「芝浦将棋 Softmax」のチーム紹介

2019 年 3 月 22 日
芝浦工業大学情報工学科
五十嵐治一，横田直之，吉野拓真，岩本裕大

1. はじめに

本稿は，第 29 回世界コンピュータ将棋選手権（2019 年 5 月 3 日～5 日開催）に出場予定の「芝浦将棋 Softmax」（シバウラショウギ　ソフトマックス）のアピール文書です．本チームは昨年に引き続いて 3 回目の出場です．本チームの原型は 2016 年まで出場していた「芝浦将棋 Jr.」チームですが，探索手法が従来の Min-max 探索（α β探索）とは異なる Softmax 探索である点が大きく異なります. ただし, 合法手生成までは芝浦将棋 Jr.と共通で, 選手権公式ライブラリとして登録されている「芝浦将棋 Jr.合法手生成プログラム」[1]を使用しています. 棋力的には従来のα β探索手法のチームにはまだまだ及びませんが, アルゴリズムが単純でコーディングの容易さや並列性に優れています. 以下, 簡単に本チームの特徴を紹介していきます.

2．開発メンバー

五十嵐は芝浦工業大学工学部情報工学科に勤務する教員です．横田，吉野，岩本は五十嵐研究室の大学院生と学部 4 年生(いずれも 2019 年度)です.

3.「芝浦将棋 Softmax」の特徴

本チームの特徴を，以下の１）～５）のようにまとめました．このうちの２)～４）が本チーム独自の探索方式である「MC Softmax 探索」[6]に関する説明です．この探索方式は, 文献[2][4]の研究が基になっています.

１）芝浦将棋 Jr. の合法手生成ルーチンを使用

芝浦将棋 Jr. では盤面表現のデータ構造を独自の Magic bitboard を用いて駒の利き場所での駒の配置状況などを計算しています[3]．この計算を含む合法手生成のプログラムは「芝浦将棋 Jr.合法手生成プログラム」の名称で選手権公式ライブラリとして登録されています. 芝浦将棋 Softmax はこの合法手生成プログラムをそのまま使用しています.

２）モンテカルロ・ソフトマックス探索を使用

現在のチェスや将棋のプログラムは「Min-max 探索」という探索方式をほぼ 100%採用しています．これには探索木のすべてのノードを探索する必要があります（全幅探索）が，α β カットなどの枝刈りの処理により探索にかかる計算時間を短縮しています（α β 探索）．この全幅探索に対して，そのゲーム特有の知識（ヒューリスティクス）を用いて探索するノードを限定したり，優先順位をつけて選択的に探索する「選択探索」という探索方式があります．本チームはノードの選択方式としてノード評価値の min-max 演算ではなく，確率分布に基づく選択（Softmax 探索）を使用しています．したがって，探索木をルートノード（実際の盤面の局面）から選択して降りていく（読んで行く）際には，実際にサイコロをふりながら確率的に選んで末端局面まで降りていきます．この確率的選択方式は，AlphaGo のようなコンピュータ囲碁ソフトで用いられている「モンテカルロ木探索」における決定論的な木の選択方法（UCT など）とは一線を画しており，我々は「モンテカルロ・ソフトマックス探索方式」(MC Softmax 探索方式)と呼んでいます．

３）ノードの評価関数を用いたボルツマン分布による確率的なノード選択

前項で述べた Softmax 探索には，本チームでは指し手の良さを用いたボルツマン分布を利用します．すなわち，各ノードでの指し手の選択確率を次の式で計算し，その確率に従ってノードを選択していきます．

$$\pi(a|s) = exp(E_a(a;s)/T) / \sum_{x \in A(s)} exp(E_a(x;s)/T) \qquad (1)$$

ただし，s は局面（ノード），a は指し手，$E_a(a;s)$は局面 s における指し手 a の良さですが，指した後の局面ノードの評価値 $E_s(s)$で置き換えることにします．$A(s)$は s における合法手の集合，T は温度と呼ばれているパラメータです．温度が低ければ最良優先探索に，温度が高ければランダム探索に近づきます．ノードの評価値は，探索木の末端ノード（leaf）であればそのノードの局面評価関数により計算します（実際にはそこで静止探索も行っています）．

一方，内部ノードであれば子ノード $v(x;s)$の評価値 $E_s(v)$をその子ノードの選択確率 $\pi(x|s)$で重みづけた期待値

$$E_s(s) = \sum_{x \in A(s)} \pi(x|s) E_s\big(v(x;s)\big) \qquad (2)$$

で定義します．したがって，読んだ先（子孫ノード）に評価の高い手があるような手は高く評価されます．また，十分探索が進んで探索木をすべて展開した後では，(1)で T をゼロに近づける（低温化）と，Softmax 探索による探索結果は Min-max 探索の探索結果に近づいて行きます．

４）深さ制御とバックアップ操作

MC Softmax 探索の全体の流れを図１に示します．ルートノードから，３）の選択法に従ってノードを選択し，末端ノードまで到達すると，そのノードの子ノードを一段階だけすべて展開します．展開後は新たな末端ノードの評価値を局面評価関数で計算し，その値をルートノードへ向けて(2)の計算を繰り返し，ルートノードまでの経路上のノード評価値を更新していきます．我々はこの更新操作を「バックアップ操作」と呼んでいます．

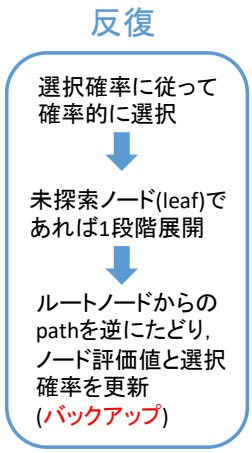

図１

また，MC Softmax 探索の名前の由来は，ルートノードから末端ノードへ到達するまで，(1)の選択確率に従って確率的にノード選択を行って経路が生成される３）の過程は，ルート局面における各指し手の良さを求めるためのモンテカルロ・サンプリング（一種のシミュレーション）に相当するからです．

上記のモンテカルロ・サンプリングを一定回数あるいは一定時間行った後，確率値の最も高い子ノードだけを次々に選択して得られた手順を最善応手手順であると決定します．

５）評価関数について

現在のところ，評価関数の特徴量は，選手権公式ライブラリである Bonanza (Ver. 6.0.0) [5] のものをそのまま使用しています．Bonanza は「Bonanza メソッド」と呼ばれる教師付学習方式が有名です．我々の研究グループは，この方式をより一般化した「方策勾配を用いた教師付学習法」を提案しています[7]．通常の教師付学習では，棋譜の着手を正解手として，この正解手の情報だけを用います．本学習法では，正解手以外の手の評価値も学習データとして利用することが可能です．

さらに，３）で述べたモンテカルロ・サンプリングで生成された探索木において，全 leaf に出現する特徴量の重みを探索時と同様なモンテカルロ・サンプリングとバックアップ操作だけで学習することが可能です[6]．将来的にはこの学習法も実装していく予定です．また，上記のような教師付学習だけではなく，報酬の最大化を目的とする強化学習（TD 法や方策勾配法），勝敗の予想確率を学習する回帰法，深い探索結果を利用する Bootstrap 法（RootStrap 法や TreeStrap 法）も，Softmax 探索とモンテカルロ・サンプリングの組合せで実行することが可能です．これにより，最善応手手順だけでなく，有力変化手順の近傍局面に出現する特徴量パラメータも，その重要度に応じて積極的に学習できるので，学習の精度や速度の向上に繋がると期待しています[10]．

なお，末端ノードでの局面評価には静止探索（駒の取り合いだけを考慮する探索）を行って，その結果を局面評価として返す処理を行っています．現バージョンのプログラムでは，この静止探索においては高速化のために従来の α β 探索を使用しています．

4. 昨年のバージョンからの更新箇所

今年のバージョンでは，以下の機能を追加しました．これまで探索木中に同一局面が存在しても全く別のノードを用意してきました．したがって，同じ局面に対してその局面からの探索処理を何度も繰り返してきました．そこで，同じ局面には一つのノードを割り当てる処理を追加しました．このノード（合流点）の管理のために，専用のハッシュテーブルを新たに用意しました．探索木中の各ノードは複数の親ノードを持つようになりますが，同一局面の展開処理を繰り返すことがないので，探索時間とノード数を削減することができます．ただ，現在は，動作が不安定な点があり完全に実装できていませんが，大会までに間に合えば実装します．

5. 今後の課題

今年は昨年同様，36 コア(72 スレッド)のワークステーションを使用する予定です．各スレッドが探索木を共有し，図 1 に示した処理を独立に行っています．しかし，有力手順を重点的に探索するためにはスレッドの割当方法を工夫する必要があります．また，各スレッドが同じ温度パラメータを持って枝を選択して降りて行く必要性は必ずしもありません．選択時に温度の高いスレッドや低いスレッドがあって，広く浅く探索するスレッドや狭く深く読むスレッドなどのバリエーションを持たせることも可能と考えています（ただし，バックアップ時の温度は一定としておく）．大会までに温度調整がうまくいけば実装します．

さらに，親ノードとそれ以下の探索木とをスレッドに分散して割当て，完全な並列分散化処理を行うことも可能です．上記のスレッド割当と探索木の分割処理とをうまく動的に行うことが今後の課題の一つです．今のところ，最善応手手順や有力手順の近傍を中心に，スレッドと探索木を探索途中で動的に割り当てることを考えています．

また，MC Softmax 探索方式は，ニューラルネットワークモデルによる評価関数表現と非常に相性が良いとされています．実際，2017 年 11 月開催の第 5 回将棋電王トーナメントでも mEssiah というチームが早速採用してくれました[9]．今後，ディープラーニングを用いた学習方式がコンピュータ囲碁だけではなく将棋へも波及して来ると予想されます．mEssiah の開発者の話によれば，ニューラルネットワークモデルによる評価値計算にはかなりの時間的コストがかかるが，GPU などを用いると多くの局面の評価値計算を一度に並列化して計算することができ，例えば，図 1 における子ノードの一斉展開と評価値計算には適しているとのことです[9]．このように，局面評価関数としてニューラルネットワークモデルを用いることも本チームの今後の課題の一つです．

6. おわりに

現在のコンピュータ将棋プログラムの多くは，探索方式（Minimax 探索の高速版である $\alpha\beta$ 探索）からソースコードのレベルまで，Stockfish[8]などのチェスプログラムから大きな影響を受けています．それに対して，本チームは Softmax 探索とモンテカルロ・サンプリン

グをベースにしています．本探索方式は囲碁プログラムで用いられているモンテカルロ木探索の一種と思われますが，プレイアウトを行わない点や，確率的選択を行っている点が異なっています．また，本探索方式は，プログラム作成が容易で，並列化の効果も高い上に，他のゲームプログラムへの適用も容易であるという点で汎用性にも優れていると考えています．

まだまだ問題点も多いのですが，新しい探索方式と学習方式を研究する上では面白さが多く[10]，開発者自身，今後の展開を楽しみにしております．最終的には，プロ棋士の棋譜を用いることなく，コンピュータ自身が自己対局を（あるいは他者との他流試合も）通して，探索法や局面評価関数を学習し，人類の棋力を超えて，新しい定跡や戦法を創出し，棋士や将棋ファンを大いに楽しませてくれることを目標としております．

参考文献

[1] 「芝浦将棋 Jr.合法手生成プログラム」の機能説明書とプログラムは次のページからダウンロードできます：http://www2.computer-shogi.org/library/

[2] 五十嵐治一，森岡祐一，山本一将，“方策勾配法による静的局面評価関数の強化学習についての一考察”，第 17 回ゲームプログラミングワークショップ 2012 予稿集, pp.118-121 (2012).

[3] 例えば，http://www2.computer-shogi.org/wcsc26/appeal/Shibaura_Shougi_Jr./appeal.pdf に記載されています．

[4] 原悠一，五十嵐治一，森岡祐一，山本一将，“ソフトマックス戦略と実現確率による深さ制御を用いたシンプルなゲーム木探索方式”，第 21 回ゲーム・プログラミング・ワークショップ 2016 予稿集，pp.108-111(2016).

[5] Bonanza のホームページ，http://www.geocities.jp/bonanza_shogi/

[6] 桐井杏樹，原悠一，五十嵐治一，森岡祐一，山本一将，“確率的選択探索の将棋への適用”，第 22 回ゲーム・プログラミング・ワークショップ 2017 予稿集，pp.26-33（2017）.

[7] 古根村光，山本一将，森岡祐一，五十嵐治一，“方策勾配を用いた将棋の局面評価関数の教師付学習：静止探索の導入と AdaGrad の適用”，第 22 回ゲーム・プログラミング・ワークショップ 2017 予稿集，pp.1-7（2017）.

[8] Stockfish のホームページ，https://stockfishchess.org/

[9] コンピュータ将棋ソフト mEssiah の内部構造，https://qiita.com/sakuramaru7777/items/ebb397eef94fc02be2d8

[10] 五十嵐治一，森岡祐一，山本一将，“MC Softmax 探索における局面評価関数の学習”，第 23 回ゲーム・プログラミング・ワークショップ 2018 予稿集，pp.212-219 （2018），