第 25 回世界コンピュータ将棋選手権
Apery アピール文書

平岡 拓也、杉田 歩、山本 修平、白岩 大地

2015 年 5 月 12 日

1 Apery とは

- 読み方は「えいぷりー」です。
- 意味は「猿真似」です。
- 名前はこんなですが、新しいことが出来たら良いなと思います。
- GPL v3 ライセンスでオープンソースで開発しています。
  https://github.com/HiraokaTakuya/apery

2 作者紹介

- 平岡 拓也 (ひらおか たくや)
  趣味で作っています。
  Mail : hiraoka64@gmail.com
  Twitter: @HiraokaTakuya
- 杉田 歩 (すぎた あゆむ)
  大阪市立大学工学部数理工学研究室の教員です。
- 山本 修平 (やまもと しゅうへい)
  同研究室の M 1 の学生です。
- 白岩 大地 (しらいわ だいち)
  同研究室の 4 回生です。新メンバーです。

3 実績

- 第 22 回世界コンピュータ将棋選手権 22 位 (2 次予選敗退)
- 第 23 回世界コンピュータ将棋選手権 9 位 (2 次予選敗退)
  決勝進出に一歩足りず。
  新人賞に一歩足りず。

- 第 1 回将棋電王トーナメント 6 位 (5 位決定戦敗退)
  第 3 回電王戦出場に一歩足りず。
- 第 24 回世界コンピュータ将棋選手権 優勝
- 第 2 回将棋電王トーナメント 5 位
- 将棋電王戦 FINAL 斎藤慎太郎五段に完敗

4 技術的特徴

- 全体的に Stockfish ベースで、Bonanza の 3 駒関係による局面評価などの将棋に適した機能も盛り込んでいます。
- 出来るだけ簡潔なプログラムになるように心掛けています。
- 探索において特徴的な点は、多くのプログラムに存在する王手のみを生成する機能が無いことでしょうか。
  静止探索で王手のみを調べることが無く、その為か終盤が弱いとの評判ですが、それでも中盤は十分戦えることが分かりました。
  プログラムをコンパクトにしたいなら王手生成は省いても問題無さそうです。(今後は王手生成するかも知れません。)
- 評価関数は対局時には KP 要素の排除や持ち駒 0 枚の排除、KPP と KKP のインデックスの共通化、三角配列から矩形配列への変更などにより Bonanza よりも高速化しています。
- コンピュータ将棋で (おそらく) 初めて PEXT Bitboard を採用しました。Magic Bitboard よりも 1% 強の高速化を実現しています。
- Bonanza Method による評価関数の機械学習をチームで手分けして色々な方法で試しています。
  Apery の学習部は最近書き直し、単純に相対位置と絶対位置を重ねあわせる方法を試しています。
  また、AWAKE がおこなっている利き情報を 3 駒関係に落とし込む方法も取り入れました。Bonanza の学習部を改造したものも実験で使用しています。その為、Bonanza Ver.6 ライブラリ使用としています。そこでは相対位置の同じ位置関係の点数が滑らかになるようにフーリエ変換でペナルティ項を調整しています。(この手法は結局本番バージョンでは採用しなかったのですが、他の人の参考になるかと思って少し詳しく付録を書きました。)

5 昨年との違い

- 評価関数の機械学習の工夫による精度改善がほとんどです。
  昨年版と直接比較はしていませんが、半年ごとに 1 度ずつ、自己対局で前のバージョンに勝率 7 割を達成しました。

付録 A　評価関数の近似的並進不変性を考慮したペナルティ項の付け方について

A.1　まえおき

コンピュータ将棋においてよく用いられる３駒関係の評価関数は膨大な数のパラメータを持つため、全ての項目に正しく点数をつけるにはプロ棋士の棋譜データだけでは圧倒的に情報不足で、何も工夫せずに学習すると過学習によって非常に弱い評価関数になります。従って、教師データから決めきれない項の振る舞いをなんらかのやり方でコントロールしてやる必要がありますが、このために通常用いられるのがペナルティ項です。

普通用いられるペナルティ項は、L1 型とか L2 型とかありますが、どちらにしても、基本的な考え方は「よく分からないときは点数は小さめにつけろ」というものです。特に、教師データから情報が得られない項目は０になります。これは悪くないやり方で、確かに過学習を防ぐことは出来ますが、一方、分からない項目を全部０にしてしまうのは安直すぎて、せっかくの３駒関係の表現力を無駄にしているような気もします。そこで、将棋における評価関数の性質を考えれば、もうちょっとマシな推定が出来るのではないか、ということを以下で考えていきます。

A.2　近似的並進不変性と「次元下げ」

将棋において駒の配置に点数を付ける場合、駒達の間の相対的な位置関係は非常に重要ですが、絶対的な位置はそれと比べれば低い重要性しか持ちません。例えば、玉と金の位置関係を考える場合、玉と金がどれくらい離れているか、ということは非常に重要ですが、玉の隣に金がいるという配置は、玉の位置がどこであろうと良い配置であると考えることができます。各駒の相対位置を保ったまま駒を移動させる操作を並進と呼ぶことにすると、このことは「評価関数が近似的な並進不変性を持つ」というふうに表現することができます。

この近似的並進不変性を考えれば、上で述べた、データから決められない３駒関係の価値についても、０以外のある程度意味のある値を与えることができます。現在よく用いられているのは、相対位置のみによって決まる評価関数を別に用意し、絶対位置を含む評価関数と重ね合わせるという方法です。（いわゆる「次元下げ」の一種。）

A.3　フーリエ変換したペナルティ項による処方

一方、上で述べた次元下げと同じようなことを、ペナルティ項の形を変えることで実現することもできます。ペナルティ項のとり方は色々考えられますが、ここでは Apery チームで試しているフーリエ変換による方法を紹介します。

まず玉とそれ以外の２駒の相対位置を固定して、玉の位置の座標を x とします。このとき、各項につく点数は x の関数なので $f(x)$ と書きます。（本当は座標は２次元ですが、簡単のため１次元で考えます。）x の取りうる値が 0 から $N-1$ の N 個だとします。（これも実際の盤上の座標からは適宜変換して考えて下さい。）このとき関数 $f(x)$ はそのフーリエ変換 $\tilde{f}(k)$ と以下のような関係で結ばれています。

$$f(x) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \tilde{f}(k) e^{2\pi ikx/N}$$

$$\tilde{f}(k) = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) e^{-2\pi ikx/N}$$

ここで複素数を使ったのは単に見た目上表現を簡単にするためで、オイラーの公式 $e^{i\theta} = \cos\theta + i\sin\theta$ から分かるように、これは本質的には関数 $f(x)$ を三角関数を組み合わせて表現する手法です。$k = 0$ の項は x によらない、つまり、並進に対して不変な部分を表しています。$k \neq 0$ の項は x にの変化に対して振動する部分で、k が大きいほど振動が激しくなります。この意味で、k が大きい項はより強く並進不変性を破っていると言うことができます。

通常のボナンザメソッドのペナルティ項は、各点 x における値 $f(x)$ を使って $\lambda\sum_{x=0}^{N-1}|f(x)|$（L1 型）や $\lambda\sum_{x=0}^{N-1}|f(x)|^2$ （L2 型）のように書かれます。λ はペナルティの強さを決めるパラメータです。

ここで、$f(x)$ ではなく、そのフーリエ変換 $\tilde{f}(k)$ を使ってペナルティ項を書くことを考えます。k の値が「並進不変性を破る度合い」に対応していることを思い出して、各項に異なる重みをつけたペナルティ項を考えます。例えば、L1 型の場合であれば

$$\sum_{k=0}^{N-1} \lambda_k |\tilde{f}(k)| \tag{1}$$

のような形です。ここで、k が小さいときに λ_k が小さくなるようにしておけば、並進不変性をより尊重した点数の付き方が実現できると考えられます。

例えば、$k = 0$ の項だけペナルティを弱くしておくと、玉の絶対位置によらない、相対位置のみで決まる項目の点数がつきやすくなり、これは次元下げと同様の効果が期待できます。また、k が 0 ではないが小さい項のペナルティを弱くしておくと、玉の位置によってゆるやかに変動するような効果を取り入れることができます。

A.4　フーリエ変換で強くなるのか？

さて、問題はこの手法でより良い評価関数が実現できるかどうかです。理屈で考えると、フーリエ変換したペナルティ項は、次元下げのような効果を含み、さらに一般的に点数の付き方をコントロールする手段を与えるので、次元下げより悪い理由はないように思われます。しかし、現状ではいくつか細かい問題があってそれほど上手くいっていません。問題の一つは、L1 型の (1) の形でペナルティをつけた場合、絶対値の中が $f(x)$ に対してかなり複雑な形となるため、収束性に問題が出てくることです。この点については、点数のアップデートをするときに $f(x)$ を書き換えるのではなく、フーリエ変換後の $\tilde{f}(k)$ のほうを書き換えることで収束性が改善することを確認しました。しかし、$\tilde{f}(k)$ を動かすようにするとペナルティの効き方が変わる（より強くなる）ようで、ペナルティ項の強さの再調整をしないといけないようです。これは現在試みている最中ですが、残念ながら今回の選手権には間に合いませんでした。

また、収束性の問題は、L1 型ではなく、微分不可能性の問題のない L2 型等を用いることで改善する可能性があります。ただし、この際にもペナルティの強さの再調整が必要になるので、少し時間がかかります。今後機会があればこれらの実験結果についても報告したいと思います。

A.5　評価関数の関数系による展開について

ここで用いたフーリエ変換は、評価関数を三角関数を使って展開するという考え方ですが、三角関数に限らず、評価関数を何らかの関数系によって展開して表現するというのはおそらく自然なアイデアで、例えば多項式による表現については既に 2006 年に Bonanza 開発者の保木邦仁氏によって述べられています。(「局面評価の学習を目指した探索結果の最適制御」, 第 11 回ゲームプログラミングワークショップ, pp. 78-83, 2006）昨年の選手権のときに保木さんにお聞きしたところ、保木さんは三角関数系を含む様々な関数系で評価関数を展開し、高次項を落とすことで次元の低い近似的表現を求めることを試したそうですが、どれもうまくいかなかったそうです。おそらくこれは、評価関数の絶対位置による表現にそのまま関数展開を適用したためで、この場合、評価項目の点数は非常に激しく変化し得るので、近似的な表現をとることが難しかったのだろうと考えられます。

一方、ここで述べたように、相対位置を固定して絶対位置を変化させた場合の点数は緩やかに変化することが期待できるので、関数展開による近似はうまくいきそうに思えます。Apery チームではまず多項式近似を試してみたのですが、これはデータのない領域に外挿していったときにとんでもなく大きい値がついてしまうことがあり、うまくいきませんでした。その後、三角関数系による近似を試しましたが、これはそれなりにもっともらしい点数の付き方が実現できました。しかし、次元を落として近似すると情報量が減るので、やはり弱くなってしまうようでした。現在では、次元を落として近似的表現を求めるよりも、ペナルティ項の強弱で各項の点数の付き方のコントロールをするのが正しいのではないかと考えています。あるいは、現在行われている次元下げの一般化として、(関数展開による近似的評価関数）＋（通常の評価関数）の重ね合わせの形で評価関数を作る方向は有望かもしれません。

A.6　ペナルティ項の積極的活用

パラメータフィッティングをベイズ推定の立場から見ると、ペナルティ項は事前分布の役割をしていると見ることができます。その観点から言うと、現在のボナンザメソッドのペナルティ項は、「評価関数の評価項目に大きい点数が付く確率は低い」ということを事前情報として導入していることになります。もちろんこれはこれで有用なのですが、我々は将棋の評価関数について事前にもっと詳しい情報を知っているのだから、それをペナルティ項に導入して、積極的に点数の付き方をコントロールすることが可能なはずです。今回は並進不変性について考えましたが、これからは他にも様々な観点からペナルティ項を積極的に活用する方法を考えていきたいと思っています。