Language Learning from Stochastic Input

Shyam Kapur
Institute for Research in Cognitive Science
University of Pennsylvania
3401 Walnut Street–Rm 412C
Philadelphia PA 19104
skapur@linc.cis.upenn.edu

Gianfranco Bilardi
Dipartimento di Elettronica ed Informatica
Universitā di Padova
Via Gradenigo 6/A
35131 Padova Italy
bilardi@dei.unipd.it

Abstract

Language learning from positive data in the Gold model of inductive inference is investigated in a setting where the data can be modeled as a stochastic process. Specifically, the input strings are assumed to form a sequence of identically distributed, independent random variables, where the distribution depends on the language being presented. A scheme is developed which can be tuned to learn, with probability one, any family of recursive languages, given a recursive enumeration of total indices for the languages in the family and a procedure to compute a lower bound to the probability of occurrence of a given string in a given language. Variations of the scheme work under other assumptions, e.g., if the probabilities of the strings form a monotone sequence with respect to a given enumeration. The learning algorithm is rather simple and appears psychologically plausible. A more sophisticated version of the learner is also developed, based on a probabilistic version of the notion of tell-tale subset. This version yields, as a special case, Angluin's learner for the families of languages that are learnable from all texts (and not just from a set of texts of probability one).

1 Introduction

In the Gold paradigm for inductive inference [Gol67], the learner is presented with the *text* of a language (all strings in any order with possible repetitions) that belongs to a specified family of languages. This model is motivated by the well-established hypothesis that the child learns her native language from positive evidence alone. (For a discussion, see [Ber85].) The learner is said to learn a language if, on any text for it, the learner's guess *converges* to the same language, i.e., from some point onwards, the guess coincides with the language being presented. The learner is said to learn the family if it learns each language in the family.

Angluin [Ang80] characterized the families learnable in the Gold paradigm. The requirement of convergence on *every* text of each language turns out to be too stringent. Gold [Gol67] suggested that by imposing probabilistic assumptions on texts for a language, and requesting convergence only with probability one, the class of identifiable families could be enriched. Stochastic input could provide some form of *indirect* negative evidence of the type that has often been suggested for natural language acquisition [Pin84, Cla90]. Angluin [Ang88] studied the case of a stochastic input where the distribution of each language is essentially known to the learner in the form of a procedure that allows to compute it. It is shown that families not learnable from all texts become learnable with probability 1. Furthermore, the distribution itself is learned, not just the supporting language.[1]

In this work, we study the learning problem in the case of stochastic input, under relatively mild assumptions on the input distribution (e.g., a lower bound to the distribution is computable, or the distribution is monotone with respect to a canonical enumeration of the strings). The target is the identification in the limit, with probability one, of the language being presented, not of the distribution according to which it is presented. Indeed, the distribution need neither be computable nor even be representable in any finite form.

We develop a learning algorithm computationally simple and psychologically plausible. (For some applications to natural language acquisition, see [Kap92].) No complicated functions are computed or distributions estimated. Learning proceeds as if each language is in isolation, and, at any stage, only the guessed language and any parameters associated with it play any role.

[1] Angluin also showed that her work subsumes the previous work. (In particular, [Hor69] and [vdMW78].)

COLT'92-7/92/PA,USA

In contrast, many learning algorithms proposed in the literature continue evaluating various functions of languages different from the one being presented, even after they have converged.

In Section 2, we define the basic framework of our model. In Section 3, we define a *recognition problem* which is the problem to recognize whether the language being presented is the same as a given language of the family. We show a systematic way to obtain a learner from a recognizer. Since the recognizer is simpler to define and to analyze than the learner, this approach is of independent interest.

In Section 4, we specify a particular recognizer and analyze its behavior. Our recognizer can be tuned to work correctly whenever a lower bound is computable for the probability that a given string occur in the input when a given language is being presented. We also discuss variations of the scheme which work in other situations.

Our recognizer makes no assumptions on—and takes no advantage of—the structure of the family. Indeed, under the proper probabilistic assumptions, it will work for any family, but without such assumptions it could fail even on a family that is learnable from all texts. This is not surprising if one considers that, as shown by Angluin [Ang80], learning a family from all texts is equivalent to the ability of recursively enumerating a so-called tell-tale subset for each language in the family. At the same time, a tell-tale subset enumerator is not in general computable from the description of the family (even if there is one) [KB91, Kap91]. Motivated by the preceding considerations, in Section 5, we show how the recognition algorithm can be generalized so that learning from all texts arises as a special case within this setting. We conclude with the hope that our development could provide useful hints for understanding the role played by indirect negative evidence in the learning process.

2 Model

Let Σ be a finite alphabet and Σ^* be the set of all finite strings formed by concatenating elements of Σ. Let there be a canonical enumeration of Σ^*. We will use the notation $x < y$ to indicate that the string x appears before the string y in this enumeration. (The special symbol x_0, which is not a string in Σ^*, is considered to be the least string and is output in front of the enumeration.) Let $M_1, M_2, M_3, \ldots$ be any standard enumeration of all Turing machines over Σ. For any *index* $I \in Z_+$, let W_I denote the *language* (subset of Σ^*) accepted by the machine M_I. Thus, the W_Is form an enumeration of all *recursively enumerable (r.e.)* languages.

An index I is *total* if the corresponding machine M_I is total and accepts a non-empty language. A recursive enumeration of total indices $\mathcal{I} : I_1, I_2, \ldots, I_k, \ldots$ defines a family $\mathcal{F} = \{W_I : I \in \mathcal{I}\}$. Let $\mathcal{P} : p_1, p_2, \ldots, p_k, \ldots$ be a sequence of functions, where for each k, $p_k : \Sigma^* \mapsto [0,1]$ is a probability distribution on strings, i.e., $\sum_{x \in \Sigma^*} p_k(x) = 1$. In addition, $p_k(x) > 0$ if and only if $x \in W_{I_k}$. (The language W_{I_k} is said to be the *support* of p_k.) No assumption is made regarding the computability of the p_ks or the enumerability of the sequence $\mathcal{P}$. We will refer to $(\mathcal{I}, \mathcal{P})$ as a stochastic family. Based on p_k, we can define a unique, complete probability measure Pr_k on the infinite product of W_{I_k} with itself [Neu73]. A text t for (I_k, p_k) is a stochastic process consisting of a sequence $t_1, t_2, \ldots, t_n, \ldots$ of independent random variables, all distributed according to p_k. In other words, $\forall n_1, n_2, \ldots, n_s \in Z_+$, where $n_1 < n_2 \ldots < n_s$, and $\forall x_1, x_2, \ldots, x_s \in \Sigma^*$,

$$Pr_k[t_{n_1} = x_1, \ldots, t_{n_s} = x_s] = p_k(x_1) \cdots p_k(x_s).$$

An *inductive inference machine (IIM)* M is an algorithmic procedure (say, a Turing machine) whose input is a text t and whose output is a sequence of non-negative integers $M(\bar{t}_1), M(\bar{t}_2), \ldots$ constrained to be either 0 or total indices. (We use the notation $\bar{t}_n$ to denote the sequence of values of the first n random variables in t. We denote by $content(\bar{t}_n)$ the set of these values.) The procedure works in stages, but it may never complete some stage. At the nth stage, the value of t_n is input and $M(\bar{t}_n)$ is output. The intended interpretation is as follows: if $M(\bar{t}_n) = 0$, then the IIM makes no guess; otherwise, it guesses the language $W_{M(\bar{t}_n)}$.

An IIM M is said to converge to an index I if there is a k such that $M(\bar{t}_k) = I$ and, for all $n > k$, $M(\bar{t}_n) = M(\bar{t}_k)$. We let $M(t)$ be I if, on the text t, M converges to the index I, and we let $M(t)$ be $\perp$ if, on the text t, M does not converge to any index. It can be shown that the set of infinite sequences of strings from W_{I_k} on which an IIM M converges to an index for W_{I_k} is a measurable set, and the Pr_k measure of the subset of infinite sequences from W_{I_k} which are not texts for it is zero. (For details, see [Kap91].) Thus, the set of texts from W_{I_k} on which an IIM M converges to an index for W_{I_k} is measurable.

Definition 1 An IIM M learns $(\mathcal{I}, \mathcal{P})$ with probability one (henceforth, w.p.o.) if, for all $I_k \in \mathcal{I}$,

$$Pr_k[W_{M(t)} = W_{I_k}] = 1.$$

It should be observed that, in a stochastic family $(\mathcal{I}, \mathcal{P})$, we allow a language to have more than one index in the set $\mathcal{I}$. In case $W_{I_h} = W_{I_k}$, it is not required that p_h be identical to p_k. Further, we only require that the learner converges w.p.o. to an index for the language being presented, and not that the distribution itself is learned in any sense.

There is no IIM that would learn every possible pair of indexed families and probability distributions. Angluin [Ang88] showed that in a 'distribution-free' setting (where the input distribution can be arbitrary), the families that can be learned w.p.o. can also be learned from all texts. However, new families could be learned from inputs guaranteed to be generated from a restricted class of distributions.

3 Recognizer

Consider first the standard setting where a text t for a language L is any infinite sequence of all and only the strings of L and a family $\mathcal{F}$ is said to be learned by an IIM M if on every text t for any $L \in \mathcal{F}$, $M(t)$ is an index for L. In order to simplify the study of learning from stochastic input, let us first define a *recognition* problem in this setting that has intimate connections to learning. Intuitively, in order to recognize a language in a family, an IIM must converge to the language if and only if the input is a text for that language. Let a recursive enumeration of total indices $\mathcal{I} : I_1, I_2, \ldots, I_k, \ldots$ define a family $\mathcal{F} = \{W_I : I \in \mathcal{I}\}$.

Definition 2 An IIM R is said to *recognize* a language $L \in \mathcal{F}$ if the following conditions are met:

1. On every text t for L, $R(t) \neq \perp$ and $W_{R(t)} = L$.
2. On every text t for any $L' \in \mathcal{F}$ such that $L' \neq L$, $R(\bar{t}_n) = 0$ infinitely often.

The connection between the recognition problem and learning is brought out by the proposition that follows the definition below of a *Uniform Recognizer*.

Definition 3 A Uniform Recognizer for $\mathcal{F}$ is a procedure that, given a total index I such that $W_I \in \mathcal{F}$, returns an IIM R_I which recognizes the language W_I.

Proposition 1 *Given an IIM M that learns a family $\mathcal{F}$, we can effectively construct a Uniform Recognizer for $\mathcal{F}$. Given an $\mathcal{I}$-enumerator and a Uniform Recognizer for $\mathcal{F}$, we can effectively construct an IIM M that learns $\mathcal{F}$.*

Proof: Consider first that a Uniform Recognizer is given. Let $R_{I_1}, R_{I_2}, \ldots$ be a recursive enumeration of the recognizers obtained by giving this Uniform Recognizer as input the indices I_1, $I_2, \ldots$ enumerated by the $\mathcal{I}$-enumerator. Consider a recursive enumeration of these recognizers $R_{I_1}, R_{I_1}, R_{I_2}, R_{I_1}, R_{I_2}, R_{I_3}, \ldots$, in which each recognizer appears infinitely often. Let $R_{\phi(n)}$ be the nth machine in this enumeration. Then, it is easy to learn the family by running these recognizers in a systematic manner such that each recognizer potentially gets the control infinitely often. The control from a particular recognizer is taken away whenever it outputs a 0. The IIM M that we claim learns $\mathcal{F}$ is described precisely below.

Initialization: $j := 1$
Stage n $(n \geq 1)$: $M(\bar{t}_n) := R_{\phi(j)}(\bar{t}_n)$;
if $R_{\phi(j)}(\bar{t}_n) = 0$ **then** $j := j + 1$.

For any $k > 0$, consider any text t for W_{I_k}. The following two claims establish that M must converge on t to an index for W_{I_k}.

Claim 1 *If the variable j in the execution of M on t reaches a final value of j^*, then $R_{\phi(j^*)}$ must be a recognizer for W_{I_k}.*

Proof: If j never increases beyond j^*, then the recognizer $R_{\phi(j^*)}$ must make a non-zero output at every stage subsequent to the stage at which j got to j^*. By the definition of recognition, only the recognizer for W_{I_k} can behave in such a fashion on a text for W_{I_k}. ∎

Claim 2 *The variable j in the execution of M on t must reach a final value.*

Proof: Suppose j does not reach a final value. Let j^* be the least j such that $R_{\phi(j)}$ is a recognizer for W_{I_k}. Then, at an infinite number of stages, the variable j must be such that $R_{\phi(j)} = R_{\phi(j^*)}$. Since $R_{\phi(j^*)}$ recognizes every text for W_{I_k}, we have that $W_{R_{\phi(j^*)}(t)} = W_{I_k}$. By definition, there is an n^* such that, for all $n \geq n^*$, $R_{\phi(j^*)}(\bar{t}_n) \neq 0$. Suppose at some stage $k^* \geq n^*$, j is such that $R_{\phi(j)} = R_{\phi(j^*)}$. Then, j would not increase beyond the stage k^*, which is a contradiction. ∎

For the converse, suppose an IIM M learns $\mathcal{F}$. Let $\rho(i, j, n)$ be *true* if and only if W_i and W_j agree up to the nth string in the enumeration of Σ^*. Consider the procedure that, given as input a total index I such that $W_I \in \mathcal{F}$, returns an IIM R_I which behaves as follows:

Stage n $(n \geq 1)$:
if $\rho(I, M(\bar{t}_n), n)$ **then** $R_I(\bar{t}_n) := M(\bar{t}_n)$
else $R_I(\bar{t}_n) := 0$.

We claim that the procedure described is a Uniform Recognizer. Clearly, the construction of R_I from I is uniform. In order to show that R_I recognizes W_I, we distinguish two cases. Suppose first that t is a text for W_I. Then M must converge to an index for W_I and, since for any n beyond the onset of convergence $\rho(I, M(\bar{t}_n), n)$ will be *true*, R_I will also converge to an index for W_I. On the other hand, if t is a text for some other language in $\mathcal{F}$, then M will converge to an index for that other language. Beyond the point of convergence, for all n beyond some n^* (at which the guessed language and W_I first differ), $\rho(I, M(\bar{t}_n), n)$ will be *false*. Thus, R_I will output 0 at all subsequent stages. ∎

There is a probabilistic analog of the recognition problem which has useful connections to probabilistic learning. Let $(\mathcal{I}, \mathcal{P})$ be a stochastic family.

Definition 4 An IIM R is said to recognize (I_k, p_k) w.p.o. if the following conditions are met:

1. On t for (I_k, p_k), $Pr_k[W_{R(t)} = W_{I_k}] = 1$.
2. On t for any (I_h, p_h) such that $W_{I_h} \neq W_{I_k}$,
$$Pr_h[R(\bar{t}_n) = 0 \text{ } \textit{infinitely often}] = 1.$$

Notice that for recognition of (I_k, p_k) w.p.o., no condition is required on the behavior of the IIM on a text for (I_h, p_h) such that $W_{I_h} = W_{I_k}$.

The connection between recognition and learning w.p.o. is brought out by the proposition that follows the definition below. The proof of the proposition is similar to that for Proposition 1.

Definition 5 A Uniform Recognizer for $(\mathcal{I},\mathcal{P})$ is a procedure that, given I_k, returns an IIM R_{I_k} which recognizes (I_k,p_k) w.p.o..

Proposition 2 *Given an IIM M that learns $(\mathcal{I},\mathcal{P})$ w.p.o., we can effectively construct a Uniform Recognizer for $(\mathcal{I},\mathcal{P})$. Given an $\mathcal{I}$-enumerator and a Uniform Recognizer for $(\mathcal{I},\mathcal{P})$, we can effectively construct an IIM M that learns $(\mathcal{I},\mathcal{P})$ w.p.o..*

In the next section, we consider one particular approach to obtaining a Uniform Recognizer for $(\mathcal{I},\mathcal{P})$. Due to Proposition 2, this can be used to learn the family w.p.o. from stochastic input.

4 Window-based Learners

One way to solve the recognition problem is based on the idea of *confirmation* of strings. The IIM could wait within some suitable *window* of the input for a particular string in the language. In case the string shows up, the string is said to be confirmed and the machine continues to output the index for the language. The machine next tries to confirm the next string in the language. Otherwise, at the particular stage at which the window got over, the machine outputs a 0 and tries to confirm the same string again. In this way, the machine makes progress through a sequence of windows of various lengths, during each of which it is selective for a specific string from the language. We next specify the recognizer formally.

Let $(\mathcal{I},\mathcal{P})$ be a stochastic family. Consider the IIM R_{I_k} obtained from the total index I_k as shown in Figure 1. We assume that W_{I_k} is infinite. (The case that W_{I_k} may be finite can easily be handled.) Notationally, $content(\bar{t}_0)$ is the empty set; $next(x,I)$ denotes the smallest string in W_I greater than x, and $\lambda_k(x)$ returns a positive integer. Next, we investigate under which conditions R_{I_k} is a recognizer w.p.o. for (I_k,p_k) in the stochastic family $(\mathcal{I},\mathcal{P})$.

We observe that, on a text t for a language L different from W_{I_k}, R_{I_k} will output infinitely many 0's. This is clearly the case if $L \not\subseteq W_{I_k}$. Otherwise, let u be the least string in $W_{I_k} \setminus L$. Then, R_{I_k} will fail infinitely often to confirm some $v \leq u$. Therefore, Condition 2 of Definition 4 is satisfied by R_{I_k}, for any $(\mathcal{I},\mathcal{P})$. A more careful analysis is needed for Condition 1.

Let $x_1,x_2,\ldots$ be an enumeration of the language W_{I_k} in increasing order. In the probability space defined by the text for (I_k,p_k), for each $i,j \in Z_+$, we define $A_{i,j}$ to be the event that the jth attempt at the confirmation of the string x_i took place and failed. That means that the machine for the jth time set up a window in which it was selective for the string x_i and the string x_i did

```
n := 0;
u := next(x_0, I_k);
while content(t̄_n) ⊆ W_{I_k} do
    begin {attempt at confirming u}
        found := false;
        for m := 1 to λ_k(u) − 1 do
            begin
                n := n + 1;
                if t_n = u then found := true;
                R_{I_k}(t̄_n) := I_k
            end;
        n := n + 1;
        if t_n = u then found := true;
        if found then
            begin {attempt succeeded}
                u := next(u, I_k);
                R_{I_k}(t̄_n) := I_k
            end
        else {attempt failed}
            R_{I_k}(t̄_n) := 0
    end;
while true do
    begin
        n := n + 1;
        R_{I_k}(t̄_n) := 0
    end.
```

Figure 1: The IIM R_{I_k} for recognizing (I_k,p_k) w.p.o.

not show up in that duration. Since R_{I_k} never seeks to confirm a string that has already been confirmed, there are only two kinds of divergent behaviors possible for R_{I_k}. R_{I_k} is said to undergo *static* divergence if, for some string x_i, all attempts at confirmation of x_i fail, that is, all the events $A_{i,1},A_{i,2},A_{i,3},\ldots$ take place. R_{I_k} is said to undergo *dynamic* divergence if R_{I_k} fails to confirm an infinite number of different strings in the first attempt. Thus R_{I_k} undergoes dynamic divergence whenever an infinite subsequence of events from the sequence $A_{1,1},A_{2,1},A_{3,1},\ldots$ occur.

We first consider the event D_s of static divergence and show that $Pr_k[D_s] = 0$. To this end, we define, for $i \geq 0$, the event

$$B_i = \text{there is no attempt to confirm } x_{i+1}$$

and observe that $D_s = \bigcup_{i=0}^{\infty} B_i$. Our claim that $Pr_k[D_s] = 0$ then follows from the next proposition.

Proposition 3 *For all $i \geq 0$, $Pr_k[B_i] = 0$.*

Proof: By induction on i. Trivially, the base case $Pr_k[B_0] = 0$ is true. Assuming that $Pr_k[B_{i-1}] = 0$, we easily have that

$$Pr_k[A_{i,1}] = Pr_k[A_{i,1}|\bar{B}_{i-1}] = (1-p_k(x_1))^{\lambda_k(x_1)},$$

since, given that an attempt to confirm x_i is made ($\bar{B}_{i-1}$ happens), the attempt fails if and only if $t_n \neq x_i$ for $\lambda_k(x_i)$ consecutive values of n. In general, the jth

attempt at the confirmation of the string x_i can fail if the attempt took place (i.e., the event $A_{i,j-1}$ must have taken place) and the string x_i did not show up during that attempt. Thus, for any j,

$$\begin{aligned} Pr_k[A_{i,j}] &= Pr_k[A_{i,j} \cap A_{i,j-1}] \\ &= Pr_k[A_{i,j}|A_{i,j-1}]Pr_k[A_{i,j-1}]. \end{aligned}$$

Clearly,

$$Pr_k[A_{i,j}|A_{i,j-1}] = (1 - p_k(x_i))^{\lambda_k(x_i)}.$$

By easy induction, we establish that

$$Pr_k[A_{i,j}] = (1 - p_k(x_i))^{\lambda_k(x_i)j}.$$

We now observe that

$$B_i = B_{i-1} \cup (\bar{B}_{i-1} \cap A_{i,1} \cap A_{i,2} \cap \ldots).$$

Hence, since $Pr_k[B_{i-1}] = 0$, we have

$$Pr_k[B_i] = Pr_k[\bigcap_{j=1}^{\infty} A_{i,j}].$$

Since $A_{i,1} \supset A_{i,2} \supset \ldots$ and $\lambda_k(x_i) \geq 1$, we have that

$$\begin{aligned} Pr_k[\bigcap_{j=1}^{\infty} A_{i,j}] &= \lim_{j\to\infty} Pr_k[A_{i,j}] \\ &= \lim_{j\to\infty} (1 - p_k(x_i))^{\lambda_k(x_i)j} = 0, \end{aligned}$$

and therefore $Pr_k[B_i] = 0$. ∎

As a by product of the above proof we have:

Corollary 1 *For all $i, j \geq 1$,*

$$Pr_k[A_{i,j}] = (1 - p_k(x_i))^{\lambda_k(x_i)j}.$$

Let us next consider the event of dynamic divergence, D_d. Consider the events $A_{1,1}, A_{2,1}, A_{3,1}, \ldots$. Due to Proposition 3, w.p.o., there is at least one attempt to confirm each string x_i. Since confirmation of different x_i's happens at different stages, it can be seen that the events $A_{1,1}, A_{2,1}, A_{3,1}, \ldots$ are statistically independent. Then, from a combination of First and Second Borel-Cantelli Lemmas [Neu73], we conclude that, w.p.o., only finitely many of these events occur if and only if $\sum_{i=1}^{\infty} Pr_k[A_{i,1}]$ converges. Equivalently, $Pr_k[D_d] = 0$ if and only if $\sum_{i=1}^{\infty} Pr_k[A_{i,1}] < \infty$. Recalling that R_{I_k} recognizes (I_k, p_k) if and only if there is no divergence (static or dynamic), from the above considerations and Corollary 1, we obtain the following characterization.

Theorem 1 *The IIM R_{I_k} recognizes (I_k, p_k) w.p.o. if and only if*

$$\sum_{i=1}^{\infty} (1 - p_k(x_i))^{\lambda_k(x_i)} < \infty.$$

For $0 \leq p \leq 1$, it is easy to see that $(1 - p) \leq e^{-p}$. Using this upper bound, we can establish the following corollary.

Corollary 2 *The IIM R_{I_k} recognizes (I_k, p_k) w.p.o. if*

$$\sum_{i=1}^{\infty} e^{-p_k(x_i)\lambda_k(x_i)} < \infty.$$

Consider next one example of a possible relationship between p_k and λ_k that guarantees recognition. For some $\epsilon > 0$ and for all $i \geq 1$, suppose

$$p_k(x_i)\lambda_k(x_i) \geq (1 + \epsilon) \ln i.$$

Now

$$\sum_{i=1}^{\infty} e^{-p_k(x_i)\lambda_k(x_i)} \leq \sum_{i=1}^{\infty} \frac{1}{i^{(1+\epsilon)}} < +\infty.$$

Note that this requirement for recognition w.p.o. does not depend on the computability of the input distribution p_k. Further, it is independent of both the structure of the family and the distributions according to which the other languages in the family may be presented.

The relationship between p_k and λ_k in the example above can be viewed in the following perspective. In order for R_{I_k} to recognize w.p.o., clearly we cannot let λ_k be a constant function. Since the probabilities of the strings in the language must decrease arbitrarily, it is natural to expect that λ_k must increase in proportion to the inverse of the probabilities. Making λ_k exactly the inverse of the probability function is not enough either. If, for all $i \geq 1$, $p_k(x_i)\lambda_k(x_i) = 1$, then it is easy to see using Theorem 1 that R_{I_k} will fail to recognize. The λ_k function needs to be related to the inverse of the input probability by a function that grows unbounded as a function of i. The above example illustrates that a slow-growing function such as $\ln i$ is sufficient.

We next determine sets of probability distributions (the p_k's) for which a single computable λ_k function can be constructed such that R_{I_k} recognizes (I_k, p_k) w.p.o., where p_k is one of those distributions. Our task is simplified because such sets can be obtained for a language in the family independent of both the structure of the family as well as the distributions according to which the other languages may be presented. No λ_k function exists that would work for the set of all distributions; if the set consists of a single computable distribution, we can easily construct this function. We need a definition to state the next proposition.

Definition 6 A distribution p is said to *dominate* a distribution q with the same support, if $(\forall^{\infty} x \in \Sigma^*)(p(x) \geq q(x))$.

The next proposition is easily established due to the window-based nature of R_{I_k}.

Proposition 4 *Given a computable distribution q_k whose support is W_{I_k}, the function λ_k can be constructed so that R_{I_k} recognizes (I_k, p_k) w.p.o. whenever p_k dominates q_k.*

Instead of making the function λ_k depend only on the string u, it is shown next that, if the window sizes can be adjusted during the execution of the algorithm, input from a larger class of distributions can be recognized.

Proposition 5 *Let there be a recursive enumeration of computable distributions $q_{k,1}, q_{k,2}, \ldots$, each of whose support is W_{I_k}. Then we can construct an IIM R_{I_k} that recognizes (I_k, p_k) w.p.o., whenever p_k dominates at least one of those distributions.*

Proof: We give an informal proof. Recall that the function λ_k in the construction of R_{I_k} above is used to determine the size of the window to be used for various strings. Now we set the window sizes as follows. As long as convergence has not taken place, the size of the windows is determined according to different distributions. For example, for a string x_j, the size of the window may be set according to the distribution $q_{k,i}$ by making it $2 \ln j / q_{k,i}(x_j)$. The different distributions are chosen in a systematic fashion such that each distribution potentially gets an infinite number of chances to get window-sizes set according to it. For example, this may be done by scanning them in the order $q_{k,1}, q_{k,1}, q_{k,2}, q_{k,1}, q_{k,2}, q_{k,3}, \ldots$. It can easily be seen that, if p_k dominates any distribution in the sequence, then R_{I_k} recognizes (I_k, p_k) w.p.o.. ∎

In the construction above, the sizes of the windows were set according to different distributions by switching between them only whenever a 0 was output because some string was not confirmed. Following a different approach, we show that recognition can take place with different type of *a priori* information about the input distribution. The information required is about the relative probabilities of the strings in the language.

Theorem 2 *For each string $x \in W_{I_k}$, we are given a recursive enumeration of a set $\Psi_x^{(k)} \subseteq W_{I_k}$ which includes all except a finite number of strings from W_{I_k}. A distribution p with the support W_{I_k}, which has the property that*

$$(\forall x \in W_{I_k})(\forall y \in \Psi_x^{(k)})(p(x) \geq p(y)),$$

is said to be **good**. *We claim that we can construct an IIM R_{I_k} that recognizes (I_k, p_k) w.p.o. if the input distribution p_k is good.*

Proof: Consider a text for (I_k, p_k). Suppose at some stage the recognizer R_{I_k} wants to decide on the window-size to use for the next string, say u.

The machine R_{I_k} does not decide on the window-size right at the start of the window. To determine the window-size for the string u, it begins to enumerate $\Psi_u^{(k)}$. It also reads the subsequent presentation till it either finds a string in the text common with the enumeration of $\Psi_u^{(k)}$ or a string not in the guessed language. (It is easy to see that one or the other must happen.) If a string not in the guessed language is encountered first, then an inconsistency has been found and hence the window can be immediately terminated. Otherwise, let the string found be v. Since $p_k(v) \leq p_k(u)$, then the probability that m occurrences of v will take place and none of u is certainly less than 2^{-m}. Recall that $ord(u)$ is the position of the string u in the standard enumeration of the language W_{I_k}. It is adequate to close the window whenever v has appeared m times, where m is such that $2^{-m} < (1/ord(u))$. The window-sizes constructed for various strings when W_{I_k} is the current guess can easily be shown to be sufficient to ensure that R_{I_k} recognizes W_{I_k} w.p.o.. ∎

As a special case of the above theorem, we have

Corollary 3 *A distribution p with the support W_{I_k} is* monotonic *if, for all $i \geq 1$, $p(x_i) \geq p(x_{i+1})$. We can construct an IIM R_{I_k} that recognizes (I_k, p_k) w.p.o. if the input distribution p_k is monotonic.*

Proof: Since p_k is monotonic, for each string $x \in W_{I_k}$, we can recursively enumerate exactly the set of strings $y \in W_{I_k}$ such that $p_k(x) \geq p_k(y)$. Clearly, given this enumeration for each string $x \in W_{I_k}$, the input distribution is good as defined in the statement of Theorem 2. Hence, by an application of Theorem 2, the result follows. ∎

5 Weighted Tell-tale Sets

An analysis of our window-based Uniform Recognizer reveals that it would not recognize an infinite language on all texts, even if the family where learnable from all texts. As established in [Ang80], learnability from all texts is equivalent to the existence of a uniform enumerator of tell-tale subsets for each language in the family. We recall that a finite set T_k is a *tell-tale subset* of language W_{I_k} in the family $W_{I_1}, W_{I_2}, \ldots$ if there is no $W_{I_h} \subset W_{I_k}$ such that $T_k \subseteq W_{I_h}$.

Tell-tale subsets give some information on the structure of the family. In general, they are not computable from a description of the family [KB91]. Below, we introduce the notion of a *weighted tell-tale set*. For a stochastic family $(\mathcal{I}, \mathcal{P})$, a weighted tell-tale set provides useful information on the structure of both the family and the probability distributions. This information can be exploited to construct a window-based recognizer w.p.o. which, when a weighted tell-tale set is finite, actually recognizes from all texts.

Definition 7 Consider a set V_k of pairs $\{(y_{k,1}, \lambda_{k,1}), (y_{k,2}, \lambda_{k,2}), \ldots\}$, where each pair consists of a string and a positive integer. Let $T_k = \{y_{k,1}, y_{k,2}, \ldots\}$. V_k is said to be a weighted tell-tale set of (I_k, p_k) in the stochastic family $(\mathcal{I}, \mathcal{P})$ if the following conditions are satisfied:

(i) $T_k \subseteq W_{I_k}$.

(ii) $\sum_i (1 - p_k(y_{k,i}))^{\lambda_{k,i}} < \infty$.

(iii) For every $h > 0$ such that $T_k \subseteq W_{I_h} \subset W_{I_k}$,

$$\sum_i (1 - p_h(y_{k,i}))^{\lambda_{k,i}} = \infty.$$

If a weighted tell-tale set V_k is such that T_k is finite, then T_k is a tell-tale subset. For if there were any W_{I_h} such that $T_k \subseteq W_{I_h} \subset W_{I_k}$, the summation in Condition (iii) could not diverge as it contains only a finite number of terms. Conversely, if T_k is a finite tell-tale subset for W_{I_k}, then any set V_k obtained by pairing each string in T_k with any positive integer whatsoever is a weighted tell-tale set for (I_k, p_k). Notice also that the only difference between the summations in Condition (ii) and Condition (iii) is that one involves the probability function p_k and the other p_h. As the summation represents a quantity decreasing with the probabilities, it can be viewed as an indicator of inverse likelihood of T_k. In this view, Condition (iii) says that T_k is "infinitely unlikely" for any language W_{I_h} such that $T_k \subseteq W_{I_h} \subset W_{I_k}$.

We will show that, to recognize (I_k, p_k), it is enough that the IIM confirms all and only the strings that appear in the enumeration of the weighted tell-tale set for (I_k, p_k). For the first time, we use the fact that Condition 2 for recognition (Definition 4) needs to be satisfied only w.p.o. and not always.

Theorem 3 *Given a procedure to recursively enumerate a weighted tell-tale set V_k for (I_k, p_k), an IIM R_{I_k} can be constructed which recognizes (I_k, p_k) w.p.o..*

Proof: Consider the IIM R_{I_k} obtained from the total index I_k as shown in Figure 2. R_{I_k} uses the recursive enumeration of V_k where it is assumed that, without loss of generality, no string is repeated.[2]

Suppose first that t is a text for (I_h, p_h), with $W_{I_h} \neq W_{I_k}$. We claim that R_{I_k} will output an infinite number of 0's. This is clearly the case if $W_{I_h} \not\subseteq W_{I_k}$. For $W_{I_h} \subset W_{I_k}$, suppose $T_k \not\subseteq W_{I_h}$. Let $y_{k,s}$ be the least string in $T_k \setminus W_{I_h}$. Then, R_{I_k} will fail infinitely often to confirm some $y_{k,s'}$, where $s' \leq s$. Suppose, on the other hand, that $T_k \subseteq W_{I_h} \subset W_{I_k}$. As in Section 4, we can define $A_{i,j}$ to be the event that the jth attempt at the confirmation of the string $y_{k,i}$ took place and failed. By a development parallel to that in Section 4, we can establish that, w.p.o., R_{I_k} will not converge to I_k if and only if

$$\sum_i (1 - p_h(y_{k,i}))^{\lambda_{k,i}} = \infty.$$

Thus, Condition (iii) in the definition of V_k ensures that R_{I_k} satisfies the second condition for recognition.

[2]Given any recursive enumeration of V_k, an enumeration without repetition of strings can be generated constructively.

Suppose that t is a text for (I_k, p_k). As above, we can define $A_{i,j}$ to be the event that the jth attempt at the confirmation of the string $y_{k,i}$ took place and failed. By a development parallel to that in Section 4, we can establish that the first condition for recognition is satisfied if and only if

$$\sum_i (1 - p_k(y_{k,i}))^{\lambda_{k,i}} < \infty.$$

Thus, Condition (ii) in the definition of V_k ensures that R_{I_k} satisfies the first condition for recognition.

Since the two cases considered are exhaustive, we have shown that the IIM R_{I_k} recognizes (I_k, p_k) w.p.o..

```
n := 0;
u := x_0;
w := 1;
s := 1;
u := next(x_0, I_k);
while content(t̄_n) ⊆ W_{I_k} do
    begin {attempt at confirming u}
        found := false;
        for m := 1 to w − 1 do
            begin
                n := n + 1;
                if t_n = u then found := true;
                R_{I_k}(t̄_n) := I_k
            end;
        n := n + 1;
        if t_n = u or u = x_0 then found := true;
        if found then
            begin {attempt succeeded}
                Run the enumerator for V_k
                for n steps.
                if pair (y_{k,s}, λ_{k,s}) is output then
                    begin
                        u := y_{k,s};
                        w := λ_{k,s};
                        s := s + 1
                    end
                else
                    begin
                        u := x_0;
                        w := 1
                    end
                R_{I_k}(t̄_n) := I_k
            end
        else {attempt failed}
            R_{I_k}(t̄_n) := 0
    end;
while true do
    begin
        n := n + 1;
        R_{I_k}(t̄_n) := 0
    end.
```

Figure 2: The IIM R_{I_k} that recognizes (I_k, p_k) using V_k

Intuitively, Theorem 3 suggests that a window-based IIM can succeed at learning a family w.p.o. by exploiting two different means for convergence to the right language. On a text for (I_k, p_k), some superset languages of W_{I_k} are defeated because the strings in their weighted tell-tale sets are not all contained in W_{I_k}. The remaining ones are excluded in the following way. Suppose, for some $h > 0$, $T_h \subseteq W_{I_k} \subset W_{I_h}$. At least an infinite number of strings in T_h have suitable weights $\lambda_{h,s}$'s to ensure that R_{I_h} sets window sizes that, if the input distribution is p_k, will lead, w.p.o., to an infinite number of failures in confirmation of T_h.

In case the weighted tell-tale set V_k for (I_k, p_k) is such that the set T_k is finite, then the IIM R_{I_k} will not only converge to I_k w.p.o. but, in fact, it will converge to I_k on all texts (which of course includes stochastic input according to any other distribution). If this is the case for the entire family, the learner constructed from the Uniform Recognizer will learn each language in the family on all texts.

6 Conclusion

In this paper, we defined a model of learning from stochastic input and obtained a uniform learning algorithm that works for every family of languages, under appropriate probabilistic assumptions. The results indicate that stochastic input provides a useful form of 'indirect negative evidence'. Our development could be particularly interesting in situations where it is more plausible to assume some *a priori* knowledge of the input distribution rather than that of the structure of the family to be learned.

Our results in Section 5 have also opened a promising territory for further investigation. One interesting question is whether the learning capability of the window-based IIMs can be characterized in terms of the ability to enumerate weighted tell-tale sets.

Acknowledgements

The work of S. Kapur was supported in part by the National Science Foundation grant IRI 90-16592, ARO grant DAAL 03-89-C-0031, DARPA grant N00014-90-J-1863, and Ben Franklin grant 91S.3078C-1. The work of G. Bilardi was supported in part by the Italian Ministry of University and Research and the National Research Council of Italy.

References

[Ang80] Dana Angluin. Inductive inference of formal languages from positive data. *Information and Control*, 45:117–135, 1980.

[Ang88] Dana Angluin. Identifying languages from stochastic examples. Technical Report 614, Yale University, March 1988.

[Ber85] Robert Berwick. *The Acquisition of Syntactic Knowledge*. MIT press, Cambridge, MA, 1985.

[Cla90] Robin Clark. Papers on learnability and natural selection. Technical Report 1, Université de Genève, Département de Linguistique générale et de linguistique française, Faculté des Lettres, CH-1211, Genève 4, 1990. Technical Reports in Formal and Computational Linguistics.

[Gol67] E. M. Gold. Language identification in the limit. *Information and Control*, 10:447–474, 1967.

[Hor69] J. J. Horning. *A Study of Grammatical Inference*. PhD thesis, Stanford University, 1969.

[Kap91] Shyam Kapur. *Computational Learning of Languages*. PhD thesis, Cornell University, September 1991. Technical Report 91-1234.

[Kap92] Shyam Kapur. Some (potential) applications of formal learning theory results to natural language acquisition. Presented at a symposium at Cornell University on 'Syntactic Theory and First Language Acquisition: Cross Linguistic Perspectives', April 1992.

[KB91] Shyam Kapur and Gianfranco Bilardi. On uniform learnability of language families. Cornell University, 1991.

[Neu73] M. F. Neuts. *Probability*. Allyn and Bacon, Boston, 1973.

[Pin84] Steve Pinker. *Language Learnability and Language Development*. Harvard University press, Cambridge, MA, 1984.

[vdMW78] A. van der Mude and A. Walker. On the inference of stochastic regular grammars. *Information and Control*, 38:310–329, 1978.