsymbol{P}_n(\ell)$ (exists a.s. as $n \to \infty$). Then:

$$\alpha_n^{\mathsf{MLE}} = \boldsymbol{\pi}_n(2)\boldsymbol{r}_n(2) - \boldsymbol{\pi}_n(1)\boldsymbol{r}_n(1).$$

Time-average regular policies

We optimize over time-average regular policies.

Definition

Design A is *time-average regular* if

$$\frac{1}{n}\Gamma_n(\ell, x) \xrightarrow{p} \gamma(\ell, x)$$

as $n \to \infty$ for each $x \in S, \ell = 1, 2$, and (possibly random) $\gamma(\ell, x)$.

We call $\gamma = (\gamma(\ell, x) : x \in S, \ell = 1, 2)$ the *policy limit*.

(For our theory we will require $\gamma(\ell, x) > 0$ a.s.)

Central limit theorem for NP MLE

Theorem

For any time-average regular design A with strictly positive policy limits:

$$n^{1/2}(\alpha_n^{MLE} - \alpha) \Rightarrow \sum_x \frac{\pi(2,x)\sigma(2,x)}{\gamma(2,x)^{1/2}} G(2,x) - \sum_x \frac{\pi(1,x)\sigma(1,x)}{\gamma(1,x)^{1/2}} G(1,x).$$

where:

- $G(\ell, x)$ are i.i.d. $N(0,1)$;
- $\sigma^2(\ell, x) = \mathrm{Var}\left(R_j + \tilde{g}(\ell, X_j) \mid X_{j-1} = x, A_{j-1} = \ell\right)$ (assume positive);
- $\tilde{\boldsymbol{g}}(\ell)$ solves the *Poisson equation*:

$$\tilde{\boldsymbol{g}}(\ell) = (\boldsymbol{I} - \boldsymbol{P}(\ell) + \boldsymbol{\Pi}(\ell))^{-1} \boldsymbol{r}(\ell)$$

- $\boldsymbol{\Pi}(\ell)$ is the matrix where each row is $\boldsymbol{\pi}(\ell)$.

Central limit theorem for NP MLE: Single chain

Key idea:

$$\begin{aligned}\alpha_n(\ell) - \alpha(\ell) &= \sum_x \pi_n(\ell, x) r_n(\ell, x) - \sum_x \pi(\ell, x) r(\ell, x) \\ &= \boldsymbol{\pi}_n(\ell)\big(\boldsymbol{r}_n(\ell) - \boldsymbol{r}(\ell)\big) + \big(\boldsymbol{\pi}_n(\ell) - \boldsymbol{\pi}(\ell)\big)\boldsymbol{r}(\ell) \\ &= \boldsymbol{\pi}_n(\ell)\big(\boldsymbol{r}_n(\ell) - \boldsymbol{r}(\ell)\big) + \boldsymbol{\pi}_n(\ell)\big(\boldsymbol{P}_n(\ell) - \boldsymbol{P}(\ell)\big)\tilde{\boldsymbol{g}}(\ell)\end{aligned}$$

We combine the preceding idea with martingale arguments to handle adaptive sampling.

Optimal oracle design for NP MLE

Let $\mathcal{K}$ be the (convex, compact) set of all $\big(\kappa(\ell, x) : x \in S, \ell = 1, 2\big)$ such that:

$$\kappa(1, y) + \kappa(2, y) = \sum_{\ell} \sum_{x} \kappa(\ell, x) P(\ell, x, y), \quad y \in S;$$

$$\sum_{\ell} \sum_{x} \kappa(\ell, x) = 1;$$

$$\kappa(\ell, x) \geq 0.$$

Lemma: The law of any time-average regular policy limit γ is a probability measure over $\mathcal{K}$.

Optimal oracle design for NP MLE

Let κ^* be the solution to the following convex optimization problem:

$$\begin{aligned} \text{minimize} \quad & \sum_{\ell} \sum_{x} \frac{\pi^2(\ell, x)\sigma^2(\ell, x)}{\kappa(\ell, x)} \\ \text{subject to} \quad & \kappa \in \mathcal{K}. \end{aligned}$$

Then κ^* can be realized as the policy limit of the following *stationary, Markov* design:

Run algorithm ℓ in state x with probability:

$$p^*(\ell, x) = \frac{\kappa^*(\ell, x)}{\kappa^*(1, x) + \kappa^*(2, x)}.$$

Optimal oracle design for NP MLE

Theorem

The design p^* minimizes the asymptotic variance of $n^{1/2}(\alpha_n^{MLE} - \alpha)$ over time-average regular policies.

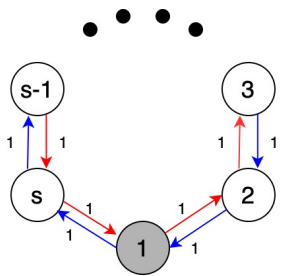
Proof idea: Use Jensen's inequality on asymptotic variance of $n^{1/2}\big(\alpha_n^{\mathsf{MLE}} - \alpha\big)$:

$$\mathbb{E}\Big[\sum_{\ell}\sum_{x}\frac{\pi^2(\ell,x)\sigma^2(\ell,x)}{\gamma(\ell,x)}\Big]$$

The value of cooperative exploration

Cooperative exploration: Two chains can yield much more efficient estimation than either chain alone.

Example: Suppose stochastic policy-dependent reward earned in state 1, and zero reward elsewhere.
Estimating using red or blue chain alone has asymptotic variance $\Theta(S)$ higher than using both together!



Optimal online design for NP MLE

Without knowledge of the primitives, we can compute $\kappa_n(\ell, x)$ as the optimal solution given $\boldsymbol{P}_n(\ell)$, and set:

$$p_n(\ell, x) = (1 - \epsilon_n) \left(\frac{\kappa_n(\ell, x)}{\kappa_n(1, x) + \kappa_n(2, x)} \right) + \frac{1}{2} \epsilon_n,$$

with $\epsilon_n = n^{-1/2}$ (forced exploration).

This yields the asymptotically optimal policy limits in an online fashion.

Challenges and opportunities

(a) Computational complexity

Computing the optimal policy using NP MLE estimation is clearly intractable in general.

A particular challenge arises because the state space of real-world queueing systems is in principle very large.

What are possible approaches to computational complexity?

(a) Computational complexity: Regenerative experiments

Fix a state x^r (the *regeneration* state).

Only change chains at visits to x^r; at each visit, choose ℓ with probability $p(\ell)$.

Estimate performance by sample averages.

(a) Computational complexity: Regenerative experiments

Fix a state x^r (the *regeneration* state).

Only change chains at visits to x^r; at each visit, choose ℓ with probability $p(\ell)$.

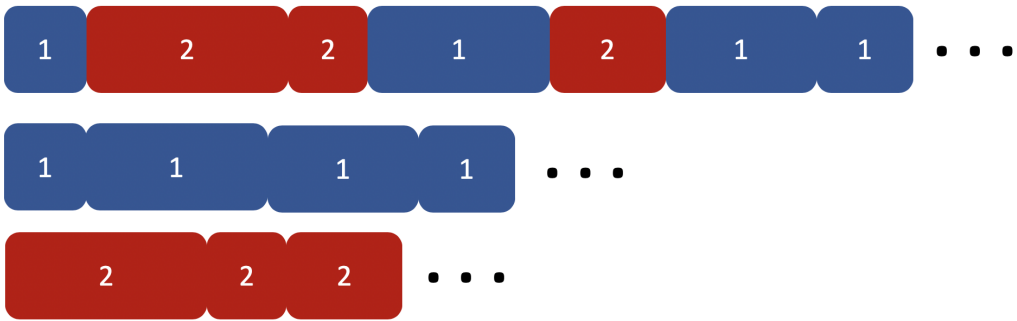
Estimate performance by sample averages.

(a) Computational complexity: Regenerative experiments

Fix a state x^r (the *regeneration* state).

Only change chains at visits to x^r; at each visit, choose ℓ with probability $p(\ell)$.

Estimate performance by sample averages.



(a) Computational complexity: Regenerative experiments

This approach offers the computational simplicity of switchback designs, but without the bias.

It is possible to similarly leverage a CLT to solve for the optimal regenerative design.

Unfortunately unboundedly worse variance w.r.t. optimal NP MLE design.

(b) The role of modeling

A key hallmark of research on queueing networks is the role of *modeling*.

We take this for granted, but it's important to note that this is not the norm in causal inference!

(b) The role of modeling

A key hallmark of research on queueing networks is the role of *modeling*.

We take this for granted, but it's important to note that this is not the norm in causal inference!

How does the causal inference literature approach interference in experimental design?

(b) The role of modeling

A key hallmark of research on queueing networks is the role of *modeling*.

We take this for granted, but it's important to note that this is not the norm in causal inference!

How does the causal inference literature approach interference in experimental design?

- Postulate a low-dimensional representation of impact of other units' treatment condition on a given unit's potential outcomes (*exposure model*).

(b) The role of modeling

A key hallmark of research on queueing networks is the role of *modeling*.

We take this for granted, but it's important to note that this is not the norm in causal inference!

How does the causal inference literature approach interference in experimental design?

- Postulate a low-dimensional representation of impact of other units' treatment condition on a given unit's potential outcomes (*exposure model*).
- Estimate treatment effect by estimating parameters in the exposure model.

See, e.g.: Aronow and Samii, 2013 on network experiments; Bojinov et al., 2020 on switchback experiments; Bajari et al., 2019, and Doudchenko et al., 2020, on marketplace experiments; and many others.

(b) The role of modeling

However, for queueing systems, appropriate exposure models can be hard to construct,
because interference between units typically grows exponentially in time.

Ongoing work in this area: combine the best of model-based experiment design and estimation, with the robustness of exposure modeling.

(b) The role of modeling: Parametric MLE

We have developed a similar approach to our NP MLE results in a setting where the Markov chains and rewards are *parameterized* by θ.

Example: For queueing control problems, parameters determine arrivals and service policy.

In this case similar techniques can be used to compute a parametric MLE for θ, and use this to estimate the difference in rewards.

This allows the possibility of *learning across states*, if multiple (ℓ, x) pairs provide information about $\theta \implies$ improvement in statistical efficiency.

(c) Comparing experimental designs

Many different approaches to dealing with interference:

- Switchback and crossover designs
- Cluster randomized designs
- Two-sided randomization (joint work with Hannah Li, Inessa Liskovich, and Gabriel Weintraub)

and many combinations of these, as well as novel estimation techniques.

However, currently lack guidance on conditions under which one design is preferable to another.

(d) Applications

As many industries digitize experimentation, the space of applications of such techniques is rapidly expanding.

E.g.: Communication networks, logistics and supply chain networks, etc.

(d) Applications

Recent collaboration with Netflix (with Bruce Spang, Nick McKeown, Te-Yuan Huang):

- When new network algorithms are A/B tested, they can create interference as the same underlying communication network is being shared
- We use switchback designs to show that this interference can be so significant that it *flips the sign* of treatment effect estimation

Many challenges in such applications, including combining *network partitioning* with *temporally adaptive experimental design*.

Related work

Related work

- *Crossover and switchback designs*

  Kershner and Federer (1981); Laska et al. (1983); Stufken (1996); Oman and Seiden (1988); Bojinov et al. (2020)

- *Mitigating network interference*

  Sobel (2006); Hudgens and Halloran (2008); Aronow and Samii (2013); Manski (2013); Ugander et al. (2013); Manski (2013); Eckles et al. (2017); Choi (2017); Baird et al. (2018); Athey et al. (2018); Basse et al. (2019)

- *Mitigating marketplace interference*

  Kohavi et al. (2009); Ostrovsky and Schwarz (2011); Bottou et al. (2013); Blake and Coey (2014); Basse et al. (2016); Wager and Xu (2019); Doudchenko et al. (2020); Johari et al. (2021); Candogan et al. (2021)

Related work (continued)

- *Estimation of a single Markov chain*

  Billingsley (1961); Kutoyants (2013)

- *Markov decision processes with minimum variance objectives*: Generally computationally intractable

  Sobel (1982, 1994); Di Castroet et al. (2012); Filar et al. (1989); Iancu et al. (2015); Mannor and Tsitsiklis (2011); Yu et al. (2018)

- *Pure exploration in reinforcement learning*: Focus on finding the best policy

  Brunskill et al. (2017); Putta and Tulabandhula (2017)

- *Offline policy evaluation in reinforcement learning*

  Precup et al. (2000), Dudik et al. (2015), Theocharous et al. (2015), Thomas and Brunskill (2016), etc.

Concluding thoughts

Looking ahead

The digitization of platforms has opened up a rich landscape of experimental design and estimation.

The applied probability community is uniquely poised to contribute to this area, given the particular need for a combination of mathematical modeling of queueing systems along with statistical guarantees.