(Information Processing Letters,
vol. 39, No. 4 (1991.8), pp. 177-182)

INDUCTIVE INFERENCE FROM ALL POSITIVE AND SOME NEGATIVE DATA

Tatsuya MOTOKI

September 18, 1990

Revised December 3, 1990

Revised May 16, 1991

College of General Education

Niigata University

Ikarashi 2-8050

Niigata 950-21, Japan

Keywords :

Formal languages,

computational learning,

identification in the limit,

usefulness of negative data

1. Introduction

Inductive inference is a fundamental component of intelligent behavior; so many researchers have discussed the computational aspect of induction process (see [2] for a survey). Gold [3] established a basic paradigm of inductive inference. He considers an induction process to be an infinite one that occasionally receives examples of some unknown rule, and simultaneously generates a sequence of guesses of the underlying rule; he then defines the process to make correct inference if the sequence of guesses eventually converges to a description of the underlying rule.

We are now concerned with inductive inference of (formal) languages, that is, those induction processes that (i) restrict the range of unknown rules to some fixed class of languages, (ii) receive *positive* or *negative* facts (, i.e., examples of elements in the unknown language, or examples of elements that are not in the language), and (iii) choose a guess language from the fixed class. Angluin [1] characterizes those classes of nonempty recursive languages that can be correctly infered when all and only positive facts are allowed to be eventually given to induction process. Shinohara [4, section 5.3] shows that when all positive facts are eventually given to induction process, finitely many more negative facts do not necessarily extend the possibility of correct inference.

In this paper, we consider negative facts to be the timely advice that observers give to induction process, and show that when all positive facts are eventually given to induction

process, some supplementary negative facts might extend the possibility of correct inference. Hence, as we intuitionally accept it, we might construct an induction process of practical use by supposing the existence of teachers who help the process to make correct inference.

2. Preliminaries

In this section, we give basic definitions and a result about inductive inference of formal languages. Section 2.1 deals with the case where all and only positive facts are eventually given to induction process; section 2.2 deals with the case where all positive and some negative facts are eventually given.

2.1. Inductive inference from positive data

Let Σ be a fixed nonempty finite set of symbols. We use Σ^* to denote the set of all finite strings of symbols from Σ, including the empty string ε, and Σ^+ to denote the set $\Sigma^* - \{\varepsilon\}$. A *language* is any subset of Σ^*. An *indexed family of nonempty recursive languages* is an infinite sequence $L_1, L_2, \ldots$ of nonempty languages such that the following membership function f is computable:

capital sigma (Greek)

lower-case epsilon (Greek)

$$f(i, w) = 1 \quad \text{if } w \in L_i;$$
$$= 0 \quad \text{otherwise.}$$

symbol for "is an element of"

An *inference machine* M is an effective procedure that occasionally requests an element in $\Sigma^* \times \{0, 1\}$ as an input and occasionally outputs a positive integer as a *guess*. [Note that we use the word "effective" to mean that all of the operations in the procedure are sufficiently basic that they can exactly be done in a finite length of time.] We regard an input $(s, 1)$ as a *positive fact* that s is in the unknown language, and an input $(s, 0)$ as a *negative fact* that s is not in the unknown language. To run the machine M we provide with an infinite or finite sequence $\sigma = \langle (s_1, d_1), (s_2, d_2), \ldots \rangle$ of inputs; when M requests an input for the ith time, we supply with the ith element (s_i, d_i). For each input sequence σ, we use $M[\sigma]$ to denote the infinite or finite sequence of guesses produced by M on σ; for each finite sequence σ, we use $M(\sigma)$ to denote the last guess in $M[\sigma]$, that is, the most recent guess produced by M on σ. Given an infinite sequence σ, we say $M[\sigma]$ *converges* to a positive integer i if either $M[\sigma]$ is finite and the last guess is equal to i, or $M[\sigma]$ is infinite and all but finitely many terms in $M[\sigma]$ are equal to i.

A *positive presentation* of a nonempty language L is an infinite sequence $\langle (s_1,1), (s_2,1), \ldots \rangle$ such that $\{s_1, s_2, \ldots\} = L$. An inference machine M is said to *infer an indexed family of nonempty recursive languages* $L_1, L_2, \ldots$ *from positive data* if for every $i \geq 1$ and every positive presentation σ of L_i, the sequence $M[\sigma]$ converges to some positive integer j such that $L_j = L_i$. An indexed family of nonempty recursive languages $L_1, L_2, \ldots$ is said to be *inferable from positive data* if there exists an inference machine that infers the family $L_1, L_2, \ldots$ from

positive data.

The following theorem characterizes when an indexed family of nonempty recursive languages is inferable from positive data.

Theorem 2.1 [1]. *An indexed family of nonempty recursive languages L_1, L_2, ... is inferable from positive data if and only if there exists an effective procedure that, given an input $i \geq 1$, enumerates (,i.e., generates all and only elements of) a finite subset T_i of L_i such that for every $j \geq 1$ the condition $T_i \subseteq L_j \subseteq L_i$ implies $L_j = L_i$.*

symbol for "is contained in"

2.2. Inductive inference from all positive and some negative data

Let L_1, L_2, ... be an indexed family of nonempty recursive languages. A (*negative*) *advisor* is a function $A : \{1, 2, 3, \ldots\} \longrightarrow 2^{\Sigma^*}$ such that $A(i) \subseteq \bar{L}_i$ for every i. For each $i \geq 1$, we regard $A(i) \times \{0\}$ as a set of supplementary negative facts that observers will eventually give to the inference machine when the unknown language is L_i.

arrow sign

For each advisor A and each $i \geq 1$, an *A-positive presentation* of L_i is an infinite sequence $\langle (s_1, d_1), (s_2, d_2), \ldots \rangle$ of ordered pairs from $\Sigma^* \times \{0, 1\}$ such that $\{s \mid (s, 1) = (s_k, d_k)$ for some $k\} = L_i$ and $A(i) \subseteq \{s \mid (s, 0) = (s_k, d_k)$ for some $k\} \subseteq \bar{L}_i$. An inference machine M is said to *infer an indexed family of nonempty recursive languages L_1, L_2, ... from positive data and an advisor A* if for every $i \geq 1$ and every A-positive presentation σ of L_i, the sequence $M[\sigma]$ converges to some positive integer j such that $L_j = L_i$. An indexed family of nonempty recursive

languages L_1, L_2, ... is said to be *inferable from positive data and an advisor A* if there exists an inference machine that infers the family L_1, L_2, ... from positive data and an advisor A.

3. Usefulness of negative data

We now establish a result that is similar to Theorem 2.1.

Theorem 3.1. *An indexed family of nonempty recursive languages L_1, L_2, ... is inferable from positive data and an advisor A if and only if there exists an effective procedure that, given an input $i \geq 1$, enumerates a finite subset T_i of L_i such that for every $j \geq 1$ the conjunction of $T_i \subseteq L_j \subseteq L_i$ and $A(j) \subseteq \bar{L}_i$ implies $L_j = L_i$.*

Proof. The proof proceeds in the similar manner as that of Theorem 2.1.

($\Leftarrow$) Suppose that there exists an enumeration procedure of T_i as specified in the theorem, and consider the inference machine M described as follows:

arrow sign

```
for n:=1 to +∞ do
  begin
    request another input data (s_n, d_n);
    compute and output the value of the expression
      min[{g | g≤n, T_g^(n) ⊆ P(n) ⊆ L_g, N(n) ⊆ L̄_g} ∪ {n+1}]
  end,
```

symbol for infinity

symbol for the union operation of sets

where we define $P(n) = \{s \mid (s,1)=(s_k,d_k)$ for some $k\leq n\}$, define $N(n) = \{s \mid (s,0)=(s_k,d_k)$ for some $k\leq n\}$, and use $T_g^{(n)}$ to denote the set of strings produced in the first n steps of the enumeration of T_g. [Note that the effectiveness of M follows from the recursiveness of L_i and the existence of enumeration procedure of T_i.]

To see that M infers L_1, L_2, ... from positive data and an advisor A, let $j\geq 1$ be arbitrary and assume that $\sigma = <(s_1, d_1), (s_2, d_2), \ldots >$ is any A-positive presentation of L_j. It remains to show that $M[\sigma]$ converges to some i such that $L_j=L_i$. For each $n\geq 1$, we use $\sigma(n)$ to denote the initial subsequence $<(s_1, d_1), \ldots, (s_n, d_n)>$.

Since T_j is a finite subset of L_j, we have $T_j \subseteq P(n)$ for sufficiently large n; so define $n_0=\min\{n \mid T_j \subseteq P(n)\}$. Then we can easily prove two claims: (i) $M(\sigma(n))\leq j$ for every $n\geq n_0$, and (ii) $M(\sigma(n))\leq M(\sigma(m))$ for every $m\geq n\geq n_0$. From these claims we see that $M[\sigma]$ converges to some positive integer, say i. Thus there must exists an integer n^* such that for every $n\geq n^*$,

$$T_i^{(n)} \subseteq P(n) \subseteq L_i \quad \text{and}$$
$$N(n) \subseteq \bar{L}_i;$$

so by letting $n \longrightarrow \infty$ two relations

$$T_i \subseteq \cup_{n=1}^{\infty} P(n) \subseteq L_i \quad \text{and} \qquad (1)$$
$$\cup_{n=1}^{\infty} N(n) \subseteq \bar{L}_i \qquad (2)$$

follow. Since we obtain $\cup_{n=1}^{\infty} P(n)=L_j$ and $A(j) \subseteq \cup_{n=1}^{\infty} N(n)$ from

the choice of σ and the definitions of $P(n)$ and $N(n)$, we now derive from (1), (2) two relations $T_i \subseteq L_j \subseteq L_i$ and $A(j) \subseteq \bar{L}_i$; so from the choice of T_i we deduce the required equation $L_j = L_i$.

($\Longrightarrow$) Suppose M is an inference machine that infers L_1, L_2, ... from positive data and an advisor A. For each $i \geq 1$, let $(s_{i,1}, d_{i,1})$, $(s_{i,2}, d_{i,2})$, ... be an enumeration of $L_i \times \{1\} \cup \bar{L}_i \times \{0\}$, and let $\rho_{i,1}$, $\rho_{i,2}$, ... be an enumeration of all finite sequences of elements from $L_i \times \{1\} \cup \bar{L}_i \times \{0\}$. [Note that effective enumeration procedures for $(s_{i,m}, d_{i,m})$'s and $\rho_{i,m}$'s are seen to exist from the existence of effective enumeration procedure of Σ^* and the recursiveness of L_i.]

To enumerate for any given $i \geq 1$ the required finite subset T_i of L_i, consider the infinite procedure $P(i)$ described as follows:

```
τ := <>;
T := {};
for k:=1 to +∞ do
  begin
    m := 0;
    repeat m:=m+1 until M(append(τ, ρ_{i,m}))≠M(τ);
    τ := append(τ, append(ρ_{i,m}, <(s_{i,k}, d_{i,k})>));
    T := T ∪{s | (s,1) is in append(ρ_{i,m}, <(s_{i,k},d_{i,k})>)}
  end,
```

where we use <> to denote the null sequence, and append(τ, ρ) to denote the sequence obtained by appending sequence ρ to the end

of finite sequence τ. Now let T_i be the limiting value of T in $P(i)$; similarly let $\tau(i)$ be the limiting value of τ in $P(i)$. We easily see $T_i \subseteq L_i$, and can regard $P(i)$ as an effective procedure to enumerate T_i. [Note that the **repeat** loop in $P(i)$ tries to find an integer m such that $M(\text{append}(\tau, \sigma_{i,m})) \neq M(\tau)$, and that the effectiveness of $P(i)$ follows from the existence of effective enumeration procedures for $(s_{i,m}, d_{i,m})$'s and $\sigma_{i,m}$'s.]

To show the finiteness of T_i, let us assume the opposite and see what happen. Since each $\sigma_{i,m}$ is finite, the execution of $P(i)$ goes infinitely many times around the **for** loop; so $\tau(i)$ is an A-positive presentation of L_i. However, we can also derive the consequence that the execution of M on $\tau(i)$ changes its guess infinitely many times and so $M[\tau(i)]$ does not converge to any integer. Therefore M fails to make correct inference when we provide with an input sequence $\tau(i)$, a contradiction.

The argument in the preceding paragraph also shows that the execution of $P(i)$ cannot go infinitely many times around the **for** loop. Hence after the execution of $P(i)$ goes some finite times around the **for** loop, the procedure $P(i)$ cannot find an integer m such that $M(\text{append}(\tau, \sigma_{i,m})) \neq M(\tau)$; after that time, program variables τ and T have already converged to $\tau(i)$ and T_i respectively. Therefore $\tau(i)$ satisfies the condition

$$M(\text{append}(\tau(i), \sigma_{i,m})) = M(\tau(i)) \text{ for every } m \geq 1. \quad (3)$$

Finally, let $j \geq 1$ be arbitrary and assume that two conditions

$T_i \subseteq L_j \subseteq L_i$ and (4)

$A(j) \subseteq \bar{L}_i$ (5)

hold. Now it remains to deduce the equation $L_i=L_j$. To do this, consider two input sequences

$$\sigma_1=\text{append}(\tau(i), <(s_{i,1}, d_{i,1}), (s_{i,2}, d_{i,2}), \ldots >) \text{ and}$$
$$\sigma_2=\text{append}(\tau(i), <(t_1, e_1), (t_2, e_2), \ldots >),$$

where $(t_1, e_1), (t_2, e_2), \ldots$ denote an enumeration of $L_j \times \{1\} \cup \bar{L}_i \times \{0\}$. Obviously σ_1 is an A-positive presentation of L_i; from (4), (5) and the equation $\{s \mid (s, 1) \text{ is in } \tau(i)\}=T_i$, the sequence σ_2 is seen to be an A-positive presentation of L_j. So $M[\sigma_1]$ converges to some integer i^* such that $L_{i^*}=L_i$; similarly $M[\sigma_2]$ converges to some integer j^* such that $L_{j^*}=L_j$.

Now, observe that for each $p \geq 1$ two equations

$$M(\text{append}(\tau(i), <(s_{i,1}, d_{i,1}), \ldots, (s_{i,p}, d_{i,p})>)) = M(\tau(i)) \text{ and}$$
$$M(\text{append}(\tau(i), <(t_1, e_1), \ldots, (t_p, e_p)>)) = M(\tau(i))$$

follows from (3). From these equations, we deduce that both $M[\sigma_1]$ and $M[\sigma_2]$ must converge to the integer $M(\tau(i))$, that is, $i^*=j^*=M(\tau(i))$; and hence we obtain the required equation $L_i=L_{i^*}=L_{j^*}=L_j$. □

square symbol

We next give an example of the application of Theorem 3.1.

Example 3.2. Let $\Sigma=\{a, b\}$, and let h be a computable bijection from $\{1, 2, 3, \ldots\}^2$ to $\{1, 2, 3, \ldots\}$. Now consider the family $L_1, L_2, \ldots$ and an advisor A defined as follows:

$$
\begin{aligned}
L_{h(i,j)} &= \{a\}^+ \cup \{b\}^+ && \text{if } i=j=1;\\
&= \{a\}^+ \cup \{b, b^2, \ldots, b^{j-1}\} && \text{if } i=1,\ j>1;\\
&= \{a, a^2, \ldots, a^{i-1}\} \cup \{b\}^+ && \text{if } i>1,\ j=1;\\
&= \{a, a^2, \ldots, a^{i-1}\} \cup \{b, b^2, \ldots, b^{j-1}\} && \text{if } i>1, j>1;
\end{aligned}
$$

$$
\begin{aligned}
A(h(i,j)) &= \{\} && \text{if } i=j=1;\\
&= \{b^j\} && \text{if either } 1<i\leq j \text{ or } 1=i<j;\\
&= \{a^i\} && \text{if either } 1<j<i \text{ or } 1=j<i;
\end{aligned}
$$

Then by setting the "telltale set"

$$
\begin{aligned}
T_{h(m,n)} &= \{\} && \text{if } m=n=1;\\
&= \{a^n\} && \text{if } m=1,\ n>1;\\
&= \{b^{m-1}\} && \text{if } m>1,\ n=1;\\
&= L_{h(m,n)} && \text{if } m>1,\ n>1;
\end{aligned}
$$

and applying Theorem 3.1, we derive the consequence that L_1, L_2, ... is inferable from positive data and an advisor A. [Therefore the class L_1, L_2, ... is considered to be inferable from positive data and one supplementary negative data.] In fact, let

$$
C(h(m,n)) = \{L_{h(i,j)} \mid T_{h(m,n)} \subseteq L_{h(i,j)} \subsetneq L_{h(m,n)},\ A(h(i,j)) \subseteq \bar{L}_{h(m,n)}\}.
$$

symbol for "is strictly contained in"

Then for each $m,n>1$ we have

$$
\begin{aligned}
&C(h(1,1)) \\
&= \{L_{h(i,j)} \mid (i,j)\neq(1,1),\ A(h(i,j)) \subseteq \overline{\{a\}^{+} \cup \{b\}^{+}}\} \\
&= \{\}
\end{aligned}
$$

$$
\begin{aligned}
&C(h(m,1)) \\
&= \{L_{h(i,j)} \mid b^{m-1} \in L_{h(i,j)} \subsetneq L_{h(m,1)},\ A(h(i,j)) \subseteq \bar{L}_{h(m,1)}\} \\
&= \{L_{h(i,j)} \mid (m\leq j \text{ or } j=1),\ 1<i\leq m,\ (i,j)\neq(m,1), \\
&\qquad\qquad A(h(i,j)) \subseteq \bar{L}_{h(m,1)}\} \\
&= \{L_{h(i,j)} \mid \text{either } (1<i\leq m\leq j,\ b^{j} \in \bar{L}_{h(m,1)}) \\
&\qquad\qquad \text{or } (j=1,\ 1<i<m,\ a^{i} \in \bar{L}_{h(m,1)})\} \\
&= \{\}
\end{aligned}
$$

$$
\begin{aligned}
&C(h(1,n)) \\
&= \{L_{h(i,j)} \mid a^{n} \in L_{h(i,j)} \subsetneq L_{h(1,n)},\ A(h(i,j)) \subseteq \bar{L}_{h(1,n)}\} \\
&= \{L_{h(i,j)} \mid (n<i \text{ or } i=1),\ 1<j\leq n,\ (i,j)\neq(1,n), \\
&\qquad\qquad A(h(i,j)) \subseteq \bar{L}_{h(1,n)}\} \\
&= \{L_{h(i,j)} \mid \text{either } (1<j\leq n<i,\ a^{i} \in \bar{L}_{h(1,n)}) \\
&\qquad\qquad \text{or } (i=1,\ 1<j<n,\ b^{j} \in \bar{L}_{h(1,n)})\} \\
&= \{\}
\end{aligned}
$$

$$
\begin{aligned}
&C(h(m,n)) \\
&= \{L_{h(i,j)} \mid L_{h(m,n)} \subseteq L_{h(i,j)} \subsetneq L_{h(m,n)}, \\
&\qquad\qquad A(h(i,j)) \subseteq \bar{L}_{h(m,n)}\}. \\
&= \{\}
\end{aligned}
$$

From Theorem 3.1 and Example 3.2, we have the conclusion

that when all positive data are eventually given to inference machine, some negative data might extend the possibility of correct inference; by receiving some negative advice from teachers, machine can develop its ability to make correct inference. Thus we might construct an inference machine of practical use by supposing the existence of teachers who help the machine to make correct inference.

Acknowledgment

The author would like to thank Dr. T. Shinohara and the referee for their valuable comments, and Prof. T. Nishizawa for drawing author's attention to this research area.

References

[1] D. Angluin, Inductive inference of formal languages from positive data, *Inform. and Control* **45** (1980) 117-135.

[2] D. Angluin and C. H. Smith, Inductive inference: theory and methods, *Comput. Survey* **15** (1983) 237-269.

[3] E. M. Gold, Language identification in the limit, *Inform. and Control* **10** (1967) 447-474.

[4] T. Shinohara, Studies on inductive inference from positive data, doctoral thesis, Kyushu University, Japan (1986).