Experimentation with Temporal Interference:
Poisson's Equation and Adaptive Markov Chain Sampling

Peter W. Glynn
Stanford University

Joint work with Ramesh Johari and Mohammad Rasouli

ICTS, Bangalore, August 16, 2019

Outline of Talk

1. What is "temporal interference"?
2. Discussion of related concepts:
   - Poisson's equation for Markov chains
   - Markov decision processes
3. Optimal adaptive Markov chain sampling

1. What is "Temporal Interference"?

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, 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 $\widehat{\text{SAE}}_A$ and $\widehat{\text{SAE}}_B$ of reward of A and B respectively.
4. Compute $\widehat{\text{SAE}}_A - \widehat{\text{SAE}}_B$ as *treatment effect estimate* $\widehat{\text{TE}}$.

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 $\widehat{\text{TE}}$, is *consistent* and *sample efficient*.
- We develop a *regenerative policy* for allocation that is *consistent* when used with the SAE for $\widehat{\text{TE}}$ (but not sample efficient).

2. Discussion of Related Concepts:

Poisson's Equation for Markov Chains

Markov Decision Processes

Poisson's Equation for Markov Chains

- $X = (X_n : n \geq 0)$, S-valued Markov chain, irreducible, $|S| < \infty$
- $P = (P(x, y) : x, y \in S)$ transition matrix

Given a function/column vector f, Poisson's equation is

$$(P - I)g = -f$$

For solvability: Need $\pi f = 0$

Poisson's Equation for Markov Chains

$$(P - I)g = -f$$

$$(P - I + \Pi)g = -f$$

$$g = (I - P + \Pi)^{-1} f \quad (\triangleq (I - B)^{-1} f)$$

$$(I - P + \Pi)^{-1} = \sum_{n=0}^{\infty} (P - \Pi)^n \quad \text{aperiodic setting}$$

$(I - P + \Pi)^{-1}$ exists in general

Remark: $(I - P + \Pi)^{-1}$ is known as the *fundamental matrix*

Poisson's Equation for Markov Chains

An application:

- Suppose we want to prove

$$\frac{1}{n}\sum_{j=0}^{n-1} f(X_j) \stackrel{a.s.}{\to} Ef(X_\infty)$$

as $n \to \infty$

Poisson's Equation for Markov Chains

An application:

- Suppose we want to prove

$$\frac{1}{n}\sum_{j=0}^{n-1} f(X_j) \overset{a.s.}{\to} Ef(X_\infty)$$

  as $n \to \infty$

- Try to write $\sum_{j=0}^{n-1} f(X_j) - nEf(X_\infty)$ in terms of a martingale

Poisson's Equation for Markov Chains

An application:

- Suppose we want to prove

$$\frac{1}{n}\sum_{j=0}^{n-1} f(X_j) \stackrel{a.s.}{\to} Ef(X_\infty)$$

  as $n \to \infty$

- Try to write $\sum_{j=0}^{n-1} f(X_j) - nEf(X_\infty)$ in terms of a martingale
- Put $f_c(x) = f(x) - Ef(X_\infty)$ and solve

$$(P - I)g = -f_c$$

Poisson's Equation for Markov Chains

- Note that

$$E\left[g(X_i) \mid \mathcal{F}_{i-1}\right] = \sum_y P(X_{i-1}, y)g(y) = (Pg)(X_{i-1})$$

so

$$D_i = g(X_i) - (Pg)(X_{i-1})$$

is a martingale difference

Poisson's Equation for Markov Chains

- Note that

$$E\left[g(X_i) \mid \mathcal{F}_{i-1}\right] = \sum_y P(X_{i-1}, y) g(y) = (Pg)(X_{i-1})$$

  so

$$D_i = g(X_i) - (Pg)(X_{i-1})$$

  is a martingale difference

- So, $M_n = \sum_{i=1}^n D_i$ is a martingale

Poisson's Equation for Markov Chains

- Note that

$$E\left[g(X_i) \mid \mathcal{F}_{i-1}\right] = \sum_y P(X_{i-1}, y)g(y) = (Pg)(X_{i-1})$$

so

$$D_i = g(X_i) - (Pg)(X_{i-1})$$

is a martingale difference

- So, $M_n = \sum_{i=1}^n D_i$ is a martingale
- But

$$\begin{aligned} M_n &= \sum_{i=1}^{n}[g(X_i) - (Pg)(X_{i-1})] \\ &= \sum_{i=0}^{n-1}[g(X_i) - (Pg)(X_i)] + g(X_n) - g(X_0) \\ &= \sum_{i=0}^{n-1} f_c(X_i) + g(X_n) - g(X_0) \quad \text{(recall: } (P-I)g = -f_c) \end{aligned}$$

Poisson's Equation for Markov Chains

- So,

$$\frac{1}{n}\sum_{i=0}^{n-1} f_c(X_i) = \frac{1}{n}M_n + \frac{1}{n}g(X_0) - \frac{1}{n}g(X_n)$$

Poisson's Equation for Markov Chains

- So,

$$\frac{1}{n}\sum_{i=0}^{n-1} f_c(X_i) = \frac{1}{n}M_n + \frac{1}{n}g(X_0) - \frac{1}{n}g(X_n)$$

- Martingale theory:

$$\frac{1}{n}M_n \overset{a.s.}{\to} 0$$

Poisson's Equation for Markov Chains

The Central Limit Theorem (CLT) for Markov Chains:

$$\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1} f_c(X_i) = \frac{1}{\sqrt{n}}M_n + \frac{1}{\sqrt{n}}g(X_0) - \frac{1}{\sqrt{n}}g(X_n)$$

- Martingale CLT implies:

$$\frac{1}{\sqrt{n}}\sum_{i=0}^{n-1} f_c(X_i) \Rightarrow \sigma N(0,1)$$

where

$$\sigma^2 = \mathsf{var}_\pi D_1 = E_\pi g^2(X_0) - E_\pi (Pg)^2(X_0)$$

Poisson's Equation for Markov Chains

Many other applications of Poisson's equation:

- stochastic control
- gradients of $E^\theta f(X_\infty)$
- non-stationary Markov chains

Markov Decision Processes

- Compute an optimal policy/control minimizing

$$\overline{\lim_{n\to\infty}} \frac{1}{n} \sum_{j=0}^{n-1} c(X_j, A_j)$$

over all adapted policies $(A_n : n \geq 0)$

Markov Decision Processes

- Compute an optimal policy/control minimizing

$$\overline{\lim_{n\to\infty}} \frac{1}{n} \sum_{j=0}^{n-1} c(X_j, A_j)$$

  over all adapted policies $(A_n : n \geq 0)$

- The optimality equation is:

$$v(x) + \gamma = \min_a \left[c(x,a) + \sum_y P_a(x,y) v(y) \right]$$

Markov Decision Processes

- The *value function* can alternatively be computed as the solution to a linear program (LP)

Markov Decision Processes

- The *value function* can alternatively be computed as the solution to a linear program (LP)
- The dual to this LP is:

$$\begin{aligned} \min_{\pi} \quad & \sum_{x,a} c(x,a)\pi(x,a) \\ \text{s/t} \quad & \sum_{a} \pi(y,a) = \sum_{x,a'} \pi(x,a')P_{a'}(x,y) \\ & \pi(x,a) \geq 0, \;\; \forall x, a \\ & \sum_{x,a} \pi(x,a) = 1 \end{aligned}$$

Markov Decision Processes

- The *value function* can alternatively be computed as the solution to a linear program (LP)
- The dual to this LP is:

$$
\begin{aligned}
\min_{\pi} \quad & \sum_{x,a} c(x,a)\pi(x,a) \\
\text{s/t} \quad & \sum_{a} \pi(y,a) = \sum_{x,a'} \pi(x,a')P_{a'}(x,y) \\
& \pi(x,a) \geq 0, \ \ \forall x, a \\
& \sum_{x,a} \pi(x,a) = 1
\end{aligned}
$$

- Optimal policy: Choose action a in state x with probability

$$
\frac{\pi(x,a)}{\sum_{a'} \pi(x,a')}
$$

3. Optimal Adaptive Markov Chain Sampling

Non-parametric Model

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

Non-parametric 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)

Non-parametric 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)
- $\boldsymbol{r}(\ell) = (r(\ell, x), x \in S)$ (column vector)

Non-parametric 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)
- $\boldsymbol{r}(\ell) = (r(\ell, x), x \in S)$ (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*:

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

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.

The Non-parametric Maximum Likelihood Estimator

Definitions:

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

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

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

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

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:

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

where:

- $G(\ell, x)$ are i.i.d. $N(0,1)$;
- $\sigma^2(\ell, x) = \mathrm{Var}\left(\tilde{g}(\ell, X_j) \mid X_{j-1} = x, A_{j-1} = \ell\right)$ (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 $\boldsymbol{\pi}(\ell)$.

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:

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

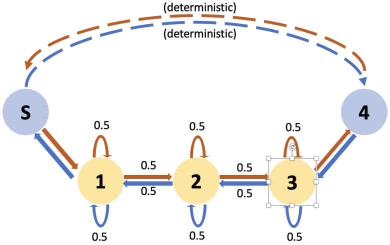
Optimal Oracle Policy for MLE

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

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!



Outline of Argument

- $(\pi_n(i) - \pi(i))(I - P(i) + \pi(i)) = \pi_n(i)(P_n(i) - P(i))$

Outline of Argument

- $(\pi_n(i) - \pi(i))(I - P(i) + \pi(i)) = \pi_n(i)(P_n(i) - P(i))$
- $$\begin{aligned} \pi_n(i) - \pi(i) &\approx \pi(i)(P_n(i) - P(i))(I - P(i) + \pi(i))^{-1} r(i) \\ &= \pi(i)(P_n(i) - P(i)) g(i) \end{aligned}$$

  where $g(i)$ is the solution to Poisson's equation for $r_c(i)$

Outline of Argument

- $(\pi_n(i) - \pi(i))(I - P(i) + \pi(i)) = \pi_n(i)(P_n(i) - P(i))$
- $$\begin{aligned}\pi_n(i) - \pi(i) &\approx \pi(i)(P_n(i) - P(i))(I - P(i) + \pi(i))^{-1} r(i) \\ &= \pi(i)(P_n(i) - P(i))g(i)\end{aligned}$$

  where $g(i)$ is the solution to Poisson's equation for $r_c(i)$
- $$((P_n(i) - P(i))g(i))\,(x) = \frac{\sum_{j=1}^{n} D_j(i,x)}{\pi_n(i,x)}$$

  where

  $$D_j(i,x) = I(X_{j-1} = x, A_{j-1} = i)\left[g(i, X_j) - (P(i)g(i))(X_{j-1})\right]$$

  is a martingale difference

Outline of Argument

- $(\pi_n(i) - \pi(i))(I - P(i) + \pi(i)) = \pi_n(i)(P_n(i) - P(i))$
- $$\begin{aligned} \pi_n(i) - \pi(i) &\approx \pi(i)(P_n(i) - P(i))(I - P(i) + \pi(i))^{-1} r(i) \\ &= \pi(i)(P_n(i) - P(i))g(i) \end{aligned}$$

  where $g(i)$ is the solution to Poisson's equation for $r_c(i)$
- $$\left((P_n(i) - P(i))g(i)\right)(x) = \frac{\sum_{j=1}^{n} D_j(i,x)}{\pi_n(i,x)}$$

  where

  $$D_j(i,x) = I(X_{j-1} = x, A_{j-1} = i)\left[g(i, X_j) - (P(i)g(i))(X_{j-1})\right]$$

  is a martingale difference
- Now, apply martingale CLT

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.

Analysis easily extends to random rewards $R_n(i)$

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

Three lectures hinting at the range of different problems of interest to the OR/MS applied probability community:

- output analysis for Monte Carlo
- Markov chains and processes in the presence of time-of-day effects, day-of-week effects, etc.
- A/B testing and temporal interference

Three lectures hinting at the range of different problems of interest to the OR/MS applied probability community:

- output analysis for Monte Carlo
- Markov chains and processes in the presence of time-of-day effects, day-of-week effects, etc.
- A/B testing and temporal interference

Just the "tip of the iceberg"

Thank you!