

# 将棋の棋譜の指し手から意図を読み取る

金子 知適

## 概要

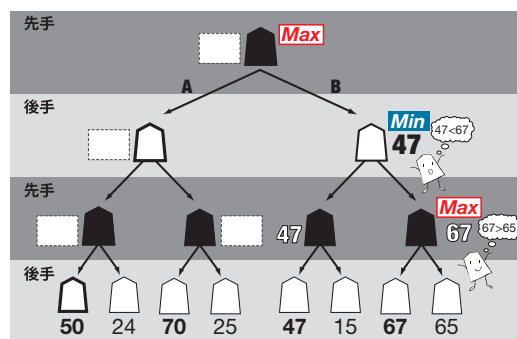
コンピュータは人間の知能を越えるだろうか？ チェス，将棋，囲碁などの思考ゲームに限定すれば多くの人がこの間を肯定的に考えるようになってきている。1997年にチェスで Deep Blue [1] が勝利をおさめた際には，先を読む探索能力が優れていると理解されたが，2013年に

将棋でコンピュータが A 級八段に勝つまでに至った [5] 原動力としては，局面の良し悪しを直感的に判断する形勢判断力の改善が有力である。熟達者の棋譜からプログラムが形勢判断を学ぶ方法について，簡単に紹介したい。

## 1 ゲームの先を読む思考

将棋・チェスなどの思考ゲームを研究するゲームプログラミングの分野は，強いプログラムを作ることは開発者の夢であるというだけでなく，人工知能の成果を測る題材として古くから研究が行われてきた。ゲームのルールを計算機上で表現することは簡単であるにもかかわらず，実際に勝つために良い指し手を選ばせることは大変に難しいためである。将棋に先行したチェスの研究では，人間のプレイヤーの考え方を計算機上で再現することにこだわらずに，計算機に適した思考方法を追求することで強いプログラムが開発された。以下に紹介する「探索」と「評価関数」の組み合わせが，その主流の枠組みである<sup>1</sup>。探索は自分と相手の指し手の組み合わせを考えて未来をシミュレーションする

こと，評価関数は予想される各局面の良し悪しを直感的に判断する役割を担う。



AとBどちらを指すと先手は得？

図 1. ゲーム木とミニマックス探索

評価関数は，入力された局面に対応する数値（評価値と呼ぶ）を出力し，評価値は先手（後手）が勝ちやすいほど正（負）に大きいものとする。

将棋の単純な例で「先手の駒の枚数 - 後手の駒の枚数」という評価関数は、駒得している方が強いというゲームの性質を反映したものである。(夢物語ではあるが)もし評価関数が完全に正しければ、一手先の局面の評価値が最大となる指手が最善手と分かる。実際には、現実の評価関数は誤差を含むので、一手だけではなく相手の応手やその先も含めて評価するほうが正確である。そのために図1のように局面を頂点、指し手を辺としたグラフを仮想的に作成する。評価値の概念も拡張し、葉では評価関数の出力をそのまま用いるが、内部節点ではその手番の立場で最善の子局面の評価値と再帰的に定義する。それは、先手番では子の評価値の最大値、後手番では最小値をとると計算できるので、ミニマックス探索と呼ばれる。先に例に挙げた駒の枚数を数える評価関数では平穏な局面と自分の駒が取られそうな局面を区別できないが、探索の過程で相手が自分の駒を取った後の局面ではその分評価が下がるので、結果として駒が取られそうな一手前の局面の評価値も低くなる。このような理由で、深く探索すればそれだけ強

くなることが知られている。一方で、評価関数で局面の分析に時間をかけると探索は遅くなり、探索で深く読めなくなるというトレードオフがある。そのため駒が取られそうかなどの各評価項目を、評価関数で判断すべきか探索でカバーすべきかについての一般的な指針は知られてない。

探索と評価関数をこのように分離したとすると、探索においては一定時間で深く読むこと、もしくは同じことであるが一定の質の結果を早く得ることが目標となる。幅(≈局面あたりの合法手の数)が $b$ で深さが $d$ の様なゲーム木を仮定すると $b^d$ 個の葉の訪問が必要となるが、不要な枝の探索を削減することで、指数の底あるいは肩を減らすことができる。直感的には、ある部分木について浅く探索しても深く探索しても評価値がほとんど変わらなければ、その部分を深く探索する必要はないだろう。古典的な $\alpha\beta$ 枝刈りは同じ結果を $b^{d/2}$ 程度で得るが、現代では様々な枝刈りが用いられ探索空間はさらに大幅に削減されている[4]。また分散・並列計算も有効である[5]。

## 2 棋譜から評価関数を学習する

評価関数をどのように作成すべきかについては、比較的研究が少なく、複数のゲームで実績のある方法は確立されていない。評価関数は本質的にゲーム毎に作成する必要があるため、ゲームの性質次第で適する手法が異なりうる。

最近になって将棋で棋譜の指手を教師として数千万という規模のパラメータを自動調整する手法が開発され、他のゲームでも有効であろう

と期待されている。チェスでも1980年代頃から似たようなアイデアが散発的に試みられてきたなかで、実際に成果をあげるまでの技術的困難など、詳細は文献[3]を参照されたい。

この手法では、評価関数の良さについて、探索で選ばれる指手がプロ棋士など熟達者が選んだ指手と一致するほど良いという仮定をおく。そのうえで、熟達者が選んだ指手の記録である

棋譜との差異を表す以下の式を最小化する:

$$\sum_{p \in \mathcal{P}} \sum_{m \in \mathcal{M}'_p} T(s(p, d_p, \mathbf{w}) - s(p, m, \mathbf{w})).$$

ここで、 $\mathbf{w}$  は評価関数のパラメータを表すベクトル (歩 1 枚の価値や、先手玉が 8 八でかつ金が 7 八に置かれている価値など) で、括弧内の左側  $s(p, d_p, \mathbf{w})$  は棋譜の局面  $p$  で指された指し手  $d_p$  をパラメータ  $\mathbf{w}$  の評価関数を使って探索して得た評価、右側の  $s(p, m, \mathbf{w})$  は同じ局面  $p$  で指されなかった指し手の一つ  $m$  に対する同様の評価である。関数  $T(x)$  は左右反転させたシグモイド関数  $1/(1 + e^{ax})$  なのでこれを階段関数と考えると、式全体は、括弧内の差が負となる個数を、学習対象の局面の集合  $\mathcal{P}$  の各局面  $p$  で、プロ棋士が選ばなかった指し手の集合  $\mathcal{M}'_p$

について数えたものである。すなわち、プロ棋士の指し手より (誤って) 高く評価した指し手の数を大まかに表す。なお、後手番の局面では符号を変えて対応する。

この学習で必須の部分は、棋譜の指し手の局面をそのまま評価関数で評価するのではなく、その局面を探索した評価値を用いるという点である。この探索が深いほど、棋譜の指し手の意図の推定がより正確になるが、約 5 万の棋譜を処理する必要がある学習時間の都合から現状では 1, 2 手 (と駒の取り合い) 程度しか読むことができていない。実際の対局では有力な展開を 20 手以上先まで読むことを考えると、学習時にはほんの浅い思考で棋譜の意図を想像していることに相当する。

### 3 現在の課題と人との協調に向けて

パラメータ調整の自動化により、評価関数で用いる評価項目を大幅に増やすことが可能となった。2003 年の時点では 100 万程度の調整でも困難があったのに対し [6]、現在では数千万から億のオーダーのパラメータから有効なもののみ調整した評価関数が実際に将棋プログラムで用いられている。この評価項目の大規模化により、評価は正確になった。

一方で、機械学習は全ての課題を解決したわけではなく、良い評価関数を作っても、探索の効率が相手に劣れば勝率は落ちる。また、評価関数の学習の過程での探索の強化も必要である。そのため、それぞれ別に発展した探索と評価関数を同時に扱い、学習中のものを含む個々の評価関数の個性に適した探索を行う手法が重

要になるだろう。

最後に、人と協調するためには、指し手を選んだ背景を説明する能力も必要である。現在の将棋プログラムは未知の局面でも平均的には良い指し手を選ぶことができるようになったが、そのプログラムの思考方法は人間とかなり離れていることから、指し手という結論にいたった理由を人間が推測することも簡単ではない。棋士とプログラムが力を合わせて対局するという状況を想定すると、対等の立場で相談して良い手を探すことは難しく、棋士が主体で指して補助的にプログラムの評価も確認するスタイルかそれともプログラムに指し手を一任するかという極端な関係となってしまう。現場で信頼される人工知能システムを実現するためには、この

ような問題を解決してゆくこともまた今後の課題である。

## 註

<sup>1</sup> なお囲碁プログラムでは、モンテカルロ木探索という乱数を用いる技法が有力だが、こちらも人間の思考方法を模倣したものとは言いがたい [2].

## 参考文献

- [1] M. Campbell, A. J. Hoane, Jr., and F.-h. Hsu. Deep Blue. *Artificial Intelligence*, 134(1–2):57–83, Jan. 2002.
- [2] S. Gelly, L. Kocsis, M. Schoenauer, M. Sebag, D. Silver, C. Szepesvári, and O. Teytaud. The grand challenge of computer go: Monte carlo tree search and extensions. *Commun. ACM*, 55(3):106–113, Mar. 2012.
- [3] K. Hoki and T. Kaneko. Large-scale optimization for evaluation functions with minimax search. *J. Artif. Intell. Res. (JAIR)*, 49:527–568, 2014.
- [4] K. Hoki and M. Muramatsu. Efficiency of three forward-pruning techniques in shogi: Futility pruning, null-move pruning, and late move reduction (LMR). *Entertainment Computing*, 3(3):51–57, 2012.
- [5] 金子, 田中. 多数の計算機を活用したゲーム木探索技術の進歩—三浦弘行八段と GPS 将棋との対局を振り返って—. *情報処理*, 54(9):914–922, 2013.
- [6] 金子, 田中, 山口, 川合. 駒の関係を利用した将棋の評価関数. 第 8 回ゲームプログラミング ワークショップ, pp. 14–21, Nov. 2003.



金子 知適 (かね ともゆき)

現在の所属: 情報学環 総合分析情報学コース

東京大学大学院総合文化研究科博士課程修了. 博士 (学術). 同大学大学院総合文化研究科助手, 助教, 准教授を経て, 2015 年より情報学環准教授. 情報処理学会, 人工知能学会, 日本ソフトウェア科学会, ACM. 2003 年頃より, オープンソースの将棋プログラムである GPS 将棋の開発に参加. ゲームプログラミング, 探索, 機械学習, 人工知能に興味を持つ. 情報処理学会論文賞 (2013, 2007), 同創立 50 周年記念論文表彰 (2010), 同山下記念研究賞 (2008), ACM-ICPC Coach Award (2013) 受賞.