「芝浦将棋Softmax」のチーム紹介

2021年3月24日

芝浦工業大学情報工学科

岩本裕大，粂川叶，村上陽大，五十嵐治一

１．はじめに

本稿は，第31回世界コンピュータ将棋選手権（2021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5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（αβ探索）とは異なるMC Softmax探索である点が大きく異なります．棋力的には従来のαβ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．岩本と粂川は五十嵐研究室の大学院生，村上は2020年度の卒業生です．

３．「芝浦将棋Softmax」の特徴

本チームの特徴を，以下の１）～４）のようにまとめました．このうちの１)～３）が本チーム独自の探索方式である「MC Softmax探索」[1]に関する説明です．この探索方式は,文献[2][3]の研究が基になっています．

１）モンテカルロ・ソフトマックス（MC Softmax）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」が伝統的な探索方式です．この探索方式で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，αβ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（αβ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．コンピュータ囲碁で有名なAlphaGoはこの探索方式を採用しています．

本チームの探索方式は後者の選択探索に分類されます．特に，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」(Monte Carlo Tree Search)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」(Monte Carlo Softmax Search)と呼んでいます．

２）ノードの局面評価関数と探索深さ指標を用いた確率的なノード選択方策

前項で述べたMC Softmax探索によりルートノードから探索木を降りていく際には，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．

$π\left(a|s\right)=exp\left(E\_{a}\left(a;s\right)/T\right)/\sum\_{x\in A\left(s\right)}^{}exp\left(E\_{a}\left(x;s\right)/T\right)$　 (1)

ただし，*s*は局面（ノード）, *a*は指し手，*Ea*(*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*s’*の評価値*Es*(*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．（今回のバージョンでは，leafでさらに静止探索を行っています）

一方，内部ノードであれば子ノード*v*(*x;s*)の評価値*E*(*v*)をその子ノードの選択確率*π*(*x*|*s*)で重み付けた期待値

$E\left(s\right)=\sum\_{x\in A\left(s\right)}^{}π\left(x|s\right)E\left(v\left(x;s\right)\right)$ (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．

　ここまでは昨年までのバージョンでした．今年は，「探索深さ指標」$d(s)$をSoftmax探索のための深さの指標として新たに定義し, これを用いて探索を行います. 探索深さ指標は以下の式で定義しています.

$d\left(n\right)≡\sum\_{l \in leaf\left(n\right)}^{}depth\left(n,l\right)\prod\_{i=1}^{depth\left(n,l\right)}P\left(n\_{i}\right)$ (3)

ここで，$leaf\left(n\right)$はノード$n$を根とする探索木の末端ノードの集合，$depth(n,l)$は根$n$から末端$ l$への経路の長さ, $n\_{i}$は根$n=n\_{0}$から末端 $l=n\_{depth\left(n,l\right)}$までの経路上のノード, $P\left(s^{'}|s\right)$はsからs'への選択確率（$s→s^{'} ⇒ P\left(s\right)=π(s^{'}|s)$）です.

この式(3)を探索の中で効率的に計算するため式変形します. 末端ノードの探索深さ指標を0として, 内部ノードの探索深さ指標を以下の式で再帰的に求めます.

$d\left(s\right)=\sum\_{s^{'}\in C\left(s\right)}^{}\left\{P\left(s^{'}|s\right)d\left(s^{'}\right)+1\right\}$ (4)

C(s)はsの子ノードの集合です. この式変形による誤差は$10^{-10}$程度とごく僅かです（予備実験で確認済み）.

この探索深さ指標$d(s)$を用いて，ノード選択時の評価値に次のような補正を掛けることで，深さを考慮した探索を行います.

$E\_{s}\left(s\right)=E\left(s\right)\left\{1-\frac{1}{2d\left(s\right)}\right\} +E\_{0}(s)\frac{1}{2d\left(s\right)}$ (5)

$E\_{0}(s)$は局面*s*の局面評価値です. 静止探索を行わないなどの精度の低い評価関数を用いた場合は特に効果が大きく，用いない場合よりもレートにして300程度棋力が向上しました.

３）バックアップ操作

図１

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，２）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．また，(1),(2)で定義される期待値操作の方法を「バックアップ方策」と呼び，２）で述べた「ノード選択方策」と区別しています．

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２）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．

４）評価関数について

現在のところ，評価関数は既存の公開された評価関数をそのまま使用しています．今年度も昨年と同じく，AperyのKPPT型評価関数を利用します．tanuki-のNNUE も実装していますが，KPPTの方が勝率が良かったためこちらを利用しています.

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値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αβ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のまとめ

　昨年のバージョンからの更新個所を以下のように簡潔にまとめました．

(1)序盤定跡の使用

通常の序盤定跡を実装してそのまま使用することもできますが，独自の定跡データを作成しています．この方法は，序盤探索木を作成しておき，開始時に読み込んで，それを対局時に成長させ，終了時には元の序盤探索木へ格納します．これの反復により，対局を通して序盤定跡を成長させることができます．現在，実験中ですが，良い結果が出れば大会で使用します．

(2)ノード選択方策中の温度パラメータの調整

探索時のノード選択方策(1)において，温度パラメータをスレッドごと，あるいはノードごとに調整する方式を検討中です．大会までに間に合えば実装します．

(3)その他変更点

　細かいリファクタリングを行っています。子ノードの保持方法を、ノードのポインタの動的配列としていたのを、ノードの可変長配列で直接保持するように変更し、ノードの生成・削除の処理を簡素化しました。

５．今後の課題

今年は，64コア(128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．

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というチームが採用してくれました[5]．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[5]．我々の研究グループもこのメリットを実験により確認しています[6]．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．

評価関数については，独自の学習法を考案しており，今後実装していく予定です．将棋では「Bonanzaメソッド」と呼ばれる教師付学習方式が有名ですが，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が，本学習法では，正解手以外の手（例えば有力な次善手）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[1]．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[8]．

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αβ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，探索と複数局面の評価に関する並列化の効果も高く，特に局面評価ではGPUを用いたニューラルネットワークモデルの並列計算に向いています．さらに，プログラム作成が容易で，他のゲームプログラムへの適用も容易あることから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，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] 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）.

[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] 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)．

[4] 岩本 裕大 , 五十嵐 治一，“コンピュータ将棋におけるMC Softmax探索のための探索深さの指標”，第25回ゲーム・プログラミング・ワークショップ2020予稿集，pp.46-52 （2020.11.14-15），

[5] コンピュータ将棋ソフトmEssiahの内部構造，

<https://qiita.com/sakuramaru7777/items/ebb397eef94fc02be2d8>

[6] 吉野 拓真 , 五十嵐 治一 , 川島 馨，“MC Softmax 探索における局面の並列評価： GPU とニューラルネットワークモデルの利用”，第25回ゲーム・プログラミング・ワークショップ2020予稿集，pp.16-21 （2020.11.14-15）

[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] 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），