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
Approx. planning and learning—(Mahajan)

Recent successes of RL

- Algorithms based on comprehensive theory

**Alpha Go**
Recent successes of RL

- Algorithms based on comprehensive theory

Approx. planning and learning—(Mahajan)
Recent successes of RL

- Algorithms based on comprehensive theory

Robotic grasping

Approx. planning and learning—(Mahajan)
Recent successes of RL

- Algorithms based on comprehensive theory
- The theory is restricted almost exclusively to systems with **perfect state observations**.
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:** $\mathbb{P}(S_{t+1} | S_t, A_t)$
- **Observations:** $\mathbb{P}(Y_t | S_t)$
- **Reward:** $R_t = r(S_t, A_t)$.

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

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

The objective is to choose a policy $\pi$ to maximize:

$$J(\pi) := \mathbb{E}^\pi \left[ \sum_{t=1}^{\infty} \gamma^{t-1} R_t \right]$$
**POMDP: PARTIALLY OBSERVABLE MARKOV DECISION PROCESS**

- **Dynamics:** $\mathbb{P}(S_{t+1} | S_t, A_t)$
- **Observations:** $\mathbb{P}(Y_t | S_t)$
- **Reward** $R_t = r(S_t, A_t)$.
- **Action:** $A_t = \pi_t(Y_1:t, A_1:t-1)$.
  - $\pi = (\pi_t)_{t \geq 1}$ is called a policy.

The objective is to choose a policy $\pi$ 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.
**Key simplifying idea**

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

- 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
  $$E[R_t | Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t).$$

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

Define **belief state** $B_t \in \Delta(S)$ 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:
$$V_t(b_t) = \max_{a_t \in \mathcal{A}} \left\{ \hat{r}(S_t, A_t) + \mathbb{E}[V_{t+1}(B_{t+1}) \mid B_t = b_t, A_t = a_t] \right\}$$
### Implications of the modeling framework

<table>
<thead>
<tr>
<th>Implications for planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Allows to use the entire machinery of fully observed Markov decision processes for partially observed systems.</td>
</tr>
<tr>
<td>- Various exact and approximate algorithms can efficiently solve the DP.</td>
</tr>
<tr>
<td>Exact: incremental pruning, witness algorithm, linear support algo</td>
</tr>
<tr>
<td>Approximate: QMDP, point based methods, SARSOP, DESPOT, ...</td>
</tr>
</tbody>
</table>
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 theoretically grounded method for RL in partially observable models which has strong empirical performance for high-dimensional environments.
## 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 approximately satisfies these properties.
- We show that an AIS always leads to an approximate dynamic program.
- Recover (and improve upon) 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

Control input: $A_t$ → Control input: $A_t$ → Output: $Y_t$
Stochastic input: $W_t$ → Stochastic input: $W_t$ → Reward: $R_t$

Stochastic System

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

$(Y_1, R_1)$ $(Y_2, R_2)$ $(Y_t, R_t)$

$A_1 W_1$ $A_2 W_2$ $A_t W_t$
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 $\pi = (\pi_1, \pi_2, \ldots)$ denote the control policy.

The objective is to choose a policy $\pi$ to maximize:

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

Approx. planning and learning—(Mahajan)
From sufficient statistics to information state

<table>
<thead>
<tr>
<th>S</th>
<th>Y</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Obs.</td>
<td>Action</td>
</tr>
</tbody>
</table>

Sufficient Statistics

\[ Z = \sigma(Y) \text{ is a sufficient statistic for (the purpose of) evaluating the reward } R = r(S, A) \text{ if} \]

\[(P1) \quad E[R \mid Y = y, A = a] = E[R \mid Z = \sigma(y), A = a] =: \hat{r}(\sigma(y), a) \]
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?
Consider a POMDP. Suppose:

\[ Z_t = \sigma_t(H_t) \] is a sufficient statistic for evaluating the reward \( R_t = r(S_t, A_t) \) if

\[(P1) \quad \mathbb{E}[R | Y = y, A = a] = \mathbb{E}[R | Z = \sigma(y), A = a] \]

\[ =: \hat{r}(\sigma(y), a) \]

From sufficient statistics to information state

<table>
<thead>
<tr>
<th>S</th>
<th>Y</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>State</td>
<td>Obs.</td>
<td>Action</td>
</tr>
</tbody>
</table>

Sufficient Statistics

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:

\[(P2) \quad \mathbb{P}(Z_{t+1} = z_{t+1} | H_t = h_t, A_t = a_t) = \mathbb{P}(Z_{t+1} = z_{t+1} | Z_t = \sigma_t(H_t), A_t = a_t) \]
Informally, an information state is a compression of the history which is sufficient for performance evaluation and predicting itself.
Formal definition of information state

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:
\[
\mathbb{P}(Z_{t+1} = z_{t+1} \mid H_t = h_t, A_t = a_t) = \hat{P}(z_{t+1} \mid \sigma_t(h_t), a_t).
\]
### Formal definition of information state

Given a Banach space $\mathcal{Z}$, a collection $\{\sigma_t: \mathcal{H}_t \rightarrow \mathcal{Z}\}_{t \geq 1}$ 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 dynamic program

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

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

Let $\pi^*(z)$ denote the arg max of the RHS. Then, the policy $\pi = (\pi_1, \pi_2, \ldots)$ 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

<table>
<thead>
<tr>
<th>Markov decision processes (MDP)</th>
<th>MDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current state $S_t$ is an info state</td>
<td>$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state</td>
</tr>
</tbody>
</table>
### Examples of information state

<table>
<thead>
<tr>
<th>Markov decision processes (MDP)</th>
<th>MDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current state $S_t$ is an info state</td>
<td>$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state</td>
</tr>
</tbody>
</table>

- **POMDP**
  - Belief state is an info state
### Examples of information state

<table>
<thead>
<tr>
<th>Markov decision processes (MDP)</th>
<th>MDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current state $S_t$ is an info state</td>
<td>$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>POMDP</th>
<th>POMDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belief state is an info state</td>
<td>$(\mathbb{P}(S_{t-\delta}</td>
</tr>
</tbody>
</table>
## Examples of information state

<table>
<thead>
<tr>
<th>Markov decision processes (MDP)</th>
<th>MDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current state $S_t$ is an info state</td>
<td>$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>POMDP</th>
<th>POMDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belief state is an info state</td>
<td>$(\mathbb{P}(S_{t-\delta}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linear Quadratic Guassian (LQG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The state estimate $\mathbb{E}[S_t</td>
</tr>
</tbody>
</table>
### Examples of information state

<table>
<thead>
<tr>
<th>Markov decision processes (MDP)</th>
<th>MDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current state $S_t$ is an info state</td>
<td>$(S_{t-\delta+1}, A_{t-\delta+1:t-1})$ is an info state</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>POMDP</th>
<th>POMDP with delayed observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Belief state is an info state</td>
<td>$(\mathbb{P}(S_{t-\delta}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linear Quadratic Guassian (LQG)</th>
<th>Machine Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>The state estimate $\mathbb{E}[S_t</td>
<td>H_t]$ is an info state</td>
</tr>
</tbody>
</table>
Outline

**Information state**

**Approximate information state**
- Information state is defined in terms of two properties.
- An AIS is a process which approximately satisfies these properties.
- We show that an AIS always leads to an approximate dynamic program.
- Recover (and improve upon) many existing results in the literature.

**AIS based RL**
Approximate information state (AIS)

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

(P1) Sufficient for approximate performance evaluation:
\[
|\mathbb{E}[R_t \mid H_t = h_t, A_t = a_t] - \hat{r}(\sigma_t(h_t), a_t)| \leq \varepsilon
\]

(P2) Sufficient for predicting itself approximately:
\[
d_\tilde{s}(\mathbb{P}(Z_{t+1} = \cdot \mid H_t = h_t, A_t = a_t), \hat{P}(\cdot \mid \sigma_t(h_t), a_t)) \leq \delta
\]
# Approximate information state (AIS)

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

**(P1)** Sufficient for approximate performance evaluation:

\[
|\mathbb{E}[R_t \mid H_t = h_t, A_t = a_t] - \hat{r}(\sigma_t(h_t), a_t)| \leq \epsilon
\]

**(P2)** Sufficient for predicting itself approximately:

\[
d_\mathfrak{H}(\mathbb{P}(Z_{t+1} = \cdot \mid H_t = h_t, A_t = a_t), \hat{P}(\cdot \mid \sigma_t(h_t), a_t)) \leq \delta
\]

## Metrics on probability measures

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

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

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

The Minkowski function \( \rho_{\mathcal{F}} \) with respect to \( \mathcal{F} \) is given by

\[
\rho_{\mathcal{F}}(f) = \inf\{ \rho \in \mathbb{R}_{\geq 0} : \rho^{-1} f \in \mathcal{F} \}.
\]
Integral probability metrics (IPMs)

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

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

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

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

Examples of IPM

- **Total variation distance** corresponds to $\mathcal{F} = \{ f : \|f\|_{\infty} \leq 1 \}$.
- **Kolmogorov distance** corresponds to $\mathcal{F} = \{ 1_{(-\infty, t]} : t \in \mathbb{R} \}$.
- **Wasserstein distance** corresponds to $\mathcal{F} = \{ f : \|f\|_{\text{Lip}} \leq 1 \}$.
- **Maximum mean discrepancy** corresponds to $\mathcal{F} = \{ f \in \mathcal{H} : \|f\|_{\mathcal{H}} \leq 1 \}$, where $\mathcal{H}$ is a RKHS.
AIS based approximation bounds

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

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

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:

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

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.,
  \[
  |V(h_t) - \hat{V}(\sigma_t(h_t))| \leq \alpha = \varepsilon + \gamma \rho(\hat{V}) \delta \left( 1 - \gamma \right).
  \]

- **Policy approximation**: Let \( \hat{\pi}^*: \mathcal{Z} \to \Delta(\mathcal{A}) \) be an optimal policy for \( \hat{V} \). Then, the policy \( \pi = (\pi_1, \pi_2, \ldots) \) given by \( \pi_t = \hat{\pi}^* \circ \sigma_t \) is approx. optimal:
  \[
  V(h_t) - V_t^{\pi}(h_t) \leq 2\alpha.
  \]
Examples of AIS
Example 1: Robustness to model mismatch in MDPs

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

Model mismatch as an AIS

$L (\hat{P}, \hat{r})$ is an $(\epsilon, \delta)$-AIS with $\epsilon = \sup_{s, a} |r(s, a) - \hat{r}(s, a)|$ and $\delta = \sup_{s, a} d_{\overline{\mathcal{S}}}(P(\cdot|s, a), \hat{P}(\cdot|s, a))$.

Thus, $V(s) - V^\pi(s) \leq 2 \frac{\epsilon + \gamma \rho_{\overline{\mathcal{S}}}(\hat{V}) \delta}{1 - \gamma}$.
Example 1: Robustness to model mismatch in MDPs

Model mismatch as an AIS

- (Identity, \( \hat{P}, \hat{r} \)) is an \((\varepsilon, \delta)\)-AIS with \( \varepsilon = \sup_{s,a} |r(s, a) - \hat{r}(s, a)| \) and \( \delta_{\hat{\mathcal{B}}} = \sup_{s,a} d_{\hat{\mathcal{B}}}(P(\cdot|s,a), \hat{P}(\cdot|s,a)) \).
- Thus, \( V(s) - V^\pi(s) \leq \frac{2 \varepsilon + \gamma \rho_{\hat{\mathcal{B}}} \delta_{\hat{\mathcal{B}}}}{1 - \gamma} \).

Where \( d_{\hat{\mathcal{B}}} \) is total variation.

\[ V(s) - V^\pi(s) \leq \frac{2\varepsilon}{1 - \gamma} + \frac{\gamma \delta \text{span}(r)}{(1 - \gamma)^2} \]

Recover bounds of Müller (1997).

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

Model mismatch as an AIS

- (Identity, \( \hat{P}, \hat{r} \)) is an \((\varepsilon, \delta)\)-AIS with \( \varepsilon = \sup_{s,a} |r(s, a) - \hat{r}(s, a)| \) and \( \delta_{\tilde{\mathcal{S}}} = \sup_{s,a} d_{\tilde{\mathcal{S}}}(P(\cdot|s,a), \hat{P}(\cdot|s,a)) \).
- Thus, \( V(s) - V^\pi(s) \leq 2\varepsilon + \gamma \rho_{\tilde{\mathcal{S}}}(\hat{V}) \delta_{\tilde{\mathcal{S}}} \).

\[ d_{\tilde{\mathcal{S}}} \text{ is total variation} \]

\[ V(s) - V^\pi(s) \leq \frac{2\varepsilon}{1 - \gamma} + \frac{\gamma \delta \text{span}(r)}{(1 - \gamma)^2} \]

Recover bounds of Müller (1997).

\[ d_{\tilde{\mathcal{S}}} \text{ is Wasserstein distance} \]

\[ V(s) - V^\pi(s) \leq \frac{2\varepsilon}{1 - \gamma} + \frac{2\gamma \delta L_r}{(1 - \gamma)(1 - \gamma L_p)} \]


What is the loss in performance if we choose a policy using the simulation model and use it in the real world?
Example 2: Feature abstraction in MDPs

\( \hat{S} \)

\( \varphi \)

\( (\hat{P}, \hat{r}) \) is determined from \((P, r)\) using \( \varphi \)

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

Feature abstraction as AIS

- (Identity, $\hat{P}, \hat{r}$) is an $(\varepsilon, \delta)$-AIS with $\varepsilon = \sup_{s,a} |r(s,a) - \hat{r}(\varphi(s), a)|$ and $\delta_{\hat{S}} = \sup_{s,a} d_{\hat{S}}(P(\varphi^{-1}(\cdot)|s,a), \hat{P}(\cdot|\varphi(s),a))$.

- Thus, $V(s) - V^\pi(s) \leq 2 \frac{\varepsilon + \gamma \rho_{\hat{S}}(\hat{V}) \delta_{\hat{S}}}{1 - \gamma}$.

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}) \text{ is determined from } (P, r) \text{ using } \varphi

Feature abstraction as AIS

\textbf{Identity, } \hat{P}, \hat{r} \text{ is an } (\varepsilon, \delta)-\text{AIS with } \varepsilon = \sup_{s,a} |r(s, a) - \hat{r}(\varphi(s), a)| \text{ and } \delta_{\hat{S}} = \sup_{s,a} d_{\hat{S}}(P(\varphi^{-1}(\cdot)|s,a), \hat{P}(\cdot|\varphi(s), a)).

\textbf{Thus, } V(s) - V^\pi(s) \leq 2\varepsilon + \gamma \rho_{\hat{S}}(\hat{V}) \delta_{\hat{S}}.

\hspace{2cm} d_{\hat{S}} \text{ is total variation}

\hspace{2cm} V(s) - V^\pi(s) \leq \frac{2\varepsilon}{1 - \gamma} + \frac{\gamma \delta_{\hat{S}} \text{span}(r)}{(1 - \gamma)^2}

\textbf{Improve bounds of Abel et al. (2016)}

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

\( \hat{\mathcal{S}} \) is determined from \((P, r)\) using \( \varphi \)

Feature abstraction as AIS

- (Identity, \( \hat{P}, \hat{r} \)) is an \((\varepsilon, \delta)\)-AIS with \( \varepsilon = \sup_{s,a} |r(s, a) - \hat{r}(\varphi(s), a)| \) and \( \delta_{\hat{\mathcal{S}}} = \sup_{s,a} d_{\hat{\mathcal{S}}}(P(\varphi^{-1}(\cdot)|s,a), \hat{P}(\cdot|\varphi(s), a)) \).

- Thus, \( V(s) - V^\pi(s) \leq 2\varepsilon + \gamma \rho_{\hat{\mathcal{S}}}(\hat{V}) \delta_{\hat{\mathcal{S}}} \).

<table>
<thead>
<tr>
<th>( d_{\hat{\mathcal{S}}} ) is total variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V(s) - V^\pi(s) \leq \frac{2\varepsilon}{1-\gamma} + \frac{\gamma \delta_{\hat{\mathcal{S}}} \text{span}(r)}{(1-\gamma)^2} )</td>
</tr>
<tr>
<td>Improve bounds of Abel et al. (2016)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( d_{\hat{\mathcal{S}}} ) is Wasserstein distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>( V(s) - V^\pi(s) \leq \frac{2\varepsilon}{1-\gamma} + \frac{2\gamma \delta_{\hat{\mathcal{S}}} | \hat{V} |_{\text{Lip}}}{(1-\gamma)^2} )</td>
</tr>
<tr>
<td>Recover bounds of Gelada et al. (2019)</td>
</tr>
</tbody>
</table>

What is the loss in performance if we choose a policy using the abstract model 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?
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

- $\epsilon$-sufficient statistics defined in Francois-Lavet et al. (2019) as $d_{TV}(\hat{b}_t(\cdot | \phi_t(h_t)), b_t(\cdot | h_t)) \leq \epsilon$

- We can show that an $\epsilon$-sufficient statistic is an $(\epsilon \| r \|_{\infty}, 3\epsilon)$-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

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

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

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

Approx. planning and learning–(Mahajan)
Thus, the notion of AIS unifies many of the approximation results in the literature, both for MDPs and POMDPs.
### Outline

<table>
<thead>
<tr>
<th>Information state</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approximate information state</td>
</tr>
<tr>
<td>AIS based RL</td>
</tr>
</tbody>
</table>

- 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.
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 $\lambda (\tilde{R}_t - R_t)^2 + (1 - \lambda) d_\mathcal{F}(\mu_t, \nu_t)^2$ as a surrogate loss fn.
- When IPM is Wasserstein distance or maximum mean discrepancy, $\nabla d_\mathcal{F}(\mu_t, \nu_t)^2$ can be computed efficiently.

Reinforcement learning setup

Approx. planning and learning--(Mahajan)
**Reinforcement learning setup**

**AIS Generator**

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

**Value approximator**

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

Approx. planning and learning—(Mahajan)
Reinforcement learning setup

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

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

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

---

Approx. planning and learning–(Mahajan)
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 $Q: \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$.
- 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 $\nabla J(\pi)$. 

---

Approx. planning and learning–(Mahajan)
Numerical Experiments
MiniGrid Environments

Features

- Partially observable 2D grids. Agent has a view of a $7 \times 7$ 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 Corridor

Approx. planning and learning–(Mahajan)
## Baselines

<table>
<thead>
<tr>
<th>Baseline</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIS + MMD</td>
<td>AIS based algorithm where maximum mean discrepancy (MMD) is used as an IPM.</td>
</tr>
<tr>
<td>AIS + KL</td>
<td>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.</td>
</tr>
<tr>
<td>PPO + LSTM</td>
<td>Baseline proposed in the paper introducing the minigrid environments.</td>
</tr>
</tbody>
</table>
Simple Crossing

Approx. planning and learning—(Mahajan)
Key Corridor

Approx. planning and learning—(Mahajan)

Key Corridor S3R2

Key Corridor S3R3
Obstructed Maze

Approx. planning and learning—(Mahajan)
Summary

Approx. planning and learning—(Mahajan)
**Summary**

**Review: Planning in partially observable environments**

**Key simplifying idea**

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

- 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
  
  $E[R_t | Y_{1:t}, A_{1:t}] = \hat{r}(B_t, A_t)$.

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

Therefore, we get the following results:

<table>
<thead>
<tr>
<th>Structure of optimal policy</th>
<th>There is no loss of optimality in choosing the action $A_t$ as a function of the belief state $B_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Program</td>
<td>The optimal control policy is given by the solution of the following DP: $V_t(b_t) = \max_{a_t \in A} \left{ \hat{r}(S_t, A_t) + E[V_{t+1}(B_{t+1})</td>
</tr>
</tbody>
</table>

Approx. planning and learning—(Mahajan)
Summary

Formal definition of information state

Information State

Given a Banach space $\mathcal{Z}$, a collection $\{\sigma_t: \mathcal{H}_t \rightarrow \mathcal{Z}\}_{t \geq 1}$ 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:
$$E[R_t \mid H_t = h_t, A_t = a_t] = \hat{r}(\sigma_t(h_t), a_t).$$

(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 dynamic program

Let $\{Z_t\}_{t \geq 1}$ be any information state process. Define

$$V(z) = \max_{a \in \mathcal{A}} \left\{ \hat{r}(z, a) + \gamma \int_{\mathcal{Z}} V(z') \hat{P}(dz' \mid z, a) \right\}$$

Let $\pi^*(z)$ denote the arg max of the RHS. Then, the policy $\pi = (\pi_1, \pi_2, \ldots)$ 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:

\[
\hat{V}(z, a) = \max_{a \in \mathcal{A}} \left\{ \hat{r}(z, a) + \gamma \int_{\mathcal{Z}} \hat{V}(z+) \hat{P}(dz_+|z, a) \right\}
\]

Let \( V \) denote the optimal value and action-value functions.

Then, we have the following:

<table>
<thead>
<tr>
<th>Value function approximation</th>
<th>The value function ( \hat{V} ) is approximately optimal, i.e.,</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(</td>
</tr>
</tbody>
</table>

| Policy approximation          | Let \( \hat{\pi}^* : \mathcal{Z} \rightarrow \Delta(\mathcal{A}) \) be an optimal policy for \( \hat{V} \). |
|                              | Then, the policy \( \pi = (\pi_1, \pi_2, \ldots) \) given by \( \pi_t = \hat{\pi}^* \circ \sigma_t \) is approx. optimal: |
|                              | \( V_t(h_t) - V^\pi_t(h_t) \leq 2\alpha \). |
Belief state is sufficient to evaluate rewards.

Let $B_t$ be the belief state updates in a state-like manner.

Then, we have the following:

$$V(z) = \max_{a \in A} \{ \hat{r}(z,a) + \gamma \int_{h(B^t)} B^t [R(z, a(t))|y_{1:t} - R(t)] \}$$

where $\hat{r}(z,a)$ is the approx. action-value function.

To minimize temporal difference loss, we use a NN to approx. the action-value function $Q: Z \times A \rightarrow \mathbb{R}$.

The optimal control policy is given by the solution of the following DP:

$$\pi^* = \{ a \in A | \hat{r}(z,a) + \gamma \int_{h(B^t)} B^t \hat{r}(\sigma(B^t),a(t)) \}$$

Let $\pi = \{ a \in A | \hat{r}(\sigma(B^t),a(t)) \}$ be an optimal policy for $\pi: Z \rightarrow \Delta(A)$.

Use the policy gradient theorem to efficiently compute $\nabla J(\pi)$.

Use a NN to approx. policy $\pi: Z \rightarrow \Delta(A)$.

The optimal control policy is given by the solution of the following DP:

$$\pi^* = \{ a \in A | \hat{r}(\sigma(B^t),a(t)) \}$$

Let $\pi = \{ a \in A | \hat{r}(\sigma(B^t),a(t)) \}$ be an optimal policy for $\pi: Z \rightarrow \Delta(A)$.

Use a NN to approx. policy $\pi: Z \rightarrow \Delta(A)$.

Policy approximator

Approx. planning and learning–(Mahajan)
Summary

Obstructed Maze

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.
- ...
Thank you