Adaptive Experimental Design with Temporal Interference

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

3 June 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?*
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, introducing bias.
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 $\hat{\text{SAE}}_A$ and $\hat{\text{SAE}}_B$ of reward of $A$ and $B$ respectively.
4. Compute $\hat{\text{SAE}}_A - \hat{\text{SAE}}_B$ as treatment effect estimate $\hat{\text{TE}}$.

Note: Doesn’t eliminate temporal interference.
Overview of our contributions

We cast the problem of testing two algorithms as a theoretical problem of testing two Markov chains.

We focus on consistent estimation of TE.

- We develop a Markov policy for allocation, that together with a MLE for \( \hat{\text{TE}} \), is consistent and sample efficient.
- We develop a regenerative policy for allocation that is consistent when used with the SAE for \( \hat{\text{TE}} \) (but not 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 Castrogio 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 ($\ell$ denotes algorithm)
- Unknown irreducible transition matrices $P(\ell) = (P(\ell, x, y), x, y \in S)$
- Invariant distributions $\pi(\ell) = (\pi(\ell, x), x \in S)$ (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 ($\ell$ denotes algorithm)
- Unknown irreducible transition matrices $P(\ell) = (P(\ell, x, y), x, y \in S)$
- Invariant distributions $\pi(\ell) = (\pi(\ell, x), x \in S)$ (row vector)
- Unknown reward distribution $R \sim f(\cdot | \ell, x, y)$ (finite mean and variance)
- $r(\ell, x) = \mathbb{E}[R | \ell, x]$; $r(\ell) = (r(\ell, x), x \in S)$ (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 ($\ell$ denotes algorithm)
- Unknown irreducible transition matrices $P(\ell) = (P(\ell, x, y), x, y \in S)$
- Invariant distributions $\pi(\ell) = (\pi(\ell, x), x \in S)$ (row vector)
- Unknown reward distribution $R \sim f(\cdot | \ell, x, y)$ (finite mean and variance)
- $r(\ell, x) = \mathbb{E}[R | \ell, x]$; $r(\ell) = (r(\ell, x), x \in S)$ (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:

\[ \alpha = \alpha(2) - \alpha(1) = \sum_x \pi(2, x)r(2, x) - \sum_x \pi(1, x)r(1, x) \]
\[ = \pi(2)r(2) - \pi(1)r(1). \]
The estimation problem

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

\[
\alpha = \alpha(2) - \alpha(1) = \sum_x \pi(2, x)r(2, x) - \sum_x \pi(1, x)r(1, x)
\]

\[
= \pi(2)r(2) - \pi(1)r(1).
\]

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:

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

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

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

Let \( \pi_n(\ell) \) be invariant distribution of \( P_n(\ell) \) (exists a.s. as \( n \to \infty \)). Then:

\[ \alpha_n^{\text{MLE}} = \pi_n(2)r_n(2) - \pi_n(1)r_n(1). \]
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

Theorem

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

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

where:

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

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

- $\Pi(\ell)$ is the matrix where each row is $\pi(\ell)$. 
Central limit theorem for MLE: Single chain

Key idea:

$$\alpha_n(\ell) - \alpha(\ell) = \sum_x \pi_n(\ell, x) r_n(\ell, x) - \sum_x \pi(\ell, x) r(\ell, x)$$

$$= \pi_n(\ell) (r_n(\ell) - r(\ell)) + (\pi_n(\ell) - \pi(\ell)) r(\ell)$$

$$= \pi_n(\ell) (r_n(\ell) - r(\ell)) + \pi_n(\ell) (P_n(\ell) - P(\ell)) \tilde{g}(\ell)$$

We combine the preceding idea with martingale arguments to handle adaptive sampling.
Let $\mathcal{K}$ be the (convex, compact) set of all $(\kappa(\ell, x) : x \in S, \ell = 1, 2)$ 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 $\gamma$ is a probability measure over $\mathcal{K}$.
Optimal oracle policy for MLE

Let $\kappa^*$ be the solution to the following convex optimization problem:

$$\text{minimize} \quad \sum_{\ell} \sum_x \frac{\pi^2(\ell, x)\sigma^2(\ell, x)}{\kappa(\ell, x)}$$

subject to $\kappa \in \mathcal{K}$.

Then $\kappa^*$ can be realized as the policy limit of the following stationary, Markov policy:

Run algorithm $\ell$ 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 \( n^{1/2}(\alpha_n^{\text{MLE}} - \alpha) \) over time-average regular policies.

**Proof idea**: Use Jensen’s inequality on asymptotic variance of \( n^{1/2}(\alpha_n^{\text{MLE}} - \alpha) \):

\[
E \left[ \sum_{\ell} \sum_{x} \frac{\pi^2(\ell, x) \sigma^2(\ell, x)}{\gamma(\ell, x)} \right]
\]
The value of cooperative exploration

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

*Example:* Deterministic reward $r = 1$ in states 1, 2, 3, and zero reward elsewhere. Estimating red or blue chain alone has asymptotic variance $\Theta(S')$ higher than using both together!
Without knowledge of the primitives, we can compute $\kappa_n(\ell, x)$ as the optimal solution given $P_n(\ell)$, 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.
Sample average estimation
Sample average estimation

Given a policy $A$, the sample average estimator is:

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

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 $\ell$ 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 $\ell$ with probability $p(\ell)$. 

```
1 2 2 1 2 1 1 ...
```
Regenerative policies

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

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

---

1 2 2 1 2 1 1

1 1 1 1

2 2 2
Consistency and central limit theorem

SAE of regenerative policy is *consistent* (no temporal interference asymptotically 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 $\ell, x$.

$q(\ell)$ gives the fraction of time spent with chain $\ell$.
(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:

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

where \( \sigma^2(\ell) = \sum_x \pi(\ell, x)\sigma^2(\ell, x). \)

Scaled asymptotic variance of this policy is \( (\sigma(1) + \sigma(2))^2 \) (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.
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