



4 最近のコンピュータ将棋の技術背景と激指



マンチェスター大学 鶴岡 慶雅

止まらないコンピュータ将棋

将棋プログラムの強さはいずれ頭打ちになるのではないかとわれ続けてきたが、いつのまにかアマチュアのトップといい勝負ができるようなところまでできてしまった。

もちろん、毎年着実に進歩を続けるハードウェアが棋力向上に果たした役割は大きいが、ソフトウェアの進歩もそれと同じぐらい重要だ。ソフトウェアがここまで進歩した背景には、細かな改良の積み重ねだけではなく、いくつかのブレイクスルーがあったからでもある。本稿では、コンピュータ将棋の技術背景について最近の動きに焦点をあてて簡単に解説する。

評価関数の自動学習

今年の世界コンピュータ将棋選手権で最も目立った動きの1つは、評価関数の自動学習を採用したプログラムが激増したことだ。Bonanzaというプログラムの作者である保木氏によって提案されたこの手法は、コンピュータ将棋のレベルアップに大きく貢献したブレイクスルーといってよい。決勝リーグに関していえば、8つのプログラムのうち、「激指」、「YSS」、「備後将棋」を除くすべてのプログラムが自動学習を採用している。

評価関数とは、人間でいう形勢判断に相当する部分で、局面の有利不利を定量化して表現するための関数だ。たとえば、互角の局面であれば0、先手が優勢の局面であれば+800といった数値を返す。評価関数の設計はいうまでもなく非常に重要で、いくら探索アルゴリズムが優れていても、局面の形勢判断が正確にできなくては、正しい手を選択することはできない。

従来、将棋の評価関数の設計は大変であった。まずは、駒の価値を決めるところから始まり、駒の働き、玉の危険度などを、人手で経験的に決めていかななくてはなら

ず、将棋に関する知識と細かなチューニングが必要であった。特に、序盤の駒組みに関する評価値の設計は、プログラムの将棋に関する知識がかなりダイレクトに影響するため、ある程度の将棋の知識がないと、強い評価関数を設計するのは難しかった。

評価関数を自動的に学習しようとする試みが Bonanza 以前になかったわけではない。たとえば、TD-leaf と呼ばれる強化学習ベースの手法を使うことで、駒の価値などの学習にある程度成功したという報告などがある。しかし、序盤の微妙な駒組みから終盤の玉の危険度のようなもので、すべてを自動的に学習するのは難しいとされていた。

TD-leaf 法が、「教師なし」学習であるのに対して、Bonanza 式の学習は「教師あり」学習といえる。一口に言えば、学習データの棋譜とプログラムの指し手になるべく一致するように評価関数のパラメータをチューニングするという手法だ。単純な機械学習の教師あり学習の問題とは異なり、間に探索というプロセスが挟まっているため、問題は簡単ではない。評価関数のパラメータが変われば当然探索結果も変わるのだから、普通に考えれば、指し手の一致率の最適化を、数学的に自然な形で定式化するのは難しそうに見える。

Bonanza 式の手法の素晴らしいところは、評価関数が変化しても、読み筋は変化しないという大胆な仮定をおき、問題をシンプルな最適化の問題に帰着させてしまったことだ²⁾。最終的には、スカラー関数の最小値を求めるとい問題になるため、勾配をもとにパラメータを最適化してしまえばよい。開発者の保木氏が化学の分野の出身であるため、情報科学の人にとっては少し見慣れない定式化になっているが、本質的には確率モデルを用いた機械学習の問題と同様だ。情報科学の人にとっては文献1)による表現のほうが分かりやすいかもしれない。

この手法を実装するにあたって悩ましい問題の1つは計算コストである。基本的には、学習データのすべての



局面において、正解手と、そうでない手（これが数が多いので大変だ）に対して、読み筋（最善応手手順）を計算しなくてはならないため、相当に時間がかかる。将棋の一局は約 120 局面からなり、局面ごとに数十手以上の合法手が存在するから、単純にやるとそれだけで 1 万回以上の探索が必要だ。学習には、数千から数万局の棋譜を使うことを考えると、かなりの計算コストがかかることになる。

もう 1 つの問題は、学習されるパラメータが、どうしても盤面の表層的な情報にひっぱられることだ。初期の Bonanza では、自分の角をすぐに相手の金と交換してしまうという欠点があった。プロの棋譜では角を金と交換することはよくあるが、それは、交換したあと有利になる展開まで先を読んでいるからできることだ。しかし、学習時の浅い探索では、そこまで読みきれていないことが多く、結果として、角と金はそれほど価値が変わらない、という（ある意味で）誤った学習がなされてしまう。そのため、Bonanza 式の自動学習を採用したプログラムでも、ある程度プログラムの将棋の知識を利用してパラメータに制約をかけて学習させるほうが強くなる、ということがまだあるようである。もっとも、この問題は、深い探索が学習時に使えるようになると自然に解決されてくるかもしれない。単に計算コストだけの問題であるとすれば、それほど本質的な問題ではないだろう。

評価関数が自動的に学習されるとはいっても、評価関数を表現するベースとなる特徴量に関しては、プログラムが自分で設計する必要がある。実はこれがなかなかくせもので、なんでも適当に放り込めばいいというものではない。ある程度一般性を持っていて、現実的な学習データでスパースにならないような、「筋のいい」特徴量を設計するには、それなりに経験とセンスが必要だ。このあたりは、普通の機械学習の問題とよく似ていて、結局最終的に精度の善し悪しを決めるのは、特徴量の設計次第という面もある。

評価関数の人手によるチューニングは、将棋に関する経験や知識もさることながら、とにかく手間と時間の必要な作業である。そのため、特に開発に時間をかけられないプログラムにとっては、プログラムを強くする上での大きな障害であった。それが、この手法によって、棋譜さえあれば、まともな評価関数が自動的に作れるようになったのだから、コンピュータ将棋にとって非常に大きなブレイクスルーである。今年は特に、予選を含め選手権に参加しているプログラムの全体的なレベルアップが目立ったが、Bonanza 式の評価関数の学習手法がそれに大きな役割を果たしたことは疑いない。

探索

評価関数が、人間でいうところの「形勢判断」や「大局観」といったものに対応するのに対し、探索は「読み」に相当する。重要な展開をいかに深く読めるかが将棋プログラムの棋力を大きく左右するため、探索アルゴリズムの設計は非常に重要だ。

探索に対するアプローチは、大まかにわけて、全幅探索を基本とするものと、将棋の知識を使って積極的に前向き枝刈りをするものとにわかれる。従来、将棋プログラムのほとんどは後者であったが、Bonanza が全幅探索を基本とした手法で成功したために、最近では、全幅を基本とするプログラムも増えている。その場合、基本的にはすべての局面で可能な合法手をすべて生成し、それにコンピュータチェスの分野で提案されてきた、null move forward pruning や futility pruning といった枝刈り手法を組み合わせ探索を行うことになる。

積極的に前向き枝刈りをするアプローチの場合、指し手の種類ごと（駒を取る手、成る手、王手をかける手など）に専用の指し手生成ルーチンを用意し、探索の残り深さに応じて、有効そうな指し手のみを生成するという手法をとることが多い。

激指の探索手法

通常、探索範囲の制御には深さ（手数）を用いることが多いが、激指では、局面の実現確率による手法⁴⁾を利用している。この手法では、探索木のルートの局面の実現確率を 1 とし、それ以外の局面の実現確率を、

$$(\text{局面の実現確率}) = (\text{親の局面の実現確率}) \times (\text{遷移確率})$$
という形で再帰的に定義する。探索範囲を決めるのに、実現確率を閾値とすれば、実際に起こりやすい展開を重点的に探索することができるといわけだ。遷移確率をどの手でも同じ値にしてしまえば、通常の、深さを利用した探索アルゴリズムと等価になる。

遷移確率は、その遷移に対応する手が指される確率なので、大量の棋譜を利用して統計的に求めることができる。以前の激指では、遷移確率を算出するのに、その指し手に関する特徴を 1 つだけ利用して算出するという、かなり荒っぽい手法を用いていたが、2、3 年ほど前から、ロジスティック回帰モデル（最大エントロピーモデルと呼ばれることもある）を利用して、指し手に関するさまざまな特徴量を総合的に使って算出する方式に変更している。

選手権に出場したバージョンでは、プロ棋士の棋譜約 4000 局の棋譜データを利用して学習を行っている。たとえば図-1 の局面を与えると、確率の高い順に以下のよう



な指し手が得られる(カッコ内が確率値)。

- ▲ 7 一角 (0.28)
- ▲ 3 四歩 (0.09)
- ▲ 5 三角 (0.09)
- ▲ 5 三銀 (0.02)
- ▲ 5 七飛 (0.02)
- ▲ 4 一銀 (0.02)
- :

完璧とは程遠いが、それなりに人間の直感に近い感じといえなくもなさそうだ。学習に使用した棋譜とは別の棋譜 100 局を使って、それぞれの局面において、最も確率値の高い手がどれくらい棋譜の指し手と一致するかを評価してみたところ、序盤で 47%，中終盤で 36% の一致率であった。盤面を見ただけ、つまりまったく探索を行わずにこれだけ正解できるのだからまずまずではないだろうか。もっとも、プロ棋士の場合、盤面を見た瞬間に有望な手を 2, 3 手に絞り込むことができるというから、人間の高度なパターン認識的な能力にははるかに及ばない。

学習に用いる指し手に関する特徴量としては、移動先、移動元の盤面情報 (3 × 3 の盤面パターン)、駒の種類、自玉・相手玉からの相対位置、駒の取り合い評価、あたりをかけている駒の種類などを利用している。学習の際には、重みの絶対値に関して線形のペナルティがかかるようにしているので、結果として実際に使う特徴量の数 (重みがゼロでない特徴量の数) はそれほど多くなく、10 万程度で済んでいる。

激指は、評価関数に関しては今のところ自動学習を採用せず、人手による設計に頼っている。理由の 1 つとしては、将棋の評価関数はなるべくシンプルかつ本質的な要素だけで表現したい、という個人的な思いがある。自動学習の場合、表面的な特徴量を多数組み合わせさせてプロの駒組みや指しまわしをまねるようにする、といった感じだが、コンピュータが意味を理解せずに形だけプロの駒組

みをまねる、というのがあまりしっくりこない。少数の本質的な要素だけで評価関数を作って、結果的にプロと似たような指しまわしになる、というのが個人的には理想だ。もっとも、そういう試みが激指でうまくいっているわけではないので、現実的には自動学習のほうがよいかもしれない。

定跡

現在の将棋プログラムは、序盤は定跡データベースを頼りにしていることが多い。そのため、コンピュータ同士の対局で、ずいぶんともな序盤を指していると思ったら、実は定跡データベースにあるプロの棋譜をお互いなぞっているだけ、ということがよくある。基本的には、局面を完全一致でデータベースから検索して、その局面が存在すれば、データベースに保存されている手を指す、というだけである。定跡データベース自体は、プロの棋譜などから自動的に作ることもできるし、小さなものを人手で作ることもある。定跡データベースを使って指している間は、お互い思考時間をまったく消費しないため、パタパタと局面が進んでいくので、見ていてすぐに分かる。場合によっては、そのまま終盤の入り口まで進んでしまい、実際にコンピュータが考え始める局面になったときには、大差がついていて将棋が実質終わっているということもあるが、こうなってしまうと興ざめである。

定跡データベースには頼るべきではないとして、YSS のように定跡データベースをほとんど使用しないプログラムもある。激指の場合は、今年の選手権では、定跡データベースの使用を強制的に最初の 20 手までで打ち切っている。

人間がコンピュータと対戦する場合、序盤早々に定跡データベースに存在しないであろう手を指すというのが有力な戦術の 1 つだ。そうすると、コンピュータが不得意な序盤を自力で考えなくてはならないため、人間が相対的に有利になることが多い。

並列探索

将棋プログラムは、原則的に、深く読めれば読めるほど強くなるため、探索の速度を向上させることは非常に重要だ。ハードウェアによる速度向上の手段としては、数年前までは、CPU 単体の処理速度が毎年数十パーセント向上していたので、単に最新のプロセッサに買い換えるだけでよかったが、最近では、クロックの向上が頭打ちになってきている。そのかわり、1 つにプロセッサの複数のコアを搭載した CPU が登場してきたことで、探索を並列化

	9	8	7	6	5	4	3	2	1	
香	桂								香	一
飛						金	王			二
歩		銀				桂	歩	歩	歩	三
	歩	歩		歩	金					四
										五
		歩	歩	歩	歩	王				六
歩	歩	銀	金				歩	歩		七
		金		飛						八
香	桂	玉							香	九

図 -1 △ 4 四金 まで



して高速化するアプローチが重要になっている。今年の選手権では、上位のプログラムの多くは 8 コアによる並列化を実装している。

並列探索に関するよくある誤解は、プロセッサの数を n 倍にすれば、探索速度も n 倍になるというものだ。

残念ながらゲーム木探索を並列化するのは、実はそう簡単ではない。ゲーム木探索では、アルファ・ベータ法による枝刈りのプロセスが不可欠で、基本的に、逐次的に探索を行った場合に、最も枝刈り効率が高くなるようになっている。そのため、並列化によるオーバーヘッドがたとえゼロでも、完全な台数効果を発揮させることはできない。仮に秒間あたりの探索ノード数を n 倍にすることができたとしても、実は無駄な探索が増えているので本質的な速度向上は n 倍よりも小さくなるのだ。

並列化の実装手法に関しては、Crafty というオープンソースのチェスプログラムの作者による手法³⁾を参考に行っているプログラムが多いようだ。Hyatt 自身は、かなり高い台数効果 (4 プロセッサで 3.7 倍、8 プロセッサで 6.6 倍、16 プロセッサで 11.1 倍の速度向上) を報告しているが、将棋プログラムでここまで並列効果が出ているケースは少ない。

将棋ハードウェア

今年の選手権の 2 次予選でひときわ注目を集めたのが、「A 級リーグ指し手 1 号」というプログラムだ。FPGA で作られた将棋専用チップを利用して探索を行っており、秒間 800 万局面を探索するという。激指が 8 コアを利用して秒間 100 万から 200 万局面程度なのと比べると相当に高速だ。コンピュータチェスでは、初めて世界チャンピオンを破ったのが、IBM の DeepBlue という専用ハードウェアを利用したプログラムだっただけに、将棋専用ハードウェアに対する期待は大きい。将棋は、チェスと比較して、

盤面情報が大きい、評価関数が複雑といった点があるため、チェスよりもハードウェア化する際の敷居はずっと高そうにみえる。将棋でここまで成功したというのは、開発者の技術力の高さの賜物だろう。「A 級リーグ指し手 1 号」は cyclone III という汎用の FPGA ボードを使用しているが、そのボードのクロックは 15.6MHz と、普通の CPU と比較するとかなり遅い。仮に ASIC などを利用して、もっと高速の回路で実装することができれば、それだけでも相当強くなりそうだ。将棋専用ハードウェアがまっさきに名人を破るというシナリオもあり得るのではないだろうか。

詰み探索

コンピュータの終盤が強いのは高性能な詰み探索アルゴリズムのおかげ、という認識は正確ではなかったかもしれない。将棋では、詰むや詰まざるやという勝負になることは実際にはかなり少なく、勝敗はそれ以前にはっきりしていることのほうが多いからだ。また、短手数の詰みであれば、通常の探索の中で自然に発見されることも多い。詰み探索のための処理は、通常の探索に比べるとはるかに高速であるとはいえ、探索中に数多く呼び出せば、それなりに思考時間を消費してしまう。下手に詰め探索ルーチンに計算リソースを割り振ってしまうと、肝心の通常探索のところが弱くなるために、実は詰み探索ルーチンをまったく使わないという選択肢も、総合的に見ればそれほど不利には働かない可能性もある。実際、2006 年の選手権で優勝した Bonanza は専用の詰みルーチンは搭載していないとのことである。

コンピュータ将棋の棋力

将棋の棋力といえば、昔は「段」「級」を使って表現することが多かったが、最近では、レーティングと呼ばれる指標を使うことが多い。大雑把にいうと、A さんが B さんに対して 2 勝 1 敗ペースで勝てるとすると、A さんは B さんよりも約 100 点高いレーティングを持つということになる。レーティングはあくまで相対評価であるため、母集団によってその基準が変わる。最近最もよく使われるのは、「将棋倶楽部 24」というインターネット将棋道場で使われているレーティングシステムだ。これだと、アマチュア初段で 1600 点前後、プロのトップクラスで 3000 点超といわれている。

将棋プログラムは、毎年レーティングにして約 100 点ずつ強くなってきた (図-2)。内訳は、大雑把にいうと、ハードウェアの高速化・並列化による上昇分が 50 点、ソフ

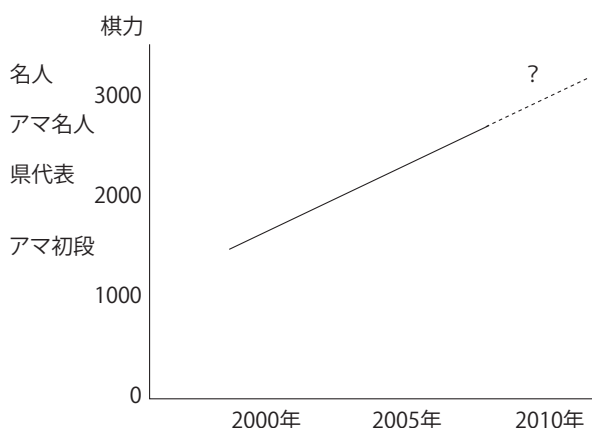


図-2 コンピュータ将棋の棋力



トウェアの進歩による上昇分が 50 点とっていいだろう。

2008 年の時点で、トップクラスの将棋プログラムは、いろいろな意見があるものの 2700 点以上はあるといわれている。もっとも、この数字はやや過大評価の可能性があるので注意する必要がある。1 つには持ち時間の問題で、将棋倶楽部 24 の対局はかなり早指しなので、人間が相対的に不利である可能性があることだ。

もう 1 つの大きな要因は、まだそれほど人間側がコンピュータの苦手な展開をよく知らないことだ。終盤はプロ以上といわれることもあり、中盤以降はコンピュータが非常に強い。特に切り合いになるような展開はコンピュータの最も得意とするところだ。したがって、人間側が勝つためには、序盤でしっかりと優位を確立する必要があるのだが、先に述べたように、特に高段者の場合、定跡をよく知っているがために、序盤を定跡どおりに進めてしまい、形勢互角のまま中盤に入ってしまうことが多い。これではコンピュータの思う壺である。

人間側が勝つための戦略として、序盤で定跡にない手を指して、定跡データベースを役に立たなくするというのは非常に有効だ。チェスとちがって、不利にならずに定跡をはずすことは難しくない。その後はとにかく派手な展開になることを避け、駒組みの優位をじりじり積み重ねていく指し方が有効だろう。コンピュータにとって、駒組みの優位性を認識することはむずかしく、コンピュータが自分の不利に気づくころには実際には逆転不可能な大差になっている、という展開が人間の有力な勝ちパターンの 1 つだ。

ところで、持ち時間に関してもう少しつけくわえると、先に述べたのとやや矛盾するが、名人戦のように 2 日ばかりで対局するような非常に長い持ち時間だと人間が相対的に有利になるかというと、そうでもないように思う。思考時間が短くなると、コンピュータも実はかなり間違えるのだ。少なくとも中盤以降になれば、探索すればするだけコンピュータの読みも正確になるため、持ち時間を長くすることで、それほど人間側が有利になるということもないだろう。個人的には、むしろ、2 日制の対局のように非常に持ち時間が長い場合、疲れることを知らないコンピュータ側にとって逆に有利なのではないかという気さえする。

効果をあげている。

これらの手法はもちろん、コンピュータ将棋の進歩という上では、少ない労力で強いプログラムが作れるようになったのだから、すばらしいことには違いない。ただ、一開発者としてみたときには、やや複雑な気分があるのも確かだ。昔は、評価関数も手作り、指し手の生成ルーチンも手作りで、その中には作者の将棋の知識がいっぱいに詰め込まれていた。そのため、個々の将棋のプログラムにはっきりとした個性があり、棋譜を見ただけで、どのプログラムが指した将棋か分かったものである。進歩にはちがいないのだが、職人芸の生きる場所が、1 つ 1 つ減っていくことに寂しさを覚えなくもない。

コンピュータ将棋の今後

前述のように、ハードウェアの進歩による棋力の向上は続いていくだろう。CPU のクロックが上昇すれば、それはそのまま棋力の向上に直結し、並列探索に利用可能なコア数が増えれば、台数効果を発揮させるのが難しくなるといえる問題はあっても、探索速度の向上に貢献することは確かだ。

ソフトウェアの進歩は、なんとも予測のしようがないが、小さな改良の積み重ねによって強くなる余地はまだあるし、新たなブレイクスルーとなる手法が出てくる可能性もある。プロのトップレベルに近づくためには、いろいろと解決すべき問題は存在しているが、コンピュータ将棋の棋力が、現状で頭打ちになると予測するのはさすがに無理がありそうだ。

参考文献

- 金子知通：兄弟節点の比較に基づく評価関数の調整，第 12 回ゲームプログラミングワークショップ，pp. 9-16 (2007)。
- 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，pp.78-83 (2006)。
- Hyatt, R.: The Dynamic Tree Splitting Parallel Search Algorithm, ICCA Journal, Vol.20, No.1, pp.3-19 (1997)。
- Tsuruoka, Y., Yokoyama, D. and Chikayama, T.: Game-tree Search Algorithm based on Realization Probability, ICGA Journal, Vol.25, No.3, pp.145-152 (2002)。

(平成 20 年 6 月 12 日受付)

コンピュータ将棋と機械学習

ここ 2, 3 年のコンピュータ将棋の進歩を振り返ると、なんといっても、大量の棋譜を利用した自動学習の成功が目立つ。前述のように、評価関数に関しては、Bonanza 式の学習が成功し、探索の制御に関しては、ロジスティック回帰モデルによる指し手の確率の学習が激指で

鶴岡 慶雅

yoshimasa.tsuruoka@manchester.ac.uk

マンチェスター大学 研究員。1997 年東京大学工学部電気工学科卒業。2002 年同大学院博士課程修了。工学博士。1999 年に激指の開発を開始。自然言語処理・テキストマイニングに関する研究に従事。