グラフ畳み込み構造を持つ予測状態表現を用いた深層強化学習による移動ロボットナビゲーション

○松本耕平（九州大学）　河村晃宏（九州大学）　安琪（東京大学）　倉爪亮（九州大学）

1. はじめに

生活環境で活動するサービスロボットの実現には，歩行者などが存在する動的な環境での自律移動が不可欠である．歩行者の行動は様々な影響によって変化する可能性があり，ロボットの行動にも影響を受ける可能性がある．本研究で提案する自律移動ロボットナビゲーションのための深層強化学習手法は，エージェントの行動後の環境の変化を予測できる予測状態表現（PSR）に基づいて，ロボットの行動の影響による歩行者の行動の変化に対応する．さらに，従来の PSR モデルでは考慮できない，歩行者同士の影響による歩行者の行動の変化に対応するために，グラフ畳み込み構造を用いた新しい PSR モデルを提案し．従来手法と比較して有効であることをシミュレーションによって示す．さらに，歩行者数の変化に対応するために，PSR の状態を統合する 2 つの手法を提案し，この 2 つの手法の性能を比較する．

2. Predictive State Representation

予測状態表現 (Predictive State Representation : PSR)[1] は，考え得る全てのパターンの観測・行動から予想される結果が全て分かっていれば，動的システムを完全に把握できているという考え方に基づいている．

基本的な PSR モデルは離散的な観測・行動からなるシステムのみに適用できる手法であるが，これまでに，PSR を連続的なシステムに対応できるようにした手法が提案されている [2, 3]．本稿では，これらの手法を総称して Recurrent PSR (RPSR) と呼ぶ．

RPSR の状態の更新は 2 つの手順で行われる．

- Extension : 状態 q_t に線形写像 W_{ext} を適用し，拡張状態 p_t を得る．拡張状態 p_t は，拡張された行動 $a_{t:t+k}$ によって条件付けられた拡張された観測 $o_{t:t+k}$ の条件付き分布である．また，W_{ext} は学習によって最適化されるパラメータである．

$$e_t = W_{\text{ext}} q_t \tag{1}$$

- Conditioning : 時刻 t における行動 a_t と観測 o_t に，既知の条件付け関数 f_{cond} により，以下のように状態が更新される．

$$q_{t+1} = f_{\text{cond}}\left(e_t, a_t, o_t\right) \tag{2}$$

q_t と e_t は条件付き確率であり，f_{cond} はベイズ則を適用する．連続的なシステムに応用するために，これらに対して，分布のヒルベルト空間埋め込み [2] とカーネルベイズ則 [4] を用いる．

本研究では，RPSR のモデルとして RFF-PSR [3] を用いる．RFF-PSR では，観測と行動のデータに RBF カーネルによる写像を適用後，ランダムフーリエ特徴 [5] を抽出し，ランダム主成分分析 [6] を用いて次元削減したものを，それぞれ観測と行動データの特徴量として用いる．この特徴量を抽出する関数を ϕ で表す．これを用いて，観測予測関数 f_{pred} によって以下のように時刻 t における観測が予測される．

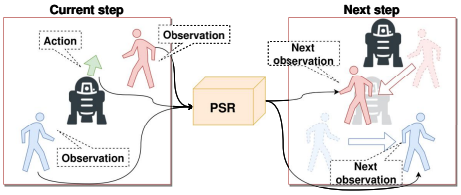

図 1 歩行者による動的環境における移動ロボットナビゲーションに対する PSR のモデリングの概念図

$$\begin{aligned}\hat{o} &= f_{\text{pred}}\left(q_t, \phi\left(a_t\right)\right) \\ &= W_{\text{pred}}\left(q_t \otimes \phi\left(a_t\right)\right)\end{aligned} \tag{3}$$

ここで，W_{pred} は学習で最適化される線形写像であり，$\otimes$ はクロネッカー積を表す．

3. 提案手法

3.1 歩行者による動的環境における移動ロボットナビゲーションへの PSR の適用

本研究では，PSR に基づく深層強化学習手法を，複数の歩行者が存在する動的環境における移動ロボットナビゲーションに適用した．移動ロボットのナビゲーションにおける PSR の要素は，以下のように構成することができる．

- 観測：歩行者とロボットの位置と速度データを観測として扱う．各歩行者，ロボットの観測はベクトル $(p_x^i, p_y^i, v_x^i, v_y^i)$ であり，i 番目の歩行者またはロボットにおいて，(p_x^i, p_y^i) は位置，(v_x^i, v_y^i) は速度を表す．
- 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．

これによって構成される PSR は，ロボットの指令速度を入力することで，歩行者の位置と次のステップの速度を予測することができるモデルとなる．PSR のモデルの概念図を図 1 に示す．

3.2 提案手法のアーキテクチャ

提案手法のアーキテクチャを図 2 に示す．提案手法は Feature extractor, State updater, Observation predictor, State integrator, Value estimator により構成される．

- Feature extractor：入力情報から特徴を抽出する．

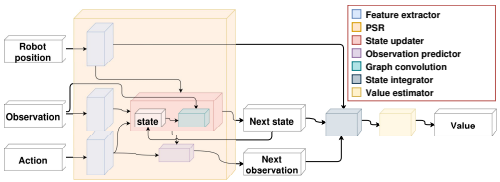

図 2　提案手法のアーキテクチャ

- State updater：観測と行動から抽出された特徴量を用いて，状態を更新する．
- Observation predictor：行動から抽出された状態特徴量を用いて，次の観測を予測する．
- State integrator：各歩行者に対応する PSR の状態を統合する．
- Value estimator：統合された状態とロボットの特徴から，価値を推定する．

3.3　Graph convolutional RPSR

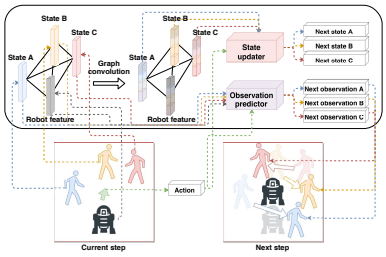

図 3　GC-RPSR の状態更新の概念図

RPSR モデルは，エージェントの行動が環境に与える影響を考慮することができ，これにより，ロボットの行動が各歩行者に与える影響を考慮することができる．しかし，エージェントの行動に起因しない影響は考慮されていないため，歩行者同士の影響を考慮することはできない．この問題に対処するために，Graph Convolutional RPSR（GC-RPSR）を提案する．

本アーキテクチャは，RPSR の状態更新時において，歩行者間の相互作用を表現するためのグラフ畳み込み構造を有している．この GC-RPSR の状態更新処理の概念図を図 3 に示す．グラフ畳み込みに用いる隣接行列 A_t のカーネルは，式 (4) に示すように，各位置の差分の L_2 ノルムの逆数に基づく [7].

$$a_t^{ij} = \begin{cases} 1/\left\|p_t^i - p_t^j\right\|_2 & \text{if } \left\|p_t^i - p_t^j\right\|_2 \neq 0 \\ 0 & \text{Otherwise} \end{cases} \tag{4}$$

さらに，式 (5) に示す正規化処理を行い $\tilde{A}_t$ を実際の計算に用いる．

$$\tilde{A}_t = \Lambda_t^{-\frac{1}{2}} \hat{A}_t \Lambda_t^{-\frac{1}{2}} \tag{5}$$

ここで，$\hat{A}_t = A_t + I$ であり，Λ_t は次数行列を表す．グラフ畳み込みと式 (1), (2) を用いることで，GC-RPSR の状態更新は以下のように表される．

$$\tilde{q}_t = f_{gc}\left(q_t, \tilde{A}_t\right) \tag{6}$$

$$e_t = W_{\text{ext}} \tilde{q}_t \tag{7}$$

$$q_{t+1} = f_{\text{cnd}}\left(e_t, a_t, o_t\right) \tag{8}$$

ここで，f_{gc} はグラフ畳み込み関数を表す．

3.4　状態統合手法

各歩行者に対応する状態から，歩行者数の変化に対して，スケーラブルに価値評価を行うための統合処理が必要である．これに対して．グラフの畳み込みを用いる手法と占有地図に基づく手法の 2 種類を提案する．

3.4.1　グラフ畳み込みによる統合

各歩行者に対応する PSR の状態を統合するために，グラフ畳み込みを用いる．Value estimator はグラフ畳み込みが適用されたロボット特徴量のみを受け取る．隣接行列に用いるカーネルは，式 (4) で表される GC-RPSR の状態更新と同じである．この手法の概念図を図 4 に示す．

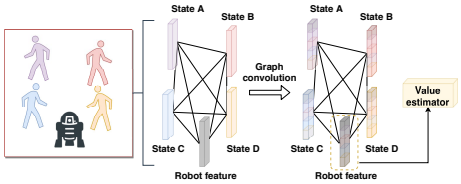

図 4　グラフ畳み込みによる状態統合の概念図

3.4.2　占有地図による統合

各歩行者に対応する PSR の状態を統合するために，占有地図を使用する．歩行者の状態は，歩行者の位置に応じて対応するセルに格納される．同じセルに複数の歩行者の状態を格納する必要がある場合は平均値を格納する．この手法の概念図を図 5 に示す．

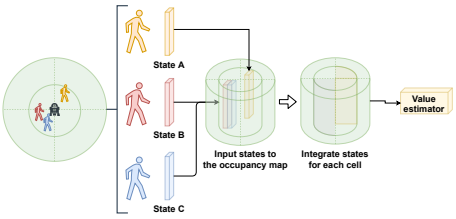

図 5　占有地図による状態統合の概念図

3.5　行動生成

行動は，Value estimator f_v，PSR モデル f_p の State updater f_p^q，Observation predictor f_p^o を用いて，得られる報酬が最大になるように，行動空間 $\mathcal{A}$ から式 (9) に従って選択される．ここで，γ は割引率，s_t は時刻 t における統合された状態，p_t は時刻 t におけるロボットの位置を表し，$\hat{o}_t = f_p^o\left(q_t, a_t\right)$ である．

$$a_t \leftarrow \text{argmax}_{a_t \in \mathcal{A}} R\left(\hat{o}_t\right) + \gamma^{\Delta t} f_v\left(s_t, p_t\right) \tag{9}$$

さらに，$R(o_t)$ は時刻 t における報酬関数であり，式 (10) で表される．d_t はロボットと歩行者間の最小距離を表し，p_g はロボットの目標位置を示す．

$$R(o_t) = \begin{cases} -0.25 & \text{if } d_t < 0 \\ -0.1 + d_t/2 & \text{else if } d_t < 0.2 \\ 1 & \text{else if } p_t = p_g \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

さらに，d-step planning[8] を利用し，式 (11) に従っ

て，d ステップ先を考慮して行動を生成する．

$$f_v^d(s_t, p_t) = \begin{cases} f_v(s_t, p_t) & \text{if } d = 1 \\ \frac{1}{d} f_v^1(s_t, p_t) + \frac{d-1}{d} \max_{a_t}(& \\ R(\hat{o}_{t+1}) + \gamma f_v^{d-1}(\hat{s}_{t+1}, \hat{p}_{t+1})) & \text{otherwise} \end{cases} \quad (11)$$

3.6 学習手法

学習は 2 つのステップで行われる．最初のステップでは，PSR パラメータを事前学習する．この事前学習は，ORCA [9] に従った探索方針でデータを収集した後，two-stage regression [10] によって行われる．その後，収集したデータを用いてアルゴリズム 1 に従い，強化学習によって Value estimator の学習させ，教師あり学習で PSR モデルを学習させる．

Algorithm 1: 提案手法の学習

PSR f_p と Value estimator f_v を two-stage regression と模倣学習によって事前学習
Value estimator のターゲットネットワーク $\hat{f}_v$ を初期化
for $i = 1$ **to** E **do**
  探索方策に従い a_t を選択し，報酬 r_t，観測 o_t，ロボットの位置 p_t を取得
  エピソード終了後，軌跡 (o_t, a_t, r_t, p_t) をバッファ $\mathcal{B}$ に保存
  バッファ $\mathcal{B}$ から M セットの軌跡をサンプルする
  for $j = 1$ **to** M **do**
    以下の軌跡を取得する
      観測 $\mathbf{o}^j = \left\{o_1^j, o_2^j, \ldots, o_T^j\right\}$,
      ロボットの位置 $\mathbf{p}^j = \left\{p_1^j, p_2^j, \ldots, p_T^j\right\}$,
      行動 $\mathbf{a}^j = \left\{a_1^j, a_2^j, \ldots, a_T^j\right\}$
    以下の軌跡を計算する
      状態 $\mathbf{q}^j = \left\{q_1^j, q_2^j, \ldots, q_T^j\right\}$,
      価値の目標値 $\mathbf{y}^j = \left\{y_1^j, y_2^j, \ldots, y_T^j\right\}$,
      統合された状態 $\mathbf{s}^j = \left\{s_1^j, s_2^j, \ldots, s_T^j\right\}$
  end
  $L_{\text{pred}} = \text{MSE}(f_p^o(\mathbf{q}, \mathbf{a}), \mathbf{o})$ を最小化するように f_p を更新
  $L_{\text{value}} = \text{MSE}(f_v(\mathbf{s}, \mathbf{p}), \mathbf{y})$ を最小化するように f_v を更新
  if i mod d **then**
    $\hat{f}_v \leftarrow f_v$ によって $\hat{f}_v$ を更新
  end
end

4. シミュレーション実験

4.1 シミュレーション環境

本研究では，CrowdNav 環境での circle crossing シナリオを利用する [11, 12]．このシナリオでは，ロボットは初期位置 $(x, y) = (0, -4)$ からゴール地点 $(x, y) = (0, 4)$ を目指して進む．歩行者は ORCA 従って行動し，初期化時に半径 4m の円上にランダムに配置される．ロボットと歩行者は互いに影響し合い，位置や速度によって挙動が変化する．評価は 500 パターンのテストケースを用いて行う．

4.2 GC-RPSR と RPSR の比較

GC-RPSR と RPSR を比較することで，その提案手法の有効性を確認した．成功率，衝突率，実行時間，平均収益を比較する．この実験では State integrator には単純な連結を使用した．結果を表 1 に示す．

表 1 RPSR と GC-RPSR の比較

手法	成功率 [%]	衝突率 [%]	実行時間 [s]	平均収益
RPSR	57.8	33.8	8.84 ± 1.32	0.258
GC-RPSR	64.0	11.0	8.73 ± 1.31	0.356

この表から，GC-RPSR はすべての評価項目で RPSR を上回っていることがわかる．したがって，提案する PSR モデルは基本的なモデルである RPSR と比較して，移動ロボットナビゲーションタスクにおいて優れていると言える．

4.3 ベースラインと状態統合手法を組み合わせた提案手法の比較

図 6 にグラフ畳み込みを用いた状態統合を行った提案モデル（GGC-RPSR）の結果の軌跡と，占有地図を用いた状態統合を行った提案モデル（OGC-RPSR）の結果の軌跡を示す．各図において，黒い軌跡はロボット，その他の色の軌跡は歩行者，数字は時間ステップを表している．どちらの手法も，歩行者を避けながらロボットをゴールまで誘導できることが分かる．しかし，GGC-RPSR の方がより短い経路を生成できるため，効率が良いと言える．

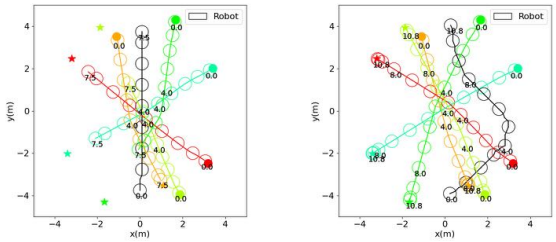

図 6 GGC-RPSR により生成される軌跡（左）と OGC-RPSR により生成される軌跡（右）

また，関連研究で提案された手法 [12] をベースラインとして提案モデルとの比較を行う．提案する 2 つの手法とベースライン手法の数値比較を表 2 に示す．GGC-RPSR はすべての評価項目で他の手法を上回っている．この結果は，GC-RPSR と，グラフ畳み込みを用いた状態統合を用いた提案手法の有効性を示している．これは，PSR の構造がより最適な経路計画を実現するために有効であったためであると考えられる．また，グラフ畳み込みを用いた状態統合手法の方が最適化性能が高いため，学習時とテスト時の歩行者数が同じ本実験では，GGC-RPSR の方が OGC-RPSR より性能が高かったと推察される．

表 2 提案手法とベースラインの比較

手法	成功率 [%]	衝突率 [%]	実行時間 [s]	平均収益
RGL [12]	92.0	7.0	9.09 ± 1.31	0.560
GGC-RPSR	94.6	5.4	7.79 ± 0.10	0.587
OGC-RPSR	91.4	6.8	11.54 ± 1.03	0.512

4.4 歩行者数が学習時とテスト時で異なる場合の比較

歩行者の数が学習時とテスト時で異なる状況で，2 つの提案手法を比較した．両モデルとも歩行者数が 5 人の環境で学習し，歩行者数が 1~10 人の環境でテストを行った．各モデルの歩行者 5 人の場合での結果との成功率の差を図 7 に示す．GGC-RPSR の成功率の最大低下率は 43.4%であり，これに対し OGC-RPSR の最大低下率は 7.4%である．この結果から，OGC-RPSR は学習時とテスト時で歩行者数を変化させた場合により安定していることがわかる．

両者を比較すると，GGC-RPSR は歩行者数によらず，各歩行者に対応する状態を 1 つの特徴量に統合し

ている．そのため，未学習の歩行者数ではうまく機能しない可能性があると推察される．一方，OGC-RPSR は，各セルの歩行者に対応する状態を統合しているため，歩行者の数に関わらず，各セルに統合される状態の数が一定になることが多いのではないかと考えらる．そのため，OGC-RPSR の方が安定した性能を発揮したと考察する．

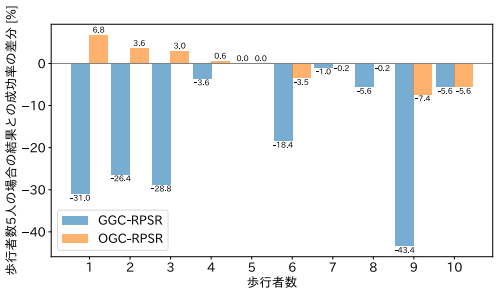

図 7　歩行者が 5 人の場合での結果との成功率の差の比較

4.5　歩行者数を変えながらの学習

4.4 節で不安定な結果であった GGC-RPSR において，歩行者数を 1~10 人で変化させながら学習を行い，歩行者数 5 人で学習した場合と成功率を比較した．結果を図 8 に示す．この結果より歩行者数を 1~10 人で変化させながら学習を行った場合，成功率の最大値が下がるものの，各歩行者数においての成功率の差は小さくなっていると言える．各歩行者数での成功率の平均及び標準偏差を見ると，歩行者数 5 人で学習した場合 78.2 ± 15.3%であるが，歩行者数を変化させながら学習を行った場合 85.7±7.7%であり，より安定した結果になっていることが分かる．

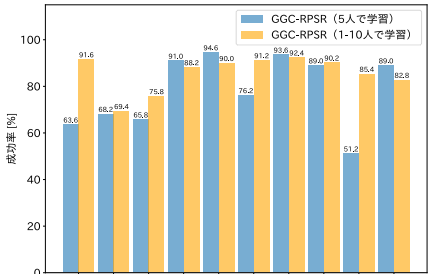

図 8 歩行者数を変化させながら学習を行った場合と歩行者数 5 人で学習した場合の成功率の比較

5.　結論

本研究では，歩行者による動的環境における移動ロボットナビゲーション課題において，PSR に基づく深層強化学習法を用いた．グラフ畳み込みを利用することで歩行者間の相互作用を考慮した，新しい PSR の構造を提案した．また，学習時とテスト時の歩行者数の変化に対応するため，各歩行者に対応した PSR の状態を統合する方法を提案し，シミュレーション実験により提案モデルの有効性を確認した．また，訓練時とテスト時で歩行者数が異なるシナリオでは，占有地図を用いた状態統合を行う OGC-RPSR がより安定していることを確認した．さらに，GGC-RPSR については，歩行者数を変化させながらの学習によって，安定性が増すことを確認した．

今後は，本研究で提案した 2 種類の状態統合手法のより詳細な解析を行い，本研究の考察が妥当であるかを確認する．加えて，占有地図ベースの手法についても，4.5 節と同様の条件で学習を行い，今回の結果との比較を行う．さらに，本研究の実験は全てシミュレーション環境で行ったが，今後，実際のロボットへの適用も進める予定である．

参　考　文　献

[1] S. Singh, M. James, and M. Rudary, “Predictive State Representations: A New Theory for Modeling Dynamical Systems,” in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 512–519, 2004.

[2] B. Boots, A. Gretton, and G. J. Gordon, “Hilbert space embeddings of predictive state representations,” in *Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 92–101, 2013.

[3] A. Hefny, C. Downey, and G. Gordon, “An efficient, expressive and local minima-free method for learning controlled dynamical systems,” in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pp. 3191–3198, 2018.

[4] K. Fukumizu, L. Song, and A. Gretton, “Kernel Bayes’ rule: Bayesian inference with positive definite kernels,” *Journal of Machine Learning Research*, vol. 14, pp. 3753–3783, 2013.

[5] “Random features for large-scale kernel machines,” in *Advances in Neural Information Processing Systems*, 2008.

[6] N. Halko, P. G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[7] A. Mohamed, K. Qian, M. Elhoseiny, and C. Claudel, “Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction,” in *Proceedings of the IEEE/CVF Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14412–14420, 2020.

[8] J. Oh, S. Singh, and H. Lee, “Value prediction network,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6119–6129, 2017.

[9] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in *Proceedings of the International Symposium of Robotic Research*, pp. 3–19, 2011.

[10] A. Hefny, C. Downey, and G. J. Gordon, “Supervised learning for dynamical system learning,” in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1963–1971, 2015.

[11] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, “Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6015–6022, 2019.

[12] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, “Relational graph learning for crowd navigation,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10007–10013, 2020.