線形時相論理制約のもとでの階層的制御器の強化学習

学籍番号：09C13151　潮 研究室　山倉 佑馬

1　緒論

対象システムを有限 Markov 決定過程 (MDP) としたとき, 線形時相論理式 (LTL 式) で与えられた制御仕様を確率 1 で満たす制御器を強化学習で獲得する方法 [1] が提案された. 一方, 最近 LTL 式を受理条件が遷移で記述される GDRA [2] の構成方法が提案された. しかし, [1] の方法にこの GDRA を用いても学習ができない場合があることが明らかになった. 本研究では, 問題の解決のために, 階層的強化学習 [3] を用い, LTL 式で記述された制御仕様を満たす階層的制御器の強化学習の手法を提案する.

2　階層的制御器の強化学習

システムの制御仕様として記述される LTL 式 ϕ に対する GDRA $\mathcal{R}_\phi$ は, $\mathcal{R}_\phi =< Q, AP, \delta, q_0, ACC >$ で表現される.

- Q : LTL 式 ϕ を GDRA に変換したときの状態の有限集合
- AP : 原子命題の有限集合
- $\delta \subseteq Q \times 2^{AP} \times Q$: 状態の遷移関数
- $q_0 \in Q$: 初期状態
- ACC : 受理条件
  $ACC = \{ACC_i\}_{i=1}^{n_{ACC}}$, $ACC_i = (G_i, B_i)$, $i = \{1, ..., n_{ACC}\}$
  $G_i = \{G_{i,1}\ ,\ \ldots\ ,\ G_{i,l_i}\} \subseteq \delta$
  $B_i \subseteq \delta$

ラベル付き MDP $\mathcal{M}_\phi$ は, $\mathcal{M}_\phi =< S, A, P, s_0, AP, L >$ で表現される.

- S : システムの状態の有限集合
- A : エージェントがとりうる行動の集合
- $P : S \times A \times S \to [0, 1]$: システムの状態の遷移確率
- $s_0 \in S$: システムの初期状態
- AP : 原子命題の有限集合
- $L : S \to 2^{AP}$: 各状態に原子命題を割り当てるラベル関数

状態 $s \in S$ で行動 $a \in A$ をとり, 状態 $s' \in S$ に遷移する確率が $P(s, a, s')$ である. P は時刻 t 以前の行動や状態に依存せずに遷移確率を決定できる Markov 性を有するものとする. さらに, $A(s) = \{a \in A \mid \exists s' \in S,\ P(s, a, s') > 0\}$ とおく. $A(s) \subseteq A$ は状態 s において選択できる行動の集合である.

GDRA $\mathcal{R}_\phi$ とラベル付き MDP $\mathcal{M}_\phi$ の合成積をとり, さらに目標という概念を加えた Rabin 重み付き合成 MDP $\mathcal{P}$ は, $\mathcal{P} =< S_\mathcal{P}, A_\mathcal{P}, P_\mathcal{P}, s_{\mathcal{P}_0}, ACC_\mathcal{P}, Goal, W_\mathcal{P} >$ で表現される。

- $S_\mathcal{P} = S \times Q$: 状態の有限集合
  $\mathcal{P}$ の状態 $S_\mathcal{P}$ から $\mathcal{M}_\phi$ の状態 S または $\mathcal{R}_\phi$ の状態 Q を返す射影関数を $[\![\cdot]\!]_s : S_\mathcal{P} \to S$, $[\![\cdot]\!]_q : S_\mathcal{P} \to Q$ と定義する.
  $s_\mathcal{P} = (s, q) \in S_\mathcal{P}$に対して, $[\![s_\mathcal{P}]\!]_s = s, [\![s_\mathcal{P}]\!]_q = q$ である.
- $A_\mathcal{P}$: 行動の有限集合
  $A_\mathcal{P}(s_\mathcal{P}) = A([\![s_\mathcal{P}]\!]_s)$
- $P_\mathcal{P} : S_\mathcal{P} \times A_\mathcal{P} \times S_\mathcal{P} \to [0, 1]$: 状態の遷移確率

$$P_\mathcal{P}(s_\mathcal{P}, a_\mathcal{P}, s'_\mathcal{P}) = \begin{cases} P([\![s_\mathcal{P}]\!]_s, a_\mathcal{P}, [\![s'_\mathcal{P}]\!]_s) \\ \quad \text{if } ([\![s_\mathcal{P}]\!]_q, L([\![s'_\mathcal{P}]\!]_s), [\![s'_\mathcal{P}]\!]_q) \in \delta \\ 0 \quad \text{otherwise} \end{cases}$$

- $s_{\mathcal{P}_0} = (s_0,\ q_0) \in S_\mathcal{P}$: 初期状態
- $ACC_\mathcal{P}$: 受理条件
  $ACC_\mathcal{P} = \{ACC_{\mathcal{P}i}\}_{i=1}^{n_{ACC}}$, $ACC_{\mathcal{P}i} = (\mathcal{G}_i, \mathcal{B}_i)$, $i = \{1, ..., n_{ACC}\}$
  $\mathcal{G}_i = \{\mathcal{G}_{i,1}\ ,\ \ldots\ ,\ \mathcal{G}_{i,l_i}\}$, $\mathcal{G}_{i,k} = S \times G_{i,k}$, $k = \{1, \ldots, l_i\}$
  $\mathcal{B}_i = S \times B_i$
- $Goal$: 目標集合
  $Goal = \{Goal_i\}_{i=1}^{n_{ACC}}$, $Goal_i = \mathcal{G}_i$
  $goal_i \in Goal_i$ は目標と表記
- $W_\mathcal{P}$: 報酬関数
  報酬は, 受理条件 $(\mathcal{G}_i, \mathcal{B}_i)$ それぞれに対して定義される.
  $W_\mathcal{P} = \{W_{\mathcal{P}i}\}_{i=1}^{n_{ACC}}$, $W_{\mathcal{P}i} : S_\mathcal{P} \times S_\mathcal{P} \times Goal_i \to \mathbb{R}$

$$W_{\mathcal{P}i}(s_\mathcal{P}, s'_\mathcal{P}, goal_i) = \begin{cases} w_G(>0) & \text{if } (s_\mathcal{P}, s'_\mathcal{P}) \in goal_i \\ w_B(<0) & \text{if } (s_\mathcal{P}, s'_\mathcal{P}) \in \mathcal{B}_i \\ 0 & \text{othewise} \end{cases}$$

目標とは, $\mathcal{G}_i$ の要素の中でエージェントが目指す遷移を示す. つまり, 目標と同じ遷移をしたときしか正の報酬を得ないため, すべての状態から目標を目指すような経路を学習するということである. そして, 適切に目標を切り替えることで, 受理条件を満たす遷移が生起するようにエージェントを制御する. 強化学習の枠組みを図 1 に示す. エージェントを制御するメイン制御器 C_{main} は $\mathcal{P}$ 上で定常政策 π を学習する. 目標を提示するメタ制御器 C_{meta} が適切に C_{main} に目標を提示することで, C_{main} は目標と最適政策をもとにエージェントを制御する.

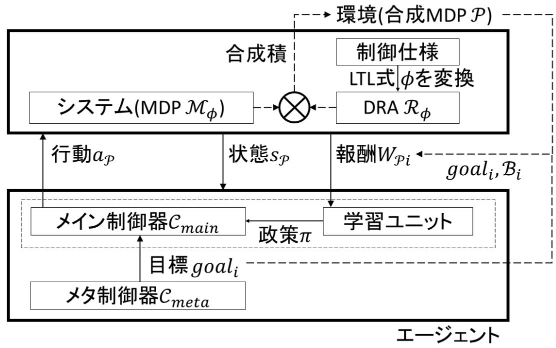

図 1　強化学習の枠組み

3　シミュレーション

図 2 で示される 5 × 5 の格子状の世界を考える. この世界の中を移動するエージェントに対する制御仕様を表す LTL 式は, $\phi = \mathbf{GF}A \land \mathbf{GF}B \land \mathbf{G}\neg C$ とする. $Goal = \{A$ が真になる遷移, B が真になる遷移 $\}$ であり, $\mathcal{B} = \{C$ が真になる遷移 $\}$ である. それぞれのエージェントは"右上", "左上", "右下", "左下"の 4 つの行動を持つ. 右上の行動は"右", "上", "その場にとどまる"の 3 つの遷移があり, それぞれの状態での遷移確率が定義されている. 以上のダイナミクスはそれぞれの行動に対して同様に定義される.

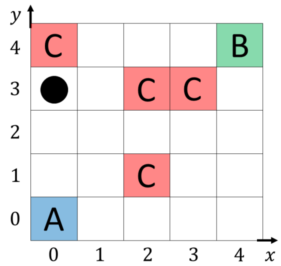

図 2　シミュレーションの対象システム

初期状態は図 2 の黒丸の位置である. C_{meta} は片方の目標にエージェントがたどり着いたらもう片方の目標に切り替えるというメタ制御を行った. C_{main} は *Q-learning* で学習した政策によるメイン制御を行い, 列を作成した. その列は LTL 式 ϕ を確率 1 で満たすことが確認できた. 直感的には, C が真になる遷移を避けながら, A が真になる遷移または B が真になる遷移がそれぞれ無限回生起したということである.

4　結論

LTL を確率 1 で満たす C_{main} を学習により獲得できた. LTL 式を用いることで制御仕様が正確に記述でき, 報酬も w_G, w_B, 0 の 3 種類を設定するだけでよいメリットがある.

問題点として, エージェント数や状態空間の増加により計算時間が爆発的に多くなってしまう点, C_{meta} の目標の切り替え方を既知のものとしている点が挙げられる.

今後の課題は, 複雑な例題に対して Deep Q-network や並列化の適用, C_{meta} の目標の切り替え方を学習するアルゴリズムの提案, 分散強化学習への拡張である.

参考文献

[1] D.Sadigh et al., "A Learning Based Approach to Control Synthesis of Markov Decision Processes for Linear Temporal Logic Specifications,"Technical Report UCB/EECS-2014-166, University of California, Berkeley, 2014

[2] E.Javier et al., "From LTL to Deterministic Automata–A Safraless Compositional Approach, "Formal Methods Syst. Des. vol. 49, pp. 219–271, 2016.

[3] Tejas D. Kulkarni et al., "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation," CoRR, abs/1604.06057, 2016