

Statistical Inference in Reinforcement Learning

Chengchun Shi

Assistant Professor of Data Science
London School of Economics and Political Science

Developing AI with reinforcement learning



THE ULTIMATE GO CHALLENGE

GAME 3 OF 3

27 MAY 2017



vs



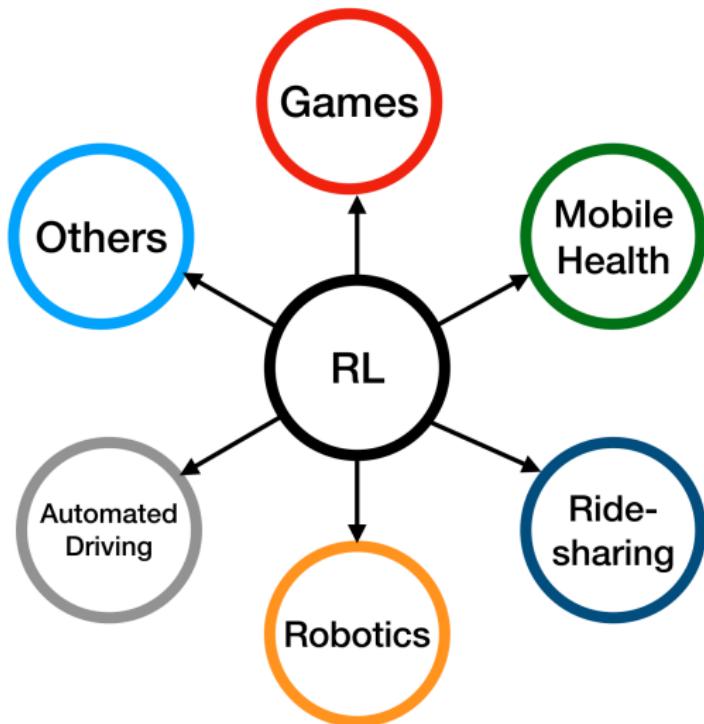
AlphaGo

Winner of Match 3

Ke Jie

RESULT B + Res

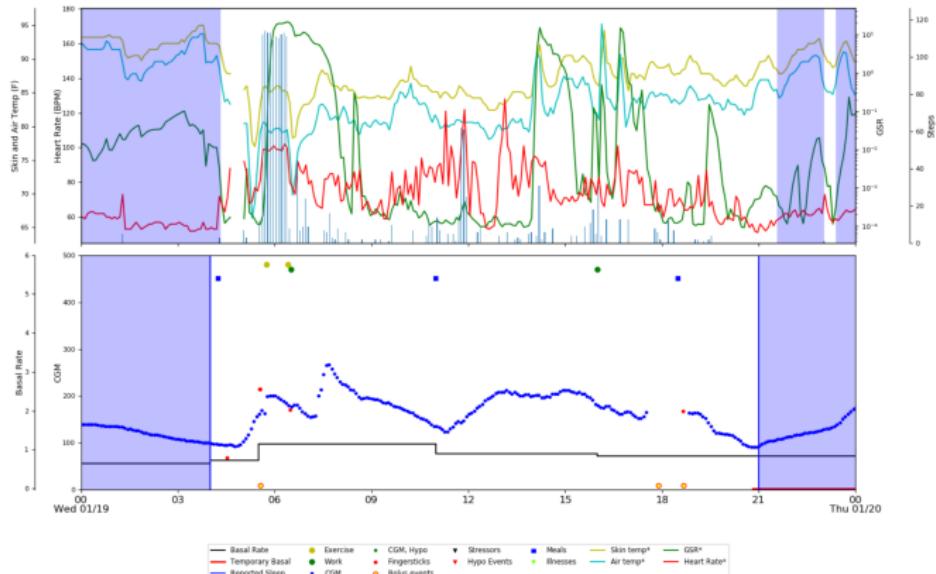
Reinforcement learning applications



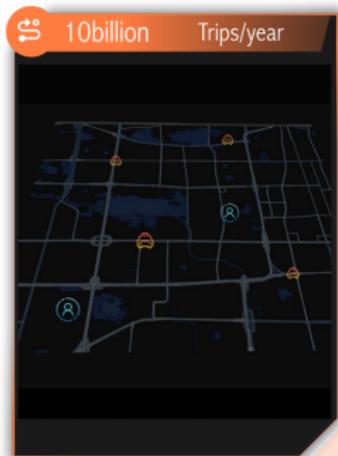
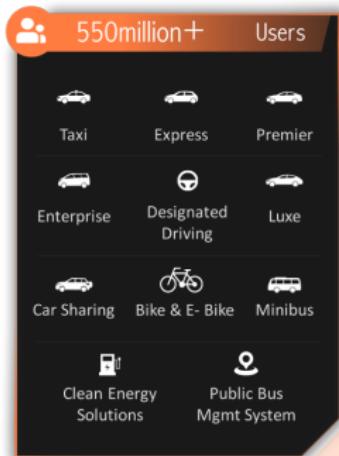
We focus on applications in **mobile health** and **ride-sharing**.

Applications in mobile health (mHealth)

- Management of **Type-I diabetes**.
- **Subject:** Patients with Type-I diabetes.
- **Intervention:** Determine whether a patient needs to inject insulin or not based on their glucose levels, food intake, etc.
- **Data:** OhioT1DM dataset (Marling and Bunescu, 2018)



Applications in ridesharing



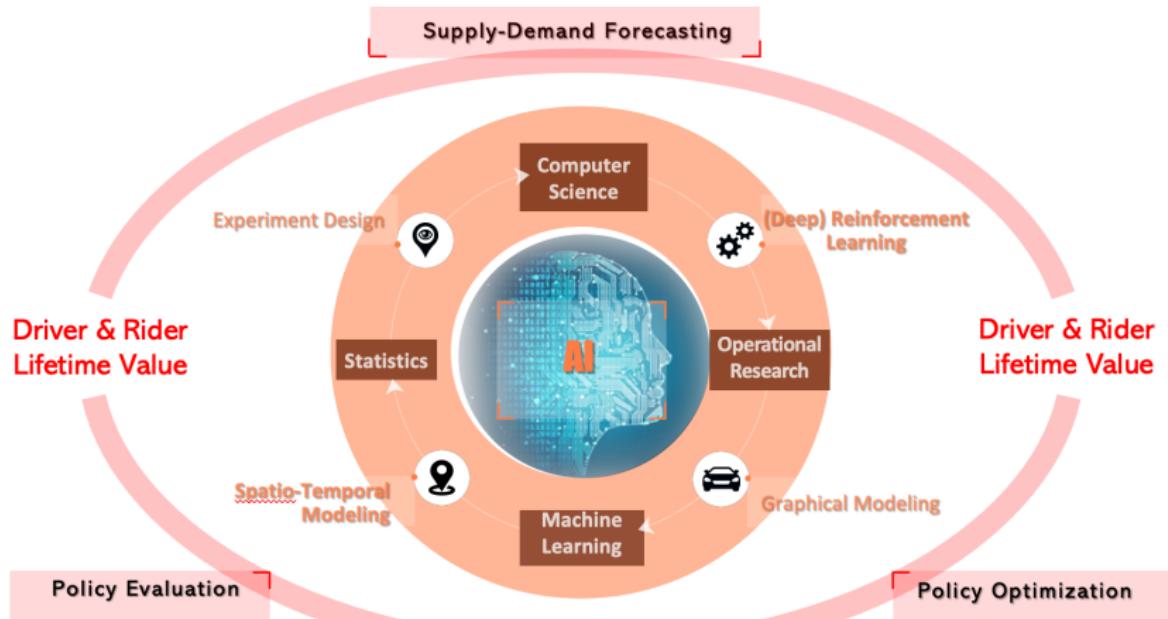
106TB+
vehicle trajectory data/day

4875TB+
data processed/day

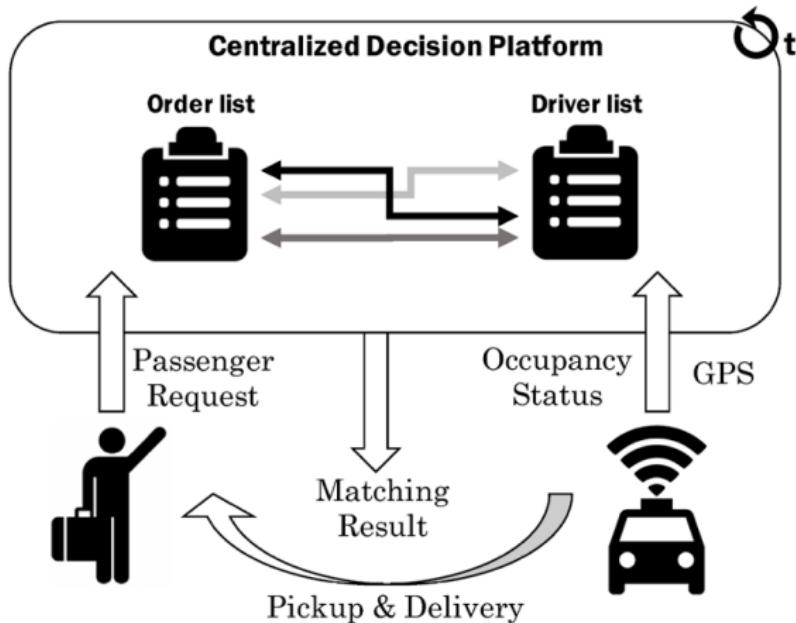
40billion+
routing requests/day

15billion+
location points/day

Applications in ridesharing (Cont'd)



Order dispatch



In this talk, we will focus on...

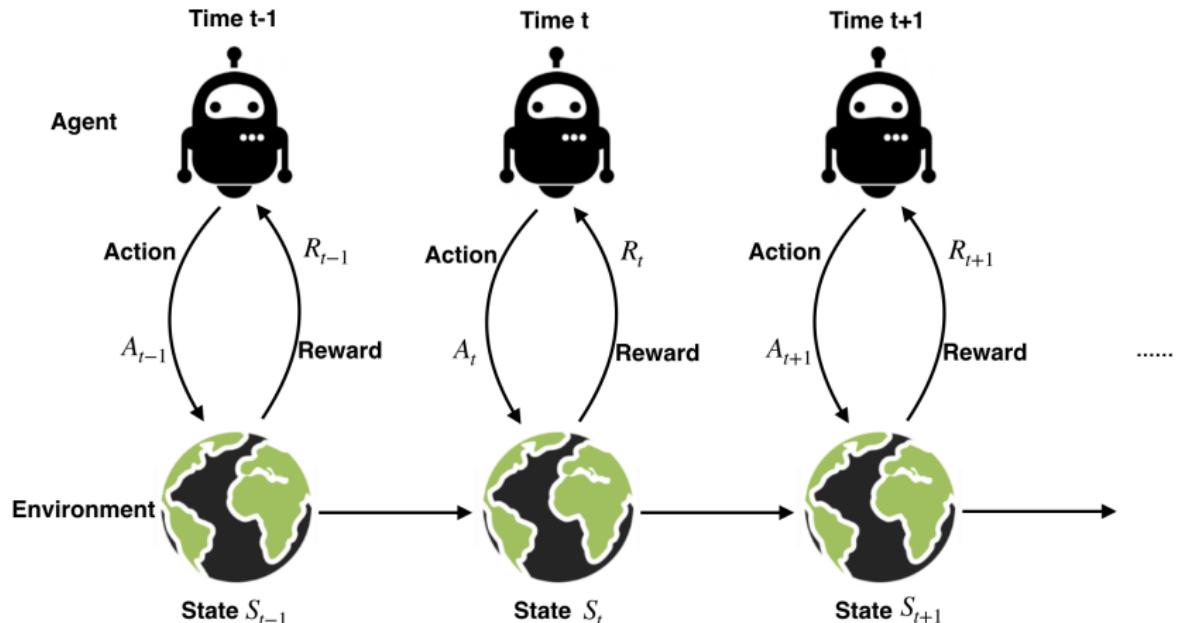
- **Statistical inference in reinforcement learning (RL).**
- Is statistical inference useful for RL?

Does the Markov Decision Process Fit the Data: Testing for the Markov Property in Sequential Decision Making

joint work with Runzhe Wan, Rui Song, Wenbin Lu (NCSU) and Ling Leng (Amazon)

—*ICML, 2020.*

Sequential decision making



Objective: find an optimal policy that maximizes the cumulative reward

The agent's policy

- The agent implements a **mapping** from the observed data to a probability distribution over actions at each time step
- The collection of these mappings $\pi = \{\pi_t\}_t$ is called **the agent's policy**:

$$\pi_t(a, \bar{s}) = \Pr(\textcolor{red}{A}_t = a | \bar{S}_t = \bar{s}),$$

where $\bar{S}_t = (\textcolor{blue}{S}_t, \textcolor{red}{A}_{t-1}, S_{t-1}, \dots, \textcolor{red}{A}_0, S_0)$ is the set of observed state-action history up to time t

- **History-dependent** policy: π_t depends on \bar{S}_t
- **Markov** policy: π_t depends on \bar{S}_t only through S_t , $\forall t$
- **Stationary** policy: π is Markov & π_t is homogeneous in t , $\forall t$

The Agent's Policy (Cont'd)



Reinforcement learning

- **RL algorithms:** trust region policy optimization (Schulman et al., 2015), deep Q-network (DQN, Mnih et al., 2015), asynchronous advantage actor-critic (Minh et al., 2016), quantile regression DQN (Dabney et al., 2018).
- **Foundations** of RL:
 - **Markov decision process** (MDP, Puterman, 1994): ensures the optimal policy is *stationary*, and is *not* history-dependent.
 - **Markov assumption** (MA): conditional on the present, the future and the past are independent,

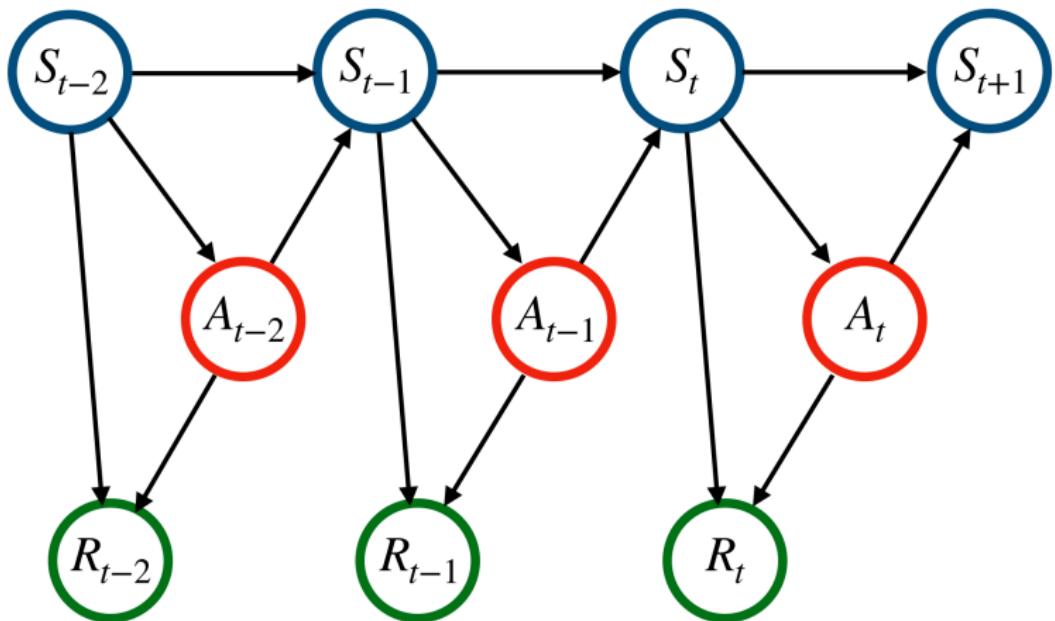
$$S_{t+1}, R_t \perp\!\!\!\perp \{(S_j, A_j, R_j)\}_{j < t} | S_t, A_t.$$

When R_t is a deterministic function of (S_t, A_t, S_{t+1}) :

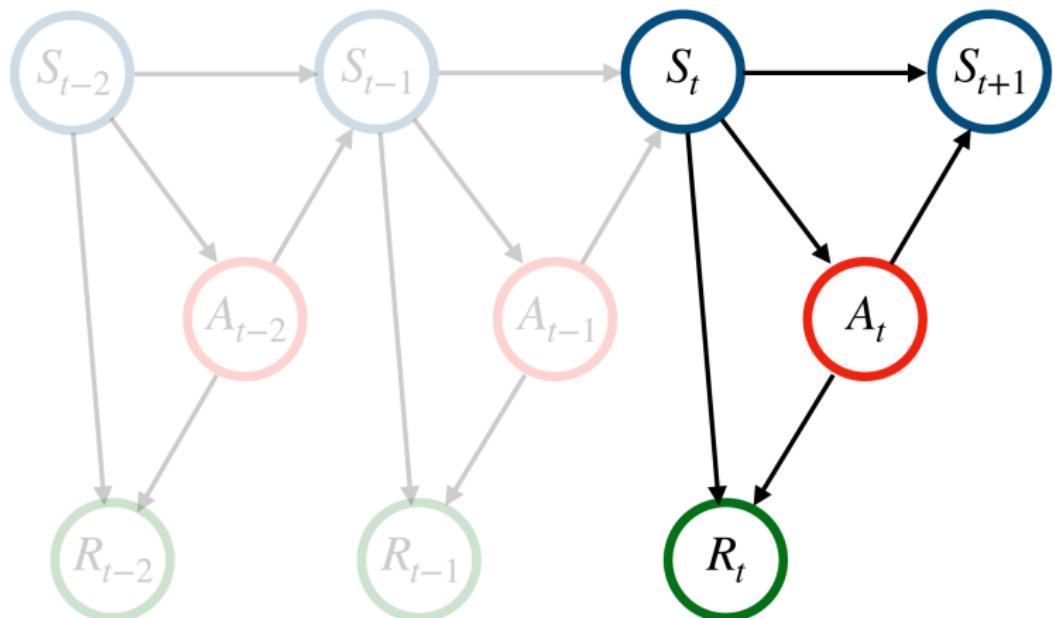
$$S_{t+1} \perp\!\!\!\perp \{(S_j, A_j)\}_{j < t} | S_t, A_t.$$

The Markov transition kernel is homogeneous in time.

Markov Decision Process



Markov Decision Process



RL models

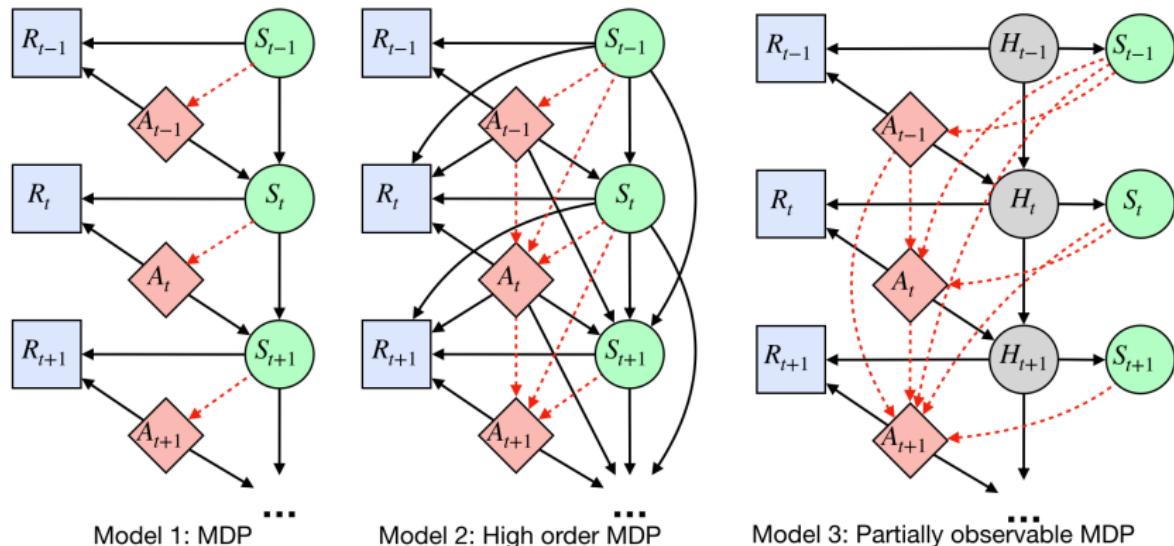


Figure: Causal diagrams for MDPs, HMDPs and POMDPs. The solid lines represent the causal relationships and the dashed lines indicate the information needed to implement the optimal policy. $\{H_t\}_t$ denotes latent variables.

Contributions

- **Methodologically**
 - propose a **forward-backward learning** procedure to test MA;
 - **first** work on developing consistent tests for MA in RL;
 - sequentially apply the proposed test for RL **model selection**;
 - critical to **offline** domains:
 - For **under-fitted** models, any stationary policy is not optimal;
 - For **over-fitted** models, the estimated policy might be very noisy due to the inclusion of many irrelevant lagged variables.
- **Empirically**
 - identify the optimal policy in **high-order** MDPs;
 - detect **partially observable** MDPs.
- **Theoretically**
 - prove our test **controls type-I error** under a **bidirectional** asymptotic framework.

Applications in high-order MDPs

- **Data:** the OhioT1DM dataset (Marling & Bunescu, 2018).
- Measurements for 6 patients with type I diabetes over 8 weeks.
- One-hour interval as a time unit.
- **State:** patients' time-varying variables, e.g., glucose levels, food intake, exercise intensity.
- **Action:** to inject insulin or not.
- **Reward:** the Index of Glycemic Control (Rodbard, 2009).

Applications in high-order MDPs (Cont'd)

- **Analysis I:**

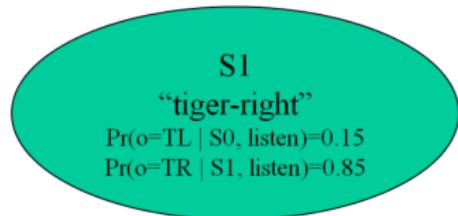
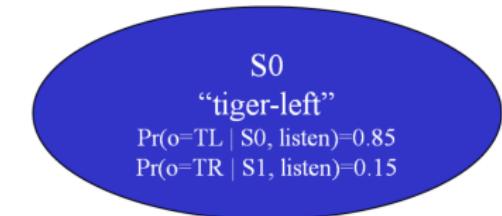
- sequentially apply our test to determine the order of MDP;
- conclude it is a **fourth-order** MDP.

- **Analysis II:**

- split the data into training/testing samples;
- policy optimization based on fitted-Q iteration (Ernst et al., 2005), by assuming it is a k -th order MDP for $k = 1, \dots, 10$;
- policy evaluation based on fitted-Q evaluation (Le et al., 2019);
- use random forest to model the Q-function;
- repeat the above procedure to compute the average value of policies computed under each MDP model assumption.

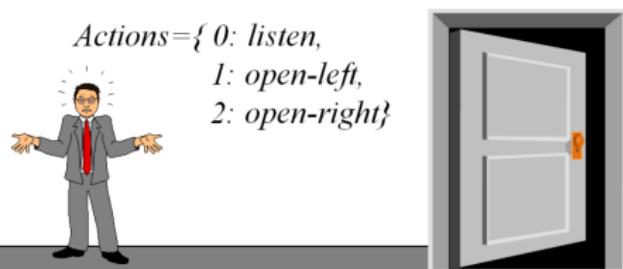
order	1	2	3	4	5	6	7	8	9	10
value	-90.8	-57.5	-63.8	-52.6	-56.2	-60.1	-63.7	-54.9	-65.1	-59.6

Applications in partially observable MDPs



Reward Function

- Penalty for wrong opening: -100
- Reward for correct opening: +10
- Cost for listening action: -1

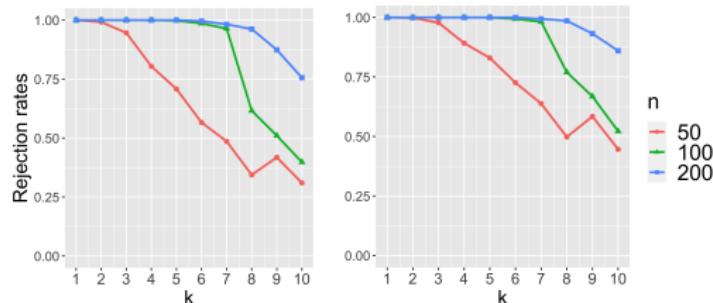


Observations

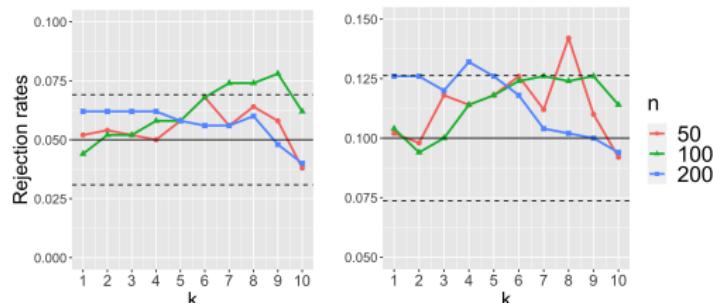
- to hear the tiger on the left (TL)
- to hear the tiger on the right (TR)

Applications in partially observable MDPs (Cont'd)

- Empirical rejection rates under the alternative hypothesis (MA is violated). $\alpha = (0.05, 0.1)$ from left to right.



- Empirical rejection rates under the null hypothesis (MA holds). $\alpha = (0.05, 0.1)$ from left to right.



Methodology

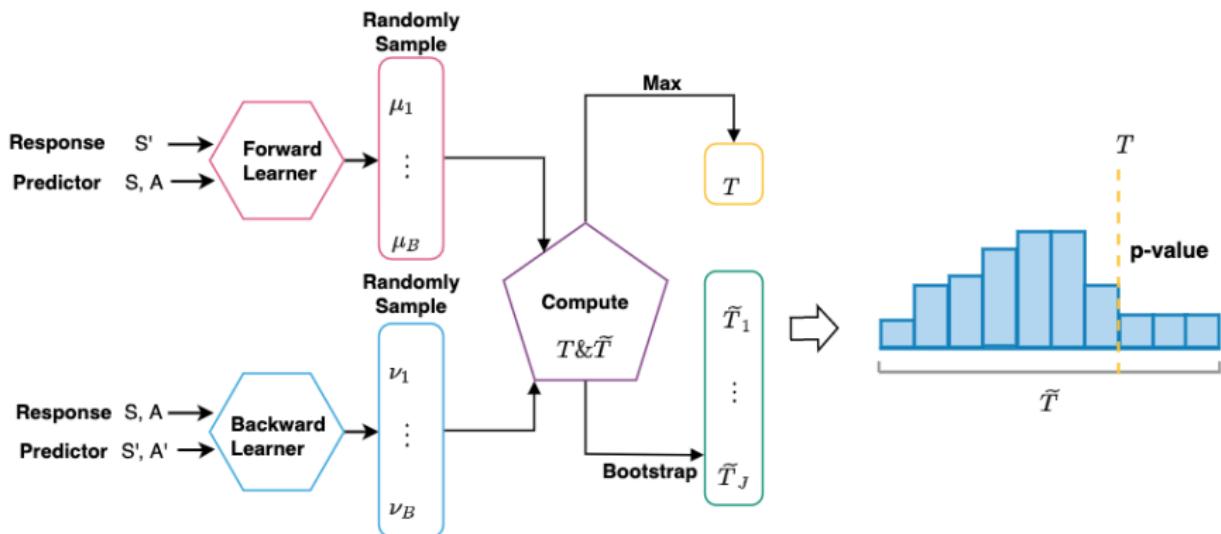
- **First work** to test MA in sequential decision making
- Existing approach in time series: Cheng and Hong (2012)
 - characterize MA based on the notion of **conditional characteristic function** (CCF);
 - use local polynomial regression to estimate CCF.
- Challenge:
 - develop a valid test for MA in **moderate or high-dimensions**
 - the dimension of the state increases as we concatenate measurements over multiple time points in order to test for a high-order MDP.
- This motivates our **forward-backward learning** procedure.

Methodology (Cont'd)

Some key components of our algorithm:

- To deal with moderate or high-dimensional state space, employ modern machine learning (ML) algorithms to estimate CCF:
 - Learn CCF of S_{t+1} given A_t and S_t (**forward learner**);
 - Learn CCF of (S_t, A_t) given (S_{t+1}, A_{t+1}) (**backward learner**);
 - Develop a random forest-based algorithm to estimate CCF;
 - Borrow ideas from the quantile random forest algorithm (Meinshausen, 2006) to facilitate the computation.
- To alleviate the bias of ML algorithms, construct **doubly-robust** estimating equations by integrating forward and backward learners;
- To improve the power, construct a **maximum-type** test statistic;
- To control the type-I error, approximate the distribution of our test via **multiplier bootstrap** (Chernozhukov, et al., 2014).

Algorithm



Bidirectional theory

- N the number of trajectories;
- T the number of decision points per trajectory;
- bidirectional asymptotics: a framework allows either N or $T \rightarrow \infty$;
- large T , small N (OhioT1DM dataset)



- large N , small T (Framingham Heart Study)



- large N , large T (games)

Bidirectional theory (cont'd)

- (C1) Actions are generated by a fixed behavior policy.
- (C2) The process $\{S_t\}_{t \geq 0}$ is exponentially β -mixing.
- (C3) The ℓ_2 prediction errors of forward and backward learners converge at a rate faster than $(NT)^{-1/4}$.

Theorem

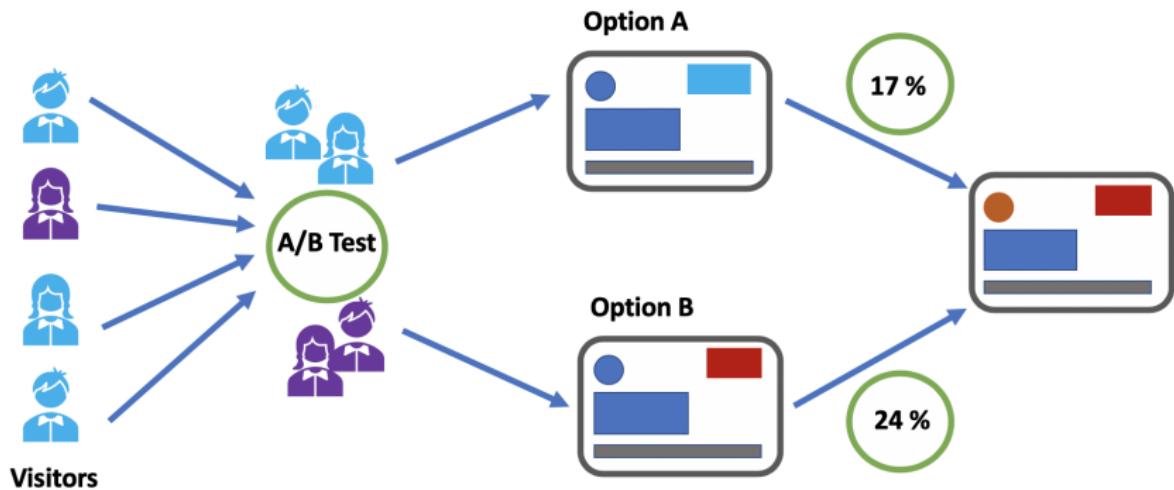
Assume (C1)-(C3) hold. Then under some other mild conditions, our test controls the type-I error asymptotically as either N or T diverges to ∞ .

A Reinforcement Learning Framework for Time-Dependent Causal Effects Evaluation in A/B Testing

joint work with Xiaoyu Wang, Shikai Luo, Rui Song, Hongtu Zhu
and Jieping Ye

—JASA, *under revision.*

A/B testing



Taken from <https://towardsdatascience.com/how-to-conduct-a-b-testing-3076074a8458>.

Motivation and challenges

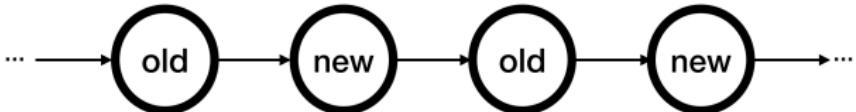
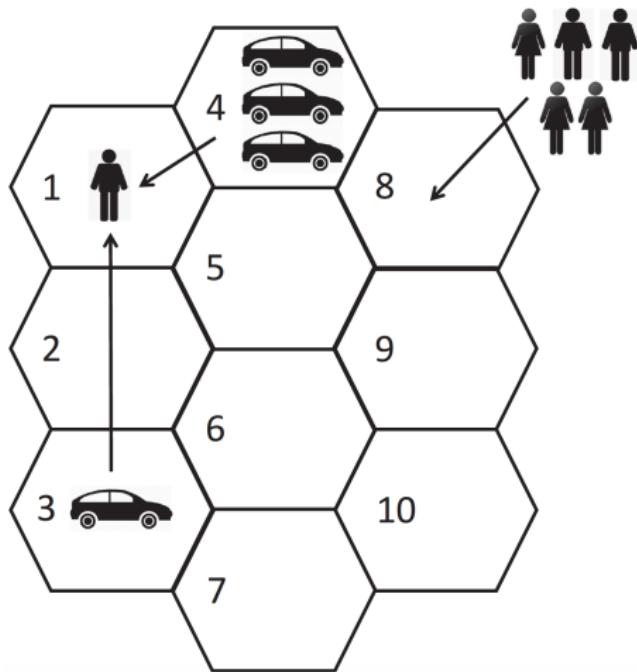
- The project is motivated by the need for comparing the **long-term rewards** of different **order dispatching policies** in **ridesharing platforms**.
- Challenges:
 - ① The existence of **carryover effects**:
 - Alternating time-interval design
 - Past actions will affect future outcomes.
 - ② The need for **early termination**:
 - Each experiment takes a considerable time (at most 2 weeks);
 - Early termination to save time and budget.
 - ③ The need for **adaptive randomization**:
 - Maximize the total reward (e.g., epsilon-greedy);
 - Detect the alternative faster.
- No existing test has addressed three challenges simultaneously.

Illustration of carryover effects



Limitations of existing A/B tests

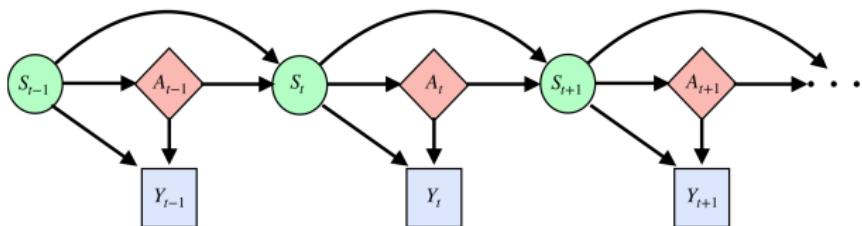
- Most existing tests **cannot** detect carryover effects.
- \mathcal{H}_0 : The old policy ($A = 0$) has larger cumulative rewards.
- \mathcal{H}_1 : The new policy ($A = 1$) has larger cumulative rewards.
- **Example 1.** $S_t \sim N(0, 0.25)$, $R_t = S_t + \delta A_t$
- **Example 2.** $S_t = 0.5S_{t-1} + \delta A_{t-1} + N(0, 0.25)$, $R_t = S_t$

Example 1	t-test 0.76	DML-based test 1.00	our test 0.98
Example 2	t-test 0.04	DML-based test 0.06	our test 0.73

Table: Powers of t-test, DML-based test (Chernozhukov et al., 2018) and the proposed test with $T = 500$, $\delta = 0.1$.

Contributions and advances of our proposal

- Introduce an RL framework for A/B testing



- **① A_{t-1} impacts R_t indirectly through its effect on S_t**
- **② S_t shall include important mediators between A_{t-1} and R_t**
- Propose a test procedure for detecting value difference
 - ① allows for **sequential monitoring**
 - ② allows for **online updating**
 - ③ applicable to a wide range of designs, including the **Markov** design, **alternating-time-interval** design and **adaptive** design

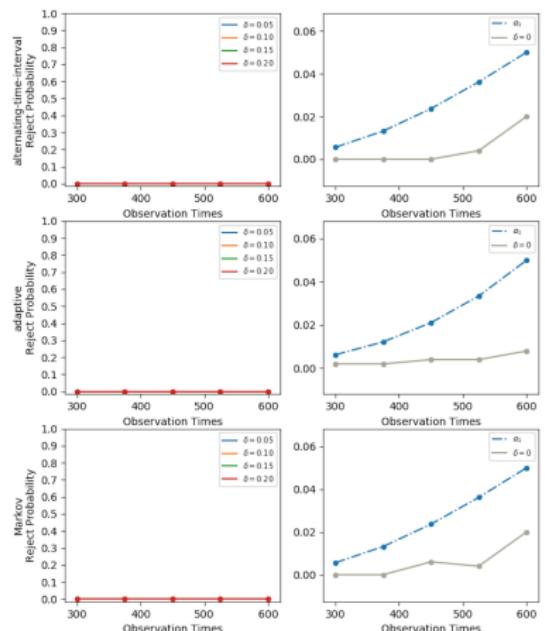
Methodology and theory

- Apply **temporal difference learning** to evaluate value difference and provide **uncertainty quantification**
- Adopt the **α -spending approach** (Lan & DeMets, 1983) for sequential monitoring
- Develop a **bootstrap-assisted procedure** for determine the stopping boundary
 - The numerical integration method designed for classical sequential tests is **not** applicable in adaptive design, due to the carryover effects

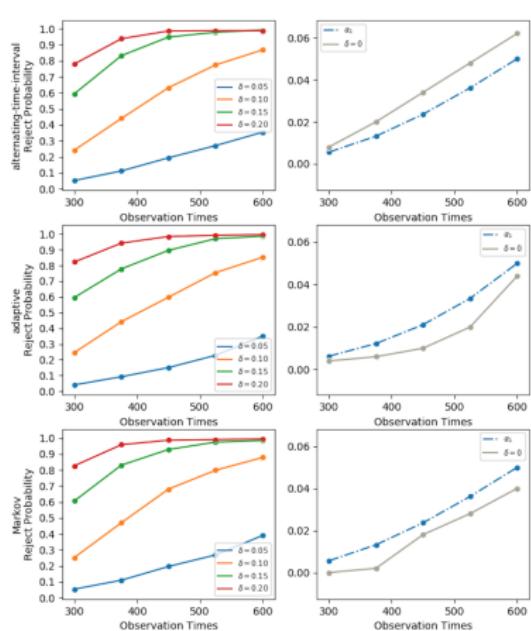
Theorem (Informal statement)

Under the Markov, alternating-time-interval or adaptive design, the proposed test can control type-I error and is consistent against alternatives that converge to the null at the $T^{-1/2}$ rate.

Simulation



(a) Power and size of t-test



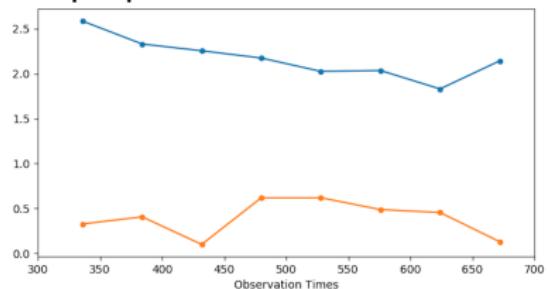
(b) Power and size of our test

Application to ridesharing platform

- **Data:** a given city from December 3rd to 16th (two weeks)
- **30 minutes** as one time unit, sample size = **672**
- **State:**
 - ① number of drivers (supply)
 - ② number of requests (demand)
 - ③ supply and demand equilibrium metric (mediator)
- **Action:** new policy **$A = 1$** v.s. old **$A = 0$**
- **Reward:** drivers' income
- The new policy is expected to have **better** performance

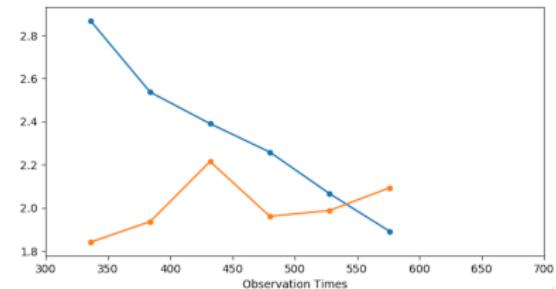
Application to ridesharing platform (Cont'd)

- The proposed test



(a) AA experiment

Blue line: stopping boundary



(b) AB experiment

Orange line: test statistics

- T-test: **fail** to reject \mathcal{H}_0 in A/B experiment with p -value 0.18

Deeply-Debiased Off-Policy Interval Estimation

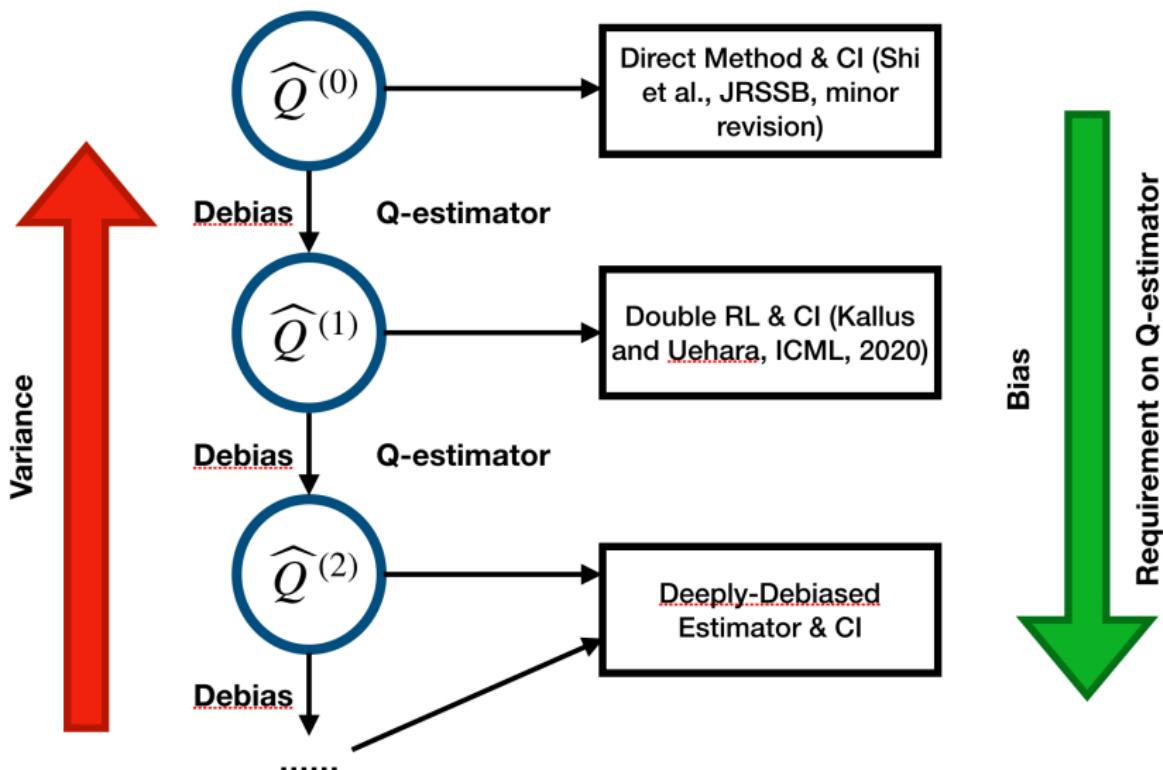
joint work with Runzhe Wan, Victor Chernozhukov and Rui Song

—ICML 2021 (*long talk, top 3% of the submissions*).

Off-Policy Interval Estimation

- **Objective:** Evaluate the impact of a target policy **offline** using historical data generated from a different behavior policy and provide rigorous **uncertainty quantification** (healthcare, automated driving, ridesharing, robotics, e.g.).
- Most existing methods focus on providing point estimators.
- **Method:** Develop a **deeply-debiasing** process

Deeply-Debiased Off-Policy Interval Estimation



Thanks!

Papers and softwares can be found on my personal website
callmespring.github.io