Vol.2011-GI-25 No.3
2011/3/5

コンピュータ大貧民に対するモンテカルロ法の適用

小沼 啓[†1] 西野哲朗[†2]

モンテカルロ法による最善手の推定に ϵ-GREEDY 法を用いた, コンピュータ大貧民のプレイヤープログラムを開発した. さらに, 最善手を得る確率を高めるために推定回数を増やす手法を採用し拡張した. この手法は, 従来のプレイヤープログラムより強いことが示された.

Application of Monte Carlo Method to Computer Daihinmin

Satoshi Konuma[†1] and Tetsuro Nishino[†2]

In this paper, we propose ϵ-GREEDY method for finding the best legal move in computer Daihinmin while applying the Monte Carlo method. Furthermore, we increase the number of estimation of our ϵ-GREEDY method. Our result suggests that our method is the strongest among computer Daihinmin methods.

1. はじめに

完全情報 2 人ゲームの 1 つである囲碁では, Crazy Stone[1)] の登場によりプレイヤープログラムの強さが飛躍的に向上した. Crazy Stone は探索木にモンテカルロ法を用いる手法を用いたプレイヤープログラムである. Crazy Stone と同じように, 探索木にモンテカルロ法を用いる手法を採用しているプレイヤープログラム MoGo[1)] は, 囲碁のプロ棋士に勝利した.

そのため, モンテカルロ法が他のゲームのプレイヤープログラムへ応用されることが期待されている. プレイヤープログラムが人間に完全情報 2 人ゲームで勝つことができるようになったが, 不完全情報多人数ゲームでは, 未だ人間に勝つようなプレイヤープログラムは開発されていない.

不完全情報多人数ゲームの 1 つであるトランプゲーム大貧民で, 須藤らは, モンテカルロ法と, その制御に UCB1-TUNED と呼ばれるアルゴリズムを用いた [2)]. そして, 第 4 回 UEC コンピュータ大貧民大会 (UECda-2009) において優勝を収めている.

須藤らが用いた UCB1-TUNED とは多腕バンディット問題[3)] を解決するためのアルゴリズムである. この UCB1-TUNED とは異なる考えに基づいた多腕バンディット問題を解決するためのアルゴリズムに, ϵ-GREEDY と呼ばれるアルゴリズムがある[3)].

そこで本研究では, コンピュータ大貧民に対して, モンテカルロ法の制御に ϵ-GREEDY を用いて, 須藤らのプレイヤープログラムより強いプレイヤープログラムを提案することを目的とする. なお本稿では, 用語の諸定義については文献[3)], [4)], [5)] に従う.

2. 提案アルゴリズム

2.1 ϵ-GREEDY

多腕バンディット問題[3)] を解決するためのアルゴリズムの 1 つである ϵ-GREEDY をモンテカルロ法におけるプレイアウトの制御アルゴリズムとして用いることを提案する. 具体的には, ϵ-GREEDY により n 回目のプレイアウトを行う合法手の選択を以下のように行う.

(1) 確率 $1-\epsilon_n$ で, $n-1$ 回目までの平均報酬値 $\overline{X_i}$ が最も大きい合法手を選択
(2) 確率 ϵ_n で全 K 個の合法手の中からランダムに選択

ここで, ϵ_n は式 (1) で計算される.

$$\epsilon_n = \min\{1, \frac{cK}{d^2 n}\} \quad (0 \leq \epsilon_n \leq 1) \tag{1}$$

このように ϵ-GREEDY を用いることにより, n 回目のプレイアウトを行う合法手を選択する際, $\overline{X_i}$ が最も大きい合法手を選択する確率は $1-\epsilon+\frac{\epsilon}{K}$ となる. そして, 他の合法手を選択する確率はそれぞれ $\frac{\epsilon}{K}$ となる. これらのことから, 局面において最善手である可能性が大きい合法手に, 多くのプレイアウトを行うことができる. 一方, 一定の確率を他の合法手にも割り当てることで, 各合法手に対しても探査を行うことができる.

ここで ϵ-GREEDY について解析を行った結果, c と d の 2 つのパラメータのうち, c を ϵ_n が 1 より小さくなる起点となる n を求めるパラメータとした. 具体的には, $c=\frac{N}{3K}$ とすると

†1 電気通信大学 情報通信工学科
Information and Communication Engineering, The University of Electro-Communications
†2 電気通信大学 総合情報学科
Departmant of Informatics, The University of Electro-Communications

© 2011 Information Processing Society of Japan

Vol.2011-GI-25 No.3
2011/3/5

$$\epsilon_n = \frac{\frac{N}{3K}K}{d^2 n} = \frac{N}{3d^2 n} \quad (2)$$

となる. したがって, $d = 1$ とすると $n = \frac{1}{3N}$ のとき $\epsilon_n = 1$ となる. そして, n が $\frac{1}{3N}$ よりも大きくなると, ϵ_n の値が 1 より徐々に小さくなる. そのため, n が $\frac{1}{3N}$ よりも大きくなるとともに, $\overline{X_i}$ が最も大きい合法手を選択する確率 $1 - \epsilon_n$ が 0 より徐々に大きくなる. 以上より, c の値を変更することにより, $\overline{X_i}$ が最も大きい合法手を選択する確率が生まれる起点を変更することができる. これにより探査と得られた情報の利用の重み, すなわちランダムに合法手を選択し新たな情報を得ることと, 得た情報を利用し報酬値が最大である合法手を選択することとの割合を変化させることができる. 例として, $c = \frac{N}{3K}$, $d = 1$, $N = 1500$ とすると ϵ_n は図 1 のようになる.

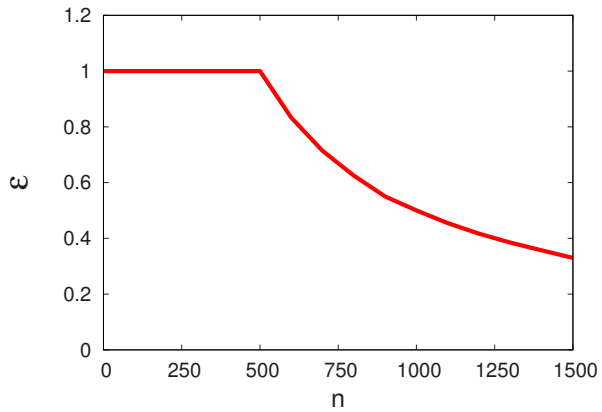

図 1 ϵ_n の変化.

また, 極端な例として, $c = \frac{N}{K}$ とするとプレイアウトを行う合法手をランダムにしか選択できなくなるよう定めることができ, $c = 0$ とすると $\overline{X_i}$ が最も大きい合法手しか選択できなくなるよう定めることもできる (貪欲法). このように, ϵ-GREEDY におけるパラメータ c を調節することで, 対象となる問題に適した, 探査し情報を得ることと, 得た情報を利用することとの割合を求められるのではないかと考えられる.

また, パラメータ d を変更することで曲線部分の曲がり具合に緩急をつけることができる. しかし, ϵ-GREEDY のパラメータによる性能差は, d よりも c に因るところが大きいと考え, c のパラメータの変更を優先した. そのため, 本研究では $d = 1$ とした. なお, モンテカルロ法による合法手の探査に ϵ-GREEDY を用いたコンピュータ大貧民のプレイヤープログラムを本研究では epsilon と呼ぶ.

2.2 成功確率の増幅

epsilon により, ゲームにおける各局面の最善手を導き出す際には, プレイアウトを行う回数を増やし, ゲームルールに反しない限り, シミュレーションに時間を費やすことが理想である. しかし, 乱数を用いた制御アルゴリズムでは, 同じ局面に対して同じ合法手を最善手として推定するとは限らない. すなわち, ある局面に対して K 個の合法手の中に最善手が 1 つのみ存在すると仮定すると, 局面における最善手を推定する確率 (成功確率) が $\frac{1}{K}$ である可能性も考えられる.

そこで, epsilon による最善手の推定を 5 回行わせ, そのうち最善手と推定された回数の最も多い合法手を最善手とすることで, 成功確率を増幅させる手法を提案する. 具体的には, α という合法手を最善手として 3 回推定し, β という合法手を最善手として 2 回推定した場合, 推定結果を α とする. この手法は乱数を用いたアルゴリズムにおける成功確率を上げる基本的な考えであり, 成功確率が $\frac{1}{2}$ よりも大きければ成功確率を飛躍的に高められることが知られている[6]. このように実装したコンピュータ大貧民のプレイヤープログラムを本研究では majority と呼ぶ. また, このように最善手を推定しても, 最終的な最善手を求めた際のプレイアウト回数は epsilon, majority 共に N であるため, 強さを直接比較することできる.

3. 比較プレイヤープログラム

3.1 プレイアウト選択時におけるランダムサンプリング

epsilon は, ϵ_n を変化させてプレイアウトを行う合法手を選択するプレイヤープログラムである. しかし, epsilon が, 局面における最善手を推定する上で有用であるかは分からない. そこで, 全 K 個の合法手の中から合法手をランダムに選択し, プレイアウトを行う合法手を選択する場合 (ランダムサンプリング) と epsilon との強さの比較を行うこととした. これにより, ϵ-GREEDY の有用性を検証できるのではないかと思われる. なお, このように実装したコンピュータ大貧民のプレイヤープログラムを本研究では random-sampling と呼ぶ.

3.2 プレイアウト選択時における soft max 関数の応用

epsilon では $\overline{X_i}$ が最大の合法手のみに, 他の合法手よりも大きな確率 $1 - \epsilon + \frac{\epsilon}{K}$ が与え

© 2011 Information Processing Society of Japan

られる. そして, 他の合法手に関しては, $\overline{X_i}$ がどの程度小さいかは考慮されず, 確率 $\frac{\epsilon}{K}$ が与えられる. そこで, $\overline{X_i}$ の大きさに応じて, 合法手 i を選択する確率 $P(i)$ を与えるために式 (3) で表される soft max と呼ばれる関数を導入し比較を行うこととした. soft max 関数を用いて実装したコンピュータ大貧民のプレイヤープログラムを本研究では soft-max と呼ぶ.

$$P(i) = \frac{e^{\frac{\overline{X_i}}{r}}}{\sum_{j=1}^{K} e^{\frac{\overline{X_j}}{r}}} \tag{3}$$

soft max 関数で用いられている r は正定数のパラメータである. この soft max 関数は, $r \to 0$ の極限では貪欲法と動作が一致し, $r \to \infty$ の極限ではランダムサンプリングと動作が一致することが知られている[7].

UEC コンピュータ大貧民大会の公式ルールでは, 1 位から順に 1 試合につき 5, 4, 3, 2, 1 点が与えられると定めている (プレイヤー数は 5)[8]. 本研究では, このルールに基づき, $\overline{X_i}$ を 1 以上から 5 以下の実数とした. よって, パラメータ r は 0 より大きく, 5 以下の実数とした. 以上より, $r \to 0$ で貪欲法と同じ動作をし, $r = 5$ でランダムサンプリングに近い動作をすると考えられる.

ここで, ϵ-GREEDY と soft max 関数のどちらが優れているかは, 対象となる問題と密接に関わっているとされ, いまだ詳しくは知られていない[7]. したがって, epsilon と soft-max の強さを比較することで, ϵ-GREEDY と soft max 関数のどちらがコンピュータ大貧民に対してモンテカルロ法を用いる際に, 有効であるのかを調べることができる.

4. 対戦による強さの比較実験

epsilon におけるパラメータである c と soft-max におけるパラメータである r を変化させ, パラメータの変化が強さとどのような関係があるかを調べた. なお, 5 つのプログラムでゲームを行わなければならないため,

- 4 つの UCB1-T との対戦
- 4 つの random-sampling との対戦
- 4 つの default(UECda 標準プレイヤープログラム) との対戦

を行った. ここで, UCB1-T は先行研究[2] により実装された, モンテカルロ法に UCB1-TUNED を応用したプレイヤープログラムである. そして, これらのプレイヤープログラムからどれだけの点数を獲得できるかで epsilon と soft-max における最も適したパラメータを調べた. なお, プレイアウトを行う総回数 N は, 先行研究において 1500 回とされていた. そこで, 全てのプレイヤープログラムにおいて $N = 1500$ として実験を行うこととした. また, 実験では UECda で用いられているルール[8] に基づき実験を行った.

その後, 最適と思われるパラメータを用いた epsilon, soft-max, majority と UCB1-T, random-sampling の 5 つのプレイヤープログラムを同時に対戦させて, 最終的な強さの比較を行った.

5. 実験結果

epsilon におけるパラメータ c を変化させ対戦させた. その結果, $c \times \frac{K}{1500} = 0$ から $c \times \frac{K}{1500} = 0.3$ にかけて獲得点数は増加した. そして, $c \times \frac{K}{1500} = 0.3$ から $c \times \frac{K}{1500} = 0.6$ の間は獲得点数が高くなった. その後, $c \times \frac{K}{1500} = 1$ に近づくにつれ獲得点数は減少した (図 2 左). また, 順位の結果も獲得点数の結果と同じように, $c \times \frac{K}{1500} = 0.3$ から $c \times \frac{K}{1500} = 0.6$ の間, 好成績を収めた (図 2 右).

次に, soft-max におけるパラメータ r を変化させ対戦させた. その結果, $r = 0$ から $r = 1$ にかけて獲得点数は増加し, その後, r の値が大きくなるにつれ徐々に獲得点数は減少した (図 3 左). また, 順位の結果も獲得点数の結果と同じように $r = 1$ の際に好成績を収めた (図 3 右).

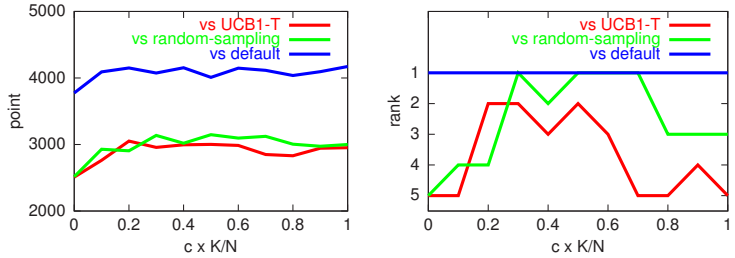

図 2 $N = 1500$ における epsilon の対戦結果.

ここで, 図 2 および図 3 の左側の図において, 横軸はパラメータの値, 縦軸は 4 つのプレイヤープログラムに対して獲得した点数を表す. また, 図 2 および図 3 の右側の図において, 横軸はパラメータの値, 縦軸は順位を表す.

以上の結果より, epsilon におけるパラメータである c の最適値は $c = \frac{1500}{2K}$ となり, soft-max におけるパラメータである r の最適値は $r = 1$ となった. これらのパラメータを用いた epsilon, soft-max, majority と UCB1-T, random-sampling を対戦させたところ, 1 位から

© 2011 Information Processing Society of Japan

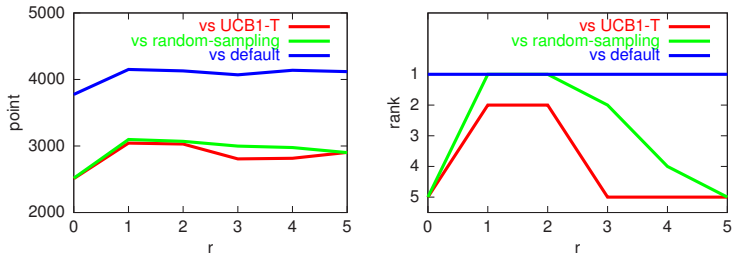

図 3 $N = 1500$ における soft-max の対戦結果.

順に majority, epsilon, UCB1-T, random-sampling, soft-max となり, majority が他のプレイヤープログラムより群を抜いて強いことが示された (図 4).

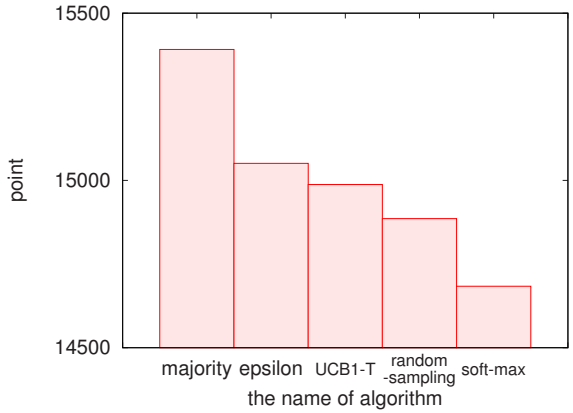

図 4 対戦結果.

6. おわりに

本研究では, モンテカルロ法におけるプレイアウトの制御に ϵ-GREEDY を用いて, コンピュータ大貧民のプレイヤープログラム epsilon を開発した. ϵ-GREEDY のパラメータを適切に選ぶことによって文献[2] のプレイヤープログラムよりも多くの点数を得ることができた. さらに, 本プレイヤープログラムは, モンテカルロ法のプレイアウトを制御する random-sampling や soft-max を用いたプレイヤープログラムよりも多くの点数を得ることができた. 以上より, コンピュータ大貧民では, モンテカルロ法のプレイアウトの制御に ϵ-GREEDY を用いることが最も有効であることが示唆された.

さらに, 本プレイヤープログラム epsilon が最善手を推定する確率 (成功確率) を高めるために, 推定回数を 5 回に増やしたプレイヤープログラム majority を開発した. コンピュータ大貧民において, majority は epsilon に勝利することができた. これは, ϵ-GREEDY は推定回数を増やすことで成功確率を高めることができることを示している.

謝　辞

本研究を進めるにあたり, 電気通信大学の田中繁特任教授, 保木邦仁特任助教, 本多武尊氏には大変貴重な御助言を賜りました. 深く感謝申し上げます.

参 考 文 献

1) 美添 一樹, “コンピュータ囲碁におけるモンテカルロ法理論編”, http://minerva.cs.uec.ac.jp/ito/entcog/contents/lecture/date/5-yoshizoe.pdf.
2) 須藤 郁弥, “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/download/suto_sym.pdf.
3) Peter Auer, Nicolo Cesa-Bianchi and Paul Fischer, “Finite-time Analysis of the Multiarmed Bandit Problem”, Machine Learning, 47:pp.235-256, 2002.
4) 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.
5) Stuart Russell, Peter Norvig 原著, 古川 康一 訳, “エージェントアプローチ人工知能”, 共立出版, 1997.
6) 玉木 久夫, “乱択アルゴリズム”, 共立出版, 2008.
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) 電気通信大学, “UEC コンピュータ大貧民大会”, http://uecda.nishino-lab.jp.

© 2011 Information Processing Society of Japan