Adaptive Experimental Design with Temporal Interference

Mohammad Rasouli — Peter Glynn — *Ramesh Johari*
Stanford University
rjohari@stanford.edu

24 May 2019

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

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

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?

Temporal interference: Each algorithm's action changes the *state* as seen by the other algorithm.

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

(Less naive) industry practice: 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 SAE_A and SAE_B of reward of A and B respectively.
4. Compute $\text{SAE}_A - \text{SAE}_B$ as *treatment effect estimate* $\widehat{\text{TE}}$.

Note: Doesn't eliminate temporal interference.

In their own words...

Experimentation in a Ridesharing Marketplace

Nicholas Chamandy Follow
Sep 2, 2016 · 9 min read

Part 1 of 3: Interference Across a Network

Technology companies strive to make data-driven product decisions—and Lyft is no exception. Because of that, online experimentation, or A/B testing, has become ubiquitous. The way it's bandied about, you'd be excused for thinking that online experimentation is a completely solved problem. In this post, we'll illustrate why that's far from the case for systems—like a ridesharing marketplace—that evolve according to network dynamics. As we'll see, naively partitioning users into treatment and control groups can bias the effect estimates you care about.

Switchback Tests and Randomized Experimentation Under Network Effects at DoorDash

DoorDash Follow
Feb 13, 2018 · 12 min read

David Kastelman, Data Scientist & Raghav Ramesh, Machine Learning Engineer

customers. As a result, this treatment algorithm may be observed to have a faster average time to completion even if it is not a better algorithm simply because it is cherry-picking short deliveries and leaving the control group to complete longer deliveries. Bias is more likely to occur if your regions get too small or you switch back too frequently; when we switchback less frequently,

Overview of our contributions

We cast the problem of testing two algorithms as one of *testing two Markov chains*.

We focus on *consistent* estimation of TE.

We develop two pairs of designs and consistent estimators:

- A Markov policy for allocation, together with a MLE for $\hat{\text{TE}}$, that is *sample efficient but not practical*.
- A regenerative policy for allocation, together with the SAE for $\hat{\text{TE}}$, that is *practical but not as sample efficient*.

Related work

- *Mitigating network interference*

  Sobel (2006); Hudgens and Halloran (2008); 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)

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.

Preliminaries

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 policy:

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

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

Maximum likelihood estimation

The maximum likelihood estimator

Definitions:

$$\Gamma_n(\ell, x) := \sum_{j=0}^{n-1} I(X_j = x, A_j = \ell)$$

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

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

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

Policy 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 MLE: Single chain

Suppose we sample only algorithm ℓ. Then $\gamma(\ell, x) = \pi(\ell, x)$, and:

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

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 following *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 MLE: Single chain

Key idea:

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

Central limit theorem for MLE: Adaptive sampling

Theorem

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

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

Proof intuition: Use the single chain argument, together with martingale arguments to handle adaptive sampling.

Optimal oracle policy for 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 policy for 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* policy:

Run algorithm ℓ in state x with probability:

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

Optimal oracle policy for MLE

Theorem

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

Optimal oracle policy for MLE: Proof idea

For a given policy, the asymptotic variance of $n^{1/2}\big(\alpha_n^{\mathsf{MLE}} - \alpha\big)$ is:

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

$$\geq \sum_{\ell}\sum_{x}\frac{\pi^2(\ell,x)\sigma^2(\ell,x)}{\mathbb{E}[\gamma(\ell,x)]} \quad \text{(via Jensen's inequality)}$$

$$\geq \inf_{\kappa\in\mathcal{K}}\sum_{\ell}\sum_{x}\frac{\pi^2(\ell,x)\sigma^2(\ell,x)}{\kappa(\ell,x)}.$$

Optimal online policy for 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.

Optimal online policy for 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.

But...computationally complex, and typically poor finite horizon performance.

Sample average estimation

Sample average estimation

Given a policy A, the sample average estimator is:

$$\alpha_n^{\mathsf{SAE}} = \frac{\sum_{j=0}^{n-1} I(A_j = 2) R_{j+1}}{\sum_{j=0}^{n-1} I(A_j = 2)} - \frac{\sum_{j=0}^{n-1} I(A_j = 1) R_{j+1}}{\sum_{j=0}^{n-1} I(A_j = 1)}$$

This estimator is computationally much less intensive.

However, it suffers from *temporal interference* every time the policy switches chains.

Regenerative policies

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

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

Regenerative policies

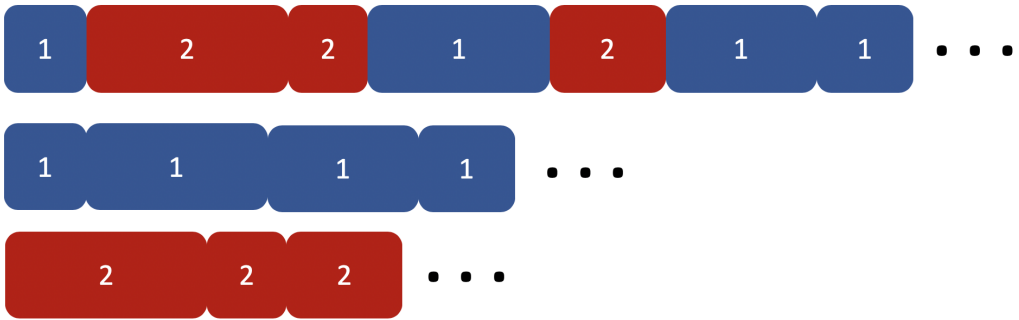
Fix a state x^r (the *regeneration* state).

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

Regenerative policies

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

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



Consistency and central limit theorem

SAE of regenerative policy is consistent asymptotically
(no temporal interference by design).

Can show: There exists $q(\ell)$ (depending on x^r and p) such that $q(1) + q(2) = 1$ and $\gamma(\ell, x) = q(\ell)\pi(\ell, x)$ for all ℓ, x.

$q(\ell)$ gives the fraction of time spent with chain ℓ.
(Can choose any q we want by varying p.)

Since *as if* we have two parallel runs of each chain,
convergence is at rate $n^{1/2}$ and CLT holds.

Optimal oracle regenerative policy

Easy to show that optimal oracle regenerative policy has:

$$q^*(\ell) = \frac{\overline{\sigma}(\ell)}{\overline{\sigma}(1) + \overline{\sigma}(2)},$$

where $\overline{\sigma}^2(\ell) = \sum_x \pi(\ell, x)\sigma^2(\ell, x)$.

Scaled asymptotic variance of this policy is $(\overline{\sigma}(1) + \overline{\sigma}(2))^2$ (achievable with *any* choice of x^r).

Can similarly construct an asymptotically equivalent online algorithm.

Optimal oracle regenerative policy

Easy to show that optimal oracle regenerative policy has:

$$q^*(\ell) = \frac{\overline{\sigma}(\ell)}{\overline{\sigma}(1) + \overline{\sigma}(2)},$$

where $\overline{\sigma}^2(\ell) = \sum_x \pi(\ell, x)\sigma^2(\ell, x)$.

Scaled asymptotic variance of this policy is $(\overline{\sigma}(1) + \overline{\sigma}(2))^2$ (achievable with *any* choice of x^r).

Can similarly construct an asymptotically equivalent online algorithm.

Unboundedly *suboptimal* in general relative to MLE with Markov optimal policy: There we had $|S|$ degrees of freedom vs. only one degree of freedom here.

Practical considerations: Future work

- Finite horizon performance
  - Prefer x^r with smaller cycle length
  - Use bias reduction techniques (future work)

Practical considerations: Future work

- Finite horizon performance
  - Prefer x^r with smaller cycle length
  - Use bias reduction techniques (future work)
- Large-scale experimentation
  - Single regeneration state unlikely to exist
  - State aggregation; regenerative set (semi-regenerative approach)
  - Combine with network clustering techniques for interference mitigation

Concluding thoughts

Summary and looking ahead

We proposed a benchmark model with which to evaluate sampling efficiency of consistent estimator-design pairs for switchback experimentation.

There are several considerations we have not addressed:

- Finite horizon analysis
- Multiple treatments
- Nonstationarity
- Heterogeneous treatment effects