

囲碁の着手符号列に対する確率文法モデルの作成

梶山 貴司 中村 貞吾

九州工業大学大学院 情報工学研究科

{kajiyama, teigo}@dumbo.ai.kyutech.ac.jp

概要

囲碁は状態空間が広く、人間の思考形態を模倣するアプローチは不可欠である。筆者らは棋譜に記載している1局の手順を自然言語と同等のテキストととらえ、棋譜テキストに対してそれを記述することができるような確率文法モデルを作成することにする。確率文法モデルは盤面認識や着手候補選択に用いることができるほか、ゲーム記述言語と自然言語との相互互換を通じて自然言語システムへ応用することもできる。本論文では、まず棋譜テキストに対する確率文法モデルを定義する。つぎに棋譜データベース中の棋譜を用いて確率文法モデルを作成する手続きを示す。そしてその確率文法モデルに対し、過去の棋譜を例に次着手の候補を列挙させる実験を行い、着手候補選択に適用することの有用性を検証する。

Constructing A Stochastic Model for Encoded Strings of Go Moves

Takashi KAJIYAMA

Teigo NAKAMURA

Abstract

Game records can be regarded as text strings by suitable encoding from each move to a character. We had acquired move sequence patterns automatically from these encoded strings of Go moves, and confirmed that this text base approach worked effectively in former research.

In this paper, we propose a stochastic grammar model for texts of game records that can assign the order of priority to each next move candidate. This model can be used to reduce a lot of search space and time of Go programs, and be applied to natural language systems about the game of Go.

We describe how to construct this model from learning database of game records, and report the result of experiments to show the effectiveness of this model.

1 はじめに

近年、チェスをはじめとする有限ゲームのいくらかは計算機プレイヤーが人間の強豪を相手に勝利するようになっている。しかし同じ有限ゲームに分類する囲碁や将棋などの計算機プレイヤーはいまだにプロに勝利するには至っていない。後者らのゲームが共通している特徴である状態空間の圧倒的な広さは、状態に対して静的な評価関数の設定や読みの作業を模した探索を困難なものにしている。近年、囲碁などのゲームにおいては、なかなか強くないこれら対局プログラムの補助や、あるいは着手決定のルーチンそのものに、メモリベースのパターンマッチや、自己学習などの概念を導入するようになった。今現在では囲碁の対局プログラムにおいてパターン知識の利用は欠かせないものとなっている。

これらの新しい試みの中に、メモリベースのアプローチがある。このアイデアは「プロが打った良い手をたくさん覚えておいて、着手とのパターンマッチングによって検索し、一番近い状況でプロ打った手と同じ手を打てばきっと強いのではなからうか？」という単純明快なものである。この発想を実践したものに清らの「勝也」があり、興味深い報告がなされている [1][2]。

われわれは、メモリベースのアプローチはその発想の単純さからくる予想以上の効果があるとみている。そして、このアプローチにもとづき、パターン知識として定石や手筋といった定型手順に着目し、定型手順の自動獲得と利用を目標に研究をしてきている。

以前よりわれわれは、定石や手筋といった定型手順と自然言語テキストの定型表現の持つ特徴の類似性に注目している。これまで、棋譜に記述してある一連の着手を、時系列に沿って記述したテキストととらえ、その中で定型的な手順の出現を獲得することを試みた。

結果として、連続着手の頻度情報から多くの定型的な手順を獲得することができ、囲碁プログラミングにおいて棋譜テキストをベースにしたアプローチが有効である根拠を得た。

このように、初めに「棋譜テキスト在りき」でこれをもとに定型手順を獲得した。そこで今回の研究では、次なる段階として、以前とは逆に棋譜テキストを生成する文法を構築することを目標とする。この文法モデルは盤面認識や着手候補選出など対局システムの中核として利用できるだけでなく、ゲーム記述言語と自然言語との相互互換を通じて、棋譜からの解説文作成や自然言語によるデータベース検索など、知的ゲームに関する自然言語システムへと幅広く応用することができる。

次節以降では、文法モデル構築の具体的な方法を説明し、構築された文法モデルの適用実験の結果と評価について述べる。

2 確率文法モデル

われわれは1局の棋譜に記載されている着手の応酬、つまり時系列に沿った着点の連結を棋譜テキストと呼んでいる。連続した着手列を「テキスト」と呼ぶのは、棋譜テキストに含まれる定型手順の特徴と自然言語テキストにおける定型表現の特徴が類似しているからである。また、棋譜テキストの中に含まれている定型手順の獲得を言語処理の定型表現獲得の手法を用いて実現できている。

今回は、1つの対局の一部または全部を表現する棋譜テキストに対して、その生成確率を与えてくれる確率文法の構築をおこなう。

本節では、確率文法の作成目的と概念を示し、ゲームプログラミングに適用する具体的方法を述べる。

2.1 確率文法モデルで何をするのか?

棋譜テキストを記述する確率文法モデルの作成の目的の一つは

- 確率文法モデルに着手予想をさせる

ことである。今回はこの点に注目して文法モデルの作成をおこなう。

ある静的な局面を見て次着手の決定を行うとき、その局面に至る過程を知っている場合と知らない場合とで対局者が行う探索の範囲が大いに異なる。たとえば、局所的に激しい戦いが繰り広げられており、直前の着手に対する応手が必要な局面が手順情報なしに与えられたとき、応手は盤面全体を認識した上で探さなければならないが、直前の数手の着点分かっているならば、探索の範囲は最初から絞り込めるはずである。一般に、人間の対局者は(あるいは多くの囲碁プログラムも同様に)、対局開始から n 手すすめたとき、 $n+1$ 手目の決定は n 個の石が成す静的な局面から判断するだけでなく、 n 手目までに行われたの着手の系列からも判定する。そこで、対局開始から n 手の着手を表現する棋譜テキストに対して、次の1手の決定を静的な局面からではなく、その局面に至る着手の系列から判定することを、今回作成する確率文法モデルが行う次着手予想の戦略とする。

2.2 確率文法

今回適用する確率文法は、形式言語の構造を記述するものである。オートマトン論や言語理論では、文法に関する多くの定理も存在し [5]、棋譜テキストを記述する文法を構築することは大変有用である。

確率文法 G は5つ組

$$G = (\Sigma, N, S, P, p)$$

によって定義する。各成分は次のものとする。

Σ : 空でない有限集合で、終端アルファベットと呼ぶ。 Σ の要素を終端記号 (terminal symbol) という。

N : 空でない有限集合で、非終端標識アルファベットと呼ぶ。 N の要素を非終端記号 (nonterminal symbol) という。

S : $S \in N$ で、開始記号 (start symbol) という。

P : P は、 $(N \cup \Sigma)^+ \times (N \cup \Sigma)^*$ の有限部分集合。 P の要素 (α, β) は生成規則 (production) とよばれ、 $\alpha \rightarrow \beta$ と書く。

p : p は、関数 $p: P \rightarrow (0, 1]$ である。 p は生成規則の G への適用確率と呼び、記号列 α が生成規則 $\alpha \rightarrow \beta$ に適用される確率をあらわす。言い換えれば p は記号列 α の発生した状況下で記号列 β が発生する条件付き確率と定義できる。以降、生成規則 $\alpha \rightarrow \beta$ の G への適用確率 p の値を $p(\alpha \rightarrow \beta)$ という形式で表現する。

語 $w \in \Sigma^+$ と開始記号 S について、有限回の生成規則の適用で S より w へ導出するパス $S \xrightarrow{\pi} w$ (π は適用する生成規則の系列) が存在するとき、 w は G の生成する語という。導出パス $S \xrightarrow{\pi} w$ がただ一つ存在するとき、生成確率 $p(w)$ は $p(S \xrightarrow{\pi} w)$ に等しく、その値は、

$$p(S \xrightarrow{\pi} w) = \prod_{(\alpha \rightarrow \beta) \in \pi} p(\alpha \rightarrow \beta)$$

で与えられる。文法 G の生成する言語 $L(G)$ を

$$L(G) = \{w \in \Sigma^* | p(w) > 0\}$$

と書く。言語 $L \subseteq \Sigma^*$ に対して $L = L(G)$ となるとき、 G は L を生成するという。

2.3 棋譜テキストを記述する確率文法モデル

文法は、そのレベルによって生成できる言語の範囲に大きな差異があることが知られている。将来、文法のレベルと、文法が生成する言語との関わりを調べることにするが、今回は、最もレベルの低い正規文法¹を作成することにする。棋譜テキストを生成する正規文法は、棋譜データベースにおける着手の出現頻度に基づいた生成規則と適用確率の定義をすることで作成できる。

2.3.1 終端アルファベット Σ

棋譜テキストの記述単位は、1手の着点に対応した符号である。まず容易に思いつく終端アルファベットは、着点の座標に対して一意に定まる終端記号の集合である。ただ、確率文法モデルを構築するためのソースとなる棋譜データベースが十分な量を得られない場合、後述する sparseness 問題の影響を大きく受ける。そこで、対策として何らかの共通の特徴を持った着点をひとつの記号に集約し、記号の総数を削減する方法がある。

今回の研究では、3種類の符号化法を用い、それぞれの符号化法にもとづく確率文法モデルを構築して、その影響を調べることにする。以下に符号化法を示す。

その1：符号化法 E_p ；一般に、中央、辺、隅で、1つの石の持つ役割の大きさが異なる。そこで中央からの相対位置が同じところを同じ符号とする。まず、着点 (x, y) と天元 $(10, 10)$ との相対位置 $(i, j) = (|x - 10|, |y - 10|)$ をもとめる。これを $0 \leq i \leq j$ となるように i, j を入れ替え。 $c_{i,j}$ を符号とする。この符号化法の符号の総数は55個になる。

その2：符号化法 E_r ；同型の手順は回転、鏡

¹正規文法ともいう。

像、平行移動に対し同じ符号列であるべきと考える。そこで、直前の着点からの相対位置が同じところを同じ符号とする。まず、着点 (x, y) と直前の着点 (x_p, y_p) との相対位置 $(i, j) = (|x - x_p|, |y - y_p|)$ を求める。これを $0 \leq i \leq j$ となるように i, j を入れ替え。 $c_{i,j}$ を符号とする。なお、初手は天元 $(10, 10)$ との相対位置をとることにする。符号の総数は190個になる。

その3：符号化法 E_s ；周辺の配石状況は着手を決定する重要な要素である。そこで、着点の周辺配石が同一のところを同じ符号にする。ただし、符号の総数をむやみに増やさないために、着点 (x, y) の4つの隣接点の配石状況に対し、一意に定まるように符号を求める。辺、隅の場合も同様にするが、辺、隅とそれ以外の地点は必ず異なる符号となるように割り当てる、符号の総数は45個になる。

また、終局を示す記号として“#”を終端アルファベットに追加する。一局全体を記述する棋譜テキストは末尾に必ず記号“#”が出現することになる。

2.3.2 非終端アルファベット N

今回の確率文法モデルの場合、非終端アルファベットの要素は、棋譜テキストデータベースから n -gram 統計をとり、出現頻度2以上の任意の符号列 $w \in \Sigma^*$ に対して記号 $[w]$ とした。

たとえば、 $w \in \Sigma^*, c \in \Sigma$ として、非終端記号 $[wc], [w]$ がともに存在するとする。これはすなわち、棋譜データベースに頻度2以上の符号列 wc, w がともに存在するときである。このとき、後述する適用確率の値 $p([w] \rightarrow [wc])$ は、符号列 x のデータベース上の出現頻度

$f(x)$ を用いて,

$$p([w] \rightarrow [wc]) = \frac{f(wc)}{f(w)}$$

で与えることができる.

また, 開始記号は $[\varepsilon]$ とする. ここで, ε は空文字を表し, $f(\varepsilon)$ を棋譜データベースの総手数と定義しておく.

2.3.3 生成規則の集合 P

次着手は直前までの着手系列から決定する. 今回はこの概念を単純に確率文法モデルにあてはめることにする. $[w], [wc], [w_p w] \in N, w_p, w \in \Sigma^*, c \in \Sigma$ とする. 生成規則の形は以下のいずれかとする.

$$[w_p w] \rightarrow w_p [wc] \quad (1)$$

$$[w\#] \rightarrow w\# \quad (2)$$

ただし, 生成規則 (1) は, 導出の一義性を保つため, $|w|$ が最大となるような $w_p w$ の分割に限るとする.

この生成規則の形は正則文法となっている.

2.3.4 適用確率 p

非終端記号 $[w]$ の定義から $p([w_p w] \rightarrow w_p [wc])$ は棋譜テキストデータベース中のテキスト $w_p wc$ の出現頻度 $f(w_p wc)$ と $w_p w$ の出現頻度 $f(w_p w)$ の比であるとすればよい. まず考え付くのは,

$$\begin{aligned} p([w_p w] \rightarrow w_p [wc]) &= \frac{f(w_p wc)}{f(w_p w)} \\ p([w\#] \rightarrow w\#) &= 1 \end{aligned}$$

とすることである. しかし, 棋譜データベースは有限なので, 一般に符号列 x の長さ $|x|$ が大きくなると $f(x) \rightarrow 0$ となる. このことにより, 符号列が長くなると, $p([w_p w] \rightarrow w_p [wc])$ を求められなくなる sparseness 問題が生じる. この解決は補間や平滑化などの方法 [4] があるが, 今回は簡単のために出現頻度を補正

することによって対処する. 以下にテキスト $c_n c_{n-1} \cdots c_1 \in \Sigma^n$ 補正頻度 $f_s(c_n c_{n-1} \cdots c_2 c_1)$ を定義する.

$$\begin{aligned} f_s(c_n c_{n-1} \cdots c_1) &= \sum_{i=1}^n \lambda_i f(c_i \cdots c_1) \\ \lambda_i &= \frac{|\Sigma|^{i-n}}{n} \end{aligned}$$

これは, 学習データベース中の長い符号列に一致する符号列ほど, 「頻度が高い」と補正しようとするものである. また, 係数 λ_i は長さが k だけ異なる 2 つ符号列 $xw, w(|x|=k)$ の頻度は $|\Sigma|^k$ 程度異なると仮定し, 長さ n の符号列と長さ $n-k$ の符号列の頻度の比が $1:|\Sigma|^k$ であれば双方の生成確率が同じとなるように設定した. これを用いて,

$$p([w_p w] \rightarrow w_p [wc]) = \frac{f_s(w_p wc)}{f_s(w_p w)}$$

で適用確率の推定値を求める.

この場合でも, 全ての事象の生起確率の総和は 1 であること, すなわち,

$$\sum_{c \in \Sigma} p([w_p w] \rightarrow w_p [wc]) = 1$$

は保存される.

2.4 確率文法モデルの作成の手順

実際に確率文法モデルを作成する手順を示す.

1. 棋譜データベースを符号化する.
2. 符号化テキストに対し任意の n に対して n -gram 統計をとる,
3. 任意のテキストに対する頻度データベースを作成する.
4. 前出の仕様に沿って文法の構成素を定義する.

3 着手予想実験

今回の言語モデルが実際のどの程度有用であるかを調べるために、同一の棋譜テキストのデータベースを用いて符号化法ごとに3つの確率文法モデルを構築し、別の棋譜テキストデータベースの着手列を出題し、次着手を予想させ、どれぐらい少ない着手候補で正解するかを調べる実験を行った。

3.1 使用データ

実験に際し、以下のデータを使用した。

学習用データ：日本棋院「棋譜データベース'96」より、1993年から1995年までの棋譜の中から3426局、総手数722118手を使用した。

テストデータ：日本棋院「棋譜データベース'96」より、1960年の棋譜の中から10局、2203手を使用した。

3.2 実験と結果の評価

まず、学習用データベースを用いてそれぞれの符号化法 E_p, E_r, E_s に基づく確率文法モデル G_p, G_r, G_s を構築する。次に作成した文法モデルにテスト用棋譜データベースから、初手から n 手目までの着手 ($n = 1, 2, 3, \dots$ (その局の最終手) - 1) を出題する。そして次の着手の候補を列挙させ、その特徴を調べた。

はじめに、各文法モデルに対し、列挙する着手候補の平均分岐数を調べた。平均分岐数 $B = 2^H$ の指数部 H は、着手候補のエントロピーの平均である。設問の集合を T とおくと、 H は以下の式で表される。

$$H = -\frac{1}{|T|} \sum_{x \in T} \sum_{c \in \Sigma} p(c|x) \cdot \log_2(p(c|x))$$

次着手候補が全て同確率の場合、 B は最大値 $|\Sigma|$ 、次着手候補が1つしかない場合、 B は最小値1となる。

表1は、各文法モデルが列挙する次着手候補の平均分岐数である、1局全体の平均分岐数の他に、序盤から終盤まで20手単位で区切り、それぞれの部分で平均分岐数を調べてみた。符号ベースの場合には、1符号に対して一般に盤上の複数の着点が該当する。そこで、3つの文法モデル間での比較ができるように、ある符号の次着手評価値が a で、その符号に対応する着点が n 個存在するとき、該当する着点の評価値を $\frac{a}{n}$ とした。この値では、平均分岐数は G_R が最も少なく候補の列挙が出来る。一方、 G_s は符号の分布に偏りが多く、符号ベースでは平均分岐数は少なくできても、着点に反映させたときは同一符号になる点が多く分岐数が増大してしまった。今回適用した符号化法ではいずれも符号のもつ情報量が少なく、同一の符号になる着点が多く存在する。この問題の解決には文献[7]にある符号化法のように、同一符号となる着点の数を少なくする符号化法を適用することが必要である。

次に、正解の符号が出現した順位を記録した。今回は作成した文法モデルの比較のため、以下に補正順位を定義し、記録には補正順位を使用した。まず出題した手順を実際に盤面に反映させ、次に着手可能な空点全てに対して評価値を求める。そして評価値の降順に通常の順位を割り当てる。ここで、同一の符号になる n 個の m 位の着点は、全て補正順位 $m + \frac{m}{n}$ とする。これは、正解符号が n 個の着点 ($n \geq 2$) にあったとしたとき、正解手(実際に打った手と同一の着点)は n 目のうちどこか1目存在することになる。候補の着点をランダムに選択したとすれば平均して $\frac{n}{2}$ 回めの試行で正解手に的中することから、上記の順位づけを割り当てた。

補正順位を用いた評価として、再現率の算出をおこなった。再現率は第何位の候補手が実際に打った手と同一であったかを評価する、

再現率は

$$\text{再現率} = \frac{\text{実際に打った総手数}}{\text{有効順位以上の候補手の総数}}$$

であらわす。

図1は、各符号化法を用いて構築した確率言語モデルの再現率を示した。着手予想に適用する場合、実験で使用した文法モデルの中では、符号化法 E_p は最も有効である。しかし、一般に符号のもつ情報量は着点の決定に必要な情報量よりも低いため、次着手を一意に決定することができない。単独の文法モデルで一意に決定できる着手予想する場合、符号化法そのものを変更する必要がある。

それから、序盤、中盤、終盤の各部分において、再現率との関係を調査した。図2は着手候補上位8着点分における正解率を各文法モデルで調べたものである。文法モデルによらず、最初の20手および60~100手目の正解率に山が見られる。これらの部分では、次着手が過去の着手の系列に他の部分よりも大きく依存していることがわかる。また、200手以降の終盤にも正解率が向上するが、これは着手可能な点そのものが減少したことが原因である。

そして、複数の確率文法モデルが生成するテキストの出現を独立事象とみなし、各文法モデルが確率値の積を評価値とした次着手予想もおこなった。具体的には、まず着点 c 、それまでの手順 x に対し、符号化法 E_p, E_r, E_s で変換した符号または符号列を $c_p, c_r, c_s, x_p, x_r, x_s$ とする。そして文法モデル G_p, G_r, G_s のうち2つまたは3つの生成確率を組み合わせた評価値 $p_{(p,r)}, p_{(p,s)}, p_{(r,s)}, p_{(p,r,s)}$ の値を以下のように定めた。

$$\begin{aligned} p_{(p,r)}(c|x) &= p(c_p|x_p) \cdot p(c_r|x_r) \\ p_{(p,s)}(c|x) &= p(c_p|x_p) \cdot p(c_s|x_s) \\ p_{(r,s)}(c|x) &= p(c_r|x_r) \cdot p(c_s|x_s) \\ p_{(p,r,s)}(c|x) &= p(c_p|x_p) \cdot p(c_r|x_r) \cdot p(c_s|x_s) \end{aligned}$$

図3は、上式の評価値で着手予想をしたとき

の再現率の推移を示したものである。 G_p と G_r を組み合わせた評価値を適用した場合、上位の手の再現率が高くなっている。着手予想に文法モデルを組み合わせる場合、複数の文法モデルが決定する評価値を組み合わせる方法が有効であることがわかった。一方、3つの文法モデル全てを組み合わせさせた評価値を適用した場合、 G_p と G_r を組み合わせたときより再現率が低くなった。これは、 G_s がほとんど有効に働かないからで、4近傍から符号を決定するのでは一つの符号が持つ情報量が不足しているといえる。

4 おわりに

棋譜テキストに対し、それを記述する言語を生成する確率文法モデルの作成の手法を示した。確率文法の具体的利用として着手候補生成を取り上げ、次着手予想実験を行った。その結果、最初の20手および60手目から100手目付近の着手は比較的過去の着手に依存する手が選択されていることがわかった。また学習用データベースに70万手分余りの棋譜を用いた場合、複数の文法モデルを併用することで着手候補を絞り込むことができることがわかった。今後は、文法レベルの向上や情報量の多い符号の割り当て、学習用データベースの増補を施し、文法が生成する言語の範囲の調査や、着手予想の正解率の向上を図りたい。

参考文献

- [1] 清, 川島: "プロの手を真似する囲碁プログラムは強くなるか?", CGF ジャーナル Vol.2, pp.1-10, 1999.
- [2] 斎藤 "これからは囲碁プログラムが面白い", bit 別冊, ゲームプログラミング, pp.60-72, 1997.

- [3] 長尾, 森信: "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", 情報処理学会言語処理研究会報告 NL 96-1, pp.1-8, 1993.
- [4] 津田: "岩波講座ソフトウェア科学-9 数値処理プログラミング", 岩波書店, pp.151-218, 1988.
- [5] 有川, 宮野: "情報処理シリーズ-9 オートマトンと計算可能性", 培風館, pp.33-66, 1986.
- [6] 中村: " 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", ゲームプログラミングワークショップ'97, pp.96-105, 1997.
- [7] 中村, 梶山: "着手記号列の出現頻度に基づく囲碁棋譜からの定型手順の獲得", 情報処理学会ゲーム情報学研究会報告 99-GI-1, pp.107-114, 1999.

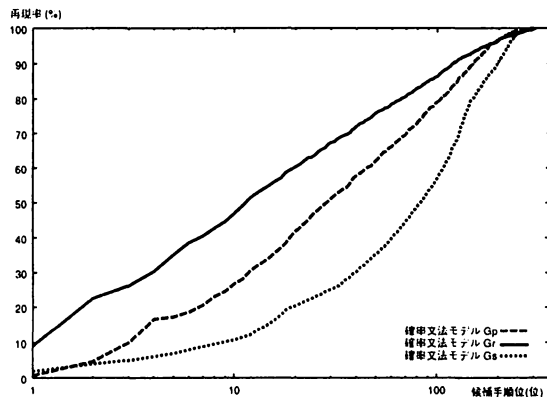


図 1: 確率文法モデルに対する着手予想テストの再現率

表 1: 文法モデルの平均分岐数の範囲

対象範囲(手)	G_p		G_r		G_s	
	符号	着点	符号	着点	符号	着点
1 ~ 20	9.6	70.9	12.1	49.5	2.3	247.0
21 ~ 40	22.6	149.4	11.8	45.7	3.0	107.5
41 ~ 60	24.4	153.7	7.9	28.1	4.1	100.8
61 ~ 80	22.9	133.7	8.1	27.3	4.6	103.5
81 ~ 100	19.6	103.2	7.1	22.9	4.5	101.7
101 ~ 120	28.2	134.8	7.8	22.3	6.4	113.1
121 ~ 140	29.0	128.0	10.8	30.2	6.2	100.9
141 ~ 160	34.3	140.0	12.8	29.9	8.6	105.6
161 ~ 180	35.6	131.6	15.3	34.5	9.8	102.7
181 ~ 200	29.9	94.9	16.1	36.1	10.1	94.6
201 ~	32.2	81.9	16.5	29.1	13.3	68.2
1局全体	25.1	114.8	11.1	31.0	6.0	104.7

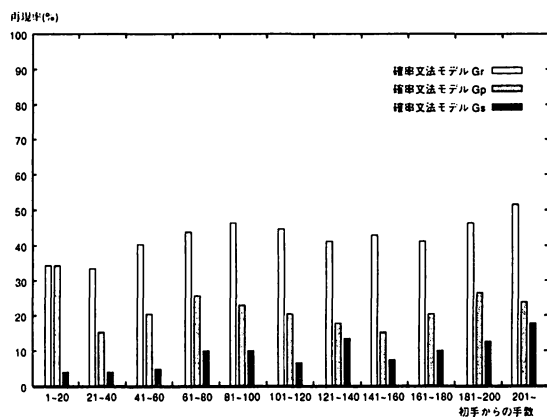


図 2: 初手からの手数と再現率の関係

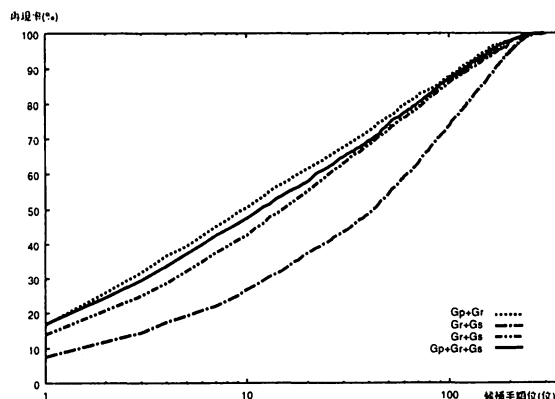


図 3: 確率文法モデルの併用時の着手予想テストの再現率