

囲碁における決定木を使った着手の価値の学習

阿部 能明 小谷 善行

東京農工大学

概要

囲碁はゲーム木の平均分岐数が 200 を超えるため、全探索による着手決定は困難であり、さまざまなアプローチによって候補手を絞ることが行われている。過去には筆者は石の強度による候補手生成も行った。本稿では棋譜から集めたデータを使い、着手を学習することで人間の手に近い有効な手の選定を試みる。着手可能な空点(合法手)の周囲の特徴を、パターンを使わずに周囲にある石や着手の座標についての情報としてプロの棋譜から集め決定木を生成し、着手の価値を数値化する。決定木より候補手を生成し棋譜の着手と照らし合わせ評価を行った。

Learning for the value of moves using Decision Tree in Go

ABE Nobuharu KOTANI Yoshiyuki

Tokyo University of Agriculture and Technology

{ abe, kotani } @fairy.ei.tuat.ac.jp

Abstract

It is necessary in a game "GO" to choose candidates of moves for searching because the total number of possible moves of a position exceeds 200 on an average. We made candidate move generation by the intensity of a stone in the past. In this paper, we try selection of promising move in the way of human play by learning selected moves with the data collected from the record. We get the features around an empty point which can be selected (legal move), using not a pattern but the information about the stone which is in the circumference of selected move or the coordinates of selected move from a professional GO player's record. And we generate a decision tree and evaluate the value of move. An evaluation of our method is shown using correspondences between our system's optimal candidates and actual moves made by professional GO players.

1. はじめに

囲碁では全探索によって着手を決定することは困難である。それはゲーム木の平均分岐数が 200 を越えることが原因である。特に序盤から中盤にかけてはさまざまなアプローチで候補手を絞っている[1]。筆者も過去に石の強度を測定し候補手生成を試みている[2]。また、コンピュータ囲碁が将棋やチェスほど強くない理由の一つに、人間の打ち手の思考を解析できないという問題がある。しかし人間の思考をプログラミング化しなくても、過去の記憶(データ)から有効な候補手を生成することは可能である

[3]。そこで、筆者は過去の研究を踏まえて、棋譜から着手に関するデータを獲得して決定木を生成することで、着手の価値の学習を試みる。本研究では着手の価値を判定し得るいくつかの情報を設定し、プロの棋譜からそれらの情報を採取して決定木を生成する。そしてその決定木の出力から候補手を生成して棋譜の着手と照合する。

2. 学習する着手

人は着手を決める際にさまざまなことを考慮する。例えば相手の大石を捕獲できるか、自分

の捕獲されそうな石が逃げられるか、自分の地や相手の地を増減するかなど、目的がはっきりしているもの。攻めの手、守りの手、消し、利かしといった明確な目的がわかりにくいものがある。

前者の方は捕獲探索などでコンピュータに理解させればよいが、後者の方は概念的な要素が加わり、人によっては意見が異なることもあって、プログラミング化することは困難である。そこで、ある着手に対して「どういう意図があるかは不明だが、着手としては質が高い」ということを着手の価値として数値化する。これにより、人間の思考や意図といったことを考慮せずに有効な着手が得られる。

3. 周囲の状況の情報

人間も明確な意図をもたない手を打つときがある。このときに考慮するのが着手の周囲の状況である。周囲の状況とは、着手の位置や周りにある石についてのさまざまな情報である。この周囲の状況を情報として獲得し、着手の価値の判断に利用する。今回は着手の周りは四方と限定し、四方にある石は各方向最近のものだけを考える。また、四方を見ると、石と石の間を通り抜けてしまうのを防ぐために各方向1路ずつの幅を持たせて調べる(図1参照)。

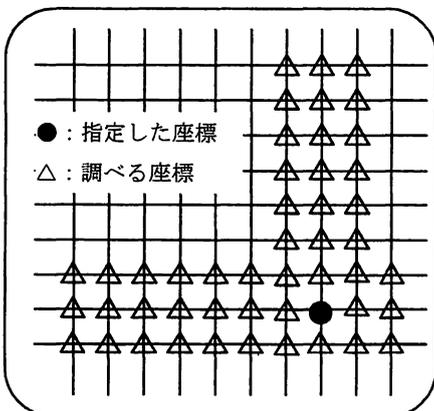


図1：調べる四方の範囲

四方にある石について調べるとき、1つの石の塊としてコスミ(斜め)でつながる連(縦横に並んだ石)を考える。以後これをコスミ連[2]と呼ぶ。考えられるコスミ連の情報に表1が挙げられる。周りの石について以外にも、表2の要素を考える。これらの要素を総じて周囲の状況と呼び、着手の価値の測定基準とする。

表1：コスミ連の情報

イ)	色
ロ)	構成する石の数
ハ)	縦幅
ニ)	横幅
ホ)	ダメの数
ヘ)	断点の数
ト)	眼型(相手からの禁手)の数
チ)	まわりの空きの数
リ)	まわりの自分の石の数
ヌ)	まわりの相手の石の数
ル)	まわりの盤端の数
ヲ)	最大高さ
ヾ)	最小高さ
カ)	中央高さ

表2：その他の情報

い)	手番
ろ)	着手の高さ
は)	座標のポテンシャル値
に)	四方にある石までの距離
ほ)	手数

各情報の中で説明が必要とされるものを次に示す。

ホ) ダメとは石が接している空点の数のこと。ダメが0になるとその石は捕獲されたことを意味する。ヘ) 断点とはコスミでつながる個所に現れる接点の一方が、相手の石によってふさがれている時の他方の点を指す。切断の可能性などが関係してくる。チ) ～ル) まわりとはコスミ連の各石からマンハッタン距離2までの点である。ヲ) ～カ) 高さとは、盤端からの距離であり、19路盤では1～10となる。は) ポテンシャル値とは周囲の石から受ける影響力の一種である[4]。

なお、コスミ連の死活については考慮に入れない。死活判定は石が多すぎても少なすぎてもできないからである。また、囲碁の対称性を考慮し、盤面を8分割して考える。

4. 決定木学習

帰納学習法の中に決定木の学習がある。決定木とは情報についての質問を繰り返し、結果を出力とする木のことであり、周囲の状況という情報を扱う本研究に適しているといえる。本研究では学習には決定木学習を使う。決定木生成アルゴリズムはID3[5]を使用する。データ集合 C に対して次の動作を行うものである。

- a) 集合 C 中の全データが同一クラス→そのクラスノードをつくり、停止する。そうでなければ属性の選択基準により1つの属性 A を選んで判別ノードを作る
- b) 属性 A の属性値により C を部分集合 C_1, C_2, \dots, C_n に分けてノードをつくり、属性値の枝を張る。
- c) それぞれのノード C_i ($1 \leq i \leq n$) について、a)、b)へ再帰的に適用する。

3. で示した周囲の状況の全組み合わせに対して人手で着手の価値を設定するのは現実的でない。よってこれらの情報を棋譜から集め、決定木を生成する。決定木の属性(木の非末端ノードで集合に対して行う質問)は3. で示したイ)～カ)、い)～ほ)であり、クラス(決定木の出力)は着手の価値を0(プロは絶対打たない)～1(プロは絶対打つ)で表す。プロの棋譜1手毎に盤上の全合法手1つ1つに対して周囲の状況と、実際の着手であるかどうか(0,1)を1組のデータとして獲得する。

ID3では末端部に到達してもクラスが同一でない場合や、質問がなくなったときに、その部分集合のクラスは多数決によって決めるのが一般的であるが、本研究ではクラスが着手であったデータの割合を返す。この値を着手の価値の数値とみなす。

5. 実験

実際に決定木を生成し、その木の妥当性を測定する。比較対照には線形一次和を使用する。本研究では序盤だけを考えるので、データは1～50手から獲得する。

5.1 実験方法

決定木と、線形一次和を使用して学習を行いそれぞれによる出力値を着手の価値として候補手を生成して実際の棋譜の着手と照合する。

5.1.1 決定木

決定木はプロの棋譜[6]90局からデータ1,500,000個を獲得して生成した。決定木の縮小化と無駄な学習を避けるため次の条件を加えてある。

- ・集合が100以下になったら末端とする
- ・木の深さは6まで、すなわち質問は5回までとする

5.1.2 線形一次和

入力(周囲の状況) $x \in R^n$ に対する出力(着手であるかどうか)を $f(x) \in R$ とするとき、観測データ $(x^1, f(x^1)), \dots, (x^m, f(x^m))$ に対する線形近似関数 $a \cdot x + b$ を解くことで線形一次和とする。 a と b は最小二乗法によって求める。決定木と同様にプロの棋譜48局からデータ800,000個を採取してネットワークにかける。

5.2 評価基準

学習に用いた棋譜とそうでない棋譜1～50手まで1手ずつ読み、すべての合法手についてその点の着手の価値を出す。価値が高い順に候補手として、実際の着手が何個目に現れるかを10手ごとの平均で判断する。また、候補手の上位20手以内に着手が現れた割合も測定する。

5.3 結果

実験した結果を表 3、表 4 に示す。A は棋譜の着手と一致するまでの候補手の数、B は候補手の上位 20 手以内に着手が現れた割合である。訓練データ、テストデータともに学習時と同じデータ数で試行している。

表 3：決定木による候補手の性能

手数	着手が現れるまでの数		上位20に現れる割合	
	訓練データ	テストデータ	訓練データ	テストデータ
1~10	22.41	27.05	0.66	0.65
11~20	31.71	41.63	0.55	0.48
21~30	29.82	53.86	0.61	0.46
31~40	34.49	65.59	0.57	0.40
41~50	33.13	75.20	0.58	0.34
1~50	30.30	52.63	0.59	0.46

表 4：線形一次和による候補手性能

手数	着手が現れるまでの数		上位20に現れる割合	
	訓練データ	テストデータ	訓練データ	テストデータ
1~10	181.59	185.50	0.00	0.00
11~20	189.93	172.00	0.00	0.00
21~30	185.58	168.5	0.00	0.00
31~40	175.42	164.61	0.01	0.01
41~50	171.01	156.50	0.01	0.01
1~50	180.769	169.502	0.00	0.01

6. 考察

決定木の方が精度が格段によいが、未知の局面に上手く対応できていない。しかし、参考に行った深さの限定をしない決定木での実験結果(表 5)と比較すると、深さを限定することである程度の一般化をはかれている事がわかる。学習に使用した棋譜が 90 局と少ないのでさらに増やしてみる必要がありそうだ。

表 5：深さを限定しない決定木

手数	着手が現れるまでの数		上位20に現れる割合	
	訓練データ	テストデータ	訓練データ	テストデータ
1~10	27.28	95.37	0.88	0.46

生成した決定木をみると、最初の質問は着手の高さであった。このことから着手を決める際に高さが重要なものであることがわかる。

線形一次和は 1~50 手の平均可能手の数が約

300 であることを考えると、ランダムと同程度であると考えられる。これは、着手であるデータと着手でないデータの比が 100 倍を超えていたため、着手でない出力を常に出すような収束におちいったためであると考えられる。このようなデータ群を扱う場合は決定木の方がより適していることがいえるだろう。

7. 今後の課題

属性値の振り分け方や属性の削除を検討してみたり、学習に使う棋譜をもっと増やして決定木を生成する。

評価方法に「次の 1 手」等の問題を使って他の手法との比較を行ってみる。

8. まとめ

本稿では周囲の状況を使って着手の価値の決定木学習を行った。生成した決定木による候補手は、平均 52.6 個でプロの着手と一致し、20 個以内に一致した割合は 0.46 であり、線形一次和で求めた候補手よりはよいものであることを示せた。

参考文献

- [1] 松原 仁, 竹内 郁雄 : 「ゲームプログラミング」, 共立出版, pp.59-70, 1998.
- [2] 阿部能明, 小谷善行 : 「囲碁における決定木を使った連の強度の学習」, ゲーム情報学研究会第 5 回研究会, 東京大学, 情報処理学会研究報告, 5-3, pp1-8, 2001.
- [3] 清 慎一, 川嶋 俊明 : 「記憶に基づく推論を使った囲碁プログラム「勝也」の試作」, ゲーム・プログラミングワークショップ'96, pp.115-122, 1996.
- [4] 実近憲昭 (電総研), 清慎一, 赤尾杉隆, 瀧和男, 吉川貞行, 吉岡勉, 村沢好信, 内田俊一, 沖廣明, 大島正行, 小木曾督, 水野勇一, 坂本淳 : 囲碁システム「碁世代」の仕様, 1991.
- [5] J.R.Quinlan : Induction of decision trees, *Machine Learning*, 1, 81-106, 1986.
- [6] BGA Software Catalogue, <http://www.britgo.org/gopres/gopres1.html>