

ノイズを考慮した人物行動の文法の教師なし学習

木谷 クリス 真実^{†1} 佐藤 洋 一^{†1} 杉本 晃 宏^{†2}

自然言語の構文解析に用いられている確率文脈自由文法は、映像による人物の行動解析にも使われており、その有効性が報告されている。しかしながら、文の単語列と異なり、映像から得られる人物行動の記号列には多くのノイズが含まれているため、行動文法の学習が困難になる。したがって、高精度の文法学習を行うためには、ノイズ記号を除外した終端記号集合を特定する必要がある。そこで本研究では、最小記述長原理に基づき、ノイズを除外した終端記号集合とそれとともなう文法の獲得手法を提案する。提案手法では、終端記号の全組合せを評価し、各々の部分集合の下で得られた文法の複雑さと観測データの記号列尤度とのトレードオフを定量化する。これにより、評価値の高い終端記号集合と文法の候補を特定することができ、記号列に含まれるノイズを除去しつつ行動文法の基本構造を獲得することが可能となる。実験により提案手法の有効性を示す。

Unsupervised Learning of Human Activity Grammars from Noisy Input Sequences

KRIS M. KITANI,^{†1} YOICHI SATO^{†1}
and AKIHIRO SUGIMOTO^{†2}

Context-Free Grammars have been shown to be useful for applications beyond natural language analysis, specifically vision-based human activity analysis. However, vision-based symbol strings differ from natural language strings, in that a string of symbols produced by video can contain noise symbols, making grammatical inference very difficult. To obtain reliable results from grammatical inference, it is necessary to identify these noise symbols. We propose a new technique for identifying the best subset of terminal symbols to acquire the best activity grammar. Our approach uses the Minimum Description Length principle, to evaluate the trade-offs between model complexity and data fit to quantify the difference between the results of each terminal subset. The evaluation results are then used to identify a class of candidate terminal subsets and grammars that remove the noise and enable the discovery of the basic structure of an activity. In this paper, we present the validity of our proposed method based on experiments with artificial and real data.

1. はじめに

文脈自由文法 (Context-Free Grammar, 以下 CFG) は自然言語のモデルとして広く知られているが、人物の行動解析にも有用である。近年、映像による人物の行動解析に CFG が適用され、その有効性が示されている^{1),2),4),5),12)}。

CFG は記号集合と生成規則で形成され、与えられた文 (単語列) の文法的な構造を階層的に表現できることから、自然言語のモデルとしてふさわしい。一方、知覚心理学の観点から、人物行動は階層構造を持つ行動の記号列として解釈できるとされていることから¹¹⁾、CFG は人物行動を表すモデルとしてもふさわしい。人物の行動解析には隠れマルコフモデルや有限状態機械といった一次構造を持つモデルも提案されているが、階層構造を持つ CFG の方が人物行動を表すモデルとして適しているといえる。

CFG の応用にあたって、重要な課題の 1 つとして、文法の定義があげられる。これまでの研究は主に文法を事前に与え、その文法の応用に着眼しているが、文法の学習には触れていない。たとえば、Ivanov ら¹⁾ は映像から人物の単位行動を HMM で出力し、確率文脈自由文法 (Stochastic Context-Free Grammar, 以下 SCFG) で指揮者の手ぶりから拍子を認識しているが、ここでは文法は手で与えられている。Moore ら⁵⁾ は SCFG を用いて Black Jack のゲームを認識し、さらに文法の事前知識を利用して、単発的な誤り (記号挿入、置換と削除) に対処したが、文法はあらかじめ決められていた。同様に、Minnen ら⁴⁾ は事前に定義した SCFG から得られる事前知識を利用して、人物行動の一時的な遮蔽に対応した行動認識システムを実装している。

一方、映像から得られた人物行動の記号列から文法を学習する研究例は少なく、自然言語解析で使用されている既存のアルゴリズムをそのまま適用することにとどまっている。Wang ら¹⁰⁾ は HMM の出力を単位行動として表し、その行動を記号化し、COMPRESSIVE⁶⁾ (CFG の学習アルゴリズム) を用いて文法を学習するという手法を提案した。ただし、彼らの手法は低次の画像処理結果の精度に完全に頼っているため、画像処理の部分で問題が発生した場合、文法の学習精度が落ちるという問題をかかえている。つまり、COMPRESSIVE

^{†1} 東京大学
The University of Tokyo

^{†2} 国立情報学研究所
National Institute of Informatics

は、与えられた記号列には誤りが存在しないことを前提としているため、画像処理結果から出力される確率的な（ノイズのある）記号列には対応していない。

これに対して、本研究ではノイズを考慮した行動文法の獲得手法を提案する。提案手法はノイズによって劣化されている行動の観測（動作単位の記号列）から、潜在している生成文法の基本構造を獲得することができる。以下、最初に提案手法の基本的なアイデアについて述べたのち、提案手法の詳細を説明する。次に提案手法を用いた実験結果を報告し、結びを述べる。

2. 基本アイデア

いま、人物行動を表す記号列 S が与えられたとする。我々の目標は S の構造を説明する、簡潔かつ曖昧性の少ない文法を求めることである（ a や b は単位行動を表す記号）。

$$S \rightarrow a x b y c a b x c y a b c x$$

文法を学習するためには、記号列から、何らかの規則を持つパターンを探す必要がある。ただし、記号列にはノイズ記号も含まれているため、パターンを探すのは容易ではない。実際、上記の S の場合も、ノイズの影響で明らかな規則性が見えない。

まず、ノイズ記号がパターンを隠していると想定すると、ノイズ記号を削除すれば、パターンが見えてくると考えられる。そこで、この記号列に含まれているノイズを発見するためには、 S の中から任意に記号を削除し、規則性が発見されるかどうかを試してみればよい。この例の場合、任意に S の中から y をノイズ記号と仮定する。そして、その仮定のうえで記号 y を省略すると、以下のような記号列が得られる。

$$S \rightarrow a x b \quad c a b \quad x \quad c a b \quad c x$$

S には部分記号列 $c a b$ が 2 回繰返されているが、これではまだ明らかな規則性が見えていない。

さらに、もう 1 つの記号 x をノイズ記号と仮定すると、今度は記号列が明らかなパターン（基本構造）で構成されていることが見えてくる。

$$S \rightarrow a b c \quad a b c \quad a b c$$

すなわち、記号列は部分記号列 $a b c$ を 3 回繰り返したものである。新たな規則 A で S を置き換えると、より少ない記号数で元の記号列の基本的な構造を表すことができる。

$$S \rightarrow A A A$$

$$A \rightarrow a b c$$

この例では、 x と y をノイズ記号と仮定することによって、 S の規則性が明らかになり、

簡潔な規則 ($A \rightarrow a b c$) が発見された。そして、その発見された規則は、記号列 S を的確に表現している（曖昧性がないという意味）。 S を的確に表す簡潔な規則が得られた結果から、 x と y は高い可能性で、ノイズ記号である。

3. 準備

3.1 ノイズの特徴

映像から得られた単位行動の記号列を用いて行動文法を学習する際、2 つの要因によるノイズが記号列に含まれていると考えられる。

第 1 の要因は、画像処理によるシステムノイズである。たとえば、照明変動や物体の遮蔽による記号の見逃し（削除）や誤検出（挿入）があげられる。このような頻繁に見逃される記号や誤検出される記号は学習に望ましくない。

第 2 の要因は、人物行動に含まれている余分な動きによる挿入ノイズである。人間の行動を認識するためには重要な単位行動を観察することが重要である。しかし、人物の行動にはばらつきがあり、認識に重要な単位行動（一次的な単位行動記号）と不要な単位行動（二次的な単位行動記号）が混在している。この二次的な記号は、重要な一次的な記号の間を埋めるという特徴を持ち、また不定期に発生する傾向があるため、文法学習の障害となる。

すべてのノイズの種類（削除と挿入）を同時に扱うことは困難であることから、本研究では、いくつかの仮定を設ける。

- (1) 挿入によるノイズ記号のみを考慮する。
- (2) 記号列にはノイズ記号と非ノイズ記号が混在している（ノイズ記号のみ、非ノイズ記号のみの記号列はない）ことを仮定する。
- (3) 記号はノイズまたは非ノイズのいずれかである（1 つの記号が同時にノイズと非ノイズにはなれない）と仮定する。
- (4) 非ノイズ記号の出現には規則性があると仮定する。

3.2 文脈自由文法を用いた人物行動モデル

冒頭で述べたように、文脈自由文法は人物の階層的な行動を表すモデルに適している。CFG は 4 つの要素の組 $G = \{T, N, S, R\}$ によって定義される。 T, N, R はそれぞれが終端記号 $\{t_1, \dots, t_m\}$ 、非終端記号 $\{N_1, \dots, N_n\}$ 、生成規則 $\{R_1, \dots, R_p\}$ の集合であり、開始記号 S は $S \in N$ である。生成規則は $A \rightarrow \alpha$ という形を持ち、 A は非終端記号、 α は終端記号と非終端記号の記号列を表す。各々の生成規則に確率 $P(A \rightarrow \alpha)$ が付加される場合、 $\sum_k P(A \rightarrow \alpha_k) = 1$ を満たす。このような文脈自由文法は確率的文脈自由文法と呼ば

Context-Free Grammar G	
$T =$	$\{walk, appear, disappear\}$
$N =$	$\{S, ENTER, EXIT\}$
$R =$	$\{S \rightarrow ENTER\ EXIT,$ $ENTER \rightarrow appear\ walk,$ $EXIT \rightarrow disappear\}$

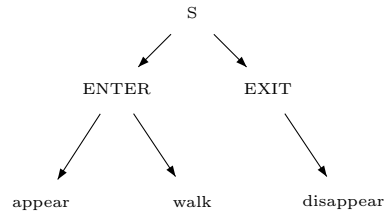


図 1 文脈自由文法が表す人物行動の階層構造

Fig. 1 An activity (i.e. a sequence of primitive actions) described by a CFG.

れている。

本研究では、終端記号 t は単一の行動（単位行動）を表し、非終端記号 N は（1 つもしくは複数の）単位行動を表している。開始記号 S は 1 つの行動シーケンス（行動の記号列）を表す記号である。したがって、文脈自由文法 G を用いることにより、人物の行動から得られる記号列の階層構造を表すことができる（図 1 参照）。

4. 提案手法

このような考えに基づき、提案手法では各々の記号をノイズ記号と仮定し、その仮定を最小記述長原理で検証する。そして、その検証結果から最も単純かつ曖昧性の少ない文法を導く、最適な終端記号集合を特定することにより、学習データから基本的な行動文法を得る。結果、画像処理および人物行動に起因するノイズを含む観測からでも行動文法を獲得することを実現する。

本章では、前述で紹介した基本アイデアの適用方法を具体的に説明し、ノイズの特徴について論じたのち、文脈自由文法の応用を説明する。そしてノイズを含んだ行動記号列からの文法の獲得手法を紹介する。ただし、ここでは映像から得られた記号列を用いた文法獲得に着目しているため、単位行動記号の抽出方法に対する議論は、本論文の範囲外であるとして行わない。

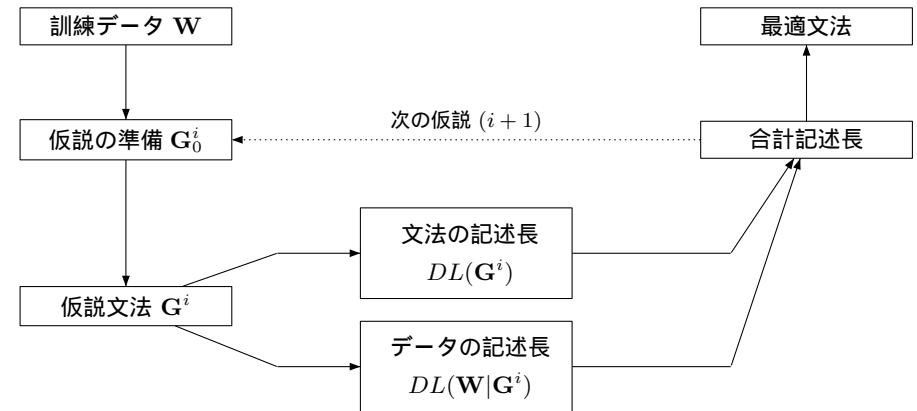


図 2 最小記述長原理を用いた文法学習の枠組み

Fig. 2 Flowchart of our MDL-based grammar learning method.

4.1 ノイズ仮説の準備

映像から得られた学習データ W から文法を学習するためには、まずノイズ記号を特定したい。ここで、学習データ W は m 個の行動記号列 $W_1 \dots W_m$ 、行動記号列（行動シーケンス） W_i は k 個の単位行動記号 w_1, \dots, w_k から形成されているものとする。ノイズ記号を直接的に探すことは困難であるため、本手法では、各々の記号に対して順番に、ノイズであるという仮説を立て、すべての仮説の妥当性を最小記述原理の枠組みで検証する（図 2 参照）。つまり、全数検索を行い、 $2^n - 2$ の仮説を検証する（ n は記号の種類の数）。たとえば、3 つの記号種類、 a, b, c が与えられた場合、6 つの仮説が得られる： $\{a\}$, $\{b\}$, $\{c\}$, $\{a, b\}$, $\{a, c\}$, $\{b, c\}$ （ $\{\cdot\}$ はノイズ記号を表すとする）。

最初に、1 つの仮説（たとえば、 c がノイズであるという仮説）に対して、仮説を反映した文法（以下、仮説文法）を学習するため、ノイズ記号に関する仮説を表現する以下のような初期文法を事前に準備する。

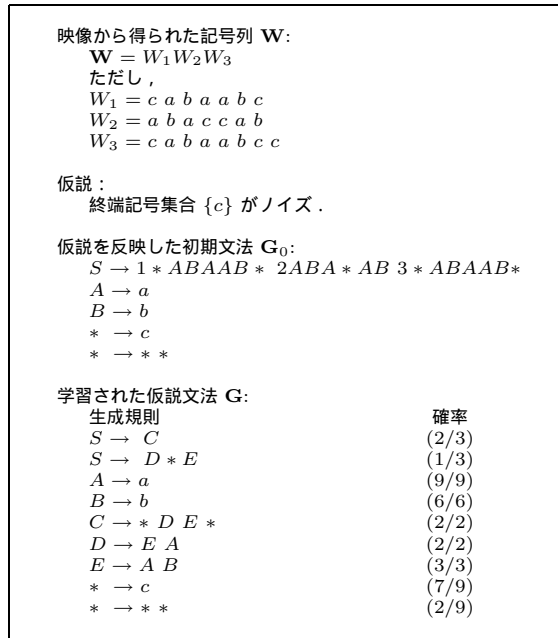


図 3 仮説文法の学習の例
 Fig. 3 Example of a single hypothesis grammar.

$$G_0 = \left\{ \begin{array}{l} N_1 \rightarrow w_1^+, \quad * \rightarrow w_1^-, \\ \dots \quad \dots \\ N_u \rightarrow w_u^+, \quad * \rightarrow w_u^-, \\ S \rightarrow \mathbf{W}', \quad * \rightarrow ** \end{array} \right\}.$$

初期文法の準備は、学習データ \mathbf{W} の前処理とも考えられ、 \mathbf{W} を仮説を反映した記号列 \mathbf{W}' に変換する処理である。

初期文法 G_0 の生成規則 $N_i \rightarrow w_i^+$ は非ノイズ記号 w_i^+ が非終端記号 N_i によって生成されることを表し、この生成規則は w_i^+ の存在を \mathbf{W}' に残す効果がある (図 3 では、 $A \rightarrow a$ によって a が A で置き換えられている)。仮説から得られた u 種類の非ノイズ記号に対して u 個の生成規則を準備する。

次に、生成規則 $* \rightarrow w_i^-$ はノイズ記号 w_i^- がノイズを意味する非終端記号 $*$ によって生成されることを表し、この生成規則によりどのノイズ記号も同じ非終端記号で表現することができる。仮説から得られた v 種類のノイズ記号に対して v 個の生成規則を準備する。

生成規則 $* \rightarrow **$ は隣接するノイズ記号が非終端記号 $*$ によって生成されることを表し、記号列内で隣接するノイズ記号を 1 つの非終端記号 $*$ で省略する効果がある (図 3 ではすべての c が $*$ で置き換えられ、隣接している c は 1 つの $*$ にまとめられている)。

最後の生成規則 $S \rightarrow \mathbf{W}'$ は、仮説の反映した学習データ \mathbf{W}' が開始記号 S によって生成されることを表し、仮説の影響を学習データに反映する。 \mathbf{W}' は、 \mathbf{W} の各々の単位行動 w_i を、対応する生成規則で置き換えたものである (図 3 の G_0 の規則 S を参照)。なお、各記号列 (行動シーケンス) W_i の開始点を示す固有な記号を、先頭に付加する (図 3 では、1, 2, 3)。この開始記号は学習後の処理で使用される。

4.2 仮説文法の取得

本節では、前節で構築した初期文法をもとに、文脈自由文法の発見的学習アルゴリズム、COMPRESSIVE⁶⁾を用いた仮説文法の学習方法を簡単に紹介する。COMPRESSIVE は、学習データ内に頻繁に出現する部分記号列から新たな生成規則を作り、逐次的に部分記号列をその生成規則で置き換えて、文脈自由文法を学習する方法である。新しい生成規則を作るための最適な部分記号列 ν は、文法全体の長さの減少 ΔDL を最大にする部分記号列である。

$$\arg \max_{\nu} \{M_{\nu} \cdot N_{\nu} - (M_{\nu} + 1) - N_{\nu}\}. \tag{1}$$

ただし、 M_{ν} は部分記号列の長さ、 N_{ν} は部分記号列の出現回数である (図 4 参照)。なお、本手法では、生成規則の適用回数を保存することにより、確率的な文脈自由文法の学習に拡張している (図 3 の文法 G の生成規則には確率が付加されている)。

COMPRESSIVE が収束したのち、生成規則の確率の計算と規則 S の分解を行う。前節で述べたように規則 S の右辺 \mathbf{W}' は m 個の行動記号列の結合であるため、最後に \mathbf{W}' を分ける必要がある。具体的には、開始記号を用いて、規則 $S \rightarrow \mathbf{W}'$ を個々の規則 $S \rightarrow W'_1, \dots, S \rightarrow W'_n$ に分ける処理を行うということである ($n \leq m$)。ただし、 $S \rightarrow W'_i$ は 1 種類の行動を表す規則であるため、重複する規則 S は一回のみ文法に加えられる。各生成規則の確率は出現頻度の比を用いて式 (2) で計算する。

$$P(N \rightarrow \nu_i) = \frac{c(N \rightarrow \nu_i)}{\sum_j c(N \rightarrow \nu_j)}, \tag{2}$$

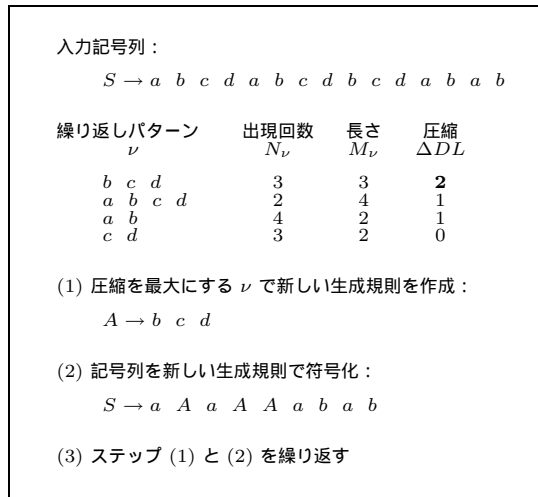


図 4 COMPRESSIVE アルゴリズムの例
 Fig. 4 Example of the COMPRESSIVE algorithm.

ただし、 $c(\cdot)$ は規則の使用回数、 N は任意の非終端記号、規則の右辺 ν は部分記号列（1 つまたは複数の記号）を意味する。

4.3 最小記述長原理による仮説文法の評価

本研究の目標は映像から得られた記号列を、簡潔かつ的確に表す文法を探すことである。この目標を達成するために、最小記述長原理の枠組みを用いて、仮説文法の簡潔さを表現する文法記の記述長 $DL(\mathbf{G})$ と文法の表現力を表すデータ尤度の記述長 $DL(\mathbf{W}|\mathbf{G})$ を最小にする最適な文法 $\hat{\mathbf{G}}$ を求める（式 (3)）。シャノンの符号法により、確率変数 X の記述長は X の確率の対数としても表せるため、最小記述長原理を対数の和としても表現できる（式 (4)）。

$$\hat{\mathbf{G}} = \arg \min_{\mathbf{G}} \{ DL(\mathbf{G}) + DL(\mathbf{W}|\mathbf{G}) \} \quad (3)$$

$$= \arg \min_{\mathbf{G}} \{ -\log P(\mathbf{G}) - \log P(\mathbf{W}|\mathbf{G}) \}. \quad (4)$$

本手法では、文献 9) で提案された手法を用いて、仮説文法の記述長を計算し、データの記述長はデータ尤度 $P(\mathbf{W}|\mathbf{G})$ から求める⁷⁾。次に、この 2 つの記述長の計算方法と意味を説明する。

4.3.1 文法の記述長

最小記述長原理の最初の項は文法の記述長である。 $DL(\mathbf{G})$ は文法の簡潔さまたは複雑さを表す値であり、間接的にデータの規則性も表している。最小記述長原理の枠組みでは、簡潔な文法ほど、文法の記述長が短い。

文法の記述長は、文法の構造 G_S （たとえば、 $A \rightarrow a$ ）と文法のパラメータ Θ_G （たとえば、 $P(A \rightarrow a) = 0.5$ ）に分けることができる。

$$\begin{aligned} DL(\mathbf{G}) &= -\log P(G_S, \Theta_G) \\ &= -\log P(\Theta_G|G_S) - \log P(G_S) \\ &= DL(\Theta_G|G_S) + DL(G_S), \end{aligned} \quad (5)$$

さらに、文法構造の記述長 $DL(G_S)$ は、各々の生成規則 R の長さの記述長 $DL(l_R)$ と、記号の記述長 $DL(s_R)$ の和として計算する。

$$DL(G_S) = \sum_{R \in \mathbf{G}} (DL(l_R) + DL(s_R)). \quad (6)$$

生成規則の長さ l_R の最小限は 2 であり（たとえば、 $S \rightarrow a$ の長さは 2）、ほとんどの生成規則の長さは 2 から 5 の間であるという性質を持つ。この性質から、 l_R はポアソン分布から生起すると近似することができる⁹⁾。したがって、 $DL(l_R)$ を次の式で表す。

$$DL(l_R) = -\log \frac{e^\mu \mu^{(l_R-1)}}{(l_R-1)!}. \quad (7)$$

ただし、 μ は生成規則の長さの事前平均である（実験では $\mu = 3$ ）。

文法で使用されている各々の記号は、文法中同じ確率で使用されると仮定すると、1 つの記号の記述長は $\log_2 |\Sigma|$ ビットとなり（ Σ は記号の集合）、長さ l_R を持つ生成規則 R は $l_R \log |\Sigma|$ ビットの記述長を要する。

$$DL(s_R) = l_R \log |\Sigma|. \quad (8)$$

次に、パラメータの記述長 $DL(\Theta_G|G_S)$ を、パラメータの確率 $P(\Theta_G|G_S)$ から求める。非終端記号 N は、1 つまたは複数の生成規則の左辺であり（たとえば、 S は $S \rightarrow C$ と $S \rightarrow D * E$ の左辺である）、左辺 N を持つ生成規則集合 \mathbf{R}_N のパラメータ（確率）は多項分布 θ_N である（図 3 では、 S のパラメータ $\theta_S = (\theta_{S \rightarrow C}, \theta_{S \rightarrow D * E})$ は $(\frac{2}{3}, \frac{1}{3})$ である）。左辺 N を持つ生成規則集合 \mathbf{R}_N のパラメータ θ_N は、どのような値をとるの事前に分らないため、多項分布の事前確率を、一様に分布したディリクレ分布で表す（ディリクレ分布の妥当性に関する議論は文献 9) を参照）。これにより、パラメータ $\theta_N = (\theta_1, \dots, \theta_n)$ の確率 $P_N(\theta_N|\mathbf{R}_N)$ は、式 (9) で表される。

$$P_N(\theta_N | \mathbf{R}_N) = \frac{1}{B(\alpha_1, \dots, \alpha_n)} \prod_{i=1}^n \theta_i^{\alpha_i - 1}. \quad (9)$$

ただし, $\alpha_1, \dots, \alpha_n$ は各生成規則確率の事前重み, 正規化項の B は多変量ベータ関数である. なお, 事前重み α_i は i 番目の生成規則が実行される確率の事前の期待値である. 具体的には, θ_i の事前の期待値は $\frac{\alpha_i}{\alpha_0}$ であり, $\alpha_0 = \sum_i \alpha_i$ は事前重みの合計値である. なお, 文献 9) と同様, どの生成規則も同じ確率を持ち, 各々の確率の事前期待値も等しいとする. そのため, 各々の生成規則の確率 θ_i は一様に分布しており, 合計値は $\alpha_0 = 1$ で, 各々の α_i は同じ値を持つものとして扱う.

最後に, 各々の非終端記号 N のパラメータ記述長の和をとることによって, 文法全体のパラメータ記述長が次の式から得られる.

$$DL(\Theta_G | G_S) = \sum_{N \in \Sigma} -\log P_N(\theta_N | \mathbf{R}_N). \quad (10)$$

4.3.2 データ尤度の記述長

小さい記述長を持つ仮説文法は簡潔である反面, 表現力に欠けている場合も多く, その表現力を評価することが必要である. 最小記述長原理の式 (3) の第 2 項はデータ尤度の記述長 $DL(\mathbf{W} | \mathbf{G})$ であり, データを詳細に記述できる文法に高い評価を与えるものである. 逆に, データ尤度の記述長を, 文法の曖昧性に対するペナルティ値と見なすこともできる. 曖昧性がない文法の記述長は 0 であり, 曖昧性が高いほど, 文法の記述長が大きくなる.

文法 \mathbf{G} が与えられたときの行動記号列 W の尤度 $P(W | \mathbf{G})$ は, 一般的に内側アルゴリズムで求めるが, 本手法で学習された文法はチョムスキー標準形ではないため, Pynadath⁷⁾ が提案したベータ確率を用いて求める (付録 A.1 参照).

すべての記号列 (学習データ) の尤度 $P(\mathbf{W} | \mathbf{G})$ は式 (11) で求まる. 最後に式 (12) で尤度を記述長に変換する.

$$P(\mathbf{W} | \mathbf{G}) = \prod_{i=1}^n P(W_i | \mathbf{G}), \quad (11)$$

$$DL(\mathbf{W} | \mathbf{G}) = -\log P(\mathbf{W} | \mathbf{G}). \quad (12)$$

これで, 各々の仮説文法 \mathbf{G} の文法記述長とデータ尤度の記述長を計算し, 最適な文法 $\hat{\mathbf{G}}$ が合計記述長 $DL(\hat{\mathbf{G}} | \mathbf{W})$ を最小にする文法として得られる.

5. 実験

最初の実験では既知の文法を用いて, 人工的にノイズを含んだ学習データを生成し, 正解文法を学習するための条件を明確にする. 2 つ目の実験では実用例を通して本手法の有用性を示す.

5.1 人工データによる実験

3.1 節で述べた 4 つの仮定を満たす様々な文法を作成し, その文法で学習データを作成した. 文法作成およびデータ作成に使用した条件は以下のとおりである.

1 つ目の条件は文法が表現できるパターン数である. 1 種類のパターンのみを記述する文法 (S 規則が 1 つ), 2 種類のパターンを記述する文法 (S 規則が 2 つ), そして 3 種類のパターンを記述する文法 (S 規則が 3 つ) を準備した. ただし, 1 つの文法から生成されるパターンは同じ非ノイズ記号で形成され, 記号の順番が異なる. 図 5 では 2 種類の記号列パターンを生成する文法の例を示す.

2 つ目の条件は文法のノイズ記号と非ノイズ記号の数の比率である. 記号の数を 6 個から 10 個に変え, ノイズ記号と非ノイズ記号の比率も変えながら評価した. ノイズ記号は *INSERT* 規則により記号列に挿入した.

3 つ目の条件は学習データの大きさである. 生成文法を用いて, ランダムにサンプル記号列を生成し, サンプル記号列の数 d を変えて, 提案手法を評価した. サンプル記号列の数

S	→	EN ACT EX	(0.5)
S	→	EX ACT EN	(0.5)
EN	→	A	(0.5)
EN	→	A INSERT	(0.5)
EX	→	B	(0.5)
EX	→	B INSERT	(0.5)
ACT	→	C	(0.5)
ACT	→	C INSERT	(0.5)
A	→	a	(1.0)
B	→	b	(1.0)
C	→	c	(1.0)
INSERT	→	nd	(0.333)
INSERT	→	ne	(0.333)
INSERT	→	INSERT INSERT	(0.334)

図 5 3 つの非ノイズ記号 (a, b, c) と 2 つのノイズ記号 (nd, ne) を含む 2 パターン文法 (挿入ノイズ)
Fig. 5 Two pattern grammar with 3 non-noise symbols and 2 noise symbols (insertion noise).

は $d = 50, 150, 300, 500, 1000$ の大きさで評価を行った。

5.1.1 挿入ノイズを含むデータ

各々の生成文法に基づく人工的学習データの解析結果を表 1 に示す。「Pat」は文法が生成できる記号列パターンの数、「Non-noise」は非ノイズ記号の数、「Noise」はノイズ記号の数、 d は学習データ（サンプル記号列）の数である。結果は、1 つの条件設定に対して、正解文法のランクで示したものであり、正解文法が本手法の最適解として得られた場合、正解文法のランクは 1 になる。

ノイズ記号の数が増えても正解文法のランクはそれほど変わらないことから、正解文法を得るために必要なサンプル記号列の数はノイズ記号の数ではなく、生成可能なパターン (pat) の数と非ノイズ記号 (non-noise) の数に依存していることが分かる。よって、生成文法が表現できるパターンの数が増えるほど、より多くの学習データが必要となり、非ノイズ記号の数が増えるほど、正解文法を得るために必要なサンプル記号列の数が増加する。

学習データの数が十分である場合、正解文法が 1 位にランクしている一方、学習データの数が十分ではない場合、正解文法より小さい文法が最適文法として得られてしまう。たとえば、表 2 では、正解文法は 3 にランクしており、最適文法（ランクが 1）としては、正解の非ノイズ記号の部分集合 (a, b) を使用した文法が得られている。ここでは、学習データの数が不十分であり、式 (3) におけるデータ尤度の記述長の項に比べ文法記述長の項の影響が支配的となり、結果的に小さい文法がより小さい記述長を得ている。学習データが不足している場合の対応策に関して 5.1.3 項で紹介する。

5.1.2 システムノイズを含むデータ

本手法では挿入ノイズ、すなわち、余分な動作によって挿入された動作記号を前提としているが、ここではシステムノイズの影響を簡単に調べる。前項と同様に 3 つの条件のもとで、複数の生成文法を提案手法で解析した。ただし、INS（挿入）規則に加え、SUB（置換）規則と削除規則を導入し、より複雑なノイズをサンプル記号列に含むこととする。記号の置換えは削除と挿入の組であり、非ノイズ記号がノイズ記号または他の非ノイズ記号で置き換えられることを意味する。図 6 で示している文法では、非ノイズ記号の置き換えが 10% の確率で起こることになっている。前項の実験と同様に、生成文法の条件と解析結果を表 3 に示す。

挿入ノイズを含んだデータの解析結果と同様に、正解文法の獲得に必要な学習データの数は、生成文法の複雑度に比例している。しかし、図 1 と図 3 を比べると、正解文法を獲得するために必要な学習データの数は、挿入ノイズよりシステムノイズの方が多いことが分

表 1 人工データによる実験結果（挿入ノイズ）
Table 1 Results using artificial data (Insertion noise).

Patterns	non-noise	noise	Number of samples (d)				
			$d = 50$	$d = 150$	$d = 300$	$d = 500$	$d = 1000$
			Rank				
1	3	3	3	1	1	-	-
1	3	4	3	1	1	-	-
1	3	5	3	1	1	-	-
1	3	6	5	1	1	-	-
1	3	7	4	1	1	-	-
1	4	3	12	4	1	1	1
1	4	4	15	4	1	1	1
1	4	5	11	4	1	1	1
1	4	6	14	4	1	1	1
1	5	3	30	15	5	1	1
1	5	4	34	15	5	1	1
1	5	5	54	15	5	1	1
2	3	3	11	4	1	1	-
2	3	4	12	4	1	1	-
2	3	5	28	4	1	1	-
2	3	6	8	4	1	1	-
2	3	7	30	4	1	1	-
2	4	3	25	11	5	1	1
2	4	4	49	11	5	1	1
2	4	5	28	11	5	1	1
2	4	6	65	13	5	1	1
2	5	3	55	35	16	6	1
2	5	4	91	43	16	6	1
2	5	5	242	34	16	6	1
3	3	3	23	5	1	1	-
3	3	4	28	5	1	1	-
3	3	5	28	6	1	1	-
3	3	6	84	7	1	1	-
3	3	7	177	7	1	1	-
3	4	3	37	26	11	4	1
3	4	4	71	18	11	4	1
3	4	5	102	43	11	4	1
3	4	6	213	89	10	3	1
3	5	3	85	69	26	16	5
3	5	4	87	80	30	16	5
3	5	5	181	136	27	17	5

表 2 ランク付けされた仮説文法．正解文法は太字で示す
Table 2 Ranked hypothesis grammars. True grammar marked in bold.

Rank	Symbols		$DL(G)$	$DL(W G)$	Total
1	a b	2	117.85	487.04	604.89
2	a c	2	126.81	489.01	615.82
3	a b c	3	348.69	327.84	676.52
4	b c	2	187.57	517.32	704.89
5	a	1	85.58	689.34	774.92
6	c	1	89.69	703.08	792.77
7	b	1	113.80	758.57	872.37
8	a ne	2	362.22	622.20	984.42
9	a nd	2	403.36	604.69	1008.05
10		1	70.82	942.46	1013.28
11	nd	1	223.37	826.17	1049.53
12	ne	1	223.37	854.10	1077.46
13	a c ne	3	664.92	415.79	1080.71
14	c nd	2	540.10	566.35	1106.45
15	c ne	2	512.91	604.70	1117.61
16	a c nd	3	749.00	399.35	1148.35
17	a b nd	3	774.74	397.38	1172.12
18	a b ne	3	758.90	422.24	1181.14
19	b nd	2	608.16	636.22	1244.38
20	b ne	2	608.30	675.64	1283.94
21	b c nd	3	981.49	398.73	1380.22
22	b c ne	3	999.61	422.70	1422.31
23	a b nd ne	4	1268.58	260.39	1528.97
24	a b c nd	4	1300.13	257.39	1557.52
25	a b c ne	4	1300.13	257.39	1557.52
26	a c nd ne	4	1300.13	257.39	1557.52
27	b c nd ne	4	1300.13	257.39	1557.52
28	a b c nd ne	5	1300.13	257.39	1557.52
29	nd ne	2	885.68	706.60	1592.28
30	a nd ne	3	1145.23	489.99	1635.22
31	b nd ne	3	1151.62	487.99	1639.61
32	c nd ne	3	1280.75	377.39	1658.15

S	→	EN	ACT	EX	(0.5)
S	→	ACT	EX	EN	(0.5)
EN	→	A			(0.45)
EN	→	A	INS		(0.45)
EN	→	SUB			(0.10)
EX	→	B			(0.45)
EX	→	B	INS		(0.45)
EX	→	SUB			(0.10)
ACT	→	C			(0.45)
ACT	→	C	INS		(0.45)
ACT	→	SUB			(0.10)
A	→	a			(1.0)
B	→	b			(1.0)
C	→	c			(1.0)
INS	→	nd			(0.25)
INS	→	ne			(0.25)
INS	→	nf			(0.25)
INS	→	INS	INS		(0.25)
SUB	→	nd			(0.30)
SUB	→	ne			(0.30)
SUB	→	nf			(0.30)
SUB	→	A			(0.0333)
SUB	→	B			(0.0333)
SUB	→	C			(0.0334)

図 6 二種類のパターンを生成する文法 (3 つの非ノイズ記号と 3 つノイズ記号)
Fig. 6 Two pattern grammar with three non-noise symbols and system noise.

表 3 人工データの実験結果 (システムノイズ)
Table 3 Results with synthetic data (inherent insertion and system noise).

Patterns	non-noise	noise	Number of samples (d)				
			$d = 50$	$d = 150$	$d = 300$	$d = 500$	$d = 1000$
			Rank				
1	3	3	12	3	1	1	1
2	3	3	15	7	4	2	1
3	3	3	23	17	7	4	1

かる。

5.1.3 不十分な学習サンプル数

学習データ数が不十分である場合，正解文法のデータ尤度 (表現力) は評価されず，少ない数の記号を使用する単純な文法が優先されてしまう。この問題に対し，先行研究ではデータ乗数 λ の導入が提案されている^{8),9)}。

$$\lambda DL(\mathbf{G}) + DL(\mathbf{W}|\mathbf{G}). \tag{13}$$

本手法では λ を文法の事前重みと呼ぶことにする。 λ が大きいときには，単純な文法が優先され，一方， λ が小さいときには，データを詳細に説明する複雑な文法が優先される。

前項で解析したシステムノイズを含んだ学習データを用いて事前重みの逆数を $1 \leq \lambda^{-1} \leq 20$

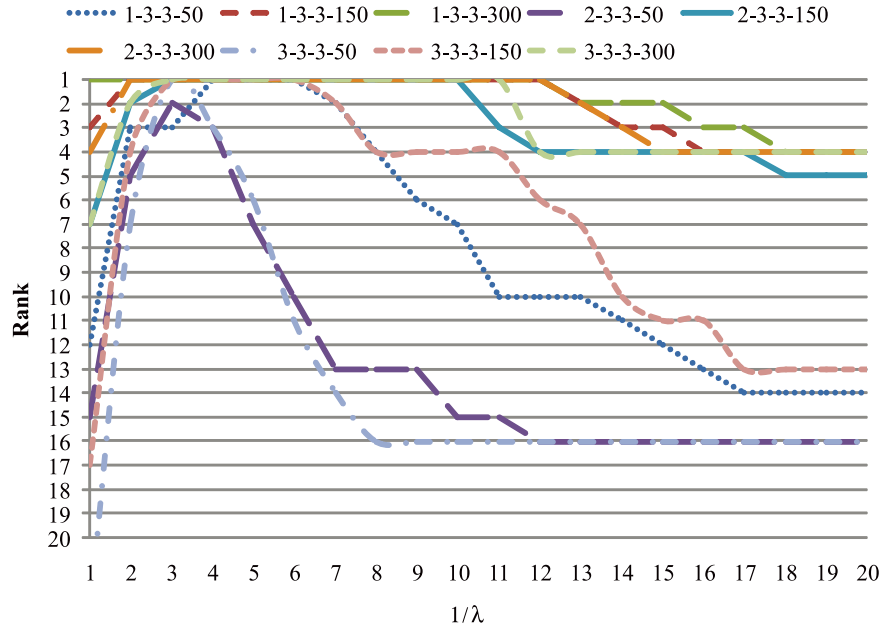


図 7 重み λ 値による正解文法のランクの変動。データラベルの 2-3-3-50 は、2 種類のパターン、3 個の非ノイズ記号と 3 個のノイズ記号、という特徴を持つ文法から得られた、50 個のサンプル記号列の解析結果を意味する
 Fig. 7 Rank of the true grammar for different values of λ . Data label 2-3-3-50 represents the results from a grammar with 2 patterns, 3 non-noise symbols and 3 noise symbols used to create 50 data samples.

に変えた場合の正解文法のランクの変動を図 7 に示す。解析データに関しては、正解文法はランク 3 以内であることから、 $\lambda^{-1} = 3$ が最適値である。このように、事前重み λ を導入することにより、学習データが多少不足している場合でも、正解文法を獲得することが可能となる。当然ながら、文法の複雑さにより、不十分な学習データから正解文法が得られないこともある。

5.2 実データによる実験

実データを用いて本提案手法の有用性を検証するために、某店舗内のレジ上に監視カメラを設置し、人物行動の文法学習を行った。図 8 に表示されているように、CCD カメラをレジの上部に設置し、店員と客の行動パターンを映像として録画した。

本実験では色情報を用いて複数の物体（手やトレイやお金等）を検出し、10 種類の単位



図 8 頭上の CCD カメラを用いて取引行動を録画
 Fig. 8 Overhead CCD cameras used to record purchase transactions.

行動（終端記号）を出力するプログラムを準備した。手とトレイの検出例を図 9 に示す。プログラムから得られた単位行動記号の説明を表 4 に示す。映像の処理はオフラインで行い、映像データ中の 60 人の取引から、60 個のサンプル記号列が自動的に得られた。ただし、各取引（各々の記号列 W_i ）の分割は手で行った。記号列の最大長は 11 で、最小長は 3 であり、すべての記号列を合わせて長さ 429 個の記号が得られた。この記号列を入力（学習データ）として、解析を行った。

各々のノイズ仮説に対して、学習データから仮説文法を学習し、最小記述長原理に基づき、仮説文法を評価した。具体的に、10 種類の記号で生成される 1022 のノイズ仮説を評価し、全数の合計記述長を計算した。

既知の生成文法を持つ人工データから得られた最適文法の評価は簡単である一方、未知の生成文法を持つ実データから得られた最適文法の真偽評価は困難である。実データから得

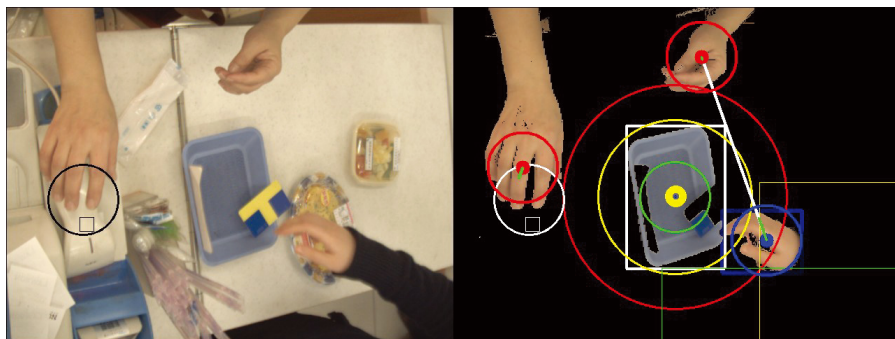


図 9 画像処理によって 10 種類の単位行動記号を抽出

Fig. 9 Ten primitive actions extracted from the image processing module.

表 4 基本動作（終端記号）の定義

Table 4 Definition of primitive actions (terminal symbols).

	終端記号	記号の意味
1	C_AdMon	お客がトレイにお金を置く
2	C_MvTry	お客がトレーを動かす
3	C_RmMon	お客がトレイからお金を取る
4	E_HndRtn	店員の手がしばらくしてから戻る
5	E_Intrct	店員がお客とやりとりをする
6	E_MvTry	店員がトレーを動かす
7	E_RmMon	店員がトレイからお金を取る
8	E_RtnScn	店員がスキャナを返す
9	E_TkRcpt	店員がレシートを取る
10	E_TkScn	店員がスキャナを取る

られた最適文法の評価が困難である理由の 1 つとしては、複数の正解文法が存在することが考えられる。たとえば、機械学習による文書要約の研究でも同様な問題が見られ³⁾、1 つの文を要約する際に、複数の正解要約文が存在する可能性がある。Knight ら³⁾は複数の上位に並んだ要約文をユーザに示し、最終決断をユーザに委ねることを提案している。また、要約文の単語数を固定し、同じ単語数を使用した要約文候補の中から最適なものを選ぶ方法も提案している。本手法の実用においては同様に文法を選択することができるが、定量的な評価を行うためには 1 つの最適文法を決めることが必要である。したがって、以下の実験では同じ記号数を使用した文法ごとの最適な文法（候補文法）をリストに示し、

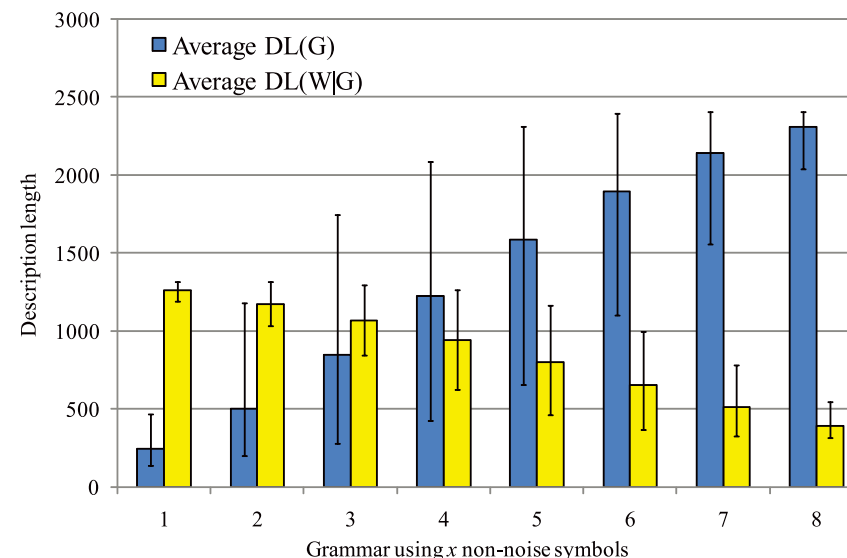


図 10 文法の記述長とデータ尤度記述長の変動割合。データ尤度の記述長より文法の記述長が広範囲に変動する（エラーバーが記述長の最大値と最小値の範囲を示す）

Fig. 10 Imbalance between description lengths - Range of the grammar description length is consistently larger than the range of the likelihood description length (error bars shows range of description length).

最小記述長を持つ文法を最適文法として 1 つを選択する。

実データのサンプル数（60 個）が少ないため、学習データが不十分である可能性が高い。なお、実データの特徴としては、人工データと異なり、文法記述長の分散の範囲がデータ尤度の記述長の分散範囲より大きく、使用する非ノイズ記号の数により、分散範囲の差も異なる（図 10）。これらの原因としては、実際の行動文法は 5.1 節の人工文法と比べて複雑であり、潜在している生成文法が出力できるパターンが多いことが考えられる。

今回の実験では、文法の記述長とデータ尤度の記述長を平等に考慮した学習文法が望ましいため、以下のような非ノイズ記号の数に依存する事前重み λ_x を導入した。

$$\lambda_x = \frac{DL_{max}(\mathbf{W}|\mathbf{G}_x^a) - DL_{min}(\mathbf{W}|\mathbf{G}_x^b)}{DL_{max}(\mathbf{G}_x^c) - DL_{min}(\mathbf{G}_x^d)} \quad (14)$$

ただし、 \mathbf{G}_x は x 個の非ノイズ記号を使用する仮説文法であり、 λ_x は x 個の非ノイズ記号

表 5 上位に現れた候補文法．最適文法は太字で表示

Table 5 Top ranking candidate grammars. Optimal grammar marked in bold.

x	Non-noise Sym	λ_x	$\lambda_x DL(G)+DL(W G)$
1	E_TkScn	0.383	1279.0016
2	E_RtnScn E_TkScn	0.2897	1160.7076
3	E_RtnScn E_TkRcpt E_TkScn	0.3096	1140.0563
4	C_MvTry E_RtnScn E_TkRcpt E_TkScn	0.3847	1211.0414
5	C_MvTry C_RmMon E_RtnScn E_TkRcpt E_TkScn	0.4246	1260.2536
6	C_MvTry C_RmMon E_MvTry E_RtnScn E_TkRcpt E_TkScn	0.4859	1353.1436
7	C_AdMon C_MvTry C_RmMon E_MvTry E_RmMon E_RtnScn E_TkScn	0.5335	1523.8244
8	C_MvTry E_HndRtn E_Intrct E_MvTry E_RmMon E_RtnScn E_TkRcpt E_TkScn	0.6228	1784.4875

表 6 最小記述長で選ばれた学習文法 (3つの非ノイズ記号を使用)

Table 6 The selected optimal grammar (using 3 non-noise symbols).

S → D	(0.02)	D → L *	(1.000)
S → H	(0.16)	E → * C	(1.000)
S → G	(0.18)	F → A *	(1.000)
S → N *	(0.04)	G → C D	(1.000)
S → J	(0.13)	H → E D	(1.000)
S → Q	(0.05)	I → * B *	(1.000)
S → *	(0.02)	J → C F	(1.000)
S → N	(0.02)	K → * D	(1.000)
S → R	(0.05)	L → F B	(1.000)
S → J B	(0.02)	M → C *	(1.000)
S → M L	(0.04)	N → E A B	(1.000)
S → M A H	(0.02)	O → E *	(1.000)
S → C K	(0.04)	P → E I	(1.000)
S → C A M F	(0.02)	Q → E K	(1.000)
S → O F	(0.02)	R → E L	(1.000)
S → M	(0.02)		
S → O L	(0.02)	* → * *	(0.309)
S → O	(0.02)	* → C_AdMon	(0.153)
S → P	(0.05)	* → C_MvTry	(0.006)
S → I	(0.04)	* → C_RmMon	(0.003)
S → K	(0.04)	* → E_HndRtn	(0.080)
A → E_ RtnScn	(1.00)	* → E_Intrct	(0.275)
B → E_ TkRcpt	(1.00)	* → E_MvTry	(0.028)
C → E_ TkScn	(1.00)	* → E_RmMon	(0.147)

を用いた，すべての仮説文法の正規化係数である． $DL_{max}(G_x^c)$ は記述長を最大にする仮説文法 G_x^c を用いたときの記述長である．このような事前重みは文法の記述長とデータ尤度の記述長の分散を強制的に同じにし，文法の記述長とデータ尤度の記述長を平等に考慮した学習文法が得られる．

評価から得られた，最適文法を含む，上位の候補文法を表 5 に示す．数少ない非ノイズ記号で形成されている文法 (表 5 の $x = 2$) は簡潔である一方，データに対する表現力が低いのでデータ尤度の記述長が増加し，合計記述長が最小ではない．その反面，数多い非ノイズ記号を利用している文法 (表 5 の $x = 5$) は行動記号列を細かく表現する能力があるものの，文法は複雑であり，文法の合計記述長も最小ではない．

最適解として得られた文法 (表 5 の $x = 3$) は簡潔かつ曖昧性の低い文法であることが分かる．最適文法を表 6 に示す．学習された文法は，10 種類の記号の中から，非ノイズ記号

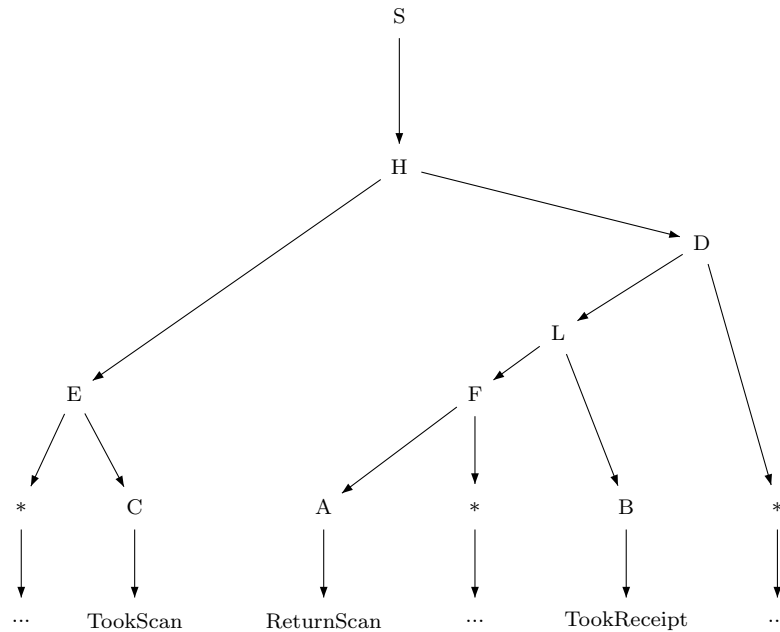


図 11 学習データに頻繁に現れた行動の構文木

Fig. 11 Parse tree of a common structure found in the training data.

E_RtnScn と E_TkRcpt と E_TkScn を用いて、数少ない記号で、取引行動を簡潔に記述している。

さらに、ここで使用されている記号は、いずれも店員の単位行動であるということにも注目されたい。取引行動を的確に表すために、最も規則性のある店員の行動が学習文法で使用されている。客の不規則な行動と比べて、規則性の強い店員の行動パターンが得られていることは理にかなっている。

学習データに頻繁に観察された取引行動 H の構文木を図 11 に示す。図 11 の構文木から「店員は取引の前半（非終端記号 E ）では何らかの行動を行い（*），スキャナを取る（ C ）。後半（非終端記号 D ）ではスキャナを返し（ A ），領収書を発行する（ B ）」という行動が、学習された文法で、説明されているのが分かる。つまり、学習された文法は、一次的な記号

の基本構造を的確に表現し、二次的な記号はノイズ記号 * として構造内で省略している。

6. 結 び

本論文では、人物行動をノイズを含む記号列として扱い、行動の基本構造を学習する枠組みを提案した。各記号の組合せに対して、その記号集合がノイズであるという仮説を立て、その仮説を最小記述長原理で評価した。最小記述長の枠組みでは、仮説文法の簡潔さと、データに対する文法の表現力の両方を評価し、仮説文法の妥当性を計った。実験では、人工データを用いて正解の生成文法が得られる条件を明確にし、さらに、実データを用いて文法の簡潔さと表現力を考慮した行動文法が得られた。

今後の研究の展開として、2つの課題がある。

1つ目は計算量の課題である。本手法では、仮説の全数検索を行ったが、記号の種類が増えるにつれ、仮説の数が指数的に増加するため、他の検索方法を検討する必要がある。たとえば、 $n-1$ の非ノイズ記号を用いる仮説から高く評価された仮説を選び、次に来る $n-2$ 個の非ノイズ記号の仮説評価は、その高く評価された仮説の部分仮説のみを評価するといった方策が考えられる。

2つ目の課題は、行動文法のモデルの選択である。文脈自由文法は、一次的なモデルと比べて、表現力が高いが、文脈自由文法には表現できない行動もある。たとえば、重なり合う行動や、同時に起こる行動は現在の枠組みでは表現することができない。次の段階として他のモデルの導入も考えなければならない。

参 考 文 献

- 1) Ivanov, Y.A. and Bobick, A.F.: Recognition of Visual Activities and Interactions by Stochastic Parsing, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.22, No.8, pp.852–872 (2000).
- 2) Kitani, K.M., Sato, Y. and Sugimoto, A.: Deleted Interpolation using a Hierarchical Bayesian Grammar Network for Recognizing Human Activity, *Proc. 2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pp.239–246 (2005).
- 3) Knight, K. and Marcu, D.: Statistics-Based Summarization — Step One: Sentence Compression, *Proc. 17th National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pp.703–710 (2000).
- 4) Minnen, D., Essa, I.A. and Starner, T.: Expectation Grammars: Leveraging High-Level Expectations for Activity Recognition, *Proc. IEEE Conference on Computer*

Vision and Pattern Recognition, pp.II: 626–632 (2003).

- 5) Moore, D.J. and Essa, I.A.: Recognizing Multitasked Activities from Video Using Stochastic Context-Free Grammar, *Proc. National Conference on Artificial Intelligence*, pp.770–776 (2002).
- 6) Nevil-Manning, C.G. and Witten, I.H.: Online and Offline Heuristics for Inferring Hierarchies of Repetitions in Sequences, *Proc. IEEE*, 88, No.11, pp.1745–1755 (2000).
- 7) Pynadath, D.V. and Wellman, M.P.: Generalized Queries on Probabilistic Context-Free Grammars, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.20, No.1, pp.65–77 (1998).
- 8) Quinlan, J.R. and Rivest, R.L.: Inferring Decision Trees Using the Minimum Description Length Principle, *Information and Computation*, Vol.80, No.3, pp.227–248 (1989).
- 9) Stolcke, A.: Bayesian Learning of Probabilistic Language Models, Ph.D. Thesis, University of California at Berkeley (1994).
- 10) Wang, T., Shum, H., Xu, Y. and Zheng, N.: Unsupervised Analysis of Human Gestures, *Proc. IEEE Pacific Rim Conference on Multimedia*, pp.174–181 (2001).
- 11) Zacks, J.M. and Tversky, B.: Event Structure in Perception and Conception, *Psychological Bulletin*, Vol.127, pp.3–21 (2001).
- 12) 三富文和, 藤原冬樹, 山本正信, 佐藤泰介: 習慣的な行動の確率文脈自由文法に基づくベイズ識別, 電子情報通信学会論文誌, Vol.J88-D2, No.4, pp.716–726 (2005).

付 録

A.1 記号列の尤度計算

ベータ確率を用いた記号列の尤度計算について簡単に紹介する(詳細は文献 7)を参照).
 まず, ベータ確率 $\beta(N, i, j, k)$ は記号 N が記号列 w_i, \dots, w_{i+j} を生成する確率(尤度)を表し, N, i, j, k をインデックスとするテーブルとして表現することができる. ここでは, 動的計画法を用いて記号列 $W = \{w_1, \dots, w_{j_{max}}\}$ ($j_{max} > 1$) と文法 G からベータ確率のテーブルを逐次的に構築する.

$$\beta(N, i, j, k) = P(N \Rightarrow w_i, \dots, w_{i+j} | G) \tag{15}$$

ただし, i は記号列の開始点, j は記号列 w_i, \dots, w_{i+j} の長さ, k は非終端記号 N の抽象度である. N をルートとする構文木の各ノードの開始点 i , 長さ j と抽象度 k を図 12 で示す. 抽象度 k は記号列 w_i, \dots, w_{i+j} を生成するために, ルート N から連続的に生成される

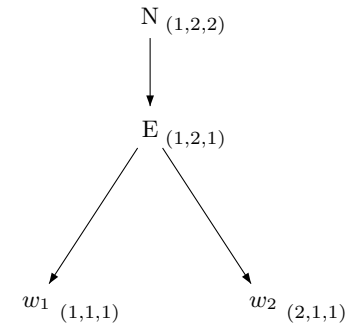


図 12 構文木のノードの指数. 指数は (i,j,k)
 Fig. 12 Indices of a parse tree. Indices are given as (i,j,k) .

抽象規則の数^{*1}であり, ここでいう抽象規則は, $N \rightarrow N'$ のような形を持つ生成規則である. たとえば図 12 では, 非終端記号 N と記号列 w_1w_2 の間には $N \rightarrow E$ という抽象規則があるので, N の抽象度 k は 2 である. 一方, E からは抽象規則が生成されていないので $k = 1$ である.

テーブル構築の初期化段階では, 記号列 W の各記号 w_i に対する $\beta(w^i, i, 1, 1)$ に初期値を与える ($1 \leq i \leq j_{max}$). 具体的に, 記号列の各記号 w_i のベータ確率 $\beta(w^i, i, 1, 1) = 1.0$ に設定し, それ以外の記号 V のベータ確率は $\beta(V, i, 1, 1) = 0$ に設定する.

次は $j = 1, k = 2$ で, 各々の i ($1 \leq i \leq j_{max}$) に対して, 文法の各々の非終端記号 N のベータ確率を以下の式で計算する.

$$\beta(N, i, 1, 2) = P(N \rightarrow w^i) \beta(w^i, i, 1, 1). \tag{16}$$

当然ながら, $N \rightarrow w^i$ のような規則を持たない非終端記号のベータ確率は 0 である.

次は $j = 2$ に固定し, $k = 1$ の場合, 各々の i ($1 \leq i \leq j_{max} - 1$) に対して, 各々の非終端記号 N のベータ確率を逐次的に以下の式で計算する.

$$\beta(N, i, 2, 1) = P(N \rightarrow N^a N^b) \beta(N^a, i, 1, 2) \beta(N^b, i + 1, 1, 2). \tag{17}$$

$k = 2$ の場合,

*1 厳密には連続的に生成される抽象規則の数足す 1.

$$\beta(N, i, 2, 2) = P(N \rightarrow N^a) \beta(N^a, i, 2, 1). \quad (18)$$

上記と同様, $N \rightarrow N^a N^b$ または $N \rightarrow N^a$ のような規則を持たない非終端記号のベータ確率は 0 である.

以後, $j = 3, \dots, j_{max}$ に対して, 各々の非終端記号のベータ確率を以下の式で計算する.
 $k = 1$ の場合,

$$\beta(N, i, j, 1) = \sum_{N \rightarrow \nu} P(N \rightarrow \nu) \sum_{\{(j_t, k_t)\}_{t=1}^m} \prod_{t=1}^m \beta(N^t, i_t, j_t, k_t). \quad (19)$$

ただし, 生成規則 $N \rightarrow \nu$ の右辺は記号列 $\nu = N^1, \dots, N^m$ であり ($1 \leq m \leq j$), $\{(j_t, k_t)\}_{t=1}^m$ は $\sum_{t=1}^m j_t = j$ を満たすようなすべての組合せの要素である. たとえば, $j = 3$ であり, $N \rightarrow N^a N^b$ の場合, 規則の右辺の記号が生成できる長さの組合せ (j_t, k_t) は $\{(1, 1), (2, 1)\}, \{(1, 1), (2, 2)\}, \dots, \{(2, 2), (1, 2)\}$ 等, 8 つの組合せがあり, 要素 $\{(1, 1), (2, 1)\}$ は $N_{j=3, k=1}$ が $N_{j=1, k=1}^a$ と $N_{j=2, k=1}^b$ を生成することを意味する.

$k = 2$ の場合,

$$\beta(N, i, j, 2) = \sum_{N \rightarrow N^a} P(N \rightarrow N^a) \beta(N^a, i, j, 1). \quad (20)$$

記号列 W の尤度は以下の式で求まる.

$$P(W_i | \mathbf{G}) = \sum_{k=1}^2 \beta(S, 1, j_{max}, k) \quad (21)$$

(平成 19 年 9 月 21 日受付)
 (平成 20 年 3 月 10 日採録)

(担当編集委員 齋藤 英雄)



木谷 クリス 真実

1999 年南カリフォルニア大学工学部電子工学専攻卒業. 2000 年ケーエルイー・テンコール株式会社入社. 2005 年東京大学大学院情報理工学系研究科電子情報学専攻修士課程修了. 現在, 同大学院同研究科同専攻博士課程在学中.



佐藤 洋一 (正会員)

1990 年東京大学工学部機械工学科卒業. 1997 年カーネギーメロン大学大学院計算機科学部ロボティクス学科博士課程修了. Ph.D. in Robotics. 同年より東京大学生産技術研究所研究機関研究員, 講師, 助教授を経て, 現在同大学大学院情報学環准教授. コンピュータビジョン, ヒューマン・コンピュータ・インタラクション, コンピュータグラフィックスに関する研究に従事. 2008 年電子情報通信学会論文賞, 2006 年電子情報通信学会論文賞, 1999 年情報処理学会山下記念研究賞, 1999 年日本バーチャルリアリティ学会論文賞等を受賞. 電子情報通信学会, 日本バーチャルリアリティ学会, ACM, IEEE 各会員.



杉本 晃宏 (正会員)

1987 年東京大学工学部計数工学科卒業. 1989 年同大学院工学系研究科修士課程修了 (数理工学専攻). 日立製作所基礎研究所, ATR, 京都大学を経て, 2002 年より国立情報学研究所. 現在, 同研究所教授. 総合研究大学院大学複合科学研究科教授併任. 2006 ~ 2007 年 Paris-Est 大学客員教授. 博士 (工学). 視覚情報処理や離散システム・アルゴリズム等に広く興味を持ち, 数理的手法に基づいた手法を確立する研究に従事. 2001 年情報処理学会論文賞. IEEE, ACM, 日本応用数理学会, 電子情報通信学会等の会員.