Does the Markov Decision Process Fit the Data

—Testing for the Markov Property in Sequential Decision Making

Chengchun Shi

Associate Professor of Data Science

London School of Economics and Political Science

Developing AI with Reinforcement Learning

Reinforcement Learning Applications

(a) Games

(b) Health Care

(c) Ridesharing


(d) Robotics

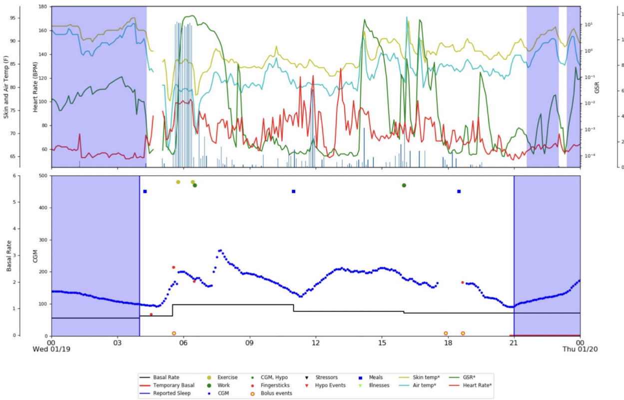

(e) Finance


(f) Automated Driving

We focus on applications in **mobile health** (mHealth)

Applications in mHealth

- Use of cellphones and wearable devices in healthcare
- 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, exercise intensity
- **Data**: OhioT1DM dataset (Marling and Bunescu, 2018)

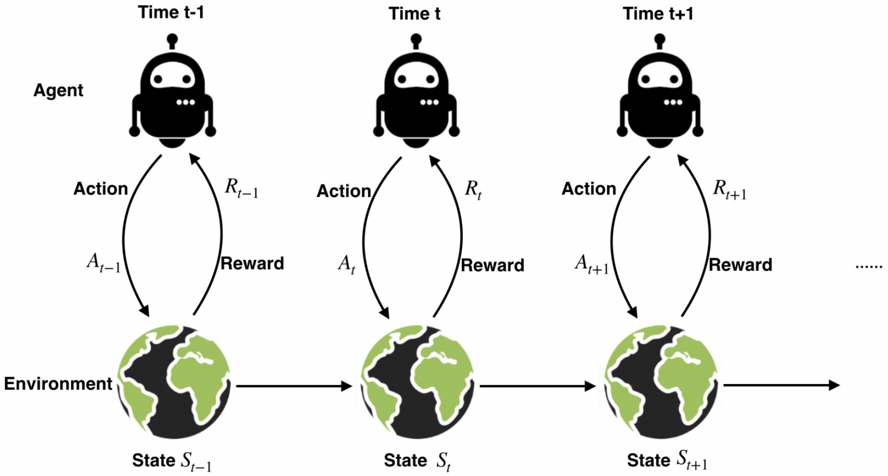


In this talk, we will focus on ...

- **Statistical inference** in reinforcement learning (RL)

- Is statistical inference useful for RL?

Sequential Decision Making



Objective: find an optimal policy that maximizes the cumulative reward

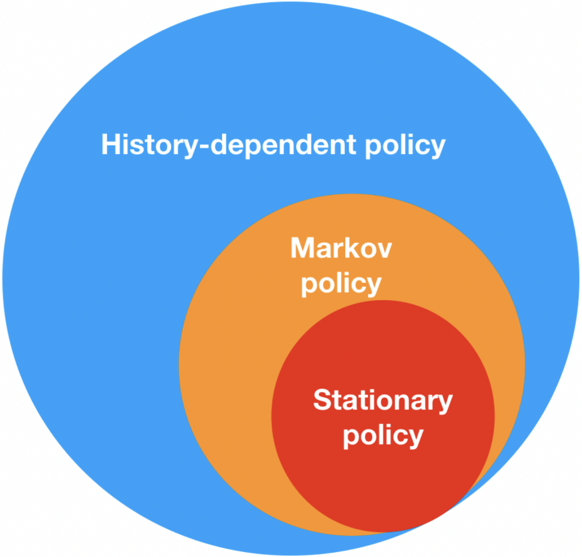
The Agent's Policy

- The agent implements a **mapping** π_t 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**:

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

  where $\bar{S}_t = (S_t, R_{t-1}, A_{t-1}, S_{t-1}, \cdots, R_0, A_0, S_0)$ is the set of **observed data 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.
- **Stationary** Policy: π is Markov & π_t is **homogeneous** in t, i.e., $\pi_0 = \pi_1 = \cdots$.

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,

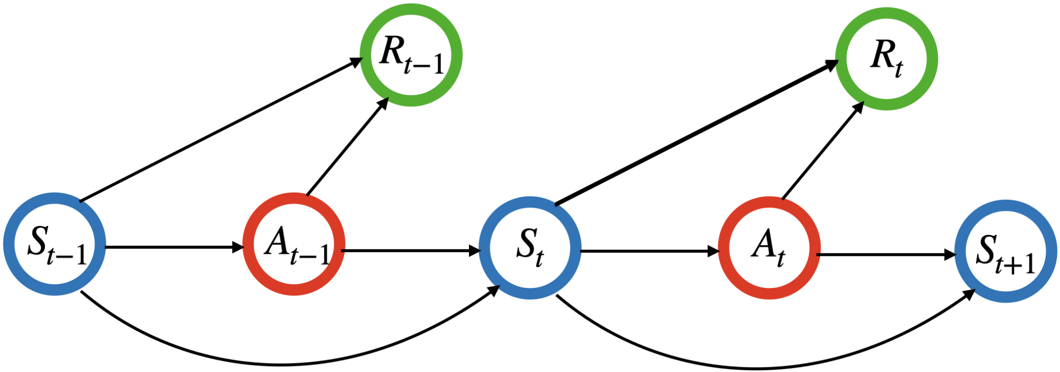
    $$S_{t+1}, R_t \perp\!\!\!\perp \{(S_j, A_j, R_j)\}_{j<t} | S_t, A_t.$$

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

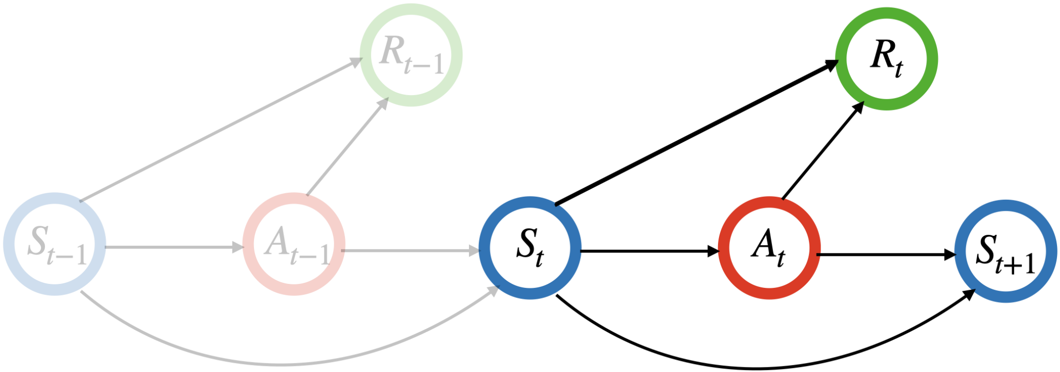
    $$S_{t+1} \perp\!\!\!\perp \{(S_j, A_j)\}_{j<t} | S_t, A_t.$$

    The Markov transition kernel is homogeneous in time

Markov Assumption



Markov Assumption



RL Models

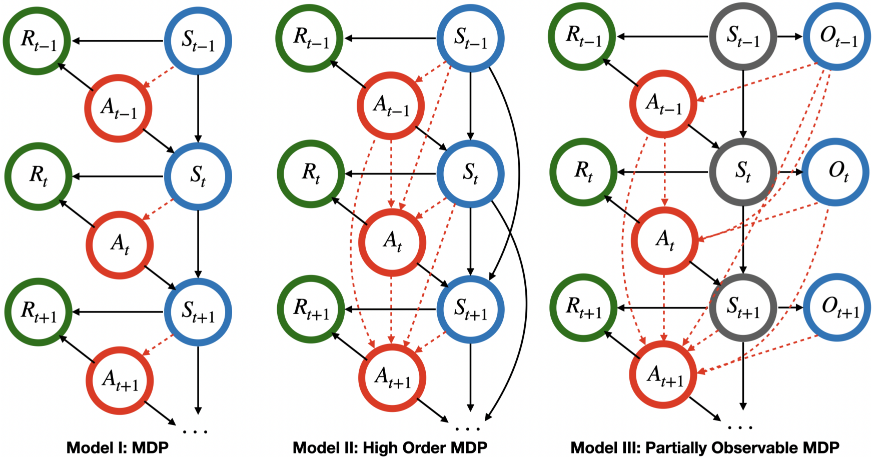


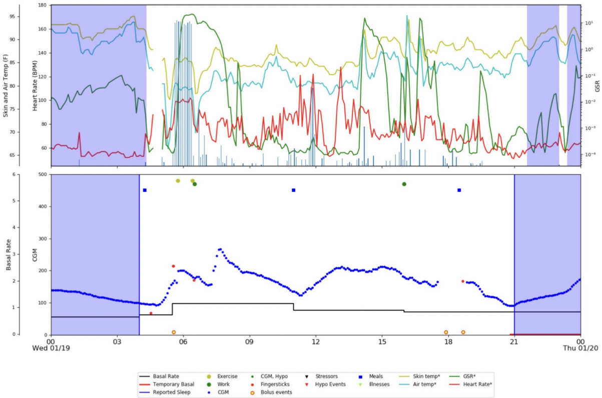
Figure: Causal diagrams for MDPs, HMDPs & POMDPs. The solid lines characterize the relationships among the variables and the dashed lines indicate the information needed to implement the optimal policy. $\{S_t\}_t$ are hidden in Model III.

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** (e.g., test kth order MDP for $k = 1, 2, \cdots$)
  - critical to **offline** domains given a historical dataset **without online collection**:
    - 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
- Measurements for 6 patients with type I diabetes over 8 weeks.
- One-hour interval as a time unit.
- **State**: glucose levels, food intake, exercise intensity
- **Action**: to inject insulin or not.
- **Reward**: the Index of Glycemic Control (Rodbard, 2009).

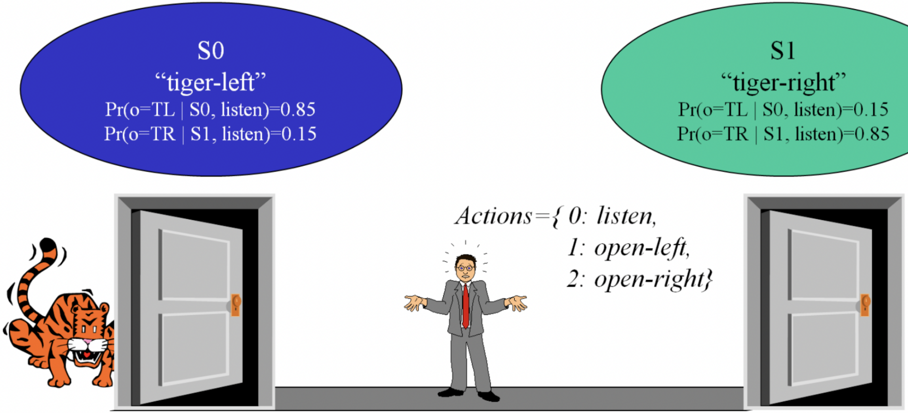


Applications in High-Order MDPs (Cont'd)

- **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**, by assuming it is a k-th order MDP for $k = 1, \cdots, 10$
  - policy evaluation based on **fitted-Q evaluation**
  - 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

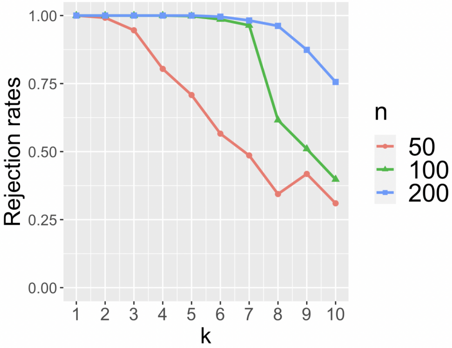
- *Penalty for wrong opening: -100*
- *Reward for correct opening: +10*
- *Cost for listening action: -1*

Observations

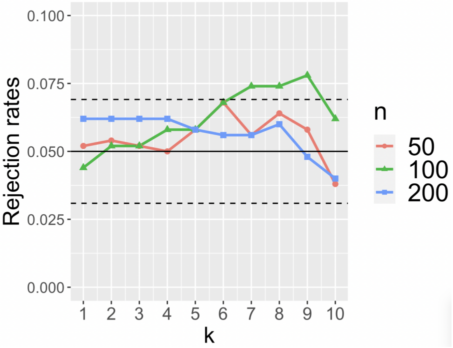
- *to hear the tiger on the left (TL)*
- *to hear the tiger on the right(TR)*

Applications in Partially Observable MDPs (Cont'd)

- Under $\mathcal{H}_1$ (MA is violated, alternative). Significance level $= 0.05$.



- Under $\mathcal{H}_0$ (MA holds, null). Significance level $= 0.05$.



Methodology

- **First** work to test MA in RL
- 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.

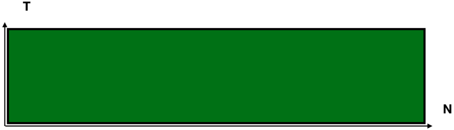
Methodology (Cont'd)

Some key components of our algorithm:

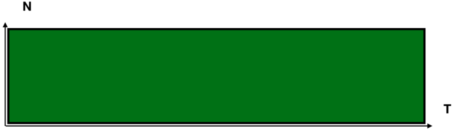
- To deal with moderate or high-dimensional state space, employ modern machine learning (ML) algorithms to estimate CCF:
    - Learn CCF of S_{t+1} given A_t and S_t (**forward learner**)
    - Learn CCF of (S_t, A_t) given (S_{t+1}, A_{t+1}) (**backward learner**)
    - Develop a **random forest**-based 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** test statistics by integrating forward and backward learners;
- To improve the power, consider a **maximum-type** test statistic;
- To control the type-I error, approximate the distribution of our test via **high-dimensional multiplier bootstrap** (Chernozhukov, et al., 2014).

Bidirectional Theory

- $\boldsymbol{N}$ the number of trajectories
- $\boldsymbol{T}$ the number of decision points per trajectory
- **bidirectional asymptotics**: a framework allows either $\boldsymbol{N}$ or $\boldsymbol{T} \to \infty$
- large $\boldsymbol{N}$, small $\boldsymbol{T}$ (Intern Health Study)



- small $\boldsymbol{N}$, large $\boldsymbol{T}$ (OhioT1DM dataset)



- large $\boldsymbol{N}$, large $\boldsymbol{T}$ (games)

Bidirectional Theory (Cont'd)

(C1) Actions are generated by a fixed behavior policy.

(C2) The observed data is exponentially β-mixing.

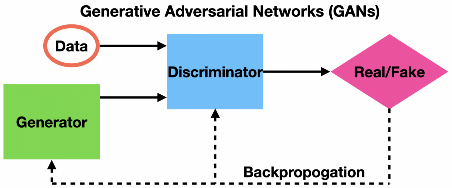
(C3) The ℓ_2 prediction errors of forward and backward learners converge at a rate faster than $(NT)^{-1/4}$.

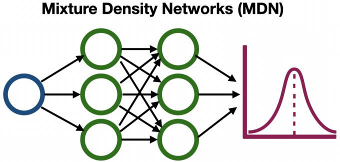
Theorem

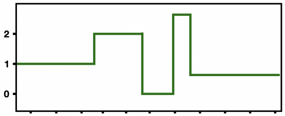
Assume (C1)-(C3) hold. Then under some other mild conditions, our test controls the type-I error asymptotically as either N or T diverges to ∞.

Some Follow-ups

- **Double GANs** for conditional independence testing (*JMLR, 2021*)
- Testing DAGs via supervised, structural learning and **GANs** (*JASA, 2023+*)
- Testing Markovanity in time series via **deep generative learning** (*JRSSB, 2023+*)
  - Derive the convergence rate of **MDN**
- A robust test for the **stationarity** assumption in RL (*ICML, 2023*)
  - Our test helps identify a better policy in the **Intern Health Study**







Thank You!

☺ Papers and softwares can be found on my personal website

`callmespring.github.io`