

## 学習を用いた枝刈りの新手法の提案

山本 一成<sup>†</sup> 田浦 健次朗<sup>‡</sup> 近山 隆<sup>††</sup>

<sup>†</sup> 東京大学 工学部 電子情報工学科 <sup>‡</sup> 東京大学大学院 情報理工学系研究科 <sup>††</sup> 東京大学大学院 工学系研究科

### 1. はじめに

強いコンピュータ将棋プログラムを作成するためには探索と評価関数それぞれの高性能化が必要不可欠である。近年 Bonanza Method<sup>1</sup>に代表されるように評価関数の分野において機械学習によるブレイクスルーが起きており、2009 年度のコンピュータ将棋選手権の決勝戦では 8 チームのうち 6 チームが評価関数に何らかの機械学習の手法をとり入れるまでに至った。一方でゲーム木探索は古典的なチェス由来の探索から大きく進展していない。しかし探索においても人手での調整に手間のかかるパラメータ調整は少なくない。そこで本研究ではこれらのパラメータ調整に機械学習を用いるという手法を提案する。局面の状態や指し手の性質、探索の動的情報など様々な特徴を用いて枝刈りの条件の最適化を行い、有効性を示した。

### 2. 関連研究

本研究は前向き枝刈りの一種 Futility Pruning<sup>2,3</sup>のパラメータの自動最適化を図る。前向き枝刈りとは指し手を作成する段階で明らかに悪そうな指し手を枝刈りして探索を打ち切り、探索時間の短縮を狙う手法である。またこれとは逆に探索の結果を変えない安全な枝刈りは後ろ向き枝刈りと呼ばれる。Futility Pruning とは探索木の末端付近の局面において評価値が悪い場合に探索を打ち切るという手法である。しかしながら評価値がどれほど悪ければ打ち切りをしてよいのかというパラメータ調整が困難である。

### 3. 提案手法

#### 3.1 本研究で用いるプログラム

本研究では提案手法の検証用として ponalgam と名づけられたプログラムを実装した。このプログラムは Bonanza<sup>4</sup>, Fruit<sup>5</sup>, Blunder<sup>6</sup>などを参考に現在将棋プログラムを作る上で一般的と考えられる以下のような技術を用いている。

- 全幅探索 + 静止探索
- 後ろ向き枝刈り
  - Beta Cut<sup>7</sup>, PV Search, 置換表, 多重反復深化 Distance Pruning<sup>8</sup>
  - Killer Move, History, SEE
- 前向き枝刈り
  - Null Move Pruning<sup>9</sup>, Late Move Reduction<sup>10</sup>
- 王手延長
- 並列探索

#### 3.2 前向き枝刈りの学習

機械学習のためにロジスティック回帰を用いた。ロジスティック回帰<sup>11,12</sup>は教師あり学習の一種であり、学習を行うにはさまざまな入力とそれに対する正解が必要である。前向き枝刈りが可能かどうかを学習する時の教師例として、本研究ではプロの棋譜の 25000 局面で深さ 5 手の探索を行い、探索の結果その局面での指し手が枝刈り可能であったか、自らの探索結果を教師値として学習を行う。ここで枝刈りしてはいけない枝を枝刈りできると判定することを FN、逆に枝刈りできる枝を枝刈りできないと判定することを FP と定義する。判定に求められる性質は両方共に低く抑えたいがより FN を下げたいというものである。そこで枝刈りしてはいけない枝に関してより大きな重みをつけてロジスティック回帰を行った。教師例として枝刈りができる枝と枝刈りできない枝を学習するのだが、すべての教師例に学習を行うと教師例のうち枝刈りできる枝ができない枝よりも 50 倍ほど多くなってしまったため、枝刈りできる枝に対しては乱数を用いて間引いた。

以下の指し手は予測の失敗時にコストが高いと考え枝刈り対象から排除した。

- 全幅探索時の PVNODE
- 王手を回避する手
- 王手となる指し手
- 残り深さ 5 以上の場合
- 他の指し手がすべて詰んでしまう場合

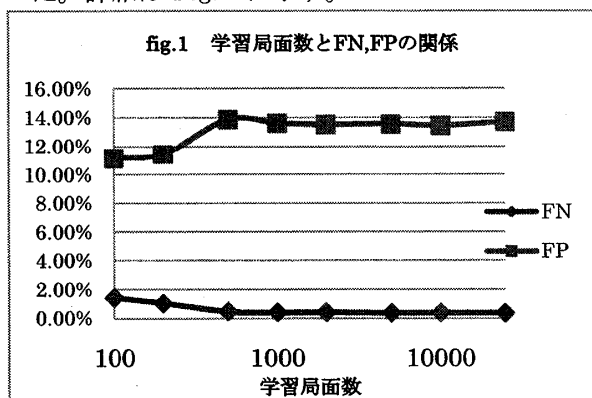
### 3.3 学習に用いた特徴

学習すべき特徴として以下のような特徴を使用した。特徴数の合計は約 240000 となっている。

- ・オーダリングの順序
- ・残り深さ
- ・PVNODE であるかどうか
- ・取った駒がある場合なにを取ったか
- ・取り返しである場合なにを取り返したか
- ・持ち駒を打つ手であるかどうか
- ・成る手であるかどうか
- ・alpha 値と現在の評価値の点差
- ・移動先の周辺 縦 5×横 3 のパターン

### 3.4 学習局面数と正答率

学習局面数と学習の進み具合との関連性を調べた。詳細は fig.1 に示す。



### 3.5 結果、考察

25000 局面を学習データとして用いて、その学習結果から作成されたデータで未知のデータ局面 1199402(内訳:枝刈できない指し手 700097 枝刈できる指し手 499305)でのテストを行った。その結果 Accuracy 91.87%、FN は 0.33%、FP は 13.67%に抑えることに成功した。また本手法を用いなかった他は同一のプログラムと 1 手 10 秒での対戦をして 61 勝 36 敗 3 引分と有意に勝ち越すことに成功した。しかしながら一方で従来手法である Futility Pruning を人手でよく調整したものを作成し対戦させた場合 1 手 10 秒での対戦をして 27 勝 72 敗 1 引分となり逆に大きく負け越した。また 1 手 10000 node 探索できとして対戦すると 41 勝 33 敗 1 引分となる。よって考えられる原因として今回提案した枝刈手法は枝刈を行うために多くの計算コストを払っており、その結果探索速度がなにも行わない場合に比べ 30-50%ほど遅くなってしまった為に同一時間内対戦では大きく負け越してしまったと考えられる。

### 4. 今後の課題

本研究で使われた手法は、手法を用いなかったものに対して優位に勝ち越すことができたが、本研究で示した手法と似たような手法である人手で設定した Futility Pruning に対しては優位であるということはできなかった。なぜなら本手法は比較的計算コストが大きく、探索速度が大幅に減少してしまうためである。本手法を今後改善していくためには計算コストが比較的軽くなおかつ有効な特徴を選択していく必要があると考えられる。

### 参考文献

- <sup>1</sup>保木邦仁, “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, Vol.11 特別号 pp. 78-83, 2006.
- <sup>2</sup>保木邦仁, “コンピュータ将棋における全幅探索と futility pruning の応用” 情報処理 Vol.47 No.8 pp.884-889, 2006.
- <sup>3</sup>Heinz, E. A, “Extend Futility Pruning”, *ICCA Journal* Vol.21 No.2 pp.75-83, 1998.
- <sup>4</sup>[http://www.geocities.jp/bonanza\\_shogi/](http://www.geocities.jp/bonanza_shogi/) Bonanza 公式ページ
- <sup>5</sup><http://www.fruitchess.com/> fruit 公式ページ
- <sup>6</sup>下山 晃, “コンピュータ将棋における機械学習を用いた指し手の順序付けと思考時間制御” 秋田大学 工学資源学部 情報工学科 卒業論文, 2009.
- <sup>7</sup>Knuth, D. E, Moore, R. W, “An Analysis of Alpha - beta Pruning”, *Artificial Intell* Vol.6 pp. 293-326, 1975.
- <sup>8</sup><http://chessprogramming.wikispaces.com/Mate+Distance+Pruning>
- <sup>9</sup>Beal, D. F, “Experiment with Null Move Advances in Computer Chess” *Elsevier Science*, pp. 65-79, 1989.
- <sup>10</sup><http://www.glaurungchess.com/lmr.html>
- <sup>11</sup><http://www.csie.ntu.edu.tw/~cjlin/liblinear/>
- <sup>12</sup>Rong-En Fan, Kai-Wei Chang, Cho-Jui Heish Xiang-Rui Wang, Chih-Jen Lin, “LIBLINEAR: A Library for Large Linear Classification”, *The Journal of Machine Learning Research* Vol.9 pp.1871-1874, 2008.