



最近のコンピュータ将棋の技術背景と GPS 将棋

金子知適 (東京大学総合文化研究科)

加速するコンピュータ将棋の進歩

2009 年の世界コンピュータ将棋選手権では、これまで上位入賞の経験のなかったチームが 1, 2, 3 位を独占するという予想外の結果となった。決勝リーグから一次予選にいたるまで参加プログラム全体の実力がはつきり向上しており、厳密な検証は難しいが、さまざまな棋風のプログラムが紙一重の実力差で並んでいる。同様に参加者の印象としては、使用ハードウェアは昨年とそれほどは変わらないため、この結果にはソフトウェア技術の進歩の貢献が大きいと思われる。トップレベルの実力の強さのプログラムの数が増えたことで競争が激化し、コンピュータ将棋は今まで以上の速度で強くなるだろう。本稿では、現在のコンピュータ将棋の技術的背景と、筆者が開発に参加している GPS 将棋について紹介したい。

探索して良い指手を探す

ほとんどのコンピュータ将棋やチェスのプログラムは、自分と相手のたくさんの指手を網羅的に探索し、自分が悪くならないように指手を決めている。この探索の対象である局面と指手からなるグラフをこの分野ではゲーム木と呼ぶ。図-1 は一手指すとゲームが終了する局面のゲーム木を表している。そのような状況は将棋では考えにくいですが、300 手目を指して詰みがなければ駒の損得で勝ち負け引き分けを決めるというルールで指したときの 300 手目を想像していただきたい。局面を節点で、指手を有向辺で表している。節点の丸と四角は手番を表し、それぞれ先手番後手番とする。根の局面 R が現在の局面で、そこには三手の合法手があり、それらを指すとそれぞれ局面 A, B, C に遷移するという状況を表している。それぞれの葉節点の肩の数値はゲームの勝敗を表す値である。ここでは 1 を先手の勝ち、0 を引き分け、-1 を後手の勝ちとしている。ここで根の局面が先手番なので、勝つためには A へ遷移する左の指手を選べば良い。

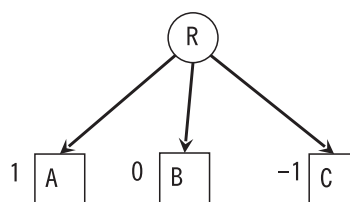


図-1 勝ち(1), 負け(-1), 引き分け(0)に関する一手の探索

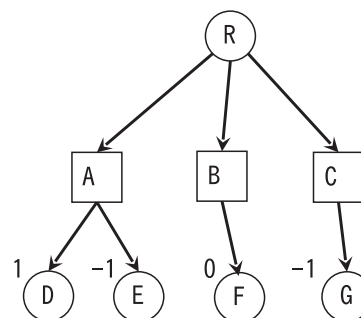


図-2 勝ち(1), 負け(-1), 引き分け(0)に関する二手の探索

図-2 は二手指すとゲームが終了する局面のゲーム木の例を表している。つまり自分が指した後に相手が指す状況である。ここで局面 C に遷移する右の指手を選ぶと相手に合法手は一手しかなく、局面 G に遷移し、先手の負けとなる。したがって(勝とうとする限りにおいて)この手を選ぶ意味はない。局面 B に遷移する中央の指手を選ぶと同様に引き分けとなる。これは堅実な選択である。局面 A では、相手に合法手が 2 つあり、1 つは先手の勝ちにもう 1 つは負けとなる。ここで先手が局面 A を選ぶ意味があるだろうか？ 相手がミスして局面 D に遷移すると予想されるならば、先手は局面 A を選ぶべきである。しかし、相手が正しく指すと予想されるならば、局面 A を選ぶと先手の負けにつながるため、局面 B を選ぶべきである。現在までの主流は、相

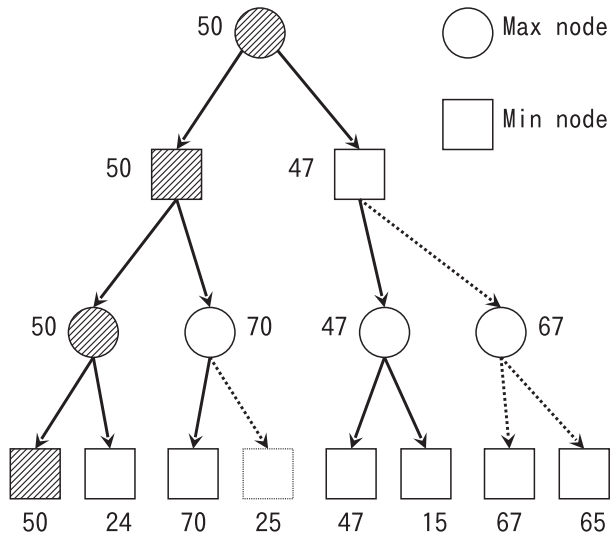


図-3 評価値を利用した三手の探索

手のミスに期待しない前提で探索を行う。その前提では図-2において、局面Aの勝敗は後手の勝ち(-1)、局面Rの勝敗は引き分け(0)となる。この計算は後手番では子の局面の値の最小値を、先手番では子の局面の値の最大値を取ることで得られる。そこで先手番、後手番の局面を、それぞれMax節点、Min節点と呼ぶ。またこのように合法手を列挙してその先の局面を展開し各節点の値を求める計算をMinMax探索と呼んでいる。2人/零和/確定/完全情報ゲームに分類されるゲームは、原理的には、MinMax探索で必勝法を求めることができる。

図-3はより現実的な三手の探索の例を表している。

まず、将棋やチェスでは、ほとんどの局面は三手以内で決着がつくことはない。そこで葉節点の数値として、ゲームの勝敗ではなく「評価値」を割り当てている。評価値は局面の形勢判断、つまり勝ちやすさを表すもので、ここでは数値が大きいほど先手が勝ちやすく値が小さいほど後手が勝ちやすいものとする。この場合の探索の目標は、相手がミスをしない前提で、最も数値が大きい局面が実現する指手を選ぶこととなる。実際に各節点で子節点のMax, Minを適切にとると図のような評価値になるため、先手はルート局面で左の指手を選ぶと良いことが分かる。図の中で斜線を示した節点はルートと同じ評価値を持つ節点であり、そのパスを最善応手手順(PV-Principal Variation)と呼ぶ。

MinMax探索以外の手法としては、モンテカルロ木探索が囲碁の世界で抜群の成功を取め、主流となっている⁵⁾。モンテカルロ木探索をコンピュータ将棋へ応用する研究も行われているが今のところトップレベルのプログラムの強さには達していない。

■ 深く読むと強くなる

さて、もし評価値に誤りがなく、たとえば符号が正ならば必ず先手勝ち、負ならば必ず後手の勝ちという性質があるとする、合法手の中で最も評価値が高い/低い指手を手番に応じて選べば良いため、図-1と同様に一手の探索で十分である。しかし現実には評価値は誤りを含み、結果として、なるべく深いMinMax探索を行う方が強いことが知られている(傍証の1つは後で述べる図-4の傾きである)。一応頭打ちというものがあり、深さを深くしてゆくと見返りが徐々に少なくなることがdiminishing returnsとして知られている。しかし、実際に強いプログラムを作る上では、一手でもより深く読むための努力が続けられている。

読みを深くする方法の1つは、単位時間あたりの探索局面数、すなわち探索速度を上げることである。まず、当然のことながら、良いハードウェアを使うことが望ましい。世界コンピュータ将棋選手権の参加者の使用ハードウェアは公開されており、上位の参加者は持ち運べる範囲で最新のコンピュータを毎年使っている。その上でハードウェアの性能を引き出す効率的な実装を行うことも重要である。搭載されている複数のCPU/コアを活用して並列に探索を行うと、さらに探索速度を向上させることができる。探索の並列化はさまざまな難しい課題を含むが、今年の決勝リーグではすべてのプログラムが並列化を行っているなど必須の技術となっている。さらに、メモリを共有しない複数の計算機が強調して探索を行う分散探索の研究も行われている。コンピュータ将棋選手権では今のところ主流ではないが、計算機の小型化が進んで設置の手間が問題にならなくなれば、将来は主流になっている可能性も高い。また、現状では多くの参加者は市販の計算機を使っているが、A級リーグ指し手1号という名前のFPGAを用いたプログラムが昨年から参加している。順調に強さを伸ばして今年も順位をあげていることから有効なアプローチであることは明らかであり、将棋専用ハードウェアの出現も楽しみな状況である。

■ メリハリをつけて読む

読みを深くするためのもう1つの着眼点は、相対的に重要でないと判断された分岐を浅く、相対的に重要と判断された分岐を深く読むことである。現実的なプログラムはこれらの手法を駆使するため、実際に探索される深さはパスによってさまざまに異なっている。まず理論的に読まなくてもよい局面が存在する。図-3のゲーム木において、点線で示した指手は探索の必要がない。点線の指手の先の節点の評価値をどのような値に変更しても、根節点の評価値が変わらないことを確認されたい。 $\alpha\beta$ 枝刈はこの性質を利用して探索を効率化する。

探索結果が変わるリスクを若干取ることで、さらに効率的な探索が可能になる。それらは、汎用的な手法とゲームの特徴を活かした方法に分けられる。ゲームの特徴を使う方法としては、王手や取り返しの場合は深く読むといった簡単なものから、「1つの局面で飛車角に利きをつける手の探索をn手までしか読まない。指手が多いときは盤上の駒が移動する手を優先させ、持駒の金銀を使う指手は優先度を下げる」といった凝ったものまである^{☆1}。このようにゲームの特徴を使って枝刈をする探索は「選択的探索」と呼ばれる。

汎用的な手法としては futility pruning, null move pruning, late move reduction, ProbCut などがある。これらの手法では、なんらかの方法で探索結果を予測してその分岐の重要性を判定する。たとえば null move pruning では、パスした上で浅い探索を行いその値が十分に良いものであれば、現局面にパスに代わる有効な手があると判断して探索を省略する。ゲームの特徴を使って枝刈をせず、汎用的な手法のみを使う探索は全幅探索と呼ばれる。

将棋では合法手が多いため、ゲーム依存の知識を用いた選択的探索が重要であると言われていた。しかし、Bonanza の活躍を受けて全幅探索の有効性が見直されている⁶⁾。もっとも全幅探索と呼ばれる Bonanza でも、王手を深く読むなどは行われているため、将棋の性質に依存した工夫がまったくないわけではない。

日本発の技術である激指の実現確率⁴⁾を用いた探索は、効果的であることが知られている。この手法では、指手が実際に実現する確率を用意する。そして、ルート局面からの現局面までの各指手の実現確率の積が、閾値を下回ると探索を打ちきる。したがって、実現確率の高い手が続く分岐をより深く、実現確率が低い手が続く分岐を浅く探索するといった効果がある。もし、すべての指手の実現確率を等しく揃えたとこの探索は、深さを打ち切基準とする従来の探索と同等になる。

実現確率を求める部分は、動かす駒の周囲、指手の履歴(連続して歩を叩いたなど)、指す駒に関する損得などの特徴を用いて棋譜を分析して求める。具体的な方法は各プログラムで異なるが、GPS 将棋では、まず指手の優先度をロジスティック回帰で求める。そして、各優先度の指手が棋譜で指された確率や、優先度の順に合法手を並べた時の各順位の指手が指された確率というものを用いて、実現確率に変換している。細かい点では、GPS 将棋は枝刈用と深さ調整用の2種類の実現確率を用いている。すなわち、ある局面である指手を読むかどうかには、局面の実現確率とその指手の枝刈用の実現確率を比

較する。そして、指した後の局面の実現確率を決める際には深さ調整用の実現確率を用いる。どちらの確率も各優先度や順位の指手が棋譜で指された確率から求めているが、枝刈用の実現確率は条件なしで、深さ調整用の実現確率はその局面でより高い優先度や順位の手が指されなかったという条件付きで測定している。この条件のために、深さ調整用の実現確率は枝刈用のものより高くなり、読むと決めたらある程度深く読むという効果を実現している。

将棋の形勢判断を学ぶ

探索の葉の局面の評価値を定める関数は評価関数と呼ばれる。その働きは人間で言えば形勢判断や大局観に相当し、評価関数をどう作るかは強さに直結する。そのため、プログラムの個性が発揮される要所であるとともに、試行錯誤、悪戦苦闘を伴う難所でもあった。近年、評価関数作成の仕上げを計算機で行う(以下、単に学習と呼ぶ)技術が広まりはじめ、今年も大きく進歩している。2009年の世界コンピュータ選手権では学習を採用したプログラムが上位を占め、従来通りに手作りで仕上げた評価関数を搭載したプログラムの成績は振るわなかった。学習が人手による調整に勝るかどうかはもう何年か見極めることが必要だが、現時点で学習が有望視されていることは間違いない。

図-3の説明で述べたように、評価関数が返す評価値は局面の勝ちやすさを数値化したものである。もし、誤りのない理想の評価関数であれば、評価値は勝ち負け引き分けを表す3値で十分である。しかし、現実には誤りを避けられないため、勝ちやすさや勝ちにくさの信頼性を評価値の絶対値の大きさと表現している。つまり、評価値が100点の局面と1000点の局面があれば、後者を選んだ方が勝ちやすいと判断して指手を決める。

実のところどのような評価関数が良いかは、あまりはっきりしたことが分かっていない。数少ない基準の1つは竹内らによる Evaluation Curve である³⁾。図-4はGPS 将棋を題材に、同じ評価値を持つ局面を集め、それらの局面の勝率(採取元の棋譜における勝者で判定)と評価値の関係をプロットしたものである。このようなグラフを描くと、グラフが右肩あがりのものが良い、凹凸がない方が良い、スケールが揃っているなら階段関数に近い傾きが急なものが良い、条件を変えた局面について複数のカーブを描いたときにグラフが重なるものが良いなどいくつかの必要条件を判定することができる。たとえば図-4で、評価関数の評価値と深さ8の静止探索を行った結果のルートの評価値を比較すると、後者の方がグラフの傾きが急になり評価値の信頼性が増している

☆1 「YSS 7.0」 — そのデータ構造、およびアルゴリズムについて
<http://www32.ocn.ne.jp/%7eyss/book.html>

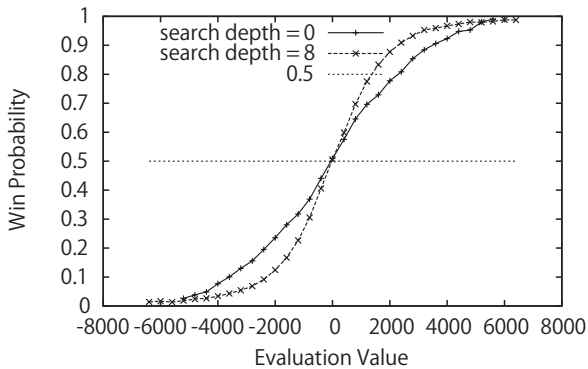


図-4 Evaluation Curve

と読み取ることができる。しかし、これらの基準を満たしたとしても強いとは限らないため、戦って強い評価関数が良い評価関数という基準が現実的には採用されている。そもそも、局面に数値を割り当てて比較できるという仮定は、将棋の高段者には違和感があるとも聞く。数値に代えて確率分布を割り当てる研究もありそちらの方が自然な印象を受けるが、現在までに強いプログラムでは採用されていない。

■ 局面のどの部分を評価するか？

将棋の形勢判断は、駒の損得、働き、玉の危険度等を評価するべきと言われている。簡単な例として、「 $100 \times$ 先手の歩の枚数 $+400 \times$ 先手の他の駒の枚数」という評価関数を考える。この関数は全体として先手の駒の枚数が増えるほど高い評価値となる。駒を取るプログラムならこれだけでできあがるが、本格的な将棋プログラムを作るためには、歩以外の駒の枚数は同一視して良いのか？とか、歩と他の駒の1枚あたりの価値は100点や400点が良いのか？といった点を手始めに改良の余地がたくさんある。後半の重みの調整を学習で行う場合は、開発者の努力は主に、前半の特徴の選定や改良に向けられる。

実際には駒の働き等の人間が指す場合の感覚を評価に組み込むことにはさまざまな困難がある。人に教える場合は「自分の玉が危険になる状況は避けるように」で済むところを、計算機に教える場合は「危険とは玉の2つ前に相手の銀がいることである」とか、学習を行わない場合はさらに「それは歩2.7枚分の損に相当する」などといちいち具体化する必要があるためである。一方で、制限時間内になるべく深く探索したいという要請もあるため、評価関数は同じ正確さであれば速い方が強い。つまり、凝った評価項目を組み込んでも、速度低下に見合う精度の向上がなければ弱くなってしまふ。速度と正確さのバランスを見極めることは非常に重要である。

Bonanza (バージョン4) では、3駒の関係の特徴に用

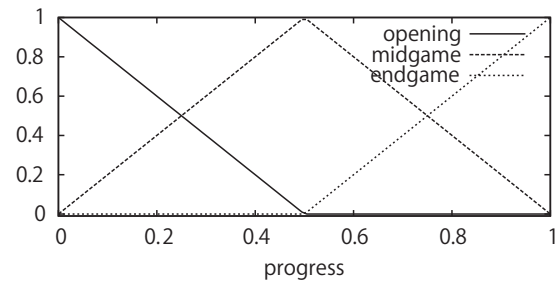


図-5 序盤・中盤・終盤の評価関数の混ぜ合わせ

いている。すなわち局面に成立する3駒の組合せそれぞれに値をつけ合計して評価値としている。3駒の関係すべてを扱うことは現実的でないため、必ず1つは玉を含み、持駒を含める場合は残りの2駒は玉とする等の制限を加えられてはいるが、それでも重み(パラメータ)の数は2億近くに達している。

GPS将棋の場合は、本稿末尾の付録1,2に掲載するように、人間が将棋を指す際の考え方に近い特徴を用いている。玉の回り25近傍に効いている攻め駒の枚数、それらのうちで攻方からの利きのある駒の枚数等、3駒の関係で表せないものも多い。特徴1つ1つは複雑な一方で、重みの数は300万程度、0でないものは80万程度とBonanzaよりはだいぶ少ない(2009年選手権参加時)。これらは、(1)特徴を考案してコードに書く(2)全体の重みを定め直す(3)元のプログラムと対戦させて勝率を見ろという手順を通じて有効と判断されたものが集められた。手順全体で1,2日かかるため、分散処理等による速度向上が課題である。

さらにGPS将棋では序盤・中盤・終盤の3つの評価関数を作り、進行度と呼ばれるゲームの進み具合に応じた重み付き平均を評価値としている。具体的には図-5に示した係数を用いて、序中盤は序盤と中盤の評価値の内分、中終盤は中盤と終盤の評価値の内分をとる。たとえば進行度が0.2であれば、評価値は $0.6 \times$ 序盤 $+0.4 \times$ 中盤となる。序盤・終盤の内分を取っていた従来手法よりも、中盤を増やした点で正確さが増していると期待される。実行速度に関しても、SSE2命令を利用して3つの値を効率的に扱っている。進行度という概念は多くの将棋プログラムで活用されており、さまざまな実装がある。GPS将棋では玉の周囲の攻め駒の利きや持駒の組合せ等を評価して、評価関数と同様の学習で重みを調整している。

なお、どのような特徴が適しているかは各プログラムの将棋盤のデータ構造にも依存する。Bonanzaの特徴は

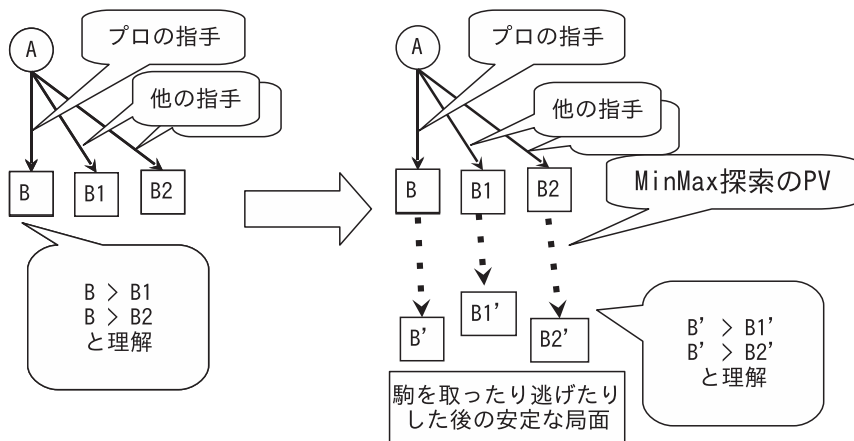


図-6 兄弟局面の比較(左:直接の比較, 右: PVを加味した比較)

駒の種類と位置を利用するため比較的移植が容易だが、GPS 将棋の特徴の場合は、直前の指手で利きが変化した駒やマス素早く知る必要がある。

ここでは Bonanza や GPS 将棋の 2 つを紹介したが、各プログラムでそれぞれ工夫を凝らした特徴を使っており、棋風の違いとして表れている。これらの特徴は現在は人手で作られているが、これらを自動的に生成し役に立つものを選び出すことは挑戦しがたい研究課題である。

■ 全体の調和をとりつつ数値化する

特徴の種類が決まったら、対応する重みを適切に定めれば評価関数が完成する。従来は重みを手で定めていたが、ここでは機械的に決める手法を紹介する。

評価関数の学習を行うにあたっては、教師をどのように設定するかが重要である。オセロでは、探索の結果(終盤は読み切りの結果の終局時のスコア(石の数の差 [-64,+64])、中盤以前は訓練済みの評価関数を使った探索結果)を教師とした最小二乗法による調整が行われ、すでに 1990 年代に圧倒的な効果をあげている¹⁾。しかし将棋の場合は終局直前に読みきっても勝ち負け引き分けの 3 値しか手に入らない上、詰み手順に入ってしまうば駒得も玉の危険度すらも関係なくなるという不安定な性質がある。そのため学習が難しく、最小二乗法による学習は成功しなかった。別の方法で、一局の勝敗から終局直前の評価値、一手先の評価値から現在の評価値へと評価値が揃う方向に伝播させる TD (-leaf) という方法がバックギャモンでは大成功を、チェスでも一定の成功を取っている。しかし、チェスでもトップレベルには達していないうえ、将棋では駒割(駒の損得)に自然な値がついたという報告はあるものの、それ以上の、たとえば囲いのような複雑な概念の学習に成功したという報告

はまだない。

将棋の学習では、プロ棋士など強いプレイヤーの棋譜を計算機に分析させ、評価値と棋譜の矛盾が少ない重みを探す方法が効果をあげている。棋譜のある局面 A においてプロがある手を指して次の局面 B になったとする。このとき、元の局面の他の合法手はプロが選ばなかった指手である。そこで、それらの局面 (B₁, B₂, ..) はプロが選んだ局面よりも悪いと考えて(図-6 左)、評価値がこれらの制約を満たすように重みを調整する。保木による Bonanza の学習⁷⁾では、指手のすぐ後の局面を比較する代わりに、簡単な探索を行いその末端局面を比較に用いている(図-6 右)。この改良は Bonanza の学習が成功した大きな要因の 1 つである。

具体的な重みの計算にあたっては、損失関数を定義し、その値を最小化する重みを求める。上述の手続きで得た局面のペアを $\langle f(a_i) > f(b_i) \rangle$ という教師と解釈する。f は評価関数、i はペアの番号、a_i や b_i は局面とする。簡単のため手番はすべて先手番で、評価値が大きいほど先手が勝ちやすい。そのうえで次の関数を最小化する：

$$\sum_i L(f(a_i) - f(b_i)) \quad (1)$$

ここで関数 L には図-7 のようにさまざまな候補が考えられるが、以下の理由で右肩下がりの関数が望ましい。L の引数が正である場合は、棋譜の指手を正しく(高く)評価しているため損失は 0 に近い。逆に負である場合は、望ましくない状況であり損失は高い。その点で最小二乗法で使われる二乗誤差は適切でないが、図-7 の他の関数は適切である。GPS 将棋では、実験の範囲で目立つ差は見られなかったため、ロジスティック回帰と同等の損失を採用している。ここでシグモイド関数を極端にしたような階段関数を使うと、上記の数式は

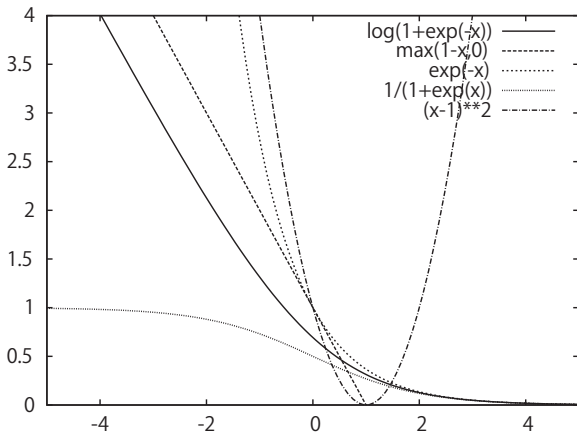


図-7 損失関数の候補

棋譜の指手よりも(誤って)高く評価した指手の数になる。保木⁷⁾は、この考え方の直接的な実現を指向して、シグモイド関数を用いている。その場合、棋譜の手を悪く評価した場合でもあまり損失が増えないため頑健性も向上すると思われる。

学習の目的は、未来の対局で出会った局面を正しく評価することである。人間に例えると、練習用に与えられた棋譜を一般化して理解して応用する能力が求められる。逆に、練習用の棋譜を丸暗記したが対局では役に立たないという状態を、機械学習の分野では過学習と呼ぶ。過学習を防ぐためには、まず、特徴の数と比べて多くの訓練例(この場合は局面のペア)を用意することである。4万棋譜を使うと約4億程度の訓練例ができるので通常は十分な数であるが、ここでは特徴の数も非常に多いためまだ問題が起こる。そこで、重みの絶対値の和や二乗和に比例した項目を式(1)の損失に加える l_1 正則化等の方法が採用されている。この項は、寄与が少ない重みを0にする方向に働く。

歴史的には、チェスのDeep Thoughtの開発者の1人が近い学習を試みていたことが後から公開された書簡で明らかになっている^{☆2}。主な違いは、最小二乗法を用いて $\sum_i (f(a_i) - f(b_i) - c_i)^2$ という関数の最小化を行った点である。そのため、局面のペア a_i と b_i の評価値の差を表す教師 c_i を用意する必要が生じ、その設定にかなり試行錯誤したことが読み取れる。また、正則化項もないため過学習の危険も高かったと想像される。実際に、後

継のDeep Blueではほとんどの重みは手で調整されたと報告されている²⁾。

強さをどのように測るか？

コンピュータ将棋の強さをどのように測定するかは難しい課題である。まず極端な例から考えると、仮にプロ棋士と対局しても勝てたとすると、ニュースを聞いた多くの方には強くなったと認めていただけるだろう。しかし、予想通りコンピュータが敗れたとして、100局指して100敗するのかそれとも1度くらいは勝てるのかは一発勝負では分からない。開発者としては数多くの対局を積みたいが、一方で、プロ棋士の方との対局機会は限られるとも予想される。1つの解として、2009年6月行われたGoogle杯将棋名人戦^{☆3}のように、職場名人対コンピュータプログラムの草の根の対戦が盛り上がることを期待している。

インターネット上で多数のプレイヤーが集うサイト(たとえば将棋倶楽部24)で対局をして実力を測ることも行われている。その際はコンピュータであると分かりやすい名前にした上で相手からの挑戦を待つことで、人間と対局したい人が間違えて対戦することを避ける工夫がされている。一方で、対戦相手が偏る危険も指摘されている。実際にコンピュータ囲碁の世界では、コンピュータとの対戦を避けるプレイヤーだけでなく、ほとんどコンピュータ相手としか指さないプレイヤーの一群が存在することが知られている。現在のところこのような対局は開発者の負担が大きいため、GPS将棋では行っていない。

コンピュータとの対局で実力を測ることは、比較的楽に行うことができる。改良前のプログラムと改良後のプログラムを対戦させて勝率を見ることはよく行われる。対局は自動で進行するため、全局終了した時点で結果を分析すれば良い。実際には、あるプログラムに強いプログラムが他の相手にも強いとは限らない。そのため、複数の相手と対戦の方が望ましい。

インターネット上でコンピュータ用対局場のfloodgateでは、15分切れ負けという早指しで30分ごとに一齐に対局が組まれている^{☆4}。メンテナンス等で臨時で止まることを除けば、24時間休みなく対局が続いていて、2008年の1年間では50,620局が行われた。作られて間もないプログラムにとっては、一局でも多くの対局を経験することが重要であり、打歩詰めや、玉で王手ができる局面等のさまざまな例外的な状況での動作を確認するテストとして有意義である。完成度の高いプログラムにとっても、プログラムの強さが相対的にどのあたりかを確かめたり、たくさんのデータを取ることで定跡の有利不利や戦形の得手不得手を確認めたりといったさ

☆2 http://www.tim-mann.org/DT_eval_tune.txt

☆3 http://googlejapan.blogspot.com/2009/06/google_29.html

☆4 floodgate という名前は当初は、毎時0分と30分に一齐に対局が流れ出す水門をイメージした開発用コードネームであった。

<http://wdoor.c.u-tokyo.ac.jp/shogi/floodgate.html>

まざまな活用法がある。ただし、公表している少数のプログラム以外のハードウェアの性能は不明であり、またそもそも本名(プログラム名)を名乗る義務もないため、ここで分かる強さはあくまで目安である。

コンピュータ同士の対局ならではの別の特徴として、評価値と読み筋がリアルタイムで記録されていることがあげられる。人間で言えば対局者が形勢判断を解説しながら対局するという(現実としては実現が難しい)状況に相当し、観戦の楽しみを充実させている。また、このような棋譜がたくさん生産されることで、将棋プログラムの棋風の違いを調べる研究等が発展することが期待される。

おわりに

最後に GPS 将棋自身について簡単に紹介する。GPS 将棋は、東京大学大学院総合文化研究科で開催されているゲームプログラミングセミナーの教員や学生が中心になって開発されているソフトウェアである。

GPS 将棋の名前は、game programming seminar からとられている。メジャーな略語に名前が重なることは避けるべきだったかもしれないが、global positioning system 将棋を開発予定という話も聞かないので、当然この名前が続きそうである。GPS 将棋の WWW サイト^{☆5}から Linux や Microsoft Windows 向けバイナリをダウンロードできる。また、オープンソースライセンスでソースコードを開発当初から公開している(修正 BSD の部分と GPL の部分がある)。興味のある方は試されたい。

開発者の 1 人は、コンピュータ将棋の開発者の役割を陸上競技のコーチに喩えている。人間の将棋プレイヤーはコーチと選手兼任なのに対して、コンピュータ将棋はコンピュータが選手で人間がコーチという対応である。少なくとも現在までのコンピュータはコーチなしでは将棋を指すことができない。評価関数の重みの自動調整や、指手の実現確率など計算機を用いて機械的に決められる部品は増えているものの、学習の方法や評価項目の設計、探索の作り方などコーチの役割の重要性は非常に大きくまた楽しいものである。興味を持たれた方にはコンピュータ将棋作りにぜひ参加されたい。GPS 将棋の名前で

☆5 <http://gps.tanaka.ecc.u-tokyo.ac.jp/gpsshogi/>



▽2四銀まで

図-8 初代 GPS 将棋の序盤

初めて世界コンピュータ将棋選手権に参加した 2003 年には、図-8 の局面のような将棋を指していた(先手番)。玉が単身で前線に展開する珍しいプログラムである。開発のある時点でこのような風変わりな出来であっても、継続的に開発することで強くなり得る例を示していることを期待している。

参考文献

- 1) Buro, M. : Improving Heuristic Mini-max Search by Supervised Learning, *Artificial Intelligence*, 134 (1-2), pp.85-99 (Jan. 2002).
- 2) Campbell, M., Joseph Hoane, J. A. and Hsu, F. : Deep blue, *Artificial Intelligence*, 134 (1-2), pp.57-83 (Jan. 2002).
- 3) Takeuchi, S., Kaneko, T., Yamaguchi, K. and Kawai, S. : Visualization and Adjustment of Evaluation Functions based on Evaluation Values and Win Probability, In *AAAI07*, pp.858-863 (2007).
- 4) Tsuruoka, Y., Yokoyama, D. and Chikayama, T. : Game-tree Search Algorithm based on Realization Probability, *ICGA Journal*, 25(3), pp.145-153 (2002).
- 5) 美添 : モンテカルロ木探索—コンピュータ囲碁に革命を起こした新手法, *情報処理*, Vol.49, No.6, pp.686-693 (June 2008).
- 6) 保木 : コンピュータ将棋の新しい動き : 3. 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 (Aug. 2006).
- 7) 保木 : 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 (Nov. 2006).

(平成 21 年 7 月 14 日受付)

金子知適 (正会員)
kaneko@graco.c.u-tokyo.ac.jp

東京大学大学院総合文化研究科助教。2008 年よりゲームプログラミングワークショップ共同プログラム委員長。

【序中終盤共通の値】

PieceEvalComponent 駒の価値

PiecePair 2 駒の関係、駒の座標に関係ないものと、X 座標固定と Y 座標固定

King25EffectAttack 玉の 25 近傍の攻撃側の利きの数 (max 127) とそこに利きを付けてる駒 (max 16) の組合せ

King25EffectAttackY King25EffectAttack を玉の Y 座標別に

ProgressBonus2 進行度による先後の進行度 (0 から 15 に正規化) の組合せ

ProgressBonusAttackDefense 各玉の、攻撃側の進行度と、防御側の進行度の値の組合せ

ProgressBonusAttackDefenseAll 両方の玉の、攻撃側の進行度と、防御側の進行度の値の組合せ

【序中終盤別の値】

PieceStand 持駒の種類別の枚数

Pin 動く和王手になってしまう駒を種類と玉からの相対位置別に

King25EffectEach 玉の 25 近傍の各枳で、どちらの駒があるか空白か、と、利きで勝ってるか負けてるか同じかの組合せ

PawnDropDefense 歩を打てる筋の自玉との X 軸の相対位置

PawnDropAttack 歩を打てる筋の相手玉との X 軸の相対位置

NoPawnOnStand 歩切れでかつ歩の枚数で負けているときの点

GoldRetreat 下がれない金を金の Y 座標別に

SilverRetreat 下がれない銀を銀の Y 座標別に

KnigtAdvance 跳べない桂を桂の Y 座標別に

AllMajor 大駒を全部持っているときの点

KingXBlocked 玉が左右どちらかに行けなときの点、玉の X 座標別

KingXBlockedY KingXBlocked を玉の Y 座標別に

AllGold 金を全部持っているときの点

PtypeX 各駒の種類につき、ある X 座標にいるときの値

PtypeY PtypeX と同様に、Y 座標別に

AnagumaEmpty 隅に居るときのまわりの 3 枳が空白かどうか

NonPawnPieceStand 歩以外の持駒の合計枚数

King25EffectDefense 玉の 25 近傍の防御側の利きの数 (max 127) とそこに利きを付けてる駒 (max 16) の組合せ

King25EffectYDefense King25EffectDefense を Y 座標別に

RookMobility 飛車が X 軸と Y 軸で動ける枳の数

BishopMobility 角が動ける枳の数

LanceMobility 香車が動ける枳の数

RookEffect 飛車の利きがある枳を相手玉との相対位置で評価

RookEffectDefense 飛車の利きがある枳を自玉との相対位置で評価

BishopEffect 角の利きがある枳を相手玉との相対位置で評価

BishopEffectDefense 角の利きがある枳を自玉との相対位置で評価

PawnAdvance 前に進めない歩を Y 座標別に

PawnDropYAttack 歩の打てる筋の相手玉からの X 軸の相対位置を玉の Y 座標別に

PawnDropYAttack 歩の打てる筋の自玉からの X 軸の相対位置を玉の Y 座標別に

KnightCheck 桂馬で王手がかかるとどうか

PieceKingRelativeBoth ある駒の盤上での各玉からの相対位置

NonPawnPieceStandCombination 歩以外の持駒の種類別の枚数の組合せ

NonPawnPieceStandTurn 歩以外の持駒の種類別の枚数を手番別に

King25EffectEachX King25EffectEach を玉の X 座標別に

King25EffectEachY King25EffectEach を玉の Y 座標別に

RookPawnY 飛車の Y 座標とその筋の歩の Y 座標

RookEffectPiece 飛車の利きが付いてる駒

BishopEffectPiece 角の利きが付いている駒

RookEffectPieceKingRelative 飛車の利きが付いている駒を玉からの相対位置で

BishopEffectPieceKingRelative 角の利きが付いている駒を玉からの相対位置で

RookPawnYX RookPawnY を玉からの X 軸の相対位置で

PawnPtypePtype 歩の前の駒とその前の駒の組合せ

CanCheckNonPawnPieceStandCombination 王手がかかりそうときの歩以外の持駒の組合せ

付録 1 GPS 将棋の評価関数、2009 年選手権出場時 (前半)



PromotedMinorPieces 相手玉と同じ側にある 2 枚目以降の小駒の成駒の相手玉からの X 軸の相対位置
PieceKingRelativeNoSupport 紐がついてない駒の各玉からの相対位置
NonPawnAttacked 相手からの利きがついている歩以外の駒を手番別に
PtypeYY 各玉の Y 座標別に駒の Y 座標で評価
PawnPtypePtypeY PawnPtypePtype を歩の Y 座標別に
PawnDropX 各玉の X 軸別に, 歩を打てる筋の X 軸の評価
King3Pieces 縦, 横, 斜めの玉を中心とする連続した三駒の評価
King3PiecesXY King3Pieces の玉の X 座標別のもとの Y 座標別のもの
King25EffectEachXY King25EffectEach の玉の X 座標別と Y 座標別のもの
BishopHead 角の頭に味方の利きがないときにそこにある駒
BishopHeadKingRelative BishopHead を自玉との相対位置で
KnightCheckY 玉の Y 座標別に桂馬で王手がかかるときの評価
KnightHead 桂の頭に相手が歩を打てそうなときに, 桂の Y 座標別に評価
RookPromoteDefense 飛車成りを受けている駒と, その駒に利きをつけている駒の種類
PawnDropPawnStand 持駒に歩があるときに, 各玉からの X 軸の相対位置で歩を打てる筋を評価
PawnDropPawnStandX PawnDropPawnStand を玉の X 座標別に
PawnDropPawnStandY PawnDropPawnStand を玉の Y 座標別に
King25Effect2 玉の 25 近傍の攻撃側の利き (max 63) と, 利きを付けている駒の数と, 持駒の飛車角金銀の数の合計の組合せ
King25EffectY2 King25Effect2 を玉の Y 座標別に
KnightHeadOppPiecePawnOnStand 桂頭に相手の駒があって, 相手の持駒に歩がある場合に, 桂馬の Y 座標と, 相手の駒の種類別に評価
KingXBothBlocked 玉が両側に動けないときに玉の X 座標別に評価
KingXBothBlockedY 玉が両側に動けないときに, 玉の絶対座標で評価
KingRookBishop 王と飛車の相対位置と王と角の相対位置の組合せ
PromotedMinorPiecesY PromotedMinorPieces を各玉の Y 座標別に
King25EffectSupported 玉の 25 近傍にある攻め駒の枚数と利きの付いてる攻め駒の枚数の組合せ
King25EffectSupportedY King25EffectSupportedY を玉の Y 座標別に
NonPawnAttackedKingRelative 歩以外の相手の利きがある駒を各玉との相対位置で
NonPawnAttackedPtype 歩以外の相手の利きのある駒を, 利きを付けている駒の種類別に
PtypeCount 駒の種類別に, 盤上にある数と, 持駒も含めて持っている数を評価
PtypeCountXY PtypeCount を自玉の X 座標別と Y 座標別に
PtypeCountXYAttack PtypeCountXY と同様, 相手玉版
LanceEffectPieceKingRelative 香の利きがついている駒を各玉との相対座標で
KingMobility 玉の絶対座標別に各向きに続いている空白棋の数
KingMobilitySum 玉の絶対座標別に, 各向きに続いている空白棋の数の総和
PtypeYPawnY 歩の Y 座標と, その筋にいる他の駒の Y 座標
GoldAndSilverNearKing 玉からある距離 (1, 2, 3) 以内の金と銀の枚数の総和
PtypeCombination 味方の駒の種類組合せ
PieceStandCombinationBoth 先後の持駒の持っている種類の組合せ
King25BothSide 玉の 25 近傍の Y 軸の攻撃側の利きの組合せ 2 つを, 玉を挟む形で組合せ (挟撃ボーナス)
King25BothSideX King25BothSide を玉の X 座標別に
King25BothSideY King25BothSide を玉の Y 座標別に
GoldAndSilverNearKingCombination 玉からある距離 (1, 2, 3) 以内の金と銀の枚数の総和の組合せ
KingMobilityWithRook KingMobility の相手の持駒に飛車がある場合
KingMobilityWithBishop KingMibily の相手の持駒に角がある場合
NumPiecesBetweenBishopAndKingSelf 相手玉が角筋にいる場合の間にいる玉側の駒の総和
NumPiecesBetweenBishopAndKingOpp 相手玉が角筋にいる場合の間にいる攻め側の駒の総和
NumPiecesBetweenBishopAndKingAll 相手玉が角筋にいる場合の間にいる駒の総和

付録 2 GPS 将棋の評価関数, 2009 年選手権出場時(後半)