F-001

Q 学習を用いたルール優先度自動決定機構の構築
Construction of Rule Priority Automatic Setting System Using Q-Learning

板津呂 翔† 打矢 隆弘† 内匠 逸†
Syo Itazuro Takahiro Uchiya Ichi Takumi

1. はじめに

近年，インターネットの急速な普及に伴い，ネットワークサービスは大きく発展し，我々を取り巻く社会におけるネットワークサービスの重要性が高まっている．新たなサービスの登場や既存のサービスの更新などにより，ユーザのニーズは多様化し，頻繁に変化する．このような背景から，動作環境に対応して自律的に処理を行う，柔軟なシステムの必要性が高まっている．

柔軟なシステムを実現する手段として，エージェント指向コンピューティングに注目が集まっている．エージェント指向とはオブジェクト指向を発展させた手法であり，オブジェクト指向におけるオブジェクトに，自身が持つパラメータや手続きを環境に応じて変化させる機能を付加し，自律的に動作するエージェントを生成する手法である．またエージェントには，過去の行動の結果から最適な行動を学習する，「学習性」と呼ばれる特性を付加することができる．学習性を実装することで，より効率的な動作が期待できる．

エージェント指向コンピューティングに基づきエージェントを効率的に開発・運用するためには，エージェントフレームワークと呼ばれる枠組みを用いる必要があり，用途に応じて様々なフレームワークが開発されている．しかし多くのフレームワークにおいて，学習性の実装はエージェント設計者に全て任されており，フレームワーク側からの支援は存在しない．そのため，学習エージェントの設計には機械学習についての専門知識と多大な労力を要する．

このような背景から，本研究では，学習エージェント設計者の支援を目的とする．そのための仕組みとして，エージェントフレームワークDASH[1]を基に，Q学習を用いた自動的な学習機能を付加した機構を提案する．本機構を用いて学習性実装に必要な記述を削減することで，専門知識が無くとも学習エージェントの設計を可能とし，設計者の負担軽減を実現する．

本稿では，前提知識であるDASH，Q学習について説明した後，提案手法であるQ学習を用いたルール優先度自動決定機構について説明する．最後に評価実験を通して提案手法の有効性を検証し，評価を行う．

2. エージェントフレームワーク DASH

本研究では，エージェントフレームワークの 1 つである DASH(Distributed Agent System based on Hybrid architecture) を用いる．DASH エージェントは if-then 型のルールに基づき行動するルールベース型エージェントであり，推論機構と呼ばれる，自身の行動を決定する仕組みを有している (図 1)．推論機構は，推論エンジン，ワーキングメモリ(以下 WM)，ルール集合から成る．

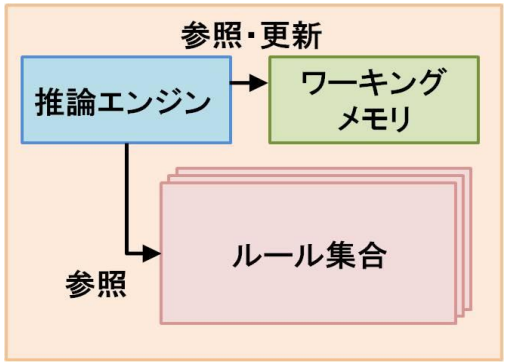

図 1：推論機構

DASH エージェントは次のように動作を行う．

1. 動作知識の制御を行う推論エンジンが WM を参照し，その内容にマッチするルールを探索する．
2. マッチしたルールを実行する．
3. ルールを実行した結果 WM の内容が更新された場合，1 へ戻る．

3. Q 学習

3.1 概要

Q 学習[2][3]は，強化学習分野における代表的な手法の 1 つとして知られている．強化学習とは，エージェントのとった行動の結果に対して評価を与え，エージェントがその評価を最大化するよう，試行錯誤的に学習を行う手法である．

Q 学習はマルコフ決定過程[1]において十分な回数の試行を繰り返した場合，最適解への収束が証明されている．本研究では問題の簡単化のためマルチエージェントは扱わず，マルコフ性を持ったシングルエージェントをのみを想定している．よって，信頼性の高い学習が可能であると考えられる．

3.2 学習のプロセス

Q 学習では，エージェントが持つ各ルールの優先度を，行動の結果を基に更新していくことで学習を行う．具体的な学習の流れを以下に示す．

1. エージェントは現在の状態(環境)を観測し，それにマッチするルールを探索する．状態の定義はエージェントにより異なるが，DASH においては，状態とは WM の状態を指す．
2. マッチしたルールが複数存在した場合，エージェントの持つ行動選択手法に基づき，その中から 1 つを選択する．
3. 2 で選択したルールを実行する．

†名古屋工業大学 大学院 工学研究科 情報工学専攻
Nagoya Institute of Technology，Graduate School of Engineering

[1] 現状態と選択した行動により次状態が決定する数学モデル．

4. 実行結果から更新式に従いルール優先度を更新する．
5. エージェントの動作の終了判定を行う．判定方法はエージェントにより異なるが，さらにエージェントの動作が続く場合は，1 に戻る．

3.3 行動選択手法

強化学習において，学習のみを目的とする場合，本来は行動選択手法を考える必要は無い．ルールを実行することで試行錯誤的に学習が進むため，全ルールが十分な回数実行されれば，学習は完了する．よって，ランダムにルールを選択しているだけでよい．しかし実際には，学習とシステム運用が同時に求められる場合がほとんどである．そのような場合には，なるべく全てのルールを実行しつつ，優先度の高いルールを実行するよう，行動選択手法を考える必要がある．
代表的な行動選択手法として，ε-greedy法，ソフトマックス法などが知られている．ε-greedy法は確率$1-\varepsilon$で優先度が最大のルールを選択し，確率 ε でランダムにルールを選択する手法である．ソフトマックス法はルールの優先度の比によって各ルールの選択確率を計算し，その確率に基づいて行動選択を行う手法である．

3.4 更新式

3.4.1 概要

Q 学習では，以下に示す更新式を用いてルール優先度の更新を行っている．

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\left[r + \gamma \max_{a' \in A(s')} Q'(s',a')\right]$$

更新式において，Q(s,a)は状態sにおいて行動aをとるというルールの優先度，$Q'(s',a')$は遷移先の状態s'において行動a'をとるというルールの優先度を表している．またrは状態sからs'への遷移において得られる報酬，$A(s')$は状態s'で実行可能な行動全体の集合を表し，$\alpha(0<\alpha<1)$は学習率，$\gamma(0<\gamma<1)$は割引率を表している．
優先度の更新はエージェントが行動を実行し，次の状態に遷移する度に行われるが，エージェントは遷移によって得られた報酬rと，遷移先で実行可能なルールの中で優先度が最大であるルールの値$(\max_{a' \in A(s')} Q'(s',a'))$を基にして，ルールの優先度の更新を行っている．

3.4.2 学習率と割引率

学習率とは，ルールの優先度の更新を行う際に，いままでの優先度の値を優先するか，得られた結果（ここでは，得られた報酬・遷移先で実行可能なルールの情報）を優先するかのバランスを表したパラメータである．更新式より，αが0 に近づくほどいままでの優先度の値を重視し，逆に1 に近づくほど得られた結果を重視して優先度の更新を行うことがわかる．なお，学習率は0.1 程度の値を設定することが一般的である．
割引率とは，将来獲得予定の報酬$(\max_{a' \in A(s')} Q'(s',a'))$を現時点でどれだけ重要視するかの度合を表したものである．更新式では，状態sからs'への遷移によって報酬rが得られたが，まだ状態s'では行動していないため，その先の行動から良い結果が得られるという完全な保証は無い．よって，状態s'で実行可能なルールの優先度は考慮に入れる必要はあるが，ある程度割り引いて考える必要がある．その度合を表したものが割引率である．更新式より，γが0に近づくほど将来獲得予定の報酬を軽視し，逆に1 に近づくほど重視することがわかる．なお，割引率は0.9～0.99 程度の値を設定することが一般的である．

2. プロトタイプの実装

4.1 概要

本研究で提案するルール優先度自動決定機構は，DASH エージェントが持つ if-then 型のルールに優先度を付加し，Q 学習を用いて優先度の調整を行うものである．本機構は，4.2 節で示す各機能を有している．

4.2 機能

4.2.1 自動学習機能

Q 学習を用いて，自動的にルール優先度の更新を行う．優先度の更新を行うための仕組みとして，自動学習機構を新たに導入し，既存の DASH に組み込まれている推論機構と連携して運用する．自動学習機構は，行動選択エンジンと学習エンジンによって構成されている．

● 行動選択エンジン

行動選択を行う．行動選択手法としてε-greedy法とソフトマックス法が実装されており，ユーザがエージェント起動時にどちらの手法を使用するかを決定する．

● 学習エンジン

行動選択エンジンが選択したルールを実行した結果から，更新式を用いて実行したルールの優先度を更新する．

4.2.2 学習データの保存及び参照機能

ある程度学習を行った後，各ルールのルール名，優先度を学習データとして CSV 形式のファイルで保存する．このファイルを学習データファイルと呼ぶ．また次に同じエージェントを動かす際，先に学習データファイルを読み込み，各ルールの優先度をセットしてから動作を開始する．
この機能により，学習の中断・再開が可能となる．また将来的な応用例として，他のユーザが行った学習結果を，新規ユーザが利用することを考えている．

4.2.3 グラフ自動描画及び保存機能

学習が進むにつれて，エージェントの動作が効率化していく様子を視覚的に確認するために，学習過程を示すグラフの自動描画を行う．エージェントが初期状態から目標状態に辿り着くまでを 1 試行とし，1 試行毎のルール実行回数をグラフ形式で自動的に表示する．また，過去の学習過程を確認するために，グラフの座標を DAT ファイルに保

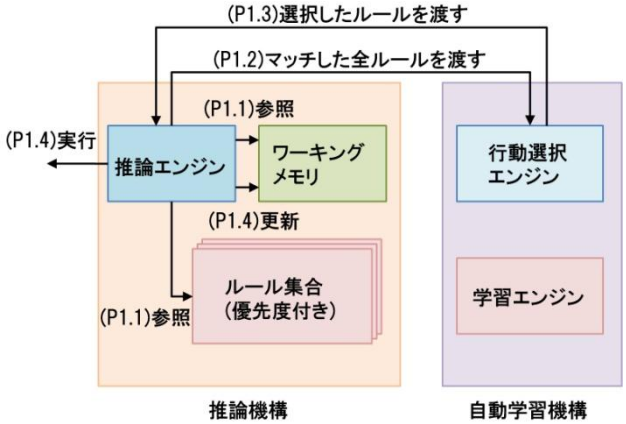

図2：行動選択・実行フェーズ

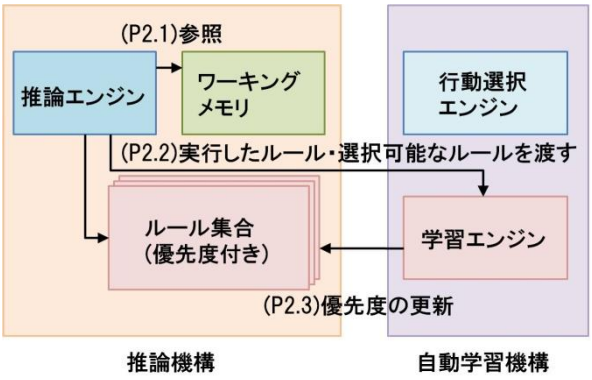

図3：優先度更新フェーズ

存する機能も有する．

4.3 動作の流れ

本機構の動作は，行動選択・実行フェーズ(図2)，優先度更新フェーズ(図3)の2段階に分かれる．

- **行動選択・実行フェーズ**
  - **(P1.1) WMの内容を参照**
  - **(P1.2) 選択可能なルールを行動選択エンジンへ送る**
  - **(P1.3) 実行するルールを選択**
  - **(P1.4) ルールを実行**
- **優先度更新フェーズ**
  - **(P2.1) WMの内容を参照**
  - **(P2.2) 実行したルール，選択可能なルールを学習エンジンへ送る**
  - **(P2.3) 実行したルールの優先度を更新**
  - **(P2.4) 終了判定**

5. 評価実験

5.1 実験環境

実験に用いた計算機環境を以下に示す．

- OS：Windows 7 Professional
- CPU：Intel(R) Celeron(R) CPU 450 @2.20GHz 2.19GHz
- メモリ：2.00GB
- システムの種類：32bit OS
- JAVA：version 1.6.0_20

5.2 実験 1：動作確認

5.2.1 概要

以下に示すサンプルエージェントを作成し，学習率を 0.1，割引率を 0.9 と設定し，学習を行わせることにより，適切に学習が行われることの確認を行った．また，グラフ自動描写機能の動作確認も同時に行った．ε-greedy 法を用いた場合とソフトマックス法を用いた場合の各々について 100 回ずつ試行を繰り返し，目標状態に達するまでのルール実行回数の変化を観察した．

meiro.dash

meiro.dashは図4に示す迷路問題を解くエージェントである．エージェントはそれぞれの部屋から移動可能な部屋の情報のみを持ち，スタート(S)からゴール(G)までの最短経路を探索する．

5.2.2 結果・考察

実験を行った結果，グラフ自動描写機能により得られたグラフを図5，図6に示す．横軸は試行回数(Number of Trials)，縦軸は移動回数(Number of Episodes)を表している．

図4：迷路問題

どちらの場合においても，移動回数が徐々に最適解に収束していくことが確認できた．よって，適切な学習が行われているものと考えられる．

5.3 実験 2：記述量削減率の調査

5.3.1 概要

提案手法により学習エージェント設計者の負担が軽減されることを示すため，学習に関連するプログラム記述量削減率の調査を行った．

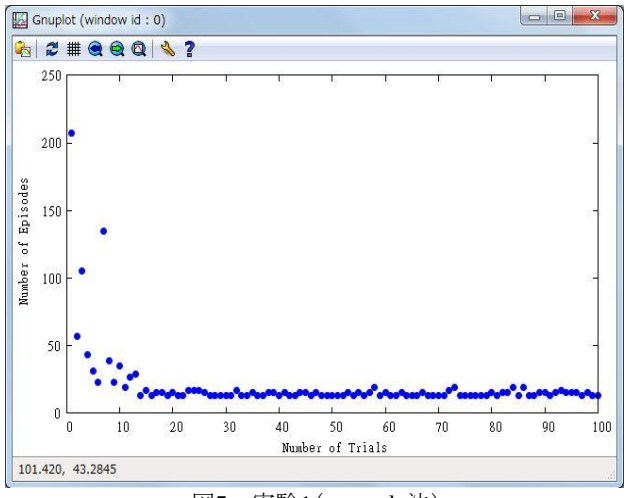

図5：実験1(ε-greedy法)

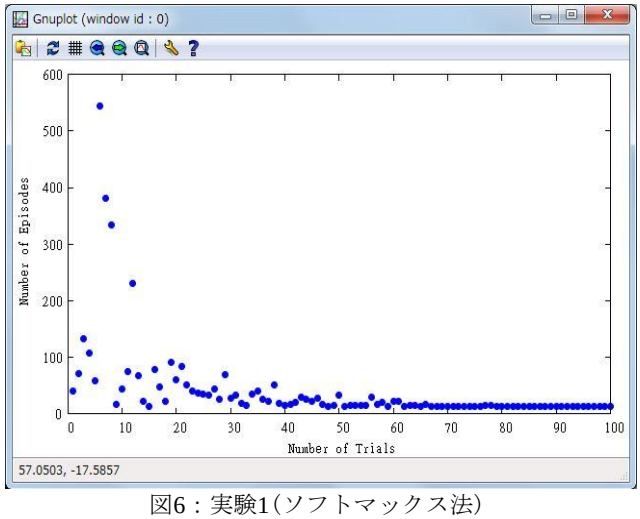

図6：実験1(ソフトマックス法)

- 記述量A：従来のDASHに対して学習機能を付加し，学習エージェントを作成した場合の，学習機能使用に必要な全記述量(単位：行)
- 記述量B：提案手法を用いて学習エージェントを実装した場合の，学習機能使用に必要な記述量(単位：行)

これらを測定し，比較することにより，学習に関する記述量削減率を計算した．本実験で用いた記述量削減率は，

$$\text{記述量削減率} = (1 - \frac{B}{A}) \times 100$$

として導出した．

5.3.2 結果・考察

実験 2 の結果を以下に示す．

表 1：学習に関する記述量削減率

記述量 A(記述行数)	記述量 B(記述行数)	削減率 (%)
187	4	97.6

表1より，学習に関する記述量が約97.6 %削減され，学習性実装に要する労力を軽減できていることがわかる．よって，本研究の目的である，専門知識を必要としない学習エージェントの設計，及び，設計者の負担軽減が達成されており，提案手法は有効であると考えられる．

5.4 実験 3：アンケート調査

5.4.1 概要

本研究で提案した優先度自動決定機構の有用性を検証するため，エージェントフレームワークDASHを用いたエージェント開発経験のある被験者9名にプロトタイプを使用してもらい，各機能の利便性・操作性や改善点・要望に関するアンケートを実施した．利便性・操作性は，1〜5の5段階で評価を依頼した．

5.4.2 結果・考察

アンケートの結果を以下に示す．利便性，操作性についての評価をそれぞれ表2，表3に，全体的な総合評価を表4に示している．

アンケート結果から，自動学習機能，グラフ自動描画機能の利便性に関しては，比較的高い評価を得ていることがわかる．しかし操作性の評価がやや低くなっているため，点数評価とは別に実施した自由記述による要望・改善案を参考に，インタフェースの改善を行う必要がある．また，DATファイル保存機能，学習データファイルの保存・参照機能に関しては，利便性・操作性共に比較的低い評価を得ている．低評価の理由として，これらの機能は実装途上にあり，現段階では，どのような場面で使用して良いかわかりにくいという意見が寄せられている．よって，更なる機能拡張により，使い方がわかりやすい機能を実装する必要がある．

表2：アンケート結果(利便性)

点数	1	2	3	4	5	平均
自動学習機能	0	0	1	4	4	4.33
グラフ自動描画	0	0	0	5	4	4.44
DAT ファイルの保存	0	1	5	3	0	3.22
学習データファイルの保存・参照	0	0	6	3	0	3.33

表3：アンケート結果(操作性)

点数	1	2	3	4	5	平均
自動学習機能	0	0	3	6	0	3.67
グラフ自動描画	0	0	3	4	2	3.89
DAT ファイルの保存	0	1	6	1	1	3.22
学習データファイルの保存・参照	0	2	6	1	0	2.89

表4：アンケート結果(総合評価)

点数	1	2	3	4	5	平均
総合評価	0	0	2	7	0	3.78

6. まとめ

本研究では，学習エージェント設計者の支援を目的とし，エージェントフレームワークDASH を対象に，Q学習を用いた優先度自動決定機構の提案と実装を行った．本機構は主に，以下の3つの機能を持つ．

- **自動学習機能**
- **学習データの保存及び参照機能**
- **グラフ自動描画及び参照機能**

評価実験を通して，自動学習機能によりエージェントの動作が効率化されることを確認した．また，各機能の評価に関するアンケートや記述量削減率に関する実験を行い，本機構の有効性の検証を行った．その結果，本機構は自動学習機能，グラフの自動描画に関しては，現段階で有用であると示された．特に自動学習機能については，高い記述量削減率を示したため，有用な機能であると言える．しかし，学習データの保存及び参照機能，グラフ(DATファイル)保存機能に関しては，現段階ではあまり有用とは言えず，改善・拡張が必要であると思われる．

今後の課題は以下の通りである．

- **マルチエージェント環境への適用**

本研究ではシングルエージェントを想定しているが，エージェントの利点を活かすためには，マルチエージェント環境への適用が求められる．しかしマルチエージェント環境では，マルコフ性が保証されないという問題がある．Q学習の最適性はマルコフ性を前提としたものであり，マルチエージェント環境では正しく動作しない可能性がある．

- **動作知識記述の支援**

現状では，適切に学習を行うためにはエージェントを実装する際，エージェントが取り得る全ての状態と行動についての動作知識(ルール)を記述する必要がある．よって，実験1で使用した迷路問題のような単純な問題であっても，迷路を大きくすると状態数が急激に増加する．マルチエージェント環境を想定すると，さらに爆発的に状態数の増加が起こる．

莫大な数の動作知識を記述することは設計者にとって大きな負担となるため，支援方法を考案する必要がある．

- **機能の改善・拡張**

評価実験で得たアンケート結果を参考に，プロトタイプが持つ各機能の改善・拡張を行い，ユーザビリティの向上を目指す．

参考文献

[1] “DASH ユーザマニュアル”, http://uchiya.web.nitech.ac.jp/idea/html/index.html
[2] 高玉 圭樹，“マルチエージェント学習-相互作用の謎に迫る-”，コロナ社，2003
[3] 謝 孟春，立花 敦，“マルチエージェントの協調行動の取得における Q 学習に関する考察”，合同エージェントワークショップ＆シンポジウム論文集，pp.441-448(JAWS2007)