1D4-03

事例に基づく関係的な強化学習のエレベータ制御問題への適用

Applying Relational Reinforcement Learning to Elevator Dispatching Problems

大久保 隆晴　亀谷 由隆　佐藤 泰介
Takaharu OKUBO　Yoshitaka KAMEYA　Taisuke SATO

東京工業大学 大学院情報理工学研究科 計算工学専攻
Dept. of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology

Q-learning is one of the most typical methods of reinforcement learning. But using Q-learning we must prepare the table for Q-values, and it often requires very large amount of memory. Relational Reinforcement Learning (RRL) is known as a method solving such a problem with relational representations. In this paper, we applied RIB-algorithm(one of RRL algorithms) to elevator dispatching problem, with a slight inprovement, and examined the algorithm's behavior. In the result, we found the algorithm could learn with small amount of memory. Additionally, we tried to use learned examples to a different environment.

1. はじめに

機械学習の分野の 1 つに強化学習 [6] がある。強化学習は、環境に関する知識を持たない状態から試行錯誤を繰り返して、行動選択を学習するという特徴を持つ。

強化学習の代表的な手法の 1 つに Q 学習 [7] が挙げられる。これは、ある状態においてある行動を取ったときの期待収益を表す行動価値関数を学習するものである。しかし、全ての状態及び行動に対する行動価値をテーブルとして保持するため、状態や行動の数の莫大である問題に対して適用した場合テーブルの大きさが膨大になり学習を困難にしてしまう。このような問題に対しては、ニューラルネットワーク等によって、過去の事例 (状態、行動、行動価値の組) より一般化を行うことにより学習を行うことがしばしばある ([1] 等)。

関係強化学習 [4](以下 RRL) は、一般化を Q 学習に組み込んだ手法として知られている。RRL では、得られた事例同士の関係に関する表現を導入することにより、保持している事例から行動価値関数を一般化する。本稿では、RRL の手法の 1 つである RIB アルゴリズムの実応用として、状態の数が莫大であるエレベータ制御問題に適用し、その動作を考察した。

2. 関係強化学習・RIB アルゴリズム

2.1 関係強化学習

Q 学習では、行動価値関数を記憶する膨大なサイズのテーブルが必要である。この問題を解決する手法の一つとして、関係強化学習 (relational reinforcement learning, 以下 RRL) [4] が知られている。RRL は、様々な状態や行動の間の関係の表現を Q 学習に取り入れ、既に得た事例から一般化された行動価値関数を導くものである。

関係表現に基づき一般化を行う帰納論理プログラミングを用いた TG アルゴリズム [3]、事例に基づく学習[*1]を用いた RIB アルゴリズム [2]、ガウス過程[*2]を用いた KBR アルゴリズム [5] が既に報告されている。

本稿では、事例に基づく学習を用いた RIB アルゴリズムを使用する。まず、関係強化学習のアルゴリズムは一般的に次のように書ける。

A: 東京工業大学 大学院情報理工学研究科 計算工学専攻 佐藤研究室 大久保 隆晴 (ohkubo@mi.cs.titech.ac.jp)

∗1　k-nearest neighbor の考え方を利用する。

∗2　共分散としてカーネルを使用する。

1. 全ての状態行動対に対して行動価値を 0 に初期化し、$\hat{Q}_0$ とする。事例の集合を空集合、$e := 0$ に初期化する。
2. $e := e + 1$、$i := 0$ とし、ランダムに初期状態 s_0 を生成する。
3. $\hat{Q}_e$ を用いた行動選択手法により、行動 a_i を確率的に選択する。
4. a_i を実行して r_i を受け取り、s_{i+1} を観測する。
5. $i := i + 1$
6. 状態 s_i がゴールに達していなければ 3 に戻る。
7. $j := i$
8. $j := j - 1$
9. $\hat{q}_j := r_j + \gamma \max_{a'} \hat{Q}_e(s_{j+1}, a')$ を用いて、事例 $x = (s_j, a_j, \hat{q}_j)$ を生成する。
10. もし事例 $(s_j, a_j, \hat{q}_{old})$ が事例の集合内に存在したら、それと x を取り換える。そうでなければ x を事例の集合に追加する。
11. $j = 0$ でなければ 8 に戻る。
12. 事例の集合より関係表現に基づく一般化アルゴリズムを用いて $\hat{Q}_e$ を $\hat{Q}_{e+1}$ に更新する。
13. 2 に戻る。

2.2 RIB アルゴリズム

RIB アルゴリズム [2] はメモリに保持する事例 (状態・行動・行動価値 の組) の個数を制限することのできるアルゴリズムである。Q 学習と同様に探索を行いデータベースに事例を追加していくことで学習を進める。後述するように、追加する事例は適当な条件を用いて制限する。また、事例が制限された個数を超えた場合も適当な条件を用いて事例を放棄する。保持している限られた個数の事例より一般の行動価値を予測するため、２つの状態行動対の間の距離 $dist_{ij}$ を定義[*3]し、状態行動対 i に対する行動価値 $\hat{q}^{(i)}$ を予測するのには下の式を用いる。

$$\hat{q}^{(i)} = \frac{\sum_j \frac{q^{(j)}}{dist_{ij}}}{\sum_j \frac{1}{dist_{ij}}} \quad (1)$$

ここで、j は事例のデータベース中に存在する状態行動対を表し、$q^{(j)}$ は状態行動対 j に対して保持している行動価値である。但し、0 による除算を避けるため、距離には小さな値 δ を足しておく。

∗3　似たような行動価値を持つ状態行動対間の距離が小さくなるように、問題に合わせて人間が適宜距離を定義する。

例の追加の制限

Driessens and Ramon は [2] において、2 つの方法を併用して追加する事例の制限を行っている。

まず、距離の近い事例の集合[*4]から予測出来る行動価値と大きく違わない行動価値を持つ事例は不要と判断して、データベースへの追加を行わない。下の式を満たす事例のみを追加する。

$$|q - \hat{q}| > \sigma_{local} \cdot F_l \tag{2}$$

ここで、q は新しい事例の探索の過程で予測された行動価値[*5]、$\hat{q}$ は保持している事例より予測された行動価値[*6]である。また、σ_{local} は、距離の近い事例の持つ行動価値の標準偏差であり、F_l は変更可能なパラメータである。

もう 1 つの方法では、保持している行動価値にばらつきの少ない領域よりもばらつきの多い領域[*7]に多くの事例を追加するために、ばらつきの少ない領域に属する事例の追加を制限する。下の式を満たす事例のみを追加することになる。

$$\sigma_{local} > \frac{\sigma_{global}}{F_g} \tag{3}$$

ここで、σ_{local} は、比較的距離の近い事例の持つ行動価値の標準偏差である。また、σ_{global} は、保持している全ての事例における行動価値の標準偏差であり、F_g は変更可能なパラメータである。

溢れた事例の放棄 (Error Proximity)

保持している値と予測される値との誤差が大きい事例を近い距離に多く持つ事例を、他の例に悪影響を及ぼしていると判断して放棄する。

$$Score_i = \sum_j \frac{|q^{(j)} - \hat{q}^{(j)}|}{dist_{ij}} \tag{4}$$

ここで、$q^{(j)}$ は保持している行動価値、$\hat{q}^{(j)}$ は予測される行動価値である。また、$dist_{ij}$ は事例 i と事例 j の間の距離である。このようなスコアをデータベース内の全ての事例に対して計算し、スコアの最も大きい事例を放棄する。

この他に、同論文において Error Contribution, Maximam Variance という手法も提案されている。

3. エレベータ制御問題への適用

3.1 本稿での問題設定

本稿では、エレベータを制御するエージェントは、状態として以下の情報を持っているものとする。

- 全てのエレベータの現在の階 (4 階建てならば、1-4 階)
- 全てのエレベータの動く方向 (上, 下, 停止)
- 全てのエレベータのドアの状態 (閉, 開, 停止 1, 停止 2)
- 全てのエレベータ内の各階に対する停車要求ボタンの状態 (ON,OFF)
- 全ての階の乗車要求ボタンの待ち時間 (99 タイムステップまで保存するならば、0-99 タイムステップ)

エレベータ 2 台、4 階建て、待ち時間を 99 タイムステップまで保存するとして、このときの状態数は、およそ 5.9×10^{17} となり、全ての状態（および行動）に対して行動価値関数のテーブルを持つ通常の Q 学習をそのまま適用することは困難である。各エレベータの取り得る行動は以下の 5 つとする。

- 動く方向を「停止」にする
- 動く方向を「上」にする
- 動く方向を「下」にする
- 動く方向を「上」にして、ドアを開ける
- 動く方向を「下」にして、ドアを開ける

また、問題を簡単にするため、以下のような制約を設ける。

- 1 つの階を移動するのにかかる時間は 1 タイムステップとし、加速などの物理的な問題は無視する。
- ドアを開けたあとは、3 タイムステップの間停車するものとし、3 タイムステップ目にドアを閉じる。
- 1 階より下に進むことはできず、最上階より上に進むこともできない。
- 乗客は、進行方向にある階の停車要求のどれかを押す。
- 乗客が停車要求を押したらその方向に進まなくてはならず、到着したらドアを開く (途中の階に停車するかは任意)。
- 停車要求が無く、動く方向と一致する乗車要求も無い階でドアを開くことはできない。

停車要求、乗車要求は事前に設定した確率に基づき、ランダムに与える。但し、これは環境のモデルに属するものであり、学習を行うエージェントは確率を知らないものとする。

3.2 学習方法

RRL アルゴリズムの一般形 (2.1 節参照) との違いは以下の点である。

- RRL アルゴリズムでは、$j = 0$ よりゴールにたどりつくまで状態を進めてから、遡って $\hat{q}_j := r_j + \gamma \max_{a'} \hat{Q}_e(s_{j+1}, a')$ を適用しているが、本稿では、Q 学習と同様に状態を訪れたときに逐一式を適用した。
- エレベータ制御問題には明確なゴールが存在しないので、100 ステップ経過した時点でゴールとする。
- 事例の数が 30 個に達するまでは、探査を重視し多様な事例を追加するため、ランダムに行動選択を行い、100 ステップ経過時にのみ事例の追加を行った。

行動選択手法には ϵ グリーディ行動選択を使用し、溢れた事例の放棄は、はじめは Error proximity を用いて行った。4.1 節より後の実験では、別の方法を用いている (4.1 節で述べる)。

報酬は、次の状態における各乗車要求の待ち時間の二乗を足し合わせたものの符号反転である。この条件のもとで得られる平均二乗待ち時間を評価する。学習の流れは以下の通りである。

1. 事例の集合を空集合、$e := 0$ に初期化する。
2. $e := e + 1$、$i := 0$ とし、ランダムに初期状態 s_0 を生成する。
3. ϵ グリーディ行動選択により、行動 a_i を確率的に選択する。但し、事例の数が 30 に満たない場合はランダムに選択する。
4. a_i を実行する。
5. a_i の結果を反映し、停車要求・乗車要求を与えて s_{i+1} を生成する。また、r_{i+1} を計算する。
6. $\hat{q}_j := r_{j+1} + \gamma \max_{a'} \hat{Q}_e(s_{j+1}, a')$ を用いて、事例 $x = (s_j, a_j, \hat{q}_j)$ を生成する。
7. 事例の追加の制限の式を満たすならば、x を事例の集合に追加する。
8. 事例の数が限定した個数に達しているなら、Error proximity を用いて事例を 1 つ放棄する。
9. 状態 s_i が 100 ステップに達していなければ 3 に戻る。
10. 2 に戻る。

また、[2] にならい、RIB アルゴリズムにおける 2 つの事例の間の距離は以下の方法で与えた。現段階では関係的知識を間接的な形で用いている。

---

*4 事前に個数を設定する。本稿では 30 個に固定した。
*5 Q 学習の更新式を用いて得られた値
*6 式 1 を用いて得られた値
*7 距離の近い事例の間での分散が大きい。

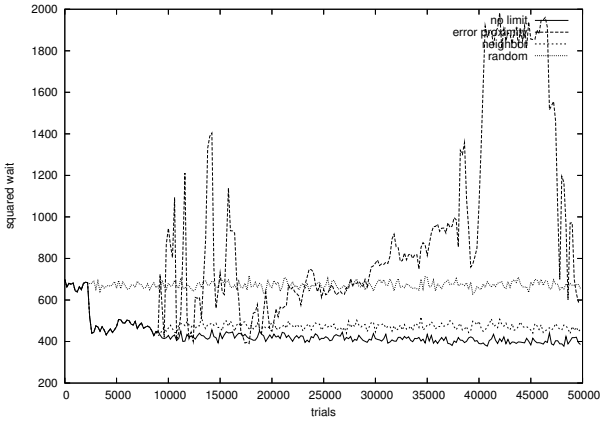

図 1: 溢れた事例の放棄に関する改良

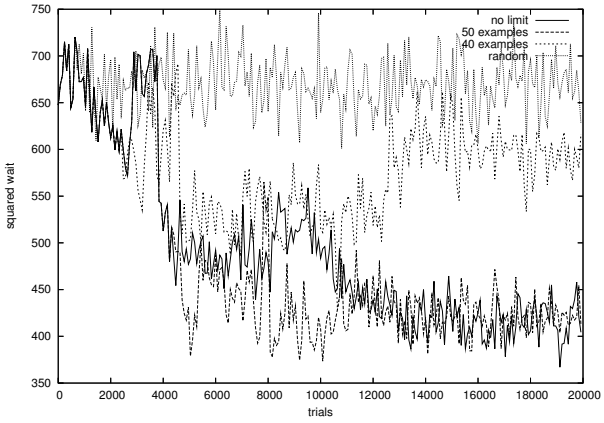

図 2: 2 台 10 階建て

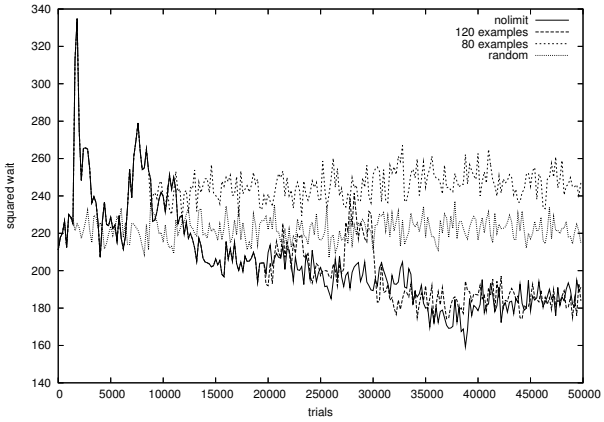

図 3: 4 台 10 階建て

- 階数、乗車要求の待ち時間については事例の間でその差を取り、停車要求、進行方向、扉の状態、行動については異なるものの数を数え、重み付けの上、*8加算する。
- 行動を次の 3 つのグループに分け、違うグループの行動を取っている場合ペナルティ*9を加算する。
  1. 「停止する」、「何もしない」
  2. 「上に動く」、「下に動く」
  3. 「上に行くと表示してドアを開ける」、「下に行くと表示してドアを開ける」
- 片方の事例において現在の階で進行方向と一致する乗車要求があり、もう片方の事例ではそれがない場合、ペナルティ(3000) を加算する。
- 片方の事例において進行方向の先の階で進行方向と一致する乗車要求があり、もう片方の事例ではそれがない場合、ペナルティ(2000) を加算する。
- 片方の事例において進行方向の先の階での停車要求があり、もう片方の事例ではそれがない場合、ペナルティ(1000) を加算する。
- 片方の事例において 1 階に居てもう片方の事例では他の階にいる場合、ペナルティ(500) を加算する。
- 片方の事例において最上階に居てもう片方の事例では他の階にいる場合、ペナルティ(500) を加算する。

4. 実験

4.1 溢れた事例の放棄に関する改良

エレベータ 2 台、10 階建てのときの平均二乗待ち時間が図 1 である。各パラメータの設定は、保持する事例の数：50、割引率 γ：0.9、ϵ グリーディ行動選択における ϵ：0.05、事例の追加の制限における F_l：1、事例の追加の制限における F_g：10、各階で乗車要求が発生する確率は、1 階から上へ：2/256、その他の階から上へ：1/256、下へ：8/256、客が乗車した際に停車要求が発生する確率は、上へ行く客：各階等確率、下へ降りる客：224/256 の確率で 1 階へ、その他の階は等確率、とした。

横軸は、100 タイムステップを 1 セットとした試行を行った回数を表す。図中の “no limit” は事例の放棄を行わない場合であり、“error proximity” は Error Proximity を用いて事例を放棄した場合、“random” は 3.1 節の制約に従った上で取り得る行動をランダムに選択した場合である。*10

Error proximity を用いた場合の動作が不安定であることが分かる (最大または最小の行動価値を持つ事例が放棄されやすく、内挿のみによって予測を行う RIB アルゴリズムでは予測できる範囲が狭まる事がある)。ここでは、Error proximity を改良した新しい事例の放棄方法を提案する。

改良した事例の放棄方法

近い距離に複数の事例は不要と判断し、片方を捨てることにする。全ての 2 つの事例の組に対し距離を計算し、最も距離の近い 2 つの事例を選ぶ。その 2 つにたいして式 4 を用いてスコアを計算し、スコアの大きい方の事例を放棄する。

改良した事例の放棄方法を用いた場合の結果は、図 1 内の “neighbor” である。

4.2 必要となる事例の数

保持しておく事例の数を変化させて、エレベータ 2 台 10 階建て、4 台 10 階建ての場合について実験した。各パラメータは 4.1 節と同じである。溢れた事例の放棄では、4.1 節で改良した方法を用いている。

エレベータ 2 台 10 階建てでの平均二乗待ち時間が図 2 である。図中 “no limit” は事例の放棄を行わない場合であり、“50 examples”、“40 examples” は、それぞれ事例の数を 50 個、40 個に制限した場合である。事例数 50 個で制限を行わない場合

∗8　階数 0.3、停車要求 0.2、進行方向 3、扉の状態 0.3、乗車要求 2、行動 10。事前の実験により値を決定した。(以下同様)

∗9　1 と 2、1 と 3 の場合 30000、2 と 3 の場合 10000

∗10　3.1 節の制約によって、ランダムでもそれなりの動作はする。

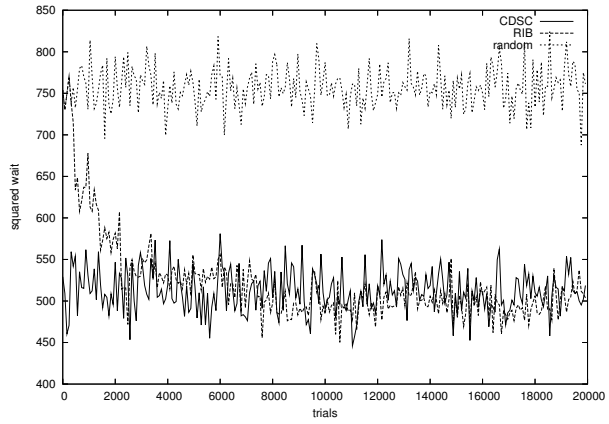

図 4: CDSC との比較

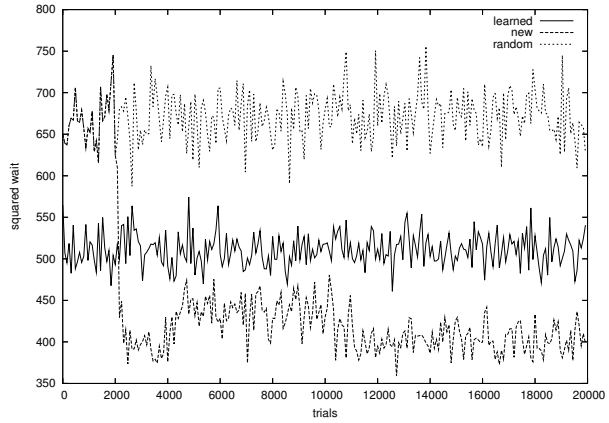

図 5: 4 → 10 階建て

とほぼ同等の結果が得られた。

エレベータ 4 台 10 階建てでの平均二乗待ち時間が図 3 である。事例数 120 個で制限を行わない場合とほぼ同等の結果が得られた。

4.3 他のアルゴリズムとの比較

RIB アルゴリズムによる結果を他のアルゴリズムによる結果を比較するため、発見的に見つけ出された簡単なアルゴリズムである、CDSC アルゴリズムを用いて比較を行った。このアルゴリズムは、実際に広く適用されているアルゴリズムである [8]。

各パラメータの設定は、保持する事例の数：200、ϵ グリーディ行動選択における ϵ：徐々に 1 から 0 へ変化させる、事例の追加の制限における F_l：1、事例の追加の制限における F_g：10、各階で乗車要求が発生する確率は、1 階から上へ：4/256、その他の階から上へ：1/256、下へ：16/256、その他は 4.1 節と同じとした。

結果が図 4 である。“CDSC” と “RIB” で、ほぼ同等の二乗待ち時間となる行動選択を得ることができた。

4.4 学習結果の一般化

エレベータ 2 台、4 階建ての場合の学習結果のデータベースに以下の操作を加え、エレベータ 2 台、10 階建ての場合の学習に用いた。保持する事例の数の制限は 150 個とした。その他のパラメータは 4.1 節と同じである。

- 最上階に居るエレベータは 4 階より 10 階に移動。途中階も同様に調整。
- 停車（乗車）要求も同様に移動し、残りの階の停車（乗車）要求は OFF に設定。
- 行動価値を $\frac{10^2}{4^2}$ 倍にする。

4 階建て→ 10 階建てのときの平均二乗待ち時間が図 5 であり、図中の “learned”が違った階数の学習結果を利用したもの、“new”がそれを利用せずに事例を持たない状態から学習したものである。

ランダムに行動選択を行うより良好な結果が得られていることより、4 階建てでの学習より 10 階建てにおいても有用な結果を学習できている、と考えられるが、10 階建てで学習を続けた場合にそれ以上の改善はほとんど見られなかった。問題が簡単なため、瞬時に新たな環境に適応したと考えられる。また、“learned” における学習結果は、“new” におけるそれに劣ることが分かった。

5. 結論・今後の課題

RIB アルゴリズムでは、少ない個数の事例より行動価値を予測し、学習を進めることができることが実験によって分かった。

溢れた事例の放棄方法を改良することにより、動作が不安定になる問題を実験では回避することができた。但し、廃棄する事例を決定する際に保持する事例の数の 2 乗のオーダの計算時間がかかり、改良の余地が残る。今後の課題としては、学習後に得られるデータベース中の事例から有用な規則を抽出することが挙げられる。

参考文献

[1] Robert H. Crites and Andrew G. Barto. Improving elevator performance using reinforcement learning. In *Advances in Neural Information Processing Systems*, Vol. 8, pp. 1017–1023, 1996.

[2] K. Driessens and J. Ramon. Relational instance based regression for relational reinforcement learning. In *Proc. of ICML 2003*, pp. 123–130, 2003.

[3] K. Driessens, J. Ramon, and H. Blockeel. Speeding up relational reinforcement learning through the use of an incremental first order decision tree algorithm. In *Proc. of ECML2001*, pp. 97–108, 2001.

[4] S. Džeroski, L. De Raedt, and K. Driessens. Relational reinforcement learning. *Machine Learning*, Vol. 43, pp. 7–52, 2001.

[5] T. Gartner, K. Driessens, and J. Ramon. Graph kernels and Gaussian processes for relational reinforcement learning. In *Proc. of ILP 2003*, pp. 146–163, 2003.

[6] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1998. 三上貞芳, 皆川雅章共訳, 『強化学習』, 森北出版, 2000.

[7] C. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, 1989.

[8] 稲元勉, 玉置久, 村尾元, 北村新三. エレベータ運行計画問題の静的最適化モデルと分枝限定法. 電気学会論文誌, Vol. 123-C, No. 7, pp. 1334–1340, 2003.