TD(λ)-MC 法を用いた評価関数の強化学習

大崎泰寛　　柴原一友　　但馬康宏　　小谷善行
東京農工大学工学部情報工学科

概要

思考ゲームを題材とした強化学習において，TD(λ)法は高い学習成果を示してきた．ところがこの学習手法では, その性質から学習前の評価関数がある程度精度の高いものでないと成果が得られないという欠点がある．本稿では TD(λ)法に MC シミュレーションの特性を活かし，効果的な改良を加えた TD(λ)-MC 法という新しいアルゴリズムを提案する．この手法は学習の教師値を MC 法によって派生した末端局面から得られた勝敗という“事実”を平均化した値に定めて学習を推し進めたものである．本稿では,従来の TD(λ)法との比較実験を通して,その新しい学習手法である TD(λ)-MC 法の成果の検証を行っている．

Reinforcement Learning of Evaluation Functions Using Temporal Difference-Monte Carlo learning method

Yasuhiro OSAKI　Kazutomo SHIBAHARA　Yasuhiro TAJIMA　Yoshiyuki KOTANI
Department of Computer Science , Tokyo University of Agriculture and Technology

Abstract

Temporal Difference learning has showed good learning result of reinforcement learning on the theme of thought games. However, by this learning method, we can not get good result if the precision of evaluation function before learning is not considerably high. In this paper, we suggest a new algorithm called TD(λ)-MonteCarlo learning, added TD to MC simulation effectively. This new method has one of the features, the teacher-values are the values given by averaging "the fact" —won_or_lost from many leaf-positions which were yielded by MC simulation. In this paper, we inspect the result of TD(λ)-MC, comparing with the former TD learning.

1.　はじめに

情報処理学会の分科会である GPCC (Games and Puzzles Competitions on Computers)2007 年度の課題として挙げられているブロックスデュオにおける，コンピュータの的確な評価関数を設計することが本研究のねらいである．

評価関数とは，思考ゲームにおいて与えられた局面がどの程度優勢にあるかを表現する関数であり，一般的にその評価値は，局面から得られる評価要素（パラメータ）に，各々の重みを掛け合わせた線形和として与えられるものである．従って，複雑な形勢判断が要される思考ゲームにおいて，より正確に状況

を捉えられる評価関数の設計こそが強いゲームプログラムには必須とされている.

今日までに，評価関数における自動調整の研究には多くの研究者が携わり，とりわけ TD 法(Temporal Difference：時間的差分学習)[1]はコンピュータ将棋やバックギャモン等において，その高い学習成果が報告されている．[2]

ただ，この TD 法は未来と現在との観測状況における静的な評価値の差と勾配から学習するものであり，未来の静的な評価値が教師値となるので，無論その精度が重要となる．よって，局面の形勢をある程度的確に捉えられる評価関数が確立されていないことには，従来の TD 法による学習成果はそう多く望めるものではない.

そこで，Minimax 法によって得られた評価値に着眼した TD(λ)-leaf 法[3]の特長に注目し，MC(Monte Carlo)法で得た潜在的な勝敗や形勢を学習に取り込んだ新たな学習手法である TD(λ)-MC 法を提案する．本手法では教師値が評価関数による値ではなく MC 法に依存したものであるので，学習前の評価関数の精度への配慮は不要である.

本稿では，研究分野として未踏のブロックスデュオの評価関数の設計に対して，従来の TD 法を拡張した汎用性の高い学習手法である TD(λ)-MC 法を提案し，従来の学習手法と比較してどの程度学習成果が向上したかを検証する.

以下に，本稿の構成を述べる．2 章では，関連研究とブロックスデュオのルールを述べ，3 章では著者が提案した新しい学習手法である TD(λ)-MC 法のアルゴリズムを具体的に説明し，4 章では実験とその結果を示し，5 章では結論と今後の展望を記す.

2. 強化学習

強化学習(reinforcement learning)とは，環境との相互作用を通して，適切な行動戦略を獲る，教師なし学習の一種である．また,その学習と意思決定を行う主体のことはエージェント(agent)と呼ばれ，エージェントの外部の全てから構成されるものは環境(environment)と呼ばれる.

このエージェントと環境との相互作用とは以下の図 1 のように「状態」「行動」「報酬」から成り立っている.

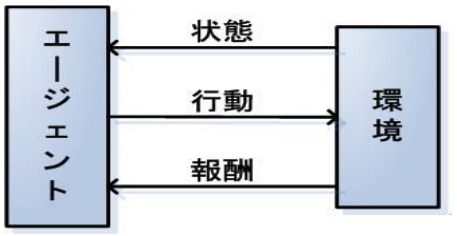

図 1 エージェントと環境との相互作用

そしてこの学習方法の最終的な目標は，環境から与えられる報酬が最大となるような行動を，エージェントが環境に対して行われるように調整することである.

最近の強化学習の代表的なものとして TD(λ)法[4]が挙げられる．TD(λ)法とは，MC 法と同様に経験から直接学習することが可能であり，動的計画法と同様に終端状態まで推移せずとも他の推定値の学習結果を利用することによって推定値を更新することができる学習手法である.

P_t は時刻(手数)t における局面の評価値であるとし，α は学習率，適格度 λ は過去の予言への依存度を表す正の定数[5](0≦ λ ≦1)としたとき，基本的な TD(λ)法の学習式は次頁の式 1 で表される.

$$\vec{w} \leftarrow \vec{w} + \alpha(P_{t+1} - P_t)\sum_{i=0}^{t} \lambda^{t-i} \nabla_w P_i \quad (式\ 1)$$

この式1のように，漸進的に評価関数が収束するまで更新するというアプローチは，強化学習において現在最も広く利用されている手法である．次の 3 章では更なる向上を図るため，MC シミュレーションの考え方をそれに組み込んだ新しい学習手法を提案する．

２．１　ブロックスデュオのルール

二人零和完全情報ゲームの一種であるブロックスデュオのルールを以下に解説する．

- 14×14 マスの正方の盤面に，モノミノからペントミノまでの計 21 種の駒を交互に置く．
- 各プレイヤは盤上のスタート地点から交互に置く．(スタート地点は右図の丸囲み部分)
- 自陣の駒の角と角をつなげて置き，このとき自陣の駒の辺と接してはならない．
  また駒は表裏好きなように置ける．
- 各プレイヤに可能手がなくなった場合，盤上の置いた累計面積の多い方の勝ちである．もし等しい場合は引き分けとする．
- 可能手がある限り駒を交互に置き，任意にパスをすることはできない．
- 一方のプレイヤに可能手がなくなった場合パスをして，他方は連続して駒を置く．

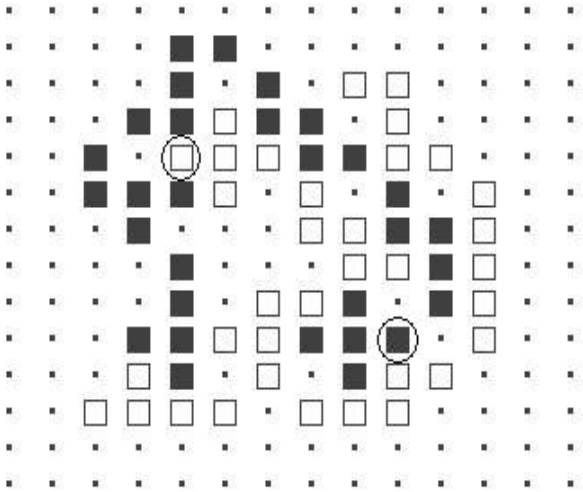

図２　ブロックスデュオの対局例

3.　　TD(λ)-MC 法アルゴリズム

本章では TD(λ)-MC 法アルゴリズムの説明とその核心に触れる．2 章で述べたとおり，従来の TD 法は各パラメータの重み列 $\vec{w}$ を現在と未来の評価値の差から学習を行っているが，翻って本手法ではある局面の静的な評価値とその局面から MC シミュレーションによって派生した末端局面の勝敗，つまり潜在的な値との差から学習するという点が決定的に異なる．従って，学習の目標となる教師値は，学習中の暫定的な評価関数による不確かな値ではなく，派生し得る末端局面の勝敗という確かな事実から構築されるべきである，という留意が本学習手法に組み込まれている．よって，本手法によって学習された評価関数は，ある局面にて MC 法によって得られた値と近似した静的な評価値を算出することが見込まれる．

ここで，M 個のパラメータからなる重み列 $\vec{w}(w_1, w_2, w_3, \ldots w_M)$ を更新する TD(λ)-MC 法のアルゴリズムを以下の図 3，図 4 に示す．

1．学習データについて，末端局面Nを除くすべての局面$p_t\ (t\ |1 \le t \le N-1)$から静的な評価値$r_t$を得る．

$r_t = T(E(p_t))$　　p_t　：学習データ内のt手目の局面

$E(p_t)$：評価関数　$E(p_t) = \sum_{i=1}^{M} w_i x_i(p_t)$　　$x_i(p_t)$：特徴量

$T(x)$：シグモイド関数　$T(x) = 1 \Big/ 1 + e^{\frac{-x}{k}}$　（k：制御定数）

$$T'(x) = \frac{1}{k} T(x)\{1 - T(x)\}$$

2．末端局面を除くすべての局面$p_t\ (t\ |1 \le t \le N-1)$から，MC シミュレーションによって

末端局面p_t^{MC}をm個派生させ，その評価値r_t^{MC}の平均値R_tを求める．

$R_t = \frac{1}{m} \sum^{m} r_t^{MC}$　　ただし，r_t^{MC}は，$r_t^{MC} = 1$　（局面p_t^{MC}が勝ちの場合）

$= 0.5$（局面p_t^{MC}が引き分けの場合）

$= 0$　（局面p_t^{MC}が負けの場合）

のいずれかの値をとる．

3．末端局面を除くすべての局面$p_t\ (t\ |1 \le t \le N-1)$における静的な評価値$r_t$の勾配を求める．

$$\nabla_w r_t = \left(\frac{\partial}{\partial w_1} r_t, \frac{\partial}{\partial w_2} r_t, \frac{\partial}{\partial w_3} r_t, \ldots, \frac{\partial}{\partial w_M} r_t\right)$$

$$\frac{\partial}{\partial w_i} r_t = \frac{x_i(p_t)}{k} r_t (1 - r_t)$$

4．パラメータの重み列$\vec{w}(w_1, w_2, w_3, \ldots w_M)$を更新する．

$\vec{w} \leftarrow \vec{w} + \alpha(R_t - r_t) \sum_{i=1}^{t} \lambda^{t-i} \nabla_w r_i$　　（α：学習率　λ：正の定数）

5．1に戻る

図3　TD(λ)-MC 法アルゴリズム

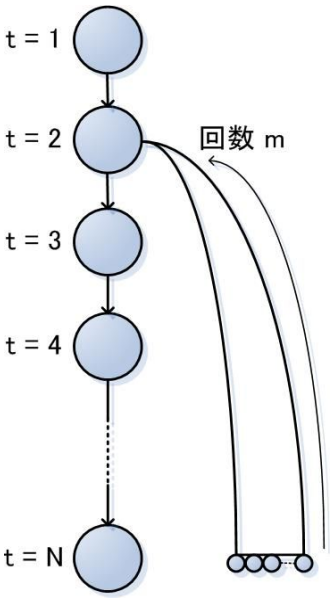

図 4　TD(λ)-MC 法の流れ

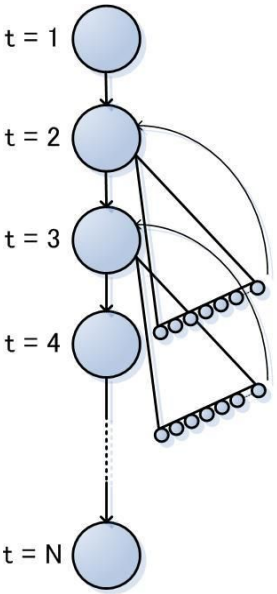

図 5　TD(λ)-leaf 法の流れ

このように，ある局面の静的な評価値とそこからＭＣシミュレーションでランダムに派生し得る末端局面の勝敗(勝ちを 1，引き分けを 0.5，負けを 0 とする)の平均値とを比較して学習することで，静的な評価値が MC シミュレーションで得られた末端局面の平均値に近似することになる．一方で，図 5 で示すように J.Baxter らによる TD(λ)-leaf 法では，Minimax 探索によって得られた動的な値を比較して学習を行ってはいるが，当然ながらその探索深度には限界があるため，手数tが末端局面の深度Nの近くでない限り，末端の勝敗という確かな報酬を得られにくい．

こうして対局ごとに評価関数内のパラメータの重み列$\vec{w}$が更新されるので，同じ思考ルーチンを持ったプログラム同士が自己対戦したところで，毎回同じ棋譜が生まれるということはなく，$\vec{w}$が収束するまで学習を行うことが可能である．

また TD(λ)-MC 法では，現在の観測局面と過去の観測局面との関連性を適格度λという過去への依存係数で表している．例えば，手数$t=3$においては以下のように重み列$\vec{w}$が更新される，

$$\vec{w} \leftarrow \vec{w} + \alpha(R_t - r_t)\sum_{i=1}^{t} \lambda^{t-i} \nabla_w r_i$$

$$\vec{w} \leftarrow \vec{w} + \alpha(R_3 - r_3)\sum_{i=1}^{3} \lambda^{3-i} \nabla_w r_i$$

$$= \vec{w} + \alpha(R_3 - r_3)\begin{Bmatrix} +\lambda^2 \nabla_w r_1 \\ +\lambda^1 \nabla_w r_2 \\ +\lambda^0 \nabla_w r_3 \end{Bmatrix}$$

適格度λは$0 \le \lambda \le 1$の定数である故，べき乗は 1 以下となり逓減していくのでこのように手数$t=3$の場合も過去に遡るほど勾配にかかる係数が小さくなっていることがわかる．これは学習の更新式が局面の末端に近づくにつれ，報酬 に信頼性が増すからである．

4.　　実験

学習成果の比較実験を以下の手順で行った.

まず，学習元となるブロックスデュオの思考ルーチンを設計した．この際，評価関数のパラメータには，盤上の駒の利き[6]や累計面積，可能手などを取り入れ，これらの評価対象となる各特徴量に重みを著者の判断で掛け，評価関数を構成した.

次に，学習元の評価関数から TD(λ)法及び TD(λ)-MC 法による学習を行った. またこの時, TD(λ)法と TD(λ)-MC 法の学習量を等しくさせるために，双方の更新回数を 10000 回として，図 3 における MC 法によって派生させる局面数 m を 200 とした．これは学習量のみならず学習にかける時間の差を極力小さくするためである.

こうして，自作のブロックスデュオの環境下にて，先手後手の全ての組み合わせ 6 種類を各 500 対局行うことで以上の三者の優劣を測った.

4.1　　実験結果

以下に TD 法及び TD(λ)-MC 法による「次の可能手」「駒の利き」パラメータの学習曲線を示す．この「次の可能手」とは駒を置いた直後の可能手の総数であり，この数が多ければ多いほど，分岐数が増えることになる.

また「駒の利き」とは，まだ駒の置かれていない盤上の空マスに対しての空間的な有利度を数値化したものであり，この値が多いほど対局を有利に展開することができることになる.

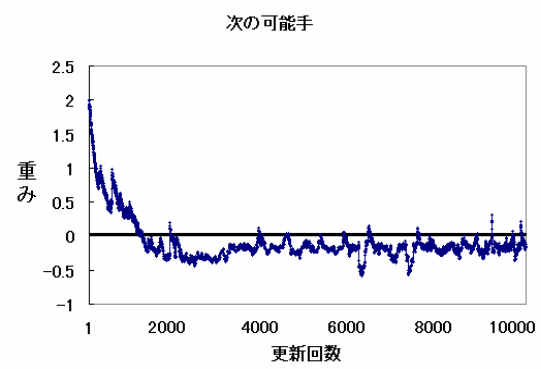

図 6　TD(λ)法による次の可能手の学習曲線

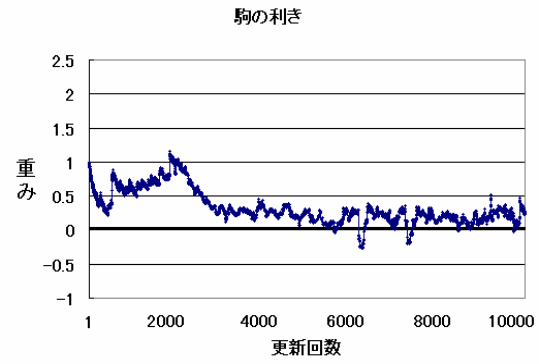

図 7　TD(λ)法による駒の利きの学習曲線

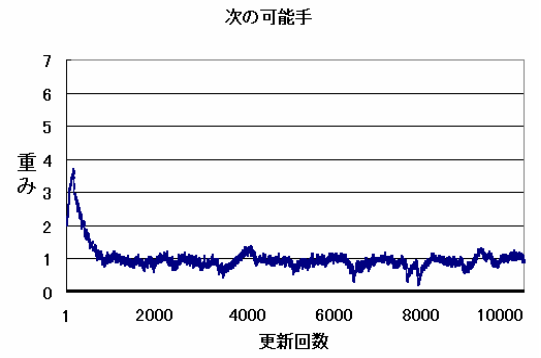

図 8　TD(λ)-MC 法による次の可能手の学習曲線

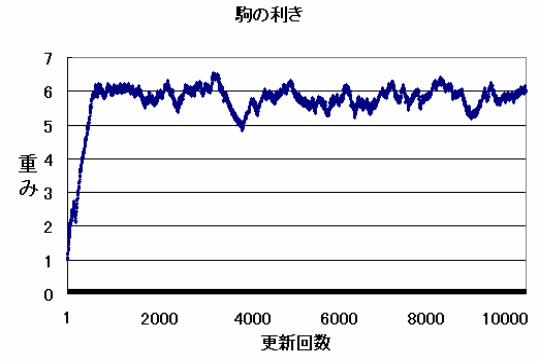

図 9　TD(λ)-MC 法による駒の利きの学習曲線

前頁の図 6 から図 9 より，二つのパラメータが徐々に収束していることがわかる．またこの時，TD(λ)法と比較して TD(λ)-MC 法の方がより収束が早いことがわかった．その他にも「駒の利き」に関しては，TD(λ)法より TD(λ)-MC 法の方が収束振動の振れ幅が大きいという特徴があった．

続いて，学習前の思考ルーチンと TD(λ)法，TD(λ)-MC 法による学習後の思考ルーチンの三者の対局結果を以下の表 1 に示す．

表 1 各思考ルーチンの 500 対局の結果

先手＼後手		TD(λ)-MC法	TD(λ)法	学習前
TD(λ)-MC法	先勝	59.2%	55.8%	74.8%
	引分	3.6%	3.0%	1.4%
	後勝	37.2%	41.2%	23.8%
TD(λ)法	先勝	52.2%	62.0%	60.6%
	引分	3.0%	4.0%	1.4%
	後勝	44.8%	34.0%	38.0%
学習前	先勝	41.4%	48.8%	51.0%
	引分	1.2%	8.2%	3.6%
	後勝	57.4%	44.2%	45.4%

上記の表 1 の対局結果から，TD(λ)-MC 法による思考ルーチンが学習前の思考ルーチンと比較して，先手後手共に勝ち越していることがわかる．特に，TD(λ)-MC 法による思考ルーチンが先手の場合は，従来の思考ルーチンと比較して，十分に勝ち越していることがわかる．

加えて僅かではあるが TD(λ)-MC 法による思考ルーチンが TD(λ)法による思考ルーチンに勝ち越していることがわかる．これは直接対局の結果からのみならず，お互いに学習前の思考ルーチンとの対局結果から推察されることである．

尚，この学習前の思考ルーチンはランダムに駒を置く思考ルーチンには先手後手共に全勝できる棋力を持っている．

これらの結果から，新しい学習アルゴリズムである TD(λ)-MC 法はブロックスデュオの環境下において学習成果が挙げられていることがわかった．

5. おわりに

TD(λ)法は，いかなる思考ゲームにおいても一律に高い学習成果が挙げられるわけではなく，やはり各々の環境やモデルに見合った学習法が存在するのである．

本稿では TD 学習アルゴリズムの拡張を目的として，TD(λ)-MC 法を提案した．今回，TD(λ)法，学習前の思考ルーチンとの比較実験を経て，TD(λ)-MC 法によって学習されたブロックスデュオのコンピュータの思考ルーチンを搭載したプログラムは，今秋 10 月末の GPCC 主催のブロックスデ

ュオの大会への参加が決まっているので，その大会で挙げられた成果と改善点を追って報告する．

また，今回の比較実験では TD(λ)-leaf 法による思考ルーチンとの優劣を測ることができなかったことを今後の課題に挙げる．

今後の展望として，TD(λ)-leaf 法には更なる拡張の余地があると思われる．例えば，TD(λ)-leaf 法を更に MC 法によって拡張した場合，従来ならば探索深度に限界のあった Minimax 法によって得られる動的な評価値も，MC 法ならば回数をこなせばより正確な動的な評価値が得られると思われる．

これを TD(λ)-leaf-MC 学習アルゴリズムと名付け，以下図 10 に示す．

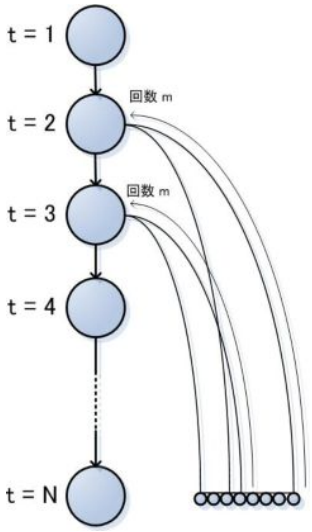

図 10　TD(λ)-leaf-MC 法の流れ

$$\vec{w} \leftarrow \vec{w} + \alpha(R_{t+1} - R_t)\sum_{i=1}^{t} \lambda^{t-i} \nabla_w r_i \quad (式\ 2)$$

(ただし，各 t における学習の更新式 2 の記号の定義は図 3 で用いたものとする．)

今後の研究の指針は TD(λ)-MC 法と TD(λ)-leaf-MC 法とを比較することによって，TD 法の更なる拡張，発展のために貢献することである．

6.　参考文献

[1]Richaed S Sutton and Andrew G.Barto 三上 貞芳・皆川 雅章共訳: 強化学習,森北出版株式会社,2000

[2]保木 邦人:局面評価の学習を目指した探索結果の最適制御,GPW 2006,pp.78-83,東北大学大学院理学研究科化学専攻

[3]J.Baxter,A.Tridgell,andL.Weaver:"Experiments In Parameter Learning Using Temporal Difference",ICGA Journal,June 1998,pp.84-99,Canberra,Australia

[4]D.F.Beal, and M.C.Smith:Temporal Difference learining for heuristic search and game playing,ICGA Journal,Vol22, No4,pp.223-235,1999

[5]薄井克俊 ： TD 法を用いたプロの将棋からの評価関数の効果的な学習,東京農工大学修士論文,1999,東京農工大学大学院工学研究科電子情報工学専攻

[6]木戸間 周平・前田 典男:数値的な特徴にもとづく囲碁局面の解析,GPW 2001, pp.124-131,東京電機大学大学院理工学科情報システム工学科