

President_X

概要

2017年5月12日 V3.1 (反省会追記)

天野 史斎

プログラム名

- ▶ 名前 : President_X
- ▶ よみがな : そうさいえっくす
- ▶ 略称 : X
- ▶ 由来 : GALAXの超AI「総裁X」

イントロダクション

①President_Xの目的

▶ コンピュータ将棋
プログラム開発から
調整の手間をなくす！
未来永劫なくす！！

※ 昨年から変わりません。

②方策

対局結果から学習。ありがちだがしかし！

1. 詰みが判明した局面から勝った局面と負けた局面（Win/Lose）を学習
2. Win/Loseのモデルは表現力が十分高いものを使う→KPPで（多分）OK
3. 上記枠組みでKPPの上にヒストグラムベースの分類器を構成し、Win/Loseの予測に関して真とする
※ ヒストグラムはサンプル数増加を前にそういつまでも嘘をつき通し得ない。
4. 項番3のWin/LoseとWin/Loseに至る前の局面の関係を説明するたくさんの仮説を生成する
5. 仮説の最良の集合で局面を分類
6. 項番5の分類ごとに評価関数（速度重視の単純なやつ）の重みを調整する。
この調整は、項番3のWin/Lose分類器の出力を真として、
通常探索結果Winが項番3によるWin判定に、
通常探索結果Loseが項番3によるLose判定に
それぞれ矛盾なく結びつくように、一貫性を基準に行う。

③詳細1

項番1～3: Win/Lose分類器

- ▶ KPPをbag of wordsとみなしたベイジアンフィルタでやる。

$$\{\text{局面}\} \rightarrow \{\log(\text{Prob}(\text{Win}|\text{局面})) - \log(\text{Prob}(\text{Lose}|\text{局面}))\}$$

どこらへんがヒストグラムベースなのかというと、このしくみがKPPを横軸（階級）とするWin局面（を構成するKPP）やLose局面（を構成するKPP）の出現回数のカウントに他ならない点、

- ▶ 対局し、当方の負けが確定した局面から勝ちと負けを学習する。

④詳細2

項番4: 仮説生成

- ▶ Win/LoseとWin/Loseに至る前の局面の関係を説明する仮説を求める。
- ▶ 将棋盤は2次元平面のような見かけをしているから、縦横斜めの平行移動の組み合わせが他よりも大きいチャンスで仮説を構成する特徴に選ばれるようにするのが妥当（なような気がする）それを愚直に実施。
- ▶ よさげな仮説（説明精度が高い）が見つかったら、小変更でさらによさげな仮説にできないか試す。

※ この特徴選択は多少しくじってもサンプル数増で帳消しが効くはず...
なぜなら、サンプルを増やしたとき、しくじったらしくじったなりの分布をWin/Lose分類器が示し始めるから、うまい仮説の小変更（上記）において、しくじりを修正する方向のバイアスの発生が期待できる（気がする）。

⑤詳細3

項番5: 局面の分類

- ▶ よさげな仮説がいくつか得られたら、良さ降順に並べて決定木を構成する。
- ▶ 局面Xを与えたときに、決定木の根から葉に至るパスがすなわち局面Xの分類コード。
- ▶ 決定木の深さの決め方は、良く知られている方法を使うと良いと思う。

⑥詳細4

項番6: 評価関数の重みの調整

- ▶ ステップ1: 所与の局面Xを、項番5の決定木で分類する。
- ▶ ステップ2: 得られた分類コードに従い、対局用評価関数（計算速度重視）のパラメータセット $PAR(X)$ を選択
- ▶ ステップ3: 所与の局面Xを根とし、Win/Lose分類器の出力を評価関数とする探索を行う。
→局面XのWin/Loseの予想が立つ
- ▶ ステップ4: 対局用評価関数での探索を何回か行い、最善手の連鎖として局面X、Y、Z、...からなる一連の局面の予想推移および評価値の系列を得る。
- ▶ ステップ3: ステップ1とステップ2の結果に矛盾（Lose予想に対し、評価値が上昇するとか、Win予想に対し、評価値が下落するとか）があれば、対局用評価関数のパラメータを上げ下げしつつステップ4を繰り返し、一致する方向に改善する。

⑦昨年（WCSC26）との違い

- ▶ WCSC26バージョンでは、通常探索の評価関数が
駒得+駒位置の得点+係数×Win/Lose分類器の出力
だったが、今回は
駒得+駒位置の得点
のみとし、Win/Lose分類器は項番6のしくみで間接的な関与
（トレーナー）とした。
- ▶ WCSC26バージョンでも対局用評価関数（計算速度重視）のパラメータ
セットPAR(X)を局面の種類ごとに切り替えていたが、局面の分類がどう
見ても効率悪かった（盤を二次元平面と見立ててはいなかった）ので項番
4の通り改良した（つもり）。

⑧その他

- ▶ 詰探索ルーチンの改良（のつもりのもの）：
 - ▶ 持駒の包含関係を考慮した詰み/不詰み判定を導入
 - ▶ GHI対策を変更 – WCSC26版では置換表のヒットを葉までの深さが一致する局面のみに限定していたが、効率が悪すぎるので、普通に深さが一致またはより大きい置換表エントリをヒットさせる。ただし一定以上古い「不詰み」エントリは無視（破棄）する。
- ▶ 時間管理の簡素化
 - ▶ WCSC26版では、探索深さと探索完了時間の統計をとって割り当て時間内に深さd+1の探索が終わりそうか否かの予測をもって探索GO/NOGO判断していた。
 - ▶ WCSC27版ではそんなややこしい仕組みを削除し、時間が来るまでとにかく反復深化を進め、時間が来たら即探索を打ち切る。ただし、深さdの不完全な探索結果を深さd-1の完全な探索結果とMIXし、深さdの探索を根の子 $\{s_1, \dots, s_k\}$ までで打ち切ったなら、根の子 $\{s_1, \dots, s_{(k-1)}\}$ を深さd、 $\{s_k, \dots, s_N\}$ を深さd-1で探索したのと等価な結果を出力する。（ $\{s_1, \dots, s_{(k-1)}\}$ の探索結果を無条件に捨てはしない。）
- ▶ 千日手/連続王手の千日手/%KACHI宣言の判定方法変更
 - ▶ 探索中の判定を廃止。代わりに、反復深化の1 iteration終了毎に、得られたPVを観察し、千日手の始まり、連続王手の千日手の始まり、または%KACHI条件成立局面があったらその旨置換表に積む。

反省会

今回は時間管理のバグで非常に不首尾な結果に終わってしまいました。

President_Xと今回対局された方々には当方の問題でつまらない内容の対局にしてしまい大変申し訳ありません。

これというのも学習部分に前のめりになりすぎたあまり、普通に指したり打ったりする基本機能の作成のための持ち時間を使い切ったからでして、切れ負け連発という結果も何かの因果かもしれません...

ただし上記問題は連休中に解消し、学習分はもともとそれなりにデバッグ済みの状態であり理屈的にも行けると思っておりますので、次回に向けて引き続き調整を進める所存です。

不完全なデータからの学習を効率化すべく、目下探索の深さを改善する取り組み中です。今後ともよろしくお願いいたします。