n-gram 統計を用いた棋譜データベースからの定型手順の獲得

中村 貞吾

九州工業大学 情報工学部 知能情報工学科
E-mail: teigo@ai.kyutech.ac.jp

囲碁は探索空間が広くまた静的局面評価が難しいため，定石，手筋，石の形といった様々なパターン知識の使用が不可欠となる．これまで，このようなパターンは，定石書などから人手で収集したり，ごく狭い固定された窓の範囲内で形パターンを獲得するといった手法が取られていた．本研究では，棋譜を各着手毎の着点が符号化された文字列であるととらえ，文字列の n-gram に基づいて定型的な表現単位を抽出する手法を用いて，棋譜データベースからの定型手順の抽出を行なう．ここで示す手法は，固定窓のような局面範囲の限定を必要とせず，また，長さの異なるさまざまな手順を棋譜から直接抽出できるという特徴を持つ．そして，日本棋院「棋譜データ集 96」の全棋譜（約 34,000 局，総手数約 700 万）を対象として行なった定型手順抽出実験の結果を示す．

Acquisition of Move Sequence Patterns from Game Record Database Using n-gram Statistics

Teigo NAKAMURA

Department of Artificial Intelligence, Kyushu Institute of Technology
E-mail: teigo@ai.kyutech.ac.jp

Move sequence patterns such as Joseki or Tesuji can be used to reduce a lot of search space and speed up the search. To acquire these patterns, we propose a new method based on n-gram statistics, where we encode each move into a character and consider a game record, that is Kifu, as a string. This method can acquire a lot of move sequences of variable length and extent from a game record database. We describe the result of acquisition from "Kifu Database 96" which contains about 34,000 games and seven million moves.

1　はじめに

囲碁は，チェスや将棋などと比べるとはるかに探索空間が広く，また，静的な局面評価が難しいため，定石，手筋，石の形といった様々なパターン知識の使用が不可欠なものとなる．これまで，このようなパターンは，定石書などから人手で収集したり，ごく狭い固定された窓の範囲内で静的な形パターンを獲得するといった手法が取られていた [6][8]．しかし当然のことながら，固定窓を使う手法では，あらかじめ与えた範囲を越えるようなパターンを獲得することはできないし，また，定石などのような長さの異なる手順パターンを効果的に獲得することも難しい．そこで，小島らは生態学的アナロジーに基づいたより柔軟なパターン知識の獲得法を提案している [7]．我々も小島らと同様に，パターンの範囲を固定せず，さらに，静的な局面パターンではなく定石や手筋などのような一定の手順をパターンとして獲得することを目標とする．定型的な手順は，定石知識や着手候補探索の高速化のために使用できることはもちろん，これ以外にも，棋譜検索のインデックス，棋譜クラスタリングのための特徴，棋譜と自然言語表現との対応づけにおける言語化の単位としてなど様々な利用が期待できる．

一般に，定石とは序盤に部分的に出現する一定の石の形およびそこに至る手順を指すが，中盤の定石という類のものもあることからわかるように，序盤に限らず，碁の法則から導かれる一定の理にかなった着手の応酬というものが存在する．そこで，一局の棋譜を通じて定型的であると認められる手順を全て獲得することが望まれる．我々は，棋譜を各着手毎の着点が符号化されてできた文字列であるととらえ，自然言語処理の分野で行なわれている n-gram 統計に基づいた定型表現の抽出法を用いて，このような定型的な手順の獲得を行なう．

2　定型手順の獲得法

近年，自然言語処理，特に日本語処理の分野では，大量のテキストデータから定型的な表現パターンを自動的に抽出する試みが盛んに行なわれている．日本語は，単語の間に空白を置かずにべた書きされるため，ここから単語やそれが連なった定型的な表現を切り出すためには，通常，辞書と文法を用いた形態素解析が行なわれるが，辞書未登録語や新出の表現などにも対応するために，形態素解析を行なわずに文字列の出現頻度情報を用いて定型表現を自動的に抽出する様々な研究が行なわれている [1][2] [3][4]．そこで，これらの手法のうちのいくつかを棋譜データからの定型手順獲得に適用することを試みる．

2.1　n-gram 統計

n-gram 統計とは n 個の文字が隣接した文字列がテキスト中にどのような頻度で出現するかを調査したものを指す．長尾らは大量のテキストから任意の n に対する n-gram 統計を簡単に作成する手法を示し，日本語テキストに対して種々の n に対する n-gram を比較することによって意味のある表現単位の抽出を行なった [1]．この手法は，形態素解析を必要としないという利点があるが，基本的に出現頻度の高い文字列を網羅的に収集するため，断片的な（まとまった表現とは認めがたい）文字列も多数抽出されるという問題があった．複数の n に対する n-gram を用いた場合，ある文字列 x が対象テキスト中に出現する頻度 $f(x)$ と x の部分文字列 y の出現頻度 $f(y)$ の間には $f(x) \leq f(y)$ が成立する

ため，単純な出現頻度の比較だけでは文字列の定型性を判断することは難しい．この問題を解決するために，出現頻度を正規化する方法 [2] や注目している文字列に隣接する文字のエントロピーを用いる方法 [3] などいくつかの手法が提案されている．

2.2 正規化頻度法

文字列 x, y に対して $f(x) = f(y)$ であったとしても，$|x| < |y|$ ($|x|$ は文字列 x の長さ) であれば y の方が重要であると考えることができる．これは，文字列の長さが長くなるにつれて文字列の可能な種類が増加し，長い文字列の出現頻度分布が総体的に小さくなるためである．中渡瀬は，これを補正するために，出現頻度に文字列の長さ n に応じた係数 $\alpha(n)$ を乗じた正規化を行なう方法を提案している [2]．そこでは，対象テキスト中に実際に出現した n-gram の異なり数を $\beta(n)$ として，正規化係数 α と正規化頻度 Nf は次のように計算される．

$$\alpha(n) = \sum_{i=1}^{n} \beta(n)$$
$$Nf(x) = (f(x) - 1) \cdot \alpha(|x|)$$

そして，この正規化頻度の大きい順に表現を獲得する．また，正規化頻度で上位の文字列の部分列で下位の順位にあるものを排除することで，断片的な文字列の抽出を防いでいる．

2.3 隣接文字エントロピー法

下畑らは，テキストから抽出された文字列が意味のある表現のまとまりであるかどうかをその文字列の前後に出現する文字の分散の度合を基準に判断して, 断片的な文字列を排除する方法を提案している [3]．

文字列 x と文字 c に対して，x の直後に c が生起する確率 $P(c|x)$ は次のように求めることができる．

$$P(c|x) = \frac{f(xc)}{f(x)}$$

x に後接する文字集合を $C(x)$ とすると，x の後接文字のエントロピー $H_R(x)$ は次式で計算される．

$$H_R(x) = -\sum_{c \in C(x)} P(c|x) \cdot \log P(c|x)$$

$H_R(x)$ は，後接文字の種類が多く出現の度合が均等であるほど大きくなり，すべての後接文字が等確率で出現するときに最大となる．逆に，後接文字の種類が少なく出現の度合が偏っているほど $H_R(x)$ は小さくなり，$|C(x)| = 1$ のときに 0 となる．x の前接文字に対するエントロピー $H_L(x)$ も同様にして計算し，$H_R(x)$ と $H_L(x)$ の小さい方の値をエントロピーの有効値 $H(x)$ とする．そして，$H(x)$ の高い順に定型表現として抽出する．

2.4 部分列頻度プロファイル

正規化頻度法および隣接文字エントロピー法は，n-gram を直接用いる手法に比べて断片的な文字列の抽出を避けるような改良がなされているが，基本的にセグメンテーションを行なわない手法であるため，それでもなお互いに重なりを持つ文字列や断片的な文字列を抽出してしまうことがある．そこで我々は，n-gram 統計を用いて文字列から定型表現を直接切り出すために，新たに，部分列頻度プロファイル (Substring Frequency Profile; SFP) を提案する．以下に，棋譜データを例にしてそのアイデアを説明する．

SFP とは，注目する部分列の長さ n を固定し，ある棋譜データについて $i - n + 1$ 手目から i 手目までの n 長さの着手列がデー

タベース中に出現した度数を各 i に対して記録したものであり，これは，各着手の近傍における定型性の指標とみなすことができる．

図 1 の棋譜 (日本棋院「棋譜データ集 96」より) に対して，$n = 8$ および $n = 6$ で作成した SFP を 図 2，図 3 に示す．

図 1 では，序盤，右下と左上に基本定石が出現している．2.1 節で述べたように，ある文字列の部分列は元の文字列よりも出現頻度が高いため，注目している部分列が頻出する基本定石手順の内部にあるときはその出現頻度は高い値をとるが，定石が一段落した境界をまたぐ部分やその外部では出現頻度は低い値となる．したがって，SFP において “山” の部分を切り出すことでひとまとまりの手順パターンを獲得することができる．図 2 の長さ 8 の部分列による SFP では，基本定石部分が明瞭に浮き出てきている．図 3 では，この 2 つの基本定石以外に 6 つの “山” が認められる．これに対応する部分を抽出したものを図 4 に示す．

SFP を用いて実際に定型手順を獲得するには，データベース中の各棋譜に対して SFP を作成し，切り出されたパターン候補の出現頻度の高い順に定型手順として獲得すればよい．

通常の n-gram を用いた方法では，データベース中のすべてのデータをおしなべて集計した結果のみを用いて定型性の判定を行なっているため，それが出現する周辺の状況は無視されているが，SFP を用いることによって 1 局を通じての出現頻度の変化に基づいた，より柔軟な手順の獲得を行なうことが可能となる．

また，定石や定型手順が一局の中でどの時期にどのように使用されているかは，その対局を特徴づける指標であるとも考えられるので，SFP は棋譜のクラスタリングをする上で有効な手段となるのではないかと考えている．

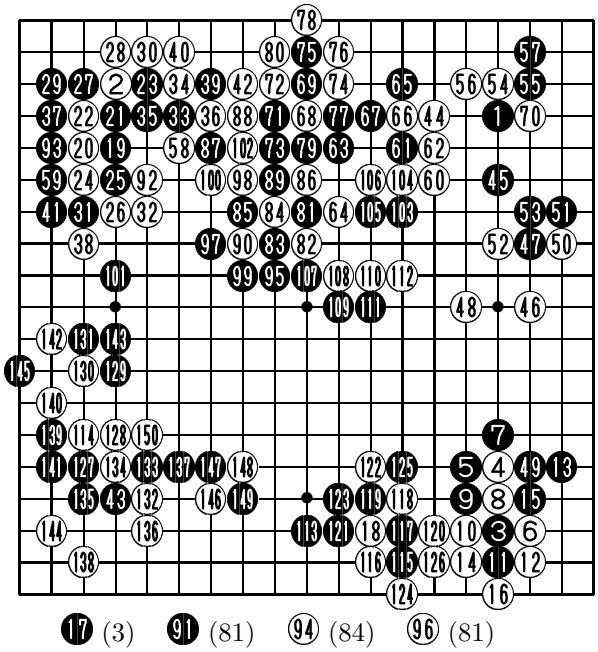


Fig. 1: Sample Game

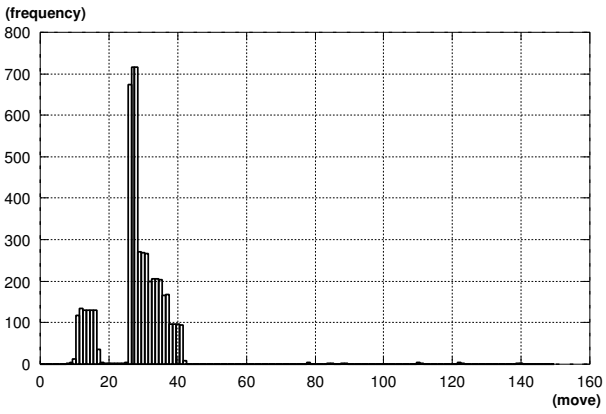


Fig. 2: SFP (substring length = 8)

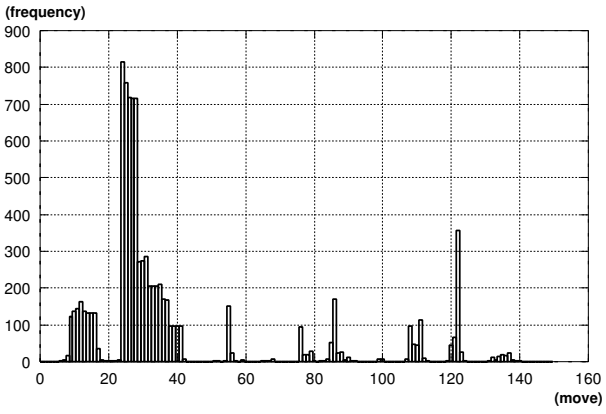


Fig. 3: SFP (substring length = 6)

3　棋譜からの定型手順の獲得

前章で述べた 3 つの手法を実際に棋譜データベースから定型手順獲得に適用し，各手法の比較を行なう．

3.1　使用データ

日本棋院「棋譜データ集 96」は，1960 年 ～ 1995 年の間に打たれたプロ棋士の対局約 34,000 局分の棋譜を収録しており，このようなパターンを獲得するための情報源として格好のデータベースである．実験には，データ不備などの問題があった 57 局分[1] を除いた残りの 33,708 局，総手数 6,951,898 手分のデータを用いた．

3.2　着手の符号化

n-gram ではパターンが一致するかどうかを文字列の一致性で判断しなければならない．盤面上で，回転，鏡像，平行移動などの関係にあるパターンを同一視するために，棋譜中の各着手に対して，直前の（相手の）着手との相対的な位置の差分をもとにした符号化を行なう．例えば，現在の着点が $(5, 3)$ で直前の相手の着点が $(3, 4)$ である着手に対して $c_{1,2}$ という符号を与える（$c_{i,j}$ において $0 \leq i \leq j$ となるように，必要であれば順序を交換する）．また，初手は絶隅からの差分とし，他の着手とは異なる符号系列を与える．

3.3　実験と結果の評価

まず，棋譜データベース中の各棋譜を符号化してできた文字列に対して，文献 [1] の手法を用いて n-gram を作成した．

次に，2 章で述べた各手法を用いて，手法毎に規定される順位をつけて定型手順の獲得を行なった．それぞれの手法で獲得された定型手順の上位のものを付録に示す．これらの手法で直接獲得されるものは注目している手順のみであり，そこには周辺の配石は含まれていない．そこで，その手順が適用される際の周辺の配石は棋譜から別途収集する．今回は，手順中の石を丁度含むところから外側に各々 2 だけ拡大した矩形の範囲にある配石を収集した[2]．また，付録の図中の ( ) 内の数値は，左から順に，順位,（手順のみを構成する）着手記号列の出現頻度，周辺の配石を含めた盤面パターンの頻度[3] である．図には，手順を構成する石の配置が同一で周辺の配石だけが異なるものについては「盤面パターンの頻度」が最大のもののみを掲載している．

各手法で獲得した個々のパターンを眺めてみると，まず，正規化頻度法では「大ナダレ内マガリ定石」の部分的な手順が多数抽出されていることが目につく．正規化頻度法は，長手順でかつ頻出するものに高い優先順位を与えるが，それがまとまった単位として機能しているかどうかの判定をしていないため，互いに重なり合う文字列の組合せが多数抽出される傾向にある．それに対して，隣接文字エントロピー法と部分列頻度プロファイル法では，まとまった単位かどうかの判定が行なわれているため，正規化頻度法の抽出結果のような違和感はない．

次に，これらの手法によりいわゆる定石手順がどの程度獲得されたかを数量的に比較するために,「基本定石事典 [5]」に代表的な基本

[1] 他の棋譜と全く同一なもの（入力ミスと思われる）が 3 局あった．残りは，途中に着手位置不明な手が含まれるものや 2 手打ちの反則があるもの．

[2] 手順の適用の可否を厳密に判断するための配石を獲得するには，清ら [6] のようにマンハッタン距離を用いるべきかもしれないが，ここでは抽出した手順の表示が主たる目的なので簡単のためこのようにした．

[3] 盤面パターンの頻度においては，平行移動の関係にあるものは別物として数えた．

定石として掲載されている定石[4] が獲得できたかどうか，獲得できた場合は何位で獲得されたかを調査した．その結果を表 1 に示す．表中の値は再現率 (recall ratio) でこれは次式で計算される．

$$\text{再現率} = \frac{\text{抽出された基本定石数}}{\text{定石事典中の基本定石数}}$$

ここでは，隣接文字エントロピー法よりも正規化頻度法の方が良い結果となっている．その理由は，定石事典に掲載されている基本定石は長手順のものが多く，正規化頻度法はこのようなものに高い優先順位をつけるということと，隣接文字エントロピー法は比較的短いまとまった単位に高い順位をつける傾向にあるためであると考えられる．

また，元々の隣接文字エントロピー法は H_R と H_L のうち小さい方の値 H を有効値として用いているが，定石手順の場合はその開始位置に多少のばらつきがあるため H_R を有効値として採用する方法も試みた．その結果，棋譜からの手順獲得には H よりも H_R を用いた方が有効であることがわかった．

次に，部分列頻度プロファイル法では $n = 6$ が少ない抽出数において最も良い結果となり，単独の手法としてはこの手法が定型手順の獲得に最も有効であると考えられる．

正規化頻度法および隣接文字エントロピー法は，それぞれ単独の使用では一長一短があったが，正規化頻度法で出力される候補に対して隣接文字エントロピーがある閾値を越えるもののみを選択するようなフィルタとして使用することにより，再現率を向上させることができた．

[4]「基本定石事典」には代表的な基本定石に「◆◆」印が付されている．ここでは，「◆◆」が付された定石と，これ以外にも基本定石と思われるものを若干数加えた計 414 個を使用した．

4 おわりに

棋譜データベースから定石などの定型的な手順を獲得するにあたって，棋譜を着手が符号化された文字列であろととらえ n-gram 統計を用いて定型的なパターンの獲得を行なう手法を示した．中でも部分列頻度プロファイル法 (SFP) は，少ない抽出数で基本定石を獲得することができ，定型手順の獲得に有効であることがわかった．今後は，抽出された手順パターンが適用される条件（周囲の配石）の調査やその結果得られたパターンの対局システムへの利用を行ないたい．

参考文献

[1] 長尾真, 森信介：“大規模日本語テキストの n グラム統計の作り方と語句の自動抽出”, 情報処理学会自然言語処理研究会報告 NL 96-1, pp.1–8, 1993.

[2] 中渡瀬秀一：“統計的手法によるテキストからのキーワード抽出法”, 電子情報通信学会データ工学研究会報告 DE 95-2, pp.9–16, 1995.

[3] 下畑さより, 杉尾俊之, 永田淳次：“隣接文字の分散値を用いた定型表現の自動抽出”, 情報処理学会自然言語処理研究会報告 NL 110-11, pp.71–78, 1995.

[4] M. Takeda and F. Matsuo： “Markov String Grammar”, Memories of the Faculty of Engineering, Kyushu University, 55(3), pp.279–284, 1995.

[5] 石田芳夫：“基本定石事典”, 上, 下巻, 日本棋院, 1975.

[6] 清愼一, 川嶋俊明：“「局所パターン」知識主導型の囲碁プログラムの試み”, ゲームプログラミングワークショップ’94, pp.97–104, 1994.

[7] 小島琢矢, 植田一博, 永野三郎：“生態学アナロジーを用いた囲碁パターン知識の獲得”, ゲームプログラミングワークショップ’96, pp.133–140, 1996.

[8] 斎藤康己：“囲碁: これからは囲碁プログラミングが面白い”, bit 別冊「ゲームプログラミング」, pp.59–72, 1997.

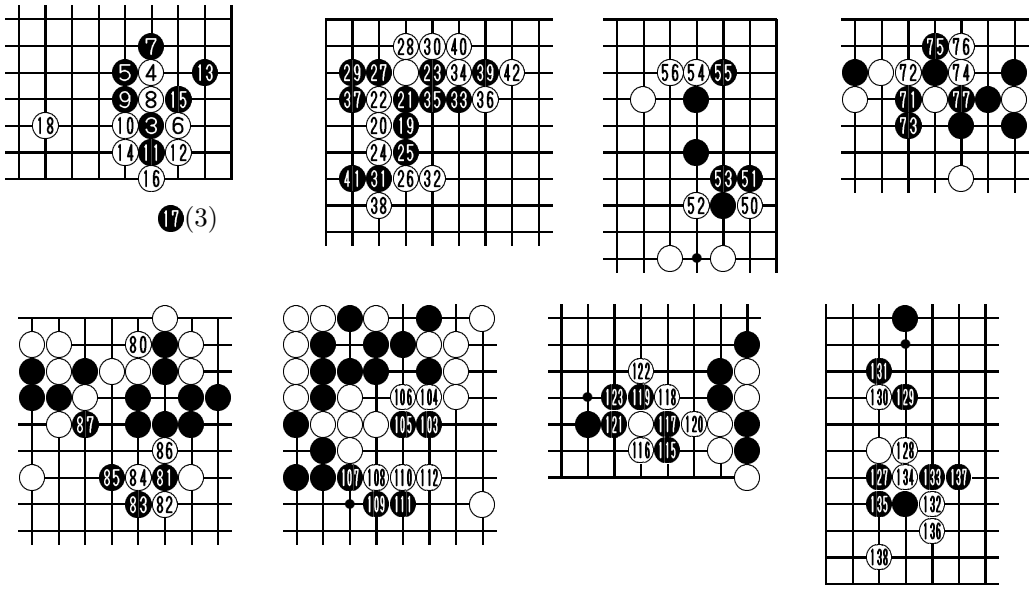


Fig. 4: Sequence patterns extracted from SFP(n=6)

順位	NF	E		SFP			NF+E			
		H	H_R	$n=4$	$n=6$	$n=8$	$H \geq 1$	$H \geq 2$	$H_R \geq 1$	$H_R \geq 2$
1000	19.1	1.2	5.8	23.1	29.6	22.8	27.5	24.6	25.6	24.6
2000	25.8	3.6	9.7	26.7	35.9	26.5	34.1	30.2	32.1	31.9
3000	30.2	6.5	11.4	28.9	38.6	28.6	38.6	33.6	36.0	35.0
4000	33.6	11.6	14.0	31.3	39.6	31.3	41.5	35.0	39.9	38.2
5000	35.7	13.3	15.5	33.0	42.2	32.0	43.0	36.2	41.3	39.1
6000	37.9	15.0	17.6	36.2	42.5	32.3	43.5	37.2	43.5	41.5
7000	39.4	16.4	21.3	37.9	43.2	—	45.7	38.4	44.0	42.8
8000	41.1	17.6	22.7	39.1	44.2	—	47.3	39.4	44.2	43.5
9000	42.0	19.1	24.6	40.8	45.1	—	48.8	39.9	46.6	44.4
10000	43.2	19.6	25.8	41.5	45.4	—	48.8	39.9	47.1	45.2
15000	45.9	23.2	31.4	45.9	48.1	—	51.7	42.3	51.0	47.3
20000	49.5	26.3	35.0	46.8	49.5	—	52.9	42.8	53.4	48.8
25000	52.4	29.5	38.4	48.5	—	—	55.3	43.7	54.8	49.5
30000	54.6	32.1	40.6	48.8	—	—	56.3	44.4	56.8	49.8
35000	56.3	33.3	42.3	49.0	—	—	57.0	44.9	57.5	50.5
40000	57.0	35.7	42.8	50.0	—	—	58.2	44.9	58.0	51.4
45000	58.0	37.9	44.2	51.7	—	—	59.4	45.4	58.7	51.7
50000	58.2	39.4	45.7	51.7	—	—	61.4	45.4	59.4	52.2

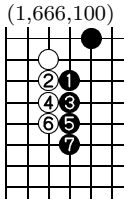
NF : 正規化頻度法 , SFP : 部分列頻度プロファイル法
E : 隣接文字エントロピー法 , NF+E : NF と E の組み合せ

Table. 1: Recall ratio of each method

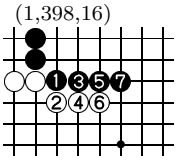
付録 A　正規化頻度法

正規化頻度の大きい順．手順長 6 以上のもの．() 内の数値は，左から順に，順位，着手符号列の頻度，周辺の石を含めた盤面パターンの頻度．

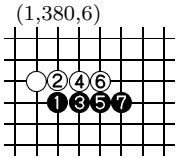
(1,666,100)



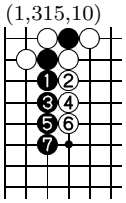
(1,398,16)



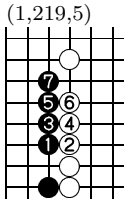
(1,380,6)



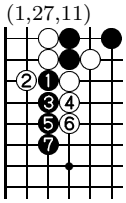
(1,315,10)



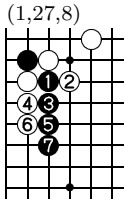
(1,219,5)



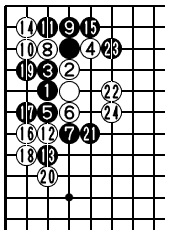
(1,27,11)



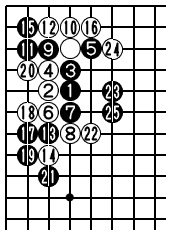
(1,27,8)



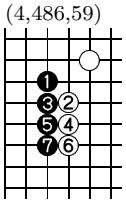
(2,339,295)



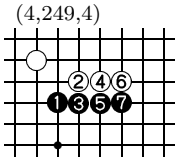
(3,316,274)



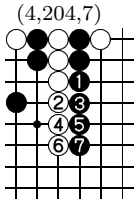
(4,486,59)



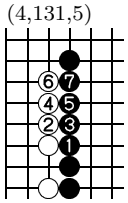
(4,249,4)



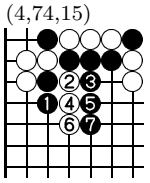
(4,204,7)



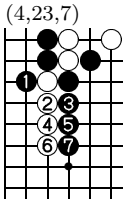
(4,131,5)



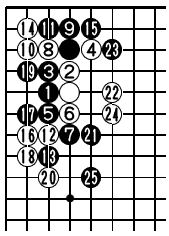
(4,74,15)



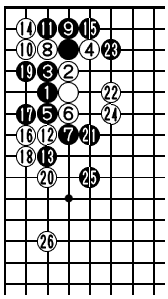
(4,23,7)



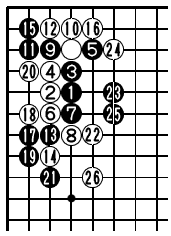
(5,256,228)



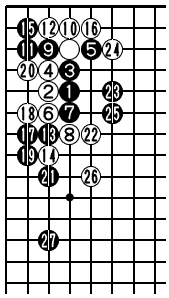
(6,243,219)



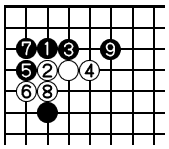
(7,234,208)



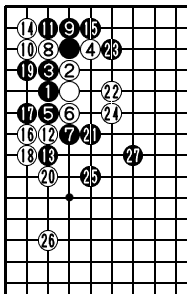
(8,223,201)



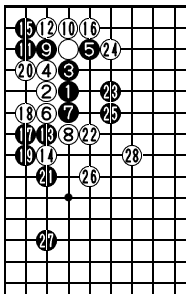
(9,1256,856)



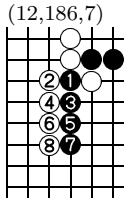
(10,195,174)



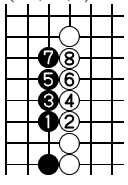
(11,176,157)



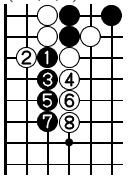
(12,186,7)



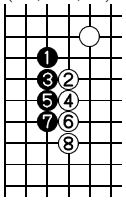
(12,79,5)



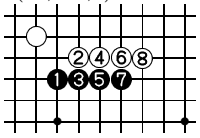
(12,9,4)



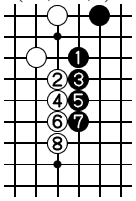
(13,254,48)



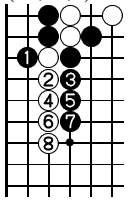
(13,123,4)



(13,100,4)



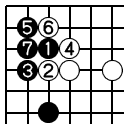
(13,14,6)



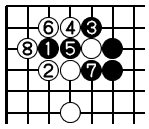
付録 B　隣接文字エントロピー法

$H = \min\{H_L, H_R\}$ の大きい順．手順長 6 以上のもの．

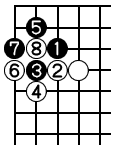
(1,806,362)



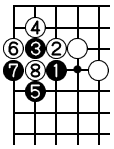
(2,75,19)



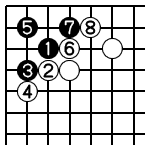
(3,79,24)



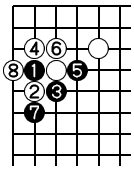
(3,32,4)



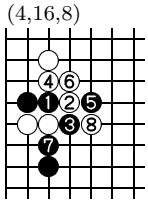
(3,15,4)



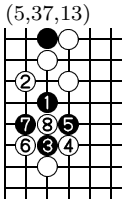
(4,188,28)



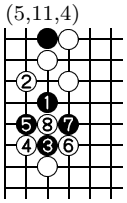
(4,16,8)



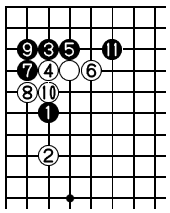
(5,37,13)



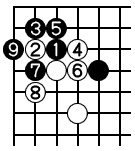
(5,11,4)



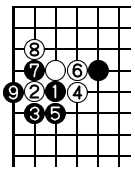
(6,429,295)



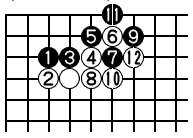
(7,55,4)



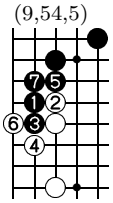
(7,30,4)



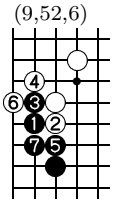
(8,189,108)



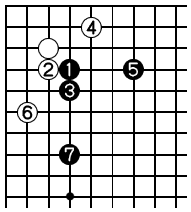
(9,54,5)



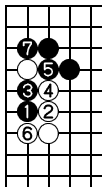
(9,52,6)



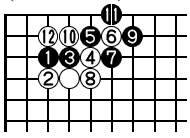
(10,109,59)



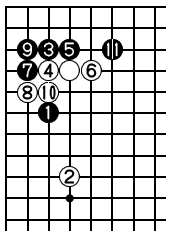
(11,42,22)



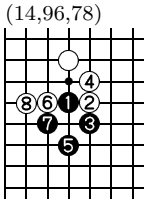
(12,229,125)



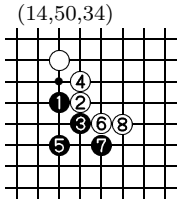
(13,223,158)



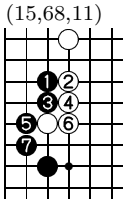
(14,96,78)



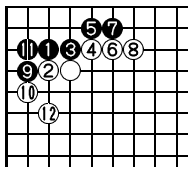
(14,50,34)



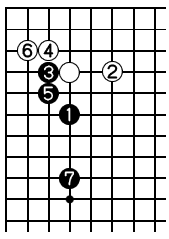
(15,68,11)



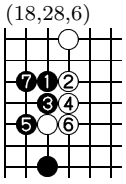
(16,282,159)



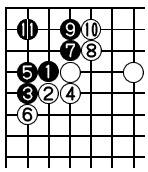
(17,49,33)



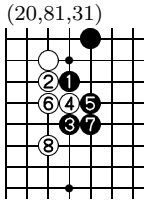
(18,28,6)



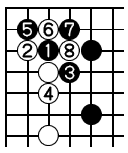
(19,39,8)



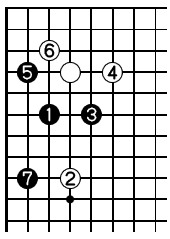
(20,81,31)



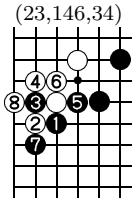
(21,28,5)



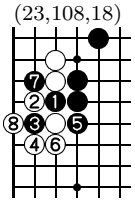
(22,207,142)



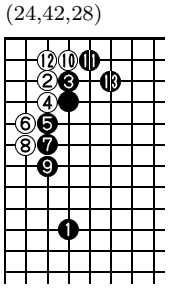
(23,146,34)



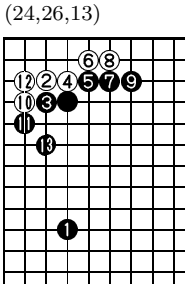
(23,108,18)



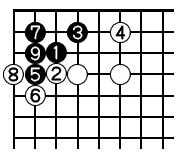
(24,42,28)



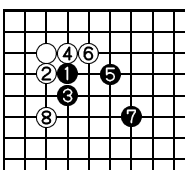
(24,26,13)



(25,38,22)



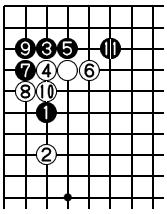
(26,28,18)



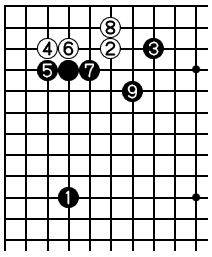
付録 C　部分列頻度プロファイル

$n = 8$ の SFP を用いて切り出された候補内での頻度順．手順長 6 以上のもの．

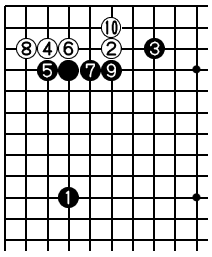
(1,429,295)



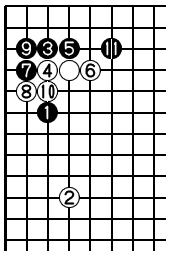
(2,309,228)



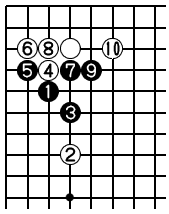
(3,218,187)



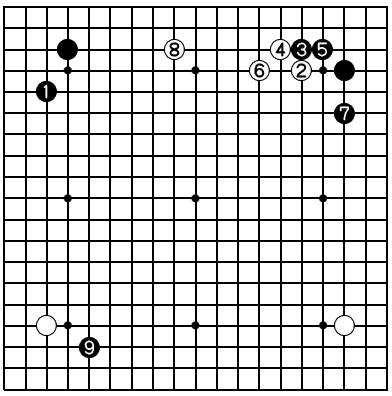
(4,230,144)



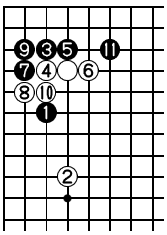
(5,208,166)



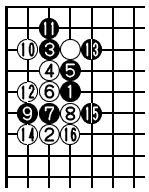
(6,496,197)



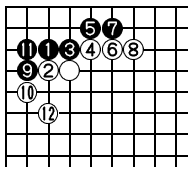
(7,223,158)



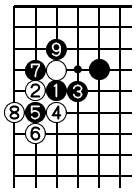
(8,249,178)



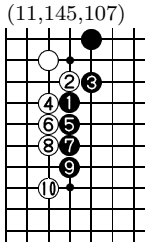
(9,282,159)



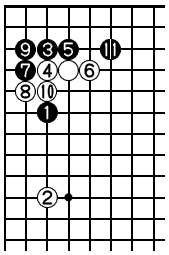
(10,208,151)



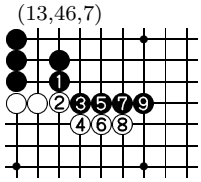
(11,145,107)



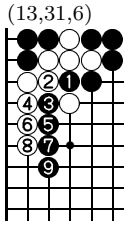
(12,90,62)



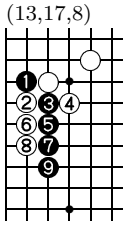
(13,46,7)



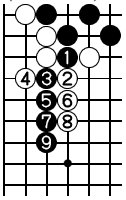
(13,31,6)



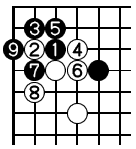
(13,17,8)



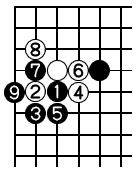
(13,16,15)



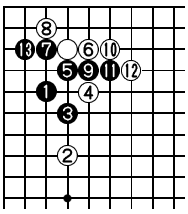
(14,55,4)



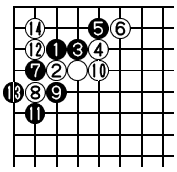
(14,30,4)



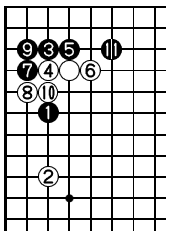
(15,102,79)



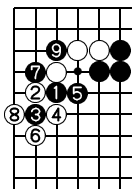
(16,124,73)



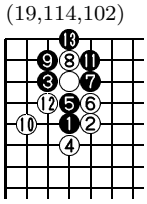
(17,57,37)



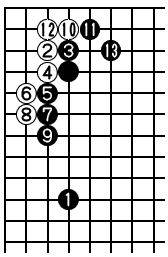
(18,52,4)



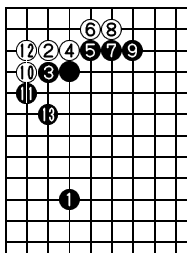
(19,114,102)



(20,42,28)



(20,26,13)



(21,79,67)