

GRAMMATICAL INFERENCE ON THE BASIS OF LINEAR ENVIRONMENTS OF SYMBOL STRINGS

Yoshiyuki Kotani
Department of Information Science
Tokyo University of Agriculture and Technology
Koganei 2-24-16, Tokyo 184, Japan

A new method of grammatical inference is proposed. The inference procedure, receiving only positive sample sentences, produces a grammar. A notion of "bounded context" is introduced which represents linear environment of a string in sample sentences. The procedure consists of (1) calculating the bounded context of strings, (2) finding a pair of context-equivalent strings, (3) reducing the sample sentences by means of the pair, and (4) iterating (1) to (3) until there is no pair of context-equivalent strings in the reduced sample sentences. The output grammar is constructed from the reduced sentences and the pairs of strings obtained in (2). Sample sentences are grammatically correct with respect to the grammar inferred from the sentences. All of regular languages and a number of non-trivial context-free languages are inferred correctly by the method, when sufficient but a finite number of sample sentences are given.

1. INTRODUCTION

Grammatical inference is a problem of inferring a grammar which describes a language by means of a finite set of sample sentences of the language. According to the development of formal linguistics, grammatical inference has recently received increasing attention as one of the most important information-processing problems. It serves as a tool for automatic acquisition of many sorts of grammatical descriptions which are dealt with in pattern recognition, graph theory, programming language design, information retrieval and other parts of artificial intelligence study.

We propose here a practical, systematic grammatical inference procedure. It has the following properties. (1) The procedure can be applied to languages which are impossible of description by any regular grammar. (2) The procedure does not need any structural information of sample sentences. (3) The procedure works without feedback by teachers. (4) The procedure is efficient, i.e., enumeration of grammars is not employed.

In many effective methods of grammatical inference, production rules are often inferred on the basis of some supposed relationship between similar parts of sample sentences. We use a notion of "bounded context" which represents linear environment of a symbol string in sample sentences, and define an equivalence relation of strings by equality of the bounded

context.

The next simple example explains our method. Suppose sample sentences are the strings

ab, aabb, aaabbb, ...

The two substrings 'ab' and 'aabb' appear in the same environment in the sentences, because they are put in the places denoted by $[_]$ in the following $[_]$, $a[_]b$, $aa[_]bb$, ... This means the two substrings are context-equivalent.

Grammars are constructed from production rules which consist of pairs of context-equivalent strings and from reduced sample sentences. Here the production rule $ab \rightarrow aabb$ is inferred on the basis of the context equivalence. The reduction of sample sentences is carried into effect by replacing all 'aabb' in them by 'ab' repeatedly until no 'aabb' exists. Then all of the sample sentences are reduced to a single sentence 'ab'. We call it a reduced sample sentence. The inferred grammar is described by the triplet $(\{a,b\}, \{ab \rightarrow aabb\}, \{ab\})$. This description of grammar is a little different from that of ordinary formal grammars, but obviously grammars of this type can be transformed into ordinary context-sensitive grammars.

2. FUNDAMENTAL DEFINITIONS

We establish the basic concepts necessary for the expression of our procedure. First, the preliminary notations are provided. An alphabet is denoted by Σ , and we use $a, b, c, \dots$ as symbols. Symbol strings are denoted by $u, v, \dots, z$ except the empty string λ . A set AB consists

of concatenation strings of elements of A and B. The notation A^* is the closure for A. A language is a set of strings and denoted by L, L' , etc. We call an element of a language a sentence. The length of a string x is denoted by $|x|$.

DEFINITION 2.1. The $k\ell$ -bounded context of a string w in a language L is a set of pairs of strings

$$C_L^{k\ell}(w) = \{(\text{last}_k(x), \text{first}_\ell(y)) \mid xwy \in L\},$$

where the parameters k, ℓ are non-negative integers or infinity and $\text{first}_\ell(y)$ ($\text{last}_k(y)$) is y's leftmost(rightmost) substring of length k, if $|y| < k$, otherwise w. The context equivalence $\approx_L^{k\ell}$ is defined as follows: $w \approx_L^{k\ell} w'$ if and only if $C_L^{k\ell}(w) = C_L^{k\ell}(w')$.

DEFINITION 2.2. A symbol substitution grammar is a triplet $G = (\Sigma, P, S)$, where Σ is an alphabet, P is a set of production rules which are of the form $x \rightarrow y$ and S is a finite set of strings called an initial set of sentences. In correspondence with ordinary formal grammars, we call it a type-1 ssg if x is not longer than y for any x, y in $x \rightarrow y \in P$. The language $L(G)$ generated by a grammar $G = (\Sigma, P, S)$ is the set of strings generated from S by P, denoted by $g_P(S)$. More precisely, $g_P(S)$ is defined by

$$\{u \mid v \xrightarrow{P}^* u, v \in S\},$$

where the relation of strings $v \xrightarrow{P}^* u$ means that $v = u$ or there is a sequence of strings $v_0, \dots, v_n$ such that $v = v_0 \xrightarrow{P} v_1 \xrightarrow{P} \dots \xrightarrow{P} v_n = u$ while $v \xrightarrow{P} v'$ denotes that there exist x, y, w, w' which satisfy $v = xwy$, $v' = xw'y$, $w \rightarrow w' \in P$.

Our procedure infers grammars of this type.

3. GRAMMATICAL INFERENCE PROCEDURE

Before presenting our procedure, we need some preliminaries. An order relation $\prec$ of strings is defined as follows: i) $x \prec y$, if x is shorter than y, and ii) two strings of the same length follows the lexicographic order. The j-th non-empty string in the order is denoted by $z(j)$.

Next we define an operator g_P^{-1} which is a mapping from the set of strings to itself. Intuitively, $g_P^{-1}(w)$ is the reduced sample sentence of w.

DEFINITION 3.2. Let a set of production rules P be $\{x_m \rightarrow y_m \mid m=1, \dots, j\}$. The reduced string of w, denoted by $g_P^{-1}(w)$, is the string obtained by applying the production rules in P to w in such a manner that any y_m in w is replaced by x_m repeatedly, until no y_m remains in w. The

operator can be applied to a set of strings.

Our grammatical inference procedure produces grammars by receiving as input a finite number of positive sample sentences. The system includes two arbitrary parameters: an order relation $\prec$ of symbols, and a pair (k, ℓ) of non-negative integers and/or infinities. The system outputs an ssg defined above. It is also type-1 because of the way of making production rules in the procedure. The procedure is described in the ALGOL-like form as follows:

```
S := L'; P := ∅; jj := 1;
for i := 1 step 1 until ∞ do begin
  for j := jj step 1 until ∞ do begin
    if |z(j)| < max_{w ∈ S} |w| then go to OUT;
    if C_S^{kℓ}(z(j)) ≠ ∅ then
      for m := 1 step 1 until j-1 do
        if z(j) ≈_L^{kℓ} z(m) then begin
          P := {z(m) → z(j)} ∪ P;
          S := g_P^{-1}(L'); go to NEXTSTRING
        end
      end;
    NEXTSTRING: jj := j+1
  end;
end;
out: print( (Σ, P, S) )
```

In the procedure, L' is the set of sample sentences, and S is the set of (partially) reduced sentences. The strings $z(j)$ ($j=1, 2, \dots$) are tested in order: I) If $z(j)$ is longer than any string in S, the grammar is output, since no more rules can be found. II) Otherwise it is tested for every string $z(m)$ shorter than $z(j)$ whether or not $z(j)$ is context-equivalent to $z(m)$. If it is true, the procedure adds a production rule $z(m) \rightarrow z(j)$ to P, and recalculates S reducing the sample sentences L' by the new P.

EXAMPLE 3.1. Let sample sentences be ab, aabb, aaabbb, abc, abcc, abccc, abcccc, aabbc, aabbcc. When we choose parameters (1,1) and $\prec$ such that $a \prec b \prec c$, a type-1 ssg

$$G_{1,1}^{1,1}(L') = (\{a, b, c\}, \{c \rightarrow cc, ab \rightarrow aabb\}, \{ab, abc\})$$

is inferred. This grammar describes a language

$$L(G_{1,1}^{1,1}(L')) = \{a^i b^j c^j \mid i \geq 1, j \geq 0\}.$$

Table 1 shows the process of obtaining the ssg.

4. PROPERTY OF INFERRED GRAMMARS

We describe property of grammars inferred by our procedure and possibility of correct inference. There have been found the following facts:

i) Inferred grammars are consistent with the sample sentences. Precisely, sample sentences are grammatically correct with respect to the

TABLE 1. The Process of the Procedure for Ex.3.1.

i	j	z(j)	$C_L^{i+1}(z(j))$	P; S
1	1	a	$(\lambda, a), (\lambda, b), (a, b), (a, a)$	$c \rightarrow cc$; ab, aabb, aaabbb, abc, aabbc
	2	b	$(a, \lambda), (a, b), (b, \lambda), (b, b), (a, c), (b, c)$	
	3	c	$(b, \lambda), (b, c), (c, \lambda), (c, c)$	
	4	aa	$(\lambda, b), (\lambda, a), (a, b)$	
	5	ab	$(\lambda, \lambda), (a, b), (\lambda, c)$	
	:	:	:	
2	12	cc	$(b, \lambda), (b, c), (c, \lambda), (c, c)$	$c \rightarrow cc$, ab $\rightarrow$ aabb; ab, abc
	:	:	:	
	46	aabb	$(\lambda, \lambda), (a, b), (\lambda, c)$	

grammar obtained from the sentences, that is, the language described by the grammar includes the set of sample sentences.

ii) The grammar inferred from sufficiently many sample sentences describes correctly the source language of the sample sentences. Precisely, if a source language L satisfies a condition, which we call $\approx$ -inferability, then the language is inferred correctly from any complete sequence of L 's sample sets by the use of any $k\ell$ -bounded context in which k and ℓ are sufficiently large integers. A sequence of sets of sample sentences from a language L is complete iff for any w in L there exists a positive integer t such that for all $t' > t$ $L_{t'}^1$ includes w .

iii) The class of $\approx$ -inferable languages includes all of regular languages and many non-trivial context-free languages.

When the procedure in which non-bounded context is used produces a grammar, we call the input language is $\approx$ -inferable. We also let source languages which are usually infinite sets be input, by giving an algorithm to calculate any description of bounded context.

EXAMPLE 4.1. Let us consider the following sequence of sample sets from a language $L = \{a^n b^n \mid n \geq 1\}$: $L_1 = \{ab\}$, $L_2 = \{ab, aabb\}$, ..., $L_N = \{ab, \dots, a^N b^N\}$, ... For example, by using 4,4-bounded context, we get the sequence of grammars: $G_1, G_2, \dots, G_5, \dots$, where $G_N = (\Sigma, \phi, L_N)$, ($N=1, \dots, 5$) and $G_6 = G_7 = \dots = (\Sigma, \{ab \rightarrow aabb\}, \{ab\})$. We must notice that the grammar $G_6 = G_7 = \dots$ describes the source language L correctly.

EXAMPLE 4.2. The following table makes us understand that most of inferred grammars are not so redundant and may reflect properties of their source languages. They are similar to the cfg's we may think of from the source language. These grammars can be inferred from any sample sets which includes all strings longer than a

TABLE 2. Result Grammars by $\approx$ -Bounded Context for Various Languages

Input Languages	Inferred Grammars
$\{w \mid w \text{ consists of } a, b \text{ and the number of } a \text{ is odd}\}$	$(\{a, b\}, \{b \rightarrow aa, a \rightarrow ababa, b \rightarrow bb\}, \{a, ab, ba, bab\})$
$\{w \mid w \text{ consists of } a, b \text{ and includes } aa\}$	$(\{a, b\}, \{b \rightarrow bb, aa \rightarrow aaa, b \rightarrow bab, b \rightarrow baab\}, \{aa, aab, baa\})$
$\{w \mid w \text{ consists of } a, b, (\text{the number of } a) = (\text{the number of } b)\}$	$(\{a, b\}, \{ab \rightarrow ba, ab \rightarrow aabb\}, \{ab\})$
the set of adequate sequences of parentheses	$(\{<, >\}, \{< \rightarrow < < >, < \rightarrow < > < >\}, \{< >\})$
the set of reversed Polish formulas	$(\{a, b, *, +\}, \{a \rightarrow b, + \rightarrow *, a \rightarrow aa +\}, \{a\})$
$\{wcp(w) \mid w \text{ consists of } a, b\}$ where $p(a_1 a_2 \dots a_n) = a_n \dots a_2 a_1$.	$(\{a, b, c\}, \{c \rightarrow aca, c \rightarrow bcb\}, \{c\})$
the set of arithmetic expressions without redundant parentheses described by the cfg: $S \rightarrow S+T \mid T, T \rightarrow T * F \mid F, F \rightarrow (S+T) \mid a$	$(\{a, +, *, (,)\}, \{a \rightarrow a * a, a \rightarrow (a + a), a \rightarrow a + a +\}, \{a, a + a\})$

constant length.

5. DISCUSSION

We implemented the procedure, which was written in LISP, and tested real sample data (simple sentences of natural languages and feature sequences of letter patterns). The processing of the data helped us to recognize the following ideas: i) It is important to select good parameters k, ℓ in accordance with the application. We need too many sample sentences to obtain correct grammars, if k, ℓ are very large. ii) It may be interesting to change the parameters dynamically by observing the convergence of inferred grammars. iii) If noise is expected, we had better employ 'similarity' instead of equivalence. iv) Quick parser can be constructed, which may be simpler than that of cfg. It seems able to be deterministic, or not needed backtracking.

ACKNOWLEDGMENTS

The author is grateful to Dr. S. Amari of Univ. of Tokyo for many helpful suggestions.

REFERENCES

- [1] Biermann, A.W. and J.A. Feldman, "A Survey Results in Grammatical Inference", 4th Hawaii Int. Conf. System Sciences (1971) 31-53.
- [2] Fu, K.S. and T.L. Booth, "Grammatical Inference: Introduction and Survey", IEEE Trans., vol.SMC-5,1,95-111 and 4,409-423 (1975).