Characterization of Finite Identification

Yasuhito Mukouchi

Department of Information Systems,
Kyushu University 39, Kasuga 816, Japan

Abstract. A majority of studies on inductive inference of formal languages and models of logic programming have mainly used Gold's identification in the limit as a correct inference criterion. In this criterion, we can not decide in general whether the inference terminates or not, and the results of the inference necessarily involve some risks. In this paper, we deal with finite identification for a class of recursive languages. The inference machine produces a unique guess just once when it is convinced the termination of the inference, and the results do not involve any risks at all. We present necessary and sufficient conditions for a class of recursive languages to be finitely identifiable from positive or complete data. We also present some classes of recursive languages that are finitely identifiable from positive or complete data.

1 Introduction

Inductive inference is a process of hypothesizing a general rule from examples. As a correct inference criterion for inductive inference of formal languages and models of logic programming, we have mainly used Gold's identification in the limit[5]. An inference machine M is said to identify a language L in the limit if the sequence of guesses from M, which is successively fed a sequence of examples of L, converges to a correct expression τ of L, that is, all guesses from M become a unique τ within a certain finite time. Under this criterion, many productive results concerning inductive inference from positive data have been reported by Angluin[1], Wright[16], Shinohara[15] and Sato&Umayahara[13]. Also, many systems concerning inductive inference from complete data have been developed (cf. e.g. Shapiro[14] and Muggleton&Buntine[10]).

Considering ordinary learning process of human beings, the criterion of identification in the limit seems to be natural. However, we can not decide in general whether a sequence of guesses from an inference machine converges or not at a certain time, and the results of the inference necessarily involve some risks. Clearly, it is important to have a conclusive answer, when we want to use the result of machine learning. There are some classes of which concepts can be learned conclusively within a finite time.

In this paper, we deal with finite identification for a class of recursive languages. Originally, finite identification was introduced to inductive inference of recursive functions (cf. Freivald&Wiehagen[4], Klette&Wiehagen[7] and Jantke&Beick[6]). An inference machine M is said to finitely identify a language L if M, which is successively fed a sequence of examples of L, produces a unique guess at a certain time

and the guess is a correct expression of L. That is, the inference machine does not produce a guess until it is convinced that the guess is correct.

In Section 2, we prepare some necessary concepts for our discussions. In Section 3 and 4, we discuss necessary and sufficient conditions for a class to be finitely identifiable from positive or complete data. Angluin[1] introduced the notion of a finite tell-tale of a language to discuss inferability of formal languages from positive data, and showed that a class is inferable from positive data if and only if there is a recursive procedure to enumerate all elements in the finite tell-tale of any language of the class. In this paper, we introduce a definite finite tell-tale and a pair of definite finite tell-tales of a language, and show that a class is finitely identifiable from positive or complete data if and only if they are uniformly computable for any language of the class. We also present some classes of recursive languages that are finitely identifiable from positive or complete data.

2 Preliminaries

Let U be a recursively enumerable set to which we refer as a *universal set*. Then we call $L \subseteq U$ a *language*. We do not consider the empty language in this paper.

Definition 1. A class of languages $C = L_1, L_2, \cdots$ is said to be an *indexed family of recursive languages* if there exists a computable function $f : N \times U \to \{0, 1\}$ such that

$$f(i, w) = \begin{cases} 1, & \text{if } w \in L_i, \\ 0, & \text{if } w \notin L_i. \end{cases}$$

From now on, we assume a class of languages is an indexed family of recursive languages without any notice.

Definition 2. A *positive presentation* of a language L is an infinite sequence $w_1, w_2, \cdots$ of elements of U such that $\{w_1, w_2, \cdots\} = L$.

A *complete presentation* of a language L is an infinite sequence $(w_1, t_1), (w_2, t_2), \cdots$ of elements of $U \times \{0, 1\}$ such that $\{w_i \mid t_i = 1, i \geq 1\} = L$ and $\{w_j \mid t_j = 0, j \geq 1\} = U - L$.

We denote by σ, δ positive or complete presentations and by $\sigma[n]$ (resp., $\sigma(n)$) the finite sequence (resp., the finite set) which consists of first $n \geq 0$ data in σ.

In this paper, we use a slightly different inference machine from that of identification in the limit. That is, the inference machine is an effective procedure that requests inputs from time to time and stops with a unique output. The unique output produced by the machine is called a *guess*.

Definition 3. A class C of languages is said to be *finitely identifiable from positive (resp., complete) data* if there exists an inference machine M which satisfies the following: For any language L_i of the class C and for any positive (resp., complete) presentation σ of L_i, M which is successively fed σ's data produces a unique guess, say j, after some finite time and $L_j = L_i$ holds.

In this criterion, an inference machine produces a unique guess when the inference process terminates.

3 Finite Identification from Positive Data

In this section, we discuss necessary and sufficient conditions for a class to be finitely identifiable from positive data.

From now on, let $C = L_1, L_2, \cdots$ be an indexed family of recursive languages.

In the criterion of identification in the limit, the following definition and theorem are well-known.

Definition 4 (Angluin[1]). A set S_i is said to be a *finite tell-tale of* L_i if
(1) S_i is a finite subset of L_i, and
(2) there is no index j such that $S_i \subseteq L_j \subsetneq L_i$.

Theorem 5 (Angluin[1]). *A class C is inferable in the limit from positive data if and only if there exists an effective procedure that enumerates all elements in a finite tell-tale of L_i for any index i.*

Now, we show our definition and theorem that form a remarkable contrast to the above definition and theorem.

Definition 6. A set S_i is said to be a *definite finite tell-tale of* L_i if
(1) S_i is a finite subset of L_i, and
(2') $S_i \subseteq L_j$ implies $L_i = L_j$ for any index j.

Clearly from the definition, the definite finite tell-tale has a more specific meaning than the finite tell-tale.

In this paper, a finite-set-valued function F is said to be *computable* if there exists an effective procedure that produces all elements in $F(x)$ and then halts uniformly for any argument x.

Theorem 7. *A class C is finitely identifiable from positive data if and only if a definite finite tell-tale of L_i is uniformly computable for any index i, that is, there exists an effective procedure that on input i produces all elements of a definite finite tell-tale of L_i and then halts.*

Proof. (i) The 'only if' part. Suppose the class C is finitely identifiable from positive data. Then there exists an inference machine M which satisfies Definition 3. A definite finite tell-tale of L_i is uniformly computable by the following procedure:

Procedure $Q(i)$;
begin
  let σ be a positive presentation of L_i;
  for $k := 1$ **to** ∞ **do begin**
    feed the next datum in σ to M;
    if M produces a guess **then** output $\sigma(k)$ and stop
  end
end.

Note that we can effectively take a positive presentation of L_i, because the universal set U is effectively enumerable and whether $w \in L_i$ or not is decidable for any $w \in U$.

Since M finitely identifies the class C, this procedure is guaranteed to terminate. Now, we show by contradiction that the output of this procedure, say S, is a definite finite tell-tale of L_i. Suppose S is not a definite finite tell-tale of L_i. Clearly, S is a finite subset of L_i. Therefore, there exists an index j such that $L_i \neq L_j$ and $S \subseteq L_j$. Since M infers L_i from $\sigma[\sharp S]$, it follows that M can not infer L_j from a positive presentation δ of L_j such that $\delta[\sharp S] = \sigma[\sharp S]$. This contradicts the assumption.

(ii) The 'if' part. Suppose a definite finite tell-tale of L_i is uniformly computable for any index i, and we denote by $S(i)$ the result of computation. The class C is finitely identifiable from positive data by the following procedure:

Procedure M;
begin
  $T := \phi$;
  for $j := 1$ **to** ∞ **do begin**
    read the next datum and add it to T;
    for $i := 1$ **to** j **do**
      if $S(i) \subseteq T$ **then** output i and stop
  end
end.

Note that whether $S(i) \subseteq T$ or not is decidable, because $S(i)$ and T are explicitly given finite sets. Suppose we are going to feed a positive presentation σ of L_h.

(1) When this procedure terminates, the output is a correct guess. In fact, let g be the output of this procedure. Since $S(g) \subseteq T \subseteq L_h$, it follows that $L_g = L_h$ by Definition 6.

(2) This procedure always terminates after some finite time. In fact, let

$$a = \min\{k \mid S(h) \subseteq \sigma(k)\} \qquad \text{and} \qquad b = \max\{a, h\}.$$

Note that $h \leq b$ holds. Suppose this procedure does not terminate. Then it reaches the case of $j = b$ and $i = h$. In this case, $S(i) \subseteq T(= \sigma(b))$ holds, which contradicts the assumption. □

We can show that the above procedure M terminates with a guess c when it reaches the case $j = b$ and $i = c$, where

$$\begin{aligned} a_n &= \min\{k \mid S(m_n) \subseteq \sigma(k)\}, \qquad (n \geq 1) \\ b &= \min\{\max\{m_1, a_1\}, \max\{m_2, a_2\}, \cdots\}, \\ c &= \min\{m_k \mid \max\{m_k, a_k\} = b,\ k \geq 1\} \end{aligned}$$

and $m_1, m_2, \cdots$ are all the m's with $L_m = L_h$. Note that $L_i = L_j$ does not imply $S(i) = S(j)$.

Lange&Zeugmann[9] has obtained similar results to the above theorem in the context of monotonic language learning from positive data, independently of ours.

The following corollary is obvious from Definition 6 and Theorem 7.

Corollary 8. *If a class C has two languages L_i, L_j with $L_i \subsetneq L_j$, then the class C is not finitely identifiable from positive data.*

Here, we present an example of a class of languages which is finitely identifiable from positive data.

Example 1. Let p_i be the i-th prime and put $L_i = \{n \mid n \text{ is a multiple of } p_i\}$ $(i \geq 1)$. Since p_i is a primitive recursive function of i, the class $C = L_1, L_2, \cdots$ is an indexed family of recursive languages. This class C is finitely identifiable from positive data. In fact, we can take the set $\{p_i\}$ as a definite finite tell-tale of L_i.

4 Finite Identification from Complete Data

In this section, we discuss necessary and sufficient conditions for a class to be finitely identifiable from complete data.

The following Definition 9 and Theorem 10 form a remarkable contrast to Definition 6 and Theorem 7 concerning positive data.

Definition 9. A language L is said to be consistent with a pair of sets $\langle T, F\rangle$ if $T \subseteq L$ and $F \subseteq U - L$. A pair of sets $\langle T_i, F_i\rangle$ is said to be *a pair of definite finite tell-tales of* L_i if
(1) T_i is a finite subset of L_i,
(2) F_i is a finite subset of $U - L_i$, and
(3) if L_j is consistent with the pair $\langle T_i, F_i\rangle$, then $L_i = L_j$.

Note that if S_i is a definite finite tell-tale of L_i, then the pair $\langle S_i, \phi\rangle$ is a pair of definite finite tell-tales of L_i.

Theorem 10. *A class C is finitely identifiable from complete data if and only if a pair of definite finite tell-tales of L_i is uniformly computable for any index i.*

Proof. (i) The 'only if' part. Suppose the class C is finitely identifiable from complete data. Then there exists an inference machine M which satisfies Definition 3. Then we consider the following procedure:

Procedure $P(i)$;
begin
  let σ be a complete presentation of L_i;
  for $k := 1$ **to** ∞ **do begin**
    feed the next datum in σ to M;
    if M produces a guess **then begin**
      $T := \{w_j \mid (w_j, 1) \in \sigma(k),\ j \geq 1\}$;
      $F := \{w_j \mid (w_j, 0) \in \sigma(k),\ j \geq 1\}$;
      output the pair $\langle T, F\rangle$ and stop
    end
  end
end.

The output of the above procedure $P(i)$ is shown to be a pair of definite finite tell-tales of L_i in a similar way to the proof of Theorem 7.

(ii) The 'if' part. Suppose a pair of definite finite tell-tales of L_i is uniformly computable for any index i, and we denote by $\langle T(i), F(i)\rangle$ the result of computation. Then we consider the following procedure:

Procedure M;
begin
  $T := \phi$; $F := \phi$;
  for $j := 1$ **to** ∞ **do begin**
    read the next datum (w, v);
    if $v = 1$ **then** $T := T \cup \{w\}$ **else** $F := F \cup \{w\}$;
    for $i := 1$ **to** j **do**
      if $T(i) \subseteq T$ and $F(i) \subseteq F$ **then** output i and stop
  end
end.

We can show that the class C is finitely identifiable by the above procedure M in a similar way to the proof of Theorem 7. □

We present a sufficient condition for a class to be finitely identifiable from complete data. This condition has more specific meaning than the condition of "finite thickness", which Angluin[1] introduced as a sufficient condition for a class to be inferable in the limit from positive data.

Theorem 11. *A class C is finitely identifiable from complete data if*
(1) the set $\{i \mid w \in L_i\}$ is finite and uniformly computable for any $w \in U$, and
(2) whether $L_i = L_j$ or not is decidable for any indices i, j.

Proof. Suppose (1) and (2) hold. The definite finite tell-tale of L_i is uniformly computable by the following procedure, where the sequence $w_1, w_2, \cdots$ is an effective enumeration of the universal set U:

Procedure $P(i)$;
begin
  let k be the least number such that $w_k \in L_i$;
  $T := \{w_k\}$; $F := \phi$;
  compute the set $\{j \mid w_k \in L_j\}$ and set it to S;
  for each $j \in S$ **do**
    if $L_i \neq L_j$ **then begin**
      $m := 1$;
      while ($w_m \in L_i$ and $w_m \in L_j$) or ($w_m \notin L_i$ and $w_m \notin L_j$) **do** $m := m+1$;
      if $w_m \in L_i$ and $w_m \notin L_j$ **then** $T := T \cup \{w_m\}$ **else** $F := F \cup \{w_m\}$
    end;
  output the pair $\langle T, F\rangle$ and stop
end.

Since the while loop above is executed only when $L_i \neq L_j$, this while statement always terminates. Therefore, the procedure $P(i)$ always terminates. It is clear that the output of $P(i)$ is a pair of definite finite tell-tales of L_i. □

We present an example of a class of languages which is finitely identifiable from complete data.

Example 2. We consider the class of pattern languages. Here, we define a pattern and a pattern language briefly. (For more details, see Angluin[2] or Mukouchi[11].)

Fix a finite alphabet with at least two constant symbols. A pattern is a nonnull finite string of constant and variable symbols. The pattern language $L(\pi)$ generated by a pattern π is the set of all strings obtained by substituting nonnull strings of constant symbols for the variables of π. Since two patterns that are identical except for renaming of variables generate the same pattern language, we do not distinguish one from the other. We can enumerate all patterns recursively and whether $w \in L(\pi)$ or not for any w and π is effectively decidable. Therefore, we can consider the class of pattern languages as an indexed family of recursive languages, where the pattern itself is considered to be an index.

(i) The class of pattern languages satisfies the condition (1) of Theorem 11. In fact, fix an arbitrary constant string w. If $w \in L(\pi)$, then π is not longer than w. The set of all patterns shorter than a fixed length is finite and uniformly computable, and whether $w \in L(\pi)$ or not for any w and π is decidable. Therefore, the set $\{\pi \mid w \in L(\pi)\}$ is finite and uniformly computable.

(ii) Angluin[2] showed that $L(\pi) = L(\tau)$ if and only if $\pi = \tau$.

Therefore, we see that the class of pattern languages is finitely identifiable from complete data by Theorem 11.

By theorems in Angluin[2], we can also show that $\langle T, F \rangle$ is a pair of definite finite tell-tales of $L(\pi)$, where T is the set of all elements of $L(\pi)$ with the same length as π, and F is the set of all constant strings each of which is not longer than π and does not belong to T. Furthermore, we see that the class of pattern languages is not finitely identifiable from positive data by Corollary 8.

Note that Lange&Zeugmann[8] has obtained similar results concerning the class of pattern languages, independently of ours.

5 Concluding Remarks

In this paper, we have discussed conditions for a class of recursive languages to be finitely identifiable from positive or complete data. We also presented some classes that are finitely identifiable from positive or complete data.

Finitely identifiable classes are much smaller than those that are inferable in the limit, but the finite identification seems to be much more significant than it is thought of.

We conclude by pointing out some relations between the results on finite identification obtained in this paper and the results shown in Angluin[1]. As easily seen, the definite finite tell-tale has a more specific meaning than the finite tell-tale. Also, if a class C is finitely identifiable from positive data, then C is also inferable in the limit from positive data. It seems that whether finitely many "mind changes" are allowed or not makes the difference between recursive enumerability of a finite tell-tale and uniform computability of a definite finite tell-tale.

Acknowledgements

The author wishes to thank Setsuo Arikawa for many suggestions, that made me start this research, and productive discussions.

References

1. Angluin, D.: Inductive inference of formal languages from positive data, Information and Control **45** (1980), 117–135
2. Angluin, D.: Finding patterns common to a set of strings, Proc. 11th Annual Symposium on Theory of Computing (1979), 130–141
3. Angluin, D., Smith, C.H.: Inductive inference: theory and methods, ACM Computing Surveys **15** No. 3 (1983), 237–269
4. Freivald R.V., Wiehagen, R.: Inductive inference with additional information, Elektron. Informationsverarb. Kybern. (EIK) **15** (1979), 179–185
5. Gold, E.M.: Language identification in the limit, Information and Control, **10** (1967), 447–474
6. Jantke, K.P., Beick, H.-R.: Combining postulates of naturalness in inductive inference, Elektron. Informationsverarb. Kybern. (EIK) **17** (1981), 465–484
7. Klette, R., Wiehagen, R.: Research in the theory of inductive inference by GDR mathematicians – a survey, Information Sciences **22** (1980), 149–169
8. Lange, S., Zeugmann, T.: On the power of monotonic language learning, GOSLER-Report 05/92, Fachbereich Mathematik und Informatik, TH Leipzig (1992)
9. Lange, S., Zeugmann, T.: Types of monotonic language learning and their characterization, to appear in Proc. 5th Workshop on Comput. Learning Theory (1992)
10. Muggleton, S., Buntine, W.: Machine invention of first-order predicates by inverting resolution, Proc. 5th International Conference on Machine Learning (1988), 339–352
11. Mukouchi, Y.: Characterization of pattern languages, Proc. 2nd Workshop on Algorithmic Learning Theory (1991), 93–104
12. Mukouchi, Y.: Definite inductive inference as a successful identification criterion, RIFIS-TR-CS-52, Research Institute of Fundamental Information Science, Kyushu University, (1991)
13. Sato, M., Umayahara, K.: Inductive inferability for formal languages from positive data, Proc. 2nd Workshop on Algorithmic Learning Theory (1991), 84–92
14. Shapiro, E.Y.: Inductive inference of theories from facts, Technical Report 192, Department of Computer Science, Yale University, (1981)
15. Shinohara, T.: Inductive inference from positive data is powerful, Proc. 3rd Workshop on Comput. Learning Theory (1990), 97–110
16. Wright, K.: Identification of unions of languages drawn from an identifiable class, Proc. 2nd Workshop on Comput. Learning Theory (1989), 328–333