

囲碁の数理モデル化とその応用

佐藤 真史^{†1} 穴田 浩一^{†2} 堤 正義^{†1}

囲碁は、ボードゲームの中で特に高い複雑性を持つ興味深いゲームである。チェスや将棋といった類似したゲームに用いられた手法だけでなく、まったく別の分野からのアプローチなど様々な方法が考案、試用されている。本論文では、基本に立ち帰って、囲碁そのものを数学を用いて表現するモデルを提示する。

A Mathematical Model for the Game of Go and its Application

MASAFUMI SATOU,^{†1} KOICHI ANADA^{†2} and MASAYOSHI TSUTSUMI^{†1}

Among many board games the game of Go is the most interesting and challenging one for researchers because of its complex nature. In recent years many new approaches from various fields such as morphology, pattern matching, and Monte carlo simulation have been developed and applied as well as classical ones which had applied to Chess and Syogi. Our purpose of this paper is to provide a mathematical model for the game of Go.

1. 最初に

囲碁は、その選択可能な手の多さと有用な静的評価がゲームの大域的構造によってうまく作れないことなどから、チェスや将棋などで用いられてきた従来のツリーサーチによる方法は十分な効果を発揮できず、ツリーサーチに代わるものとして、既存の定石をデータベースとして用いるもの以外にも、終局直前のヨセに関する組み合わせ理論を用いた方法¹⁾や、距離による評価法²⁾、外部活路、CFG³⁾といったグラフ表現の利用などが考案されている。その多くは局面を部分局面と呼ばれるいくつかの領域に分けることで、計算量の削減を行っている。

一方で最近では、ツリーサーチを用いるモンテカルロ法が大勢を占めている。モンテカルロ碁はとにかく終局まで打ち(プレイアウト)、静的評価を用いてその経路の評価をしていくものであり、Coulom⁴⁾の Crazy Stone をもってその有効性が証明された。さらにUCT(UCB applied to Trees)を用いた、各探索ノードでの着手に優先順序という重みをつけることで、その精度を向上させた(Gelly et.al.⁵⁾など)。

しかし1対局での結果を別の対局への応用する際には、結果的に経験則を利用せざるを得ないことが多い。というのも、囲碁局面の特徴がチェス以来のx-y座標表示では、表現しきれないからである。

囲碁の局面を数学的にモデル化する試みは、点の状態による表現⁶⁾やビットボードによる隣接の定義⁷⁾、後で定義する連を主体としたグラフによる表現方法³⁾などで行われているが、定理自体が数学的に表現できるとは言えない。

本稿では囲碁の数理モデルとして、BWモデルを提唱する。このモデルは、局面が方形に縛られないこと、評価関数の構成要素が明確になること、グラフ化を通してに盤上の向きや角度という囲碁では用いられない概念から開放されるといった利点が挙げられる。

本発表では、まず、2節で囲碁の数理モデル化として2ベクトル1行列による表現を提案する。これは、8)で提案されたものを改良したもので、コウや手番のような盤面上に現れない情報を除いた、局面=石の配置を表現するものとして、様々な囲碁的評価、用語を明確に定義することができる。3節でこのモデルの特徴とその利点を用いた応用例を提示する。最後に4節で結論を述べる。

2. 提案する囲碁の数理モデル

本節では、佐藤¹⁰⁾や佐藤、堤⁸⁾を発展させたモデル“B-W Graph Model”を提示する。このモデルは、

^{†1} 早稲田大学

Waseda University,
School of Science and Engineering
Department of Applied Mathematics

^{†2} 早稲田大学高等学院

Waseda University Senior High School

局面を黒と白それぞれに関わる2つのグラフで表し、その点と辺にブール代数を用いた演算を施すことによって、着手可能な点や着手による局面の変化といったルールそのものから各種評価関数までを統一的に表現出来るようにするものである。

2.1 演算の定義

まず n を盤上の交点の総数とする。また本稿ではベクトルは全て n 次元 0-1 ベクトルとし、零ベクトルを $\vec{0}$ 、単位ベクトルを $\vec{e}_1, \vec{e}_2, \dots, \vec{e}_n$ 、全ての要素が1のベクトルを $\vec{1}$ とする。

ブール代数からの自然な拡張として、反転

$$(\vec{a})_i = \begin{cases} 1 & (\vec{a})_i = 0 \\ 0 & (\vec{a})_i = 1 \end{cases}$$

を考えることができ、次いで論理積、論理和、論理差、排他的論理和を

$$(\vec{a} \wedge \vec{b})_i = (\vec{a})_i \wedge (\vec{b})_i, \quad (\vec{a} \vee \vec{b})_i = (\vec{a})_i \vee (\vec{b})_i, \\ \vec{a} \setminus \vec{b} = \vec{a} \wedge \vec{b}, \quad (\vec{a} \oplus \vec{b})_i = (\vec{a})_i \oplus (\vec{b})_i$$

と定義する。行列もまた $n \times n$ の 0-1 行列とし、同様の演算を考える。そして、ベクトル、行列の掛け算を

$$\vec{a} \cdot \vec{b} = \bigvee_{i \in I} ((\vec{a})_i \wedge (\vec{b})_i) \\ (A\vec{a})_i = \bigvee_{j \in I} ((A)_{ij} \wedge (\vec{a})_j) \\ \vec{a}A = {}^t\vec{a}A \\ (AB)_{ij} = \bigvee_{k \in I} ((A)_{ik} \wedge (B)_{kj}) \\ (\vec{a} {}^t\vec{b})_{ij} = (\vec{a})_i (\vec{b})_j$$

とする。演算順序は転置と否定、掛け算、論理演算、同格のときは左からの順とし、単位行列を E とする。

2.2 局面の定義

次に、黒石のある点を「黒」、白石のある点を「白」、何も無い点を「空」と書くことにすると、このとき n 次元ベクトルの各要素は盤上の各点と一対一対応させることが出来る。特に、二つのベクトル \vec{b} と \vec{w}

$$(\vec{b})_i = \begin{cases} 1 & i \text{ is black or empty} \\ 0 & i \text{ is white} \end{cases} \\ (\vec{w})_i = \begin{cases} 1 & i \text{ is white or empty} \\ 0 & i \text{ is black} \end{cases}$$

は重要な役割を果たす。

ここでは、囲碁における隣接関係 N を

盤上の2点 u と v が同じまたは隣接しているとき、またそのときに限り uNv が成り立つ、とし、行列 F_0 を

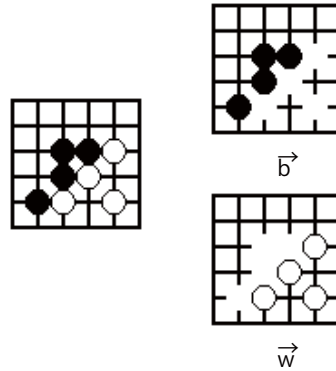


図1 局面を表す二つのグラフ
Fig.1 Two graphs \vec{b} and \vec{w} on a position.

$$(F_0)_{ij} = \begin{cases} 1 & iNj \\ 0 & \text{otherwise} \end{cases}$$

と定義すると、 \vec{b} と \vec{w} は、それぞれ図1の右上と右下の図の点、 F_0 は両図の点同士のつながり、即ちグラフの辺を表す。また使用頻度が高いので、空である点を $\vec{l} = \vec{b} \wedge \vec{w}$ と書くことにする。

2.3 着手の定義

囲碁では、各局面ごとに線を通じて隣接している同色の石は連と呼ばれ、1つにまとめられる。

連関係 M を

各局面において、盤上の2点 u, v が同じ連に含まれる、もしくは同じ点であるとき、またそのときに限り uMv が成り立つ、

と定義する。このとき、隣接 N が、反射律と対称律のみを満たすのに対し、連関係 M は推移律も満たすため同値関係となっている。以降では連という単語を M による商集合、すなわち石の作る連結集合、および空1つからなる集合に対して用いる。

これらを用いて、各局面ごとに行列 F を次のように定義する。

$$(F)_{ij} = \begin{cases} 1 & \exists k; (iMk \wedge kNj) \\ 0 & \text{otherwise} \end{cases}$$

すなわち、点 i を含む連に j の隣接点が含まれるとき、およびそのときに限り、 $(F)_{ij} = 1$ である。点 i が空であるならば、 j は i の隣接点であり、任意の2点 i, j において

$$(F_0)_{ij} = 1 \Rightarrow (F)_{ij} = 1$$

が満たされる。 $(F)_{ij} = 1$ であるとき j を i の接続点と呼ぶことにする。

以降では、下付添え字 t が付いている変数は、 t 手

目が着手されたあとの局面から生成したものとする。
ベクトル \vec{h}_t を次のように定義する。

$$(\vec{h}_t)_i = \begin{cases} 1 & |({}^t F_t \vec{e}_i) \wedge \vec{l}_t| = 1 \\ 0 & \text{otherwise} \end{cases}$$

これは $(\vec{h}_t)_i = 1$ のとき、点 i にはアタっている石があるか、石に囲まれている空の点であることを示すベクトルである。 $\vec{a}_t(i)$ と $\vec{d}_t(i)$ を

$$\vec{a}_t(i) = \begin{cases} ({}^t F_t \vec{e}_i) \wedge (\vec{b}_t \setminus \vec{l}_t) \vee \vec{e}_i & t \text{ is even} \\ ({}^t F_t \vec{e}_i) \wedge (\vec{w}_t \setminus \vec{l}_t) \vee \vec{e}_i & t \text{ is odd} \end{cases},$$

$$\vec{d}_t(i) = \begin{cases} ({}^t F_t \vec{e}_i) \wedge (\vec{h}_t \setminus \vec{b}_t) & t \text{ is even} \\ ({}^t F_t \vec{e}_i) \wedge (\vec{h}_t \setminus \vec{w}_t) & t \text{ is odd} \end{cases}$$

と定義すると、 t 手目で着手された点 (着点) を i とすれば、次の局面における \vec{b} , \vec{w} , F は、

$$\vec{b}_t = \begin{cases} \vec{b}_{t-1} \vee \vec{d}_t(i) & t \text{ is even} \\ \vec{b}_{t-1} \setminus \vec{e}_i & t \text{ is odd} \end{cases},$$

$$\vec{w}_t = \begin{cases} \vec{w}_{t-1} \setminus \vec{e}_i & t \text{ is even} \\ \vec{w}_{t-1} \vee \vec{d}_t(i) & t \text{ is odd} \end{cases},$$

$$F_t = F_{t-1} | (\vec{a}_{t-1}(i) \wedge \vec{a}_{t-1}(i) \wedge {}^t F_{t-1}) \setminus (F_0 \vec{d}_{t-1}(i))$$

と書け、 F_0 は着手による変化は無い。

$\vec{a}_t(i)$ は着手により新たに作られる連であり、着点を連結点とした隣接していた既存の先手の連の結合したものである。また $\vec{d}_t(i)$ は、着手により取られる後手の連を表す。

\vec{b} と \vec{w} の変化は、それぞれ取られた連と石に変わった連で表される。一方、 F の変化は2種類に分けられる。一つは $\vec{a}_t(i)$ による連の結合で、点 j が $(\vec{a}_t(i))_j = 1$ のとき、いままでの接続点に加え、新たに連関係になった点を経由するものが増える。もう一つは $\vec{d}_t(i)$ による連関係の初期化であり、石から空に変わること、その点の接続点は隣接点と等しくなる。このように囲碁の局面は連の拡大と初期化の繰り返しで表される。

また石のない全ての点が空である局面を初期局面と呼ぶことにすると、そのときの \vec{b} と \vec{w} は

$$\vec{b}_0 = \vec{w}_0 = \vec{1}$$

となり、 F は F_0 と一致する。ただし \vec{b}_0 と \vec{w}_0 は任意の局面において、

$$\vec{b} \vee \vec{w} = \vec{1}$$

が成り立つため、 \vec{b}_0 , \vec{w}_0 は \vec{b} , \vec{w} の初期局面での値という意味しか持たない。一方で、 F_0 は盤の形状というルールを表す変数である。

「局面」を表すだけならば、 F_0 を定数として石の配置を表す \vec{b} , \vec{w} のみで表すことが出来る。しかし、これらの情報のみから連の形を得ようとすると、各2点間の隣接関係によるルートの探索が必要となる。一方で連関係を満たす2点は、取られない限りまったく同じ扱いを受ける。そのため、局面ごとに連の形状を再計算するのではなく、点を \vec{b} , \vec{w} , 辺を F とする2つのグラフの変化として、対局を表現することで計算頻度を減らす。 F は対称行列ではないため、2つのグラフは有向グラフとなり、辺は各点からその接続点へ向きの、点と連の関係を表す情報として働く。

局面を \vec{b} , \vec{w} , F 及び F_0 の4つの値で表すこと、及び着手に伴う変化を、”B-W graph model“と呼ぶことにする。ただし対局中に F_0 が変化することはないため、混乱の恐れがない限り、局面を (\vec{b}, \vec{w}, F) という形で表記することにする。

本モデルでは、対局者の打てる点 (着手可能点) は

$$\begin{cases} F_t(\vec{b}_t \oplus \vec{h}_t) \wedge \vec{l}_t & t \text{ is even} \\ F_t(\vec{w}_t \oplus \vec{h}_t) \wedge \vec{l}_t & t \text{ is odd} \end{cases}$$

と表現できる。空点=着手可能点としない理由は、自殺手を可能とすると着手による \vec{b}, \vec{w}, F の変化が、(局面の状態に依存する) 場合分けを含む複雑なものになるゆえであり、対局者は自分の石が自分の着手によって取られることの無い空点に打つことができる。というルールの定式化でもある。一方コウに関しては、本モデルでは最も価値の低い着手として扱う。

\vec{h} の一般形として、

$$\vec{p}_t(k)_i = \begin{cases} 1 & |{}^t F_t \vec{e}_i \wedge \vec{l}_t| < k \\ 0 & \text{otherwise} \end{cases}$$

という関数を定義すれば、 \vec{h} は $\vec{p}_t(2)$ と書ける。関数 \vec{p} を数え上げ関数と呼ぶことにする。

3. 応用例

本節では、BW モデルの特徴およびその利点について説明する。1つ目は実際に Crazy Stone で用いられている評価関数を例とした本モデルによる既存の解念の定式化、次にグラフ化による部分局面の自由化、最後に定式化したことで分かった必要な演算とそれによる学習システムの提案をする。

3.1 Crazy Stone

Coulom⁴⁾ においては、緊急度 (Urgencies) と無駄な手 (Useless Moves) というパターンを用いて着手の優先順位を定めている。今回その”緊急度”および、その定義に用いられる”絶望的な拡大”を我々のモデル

を用いて表す。

”絶望的な拡大”は「隣接した空の交点は高々1つ」, 「隣接した白はアタリ状態ではなく」, 「隣接したアタリ状態でない黒の連は無い」という3つの条件で表されるが, これらはそれぞれ

$$\overline{\overline{p_t(2), F_t(\vec{h}_t \setminus \vec{b}_t), F_t(\vec{h}_t \setminus \vec{w}_t)}}$$

と表せ, ”緊急度”の増加分は次のように計算できる。

- (1) $F_t(\vec{h}_t \oplus \vec{b}_t) \wedge F_t(\vec{h}_t \oplus \vec{w}_t \vee \vec{b}_t)$ が1の時+1
- (2) $F_t(\vec{h}_t \setminus \vec{w}_t) \setminus \left(\overline{\overline{p_t(2) \wedge F_t(\vec{h}_t \setminus \vec{b}_t), \wedge F_t(\vec{h}_t \setminus \vec{w}_t)}} \right)$
が1の時 $+1,000 \times |\vec{h}_t \setminus \vec{w}_t \wedge F_t \vec{e}_i|$
- (3) $F_t(\vec{h}_t \setminus \vec{b}_t) \setminus \left(\overline{\overline{p_t(2) \wedge F_t(\vec{h}_t \setminus \vec{w}_t), \wedge F_t(\vec{h}_t \setminus \vec{b}_t)}} \right)$
が1の時 $+10,000 \times |\vec{h}_t \setminus \vec{b}_t \wedge F_t \vec{e}_i|$
- (4) $F_t(F_t(\vec{h}_t \setminus \vec{w}_t) \wedge \vec{h}_t \setminus \vec{b}_t)$ が1の時
 $+100,000 \times |F_t \vec{e}_i \wedge F_t(\vec{h}_t \setminus \vec{w}_t) \wedge \vec{h}_t \setminus \vec{b}_t|$.

3.2 部分局面

囲碁では, 盤上の一部の領域のみを対象として評価を行う部分局面による解析手法が多く用いられている。しかし, ビットボードなどを用いた従来の表現では, 部分局面も長方形である必要があった。しかし本モデルでは盤の形状は F_0 という変数で表され, F_0 を変えることで, 盤の形状というルール自体を変更することができる。また F_0 は対角要素が1である対称行列という制約しか持たず, 任意の無向きグラフ上で使用できるため, 正方形に限らず四角形でも連結でもない碁盤にも適用できる。これにより, 部分局面は任意の形状を取ることができ, 離れた二つの部分局面も扱うこともできる。

3.3 学習システム

本モデルで用いた演算は, ブール代数とその行列演算および数え上げ関数のみである。故に局面からその正着に移すような関数も, 以上の演算のみで表せる。また, どの変数も0と1の組み合わせで表現される。このことから, サポートベクターマシンのような正実数値ベクトルを入力とする2クラス線形分類器を用いて, 局面に対して関数が正着を与えるかといった命題を学習させることができる。これは, 従来の点に重みをつけるような評価ではなく, 点の満たす性質に重みを与える評価関数となる。9)は, この手法を○×ゲームに用いた例である。

4. 結 論

我々は, 囲碁を明確に定義し, ルールを含めた局面

の定式化を行った。結果, 対局は着手により変化する2つのグラフであり, ブール代数と周囲の点の数による1つの関数で表現できることが分かった。また点と連, 両方の観点を用いた2つの有向グラフの変化で対局を表現することで, 局面の変化は連の拡大と初期化で成り立っていることも分かった。さらにグラフ化に伴い, 正方形でも連結でもない自由な形状の部分局面が扱えるようになった。

今回定式化を行ったのはルールと Crazy Stonen の1例のみだが, 今後の課題としては既存の定石を中心としたより多くの概念の定式化を行い, どの演算がどのような形で表れるかを調べることがあげられる。これにより, それらに共通するような重要な概念の発見が期待できる, また離れた複数の部分局面の評価などを行う組み合わせ理論などへの応用も考えられる。

参 考 文 献

- 1) Berlekamp, E. and Wolfe, D.: *Mathematical Go: Chilling Gets the Last Point*, A K Peters Ltd, Natick, MA (1994).
- 2) Tajima, M. and Sanechika, N.: Estimating the Possible Omission Number for Groups in Go by the Number of n-th Dame, Vol.1558, pp.265–281 (1999).
- 3) Graepel, T., Goutrie, M., Kruger, M. and Herbrich, R.: Learning on Graphs in the Game of Go, *Proceedings of the Ninth International Conference on Artificial Neural Networks*, pp. 347–352 (2001).
- 4) Coulom, R.: Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search, *5th International Conference on Computer and Games* (2006).
- 5) S. Gelly, Y. Wang, R. M. and Teytaud, O.: Modification of UCT with patterns in Monte-Carlo Go, *Technical Report, INRIA*, No.6062 (2006).
- 6) Benson, D.B.: Life in the Game of Go, *Information Sciences*, Vol.10, pp.17–29 (1976).
- 7) 中村克彦, 木戸間周平: 数値的な特徴に基づく囲碁局面パタンの解析, *情報処理学会論文誌*, Vol.43, No.10, pp.3021–3029 (2002).
- 8) 佐藤真史, 堤正義: 囲碁における数学的構造の解析, *日本応用数学会 2008 年度年会*, 日本応用数学会, pp.343–344 (2008).
- 9) 佐藤真史, 堤正義, 穴田浩一: サポートベクターマシンを用いた着手の学習, *日本応用数学会 2011 年度年会*, 日本応用数学会, pp.249–250 (2011).
- 10) 佐藤真史: 囲碁における数学的構造の解析, 修士論文, 早稲田大学 (2006).