#1666 : Autonomous Exploration for Navigating in MDPs using Blackbox RL Algorithms

Pratik Gajane[1], Peter Auer[2] and Ronald Ortner[2]

[1]Eindhoven University of Technology
[2]Montanuniversität Leoben

Motivation

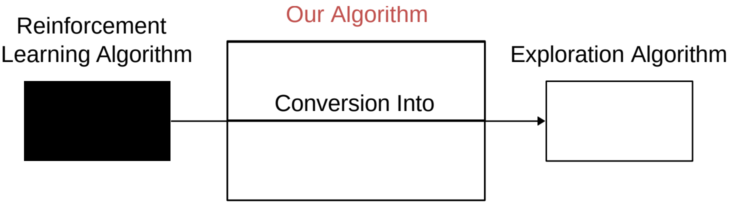
- Exploration in reinforcement learning(RL) is a hard problem.

Motivation

- Exploration in reinforcement learning(RL) is a hard problem.
- (Near)-optimal RL algorithms exist for regret minimization in various settings.

Motivation

- Exploration in reinforcement learning(RL) is a hard problem.
- (Near)-optimal RL algorithms exist for regret minimization in various settings.
- Our work :



Problem Setting

- Markov Decision Process (MDP) with
  - No external rewards and unknown transition probabilities,

Problem Setting

- Markov Decision Process (MDP) with
  - No external rewards and unknown transition probabilities,
  - Countable (possibly infinite) state space $\mathcal{S}$,

Problem Setting

- Markov Decision Process (MDP) with
  - No external rewards and unknown transition probabilities,
  - Countable (possibly infinite) state space $\mathcal{S}$,
  - Finite action space with A $\#$actions, and

Problem Setting

- Markov Decision Process (MDP) with
  - No external rewards and unknown transition probabilities,
  - Countable (possibly infinite) state space $\mathcal{S}$,
  - Finite action space with A #actions, and
  - Starting state s_0.

Problem Setting

- Markov Decision Process (MDP) with
  - No external rewards and unknown transition probabilities,
  - Countable (possibly infinite) state space $\mathcal{S}$,
  - Finite action space with A #actions, and
  - Starting state s_0.
- Assumption : In every state, RESET action available which leads back to s_0.

Problem Setting

- Markov Decision Process (MDP) with
  - No external rewards and unknown transition probabilities,
  - Countable (possibly infinite) state space $\mathcal{S}$,
  - Finite action space with A $\#$actions, and
  - Starting state s_0.
- Assumption : In every state, RESET action available which leads back to s_0.
- Input : $L \geq 1$.
  Goal : Find a policy for every state reachable from the starting state s_0 in L steps.

Reachable States

Navigation time$_\pi(s)$

Expected #steps before reaching state s for the first time following policy π from the starting state s_0.

Reachable States

Navigation time$_\pi(s)$

Expected #steps before reaching state s for the first time following policy π from the starting state s_0.

Reachable states $\mathcal{S}_L$

$\mathcal{S}_L := \{s \in \mathcal{S} \ : \min_\pi(\text{Navigation time}_\pi(s)) \leq L\}$.

Reachable States

Navigation time$_\pi(s)$

Expected #steps before reaching state s for the first time following policy π from the starting state s_0.

Reachable states $\mathcal{S}_L$

$\mathcal{S}_L := \{s \in \mathcal{S} : \min_\pi(\text{Navigation time}_\pi(s)) \leq L\}$.

- Incrementally reachable states $\mathcal{S}_L^{\rightarrow}$:= A subset of $\mathcal{S}_L$ that allows for incremental discovery.

Reachable States

Navigation time$_\pi(s)$

Expected #steps before reaching state s for the first time following policy π from the starting state s_0.

Reachable states $\mathcal{S}_L$

$\mathcal{S}_L := \{s \in \mathcal{S} : \min_\pi(\text{Navigation time}_\pi(s)) \leq L\}$.

- Incrementally reachable states $\mathcal{S}_L^{\rightarrow}$:= A subset of $\mathcal{S}_L$ that allows for incremental discovery.
- Goal : Find a policy $\forall\, s \in \mathcal{S}_L^{\rightarrow}$ with navigation time $\leq (1+\epsilon)L$.

Our proposed algorithm : META-EXPLORE

Central idea: Use an arbitrary online RL algorithm $\mathfrak{A}$ to find a suitable navigation policy for a state.

Our proposed algorithm : META-EXPLORE

- Central idea: Use an arbitrary online RL algorithm $\mathfrak{A}$ to find a suitable navigation policy for a state.
- META-EXPLORE proceeds in *rounds*.
  In each round, it evaluates a *target state*.

Our proposed algorithm : META-EXPLORE

Central idea: Use an arbitrary online RL algorithm $\mathfrak{A}$ to find a suitable navigation policy for a state.

- META-EXPLORE proceeds in *rounds*.
  In each round, it evaluates a *target state*.
- Target states are chosen from the set of candidate states.

Our proposed algorithm : META-EXPLORE

Central idea: Use an arbitrary online RL algorithm $\mathfrak{A}$ to find a suitable navigation policy for a state.

- META-EXPLORE proceeds in *rounds*.
  In each round, it evaluates a *target state*.
- Target states are chosen from the set of candidate states.
- If $(1+\epsilon)L$-step policy found for the target state ,
  Successful round and target state becomes *known*.
  Else
  Failure round.

Our proposed algorithm : META-EXPLORE

META-EXPLORE

- **Initialization:** Initialize

Set of candidate states $\mathcal{U} \leftarrow \{\}$

Set of known states $\mathcal{K} \leftarrow \{s_0\}$

Our proposed algorithm : META-EXPLORE

META-EXPLORE

- **Initialization:** Initialize

$$\text{Set of candidate states } \mathcal{U} \leftarrow \{\}$$

$$\text{Set of known states } \mathcal{K} \leftarrow \{s_0\}$$

- In each round $r = 1, 2, \ldots$

  State Discovery

  Choice of Target State

  Target State Evaluation

Our proposed algorithm : META-EXPLORE

META-EXPLORE

- **Initialization:** Initialize

$$\text{Set of candidate states } \mathcal{U} \leftarrow \{\}$$
$$\text{Set of known states } \mathcal{K} \leftarrow \{s_0\}$$

- In each round $r = 1, 2, \ldots$

  State Discovery
  Choice of Target State
  Target State Evaluation

META-EXPLORE : State Discovery

State Discovery

- Exploring the neighborhood of known states to add to the set of candidate states $\mathcal{U}$.

META-EXPLORE : State Discovery

State Discovery

- Exploring the neighborhood of known states to add to the set of candidate states $\mathcal{U}$.
- In a newly known state, every action is sampled $\tilde{O}(L)$ times.

META-EXPLORE : State Discovery

State Discovery

- Exploring the neighborhood of known states to add to the set of candidate states $\mathcal{U}$.
- In a newly known state, every action is sampled $\tilde{O}(L)$ times.
- Any newly discovered states
  and the neighboring states of previously known states
  are added to the set of candidate states $\mathcal{U}$.

META-EXPLORE

META-EXPLORE

- **Initialization:** Initialize

$$\text{Set of candidate states } \mathcal{U} \leftarrow \{\}$$
$$\text{Set of known states } \mathcal{K} \leftarrow \{s_0\}$$

- In each round $r = 1, 2, \ldots$
  - State Discovery
  - Choice of Target State
  - Target State Evaluation

META-EXPLORE : Choice of Target State

Choice of Target State

- Chosen arbitrarily from the set of candidate states.

META-EXPLORE : Choice of Target State

Choice of Target State

- Chosen arbitrarily from the set of candidate states.
- Algorithm stops when the set of candidate states is empty.

META-EXPLORE

META-EXPLORE

- **Initialization:** Initialize

  Set of candidate states $\mathcal{U} \leftarrow \{\}$

  Set of known states $\mathcal{K} \leftarrow \{s_0\}$

- In each round $r = 1, 2, \ldots$

  State Discovery
  Choice of Target State
  Target State Evaluation

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

What do we need to use an online RL algorithm $\mathfrak{A}$?

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

What do we need to use an online RL algorithm $\mathfrak{A}$?
An MDP such that regret minimization leads to time-effective navigating to the target state.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

What do we need to use an online RL algorithm $\mathfrak{A}$?
An MDP such that regret minimization leads to time-effective navigating to the target state.

- Induced MDP : In the induced MDP $\mathcal{M}_{\bar{s}}$ for target state $\bar{s}$, the learner

  has loss 0 in $\bar{s}$, and
  suffers loss 1 in every other state.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ till target $\bar{s}$ is reached f(regret($\mathfrak{A}$), L, ϵ) times.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ till target $\bar{s}$ is reached f(regret($\mathfrak{A}$), L, ϵ) times.
- Every time $\bar{s}$ is reached, record *history* of $\mathfrak{A}$ in the current round.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ till target $\bar{s}$ is reached f(regret($\mathfrak{A}$), L, ϵ) times.
- Every time $\bar{s}$ is reached, record *history* of $\mathfrak{A}$ in the current round.
- History $\equiv$ state-action-reward-next state transitions.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ till target $\bar{s}$ is reached f(regret($\mathfrak{A}$), L, ϵ) times.
- Every time $\bar{s}$ is reached, record *history* of $\mathfrak{A}$ in the current round.
- History $\equiv$ state-action-reward-next state transitions.
- A performance check (based on average #steps to reach $\bar{s}$) decides if a round is successful.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ till target $\bar{s}$ is reached f(regret($\mathfrak{A}$), L, ϵ) times.
- Every time $\bar{s}$ is reached, record *history* of $\mathfrak{A}$ in the current round.
- History $\equiv$ state-action-reward-next state transitions.
- A performance check (based on average $\#$steps to reach $\bar{s}$) decides if a round is successful.
- At the end of a successful round, $\mathcal{K} = \mathcal{K} + \bar{s}$ and all associated history points are added to the output for $\bar{s}$.

Navigation Policy for Known States

For each known state $s \in \mathcal{K}$,

1. $h \overset{\text{uniform}}{\sim}$ history points associated with s.

Navigation Policy for Known States

For each known state $s \in \mathcal{K}$,

1. $h \overset{\text{uniform}}{\sim}$ history points associated with s.
2. Run $\mathfrak{A}$ from the history point h.

Navigation Policy for Known States

For each known state $s \in \mathcal{K}$,

1. $h \overset{\text{uniform}}{\sim}$ history points associated with s.
2. Run $\mathfrak{A}$ from the history point h.
3. If s is not reached in $\approx \frac{L}{\epsilon}$ steps, RESET and go to step 1.

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$*, then with high probability, it*

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$*, then with high probability, it*

1. *discovers a set of states* $\mathcal{K} \supseteq S_L^{\rightarrow}$,

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$, *then with high probability, it*

1. *discovers a set of states* $\mathcal{K} \supseteq S_L^{\rightarrow}$,
2. *has a sample complexity better than previous work in terms of L,*

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$*, then with high probability, it*

1. *discovers a set of states* $\mathcal{K} \supseteq S_L^{\rightarrow}$,
2. *has a sample complexity better than previous work in terms of* L,
3. *for each* $s \in \mathcal{K}$*, outputs a policy with navigation time* $\leq (1 + \epsilon)L$.

Concluding Remarks

- Conversion of RL algorithms into exploration algorithms with an upper bound on sample complexity.

Concluding Remarks

- Conversion of RL algorithms into exploration algorithms with an upper bound on sample complexity.
- Not included in this presentation : Experimental results.

Concluding Remarks

- Conversion of RL algorithms into exploration algorithms with an upper bound on sample complexity.
- Not included in this presentation : Experimental results.

Thank You.

Scan the following to see the paper

See you at the poster D1.

References

[LA12] Shiau Hong Lim and Peter Auer. "Autonomous Exploration For Navigating In MDPs". In: *Proceedings of the 25th Annual Conference on Learning Theory*. 2012, pp. 40.1–40.24.

[TPVL20] Jean Tarbouriech et al. "Improved Sample Complexity for Incremental Autonomous Exploration in MDPs". In: *Advances in Neural Information Processing Systems*. 2020, pp. 11273–11284.

Diameter of an MDP

Consider the stochastic process defined by a stationary policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$ operating on an MDP M with initial state s_0. Let $T(s'|M, \pi, s)$ be the random variable for the first time step in which state s' is reached in this process. Then the diameter of M is defined as

$$D(M) := \max_{s \neq s'} \min_{\pi:\mathcal{S}\rightarrow\mathcal{A}} \mathbb{E}\left[T(s'|M, \pi, s)\right]$$

Incrementally Reachable States : Definition

Definition (Incrementally reachable states)

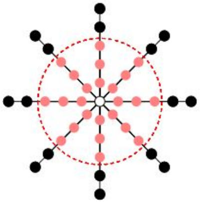
Let $\prec$ be some partial order on $\mathcal{S}$. The set $\mathcal{S}_L^{\prec}$ of states reachable in L steps with respect to $\prec$, is defined inductively as follows:

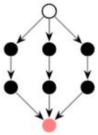
- $s_0 \in \mathcal{S}_L^{\prec}$,
- if there is a policy π on $\{s' \in \mathcal{S}_L^{\prec} : s' \prec s\}$ with navigation time$\pi(s)$ $\leq L$, then $s \in \mathcal{S}_L^{\prec}$.

We define the set $\mathcal{S}_L^{\rightarrow}$ of states incrementally reachable in L steps with respect to some partial order to be $\mathcal{S}_L^{\rightarrow} := \bigcup_{\prec} \mathcal{S}_L^{\prec}$, where the union is over all possible partial orders.

Incrementally Reachable States : Illustration

- Two environments where the starting state s_0 is shown in white.
- On the left, each transition is deterministic and is depicted with an edge.
- On the right, each transition from s_0 to the first layer is equiprobable, and the rest of the transitions are deterministic.
- For $L = 3$, states belonging to S_L are shown in pink.
- On the left, $\mathcal{S}_L^{\rightarrow} = \mathcal{S}_L$. On the right, $\mathcal{S}_L^{\rightarrow} = \{s_o\} \neq \mathcal{S}_L$.





Induced MDP :Definition

In the induced MDP $\mathcal{M}_{\bar{s}}$ for target state $\bar{s}$,

- all the actions in state $\bar{s}$ have loss 0 and lead back to s_0,

Induced MDP :Definition

In the induced MDP $\mathcal{M}_{\bar{s}}$ for target state $\bar{s}$,

- all the actions in state $\bar{s}$ have loss 0 and lead back to s_0,
- all the states $\{s|s \notin \mathcal{K} \wedge s \neq \bar{s}\}$ merged into an auxiliary state at which only RESET is possible suffering loss 1,

Induced MDP :Definition

In the induced MDP $\mathcal{M}_{\bar{s}}$ for target state $\bar{s}$,

- all the actions in state $\bar{s}$ have loss 0 and lead back to s_0,
- all the states $\{s | s \notin \mathcal{K} \wedge s \neq \bar{s}\}$ merged into an auxiliary state at which only RESET is possible suffering loss 1,
- actions in all the other states behave the same as in the original MDP and suffer loss 1.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ for $\#$*episodes* where $\#$*episodes*=f(regret($\mathfrak{A}$), L, ϵ).

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ for $\#$*episodes* where $\#$*episodes*=f(regret($\mathfrak{A}$), L, ϵ).
- Episode $:=$ begins at s_0 and ends only when $\bar{s}$ is reached.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ for $\#$*episodes* where $\#$*episodes*=f(regret($\mathfrak{A}$), L, ϵ).
- Episode $:=$ begins at s_0 and ends only when $\bar{s}$ is reached.
- At the end of an episode, record *history* of $\mathfrak{A}$ in the current round.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ for $\#$*episodes* where $\#$*episodes*=f(regret($\mathfrak{A}$), L, ϵ).
- Episode $:=$ begins at s_0 and ends only when $\bar{s}$ is reached.
- At the end of an episode, record *history* of $\mathfrak{A}$ in the current round.
- History $\equiv$ state-action-reward-next state transitions.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ for $\#$*episodes* where $\#$*episodes*=f(regret($\mathfrak{A}$), L, ϵ).
- Episode $:=$ begins at s_0 and ends only when $\bar{s}$ is reached.
- At the end of an episode, record *history* of $\mathfrak{A}$ in the current round.
- History $\equiv$ state-action-reward-next state transitions.
- A performance check (based on $\#$steps for episode completion) decides if a round is successful.

META-EXPLORE : Target State Evaluation

META-EXPLORE : Target State Evaluation

- Run $\mathfrak{A}$ on $\mathcal{M}_{\bar{s}}$ for $\#$*episodes* where $\#$*episodes*=f(regret($\mathfrak{A}$), L, ϵ).
- Episode $:=$ begins at s_0 and ends only when $\bar{s}$ is reached.
- At the end of an episode, record *history* of $\mathfrak{A}$ in the current round.
- History $\equiv$ state-action-reward-next state transitions.
- A performance check (based on $\#$steps for episode completion) decides if a round is successful.
- At the end of a successful round, $\mathcal{K} = \mathcal{K} + \bar{s}$ and all associated history points are added to the output for $\bar{s}$.

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$ *with a regret upper bound of* $B(\#\mathit{States}, \#\mathit{Actions}) \cdot T^{\alpha} \cdot D^{\beta}$, *then with prob.* $1 - \delta$, *it*

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$ *with a regret upper bound of* $B(\#States, \#Actions) \cdot T^{\alpha} \cdot D^{\beta}$*, then with prob.* $1 - \delta$*, it*

1. *discovers a set of states* $\mathcal{K} \supseteq S_L^{\rightarrow}$*,*

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$ *with a regret upper bound of* $B(\#States, \#Actions) \cdot T^{\alpha} \cdot D^{\beta}$, *then with prob.* $1-\delta$, *it*

1. *discovers a set of states* $\mathcal{K} \supseteq S_L^{\rightarrow}$,
2. *terminates after*

$$\tilde{O}\left(\frac{S^2 A \cdot [B(S,A)]^{\frac{1}{1-\alpha}} \cdot L^{2+\frac{\alpha+\beta-1}{1-\alpha}}}{\epsilon^{\max\left(4, \frac{1}{1-\alpha}\right)}}\right)$$

exploration steps, where $S := |\mathcal{K}| \leq |\mathcal{S}_{(1+\epsilon)L}^{\rightarrow}|$.

Performance Guarantees

Theorem

If META-EXPLORE *is run with an online RL algorithm* $\mathfrak{A}$ *with a regret upper bound of* $B(\#States, \#Actions) \cdot T^{\alpha} \cdot D^{\beta}$, *then with prob.* $1-\delta$, *it*

1. *discovers a set of states* $\mathcal{K} \supseteq S_L^{\rightarrow}$,
2. *terminates after*

$$\tilde{O}\left(\frac{S^2 A \cdot [B(S,A)]^{\frac{1}{1-\alpha}} \cdot L^{2+\frac{\alpha+\beta-1}{1-\alpha}}}{\epsilon^{\max\left(4, \frac{1}{1-\alpha}\right)}}\right)$$

   exploration steps, where $S := |\mathcal{K}| \leq |\mathcal{S}_{(1+\epsilon)L}^{\rightarrow}|$.

3. *for each* $s \in \mathcal{K}$, *outputs a policy with navigation time* $\leq (1+\epsilon)L$.

Relation to Existing Work

UCBEXPLORE [LA12]	$\tilde{O}(SAL^3/\epsilon^3)$
DISCO [TPVL20]	$\tilde{O}(SAGL^3/\epsilon^2)$
META-EXPLORE using UCRL2b	$\tilde{O}(S^3GA^2L^2/\epsilon^4)$