The Online Discovery Problem and Its Application to Lifelong Reinforcement Learning

Emma Brunskill
Carnegie Mellon University

Lihong Li
Microsoft Research

Multi-disciplinary Conference on Reinforcement Learning and Decision Making 2015
Edmonton, AB, Canada

Full version to be available on arXiv
Lifelong Learning Example: Intelligent Tutoring Systems
Lifelong Learning Example: Intelligent Tutoring Systems

Alice

State = (courses taken, skills mastered, grades, ...
Lifelong Learning Example: Intelligent Tutoring Systems

Action ∈ \{ test skill, teach new concept, review old lectures, ... \}

State = (courses taken, skills mastered, grades, ...)

Alice
Lifelong Learning Example: Intelligent Tutoring Systems

Action ∈ \{ test skill, teach new concept, review old lectures, ... \}

Alice

State = (courses taken, skills mastered, grades, ...)

Lifelong Learning Example: Intelligent Tutoring Systems

Action ∈ \{ test skill, teach new concept, review old lectures, ... \}

State = (courses taken, skills mastered, grades, ...)

Alice
Lifelong Learning Example: Intelligent Tutoring Systems

Action ∈ \{ test skill, teach new concept, review old lectures, ... \}

State = (courses taken, skills mastered, grades, ...)

How to benefit from past teaching experience?
Lifelong Learning Example: Intelligent Tutoring Systems

Action ∈ \{ test skill, teach new concept, review old lectures, ... \}

State = (courses taken, skills mastered, grades, ...)

How to benefit from past teaching experience?

How to teach Alice to benefit future students?
Task as Finite Markov Decision Process (MDP)

\[ M = (S, A, P, R, \gamma) \]

- **State space**
- **Action space**
- **Transition probabilities**
- **Reward function**
- **Discount factor in (0,1)**

\[ E[r_t] = R(s_t, a_t) \]
\[ s_{t+1} \sim P(\cdot | s_t, a_t) \]
A Class of Lifelong RL Problems

• Given (known): $S$ (finite), $A$ (finite), $\gamma \in (0,1)$
• Unknown: $M = \{M^1, M^2, \ldots, M^C\}$
  $\forall M \in M, M = \langle S, A, P_M, R_M, \gamma \rangle$

For $t = 1, 2, \ldots, T$
• Environment chooses an unknown $M_t \in M$
• Agent acts in $M_t$ for $H$ steps

Note: Many previous works on LLRL with different setups
A Class of Lifelong RL Problems

• Given (known): $S$ (finite), $A$ (finite), $\gamma \in (0,1)$
• Unknown: $\mathcal{M} = \{M^1, M^2, \ldots, M^C\}$
  $$\forall M \in \mathcal{M}, M = \langle S, A, P_M, R_M, \gamma \rangle$$

For $t = 1, 2, \ldots, T$
• Environment chooses an unknown $M_t \in \mathcal{M}$
• Agent acts in $M_t$ for $H$ steps

Note: Many previous works on LLRL with different setups
A Class of Lifelong RL Problems

• Given (known): \( S \) (finite), \( A \) (finite), \( \gamma \in (0,1) \)
• Unknown: \( \mathcal{M} = \{M^1, M^2, \ldots, M^C\} \)
  \[ \forall M \in \mathcal{M}, M = \langle S, A, P_M, R_M, \gamma \rangle \]

For \( t = 1, 2, \ldots, T \)
• Environment chooses an unknown \( M_t \in \mathcal{M} \)
• Agent acts in \( M_t \) for \( H \) steps

Note: Many previous works on LLRL with different setups

Finite \( S \) and \( A \)

Finitely many MDPs with “large” model differences

Examples
• Student types w/ varying learning rates [Liu&Koedinger]
• User types in human robot interaction [Nikolaidis et al.]
• User goal recognition for task assistance [Fern et al.]
Two Kinds of Exploration

\( \mathcal{M} \): set of discovered types before \( t \)

Current task

\( M_{t-1} \rightarrow M_t \rightarrow M_{t+1} \)
Two Kinds of Exploration

$\hat{M}$: set of discovered types before $t$

When in $M_t$...
- Within-task learning:
  - Goal: maximize reward in $M_t$
  - Explore promising states in $M_t$ until policy is $\epsilon$-optimal

- Cross-task knowledge transfer:
  - Goal: maximize reward in $M_{t+1}$, … with transferable info.
  - Explore possibly all states in $M_t$ to discover novel types
**Two Kinds of Exploration**

\( \hat{M} \): set of discovered types before \( t \)

When in \( M_t \)...

- **Within-task learning:**
  - Goal: maximize reward in \( M_t \)
  - Explore *promising* states in \( M_t \) until policy is \( \epsilon \)-optimal

- **Cross-task knowledge transfer:**
Two Kinds of Exploration

\( \hat{M} \): set of discovered types before \( t \)

Current task

When in \( M_t \)...
- Within-task learning:
  - Goal: maximize reward in \( M_t \)
  - Explore promising states in \( M_t \) until policy is \( \epsilon \)-optimal

- Cross-task knowledge transfer:
  - Goal: maximize reward in \( M_{t+1}, \ldots \) w/ transferable info.
  - Explore possibly all states in \( M_t \) to discover novel types
Two Kinds of Exploration

\( \bar{M} \): set of discovered types before \( t \)

When in \( M_t \)...  
  o Within-task learning:
    - Goal: maximize reward in \( M_t \)
    - Explore promising states in \( M_t \) until policy is \( \epsilon \)-optimal

  o Cross-task knowledge transfer:
Cross-task E/E tradeoff over within-task E/E tradeoff

  - Goal: maximize reward in \( M_{t+1} \), ... w/ transferable info.
  - Explore possibly all states in \( M_t \) to discover novel types
The *Online Discovery* Problem: Abstraction of Cross-task Exploration

**Environment** has an unknown set $\mathcal{M} = \{M^1, M^2, \ldots, M^C\}$

**Agent** starts with $\hat{\mathcal{M}} = \emptyset$

For $t = 1, 2, \ldots, T$

- $\epsilon \in \mathcal{M}$
- **Agent** chooses to explore ($A_t = 1$) or exploit ($A_t = 0$)
  - If $A_t = 1$, $\hat{\mathcal{M}} \leftarrow \hat{\mathcal{M}} \cup \{M_t\}$
  - Loss to agent

**Agent** aims to minimize total loss
The *Online Discovery* Problem: Abstraction of Cross-task Exploration

**Environment** has an unknown set $\mathbb{M} = \{M^1, M^2, \ldots, M^C\}$

**Agent** starts with $\hat{\mathbb{M}} = \emptyset$

For $t = 1, 2, \ldots, T$

- **Environment** selects $M_t \in \mathbb{M}$
- **Agent** chooses to explore ($A_t = 1$) or exploit ($A_t = 0$)
  - If $A_t = 1$, $\hat{\mathbb{M}} \leftarrow \hat{\mathbb{M}} \cup \{M_t\}$
- Loss to agent

**Agent** aims to minimize total loss.
The Online Discovery Problem: Abstraction of Cross-task Exploration

1. \( \mathcal{M} \leftarrow \mathcal{M} \cup \{ M_t M M M t t M t \} \)

1) or exploit \( A \leftarrow A A A t t A t t A t = 0 \)

\( \mathcal{M} \)

Environment has an unknown set \( \mathcal{M} = \{ M^1, M^2, \ldots, M^C \} \)

Agent starts with \( \hat{\mathcal{M}} = \emptyset \)

For \( t = 1, 2, \ldots, T \)

- If \( A \leftarrow t t t t = 1 \), \( \hat{\mathcal{M}} \leftarrow \hat{\mathcal{M}} \cup \{ M_t \} \)
- Agent chooses to explore \( (A_t = 1) \) or exploit \( (A_t = 0) \)
  - If \( A_t = 1 \), \( \hat{\mathcal{M}} \leftarrow \hat{\mathcal{M}} \cup \{ M_t \} \)
  - Loss to agent
The Online Discovery Problem: Abstraction of Cross-task Exploration

1. $\mathbf{M} \leftarrow \mathbf{M} \cup \{ M_t \}$

1) or exploit $(A_t = 0)$

Environment has an unknown set $\mathbf{M} = \{M^1, M^2, \ldots, M^C\}$

Agent starts with $\hat{\mathbf{M}} = \emptyset$

For $t = 1, 2, \ldots, T$

• Loss to agent
  - $M_t \in \hat{\mathbf{M}}$
    - $A_t = 0$
      - $\rho_0$
    - $A_t = 1$
      - $\rho_1$
  - $M_t \notin \hat{\mathbf{M}}$
    - $\rho_2$
    - $\rho_3$

• Agent chooses to explore ($A_t = 1$) or exploit ($A_t = 0$)

• If $A_t = 1$, $\mathbf{M} \leftarrow \mathbf{M} \cup \{M_t\}$

• Loss to agent

$(\rho_0 \ll \rho_1 \leq \rho_2 \ll \rho_3)$
The *Online Discovery* Problem: Abstraction of Cross-task Exploration

1. \( \mathcal{M} \leftarrow \mathcal{M} \cup \{M_t\} \)

1) or exploit \((A_t = 0)\)

**\(\mathcal{M}\)**

Environment has an unknown set \(\mathcal{M} = \{M^1, M^2, \ldots, M^C\}\)

**Agent** starts with \(\hat{\mathcal{M}} = \emptyset\)

For \(t = 1,2,\ldots,T\)

- Loss to agent
  - If \(A_t = 1\), \(\hat{\mathcal{M}} \leftarrow \mathcal{M} \cup \{M_t\}\)
  - If \(A_t = 0\), \(\hat{\mathcal{M}} \leftarrow \emptyset\)

- Agent chooses to explore \((A_t = 1)\) or exploit \((A_t = 0)\)

\(\rho_0 \ll \rho_1 \leq \rho_2 \ll \rho_3\)

**Agent** aims to minimize total loss
The *Online Discovery* Problem: Abstraction of Cross-task Exploration

1. $\mathcal{M} \leftarrow \mathcal{M} \cup \{ M_t \}$

1) or exploit ($A_t = 0$)

**Environment** has an unknown set $\mathcal{M} = \{ M_1, M_2, \ldots, M_C \}$

**Agent** starts with $\hat{\mathcal{M}} = \emptyset$

For $t = 1, 2, \ldots, T$

- Loss to agent
  - $M_t \in \hat{\mathcal{M}}$
    - $A_t = 0$
      - $\rho_0$
    - $A_t = 1$
      - $\rho_1$

- Agent chooses to explore ($A_t = 1$) or exploit ($A_t = 0$)
  - $\rho_0 \ll \rho_1 \leq \rho_2 \ll \rho_3$
    - $\rho_0$: successful transfer
    - $\rho_3$: negative transfer

**Agent** aims to minimize total loss
Explore-First Algorithm

Stochastic assumptions:
\[ M_t \sim \mu \ i.i.d. \ \text{over} \ M, \ \text{and} \ \mu_m := \min_{M \in M} \mu(M) \]

• Action selection
\[ A_t = \begin{cases} 1 & \text{if } t \leq E \\ 0 & \text{otherwise} \end{cases} \quad (\text{Exploration phase}) \]
\[ \text{otherwise} \quad (\text{Exploitation phase}) \]

• \[ = O(\mu_m^{-1} \log(C\mu_m T)), \text{ then} \]
\[ \text{AverageLoss} \leq \text{OptLoss} + \frac{1}{T \cdot \mu_m} \log \left( \frac{Tc\mu_m \rho_3}{\rho_1} \right) \]
Explore-First Algorithm

\[ \leq \text{OptLoss} + \frac{1}{T \cdot \mu_m} \left( 11 + T \cdot \mu_m \right) \left( \sum_{m=1}^{T} \mu_m \right) \log \left( TC \mu_m \rho 3^\rho 1 \right) \]

\[ \leq \text{OptLoss} + \frac{1}{T \cdot \mu_m} \left( 11 + T \cdot \mu_m \right) \left( \sum_{m=1}^{T} \mu_m \right) \log \left( TC \mu_m \rho 3^\rho 1 \right) \]

\[ \log \left( \sum_{m=1}^{T} \mu_m \right) \log \left( \sum_{m=1}^{T} \mu_m \right) \]

\[ \log \left( \sum_{m=1}^{T} \mu_m \right) \log \left( \sum_{m=1}^{T} \mu_m \right) \]

Stochastic assumptions:

\[ M_t \sim \mu \text{ i.i.d. over } M, \text{ and } \mu_m := \min_{M \in M} \mu(M) \]

- Action selection

\[ A_t = \begin{cases} 1 & \text{if } t \leq E \quad \text{(Exploration phase)} \\ 0 & \text{otherwise} \quad \text{(Exploitation phase)} \end{cases} \]

\[ \text{AverageLoss} \leq \text{OptLoss} + \frac{1}{T \cdot \mu_m} \log \left( \frac{TC \mu_m \rho_3}{\rho_1} \right) \]

\[ \text{AverageLoss} \leq \text{OptLoss} + \frac{1}{T \cdot \mu_m} \log \left( \frac{TC \mu_m \rho_3}{\rho_1} \right) \]
Forced-Exploration Algorithm

No stochastic assumption ($M_t$ can even be generated adversarially!)

• $\eta_2 \geq \cdots \geq \eta_T > 0$
• Algorithm chooses action $A_t \sim \text{Bernoulli}(\eta_t)$

• **Theorem**: If choose $\eta_t = 1/\sqrt{t}$, then

  $\text{AverageLoss} \leq \text{OptLoss} + \frac{1}{\sqrt{T}} (2\rho_1 + C\rho_3)$
Forced-Exploration Algorithm

Bernoulli $\eta_t \eta t \eta t \eta t \eta t \eta t \eta t$
$\eta 1 \geq \eta 2 \eta 2 \eta 2 \geq \ldots \geq \eta T \eta T \eta T T T \eta T > 0$
No stochastic assumption ($M_t$ can even be generated adversarially!)

• Algorithm chooses action
\[
A_t \sim t t t t \sim \text{Bernoulli}(\eta_t) \\
A_t \sim \text{Bernoulli}(\eta_t)
\]

• **Theorem**: If choose $\eta_t = 1/\sqrt{t}$, then
\[
\text{AverageLoss} \leq \text{OptLoss} + \frac{1}{\sqrt{T}} (2\rho_1 + C\rho_3)
\]
Forced-Exploration Algorithm

\[
\text{OptLoss} + \frac{1}{\sqrt{T}} \geq \text{AverageLoss} \leq \text{OptLoss} + \frac{1}{\sqrt{T}} (2\rho_1 + C\rho_3)
\]

\[
\text{Bernoulli } \eta_t \geq \eta_{t-1} \geq \eta_{t-2} \geq \ldots \geq \eta_T > 0
\]

No stochastic assumption (\(M_t\) can even be generated adversarially!)

- Algorithm chooses action

\[
A \sim t t t \sim \text{Bernoulli}(\eta_t)
\]

- **Theorem**: If choose \(\eta_t = \frac{1}{\sqrt{t}}\), then

\[
\text{AverageLoss} \leq \text{OptLoss} + \frac{1}{\sqrt{T}} (2\rho_1 + C\rho_3)
\]
Forced-Exploration Algorithm

\[ \text{OptLoss} + \frac{1}{\sqrt{T}}(2\rho_1 + C\rho_3) \]

1/ \ t \ t t \ t, then

Bernoulli \ \eta \ t \ \eta \ t \ \eta \ t \ \eta \ t \ \eta \ t

\eta \ 1 \geq \eta \ 2 \ \eta \ 2 \ \eta \ 2 \geq \ldots \geq \eta \ T \ \eta \ \eta \ T \ T \ T \ \eta \ T > 0

No stochastic assumption (\(M_t\) can even be generated \textit{adversarially}!)}

• Algorithm chooses action

\[ A \ t \ \sim \ t \ t \ t \ \sim \ \text{Bernoulli}(\eta_t) \]

\[ \text{AverageLoss} \leq \text{OptLoss} + \frac{1}{\sqrt{T}}(2\rho_1 + C\rho_3) \]

• \textbf{Theorem}: If choose \(\eta_t = 1/\sqrt{t}\), then

\[ \text{AverageLoss} \leq \text{OptLoss} + \frac{1}{\sqrt{T}}(2\rho_1 + C\rho_3) \]

\(\text{We have an} \ \Omega \left(\frac{1}{\sqrt{T}}\right) \text{lower bound} \)

\(\Rightarrow\) \textbf{Forced-Exploration is essentially optimal}
A Lifelong RL Algorithm based on FE

**Input:** $S, A, \gamma$

Initia
A Lifelong RL Algorithm based on FE

**Input:** $S, A, \gamma$

Initia
A Lifelong RL Algorithm based on FE

**Input:** $S, A, \gamma$

Initia
A Lifelong RL Algorithm based on FE

**Input:** $S, A, \gamma$

Initia

$= t t t = 1$ if data shows $M_t$ is novel
Sample Complexity of Exploration

Sample complexity of algorithm $A$ (given $\varepsilon$) [Kakade]

Number of steps where $Q^A_t(s_t, a_t) \leq Q^*(s_t, a_t) - \varepsilon$

Measures number of $\varepsilon$-mistakes made by the algorithm

• long enough, with high prob.

$\text{SampleComplexity(Our Algorithm)} = \tilde{O}\left(\frac{CD}{\Gamma^2}T + SAN\sqrt{T}\right)$

In contrast, single-task RL’s Sample Complexity is $\Omega(SAT')$
Sample Complexity of Exploration

\[ \text{Our Algorithm} = O(\frac{CD}{\Gamma^2} T + SAN \sqrt{T}) \]

Sample complexity of algorithm \( \mathbf{A} \) (given \( \epsilon \)) [Kakade]

Number of steps where \( Q^A_t(s_t, a_t) \leq Q^*(s_t, a_t) - \epsilon \)

Measures number of \( \epsilon \)-mistakes made by the algorithm

**Theorem**: For \( H \) long enough, with high prob.

\[ \text{SampleComplexity Our Algorithm} = \tilde{O} \left( \frac{CD}{\Gamma^2} T + SAN \sqrt{T} \right) \]

In contrast, single-task RL’s Sample Complexity is \( \Omega(SAT) \)
Sample Complexity of Exploration

Our Algorithm = \( O(OO0\ CD\ \Gamma^2\ T + SAN\ T\ CD\ \Gamma^2\ CC\ DD\ CD\ \Gamma^2\ \Gamma^2\ \Gamma^2\ \Gamma^2\ \Gamma^2\ CD\ \Gamma^2\ TT + SSAANN\ T\ TTT\ T\ CD\ \Gamma^2\ T + SAN\ T) \)

Sample complexity of algorithm \( A \) (given \( \epsilon \)) [Kakade]

Number of steps where \( Q^A_t(s_t, a_t) \leq Q^*(s_t, a_t) - \epsilon \)

Measures number of \( \epsilon \)-mistakes made by the algorithm

• **Theorem**: For \( H \) long enough, with high prob.

\[
\text{Sample Complexity} \quad \text{Our Algorithm} = \tilde{O}\left(\frac{CD}{T^2} T + SAN \sqrt{T}\right)
\]

Asymptotic performance

In contrast, single-task RL's Sample Complexity is \( \Omega(SAT) \)
Sample Complexity of Exploration

**SAT SSAATT SAT**

Our Algorithm = $O \cdot O \cdot O \cdot CD \cdot \Gamma \cdot 2 \cdot T + SAN \cdot T \cdot CD \cdot \Gamma \cdot 2 \cdot C \cdot D \cdot \Gamma \cdot 2 \cdot T + SAN \cdot T$

Sample complexity of algorithm $\mathcal{A}$ (given $\epsilon$) [Kakade]

Number of steps where $Q^A_t(s_t, a_t) \leq Q^*(s_t, a_t) - \epsilon$

Measures number of $\epsilon$-mistakes made by the algorithm

• **Theorem**: For $H$ long enough, with high prob.

  Sample Complexity $\mathcal{O}$ur Algorithm $= \tilde{O} \left( \frac{CD}{\Gamma^2} \cdot T + SAN \sqrt{T} \right)$
Experiment

4 possible MDPs with
• noisy state transitions
• different rewarding states

Algorithms for comparison
• Forced-exploration [this work]
• Explore-first [Brunskill-Li]
• Hierarchical Multi-task Learning [Wilson et al.]
Stochastic Setting with small $\mu_m$
Adversarial Setting with Changing Distribution
Conclusions

• Two kinds of exploration needed in LLRL
• Online discovery problem as abstraction for cross-task exploration
• A new lifelong RL algorithm based on optimal ODP algorithm
  o Provably sample complexity better than single-task RL
  o Proof-of-concept experiments demonstrating desired behavior

Future work
• Function approximation
• Use of prior information