Approximate planning and learning in partially observed systems

Aditya Mahajan
McGill University

Joint work with Jayakumar Subramanian, Amit Sinha, and Raihan Seraj

Communication and Signal Processing Seminar
University of Michigan
3rd December 2020

- email: aditya.mahajan@mcgill.ca
- web: http://cim.mcgill.ca/~adityam

Recent successes of RL

- Algorithms based on comprehensive theory

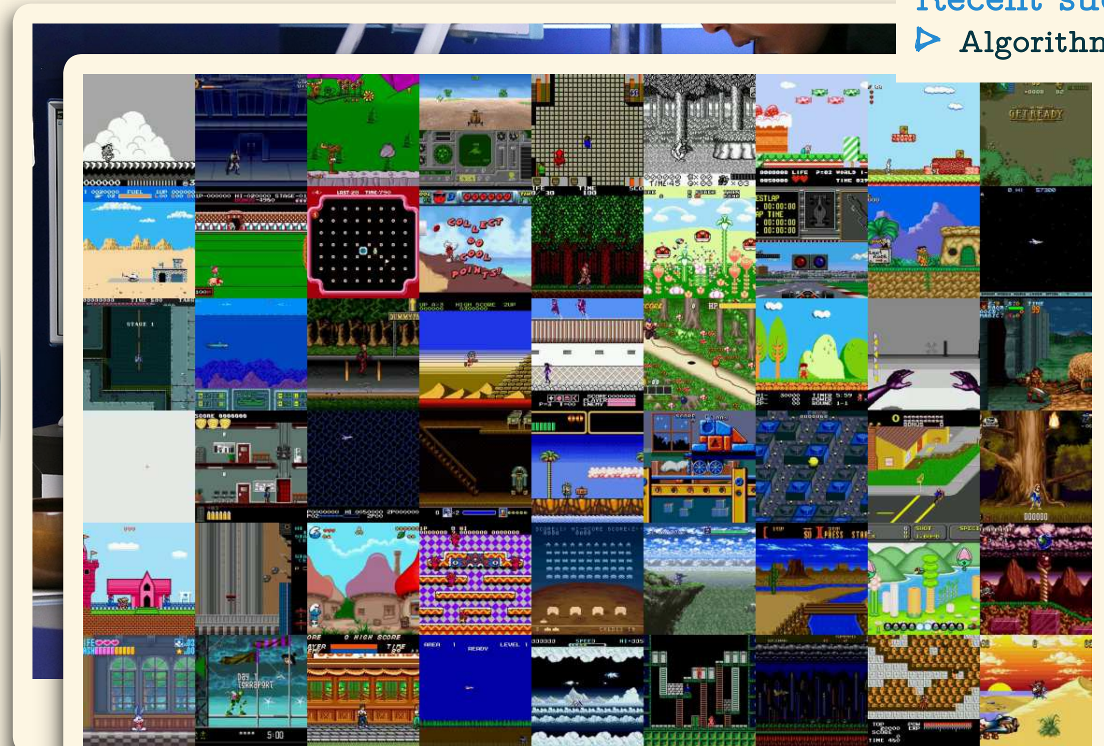
Recent successes of RL

- Algorithms based on comprehensive theory

Alpha Go

Recent successes of RL

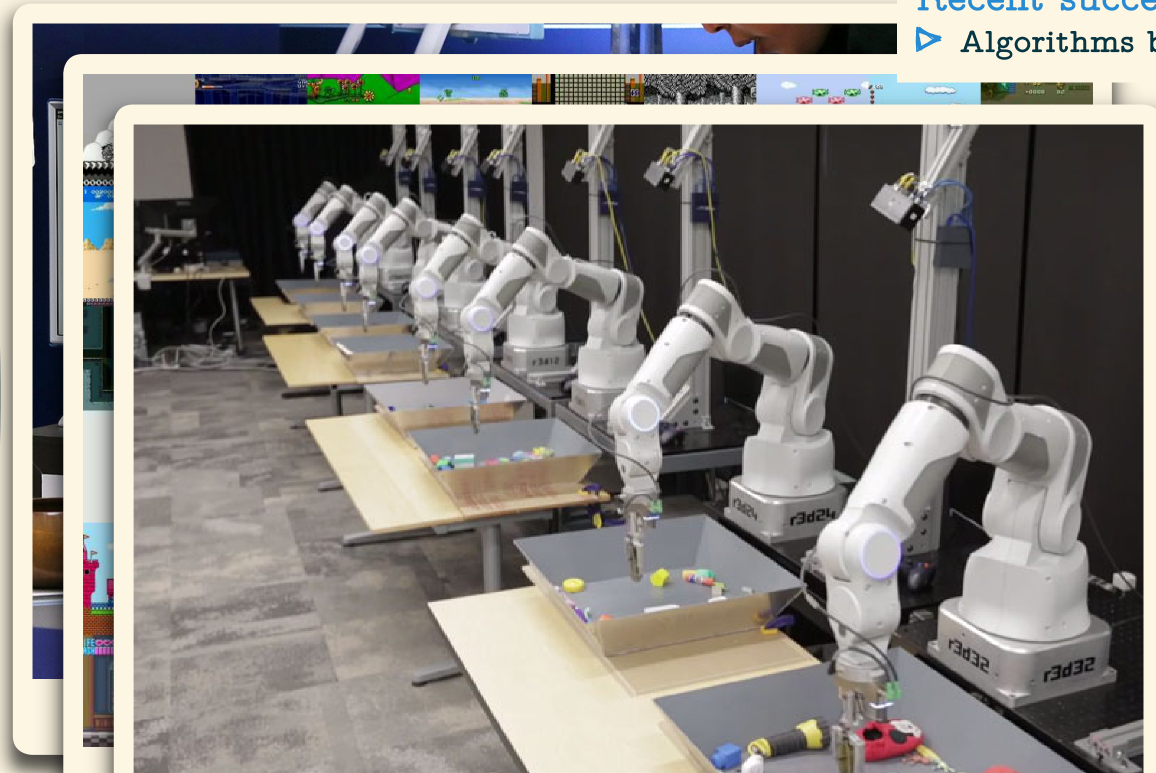
- Algorithms based on comprehensive theory



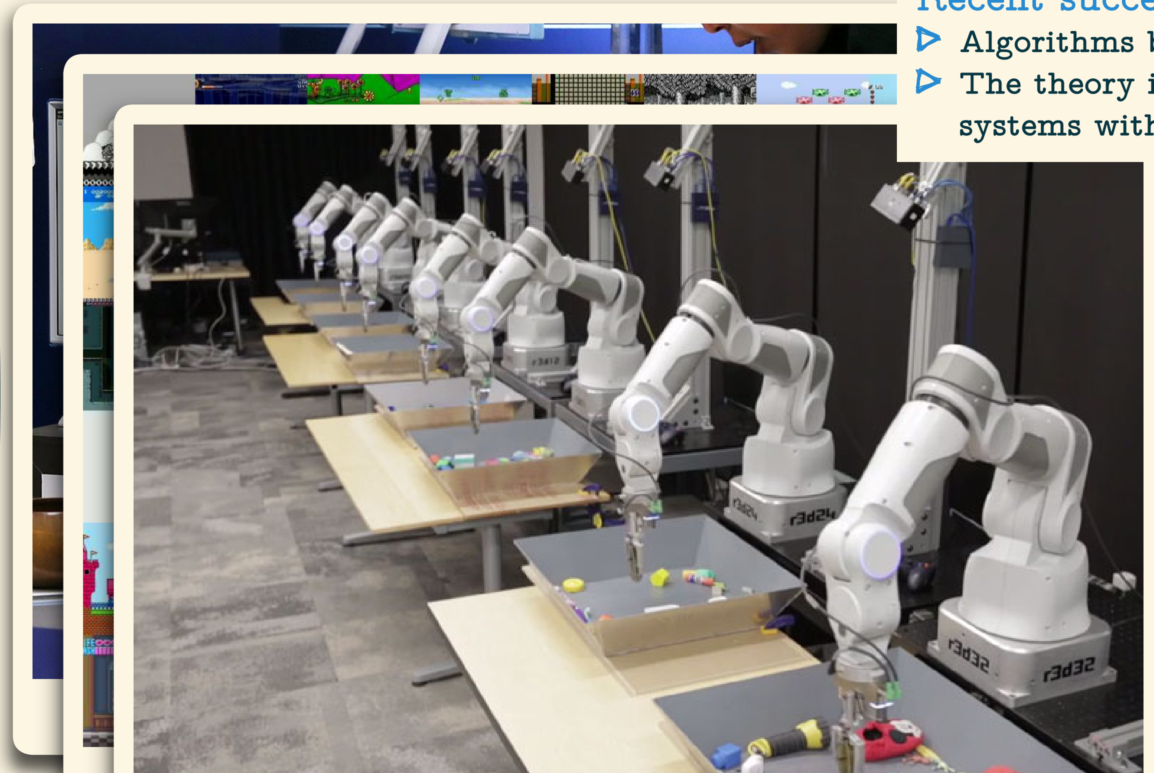
Arcade games

Recent successes of RL

- Algorithms based on comprehensive theory



Robotic grasping



Robotic grasping

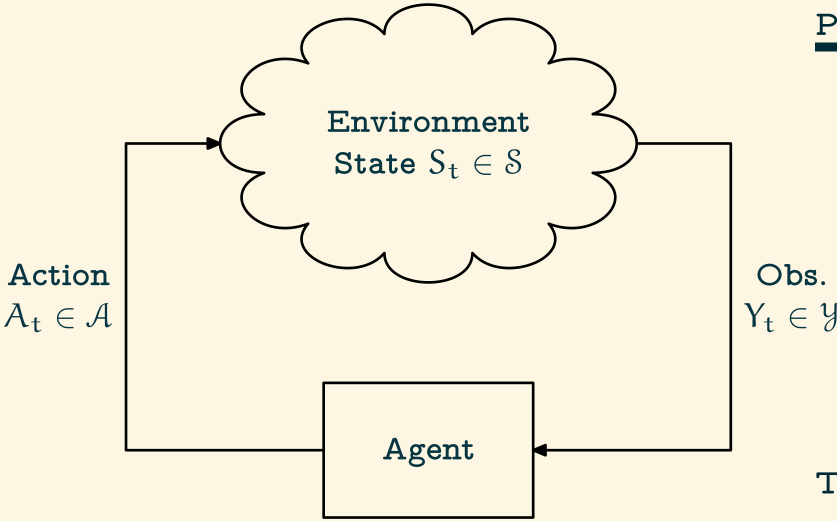
Recent successes of RL

- Algorithms based on comprehensive theory
- The theory is restricted almost exclusively to systems with perfect state observations.


HEALTH CARE
Most real world systems are partially observed

Why is it difficult to learn in partially observable environments?

Review: Planning in partially observable environments



POMDP: PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

Dynamics: $\mathbb{P}(S_{t+1} \mid S_t, A_t)$

Observations: $\mathbb{P}(Y_t \mid S_t)$

Reward $R_t = r(S_t, A_t)$.

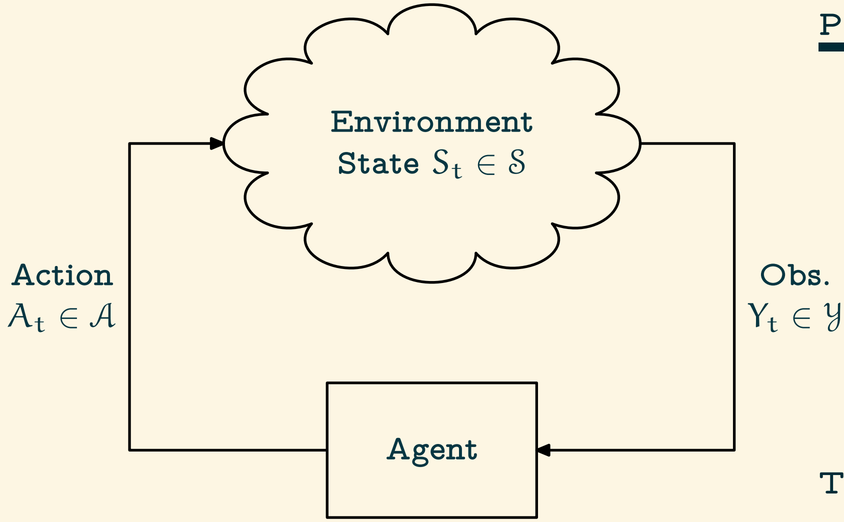
Action: $A_t = \pi_t(Y_{1:t}, A_{1:t-1})$.

$\pi = (\pi_t)_{t \geqslant 1}$ is called a policy.

The objective is to choose a policy π to maximize:

$$J(\pi) := \mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t\right]$$

Review: Planning in partially observable environments



POMDP: PARTIALLY OBSERVABLE MARKOV DECISION PROCESS

Dynamics: $\mathbb{P}(S_{t+1} \mid S_t, A_t)$

Observations: $\mathbb{P}(Y_t \mid S_t)$

Reward $R_t = r(S_t, A_t)$.

Action: $A_t = \pi_t(Y_{1:t}, A_{1:t-1})$.

$\pi = (\pi_t)_{t \geqslant 1}$ is called a policy.

The objective is to choose a policy π to maximize:

$$J(\pi) := \mathbb{E}^{\pi}\left[\sum_{t=1}^{\infty} \gamma^{t-1} R_t\right]$$

Conceptual challenge

- Action is a function of the history of observations and actions.
- The history is increasing in time. So, the search complexity increases exponentially in time.

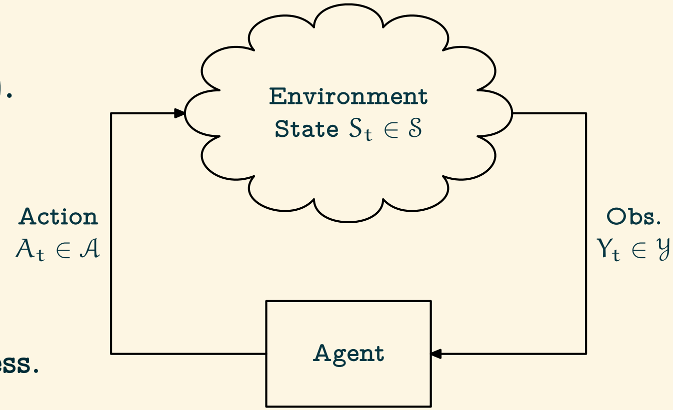
Review: Planning in partially observable environments

Key simplifying idea

Define **belief state** $B_t \in \Delta(\mathcal{S})$ as $B_t(s) = \mathbb{P}(S_t = s \mid Y_{1:t}, A_{1:t-1})$.

- Belief state updates in a state-like manner
$$B_{t+1} = \text{function}(B_t, Y_{t+1}, A_t).$$
- Belief state is sufficient to evaluate rewards
$$\mathbb{E}[R_t \mid Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t).$$

Thus, $\{B_t\}_{t \geqslant 1}$ is a perfectly observed controlled Markov process.



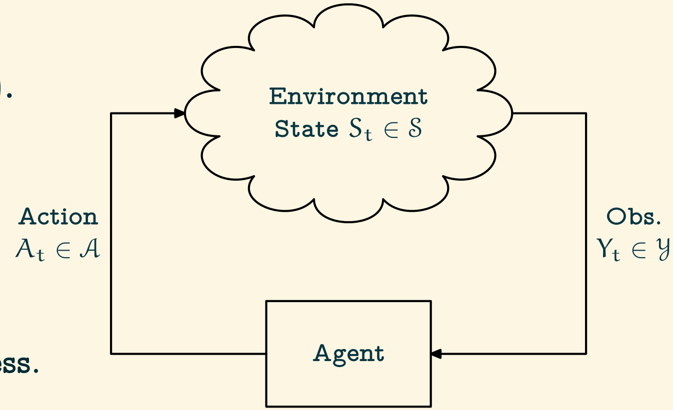
Review: Planning in partially observable environments

Key simplifying idea

Define **belief state** $B_t \in \Delta(\mathcal{S})$ as $B_t(s) = \mathbb{P}(S_t = s \mid Y_{1:t}, A_{1:t-1})$.

- Belief state updates in a state-like manner
$$B_{t+1} = \text{function}(B_t, Y_{t+1}, A_t).$$
- Belief state is sufficient to evaluate rewards
$$\mathbb{E}[R_t \mid Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t).$$

Thus, $\{B_t\}_{t \geq 1}$ is a perfectly observed controlled Markov process.



Therefore, we get the following results:

Structure of optimal policy	There is no loss of optimality in choosing the action A_t as a function of the belief state B_t
Dynamic Program	The optimal control policy is given by the solution of the following DP: $V_t(b_t) = \max_{a_t \in \mathcal{A}} \Big\{ \hat{r}(S_t, A_t) + \mathbb{E}[V_{t+1}(B_{t+1}) \mid B_t = b_t, A_t = a_t] \Big\}$

Implications of the modeling framework

Implications for planning

- Allows to use the entire machinery of fully observed Markov decision processes for partially observed systems.
- Various exact and approximate algorithms can efficiently solve the DP.

  Exact: incremental pruning, witness algorithm, linear support algo
  Approximate: QMDP, point based methods, SARSOP, DESPOT, . . .

Implications of the modeling framework

Implications for planning

- Allows to use the entire machinery of fully observed Markov decision processes for partially observed systems.
- Various exact and approximate algorithms can efficiently solve the DP.

  Exact: incremental pruning, witness algorithm, linear support algo
  Approximate: QMDP, point based methods, SARSOP, DESPOT, . . .

Implications for learning

- The construction of the belief state depends on the system model.
- So, when the system model is unknown, we cannot construct the belief state and therefore cannot use standard RL algorithms.

Implications of the modeling framework

Implications for planning

- Allows to use the entire machinery of fully observed Markov decision processes for partially observed systems.
- Various exact and approximate algorithms can efficiently solve the DP.

  Exact: incremental pruning, witness algorithm, linear support algo
  Approximate: QMDP, point based methods, SARSOP, DESPOT, . . .

Implications for learning

- The construction of the belief state depends on the system model.
- So, when the system model is unknown, we cannot construct the belief state and therefore cannot use standard RL algorithms.
- On the theoretical side:
  - Propose alternative methods: PSRs (predictive state representations), bisimulation metrics, . . .
  - Good theoretical guarantees, but difficult to scale.

Implications of the modeling framework

Implications for planning

- Allows to use the entire machinery of fully observed Markov decision processes for partially observed systems.
- Various exact and approximate algorithms can efficiently solve the DP.

  Exact: incremental pruning, witness algorithm, linear support algo
  Approximate: QMDP, point based methods, SARSOP, DESPOT, . . .

Implications for learning

- The construction of the belief state depends on the system model.
- So, when the system model is unknown, we cannot construct the belief state and therefore cannot use standard RL algorithms.
- On the theoretical side:
  - Propose alternative methods: PSRs (predictive state representations), bisimulation metrics, . . .
  - Good theoretical guarantees, but difficult to scale.
- On the practical side:
  - Simply stack the previous k observations and treat it as a “state”.
  - Instead of a CNN, use an RNN to model policy and action-value fn.
  - Can be made to work but lose theoretical guarantees and insights.

This talk: A <u>theoretically grounded</u> method for RL in partially observable models which has <u>strong empirical performance</u> for high-dimensioanl environments.

- **paper**: https://arxiv.org/abs/2010.08843
- **code**: https://github.com/info-structures/ais

The high-level view

Information state

- A classical (but perhaps not well known) concept in stochastic control.
- Informally, an information state is a sufficient statistic which can be recursively updated.
- Always leads to a dynamic programming decomposition.

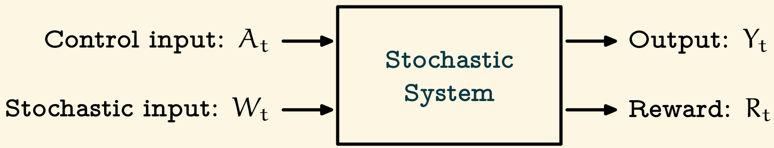
Approximate information state

- Information state is defined in terms of two properties.
- An AIS is a process which approximtely satisfies these properties.
- We show tht an AIS always leads to a approximate dynamic program.
- Recover (and improve up on) many existing results in the literature.

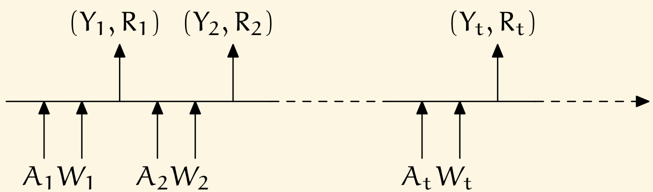
AIS based RL

- There are two approximation errors in the definition of AIS.
- Use these approximation errors as a surrogate loss
- Performs better than SOTA RL algorithms for POMDPs.

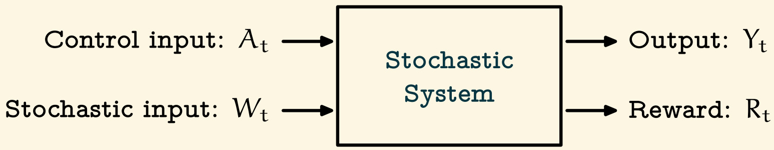
Preliminaries: Input/output modeling



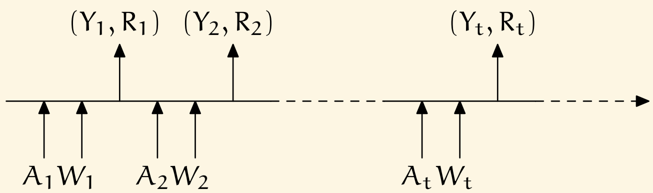
$$
\begin{aligned}
Y_t &= f_t(A_{1:t}, W_{1:t}), \\
R_t &= r_t(A_{1:t}, W_{1:t}).
\end{aligned}
$$



Preliminaries: Input/output modeling



$$Y_t = f_t(A_{1:t}, W_{1:t}),$$
$$R_t = r_t(A_{1:t}, W_{1:t}).$$



- Let $H_t = (Y_{1:t-1}, A_{1:t-1})$ denote the history of all observations and actions available to the agent before taking action at time t.
- Assume that the agent chooses an $A_t \sim \pi_t(H_t)$.
- Let $\pi = (\pi_1, \pi_2, \dots)$ denote the control policy.

The objective is to choose a policy π to maximize:

$$J(\pi) := \mathbb{E}^{\pi}\Big[\sum_{t=1}^{\infty} \gamma^{t-1} R_t\Big]$$

Outline

Information state

- A classical (but perhaps not well known) concept in stochastic control.
- Informally, an information state is a sufficient statistic which can be recursively updated.
- Always leads to a dynamic programming decomposition.

Approximate information state

AIS based RL

From sufficient statistics to information state

Sufficient Statistics

S	Y	A
State	Obs.	Action

$Z = \sigma(Y)$ is a sufficient statistic for (the purpose of) evaluating the reward $R = r(S, A)$ if

(P1) $$\mathbb{E}[R \mid Y = y, A = a] = \underbrace{\mathbb{E}[R \mid Z = \sigma(y), A = a]}_{=:\hat{r}(\sigma(y), a)}$$

From sufficient statistics to information state

Sufficient Statistics

S	Y	A
State	Obs.	Action

$Z = \sigma(Y)$ is a sufficient statistic for (the purpose of) evaluating the reward $R = r(S, A)$ if

(P1) $\mathbb{E}[R \mid Y = y, A = a] = \underbrace{\mathbb{E}[R \mid Z = \sigma(y), A = a]}_{=:\hat{r}(\sigma(y), a)}$

Information state

Consider a POMDP. Suppose:

- $Z_t = \sigma_t(H_t)$ is a sufficient statistic for evaluating the reward R_t, and
- $Z_{t+1} = \sigma_{t+1}(H_{t+1})$ is a sufficient statistic for evaluating the reward R_{t+1}.

Is Z_t sufficient for dynamic programming?

From sufficient statistics to information state

Sufficient Statistics

S State Y Obs. A Action

$Z = \sigma(Y)$ is a sufficient statistic for (the purpose of) evaluating the reward $R = r(S, A)$ if

(P1) $\mathbb{E}[R \mid Y = y, A = a] = \underbrace{\mathbb{E}[R \mid Z = \sigma(y), A = a]}_{=:\hat{r}(\sigma(y), a)}$

Information state

Consider a POMDP. Suppose:

- $Z_t = \sigma_t(H_t)$ is a sufficient statistic for evaluating the reward R_t, and
- $Z_{t+1} = \sigma_{t+1}(H_{t+1})$ is a sufficient statistic for evaluating the reward R_{t+1}.

Is Z_t sufficient for dynamic programming?

In general, no. To solve a DP, we need to be able to compute:

$$R_t + \gamma\, \mathbb{E}[V_{t+1}(Z_{t+1}) | H_t = h_t, A_t = a_t]$$

So, in addition to (P1), we need:

(P2) $\mathbb{P}(Z_{t+1} = z_{t+1} | H_t = h_t, A_t = a_t) = \mathbb{P}(Z_{t+1} = z_{t+1} | Z_t = \sigma_t(H_t), A_t = a_t)$

Informally, an information state is a compression of the history which is sufficient for performance evaluation and predicting itself.

Formal definition of information state

Information State

Given a Banach space $\mathcal{Z}$, a collection $\{\sigma_t : \mathcal{H}_t \to \mathcal{Z}\}_{t \geq 1}$ is called an information state generator if there exist a reward function $\hat{r}$ and a transition kernel $\hat{P}$ such that they are:

(P1) Sufficient for performance evaluation:

$$\mathbb{E}[R_t \mid H_t = h_t, A_t = a_t] = \hat{r}(\sigma_t(h_t), a_t).$$

(P2) Sufficient for predicting itself:

$$\mathbb{P}(Z_{t+1} = z_{t+1} \mid H_t = h_t, A_t = a_t) = \hat{P}(z_{t+1} | \sigma_t(h_t), a_t).$$

Formal definition of information state

Information State

Given a Banach space $\mathcal{Z}$, a collection $\{\sigma_t : \mathcal{H}_t \to \mathcal{Z}\}_{t \geqslant 1}$ is called an information state generator if there exist a reward function $\hat{r}$ and a transition kernel $\widehat{P}$ such that they are:

(P1) Sufficient for performance evaluation:

$$\mathbb{E}[R_t \mid H_t = h_t, A_t = a_t] = \hat{r}(\sigma_t(h_t), a_t).$$

(P2) Sufficient for predicting itself:

$$\mathbb{P}(Z_{t+1} = z_{t+1} \mid H_t = h_t, A_t = a_t) = \widehat{P}(z_{t+1} | \sigma_t(h_t), a_t).$$

Info State based dynanmic program

Let $\{Z_t\}_{t \geqslant 1}$ be any information state proces. Define

$$V(z) = \max_{a \in \mathcal{A}} \Big\{ \hat{r}(z, a) + \gamma \int_{\mathcal{Z}} V(z_+) \widehat{P}(dz_+ | z, a) \Big\}$$

Let $\pi^*(z)$ denote the arg max of the RHS. Then, the policy $\pi = (\pi_1, \pi_2, \dots)$ given by $\pi_t = \pi^* \circ \sigma_t$ is optimal.

Examples of information state

Markov decision processes (MDP)

Current state S_t is an info state

Examples of information state

Markov decision processes (MDP)

Current state S_t is an info state

MDP with delayed observations

$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state

Examples of information state

Markov decision processes (MDP)

Current state S_t is an info state

MDP with delayed observations

$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state

POMDP

Belief state is an info state

Examples of information state

Markov decision processes (MDP)

Current state S_t is an info state

MDP with delayed observations

$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state

POMDP

Belief state is an info state

POMDP with delayed observations

$(\mathbb{P}(S_{t-\delta} | Y_{1:t-\delta}, A_{1:t-\delta}), A_{t-\delta+1:t-1})$ is info state

Examples of information state

Markov decision processes (MDP)

Current state S_t is an info state

MDP with delayed observations

$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state

POMDP

Belief state is an info state

POMDP with delayed observations

$(\mathbb{P}(S_{t-\delta} | Y_{1:t-\delta}, A_{1:t-\delta}), A_{t-\delta+1:t-1})$ is info state

Linear Quadratic Guassian (LQG)

The state estimate $\mathbb{E}[S_t | H_t]$ is an info state

Examples of information state

Markov decision processes (MDP)

Current state S_t is an info state

MDP with delayed observations

$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state

POMDP

Belief state is an info state

POMDP with delayed observations

$(\mathbb{P}(S_{t-\delta} | Y_{1:t-\delta}, A_{1:t-\delta}), A_{t-\delta+1:t-1})$ is info state

Linear Quadratic Guassian (LQG)

The state estimate $\mathbb{E}[S_t | H_t]$ is an info state

Machine Maintenance

(τ, S^+_τ) is info state,
where τ is the time of last maintenance

Outline

Information state

Approximate information state

- Information state is defined in terms of two properties.
- An AIS is a process which approximtely satisfies these properties.
- We show tht an AIS always leads to a approximate dynamic program.
- Recover (and improve up on) many existing results in the literature.

AIS based RL

Approximate information state (AIS)

Approximate information state

A collection $(\sigma_t, \hat{r}, \hat{P})$ is called an (ε, δ)-approximate information state (AIS) if it satisfies properties (P1) and (P2) approximately, i.e.,

(P1) Sufficient for approximate performance evaluation:

$$\left| \mathbb{E}[R_t \mid H_t = h_t, A_t = a_t] - \hat{r}(\sigma_t(h_t), a_t) \right| \leqslant \varepsilon$$

(P2) Sufficient for predicting itself approximately:

$$d_{\mathfrak{F}}\big(\mathbb{P}(Z_{t+1} = \cdot \mid H_t = h_t, A_t = a_t), \hat{P}(\cdot | \sigma_t(h_t), a_t)\big) \leqslant \delta$$

Approximate information state (AIS)

Approximate information state

A collection $(\sigma_t, \hat{r}, \hat{P})$ is called an (ε, δ)-approximate information state (AIS) if it satisfies properties (P1) and (P2) approximately, i.e.,

(P1) Sufficient for approximate performance evaluation:

$$\left| \mathbb{E}[R_t \mid H_t = h_t, A_t = a_t] - \hat{r}(\sigma_t(h_t), a_t) \right| \leqslant \varepsilon$$

(P2) Sufficient for predicting itself approximately:

$$d_{\mathfrak{F}}\big(\mathbb{P}(Z_{t+1} = \cdot \mid H_t = h_t, A_t = a_t), \hat{P}(\cdot \mid \sigma_t(h_t), a_t)\big) \leqslant \delta$$

Metrics on probability measures

- The definition of AIS depends on the choice of metric $d_{\mathfrak{F}}$ on probability measures.
- There are various choices for choosing a metric on probability measures, e.g., total variation, Wasserstein distance, bounded-Lipsctiz metric, etc.
- We work with a class of metrics known as integral probability metrics (IPM) with respect to a class of function $\mathfrak{F}$.
- The precise approximation bounds depend on what is called the Minkowski functional $\rho_{\mathfrak{F}}$ corresponding to $\mathfrak{F}$.

Integral probability metrics (IPMs)

IPM

Given a measurable space $\mathcal{X}$ and class of real-valued functions $\mathfrak{F}$ on $\mathcal{X}$, the integral probability metric (IPM) between two distributions μ and ν on $\mathcal{X}$ with respect to $\mathfrak{F}$ is defined as

$$d_{\mathfrak{F}}(\mu, \nu) = \sup_{f \in \mathfrak{F}} \left| \int_{\mathcal{X}} f d\mu - \int_{\mathcal{X}} f d\nu \right|.$$

The Minkowski function $\rho_{\mathfrak{F}}$ with respect to $\mathfrak{F}$ is given by

$$\rho_{\mathfrak{F}}(f) = \inf\{\rho \in \mathbb{R}_{\geqslant 0} : \rho^{-1} f \in \mathfrak{F}\}.$$

Integral probability metrics (IPMs)

IPM

Given a measurable space $\mathcal{X}$ and class of real-valued functions $\mathfrak{F}$ on $\mathcal{X}$, the integral probability metric (IPM) between two distributions μ and ν on $\mathcal{X}$ with respect to $\mathfrak{F}$ is defined as

$$d_{\mathfrak{F}}(\mu, \nu) = \sup_{f \in \mathfrak{F}} \left| \int_{\mathcal{X}} f d\mu - \int_{\mathcal{X}} f d\nu \right|.$$

The Minkowski function $\rho_{\mathfrak{F}}$ with respect to $\mathfrak{F}$ is given by

$$\rho_{\mathfrak{F}}(f) = \inf\{\rho \in \mathbb{R}_{\geqslant 0} : \rho^{-1} f \in \mathfrak{F}\}.$$

Examples of IPM

- Total variation distance corresponds to $\mathfrak{F} = \{f : \|f\|_\infty \leqslant 1\}$.
- Kolmogorov distance corresponds to $\mathfrak{F} = \{\mathbb{1}_{(-\infty, t]} : t \in \mathbb{R}^m\}$.
- Wasserstein distance corresponds to $\mathfrak{F} = \{f : \|f\|_{\mathrm{Lip}} \leqslant 1\}$.
- Maximum mean discrepancy corresponds to $\mathfrak{F} = \{f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leqslant 1\}$, where $\mathcal{H}$ is a RKHS.

AIS based approximation bounds

Let $\hat{V}$ be the fixed point of the following equations:

$$\hat{V}(z, a) = \max_{a \in \mathcal{A}} \Big\{ \hat{r}(z, a) + \gamma \int_{\mathcal{Z}} \hat{V}(z_+) \hat{P}(dz_+ | z, a) \Big\}$$

Let V denote the optimal value and action-value functions.

AIS based approximation bounds

Let $\hat{V}$ be the fixed point of the following equations:

$$\hat{V}(z, a) = \max_{a \in \mathcal{A}} \Big\{ \hat{r}(z, a) + \gamma \int_{\mathcal{Z}} \hat{V}(z_+) \hat{P}(dz_+ | z, a) \Big\}$$

Let V denote the optimal value and action-value functions.

Then, we have the following:

Value function approximation	The value function $\hat{V}$ is approximately optimal, i.e., $$\lvert V_t(h_t) - \hat{V}(\sigma_t(h_t)) \rvert \leqslant \alpha = \frac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V}) \delta}{1 - \gamma}.$$
Policy approximation	Let $\hat{\pi}^* \colon \mathcal{Z} \to \Delta(\mathcal{A})$ be an optimal policy for $\hat{V}$. Then, the policy $\pi = (\pi_1, \pi_2, \dots)$ given by $\pi_t = \hat{\pi}^* \circ \sigma_t$ is approx. optimal: $$V_t(h_t) - V_t^{\pi}(h_t) \leqslant 2\alpha.$$

Examples of AIS

Example 1: Robustness to model mismatch in MDPs

Real-world model

(P, r)

Simulation model

$(\hat{P}, \hat{r})$

What is the loss in performance if we choose a policy using the simulation model and use it in the real world?

Example 1: Robustness to model mismatch in MDPs

Real-world model (P, r)

Simulation model $(\hat{P}, \hat{r})$

What is the loss in performance if we choose a policy using the simulation model and use it in the real world?

Model mismatch as an AIS

- $(\text{Identity}, \hat{P}, \hat{r})$ is an (ε, δ)-AIS with $\varepsilon = \sup_{s,a} \big| r(s,a) - \hat{r}(s,a) \big|$ and $\delta_{\mathfrak{F}} = \sup_{s,a} d_{\mathfrak{F}}(P(\cdot|s,a), \hat{P}(\cdot|s,a))$.
- Thus, $V(s) - V^{\pi}(s) \leqslant 2\dfrac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V})\delta_{\mathfrak{F}}}{1-\gamma}$.

Example 1: Robustness to model mismatch in MDPs

Real-world model (P, r)

Simulation model $(\hat{P}, \hat{r})$

What is the loss in performance if we choose a policy using the simulation model and use it in the real world?

Model mismatch as an AIS

- $(\text{Identity}, \hat{P}, \hat{r})$ is an (ε, δ)-AIS with $\varepsilon = \sup_{s,a} \big| r(s,a) - \hat{r}(s,a) \big|$ and $\delta_{\mathfrak{F}} = \sup_{s,a} d_{\mathfrak{F}}(P(\cdot|s,a), \hat{P}(\cdot|s,a))$.
- Thus, $V(s) - V^{\pi}(s) \leqslant 2\dfrac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V})\delta_{\mathfrak{F}}}{1-\gamma}$.

$d_{\mathfrak{F}}$ is total variation

$$V(s) - V^{\pi}(s) \leqslant \frac{2\varepsilon}{1-\gamma} + \frac{\gamma \delta\, \mathbf{span}(r)}{(1-\gamma)^2}$$

Recover bounds of Müller (1997).

Example 1: Robustness to model mismatch in MDPs

Real-world model (P, r)

Simulation model $(\hat{P}, \hat{r})$

What is the loss in performance if we choose a policy using the simulation model and use it in the real world?

Model mismatch as an AIS

- (**Identity**, $\hat{P}, \hat{r}$) **is an** (ε, δ)**-AIS with** $\varepsilon = \sup_{s,a} \left| r(s,a) - \hat{r}(s,a) \right|$ **and** $\delta_{\mathfrak{F}} = \sup_{s,a} d_{\mathfrak{F}}(P(\cdot|s,a), \hat{P}(\cdot|s,a))$.
- **Thus,** $V(s) - V^{\pi}(s) \leqslant 2\dfrac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V}) \delta_{\mathfrak{F}}}{1-\gamma}$.

$d_{\mathfrak{F}}$ is total variation

$$V(s) - V^{\pi}(s) \leqslant \frac{2\varepsilon}{1-\gamma} + \frac{\gamma \delta \, \mathbf{span}(r)}{(1-\gamma)^2}$$

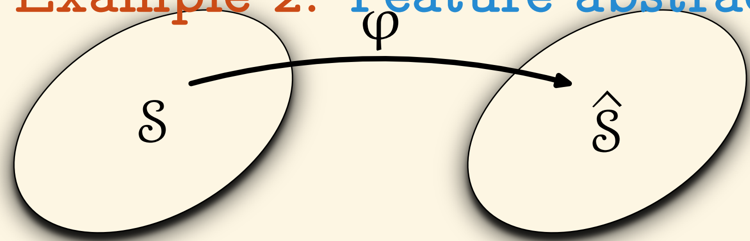
Recover bounds of Müller (1997).

$d_{\mathfrak{F}}$ is Wasserstein distance

$$V(s) - V^{\pi}(s) \leqslant \frac{2\varepsilon}{1-\gamma} + \frac{2\gamma \delta L_r}{(1-\gamma)(1-\gamma L_p)}$$

Recover bounds of Asadi, Misra, Littman (2018).

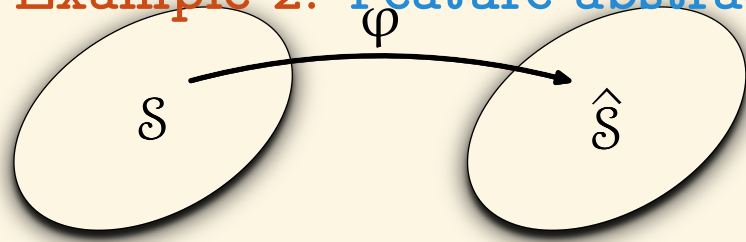
Example 2: Feature abstraction in MDPs



$(\widehat{P}, \widehat{r})$ **is determined from** (P, r) **using** φ

What is the loss in performance if we choose a policy using the abstract model and use it in the original model?

Example 2: Feature abstraction in MDPs



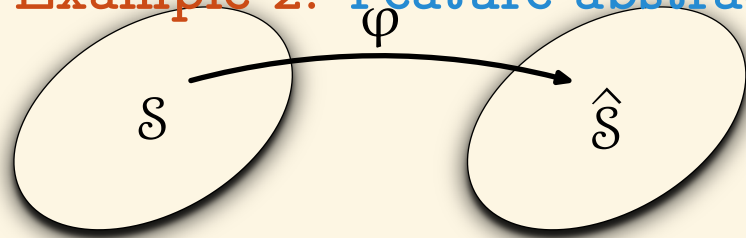
$(\hat{P}, \hat{r})$ **is determined from** (P, r) **using** φ

What is the loss in performance if we choose a policy using the abstract model and use it in the original model?

Feature abstraction as AIS

- $(\text{Identity}, \hat{P}, \hat{r})$ **is an** (ε, δ)**-AIS with** $\varepsilon = \sup_{s,a} \big| r(s,a) - \hat{r}(\varphi(s), a) \big|$ **and** $\delta_{\mathfrak{F}} = \sup_{s,a} d_{\mathfrak{F}}(P(\varphi^{-1}(\cdot)|s,a), \hat{P}(\cdot|\varphi(s),a)$.
- **Thus,** $V(s) - V^{\pi}(s) \leqslant 2\dfrac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V})\delta_{\mathfrak{F}}}{1-\gamma}$.

Example 2: Feature abstraction in MDPs



$(\hat{P}, \hat{r})$ **is determined from** (P, r) **using** φ

What is the loss in performance if we choose a policy using the abstract model and use it in the original model?

Feature abstraction as AIS

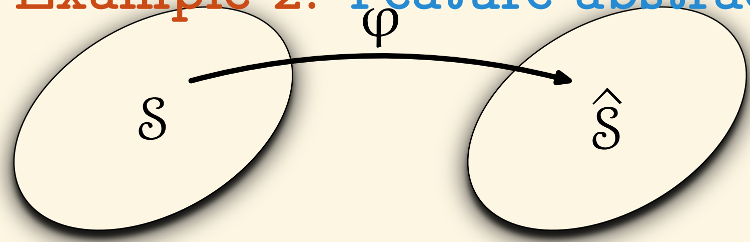
- $(\text{Identity}, \hat{P}, \hat{r})$ **is an** (ε, δ)**-AIS with** $\varepsilon = \sup_{s,a} \big| r(s,a) - \hat{r}(\varphi(s), a) \big|$ **and** $\delta_{\mathfrak{F}} = \sup_{s,a} d_{\mathfrak{F}}(P(\varphi^{-1}(\cdot)|s,a), \hat{P}(\cdot|\varphi(s),a)$.
- **Thus,** $V(s) - V^{\pi}(s) \leqslant 2\dfrac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V})\delta_{\mathfrak{F}}}{1-\gamma}$.

$d_{\mathfrak{F}}$ is total variation

$$V(s) - V^{\pi}(s) \leqslant \frac{2\varepsilon}{1-\gamma} + \frac{\gamma \delta_{\mathfrak{F}}\, \mathbf{span}(r)}{(1-\gamma)^2}$$

Improve **bounds of Abel et al. (2016)**

Example 2: Feature abstraction in MDPs



$(\hat{P}, \hat{r})$ **is determined from** (P, r) **using** φ

What is the loss in performance if we choose a policy using the abstract model and use it in the original model?

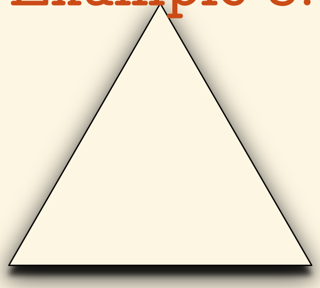
Feature abstraction as AIS

- **(Identity,** $\hat{P}, \hat{r})$ **is an** (ε, δ)**-AIS with** $\varepsilon = \sup_{s,a} \big| r(s,a) - \hat{r}(\varphi(s), a) \big|$ **and** $\delta_{\mathfrak{F}} = \sup_{s,a} d_{\mathfrak{F}}(P(\varphi^{-1}(\cdot)|s,a), \hat{P}(\cdot|\varphi(s), a)$.
- **Thus,** $V(s) - V^{\pi}(s) \leqslant 2\dfrac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V})\delta_{\mathfrak{F}}}{1-\gamma}$.

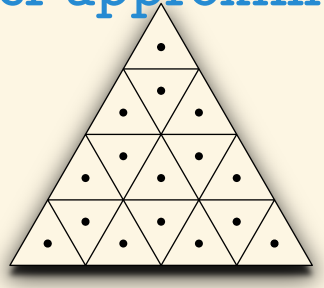
$d_{\mathfrak{F}}$ is total variation
$$V(s) - V^{\pi}(s) \leqslant \frac{2\varepsilon}{1-\gamma} + \frac{\gamma \delta_{\mathfrak{F}}\,\mathbf{span}(r)}{(1-\gamma)^2}$$
Improve **bounds of Abel et al. (2016)**

$d_{\mathfrak{F}}$ is Wasserstein distance
$$V(s) - V^{\pi}(s) \leqslant \frac{2\varepsilon}{1-\gamma} + \frac{2\gamma \delta_{\mathfrak{F}} \|\hat{V}\|_{\mathrm{Lip}}}{(1-\gamma)^2}$$
Recover bounds of Gelada et al. (2019).

Example 3: Belief approximation in POMDPs



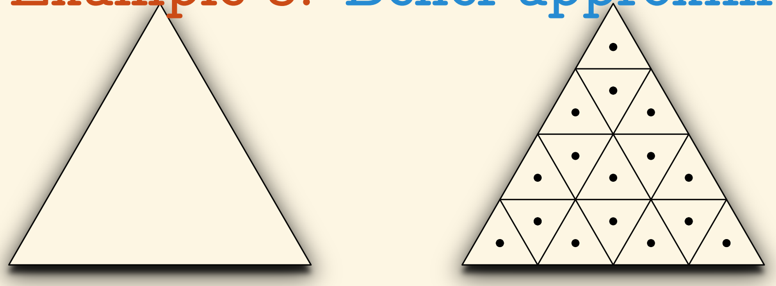
Belief space



Quantized beliefs

What is the loss in performance if we choose a policy using the approximate beliefs and use it in the original model?

Example 3: Belief approximation in POMDPs

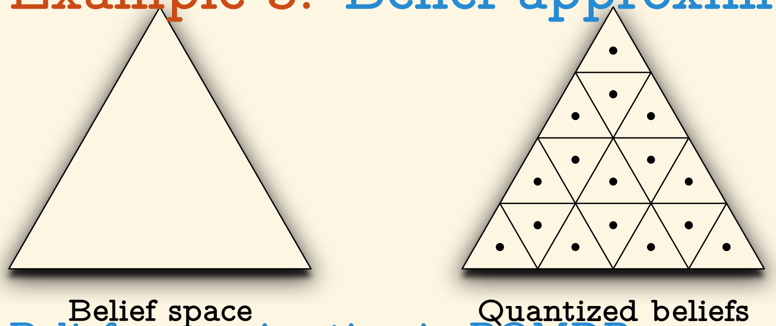


What is the loss in performance if we choose a policy using the approximate beliefs and use it in the original model?

Belief approximation in POMDPs

- ε-sufficient statistics defined in Francois-Lavet et al. (2019) as $d_{TV}(\hat{b}_t(\cdot|\phi_t(h_t)), b_t(\cdot|h_t)) \leqslant \varepsilon$
- We can show that an ε-sufficient statistic is an $(\varepsilon\|r\|_\infty, 3\varepsilon)$-AIS (wrt to the bounded Lipscitz metric).

Example 3: Belief approximation in POMDPs



What is the loss in performance if we choose a policy using the approximate beliefs and use it in the original model?

Belief approximation in POMDPs

- ε-sufficient statistics defined in Francois-Lavet et al. (2019) as $d_{TV}(\hat{b}_t(\cdot|\phi_t(h_t)), b_t(\cdot|h_t)) \leqslant \varepsilon$
- We can show that an ε-sufficient statistic is an $(\varepsilon\|r\|_\infty, 3\varepsilon)$-AIS (wrt to the bounded Lipscitz metric).

$$V(s) - V^\pi(s) \leqslant \frac{2\varepsilon\|r\|_\infty}{1-\gamma} + \frac{6\gamma\varepsilon\|r\|_\infty}{(1-\gamma)^2}$$

Improve bounds of Francois Lavet et al. (2019) by a factor of $1/(1-\gamma)$.

Thus, the notion of AIS unifies many of the approximation results in the literature, both for MDPs and POMDPs.

Outline

Information state

Approximate information state

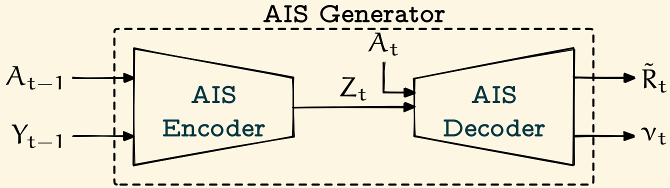
AIS based RL

- There are two approximation errors in the definition of AIS.
- Use these approximation errors as a surrogate loss
- Performs better than SOTA RL algorithms for POMDPs.

Reinforcement learning setup

AIS Generator

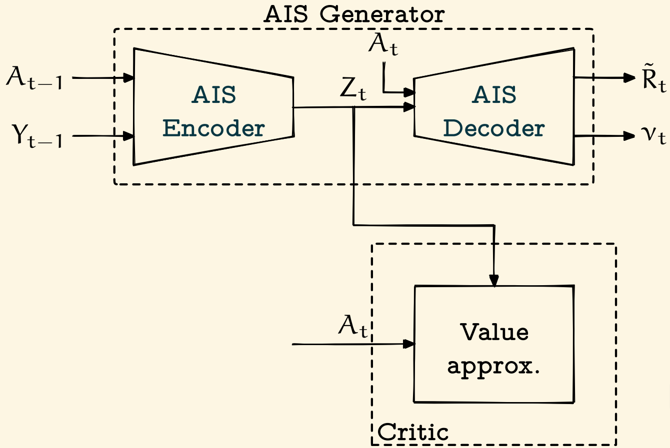
- AIS generator: an LSTM for $\sigma_t \colon \mathcal{H}_t \to \mathcal{Z}$ and a NN for functions $\hat{r}$ and $\hat{P}$.
- Use $\lambda(\tilde{R}_t - R_t)^2 + (1-\lambda) d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ as a surrogate loss fn.
- When IPM is Wasserstein distance or maximum mean discrepancy, $\nabla d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ can be computed efficiently.



Reinforcement learning setup

AIS Generator

- AIS generator: an LSTM for $\sigma_t : \mathcal{H}_t \to \mathcal{Z}$ and a NN for functions $\hat{r}$ and $\hat{P}$.
- Use $\lambda(\tilde{R}_t - R_t)^2 + (1-\lambda) d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ as a surrogate loss fn.
- When IPM is Wasserstein distance or maximum mean discrepancy, $\nabla d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ can be computed efficiently.



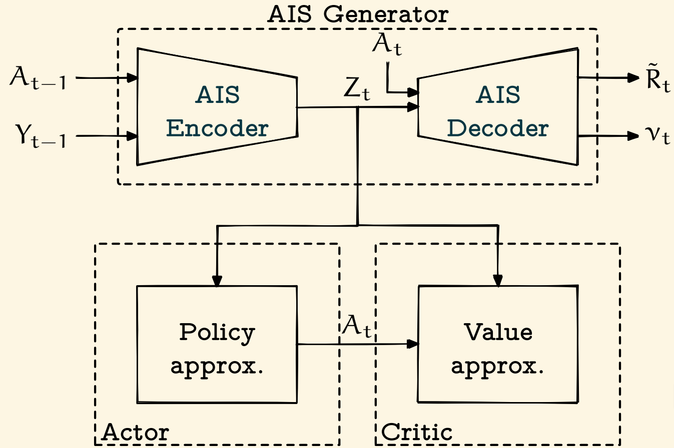
Value approximator

- Use a NN to approx. action-value function $Q : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$.
- Update the parameters to minimize temporal difference loss

Reinforcement learning setup

AIS Generator

- AIS generator: an LSTM for $\sigma_t : \mathcal{H}_t \to \mathcal{Z}$ and a NN for functions $\hat{r}$ and $\hat{P}$.
- Use $\lambda(\tilde{R}_t - R_t)^2 + (1-\lambda) d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ as a surrogate loss fn.
- When IPM is Wasserstein distance or maximum mean discrepancy, $\nabla d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ can be computed efficiently.



Value approximator

- Use a NN to approx. action-value function $Q : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$.
- Update the parameters to minimize temporal difference loss

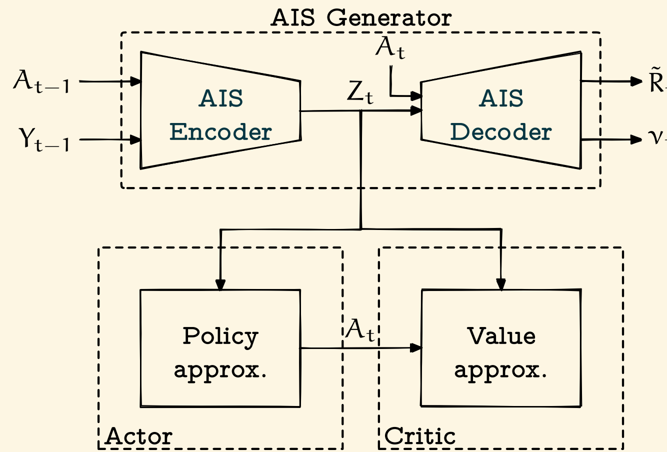
Policy approximator

- Use a NN to approx. policy $\pi : \mathcal{Z} \to \Delta(\mathcal{A})$.
- Use policy gradient theorem to efficiently compute $\nabla J(\pi)$.

Reinforcement learning setup

Convergence Guarantees

- Use multi timescale stochastic approximation to simultaneously learn AIS generator, action-value function, and policy.
- Under appropriate technical assumptions, converges to the stationary point corresponding to the choice of function approximators.



Value approximator

- Use a NN to approx. action-value function $Q\colon \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$.
- Update the parameters to minimize temporal difference loss

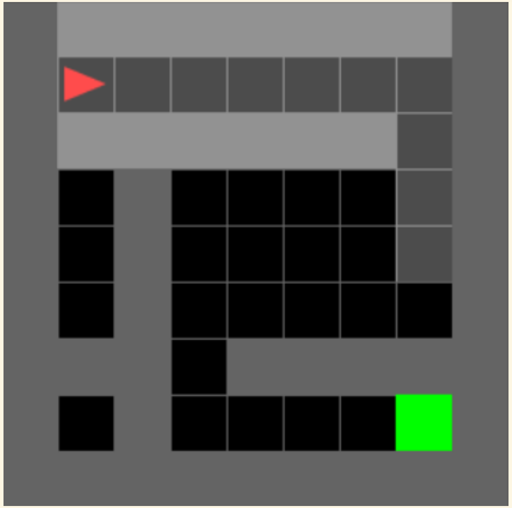
Policy approximator

- Use a NN to approx. policy $\pi\colon \mathcal{Z} \to \Delta(\mathcal{A})$.
- Use policy gradient theorem to efficiently compute $\nabla J(\pi)$.

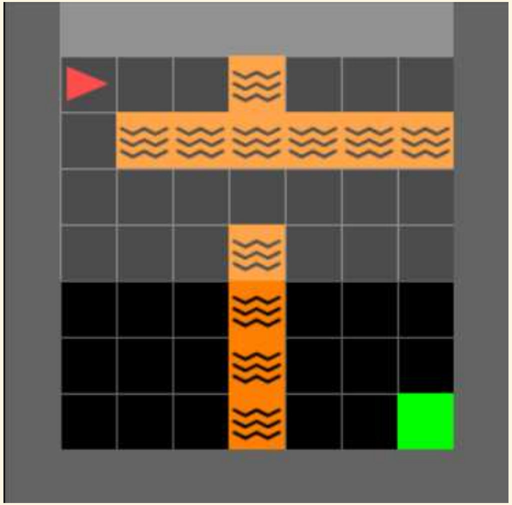
Numerical Experiments

MiniGrid Environments

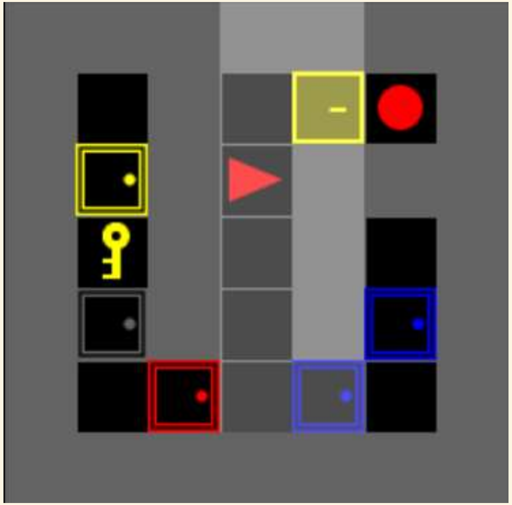
Features	
	▷ Partially observable 2D grids. Agent has a view of a 7×7 field in front of it. Observations are obstructed by walls.
	▷ Multiple entities (agents, walls, lava, boxes, doors, and keys)
	▷ Multiple actions (Move Forward, Turn Left, Turn Right, Open Door/Box, Pick up Item, Drop Item, Done).



Simple Crossing



Lava Crossing

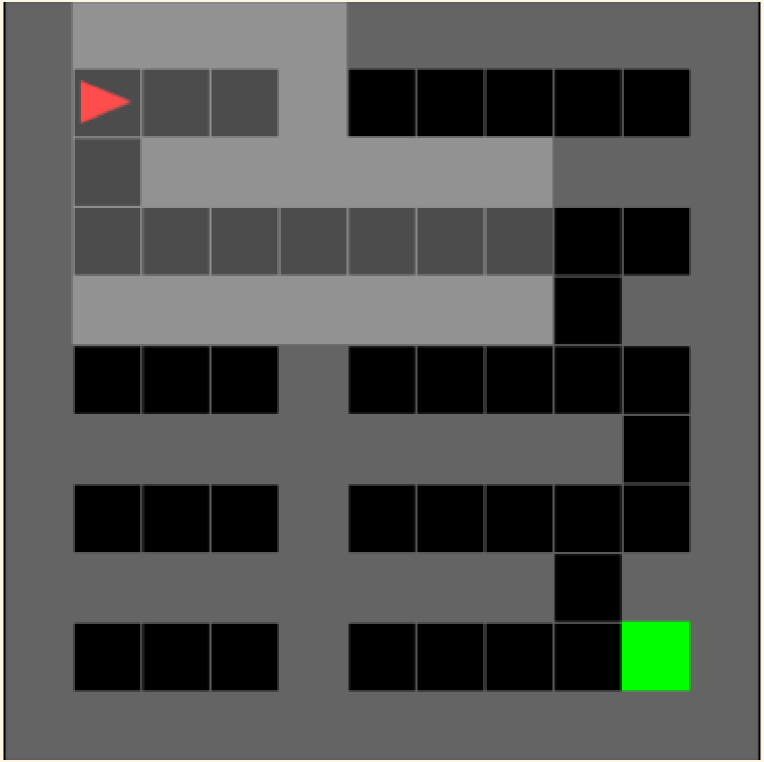


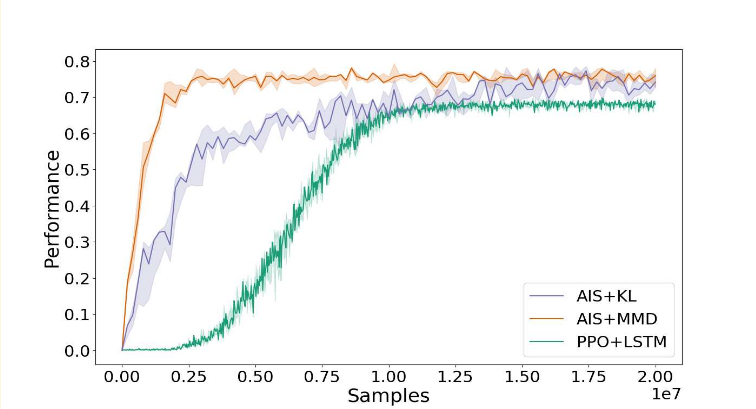
Key Corrdior

Baselines

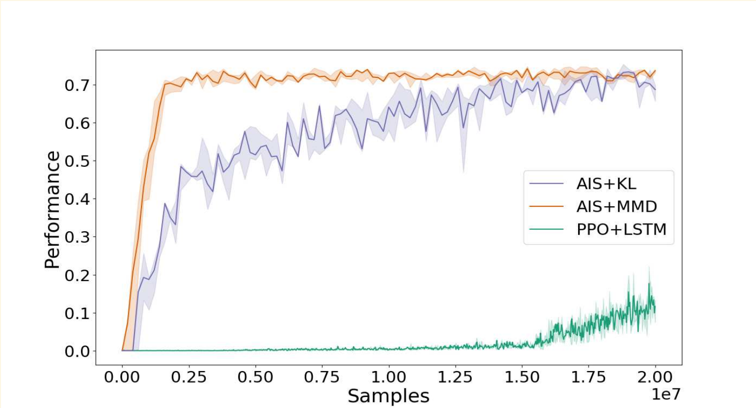
AIS + MMD	AIS based algorithm where maximum mean discrepancy (MMD) is used as an IPM.
AIS + KL	AIS based algorithm where Wasserstein distance is used as an IPM. In our experiments, we use KL divergence, which is an upper bound for Wasserstein distance and is easier to compute.
PPO + LSTM	Baseline proposed in the paper introducing the minigrid environments.

Simple Crossing



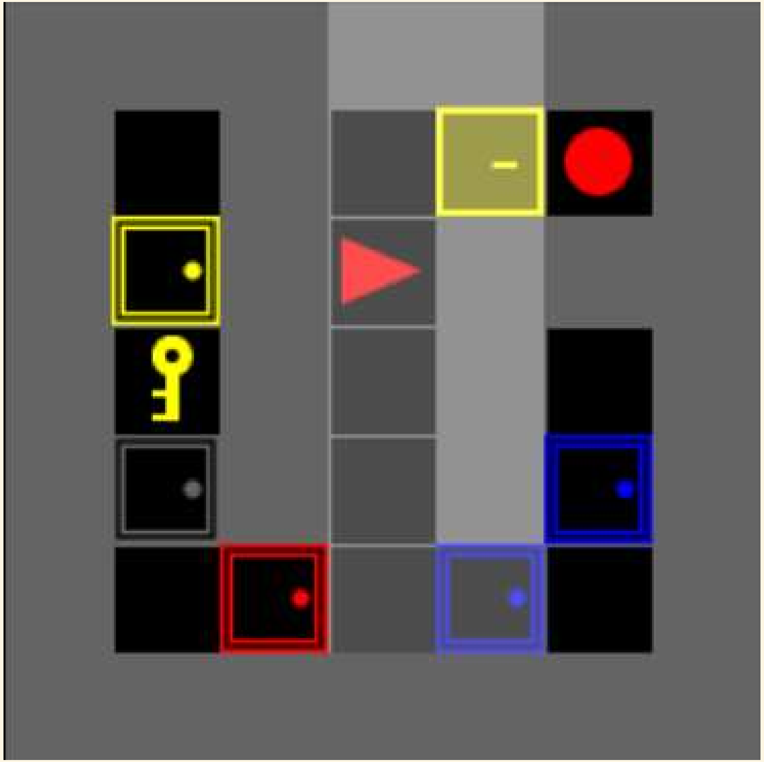


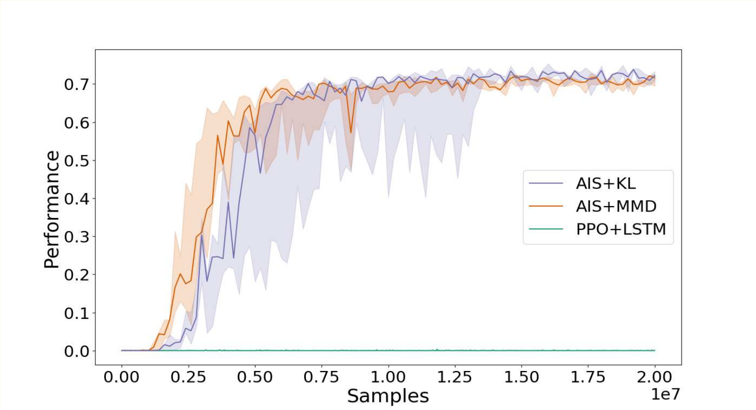
Simple Crossing S9N3



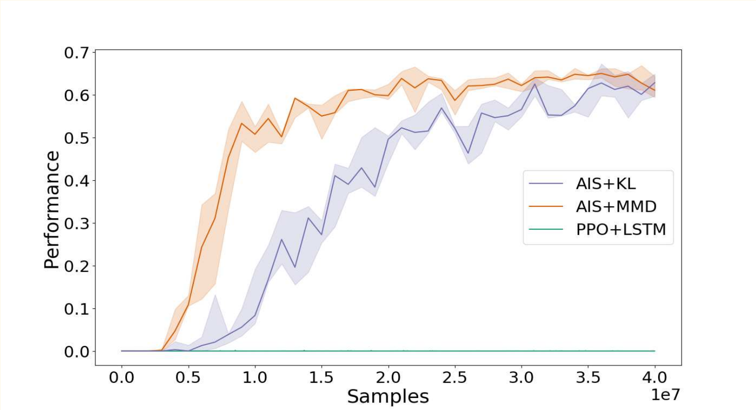
Simple Crossing S11N5

Key Corridor



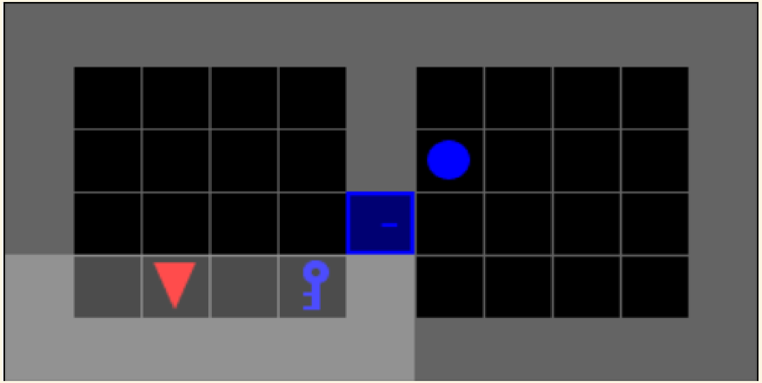


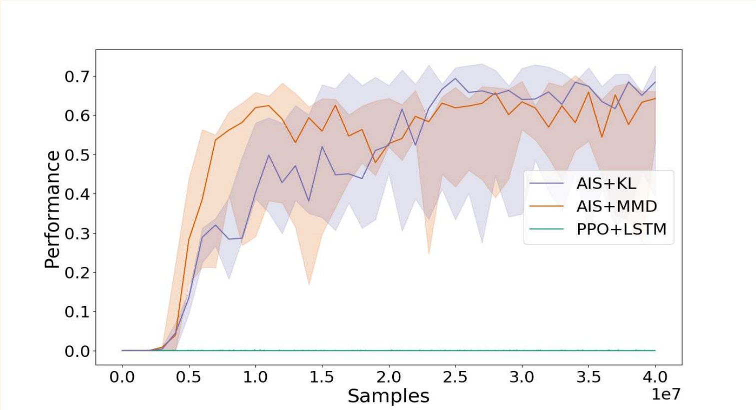
Key Corridor S3R2



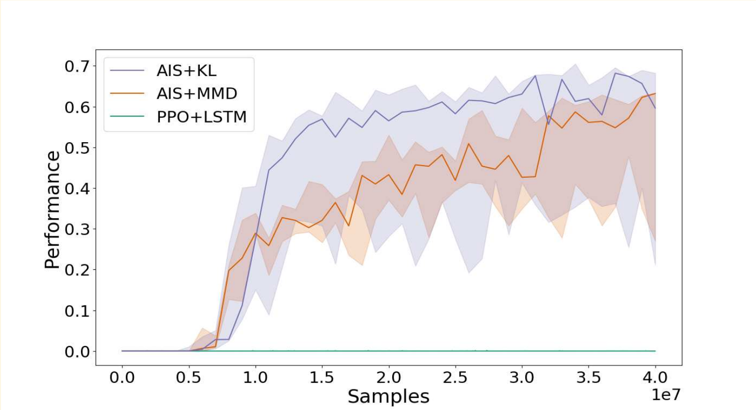
Key Corridor S3R3

Obstructed Maze





Obstructed Maze 1Dl



Obstructed Maze 1Dlh

Summary

Summary

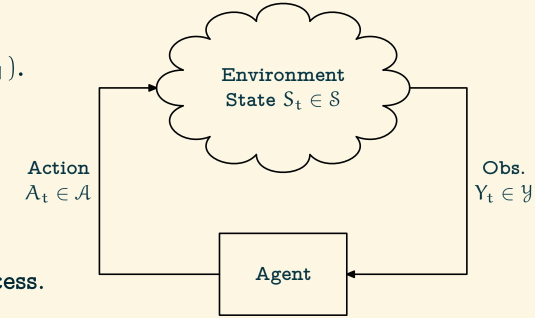
Review: Planning in partially observable environments

Key simplifying idea

Define belief state $B_t \in \Delta(\mathcal{S})$ as $B_t(s) = \mathbb{P}(S_t = s \mid Y_{1:t}, A_{1:t-1})$.

- Belief state updates in a state-like manner
  $$B_{t+1} = \text{function}(B_t, Y_{t+1}, A_t).$$
- Belief state is sufficient to evaluate rewards
  $$\mathbb{E}[R_t \mid Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t).$$

Thus, $\{B_t\}_{t \geqslant 1}$ is a perfectly observed controlled Markov process.



Therefore, we get the following results:

Structure of optimal policy	There is no loss of optimality in choosing the action A_t as a function of the belief state B_t
Dynamic Program	The optimal control policy is given by the solution of the following DP: $V_t(b_t) = \max\limits_{a_t \in \mathcal{A}} \Big\{ \hat{r}(S_t, A_t) + \mathbb{E}[V_{t+1}(B_{t+1}) \mid B_t = b_t, A_t = a_t] \Big\}$

Summary

Formal definition of information state

Information State	Given a Banach space $\mathcal{Z}$, a collection $\{\sigma_t : \mathcal{H}_t \to \mathcal{Z}\}_{t \geq 1}$ is called an information state generator if there exist a reward function $\hat{r}$ and a transition kernel $\hat{P}$ such that they are: (P1) Sufficient for performance evaluation: $$\mathbb{E}[R_t \mid H_t = h_t, A_t = a_t] = \hat{r}(\sigma_t(h_t), a_t).$$ (P2) Sufficient for predicting itself: $$\mathbb{P}(Z_{t+1} = z_{t+1} \mid H_t = h_t, A_t = a_t) = \hat{P}(z_{t+1} \mid \sigma_t(h_t), a_t).$$

Info State based dynanmic program	Let $\{Z_t\}_{t \geq 1}$ be any information state proces. Define $$V(z) = \max_{a \in \mathcal{A}} \Big\{ \hat{r}(z, a) + \gamma \int_{\mathcal{Z}} V(z_+) \hat{P}(dz_+ \mid z, a) \Big\}$$ Let $\pi^*(z)$ denote the arg max of the RHS. Then, the policy $\pi = (\pi_1, \pi_2, \dots)$ given by $\pi_t = \pi^* \circ \sigma_t$ is optimal.

Summary

AIS based approximation bounds

Let $\hat{V}$ be the fixed point of the following equations:

$$\hat{V}(z, a) = \max_{a \in \mathcal{A}} \Big\{ \hat{r}(z, a) + \gamma \int_{\mathcal{Z}} \hat{V}(z_+) \hat{P}(dz_+ | z, a) \Big\}$$

Let V denote the optimal value and action-value functions.

Then, we have the following:

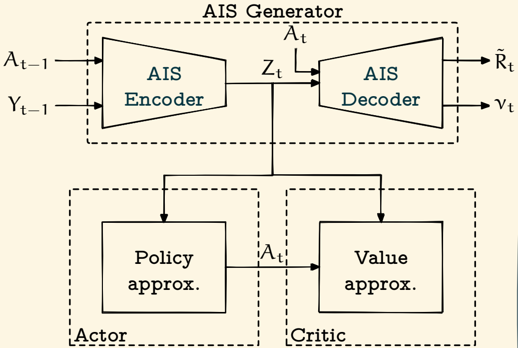
Value function approximation	The value function $\hat{V}$ is approximately optimal, i.e., $$\lvert V_t(h_t) - \hat{V}(\sigma_t(h_t)) \rvert \leqslant \alpha = \frac{\varepsilon + \gamma \rho_{\mathfrak{F}}(\hat{V}) \delta}{1 - \gamma}.$$
Policy approximation	Let $\hat{\pi}^* \colon \mathcal{Z} \to \Delta(\mathcal{A})$ be an optimal policy for $\hat{V}$. Then, the policy $\pi = (\pi_1, \pi_2, \dots)$ given by $\pi_t = \hat{\pi}^* \circ \sigma_t$ is approx. optimal: $$V_t(h_t) - V_t^{\pi}(h_t) \leqslant 2\alpha.$$

Summary

Reinforcement learning setup

AIS Generator

- AIS generator: an LSTM for $\sigma_t \colon \mathcal{H}_t \to \mathcal{Z}$ and a NN for functions $\hat{r}$ and $\hat{P}$.
- Use $\lambda(\tilde{R}_t - R_t)^2 + (1-\lambda) d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ as a surrogate loss fn.
- When IPM is Wasserstein distance or maximum mean discrepancy, $\nabla d_{\mathfrak{F}}(\mu_t, \nu_t)^2$ can be computed efficiently.



Value approximator

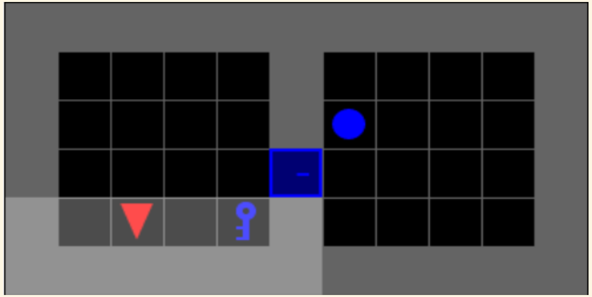
- Use a NN to approx. action-value function $Q \colon \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$.
- Update the parameters to minimize temporal difference loss

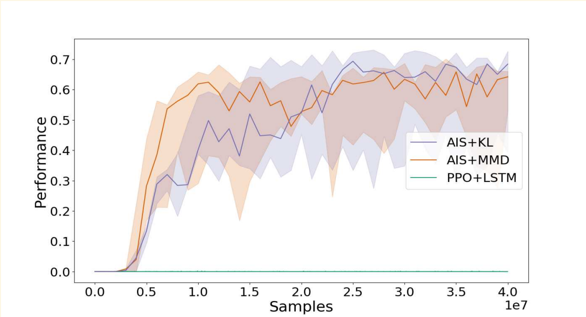
Policy approximator

- Use a NN to approx. policy $\pi \colon \mathcal{Z} \to \Delta(\mathcal{A})$.
- Use policy gradient theorem to efficiently compute $\nabla J(\pi)$.

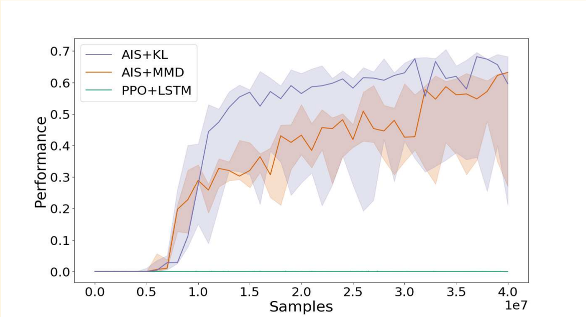
Summary

Obstructed Maze





Obstructed Maze 1Dl



Obstructed Maze 1Dlh

Approx. planning and learning–(Mahajan)

Concluding thoughts

A conceptually clean framework for approximate
DP and online RL in partially observed systems

Other results in the paper

- Generalizations to observation compression, action quantization, and lifelong learning.
- Generalizations to multi-agent systems.

Ongoing work

- Thinking about other RL settings such as offline RL, model based RL, inverse RL.
- A building block for multi-agent RL.
- . . .

- **email**: aditya.mahajan@mcgill.ca
- **web**: http://cim.mcgill.ca/~adityam

Thank you

- **paper**: https://arxiv.org/abs/2010.08843
- **code**: https://github.com/info-structures/ais

Funding: NSERC, DND