1L2-R-7-10

行動の属性に基づく模倣行動による学習

Learning through imitation based on the importance of the features of actions

坂戸達陽*1　尾関基行*1　岡夏樹*1
Tatsuya SAKATO　Motoyuki OZEKI　Natsuki OKA

*1京都工芸繊維大学 大学院工芸科学研究科
Graduate School of Science and Technology, Kyoto Institute of Technology

Learning is essential for an autonomous agent to adapt to an environment. One method that can be used is learning through trial and error. However, it is impractical because of the long learning time required. Therefore, some guidelines are necessary to expedite the learning process. Imitation can be used as the guideline. We propose a computational model of imitation and autonomous behavior, and expect that an agent can reduce its learning time through imitation. The actions are represented by a set of features, and the similarity between actions is indicative of the importance of each feature. The proposed model is evaluated using a simulator. The experimental results indicate that the model can adapt to the environment faster than a baseline model that learns only through trial and error, and that the model can shorten the learning time further if the importance of each feature can be adjusted by learning.

1. はじめに

自律的なロボットが環境に適応するためには学習が不可欠である．ロボットが環境に適応するための学習手法の一つとして，強化学習 [Sutton 98] などの試行錯誤による学習が挙げられる．しかし，試行錯誤による学習のみで環境に適応するには多くの時間が必要であり実用的ではない．ゆえに複雑な環境では，学習時間を短縮するために何らかの指針が必要である．適切な指針はロボットの学習時間を短縮させることができるはずである．ただし，指針は最終的な学習結果を変えるべきではない．Asmuth ら [Asmuth 08] や Wiewiora ら [Wiewiora 03] は，最終的な学習結果を変えずに指針を与える手法を提案している．これらの手法は静的な関数をあらかじめ定義して指針として用いている．しかし，複雑な環境においてそのような関数をあらかじめ定義することは難しい．ゆえに複雑な環境では，あらかじめ定義する必要のない指針を用いる必要がある．

環境にすでに適応しているエージェントは，その環境に適した行動をとる．ゆえに環境にすでに適応しているエージェントの行動を模倣することは，環境に適応していないエージェントの学習時間を短縮すると思われるので，このようなエージェントの行動は学習の指針として働きうる．そこで，本研究では強化学習による学習とともに他のエージェントの行動を模倣するモデルを提案する．提案モデルでは試行錯誤による学習の指針を他のエージェントの行動から動的に生成する．ゆえに提案モデルでは試行錯誤による学習の指針を事前に準備しておく必要がない．

提案モデルでは，行動を種類，位置，行動の対象などの属性の組によって表現する．すべての属性は等しく重要なわけではないので，エージェントは各属性の重要度に基づいて模倣を行うべきである．エージェントは各属性の重要度に基づいた行動間の一致度を計算し，それを用いて他のエージェントの行動を模倣しようとする．この方法で模倣を行うために，エージェントはどの属性が重要かを学習する必要がある．

提案モデルでは，エージェントが他のエージェントが存在する環境で試行錯誤による学習を行うことを想定する．エージェントは他のエージェントの行動を試行錯誤による学習の指針として用いる．エージェントは模倣行動を，学習によって順応した各属性の重要度に基づき選択する．本研究では他のエージェントによって行われた行動を模倣することで，エージェントが学習時間を短縮させることを期待する．

以下，本論文は次のように構成される．第 2. 節では提案する模倣と自律行動の計算モデルについて述べる．第 3. 節では提案モデルの主要な構成要素である模倣モジュールについて述べる．第 4. 節では提案モデルの評価実験について述べる．第 5. 節では評価実験の実験結果について，第 6. 節では実験結果の考察を述べる．最後に，第 7. 節で結論を述べる．

連絡先: 坂戸達陽，sakato@ii.is.kit.ac.jp

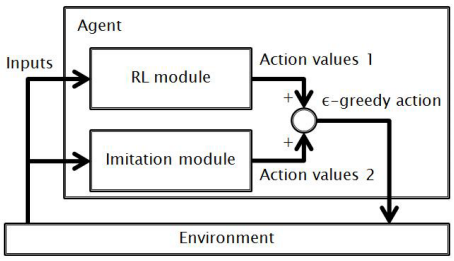

図 1: Configuration of the proposed model. The small circle represents the action selection module.

2. 提案モデル

提案モデルを図 1 に示す．

提案モデルにおいて，エージェントは強化学習モジュール，模倣モジュール，行動選択モジュールから構成される．エージェントは与えられた環境で試行錯誤によって学習する．試行錯誤による学習は強化学習モジュールによって行われる．強化学習モジュールでは Q 学習を用いる．エージェントは環境に他のエージェントが存在する場合，そのエージェントの行動を模倣しようとする．模倣に関する処理は模倣モジュールによって行われる．強化学習モジュールと模倣モジュールはそれぞれ

各行動の行動価値を計算し，行動選択モジュールへ出力する．行動選択モジュールは出力された行動価値を各行動ごとに足し合わせ，足し合わされた行動価値に基づき，行動を ϵ-greedy に決定する．学習の初期段階では，エージェントはたとえそれが負の報酬が期待される行動であったとしても，他のエージェントの行動を模倣しようとすると思われる．学習の初期段階では強化学習モジュールが未学習で，期待される報酬がわからないからである．学習が進むにつれて強化学習モジュールで行動価値が学習され，負の報酬が期待される行動を行わなくなることを期待する．

3. 模倣モジュール

模倣モジュールは他のエージェントが行った行動を観測し，その行動のもつ属性とモジュールのもつ各属性の重みに基づき各行動の行動価値を計算し，行動選択モジュールへ出力する．属性の重み w_i は属性 i のその環境における重要度を反映している．模倣モジュールが出力する各行動の行動価値は観測された行動とエージェントがとることのできる各行動の属性の一致度を表す．しかしながら，もし，2 つの行動の根本的な属性が異なっているならば，それらの行動はもはや同じ行動とは言えないので，すべての属性が等しく重要であるわけではない．ゆえに，模倣モジュールでは各属性に重みを与え，行動間の一致度 $s(a, a')$ を次のように定義する．

$$s(a, a') = \sum_{i=1}^{n} w_i \delta_{a_i a'_i} \tag{1}$$

ここで w_i は属性 i の重みであり，次の式を満たす．

$$\sum_{i=1}^{n} w_i = 1 \tag{2}$$

a と a' は行動，n は行動のもつ属性の数，a_i と a'_i はそれぞれ行動 a と行動 a' の属性 i である．また，δ_{ij} はクロネッカ・デルタである．$s(a, a')$ は $0 \leq s(a, a') \leq 1$ を満たすことに注意されたい．

行動 a' が観測されたとき，模倣モジュールは行動価値 $ActionValue(a)$ を次のように決定する．

$$ActionValue(a) = s(a, a') \tag{3}$$

$0 \leq s(a, a') \leq 1$ であるので，$0 \leq ActionValue(a) \leq 1$ である．行動が観測されないとき，$ActionValue(a)$ は 0 とする．

重要な属性は環境が異なれば異なるので，各属性の重要度は環境ごとに学習する必要がある．エージェントが行動 a_t を行ったとき，報酬 r_{t+1} を獲得したとする．このときに認識されていた行動を a' とすると，$ActionValue(a_t)$ $(= s(a_t, a'_t))$ は模倣モジュールが時刻 t での行動選択にどの程度関わっていたかを表す．

エージェントが行動 a'_t の模倣として，行動 a_t を行ったとき，次の 3 つの場合が考えられる．

1. エージェントは属性 i が一致していたので報酬 r_{t+1} 獲得した．
2. エージェントは属性 i が一致していなかったので報酬 r_{t+1} 獲得した．
3. 属性 i は報酬 r_{t+1} の獲得には無関係である．

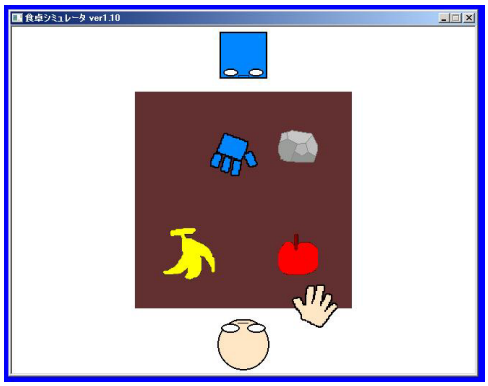

図 2: Dining table simulator.

より多くの報酬を獲得するためには，報酬 r_{t+1} が正のとき，場合 1 にあてはまる属性の重みを大きく，場合 2 にあてはまる属性の重みを小さくすればよい．逆に，報酬 r_{t+1} が負のときには，場合 1 にあてはまる属性の重みを小さく，場合 2 にあてはまる属性の重みを大きくすればよい．提案モデルでは，各属性の重み w_i を次のように更新する．

$$w'_i = w_i + \alpha s(a, a') r_{t+1} w_i \delta_{a_i a'_i} \tag{4}$$

$$w_i \leftarrow \frac{w'_i}{\sum_{i=1}^{n} w'_i} \tag{5}$$

ここで，$\alpha (0 \leq \alpha \leq 1)$ は学習率である．

4. 評価実験

4.1 食卓シミュレータ

提案モデルの性能は，図 2 のような食卓シミュレータを用いて評価する．シミュレータには 2 体のエージェントの頭部と右手が実装されている．下側のエージェントは人間が操作し，上側のエージェントは提案モデルを用いて操作する．テーブル上にはりんご，バナナ，石など，いくつかの物体が存在する．各エージェントは“テーブル上の物体をつかむ”，“テーブル上に物体を置く”，“物体を捨てる”，“食べ物を食べる”といった行動が可能である．

4.2 実験環境

実験環境は次のとおりである．

- 2 体のエージェントの行動は非同期に行われる．
- 物体は図 3 中の 0 から 3 の 4 か所に置くことができる．
- 時間が経過するとランダムな物体がテーブル上のランダムな位置に出現する．
- 自律エージェントには食べ物を食べたときに 1，行動に失敗したときに -0.1 の報酬が与えられる．食べ物を食べる以外の行動が成功した場合の報酬は 0 である．

物体を置いたりつかんだりする位置は，図 3 のように離散化されている．物体は，“りんご”，“バナナ”，“石”の 3 種類である．エージェントが可能な行動は 4.3.1 節に示すが，エージェントが失敗する行動を以下に示す．

- 何もない場所に対する“物体をつかむ”行動

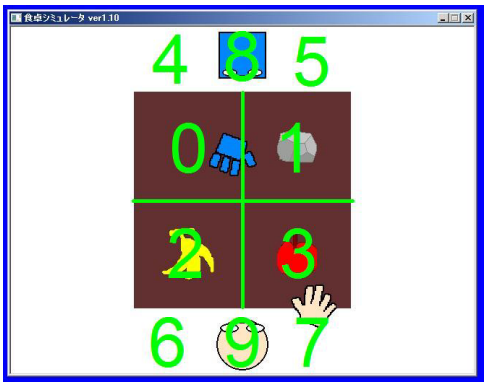

図 3: Discrete locations in the environment.

- すでに物体を持っているときの “ 物体をつかむ ” 行動
- 何も持っていないときの “ 物体を横に捨てる ” 行動
- 食べ物を持っていないときの “ 食べ物を食べる ” 行動

4.3 エージェントの仕様

4.3.1 エージェントが可能な行動

エージェントは次の 12 種類の行動が可能である.

- テーブル上の物体をつかむ行動 4 種類
- テーブル上に物体を置く行動 4 種類
- 物体を横に捨てる行動 2 種類
- 食べ物を食べる行動 1 種類
- 何もしない

エージェントは 1 つの行動を 500ms で行う. ϵ-greedy 法におけるランダムな行動を行う確率 ϵ は 0.1 と定める.

4.3.2 強化学習モジュールの仕様

強化学習モジュールは, “ テーブル上にある物体の種類 ” と “ エージェントが持っている物体の種類 ” に基づいて各状態を定める. 強化学習モジュールが扱う行動は, 先に述べた 12 種類の行動である.

4.3.3 模倣モジュールの仕様

模倣モジュールは, 行動を次に示す属性の組として扱う.

属性 1: 行動の種類 (“ つかむ ”, “ 置く ”, “ 捨てる ”, “ 食べる ”, “ 何もしない ”)

属性 2: 行動がなされる絶対的な位置 (図 3 中に示されている位置)

属性 3: 行動がなされる相対的な位置 (行動を行ったエージェントの右側など)

属性 4: 行動がなされる物体の種類 (“ りんご ”, “ バナナ ”, “ 石 ”, “ 何もなし ”)

属性 5: 行動を行うエージェントが持っている物体の種類 (“ りんご ”, “ バナナ ”, “ 石 ”, “ 何もなし ”)

行動が観測されていないときは, 模倣モジュールが出力する行動価値は 0 である.

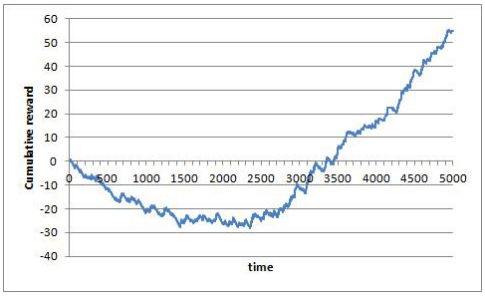

図 4: Cumulative reward in Experiment 1.

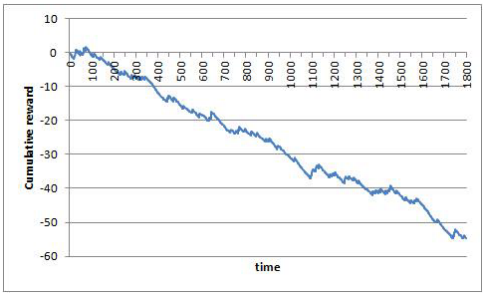

図 5: Cumulative reward in Experiment 2.

4.3.4 実験内容

以下の実験を行う.

実験 1: 強化学習モジュールのみを用いた学習実験. 強化学習モジュールにおいて, $\alpha = 0.9$, $\gamma = 0.9$ とする.

実験 2: 模倣モジュールのみを用いた学習実験. 模倣モジュールにおいて, $\alpha = 0.1$ とする.

実験 3: 提案モデルを用いた学習実験 (模倣モジュールでの学習なし). 強化学習モジュールにおいて, $\alpha = 0.9$, $\gamma = 0.9$ とする.

実験 4: 提案モデルを用いた学習実験 (模倣モジュールでの学習あり). 強化学習モジュールにおいて, $\alpha = 0.9$, $\gamma = 0.9$, 模倣モジュールにおいて, $\alpha = 0.1$ とする.

5. 実験結果

実験 1 での累積報酬を図 4 に示す. 時刻 1800 での累積報酬は -25.1 であった. 時刻は強化学習モジュール内の離散時間である. 累積報酬は時刻 1700 付近で極小である.

実験 2 での累積報酬を図 5 に, 各属性の重みの変化を図 6 に示す. 時刻 1800 での累積報酬は -54.7, 属性 i の重みを w_i とすると, 時刻 1800 での各属性の重みは, $w_1 = 0.228$, $w_2 = 0.010$, $w_3 = 0.269$, $w_4 = 0.336$, $w_5 = 0.157$ であった.

実験 3 での累積報酬を図 7 に示す. 時刻 1800 での累積報酬は 53.5 であった. 累積報酬は時刻 400 付近で極小である.

実験 4 での累積報酬を図 8 に, 各属性の重みの変化を図 9 に示す. 時刻 1800 での累積報酬は 90.0, 属性 i の重みを w_i とすると, 時刻 1800 での各属性の重みは, $w_1 = 0.221$, $w_2 = 0.00310$, $w_3 = 0.271$, $w_4 = 0.298$, $w_5 = 0.179$ であった. 累積報酬は時刻 200 付近で極小である.

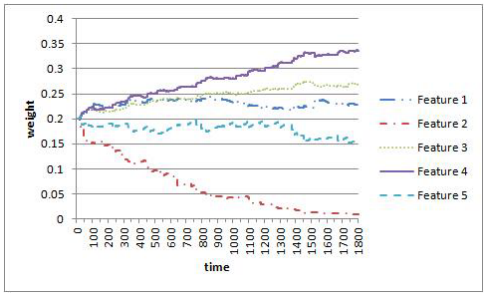

図 6: Change in feature weights in Experiment 2.

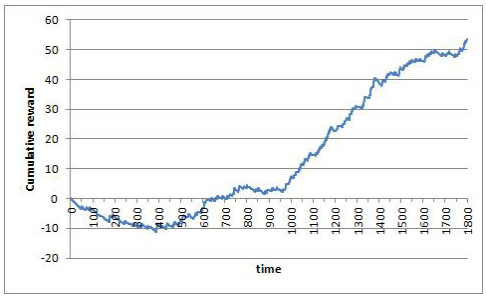

図 7: Cumulative reward in Experiment 3.

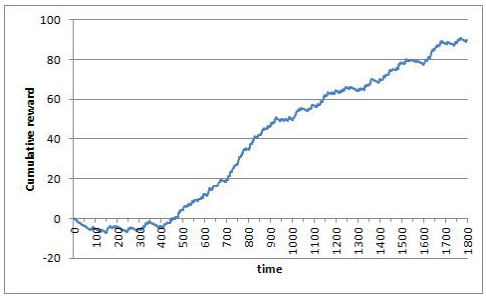

図 8: Cumulative reward in Experiment 4.

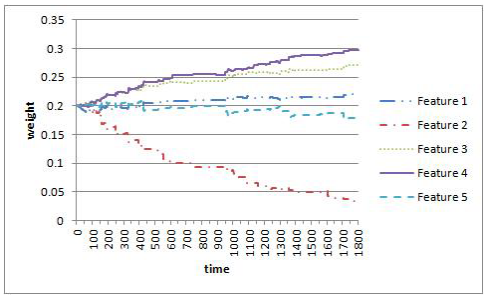

図 9: Change in feature weights in Experiment 4.

6. 考察

実験 2 と実験 3 において，属性の重み w_i を，時刻 1800 における値の高い順に並べると，$\{w_4, w_3, w_1, w_5, w_2\}$ となる．重み w_2 は 5 つの重みの中で特に小さい値をとっている．これは，属性 2，“行動がなされる絶対的な位置”が他のエージェントの行動と一致している行動は失敗しやすいためであると思われる．例えば，“場所 0 にある物体をつかむ行動”を模倣しようとすると，すでにその場所には物体が存在しないので失敗する．重み w_3 の値が高いのは，属性 3，“行動がなされる相対的な位置”が他のエージェントの行動と一致している行動に，報酬を獲得する行動，すなわち，“食べ物を食べる”行動が含まれているためであると思われる．

実験 1 での累積報酬が時刻 1700 付近で極小であるのに対し，実験 3 での累積報酬は時刻 400 付近で極小である．この結果は，実験 3 ではエージェントが実験 1 よりも早く環境に適応できたことを示している．

実験 3 での時刻 1800 における累積報酬が大きな正の値であるのに対し，実験 2 での時刻 1800 における累積報酬は大きな負の値である．これは，実験 2 ではエージェントが学習が進んでも属性の重要度に基づき，他のエージェントの行動を，たとえそれが負の報酬が期待される行動であっても模倣してしまうのに対し，実験 3 では学習が進むと正の報酬が期待される行動を自律的にとるようになり，他のエージェントが負の報酬が期待される行動を行っても模倣しなくなるためであると思われる．実験 3 の結果は他のエージェントの行動の模倣が学習の指針となりうることを示し，実験 2 の結果は他のエージェントの行動の模倣がそれだけでは環境に適応するには不十分であることを示している．

実験 3 での累積報酬が時刻 400 付近で極小をとっているのに対し，実験 4 での累積報酬は時刻 200 付近で極小をとっている．また，実験 3 での時刻 1800 における累積報酬が 53.5 であるのに対し，実験 4 での時刻 1800 における累積報酬は 90.0 である．これは，実験 4 では各属性の重みの学習を行ったためであると思われる．実験 3 と実験 4 の結果は属性の重みの学習が，エージェントが望ましい模倣行動をとるのに有効であることを示している．

7. おわりに

本研究では，模倣と自律行動の計算モデルを提案した．提案モデルを食卓シミュレータ上に実装し，累積報酬に基づいた評価を行った．さらに，模倣モジュール内で各属性の重みの学習を行い，その効果を調べた．

その結果，提案モデルを実装したエージェントが，強化学習のみを行うエージェントよりも早く環境に適応できることを示せた．また，行動の各属性の重みを学習することによって，エージェントの学習時間をより短縮できることも示せた．

本研究では提案モデルを食卓シミュレータを用いた仮想環境で評価した．提案モデルを実環境に実装して評価することが，次の我々の課題である．

参考文献

[Asmuth 08] Asmuth, J., Littman, M. L., and Zinkov, R.: Potential-based Shaping in Model-based Reinforcement Learning, in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, pp. 604–609, Chicago, IL (2008)

[Sutton 98] Sutton, R. S. and Barto, A. G.: *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA (1998)

[Wiewiora 03] Wiewiora, E., Cottrell, G., and Elkan, C.: Principled Methods for Advising Reinforcement Learning Agents, in *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 792–799, Washington, DC (2003)