動的環境における学習ベースおよびルールベースの切り替え手法を用いた移動ロボットナビゲーション

○兵頭 侑樹（九州大学），松本 耕平（九州大学）, 倉爪 亮 (九州大学)

Mobile Robot Navigation with Switching Learning-Based and Rule-Based Method in Dynamic Environments

○ Yuki HYODO (Kyushu University), Kouhei MATSUMOTO (Kyushu University),

and Ryo KURAZUME (Kyushu University)

Abstract: Mobile robot navigation in crowded environments with pedestrians is an important challenge to realize service robots that assist people in their daily lives. Although deep reinforcement learning (DRL) has been widely studied as a method to adapt to such environments, there are many challenges in dealing with unexpected environments. In this study, we propose a method for dynamically switching between learning-based and rule-based algorithms using normalized flows in such environments.

1. 緒言

人々が生活する日常環境にロボットを安全に導入するには，歩行者がいる混雑した環境でもスムーズに移動できる移動ロボットナビゲーション手法の実現が不可欠である．近年では，深層強化学習を用いた移動ロボットナビゲーションについて盛んに研究が行われている．深層強化学習による手法は学習済みの環境に対して，ルールベースによる手法より良いパフォーマンスを発揮する．しかし，想定していない環境に直面した場合の対処は難しい．また，想定されるすべての状況に対するデータセットを用意したり，混雑した環境でロボットを動作させるシミュレーションを構築したりすることは容易ではない．

本研究ではこのような課題を解決するため，状況に応じて判断しながら学習ベースによる行動とルールベースによる行動を切り替える手法を提案する．具体的には，限られたデータセットをオフラインで強化学習させた学習ベースと衝突回避アルゴリズムからなるルールベースを，グラフ正規化フローを用いた尤度推定を基に切り替える手法を提案する．想定していない環境に直面し，学習ベースによる行動で衝突が発生すると判断した場合は，ルールベースに切り替えることで歩行者との衝突を回避し，学習ベースのみの場合よりも歩行者との衝突率が低く，ルールベースよりも早く目的地に到達することができる．

2. 問題設定

本研究では，X-Y 平面においてロボットが周囲の歩行者を回避ながら目的地に到達する環境を考える．環境の状態，行動，報酬は以下のように設定する．

- 状態：ロボットの状態は $\boldsymbol{s}^r = [|\boldsymbol{p}^g - \boldsymbol{p}^r|, \boldsymbol{v}^r, \theta]$ と表し，それぞれロボットから目的地までの距離，ロボットの速度，進行方向とする．対して，歩行者の状態は $\boldsymbol{s}^n = [{}^{rc}\boldsymbol{p}^n, {}^{rc}\boldsymbol{v}^n]$ であり，歩行者の位置と速度を表すが，それぞれの歩行者の位置，速度はロボット中心の座標系に変換される．観測はベクトル $(p_x^r, p_y^r, v_x^r, v_y^r, g_x, g_y, p_x^n, p_y^n)$ であり，(p_x^r, p_y^r) はロボットの位置，(v_x^r, v_y^r) はロボットの速度，(g_x, g_y) は目的地の位置，また i 番目の歩行者に対して，(p_x^n, p_y^n) は歩行者の位置，(v_x^n, v_y^n) は歩行者の速度を表している．
- 行動：本研究では，ホロノミックな全方位移動ロボットを想定し，2 次元空間におけるロボットの x 軸方向の入力速度 v_x と y 軸方向の入力速度 v_y からなる 2 次元ベクトル (v_x, v_y) を用いる
- 報酬：報酬には式 (1) を用いる．d_t は時刻 t におけるロボットと周囲の歩行者間の最小距離を表し，p_t^r は時刻 t におけるロボットの位置，p_g はロボットの目標位置を表す．

$$R_t = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } \boldsymbol{p}_t^r = \boldsymbol{p}_t^g \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

3. 提案手法

本研究の提案手法の概念図を図 1 に示す．ロボットと歩行者の情報を学習ベースおよびルールベースに入力すると，入力速度がそれぞれのモデルから出力される．これらの速度は switching administrator からの出力よって切り替えられる．ルールベースの手法には，

SY0010/23/0000 - 0275

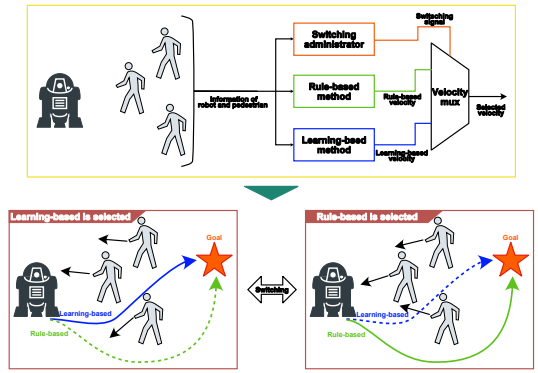

Fig.1 Conceptual diagram of proposed method.

エージェント間の衝突回避手法として広く用いられる ORCA [1] を利用する．学習ベースの手法にはオフライン強化学習に基づいた AWAC [2] によって学習された方策を利用する．switching administrator の基本構造として，グラフ正規化フローを利用する．

3.1 学習ベース手法

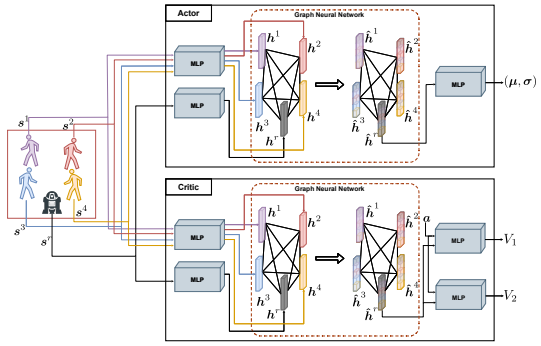

Fig.2 Architecture of actor-critic part of proposed method.

学習ベースモデルの構造を図 2 に示す．本研究の提案手法では Actor-Critic の方式をとり，Actor として確率的方策 π_θ を用いている．またこの構造は 2 つの min-Double-Q の Q 関数 Q_{ϕ^1}，Q_{ϕ^2} を持つ．学習された Actor は正規分布の平均値と分散を出力する．Actor と Critic はいずれもロボットと歩行者の関係を学習するため，GNN が含まれている．ここで GNN の構造は Embedded Gaussian とし，[3, 4] でも用いられている．このネットワークは clipping advantage [4] を用いて AWAC によって学習される．Actor は以下の式により学習が行われる．

$$L_{\text{actor}} = -\log \pi_\theta(\boldsymbol{s}, \boldsymbol{a}) \exp\left(\frac{1}{\lambda} W\right) \tag{2}$$

λ はハイパーパラメータである．重み W は以下の式によって計算される．

$$W = \max\left(0, \min_{i=1,2} Q_{\phi^i}(\boldsymbol{s}, \boldsymbol{a}) - \min_{i=1,2} Q_{\phi^i}(\boldsymbol{s}, \pi_\theta(\cdot \mid \boldsymbol{s}))\right). \tag{3}$$

学習ベースはオフライン強化学習に基づいて学習されるため，探索時に追加でデータを収集することはない．

3.2 Switching Administrator

3.2.1 アーキテクチャ

Switching Administrator の構造を図 3 に示す．Switching Administrator は正規化フローに基づいた構造をしており，歩行者の状態を MLP によりエンコードした後，正規化フローに入力され，尤度を計算される．正規化フローの構造としてはグラフ正規化フロー [5] を用いる．グラフ正規化フローは RealNVP [6] をベースとしており，affine coupling layer 内の非線形関数を，グラフニューラルネットワークで置き換えることで実現される．このグラフニューラルネットワークには Graph Attention Network (GATv2) [7] を利用する．また，バイナリマスクを利用したパーティションを採用しており，Switching Administrator 内のフローブロックの変換は以下のように表すことができる．

$$\begin{aligned}\bar{\boldsymbol{h}}^n &= \boldsymbol{b} \odot \boldsymbol{h}^n + (1 - \boldsymbol{b}) \\ &\quad \odot \left(\boldsymbol{h}^n \odot \exp(f_g^s(\boldsymbol{b} \odot \boldsymbol{h}^n)) + f_g^t(\boldsymbol{b} \odot \boldsymbol{h}^n)\right) \qquad (4)\\ \hat{\boldsymbol{h}}^n &= \tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n + (1 - \tilde{\boldsymbol{b}}) \\ &\quad \odot \left(\bar{\boldsymbol{h}}^n \odot \exp(f_g^s(\tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n)) + f_g^t(\tilde{\boldsymbol{b}} \odot \bar{\boldsymbol{h}}^n)\right), \qquad (5)\end{aligned}$$

$\boldsymbol{h}^n \in \mathbb{R}^d$ は MLP によりエンコードされた歩行者の特徴量である．また，$\boldsymbol{b}$ はバイナリマスクであり，$\tilde{\boldsymbol{b}} = 1 - \boldsymbol{b}$ である．f_g^s と f_g^t はそれぞれ GATv2 による関数である．f_g^s と f_g^t の計算は以下の式で表される．

$$f_g(\boldsymbol{h}^i) = \sum_{j \in \mathcal{N}_i} \alpha_{ij} \boldsymbol{W} \boldsymbol{h}^j, \tag{6}$$

$\boldsymbol{W} \in \mathbb{R}^{d \times d}$ は重みであり，$\mathcal{N}_i$ は歩行者の i 番目の隣接の集合である．アテンション係数 α_{ij} は以下の式で計算される．

$$e\left(\boldsymbol{h}^i, \boldsymbol{h}^j\right) = \boldsymbol{a}_w^\top \text{LeakyReLU}\left(\boldsymbol{W}_l \boldsymbol{h}^i + \boldsymbol{W}_r \boldsymbol{h}^j\right) \tag{7}$$

$$\alpha_{ij} = \text{softmax}_j\left(e\left(\boldsymbol{h}^i, \boldsymbol{h}^j\right)\right), \tag{8}$$

$\boldsymbol{a}_w \in \mathbb{R}^d$, $\boldsymbol{W}_l \in \mathbb{R}^{d \times d}$, and $\boldsymbol{W}_r \in \mathbb{R}^{d \times d}$ はいずれも学習時の重みである．

3.2.2 切り替え方法

学習ベースとルールベースの切り替えは switching administrator によって計算される尤度が，閾値 θ_{th} を下回るときに行われる．1 ステップごとに出力される行

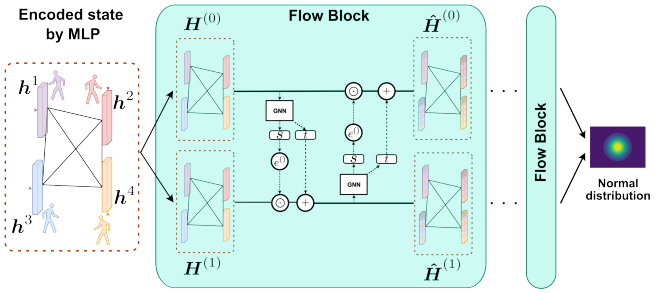

Fig.3 Architecture of switching administrator.

動は以下のように表される．

$$\boldsymbol{a}_t = \begin{cases} \boldsymbol{a}_t^r & \text{for } \tau(\boldsymbol{s}_t^h) > \theta_{th} \\ \boldsymbol{a}_t^l & \text{for } \tau(\boldsymbol{s}_t^h) \leq \theta_{th} \end{cases}, \tag{9}$$

$\boldsymbol{a}_t^r$ と $\boldsymbol{a}_t^l$ はそれぞれ時刻 t においてルールベースと学習ベースから生成される行動である．入力状態の尤度 $\tau(\boldsymbol{s}^h)$ は以下のように計算される．

$$\tau(\boldsymbol{s}^h) = -\log p_Z\left(f_{\text{sa}}(\boldsymbol{s}^h)\right). \tag{10}$$

この閾値は，学習後の switching administrator により算出した，学習データの尤度の平均値を利用する．

4. 実験

提案手法の効果を検証するため，シミュレーション環境にて評価実験を行った．

4.1 シミュレーション環境

シミュレーション環境として，先行研究 [3, 8] で採用されている CrowdNav 環境の square-crossing，circle-crossing シナリオを用いる．具体的には学習時のシナリオとして square-crossing を使用し，想定外の状況を含めた環境に対するテストとして circle-crossing を用いる．本環境において，歩行者は ORCA [1] に従って行動し，学習ベースと switching administrator は square-crossing にて 200 エピソード分のデータのみを学習させる．また circle-crossing にて 500 回の評価を行う．本実験におけるすべての学習，テストで歩行者の人数は 5 人である．

4.2 結果

提案手法による学習ベースとルールベースの切り替えによって移動ロボットナビゲーションの性能が改善するかをを数値比較により確認する．表 1 に square-crossing における学習ベースのみとルールベースのみの評価を示す．表 2 に circle-crossing シナリオでのルールベース手法と，学習ベース手法，提案手法の評価結果を示す．学習ベース手法の結果に関して，square-crossing シナリオでの結果と circle-crossing シナリオでの結果を比較すると，square-crossing シナリオでは，成功率が 90 % 以上あるのに対し circle-crossing シナリオでは 60 % 程度にまで低下している．この結果より，square-crossing でうまく動作するように学習したモデルであっても circle-crossing シナリオでは十分な性能が得られないことが分かる．次に circle-crossing シナリオでの結果に関して，学習ベースでは成功率が 63.8 % であるのに対して，提案手法で切り替えを行うことで成功率が 97.8 % になり，大幅に改善されることが分かる．またルールベースのみでは目的地に到達するまでにかかる時間は 10.0 秒程度かかるのに対し，学習ベースとルールベースを切り替えながら実行する提案手法では，8.8 秒程度で目的地に到達できている．

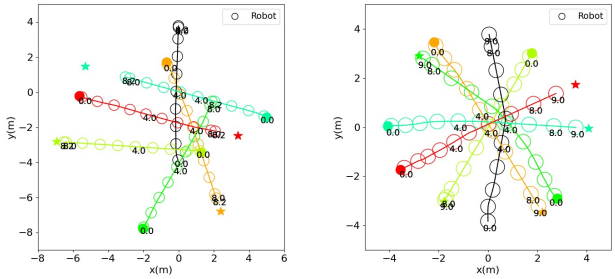

Fig.4 Trajectories of simulation environment.

Table 1 Numerical comparison in square-crossing

Method	Success [%]	Collision [%]	Exec. time [s]
Rule-based (ORCA)	100.0	0.0	8.49
Learning-based	91.2	7.8	8.29

Table 2 Numerical comparison in circle-crossing

Method	Success [%]	Collision [%]	Exec. time [s]
Rule-based (ORCA)	100.0	0.0	10.02
Learning-based	63.8	36.2	8.70
Random switching	77.8	22.2	8.98
Proposed	97.8	2.2	8.84

4.3 提案手法とランダムな切り替え手法の比較

次に学習ベースとルールベースがランダムに切り替えられる手法と提案手法との比較を行う．評価シナリオにおいて，提案手法を用いた場合に学習ベースからルールベースに切り替えられた割合は 60.4 % であったため，60.4 % の確率でランダムにルールベースに切り替えることで提案手法との差があるかを示す．提案手法とランダムに切り替える手法を比較した結果はそれぞれ，100 個のランダムなシード値でテストされた結果を平均したものである．ランダムに切り替える手法について，学習ベースのみよりも成功率の改善が見られたが，提案手法ほどの改善は見られなかった．さらに表 3 では学習ベースのシナリオで切り替えを加えた場合について，提案手法とランダムに切り替える手法

の比較を表している．それぞれの列は以下の内容を示している．

- S-S : 学習ベースのみでは成功しており，切り替えを加えた場合も成功しているエピソード数
- S-F : 学習ベースのみでは成功しており，切り替えを加えた場合は失敗しているエピソード数
- F-S : 学習ベースのみでは失敗しており，切り替えを加えた場合は成功しているエピソード数
- F-F : 学習ベースのみでは失敗しており，切り替えを加えた場合も失敗しているエピソード数

Table 3 Numerical comparison of number of change in results

Method	S-S	S-F	F-S	F-F
Random switching	260	59	129	52
Proposed	318	1	171	10

表 3 で示す通り S-F について，ランダムな切り替え手法は全エピソードのうち 31% にあたる 59 エピソードが S-F に該当していたのに対して，提案手法においては S-F に含まれるエピソード数は 1 つのみであり，全エピソードの 0.6 % に過ぎなかった．このことからランダムに切り替える場合は，学習ベースのみで成功していたシナリオでも失敗するようになってしまう場合が多くあることが分かる．このことから提案手法がランダムに切り替える手法よりも有効であることが分かる．

5. まとめと今後の展望

本研究では，歩行者が行き交う動的な環境での移動ロボットナビゲーションにおいて，正規化フローを利用して，ルールベース手法と学習ベース手法を切り替えることで，学習時に想定していない環境にも対応可能な手法を提案した．実験結果より，実際に学習時に利用していないシナリオにおいて，学習ベースのみでは性能が大幅に低下するところ，提案手法を利用することで，高い成功率を達成でき，ルールベースのみの場合よりも素早く目的地に到達できることが示された．また，ランダムに切り替える手法との比較実験により，提案手法の方が高い成功率を達成できることが分かった．また，ランダム切り替えの場合は，学習ベースのみでの場合に成功していたエピソードが失敗に転じている場合が多く見られるが，提案手法では切り替えにより失敗に転じる可能性は低いことが分かった．

今後の展望としては，実環境での実験を行うことが考えられる．また，今回の実装は switching administrator は学習したデータセットに対する尤度を評価しているため，厳密には直面している状況を方策が対応できるのかそうでないのかを直接評価できるものではない．今後はこのように方策の性能を考慮して切り替えを行うような手法の開発に取り組んでいく．

謝辞

本研究の一部は，JSPS 科研費 JP20H00230 の助成を受けたものである．

参考文献

[1] Jur Van Den Berg et al.: “Reciprocal n-Body Collision Avoidance”. *Proceedings of the International Symposium of Robotic Research*. 2011, pp. 3–19. ISBN: 9783642194566. DOI: 10.1007/978-3-642-19457-3_1.

[2] Ashvin Nair et al.: *AWAC: Accelerating Online Reinforcement Learning with Offline Datasets*. (2021). URL: https://openreview.net/forum?id=OJiM1R3jAtZ.

[3] Changan Chen et al.: “Relational Graph Learning for Crowd Navigation”. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2020, pp. 10007–10013. arXiv: 1909.13165.

[4] Xueyou Zhang et al.: “Relational Navigation Learning in Continuous Action Space among Crowds”. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2021, pp. 3175–3181.

[5] Jenny Liu et al.: “Graph Normalizing Flows”. *Advances in Neural Information Processing Systems (NeurIPS)*. 2019, pp. 13556–13566.

[6] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio: “Density estimation using Real NVP”. *Proceedings of the International Conference on Learning Representations (ICLR)*. Apr. 2017.

[7] Shaked Brody, Uri Alon, and Eran Yahav: “How Attentive are Graph Attention Networks?” *Proceedings of the International Conference on Learning Representations (ICLR)*. 2022.

[8] Changan Chen et al.: “Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning”. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*. 2019, pp. 6015–6022. ISBN: 9781538660263. DOI: 10.1109/ICRA.2019.8794134. arXiv: 1809.08835.