Vol.2012-GI-27 No.1
2012/3/2

コンピュータ大貧民に対する差分学習法の応用

小沼 啓[†1] 本多武尊[†2,†3]
保木邦仁[†4] 西野哲朗[†1]

近年, コンピュータ大貧民においてモンテカルロ法 (MC 法) が注目されている. この方法は強化学習における主要な一手法であり, 疑似乱数を用いて多様なプレイアウトを行い, ゲーム終了時点での成績を統計処理して予測精度を向上させる. 本研究では, 終了時点での成績だけではなく, プレイアウト途中の優劣評価も利用した MC 法の性能改善を目指す. 提案手法は, 遠い未来の影響は減じた統計処理により予測精度を向上させる. これは, 強化学習の分野で成功を収めている Temporal difference (TD) 学習から得た着想である. 我々はこのアルゴリズムに基づく実装を開発し, 第 6 回 UEC コンピュータ大貧民大会 (UECda-2011) で優勝した.

Application of Tempolal Difference Learning to Computer Daihinmin

Satoshi Konuma,[†1] Takeru Honda,[†2,†3]
Kunihito Hoki[†4] and Tetsuro Nishino[†1]

In recent computer Daihinmin (a Japanese shedding-type card game), attention is being paid to Monte-Calro (MC) methods. The MC method is often used to solve reinforcement learning problems. Here, diversified playouts using pseudorandom numbers are generated, and game results in the playouts are statistically processed for a better prediction. In this paper, we enhance the performance of the MC method by utilizing evaluations of game states still in progress. The proposed method is designed by using an idea provided from temporal difference (TD) learning, one of the successful methods in reinforcement learning. That is, our method discounts distant-future influences from the statistical processing for a better prediction. An implementation based on our method won the first prize in the 6th UEC computer Daihinmin championship (UECda-2011).

1. はじめに

囲碁のような 2 人完全情報ゲームに強化学習を用いる際, モンテカルロ法 (以下 MC 法とする) を応用したモンテカルロ木探索が良く用いられている. その切っ掛けとなったのは, 囲碁のプレイヤープログラムにモンテカルロ木探索を用いることでプロ棋士に勝ったことである. そして, 完全情報 2 人ゲームに関するモンテカルロ木探索・MC 法の研究がよく行われ, 強い囲碁プログラムにはこれらのアルゴリズムが組み込まれていることが多い[1)].

一方, 不完全情報多人数ゲームである” 大貧民” のプレイヤープログラムの強さを競う UEC コンピュータ大貧民大会が毎年開催されている. この大会を契機として様々なプログラムが実装され, 研究が行われている[2)]. しかし, 大貧民にモンテカルロ木探索をそのまま用いることは出来なかった. モンテカルロ木探索に用いられるゲーム木は, 完全情報ゲームのようにゲームの状態が全て開示されていなければ描けないからである.

そこで須藤らは, ランダムサンプリングした環境モデルを疑似的に作成することで, 不完全情報多人数ゲームである大貧民に応用した[3)]. 具体的には, 相手の手札をランダムに想定しゲーム木を構成した. また, 大貧民は多人数で行うゲームであるため, モンテカルロ木探索のように深く木を拡張させてしまうと, 2 人ゲームに比べてゲーム木が巨大になってしまう. そのため, 須藤らはモンテカルロ木探索ではなく単純な MC 法を用いた. MC 法は合法手の評価値により木を深く探索させない点がモンテカルロ木探索と異なり, プレイアウト・評価値の決定等の根本的な差違は存在しない. この手法を用いたプレイヤープログラムは第 4 回, 第 5 回 UEC コンピュータ大貧民大会 (UECda-2009, UECda-2010) において優勝したため, 不完全情報多人数ゲームに MC 法が有効であると考えられてきた.

本研究では, より適切な行動選択を行い, より強くなることを目指して, TD 学習に基づいた差分学習法を考案した. そして, その効果や MC 法との違いについて調べた.

†1 電気通信大学大学院 情報理工学研究科 総合情報学専攻
Department of Informatics, Faculty of Informatics and Engineering, The University of Electro-Communications
†2 理化学研究所 脳科学総合研究センター 運動学習制御研究チーム
Laboratory for Motor Learning Control, RIKEN Brain Science Institute
†3 日本学術振興会 特別研究員 (PD)
Research Fellow of Japan Society for the Promotion of Science (PD)
†4 電気通信大学 先端領域教育研究センター
The center for Frontier Science and Engineering, The University of Electro-Communications

Vol.2012-GI-27 No.1
2012/3/2

2. アルゴリズム

2.1 大貧民における MC 法の概要

須藤らは, 不完全情報多人数ゲームである大貧民に対して, 単純な MC 法を用いたプレイヤープログラムを実装した[3), 4)]. このプログラムは UEC コンピュータ大貧民大会公式ウェブサイト[5)] からダウンロード可能である. しかし, 文献[3), 4)] からは読み取れない部分が多いので, 改めてアルゴリズムを説明する (図 1). ここで用いる言葉の諸定義については, 文献[6), 7), 8)] に従う.

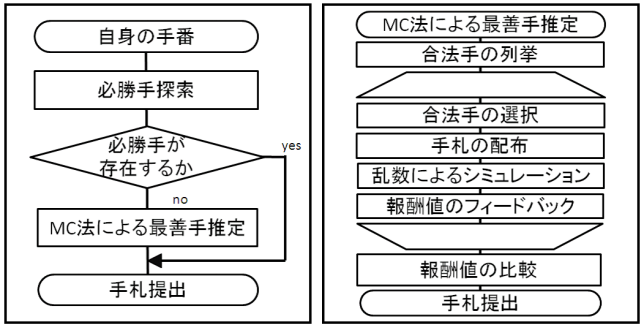

図 1 大貧民における MC 法の概略図.

図 1 をより詳細に書き下すと以下のような手順となる.

1. 合法手の列挙 自身が選択可能な合法手を列挙する

2. 以下の手順を複数回行う

- **a. 合法手の選択** 列挙した内, 1 つの合法手 i を選び, その行動選択にしたがってゲームの状態を遷移させる
- **b. 手札の配布** 乱数を用いて相手にカードをランダムに割り当てる
- **c. 乱数によるシミュレーション** 次のプレイヤーの合法手を列挙しその内 1 つをランダムに選択する. その行動選択にしたがってゲームの状態を遷移させる. 自身のカードがなくなるか, 相手全員がカードを提出し終え, ゲームが終了するまで全てのプレイヤーに順番に行動選択をランダムに行わせる
- **e. 報酬値のフィードバック** ゲームが終了したら, 報酬値 R（順位）の大きさを調べる. そして, 最初に選択した合法手 i に報酬値を与え, 合法手の評価 $\overline{X_i}$ を更新する (式 (2), (3), (4))

3. 報酬値の比較 複数回のプレイアウトを行った後, 各合法手 i に対する $\overline{X_i}$ の大小を比較し, $\overline{X_i}$ が最大の合法手を最善手と推定する

このように, 最終状態における報酬値のみを用いて, 現在の盤面における合法手の評価値を決定している. どの合法手に対してプレイアウトを行うのかについては UCB1-TUNED[6)] を用いており, 合法手の評価値と選択回数から最善手である確率が最も高いと判断された合法手を優先して選択している.

2.2 提案アルゴリズム

須藤らが用いた MC 法は, プレイアウトの途中における盤面を評価せず, 報酬値のフィードバックを行う. したがって, プレイアウトに選択した行動選択 i の評価の更新値 V は, 報酬値 R を用いて式 (1) として表される (図 2 上).

$$V = R \tag{1}$$

そして, 以下の式 (2), (3), (4) を用いて, 最初に選択した合法手 i に関して, 総報酬値 X_i, 選択した回数 n_i, 平均報酬値 $\overline{X_i}$ の更新を行う.

$$X_i \longleftarrow X_i + V \tag{2}$$

$$n_i \longleftarrow n_i + 1 \tag{3}$$

$$\overline{X_i} \longleftarrow X_i / n_i \tag{4}$$

このように, MC 法では, あらゆるゲームの盤面において, ゲームの結果のみを考慮して行動選択を行う. ここで, 途中経過も含めて強化学習させることにより, MC 法よりも適切な行動を選択するプログラムを実装できるのではないかと考えた. そこで, TD 学習に基づき式 (1) を式 (5) のように拡張し, 報酬値のフィードバックについて改良した.

$$V = \sum_{t=0}^{N-1} r_t \omega_t \tag{5}$$

すなわち, プレイアウト中に生成された盤面状態 t において, 盤面の評価値 r_t を算出する. そして, その盤面における評価の重み ω_t に従い, 最初に選択した行動選択 i に逐次的なフィードバックを行わせる (図 2 下). プレイアウトが N 回の状態遷移で表されるとき, $r_{N-1} = R$ である. 大貧民では, 相手の手番では行動選択を行えないため, 自身の手番が回ってくるまでを 1 ターンとしてフィードバックさせた.

次に, r_t や ω_t をどのように求めるかが問題である. 大貧民は多人数で行われるゲームである

Vol.2012-GI-27 No.1
2012/3/2

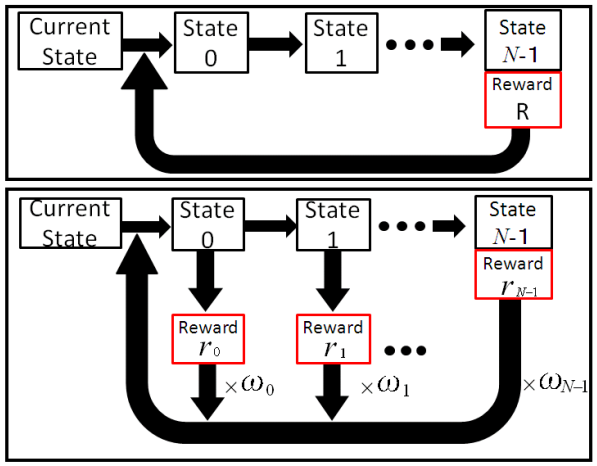

図 2 MC 法と提案アルゴリズムの比較図.

ため, どこで報酬値の差分が発生したのかを知ることが可能であると考えた. そこで, 各盤面における" ゲームに参加している人数" を評価値として用いた. これによりプレイアウトがどのような状態遷移で構成されているかを知ることができる. 具体的には, "5 人が競り合い惜しくも 1 位を逃がして 2 位で上ったプレイアウト" では, 評価値が "$5 \longrightarrow 5 \longrightarrow \cdots \longrightarrow 5 \longrightarrow 4$" という遷移となる. 一方, "1 人にいち早く上られてしまった後, 4 人で競い合い 2 位で上ったプレイアウト" では, 評価値が "$5 \longrightarrow 4 \longrightarrow 4 \longrightarrow \cdots \longrightarrow 4 \longrightarrow 4$" という遷移となる.

MC 法では, これらのプレイアウトは同じ報酬値となる. しかし, 前者のプレイアウトは何らかの要因があれば, 自身が 1 位で上がれる可能性があるが, 後者のプレイアウトは 1 位で上れる可能性は低い. そこで前者が後者よりも優れたプレイアウトであるようフィードバックさせるために式 (6) を考案した.

$$\begin{aligned} V(0) &= r_0 \\ V(t) &= V(t-1) + \alpha(r_t - V(t-1)) \\ &= (1-\alpha)V(t-1) + \alpha r_t \qquad (t>0) \end{aligned} \tag{6}$$

これにより, ターン t までのプレイアウトの評価値 $V(t)$ を, ターン $t-1$ までの評価値 $V(t-1)$ とそのターンの評価値 r_t を用いて求める. そのため, プレイアウトにおける状態遷移回数 N によらず, 1 つのパラメータ α で各ターンにおける評価の重み ω_t を制御できる.

以下に, α による ω_t の導出を示す.

$$\begin{aligned} V(N-1) &= (1-\alpha)V(N-2) + \alpha r_{N-1} \\ &= (1-\alpha)\{(1-\alpha)V(N-3) + \alpha r_{N-2}\} + \alpha r_{N-1} \\ &= (1-\alpha)\{(1-\alpha)\{(1-\alpha)V(N-4) + \alpha r_{N-3}\} + \alpha r_{N-2}\} + \alpha r_{N-1} \\ &\vdots \\ &= \sum_{t=0}^{N-1} r_t \omega_t \end{aligned} \tag{7}$$

そして, 係数を比較することによりステップ t における評価値の重み ω_t は, $\alpha \neq 1$ のとき式 (8), (9) で表される.

$$\omega_0 = (1-\alpha)^{N-1} \qquad (t=0, \alpha \neq 1) \tag{8}$$

$$\omega_t = \alpha(1-\alpha)^{N-1-t} \qquad (t>0, \alpha \neq 1) \tag{9}$$

また, $\alpha = 1$ のとき式 (10), (11) で表される.

$$\omega_t = 0 \qquad (0 \leq t < N-1, \alpha = 1) \tag{10}$$

$$\omega_{N-1} = 1 \qquad (t = N-1, \alpha = 1) \tag{11}$$

以上より, $\alpha = 1$ とすると MC 法と同じように終端状態のみを学習し, α の値を 1 より小さくするほど遷移回数が少ない非終端状態の評価の重み ω_t を大きくさせてプレイアウトにおける一連の状態遷移を学習する.

3. 結　　果

提案アルゴリズムを実装したプレイヤープログラム "Crow" で, 第 6 回 UEC コンピュータ大貧民大会に出場し優勝した. このとき, "Crow" 以外の決勝に残った 4 つのプログラムは, 須藤らのプログラムを改良した, MC 法を用いるプレイヤープログラムであった.

"Crow" の強さを検証するため, 対戦による実験を行った. まずはじめに, "Crow" のパラメータ α を MC 法を用いた前大会優勝プログラム "snowl" と同じ動作を行うよう $\alpha = 1.0$ と設定し, "Crow" と "snowl" の強さを比較した. そこで, UEC コンピュータ大貧民大会の公式ルール[5] に基づき 1000 ゲームの対戦させる実験を 100 回行った. また, 5 つのクライアントで対戦が行わなわれることから前々大会優勝クライアントを 3 つ (a, b, c) を用いた. 比較のため, 昨年の優勝プログラムのプレイアウト回数に合わせて, 5 つ全てのプログラムのプレイアウト回数を 2000 回とした. その結果, "Crow" の 49 勝 50 敗 1 引き分けの結果となった (図 3 左). これにより, 提案手法が MC 法と同じ動作を行っていることが示唆さ

れた.

次に, ”Crow” をプレイアウトにおける途中経過を含めて強化学習を行うようパラメータの値を $\alpha = 0.9$ として, 同様の対戦させる実験を行った. この結果, ”Crow” の 54 勝 46 敗となった (図 3 右). この結果より, 差分学習法を用いた提案アルゴリズムが有効であることが示唆された.

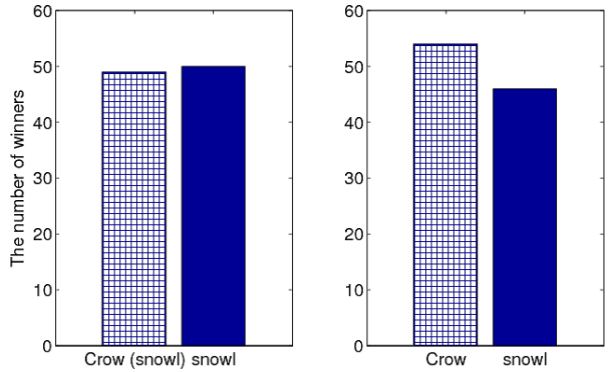

図 3 対戦による強さの比較実験結果.

最後に, 提案アルゴリズムと MC 法との間にどのような違いがあるのかを調べるため, 同じ盤面ごとに提案アルゴリズムと MC 法でどのような最善手推定をするか調べた. 具体的に, 手札 3 枚で合法手数が 3 手であるとき, それぞれのアルゴリズムで 2000 回のプレイアウトで行う最善手推定を 1000 回行わせた. このとき, 提案手法のパラメータ値は $\alpha = 0.9$ とした. 図 4 上は各合法手についてプレイアウトで選択された回数を表し, 図 4 下は各合法手について最善手推定で選択された回数を表している. ”Crow” は合法手 1 に多くのプレイアウトを割り当て, そして合法手 1 が最善手であると多く推定している. 一方, ”snowl” は合法手 1, 2 に同程度のプレイアウトを割り当てているものの, 合法手 2 が最善手であると多く推定している (図 4). 以上のことから, MC 法を提案アルゴリズムに拡張することにより, プレイアウトの選択に差異が生じることが示された.

4. おわりに

不完全情報多人数ゲームである大貧民に対して, MC 法を TD 学習の考えに基づき, 差分をとり最善手を推定するアルゴリズムに拡張した. このアルゴリズムは従来手法である MC 法より強いプレイヤープログラムとなり得ることが示された. また実験より, これらのアルゴリズムは手札枚数が少くプレイアウトが短かい状態遷移で表されるにもかかわらず, 異なる合法手を最善手としたため, 根本的な差違があることが示された. 今後の課題としては, この差違と” 大貧民というゲームにおける強さ” との関連性を示すことがあげられる.

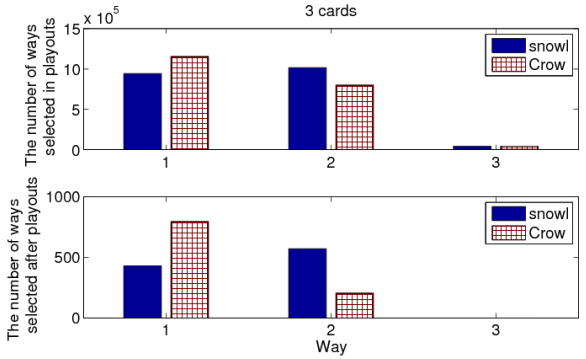

図 4 合法手に対するプレイアウト総数と最善手推定回数.

参 考 文 献

1) 美添 一樹, 村松 正和, ” コンピュータ囲碁の飛躍の背景”, 数学セミナー, pp.52-57, 2010.
2) 小沼 啓, 西野 哲朗, ” コンピュータ大貧民に対するモンテカルロ法の適用”, 情報処理学会研究報告, 第 25 回ゲーム情報学研究発表会, Vol.2011-GI-25 No.3.
3) 須藤 郁弥, 篠原 歩, ” モンテカルロ法を用いたコンピュータ大貧民の思考ルーチン設計”, http://uecda.nishino-lab.jp/2009/.
4) 須藤 郁弥, 篠原 歩, ”UEC コンピュータ大貧民大会向けクライアント「snowl」の開発”, http://uecda.nishino-lab.jp/sympo/.
5) 電気通信大学, ”UEC コンピュータ大貧民大会”, http://uecda.nishino-lab.jp/.
6) Peter Auer, Nicolo Cesa-Bianchi and Paul Fischer, ”Finite-time Analysis of the Multiarmed Bandit Problem”, Machine Learning, 47:pp.235-256, 2002.
7) Richard S. Sutton and Andrew G. Barto 原著, 三上 貞芳, 皆川 雅章 共訳, ” 強化学習”, 森北出版, 2000.
8) 電気学会 著, “学習とそのアルゴリズム”, 森北出版, 2002.