A NOTE OF CONSTRAINED EXPLORATION IN REINFORCEMENT LEARNING WITH OPTIMALITY PRESERVATION *

Author1, Author2
Affiliation
Univ
City
`{Author1, Author2}email@email`

Author3
Affiliation
Univ
City
`email@email`

1 Section 2

$|B|$ the number of elements in set B
2^B the power set of B
$\mathcal{A}$ the alphabet, whose members are characters
A string over $\mathcal{A}$ is a sequence of characters each belonging to $\mathcal{A}$
λ empty string
$|s|$ the length of string s
$\mathcal{A}^+$ the set of all possible strings over $\mathcal{A}$
$\mathcal{A}^* = \lambda \cup \mathcal{A}^+$
Prefix of string l $l = usv$, u is the prefix of l ($u \leq l$), s is the substring of l ($s \prec l$)
L A language L over $\mathcal{A}$ is a subset of $\mathcal{A}^*$
$\overline{L}$ is the prefix-closure of L, $\overline{L} = \{u \in \hat{\mathcal{A}}^* | u \leq l \text{ for some } l \in L\}$
If $\overline{L} = L$, then L is prefix-closed
$\mathcal{L}$ specification language
Automaton $(\mathcal{S}, \mathcal{A}, \delta, \Gamma, s^\circ, \mathcal{S}^\bullet)$
$\mathcal{S}$ finite set of states
$\mathcal{A}$ finite set of events (an alphabet)
δ $\mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ transition function
$\Gamma : \mathcal{S} \rightarrow 2^{\mathcal{A}}$ active event function, in RL is a set of all actions for which $\delta(s, a)$ is defined
s° initial state
$\mathcal{S}^\bullet$ marked states
$\delta(s, a)!$ there is a state $s' \in \mathcal{S}$ such that $\delta(s, a) = s'$
reachable of state s there is a string $l \in \mathcal{A}^*$ such that $\delta(s^\circ, l) = s$
co-reachable of state s there is a string $l \in \mathcal{A}^*$ such that $\delta(s, l) \in \mathcal{S}_x^\bullet$
trim automaton all of its states are both reachable and co-reachable
$\mathcal{R}_e[\mathrm{X}]$ reachable part of automaton X
$L[\mathrm{X}]$ language generated by X, defined as $L(\mathrm{X}) := \{L \in \mathcal{A}^* | \delta_x(s_x^\circ, l)!\}$
$L_m[\mathrm{X}]$ the language marked by X, all of the strings to the final state
If X is trim, then $L(X) = \overline{L_m(\mathrm{X})}$
$u \sim_x v$ equivalent strings, if there exists a state $s \in \mathcal{S}_x$ that $\delta(s_x^\circ, u) = \delta(s_x^\circ, v) = s$
$\mathbb{C}_x(s)$ the equivalent class corresponding to state s
The product of two automaton: $\mathcal{S}_z = \mathcal{S}_x \times \mathcal{S}_y, \mathcal{A}_z = \mathcal{A}_x \cup \mathcal{A}_y, \mathcal{S}_z^\bullet = \mathcal{S}_x^\bullet \times \mathcal{S}_y^\bullet, \Gamma_z((s_x, s_y)) = \Gamma_x(s_x) \cap \Gamma_y(s_y)$ with $(s_x, s_y) \in \mathcal{S}_z$

$$\delta_z((s_x, s_y), a) = \begin{cases} (\delta_x(s_x, a), \delta_y(s_y, a)) & \text{if a} \in \Gamma_{\mathrm{x}}(\mathrm{s_x}) \cap \Gamma_{\mathrm{y}}(\mathrm{s_y}) \\ \text{undefined} & \text{otherwise} \end{cases} \tag{1}$$

* *Citation*: **Authors. Title. Pages.... DOI:000000/11111.**

X is isomorphic to $X||Y$, if $\mathcal{A}_x \subseteq \mathcal{A}_y$ and $L(\mathrm{X}) \subseteq L(\mathrm{Y})$
π_b probability distribution over $\mathcal{A}_x$
$L_x^p(l)$ the probabilistic language generated by (X, π_b)

$$L_x^p(la) = \begin{cases} L_x^p(l)\pi_b(\delta_x(s_x^\circ, l), a) & \text{if } \delta_\mathrm{x}(\mathrm{s}_\mathrm{x}^\circ, \mathrm{l})! \\ L_x^p(la) = 0 & \text{otherwise} \end{cases} \tag{2}$$

2 Section 3

Unconstrained reinforcement-learning problems modeled by trim automaton

$$\mathrm{G} = (\mathcal{S}_g, \mathcal{A}_g, \delta_g, \Gamma_g, s_g^\circ, \mathcal{S}_g^\bullet)$$

with properties: $|\mathcal{S}_g| < \infty$, $\delta(s,a)!$ for all $(s,a) \in \mathcal{S}_g \times \mathcal{A}_g$, s_g° and $\mathcal{S}_g^\bullet$ are known
$\pi_b(s,a) > 0$ for all $(s,a) \in \mathcal{S}_g \times \Gamma_g(s)$
Learning process: execute A, receive a reward, update the Q
$L = \cup_{i=1}^N \bar{l}_{n_i}^{(i)}$ language, which means all action sequence's prefix-closure of all episodes
$L_m = \cup_{i=1}^N l_{n_i}^{(i)}$ the marked language, which means all sequence of all episodes.
If $N \to \infty$, then $L \to L(\mathrm{G})$ and $L_m \to L_m(\mathrm{G})$
Robbins-Monro conditions: All state-action pairs be visited infinitely often by the agent during the learning process.
Proposition 3.1 non-zero probability of being visited in unconstrained learning process

3 Section 4

Π supervisor
(G, π_b, Π) constrained learning process, use G to represent (G, π_b) or (G, π_b, Π)
$\Pi(l)$ actions belong to the admissible action set, based on l
$\Pi \subseteq \Gamma_g(s) = \mathcal{A}_g$, $\pi_b(s,a) = 0$ for all $a \notin \Pi(l)$
Definition 4.1. $\Pi : L(\mathrm{G}) \to 2^{\mathcal{A}_g}$
Explain: input the action sequence, output an admissible action set, the action set is a subset of the original action set (alphabet)
Extended feedback-control structure
$L(\Pi/\mathrm{G})$ the language generated by the agent under the supervision of Π, is defined recursively as:

$$\lambda \in L(\Pi/\mathrm{G})$$

$$[\ \ l \in L(\Pi/\mathrm{G}) \text{ and } la \in L(\mathrm{G}) \text{ and } a \in \Pi(l)\ \ ] \Leftrightarrow la \in L(\Pi/\mathrm{G})$$

$L(\Pi/\mathrm{G}) = \overline{L(\Pi/\mathrm{G})}$, $L(\Pi/\mathrm{G})$ is prefix-closed
$L(\Pi/\mathrm{G}) \subset L(\mathrm{G})$
$\mathcal{L} \subset L(\mathrm{G})$ a specification language, which is regular and prefix-closed

4 Section 5

Lemma 5.1. An effective Π exists for a prefix-closed language $\mathcal{L} \subset L(\mathrm{G})$
Proof: Define a supervisor

$$\Pi(l) = a \in \mathcal{A}_g | l \in L(\mathrm{G}) and la \in \overline{\mathcal{L}}$$

then $l \in L(\Pi/\mathrm{G})$ if and only if $l \in \overline{\mathcal{L}}$
Unconstrained RL can be implemented via the product $\mathrm{H}||\mathrm{G}$, $L(\mathrm{H}||\mathrm{G}) = L(\Pi/\mathrm{G})$ and $L_m(\mathrm{H}||\mathrm{G}) = L_m(\Pi/\mathrm{G})$
Preposition 5.1. A trim automaton H is a realization of the supervisor Π if $L(\mathrm{H}) = \mathcal{L}$
The probability of a state-action pair(s,a) being visited by the agent can be expressed as

$$p(a|s) = p_\pi(a|l) \cdot p_\pi(a|l) \cdot L_{H||G}^p(l)$$

where $L_{H||G}^p(l)$ is the probability of $l \in \mathbb{C}_g(s) \cap L(\Pi/\mathrm{G})$ generated under the supervision of Π
Lemma 5.5. $L_{H||G}^p(l) > 0$

Definition 5.2.Π is optimality-preserving if $p(a|s) > 0$ for all $(s, a) \in (\mathcal{S}_g - \mathcal{S}_g^{\bullet}) \times \mathcal{A}_g$
Definition 5.3. Let $\mathcal{L} \subset L(\mathrm{G})$. $\mathcal{L}$ covers G if, for any $(s, a) \in (\mathcal{S}_g - \mathcal{S}_g^{\bullet}) \times \mathcal{A}_g$, there exists a string $l \in \mathbb{C}_g(s) \cup \mathcal{L}$ such that $la \in \mathcal{L}$
Theorem 5.1. An effective supervisor Π is optimality-preserving if and only if $\mathcal{L}$ covers G.
Proof: based on calculation of the probability of (s, a)
Visitability of a state (in order to judge whether $\mathcal{L}$ covers G): a state s_g is visitable with respect to $\mathrm{M} = \mathrm{H}||\mathrm{G}$ if

$$\bigcup_{(s_h, s_g) \in \Omega_m(s_g)} \Gamma_m((s_h, s_g)) = \mathcal{A}_g$$

where $\Omega_m : \mathcal{S}_g \to 2^{\mathcal{S}_m}, \mathcal{S}_m = \mathcal{S}_h \times \mathcal{S}_g$ is a function.
Explanation: if we use different action sequence to reach s_g, the admission set Γ also changes. During training, we may use many sequences to reach s_g, if the total union of each Γ is the action set, then state s_g is visitable.
Lemma 5.3. If s_g is visitable with respect to M, then for any $a \in \mathcal{A}_g$, there exits a state $(s_h, s_g) \in \Omega_m(s_g)$ such that $a \in \Gamma_m((s_h, s_g))$
Definition 5.5 A automaton G is visitable with respect to M if every state in $(\mathcal{S}_g - \mathcal{S}_g^{\bullet})$ is visitable with respect to M.
Theorem 5.2. If G is visitable with respect to M, then $\mathcal{L}$ covers G.

5 Supplementary

5.1 Robbins-Monro conditions

Every state-action pair must have a non-zero probability of being visited by the agent — also known as persistent exploration.

5.2 Prefix-closed

Here's an example to illustrate this concept:

Consider the Language: Let's say we have a language L consisting of the following strings over the alphabet 0, 1: "", "0", "10", "101", "1011" . Note that "" represents the empty string.

Check for Prefixes:

The string "0" is in L, and its prefixes are "" (empty string), which is also in L. The string "10" is in L, and its prefixes are "", "1" (not in L), and "10" (in L). The string "101" is in L, and its prefixes are "", "1", "10", and "101" (only "10" and "101" are in L). The string "1011" is in L, and its prefixes are "", "1", "10", "101", and "1011" (only "10", "101", and "1011" are in L). Assess Prefix-Closed Property:

Since not all prefixes of the strings in L are also in L (for instance, "1" is a prefix of "10" but is not in L), the language L is not prefix-closed. To contrast, here is an example of a prefix-closed language:

Consider a language M = "", "0", "00", "000" . Every string in M is such that all of its prefixes are also in M. For example, the prefixes of "000" are "", "0", and "00", all of which are in M. Therefore, M is a prefix-closed language.