A CHARACTERIZATION OF PROBABILISTIC INFERENCE

Leonard Pitt

Yale University

Abstract

Inductive Inference Machines (IIMs) attempt to identify functions given only input-output pairs of the functions. Probabilistic IIMs are defined, as is the probability that a probabilistic IIM identifies a function with respect to two common identification criteria: EX and BC. Let ID denote either of these criteria. Then ID_{prob}(p) is the family of sets of functions U for which there is a probabilistic IIM identifying every \( f \in U \) with probability \( \geq p \). It is shown that for all positive integers \( n \), ID_{prob}(1/n) is properly contained in ID_{prob}(1/(n+1)), and that this discrete hierarchy is the "finest" possible. This hierarchy is related to others in the literature.

1. Introduction

Inductive inference is the study of algorithms which attempt to synthesize programs computing a function when given only examples of the function as input. Research focuses on both general theoretical properties of inference techniques, and finding specific methods for inference within particular domains. Inductive inference has applications in linguistics, artificial intelligence, pattern recognition, and the philosophy of science, among others [1].

An Inductive Inference Machine (IIM) is a Turing machine \( M \) with an oracle which presents \( M \) with initial segments of the values of some total recursive function, \( f(0), f(1), f(2), \ldots, \) on a special input tape. The IIM outputs a sequence of guesses of programs based on the examples it has seen. (There is no restriction that the guesses output by an IIM be programs which compute total functions.) Note that since new input values may not be consistent with a current guess, the IIM may not be able to determine at any point whether a particular guess is correct.

For this reason, identification of functions is seen as an infinite process which happens "in the limit".

There are two standard criteria of successful identification in the limit of an IIM on a given function \( f \): EX and BC. It has been shown that no single deterministic IIM can identify in the limit every total recursive function (according to either criterion). We allow randomization in the computations of the IIM and ask: "are more classes of functions identifiable if we only require the inference machine to be correct with some probability \( p \leq 1 \)?"

Letting ID denote both EX and BC, we define probabilistic IIMs, and the probability \( \Pr[M \text{ ID-identifies } f] \). For \( p \geq 0 \), define

\[
\text{ID}_{prob}(p) = \{ U \mid U \text{ is a set of functions such that there exists an IIM } M \text{ such that } \Pr[M \text{ ID-identifies } f] \geq p \text{ for every } f \in U \}.
\]

Our results give a description of the structure of the classes \( \text{ID}_{prob}(p) \) as a function of \( p \). For both criteria there is a discrete hierarchy of classes, with "breakpoints" at the values \( 1/2, 1/3, 1/4, \ldots \). That is, for all \( n = 1,2,3,\ldots \), \( \text{ID}_{prob}(1/n) \) is a proper subclass of \( \text{ID}_{prob}(1/(n+1)) \); and if \( p_1 \) and \( p_2 \) are in the same half-open interval \( (1/(n+1),1/n) \), then \( \text{ID}_{prob}(p_1) = \text{ID}_{prob}(p_2) \). Also, for both criteria, the sets of functions that can be identified by some machine with probability \( p > 1/2 \), can be identified by some deterministic machine.

The precise statement of our main results gives an equivalence between three different models of computation for inductive inference: probabilistic identification defined herein, frequency identification introduced by Podnieks [10], and identification by teams of machines introduced by Smith [12]. (It also
settles an open problem of Podnieks for frequency identification.) This is somewhat unusual, for in many cases the introduction of new computational models for inductive inference has given rise to new and "orthogonal" hierarchies of identifiability.

Previous work includes that of Valiant [14], who considers probabilistic "concept learning" algorithms for boolean formulae. R. Freivalds has examined the probabilistic finite identification hierarchy [5]. Freivalds proves the existence of a probabilistic inference hierarchy, where successful inference is defined with respect to halting computations. Not many of the proof techniques for finite inference are amenable to adaptation to the more prevalent notions of "in the limit" inference. We have preliminary results indicating that there is a relationship between probabilistic finite inference, and teams of finite inference machines, similar to that described in this paper for limiting inference. Also, a recent paper by Wiehagen, Freivalds, and Kimber [19] investigates the advantages of probabilistic inductive inference strategies over deterministic ones when the strategies are required to converge to a correct answer within some fixed number of changes in hypotheses. It is also independently proved that EX_{prob}(p) = EX when p > 1/2, which is a special case of our Theorem 2.3.

2. Background and Statement of Results

For the remainder of this paper, ID will stand for both EX and BC.

The symbol $\cup$ is the set operation union, together with the assertion that the operands of the union are mutually disjoint; thus $S = \cup S_i$ states that not only is $S$ the union of the sets $\{S_i\}$, but also that for all $i \neq j$, $S_i \cap S_j = \emptyset$. If $S$ is a set and $I$ a multiset, then $|I|$ is the number of (not necessarily distinct) elements of the multiset $I$, and $|I \cap S|$ is the number of (not necessarily distinct) elements of the multiset $I$ which are also elements of $S$. We write "$E(k)$ i.o. (k)" (respectively, "$E(k)$ a.e. (k)") to indicate that the equation $E$ parameterized by integer variable $k$ is true for infinitely many (respectively, all but finitely many) integers $k$. The function $f$ ranges over all total recursive functions, and the function $\phi$ over all partial recursive functions. $f|_k$ is the restriction of the function $f$ to the domain $\{x \mid x \leq k\}$. The set $U$ will range over all subsets of total recursive functions.

We assume that a particular encoding of TM transducers as nonnegative integers has been chosen [7]. We denote the function computed by program $i$ by $\phi_i$. Thus $<\phi_i>_{i \in \mathbb{N}}$ is an acceptable numbering of all and only the partial recursive functions [8]. If $\phi_i = f$, then we say $i$ is a program index, or simply an index of the function $f$.

For each $f$, we define three sets GOOD$_f$, SLOW$_f$, and WRONG$_f$, which partition the natural numbers $\mathbb{N}$, as follows:

GOOD$_f = \{i \mid \phi_i = f\}$,

SLOW$_f = \{i \mid \phi_i \neq f, \text{ and for all } x \text{ such that } \phi_i(x) \neq f(x), \phi_i \text{ diverges on } x\}$,

WRONG$_f = \{i \mid \phi_i \neq f, \text{ there exists a number } x \text{ such that } \phi_i(x) \text{ converges } \neq f(x)\}$.

If $f$ is a total recursive function, then we say that $M$ is fed the graph of $f$ iff each element input to $M$ is a pair $<x,f(x)>$, and for every $x$ in the domain of $f$, $<x,f(x)>$ is input at least once to $M$.

**Definition:** $M$ EX-identifies $f$ (written $f \in EX(M)$) iff when fed the graph of $f$ in any order, $M$ outputs infinitely many numbers, $g_1,g_2,g_3,\ldots$, and for some $n$, $g_n = g_{n+1} = g_{n+2} = \ldots$, and $g_0$ is (the encoding of) a program that computes the function $f$. (EX abbreviates "EXplains $f$".)

**Definition:** $M$ BC-identifies $f$ (written $f \in BC(M)$) iff when fed the graph of $f$ in any order, $M$ outputs infinitely many numbers $g_1,g_2,g_3,\ldots$, such that $\phi_i = f$ a.e. (k). (M eventually outputs only "Behaviorally Correct" programs.)

Thus the BC criterion requires that all guesses of $M$ be correct past some finite initial number of incorrect guesses, whereas the EX criterion requires in addition that eventually these correct guesses be identical.

**Definition:** ID = $\{U \mid \text{there exists an IIM } M \text{ such that } U \subseteq ID(M)\}$.

Clearly EX $\subseteq$ BC. It is well known that EX is a proper subset of BC [3].

We say that $M$ is order independent iff the sequence of guesses that $M$ makes is independent of the particular order in which the graph of $f$ is input to $M$. It is easily shown that if $M$ is an IIM, then $M$ can be effectively transformed into an order independent IIM.
whenever subclass of BC,Jp).

input identification: Let M be a deterministic
without loss of generality that the graph of any
function to be identified by an IIM M will be presented
to M in the canonical order <0,f(0)>, <1,f(1)>,
<2,f(2)>,..., and that every IIM outputs the guess g_n
before receiving the input f(n+1).

A set U of functions is ID-identified by a team of n
machines if and only if there exist n (deterministic)
IIMs such that every function in U is ID-identified by at
least one of the n machines. A team of n machines
represents a kind of finite nondeterminism; after an
initial n-way nondeterministic choice among the
machines, the computation is deterministic.

Let IDteam(n) denote (U | U is ID-identified by a team
of n machines). C. Smith has shown that these classes
form a proper hierarchy:

Theorem 2.1: [12] For all integers n ≥ 1, IDteam(n)
is a proper subclass of IDteam(n+1).

Podnieks [10] defines BC-frequency identification as
follows. Let M be a (deterministic) IIM, and let its
input be f(0), f(1), f(2), ... . Let F_n denote the
fraction of correct programs for f among the first n
guesses of M, i.e. if the sequence of programs output by
M is g_1,g_2,g_3,..., then

\[ F_n = \left| \{ i : i ≤ n \text{ and program } g_i \text{ computes } f \} \right| / n. \]

We say that M BC-identifies f with frequency p
iff limit_{n→∞} inf \{ F_n \} ≥ p.

Definition: BCfreq(p) = \{ M | \exists f \text{ such that } M \text{ BC-identifies every } f \in U \text{ with frequency } ≥ p \}.

Podnieks has shown:

Theorem 2.2: [10] For every ε > 0, and for every
integer n ≥ 1, BCfreq(ε + 1/n) is a proper subclass of
BCfreq(1/n).

He conjectures that this hierarchy is dense, that is,
whenever 0 ≤ p_1 < p_2 ≤ 1, BCfreq(p_2) is a proper
subclass of BCfreq(p_1).

We may analogously define EX-frequency identification: Let M be a deterministic IIM, which on
input f, outputs the sequence of guesses g_1,g_2,g_3,...,

\[ L_{x}(g) = \left| \{ j : 1 ≤ j ≤ k \text{ and } g_j=g_j \} \right| / k. \]

M EX-identifies f with frequency p iff there exists a
guess g_i such that limit_{k→∞} inf \{ F_i(g_i) \} ≥ p, and \( \phi_i = f. \)

Definition: EXfreq(p) = \{ U | \forall f \in U \text{ there exists an IIM } M \text{ which identifies every } f \text{ with frequency } p \}.

If M EX-identifies f with frequency p then there is
some particular correct guess of f that occurs in M's
output sequence with frequency p.

Our main results are the following equivalence for
ID = EX or BC.

Theorem 2.3: For all integers n ≥ 1, for all real
numbers p, if 1/(n+1) < p ≤ 1/n, then
ID_{prob}(p) = IDfreq(p) = IDteam(n).

We can sum up the consequences of this theorem as
follows:

Corollary 2.4:

- The probabilistic hierarchies for EX and BC are
  not degenerate (this follows from the team
  hierarchies shown by Smith (Theorem 2.1).
- The probabilistic hierarchy is discrete, and not
  "continuous" or "dense": i.e. if p_1 < p_2, and
  both p_1 and p_2 are in the same interval
  (1/(n+1), 1/n], then ID_{prob}(p_1) = ID_{prob}(p_2). If
  p_1 and p_2 are in different intervals, then
  ID_{prob}(p_1) properly contains ID_{prob}(p_2).
- There is also a frequency hierarchy, both for EX
  and BC, and they too are discrete (disproving
  Podnieks' conjecture [10]).
- For both EX and BC, the team, probabilistic,
  and frequency hierarchies are identical.
- If p > 1/2, then ID_{prob}(p) = IDfreq(p) = IDteam(1) = ID. Thus allowing randomness doesn't
  increase the power of IIM computations; even if
  we only require correctness 51% of the time, we
  could have done it deterministically. Similarly, if
  there is some deterministic IIM which "in the
  limit" has greater than half of its guesses correct
  for every f in the class U, then there is a
deterministic IIM which identifies every f ∈ U.

3. Probabilistic IIMs

A probabilistic IIM M is a deterministic IIM with a
random 0-1 oracle called a coin. The IIM may "query"
(or "flip") the coin from time to time, and receive the
result of the flip (which is 0 or 1 equiprobably) on a
special read-only coin tape. Without loss of generality,
we assume that every probabilistic IIM repeats the
following loop: Receive the next value of \( f \); Guess a program index; Flip the coin; Execute a finite number of deterministic steps; Receive the next value, etc.

For a particular function \( f \), \( M \) may follow different computations depending on the sequence of coin flips. Since every IIM executes the above “read-guess-flip-compute” loop, the set of possible computations of \( M \) on \( f \) are representable as an infinite, labeled, complete binary tree which we denote by \( T_{M,f} \). The nodes of \( T_{M,f} \) are numbered in breadth first search order, from left to right, with the root node being numbered ‘1’. Nodes of \( T_{M,f} \) correspond only to those configurations of the machine \( M \) where \( M \) has just guessed an index for \( f \). The edges of \( T_{M,f} \) correspond to the results of coin flips. Thus any infinite path from the root (initial configuration) corresponds to a particular infinite sequence of coin flips resulting in a computation of \( M \) on input \( f \), and the labels of the nodes on the path will correspond to the guesses that \( M \) makes for program indices for the function \( f \) during that computation. If \( n \) is a node in \( T_{M,f} \), we let \( \text{ind}(n) \) denote the program index that \( M \) has just guessed when it is in the configuration corresponding to node \( n \). The depth of a node \( n \) in \( T_{M,f} \) is denoted \( d(n) \), where \( d(n) = \lfloor \log_2(n) \rfloor \). (Hence node 1 has depth 0, nodes 2 and 3 have depth 1, etc.) \( \text{Parent}(n) \) denotes the immediate ancestor of node \( n \) in \( T_{M,f} \). Note that for any probabilistic IIM \( M \), any function \( f \), and any number \( k \), there is a Turing machine which when fed the first \( k \) values of \( f \), and the description of \( M \), constructs \( T_{M,f} \) through the \( k \)-th level.

A path \( p \) of \( T_{M,f} \) is an infinite sequence of adjacent nodes \( \langle t_0, t_1, t_2, \ldots \rangle \), starting at the root node \( t_0 = 1 \), and going “down the tree, never changing directions”, so that for all \( i \), the \( i \)-th node \( t_i \) on \( p \) is a node occurring at depth \( i \) of \( T_{M,f} \).

Definition: Let \( p = \langle t_0, t_1, t_2, \ldots \rangle \) be a path in \( T_{M,f} \), and \( j \) be a program index. The path \( p \) converges to \( j \) iff \( \text{ind}(t_j) = j \) a.e. (\( k \)).

If path \( p \) converges to \( j \), then \( p \) corresponds to a possible computation of \( M \) with input \( f \), for which \( M \) converges to outputting \( \text{"j"} \) as its guess for a program index for \( f \).

Definition: Let \( p = \langle t_0, t_1, t_2, \ldots \rangle \) be a path in \( T_{M,f} \), and \( A \subseteq \mathbb{N} \). The path \( p \) BC-converges to \( A \) iff \( \text{ind}(t_j) \in A \) a.e. (\( k \)).

If path \( p \) BC-converges to \( A \), then \( p \) corresponds to a possible computation of \( M \) with input \( f \), for which \( M \) after some initial sequence of guesses, outputs only indices from the set \( A \). If a path \( p \) BC-converges to the set \( \text{GOOD}_f \), then \( p \) contains a sequence of coin flips which causes \( M \) to output a sequence of guesses corresponding to a single deterministic BC-identification of \( f \).

We would like to define \( \text{Pr}(M \text{ BC-identifies } f) \) as the percentage of paths of \( T_{M,f} \) which BC-converge to \( \text{GOOD}_f \), and \( \text{Pr}(M \text{ EX-identifies } f) \) as the percentage of paths which converge to some index in \( \text{GOOD}_f \).

We first must precisely define probability with respect to an IIM's computation on a given input. The “experiment” for which the probability is defined is the running of \( M \) with input \( f \), and the result is the particular infinite path that \( M \) follows which depends on the infinite sequence of results of the coin flips. Thus the set of possible outcomes of the experiment is \( \{ p \mid p \text{ is a path in } T_{M,f} \} \).

Following the standard methodology, we define a class of basic subsets of possible outcomes. For each node \( n \in T_{M,f} \), \( P_n = \{ \text{paths } p \in T_{M,f} \mid p \text{ contains node } n \} \). Then the function \( \text{Pr} : P_n \rightarrow [0,1] \) is defined by \( \text{Pr}[P_n] = 2^{-d(n)} \). It is easy to see that this is what we want from our probability measure: The probability of a randomly chosen path passing through node \( n \) should be \( 2^{-d(n)} \) since every path must pass through exactly one node at depth \( d(n) \) and we'd like these to be equiprobable.

Definition: For all \( A \subseteq \mathbb{N} \),

\[
C(A) = \{ p \mid \text{path } p \text{ in } T_{M,f} \text{, and } \exists a \in A \text{ such that } p \text{ converges to } a \};
\]

\[
B(A) = \{ p \mid p \text{ is a path in } T_{M,f} \text{ and } p \text{ BC-converges to } A \};
\]

Lemma 3.1: Let \( \mathcal{B}(\{P_n\}) \) be the smallest Borel field containing \( \{P_n\}_{n \in T_{M,f}} \), then \( \text{Pr} \) is a probability measure when extended to \( \mathcal{B}(\{P_n\}) \), and for all \( A \subseteq \mathbb{N} \), \( B(A) \) and \( C(A) \) are both elements of \( \mathcal{B}(\{P_n\}) \), i.e. are both measurable.

We now may define

Definition: Let \( T_{M,f} \) be the computation tree of the probabilistic IIM \( M \) with input \( f \). Then

\[
\text{Pr}([M \text{ BC-identifies } f]) = \text{Pr}([\text{GOOD}_f]);
\]

\[
\text{Pr}([M \text{ EX-identifies } f]) = \text{Pr}([\text{GOOD}_f]).
\]
This defines the probability that $M$ BC-identifies $f$ as the fraction of paths of $T_{M,f}$ which BC-converge to a correct program index for $f$; or the fraction of $M$'s possible computations which correspond to a single deterministic BC-identification of the function $f$, and the probability that $M$ EX-identifies $f$ as the fraction of $M$'s possible computations each of which correspond to a correct EX-identification of $f$.

Let $M$ be an IIM, and ID denote both EX and BC.

**Definition:** $M$ ID-identifies $U$ with probability $p$ (written $U \subseteq \text{ID}_p(M)$) iff for all $f \in U$, $\Pr[M \text{ ID-identifies } f] \geq p$:

$$\text{ID}_p(M) = \{U \mid \exists M \text{ such that } U \subseteq \text{ID}_p(M)\}.$$

### 4. Some Proofs

Theorem 2.3 is proved by relating each of $\text{ID}_p(M)$ and $\text{ID}_q(M)$ with $\text{ID}_r(n)$. Thus we have eight containments to prove:

1. $\text{EX}_r(n) \subseteq \text{EX}_r(1/n)$
2. $\text{EX}_r(n) \supseteq \text{EX}_r(p)$ when $p > 1/(n+1)$
3. $\text{BC}_r(n) \subseteq \text{BC}_r(1/n)$
4. $\text{BC}_r(n) \supseteq \text{BC}_r(p)$ when $p > 1/(n+1)$
5. $\text{EX}_r(n) \subseteq \text{EX}_r(1/n)$
6. $\text{EX}_r(n) \supseteq \text{EX}_r(p)$ when $p > 1/(n+1)$
7. $\text{BC}_r(n) \subseteq \text{BC}_r(1/n)$
8. $\text{BC}_r(n) \supseteq \text{BC}_r(p)$ when $p > 1/(n+1)$

The above containments are best understood in pairs; for example, numbers 1 and 2 together show that if $1/(n+1) < p \leq 1/n$, then $\text{EX}_r(n) = \text{EX}_r(p)$.

**Proof of Containments 4, 5, and 7:**

1. (and 3.) The set $U$ ID-identified by a team $M_1, M_2, ..., M_n$ of deterministic IIMs is ID-identified with probability $\geq 1/n$ by a probabilistic IIM which simulates an $n$-sided coin flip with its 2-sided coin, and then equiprobably chooses to simulate $M_i$ on input $f \in U$. □

2. (and 7.) The set $U$ ID-identified by the team $M_1, M_2, ..., M_n$ of deterministic IIMs is ID-identified with frequency $p \geq 1/n$ by the deterministic IIM $M$ which, on input $f \in U$, simulates each of $M_1, M_2, ..., M_n$ and outputs the guesses of the team members interleaved, thus for each $i$, $1 \leq i \leq n$, and $k \geq 0$, $M$'s $(kn+i)^{th}$ guess is $M_i$'s $k^{th}$ guess. □

**Proof of Containment 6:**

Let $U \subseteq \text{EX}_r(p)$, with $p > 1/(n+1)$. Let $M$ be an IIM which EX-identifies $U$ with frequency $p$. To show $U \subseteq \text{EX}_r(n)$, we construct a team $M_1, M_2, ..., M_n$ of IIMs which EX-identify $U$.

Let $\text{FREQ}_k = \{g_i \mid f_i(g_i) > 1/(n+1)\}$ for the sequence of guesses output by $M$ on input $f \in U$. By definition, there is some particular guess "g" of $M$ such that $g \in \text{FREQ}_k$ a.e. (k). On input $f_{|k'}$, each $M_i$ constructs $\text{FREQ}_k$ and orders the (at most $n$) elements of $\text{FREQ}_k$ by "seniority", i.e. each $M_i$ determines for how many consecutive previous values of $k$ the particular guess has been in $\text{FREQ}_k$. $M_i$ then outputs the $i^{th}$ element of $\text{FREQ}_k$ according to this ordering (if it exists). It is not difficult to show that since $g \in \text{FREQ}_k$ a.e. (k), eventually $g$ must occupy a particular fixed position $j$ in the ordering of each of the sets $\text{FREQ}_k$ for all sufficiently large $k$, and then $M_j$ EX-identifies $f$. □

The proofs of the remaining three containments are more subtle. The proofs of containments 4 and 8 above are fairly similar, so we prove only numbers 2 and 4.

**Proof of Containment 4:**

In order to prove Containment 4, we first need some definitions and useful lemmas.

Consider IIMs which, rather than outputting a sequence of program indices, instead output a sequence of ordered pairs $<t_1, I_1>, <t_2, I_2>, ..., $ where each $t_k$ is a positive rational number, and each $I_k$ is a finite multiset of program indices.

**Definition:** $M$ BC$_{\text{threshold}}$-identifies $f$ (written $f \in \text{BC}_{\text{threshold}}(M)$), iff when fed the graph of $f$, $M$ outputs infinitely many ordered pairs $<t_k, I_k>$ such that $|I_k \cap \text{WRONG}_f| < t_k < |I_k \cap \text{GOOD}_f|$ a.e. (k).

$\text{BC}_{\text{threshold}} = \{U \mid \text{there exists an IIM } M \text{ such that } U \subseteq \text{BC}_{\text{threshold}}(M)\}$.

**Lemma 4.1:** $\text{BC}_{\text{threshold}} = \text{BC}$.

The intuition behind the proof of Lemma 4.1 is that a "threshold-plurality" vote of the programs of the list $I_k$ can be used to identify $f$.

**Proof of Lemma 4.1:**

Let $U \subseteq \text{BC}_{\text{threshold}}(M)$. We construct $M'$ which BC-
identifies $U$. $M'$'s $k$th guess for a program for $f$ is computed as follows: When fed the values of $f$, $M'$ simulates $M$ on $f$, and obtains the $k$th ordered pair $<l_k, l'_k>$ output by $M$. $M'$ then outputs the index of the program $p_k$ which on input $z$, dovetails the computations of $\{\phi_i(x)\}$ for $i \in I_k$ until $s > l_k$ elements $\{t_1, t_2, ..., t_s\}$ of $I_k$ have been found such that all $s$ computations $\{\phi_i(x)\} | 1 \leq n \leq s$ have been completed and the values of all $s$ of these computations yield the same result $y$. Then $p_k$ outputs the value $p_k(x) = y$.

To see that $M'$ BC-identifies $U$, let $f \in U$, and let $k_0$ be large enough so that for all $k \geq k_0$, $|I_k \cap \text{WRONG}_f| < t_k < |I_k \cap \text{GOOD}_f|$. Let $k \geq k_0$. Then $p_k(x)$ converges for all inputs $z$, since $|I_k \cap \text{GOOD}_f| > t_k$, i.e. the number of correct programs for $f$ in the list $I_k$ is greater than $t_k$, so after some finite number of simulation steps at least $t_k$ values must have been computed.

Now let $S = \{i_1, i_2, ..., i_s\}$ be the elements of $I_k$ which $p_k$ finds. If $y \neq f(x)$, then all $s$ elements of $S$ are in $\text{WRONG}_f$, and $|I_k \cap \text{WRONG}_f| \geq s > t_k$, which contradicts the fact that $I_k$ is a $t_k$-threshold list for $f$. Hence $p_k(x) = f(x)$ for all $x$. □

**Definition:** Let $T_{M,f}$ be a computation tree, and $A \subseteq N$ be a set of program indices. Then $L_k = \{n \mid n$ is a node at level $k$ of $T_{M,f}\}$; $L_k(A) = \{n \in L_k \mid \text{ind}(n) \in A\}$.

**Lemma 4.2:** For all $A \subseteq N$, for all $f$, and all probabilistic IIMs $M$, if $Pr[B(A)] > p$ in the tree $T_{M,f}$, then $|L_k(A)| > p2^k$ a.e. $(k)$.

That is, the fraction of nodes at level $k$ with indices in $A$ is greater than $p$ for all but finitely many levels.

We are now ready to prove Containment 4. We first note that the special case that $M$ BC-identifies $f$ with probability $> 1/2$, then by Lemma 4.2, for all but finitely many levels of the tree, the fraction of GOOD$_f$ programs at each level is $> 1/2$. A deterministic IIM on input $f$ can construct the tree $T_{M,f}$, and output as its $k$th guess the index of a program which does a majority vote of the programs found at level $k$.

Let $U \in \text{BC}_{\text{prob}}(p)$, with $p > 1/(n+1)$. Then there is a probabilistic IIM $M$ which BC-identifies every $f \in U$ with probability $> p > 1/(n+1)$. We construct a team of $n$ deterministic IIMs such that for all $f \in U$, there is a team member which $\text{BC}_{\text{thres}}$-identifies $f$, hence BC-identifies $f$.

Consider any $f \in U$, and the tree $T_{M,f}$. Then by the definition of probabilistic BC-identification, $Pr[B(\text{GOOD}_f)] \geq p > 1/(n+1)$. Lemma 4.2 asserts that $|L_k(\text{GOOD}_f)| > 2^k/(n+1)$ a.e. $(k)$. Therefore, $|L_k(\text{WRONG}_f)| < 2^k/(n+1)$ a.e. $(k)$.

There are $n$ mutually exclusive possibilities about how $|L_k(\text{WRONG}_f)|$ behaves "in the limit".

Possibility $i$: $|L_k(\text{WRONG}_f)| < 2^i/(n+1)$ a.e. $(k)$, and $|L_k(\text{WRONG}_f)| \geq 2^k/(n+1)$ a.e. $(k)$.

We use the team of $n$ deterministic IIM's to guess which case will hold for a particular $f$. The machine whose guess is correct will $\text{BC}_{\text{thres}}$-identify $f$. If a machine $M_i$ knows roughly what the fraction of "WRONG" guesses there are at each level of the tree, it can cancel most of them by witnessing that they differ from $f$. Machine $M_i$ will search for deeper and deeper levels of the tree $T_{M,f}$ such that the fraction of WRONG guesses among those output is at least $(i-1)/(n+1)$, and then cancel these wrong guesses.

If it is also true that past some point the fraction of WRONG guesses is bounded above by $i/(n+1)$, then $M_i$ will be able to form (in the limit) sets of indices for which at least the fraction $1/(n+1)$ are correct, and strictly less than this are WRONG indices. Thus $M_i$ will be able to $\text{BC}_{\text{thres}}$-identify $f$.

Machine $M_i$

$k_{old} \leftarrow 0$. REPEAT the following loop forever:

1. Simulate $M$ on input values received from $f$, and build $T_{M,f}$.
2. DOVETAIL the computations of $\phi_{\text{ind}(z)}(i)$ for all nodes $i$ and numbers $j$, comparing the outputs of completed computations with actual values of $f$. UNTIL for some level $k > k_{old}$ there are $\geq 2^k/(n+1)$ nodes in the set $\text{CANCEL}_k$, the set of nodes at level $k$ whose indices have been observed to be in $\text{WRONG}_f$.
3. $I_k \leftarrow$ the multiset of indices of nodes in $L_k - \text{CANCEL}_k$.
4. OUTPUT the ordered pair $<2^k/(n+1), I_k>$.
5. $k_{old} \leftarrow k$. GOTO 1.
We clarify the dovetail of line 2: \textit{CANCEL}_k starts out empty. A node \( n \) at level \( k \) is placed in \textit{CANCEL}_k when for some \( z, \phi_{\text{node}}(z) \) converges \( \neq f(z) \). Thus \( \text{CANCEL}_k \) contains only elements of \( L_k(\text{WRONG}_f) \).

Let \( M \) BC-identify \( U \) with probability \( p > 1/(n+1) \), and let \( M_1, M_2, \ldots, M_n \) be defined as above. Then to prove the fourth containment we only need to prove the following lemma:

\textbf{Lemma 4.3:} \ Let \( f \in U \), and let \( |L_k(\text{WRONG}_f)| \) satisfy the \( i \)th possibility above. Then \( M_i \) BC\textsubscript{threshold} identifies \( f \).

To prove the lemma, we must show

1. \( M_i \) outputs infinitely many ordered pairs \(<2^k/(n+1), I_k>\) for some value \( k \text{old} \), the dovetail of step \( 4 \) of \( M_i \) fails to satisfy its halting condition. By assumption on \( i \), \( |L_k(\text{WRONG}_f)| \geq 2^k/(n+1) \) a.e. (\( k \)), therefore there is some \( k > k \text{old} \) with \( |L_k(\text{WRONG}_f)| \geq 2^k/(n+1) \).

To show 1, we note that the only possible way for \( M_i \) to output only finitely many pairs \(<2^k/(n+1), I_k>\) is for some value \( k \text{old} \) the dovetail of step 4 of \( M_i \) fails to satisfy its halting condition.

We prove 2: By construction we have \( |L_k \cap \text{GOOD}_f| = |L_k(\text{GOOD}_f)| - |\text{CANCEL}_k| \). By assumption on \( i \), \( |L_k(\text{GOOD}_f)| < 2^k/(n+1) \) a.e. (\( k \)), and by the dovetail halting condition, \( |\text{CANCEL}_k| \geq (i-1)2^k/(n+1) \). Thus \( |L_k \cap \text{GOOD}_f| < 2^k/(n+1) \) a.e. (\( k \)).

Finally, \( |L_k \cap \text{GOOD}_f| > 2^k/(n+1) \) a.e. (\( k \)), since no node in \( L_k(\text{GOOD}_f) \) is ever cancelled, so \( |L_k \cap \text{GOOD}_f| = |L_k(\text{GOOD}_f)| \). But \( |B(\text{GOOD}_f)| \geq p > 1/(n+1) \), and Lemma 4.2 implies that \( |L_k \cap \text{GOOD}_f| = |L_k(\text{GOOD}_f)| > 2^k/(n+1) \) a.e. (\( k \)), completing the proof of Lemma 4.3, and Containment 4.

\textbf{Proof of Containment 2:} \n
We first introduce some definitions and useful lemmas. Recall that a path in \( T_M \) may converge to a particular index. This convergence must happen at some particular node:

\textbf{Definition:} Path \( p \) converges at node \( n \) iff \( n \) is the least depth node on \( p \) for which \( \text{ind}(n) = \text{ind}(m) \) for all nodes \( m \) on \( p \) with \( d(m) \geq d(n) \).

Now define \( C_n = \{ p \mid p \) is a path in \( T_M \) and \( p \) converges at node \( n \} \).

Let \( C_{n,k} \) be the set of paths \( p \) such that

- \( p \) passes through node \( n \),
- \( M \) outputs a different index at \( \text{parent}(n) \) than at \( n \) (or \( n \) is the root),
- All nodes after \( n \) on \( p \) down to depth \( k \) have the same index as \( n \).

Intuitively, \( C_{n,k} \) is the set of paths which appear to be converging to \( \text{ind}(n) \), and appear to converge at \( n \), when we examine \( T_M \) for \( k \) levels only.

Clearly \( C(A) = \bigcup_{\text{ind}(n) \in A} C_n \). Also \( C_n = \bigcap_{k \geq 1} C_{n,k} \).

\textbf{Lemma 4.4:} \ For all nodes \( n \) and for all \( T_M \),

1. For all \( k \geq d(n) \), \( C_{n,k} \supseteq C_{n,k+1} \).
2. For all \( k \geq d(n) \), \( \Pr[C_{n,k}] \geq \Pr[C_{n}] \).
3. \( \Pr[C_{n}] = \lim_{k \to \infty} \Pr[C_{n,k}] \).
4. \( \Pr[C_{n,k}] \) is computable from the first \( k \) levels of \( T_M \).

Thus the sets \( \{ C_{n,k} \} \) are increasingly better estimates of the set \( C_n \) as \( k \) increases.

\textbf{Lemma 4.5:} \ For all \( A \subseteq N \), for all \( p \in R \) such that \( 0 \leq p \leq 1 \), if \( \Pr[C(A)] > p \), then there exists nodes \( \{ n_1, n_2, \ldots, n_k \} \) such that for all \( i \), \( \text{ind}(n_i) \in A \), and \( \Pr[\bigcup_{i=1}^{k} C_{n_i}] > p \).

So most of the paths which converge to any index in the set \( A \) converge at one of a finite collection of nodes.

A finite list \( I \) of program indices is a \textit{correct list} for \( f \) if \( I \) contains at least one element of \( \text{GOOD}_f \). The class \( \text{OEX} \) was introduced in [3] (our definition here is equivalent):

\textbf{Definition:} \( M \) \( \text{OEX} \)-identifies \( f \) (written \( f \in \text{OEX}(M) \)), iff \( M \), when fed the graph of \( f \) in any order, outputs an infinite sequence \( \{ I_k \} \) of finite lists, and there is a correct list \( L \) for \( f \) such that \( I_k = L \) a.e. (\( k \)).

\( \text{OEX} = \{ U \mid \exists M \text{ such that } U \subseteq \text{OEX}(M) \} \).
Case and Smith prove a generalization of the following lemma.

**Lemma 4.6:** [3] $\text{OEX} = \text{EX}$.

It is now clear that Containment 2 follows from:

**Lemma 4.7:** If $U \in \text{EX}_{\mathcal{P}}(p), p > 1/(n+1)$, then there exists a team $\{M_1, M_2, \ldots, M_n\}$ of deterministic IIMs such that for all $f \in U$, there is some $i$ such that $M_i$ OEX-identifies (and hence EX-identifies) $f$.

The idea behind the proof of Lemma 4.7 is that rather than look only at levels of the tree, as was done in the BC case in the proof of containment 4 above, each deterministic IIM of the team will have to scan the tree, and identify converging paths, and nodes at which this convergence occurs.

**Proof of Lemma 4.7:**

Let $M$ be the probabilistic machine which identifies $U$ with probability $p > 1/(n+1)$. We will show that if a deterministic machine has a reasonable estimate of the value $\text{Pr}[C(N)]$ (the "weight", or percentage of paths which converge to any index), then it can converge to a correct list for $f$, hence OEX-identify $f$.

The finite non-determinism of the team of $n$ machines is used in the following way: each team member guesses a different range which the weight of the converging paths may fall into. For $1 \leq i \leq n$, $M_i$ assumes that the total weight of all converging paths is in the half-open interval $(i/(n+1), (i+1)/(n+1)]$. Depending on the function $f$ chosen from $U$, the weight of converging paths will fall into one of these intervals, and the associated machine will converge to a correct list for $f$.

Let $M$ be a probabilistic IIM with $U \subseteq \text{EX}_{\mathcal{P}}(M)$, and $p > 1/(n+1)$. Then let the team $\{M_1, M_2, \ldots, M_n\}$ be the following machines:

**Machine $M_i$:**

1. On input $f|_{d}$ simulate $M$ with input $f|_{d}$, and construct $T_{k}$ = the finite tree consisting of the first $k$ levels of $T_{M,i}$.
2. FOR each node $j$ in $T_{k}$, compute $\text{Pr}[C_{j,k}]$. (That this can be done is guaranteed by Lemma 4.4).
3. Let $c_{k}$ be the least numbered node in $T_{k}$ such that $\sum_{j=1}^{c_{k}} \text{Pr}[C_{j,k}] \geq i/(n+1)$.
4. Output $\{\text{ind}(i) \mid 1 \leq i \leq c_{k}\}$

We show that for all $f \in U$, there is an $i$ such that $M_i$ converges to a correct list for $f$.

$\text{Pr}[C(\text{GOOD}_i)] \geq p > 1/(n+1)$ by the definition of "$M$ EX-identifies $f$ with probability $p$.”

$C(N)$ is the set of paths which converge to any index (good or bad), so clearly $C(\text{GOOD}_i) \subseteq C(N)$, and therefore, $\text{Pr}[C(N)] \geq \text{Pr}[C(\text{GOOD}_i)] > 1/(n+1)$.

Let $m = \max \{i \mid i/(n+1) < \text{Pr}[C(N)]\}$. Clearly, $1 \leq m \leq n$. We show that $M_{m}$ converges to a correct list for $f$. By definition, $m/(n+1) < \text{Pr}[C(N)] \leq (m+1)/(n+1)$. Since $\text{Pr}[C(N)] > m/(n+1)$, Lemma 4.5 gives nodes $n_{1}, \ldots, n_{v}$ with $\text{ind}(n_{i}) \in N$ for all $i$, and $\sum_{i=1}^{v} \text{Pr}[C_{n_{i}}] > m/(n+1)$. Since all nodes $j$ have $\text{ind}(j) \in N$, this implies that there exists a smallest numbered node $s$, such that $\sum_{i=1}^{s} \text{Pr}[C_{j_{i}}] > m/(n+1)$ (choosing $s \geq \max \{n_{i}\}$ will certainly satisfy the inequality).

Now for all $k \geq d(s)$, nodes $1, 2, \ldots, s$ will be in $T_{k}$, and furthermore, by Lemma 4.4

$$\sum_{j=1}^{s} \text{Pr}[C_{j,k}] \geq \sum_{j=1}^{s} \text{Pr}[C_{j}] \geq m/(n+1),$$

hence $c_{k} \leq s$ a.e. ($k$) in step 3 of $M_{m}$.

Now Lemma 4.7 follows from:

**Claim:**

1. $M_{m}$ converges to the list $I = \{\text{ind}(1), \text{ind}(2), \ldots, \text{ind}(s)\}$.
2. $I$ contains a correct program index for $f$.

(Part 1) We've already shown that $c_{k} \leq s$ a.e. ($k$).

Now, by Lemma 4.4, for all $j$, and for all $k \geq d(j)$, $\text{Pr}[C_{j,k}] \geq \text{Pr}[C_{j,k+1}]$. It follows that the sequence $\{c_{k}\}$ is nondecreasing (a.e. ($k$)), since $c_{k}$ was chosen as the smallest value satisfying $\sum_{j=1}^{c_{k}} \text{Pr}[C_{j,k}] \geq m/(n+1)$, and since the summands are non-increasing, $\{c_{k}\}$ must be non-decreasing. Since $\{c_{k}\}$ is a nondecreasing sequence of integers bounded above by $s$, it converges. Suppose that $\{c_{k}\}$ converged to a number $t < s$. Then for all sufficiently large $k$, $\sum_{j=1}^{c_{k}} \text{Pr}[C_{j,k}] \geq m/(n+1)$. This implies that $\sum_{j=1}^{c_{k}} \text{Pr}[C_{j}] \geq m/(n+1)$, since the latter is the limit of the former. This is a contradiction, since $s$
is the least integer satisfying that inequality. Therefore, \{q_1\} converges to \(a\), and the list of program indices output by \(M_m\) converges to \(I = \{ind(1), ind(2), \ldots, ind(a)\}\).

(Part 2) \(Pr[GOOD_j] > 1/(n+1)\), and \(Pr[N] \leq (m+1)/(n+1)\), therefore \(Pr[N-GOOD_j] \leq m/(n+1)\). But \(Pr[I] > m/(n+1)\), so \(I\) must contain an element of \(GOOD\). This completes the proof of the claim, Lemma 4.7, and Containment 2. \(\Box\)

5. Anomalous Hypotheses

Allowing randomization and some probability of error for identification is only one possible way to expand the classes of functions which are identifiable. Another way to relax the definition for correct identification is to allow the hypothesized programs to disagree with the function being identified on some number of arguments.

We write \(\phi \leftarrow^k f\) to indicate that \(\{x : \phi(x) \neq f(x)\} \leq k\). Similarly, \(\phi =^* f\) indicates that \(\{x : \phi(x) \neq f(x)\}\) is finite.

Definition: Let \(M\) be a deterministic IIM, and \(a \in N \cup \{\ast\}\). Then \(M\) \(EX^a\)-identifies \(f\) iff when fed the graph of \(f\) in any order, \(M\) converges to outputting the program index \(i\), and \(\phi_i =^* f\). \(EX^a = \{U \mid \exists M\) such that for every \(f \in U, M\) \(EX^a\)-identifies \(f\}\).

Case and Smith prove that

**Theorem 5.1:** [3] For all \(k \in N\), 
\[EX^{a+1} - EX^a \neq \emptyset, \text{ and } EX^a - \bigcup_{k \in N} EX^k \neq \emptyset.\]

Smith [12] defines team inference with anomalies in the natural way; for \(a \in N \cup \{\ast\}\), \(EX^a_{\text{team}}(n) = \{U \mid \exists M_1, M_2, \ldots, M_n\) such that for every \(f \in U, there is an M_i\) which \(EX^a\)-identifies \(f\}\). It is shown that for all \(a \in N \cup \{\ast\}\), and for all integers \(n \geq 1\), \(EX^a_{\text{team}}(n) \subseteq EX^a_{\text{team}}(n+1)\). Interesting tradeoffs are also given between the number of team members, the number of anomalies, and a complexity measure - the number of "mind changes" made by an IIM before converging to a correct program. Discussion of these tradeoffs are beyond the scope of this paper; the reader is encouraged to consult [12] for further details.

Let \(M\) be a probabilistic IIM, and let \(a \in N \cup \{\ast\}\). Then \(M\) \(EX^a\)-identifies \(f\) with probability \(p\) iff \(Pr[\{\text{paths in } T_{Mi,j} \text{ which correspond to a single deterministic } EX^a\text{-identification of } f\}] \geq p\).

\[EX^a_{\text{prob}}(p) = \{U \mid \exists M\) such that for all \(f \in U, M\) \(EX^a\)-identifies \(f\) with probability \(p\}\).

Similarly, we define \(EX\)-frequency identification with anomalies; if \(M\) is a deterministic IIM, and on input \(f\), \(M\) outputs the sequence of guesses \(g_1, g_2, g_3, \ldots\), then for all \(a \in N \cup \{\ast\}\),

\(M\) \(EX^a\)-identifies \(f\) with frequency \(p\) iff there exists a guess \(g_i\) such that limit_{\to \infty} \inf \{F_i(g_i)\} \geq p\), and \(\phi_i =^* f\). (Recall \(F_i(g_i)\) is the fraction of \(M\)'s guesses among the first \(k\) which are \(g_i\).)

\[EX^a_{\text{freq}}(p) = \{U \mid \exists M\) such that for all \(f \in U, M\) \(EX^a\)-identifies \(f\) with frequency \(p\}\).

We now state

**Theorem 5.2:** For all \(a \in N \cup \{\ast\}\), for all integers \(n \geq 1\), and for all \(p \in R\), if \(1/(n+1) < p \leq 1/n\), then \(EX^a_{\text{prob}}(p) = EX^a_{\text{freq}}(p) = EX^a_{\text{team}}(n)\).

Theorem 5.2 follows from straightforward modifications of the arguments proving Theorem 2.3, and involves a generalization of Lemma 4.6 which appears in [3]. The one exception is the proof of the containment \(EX^a_{\text{team}}(n) \supseteq EX^a_{\text{prob}}(p)\) when \(p > 1/(n+1)\), which requires a significantly different approach.

Identification with anomalous hypotheses has also been studied for BC-identification, and a hierarchy analogous to Theorem 5.1 is given in [3], and extended to a BC-team-anomaly hierarchy in [12]. L. Harrington has shown that BC contains the class of all partial recursive functions [3], so relaxing this criterion in any way cannot increase the classes of functions identified. However, for \(a \in N\), we define \(BC^a_{\text{prob}}(p)\) and \(BC^a_{\text{freq}}(p)\) in the natural way, and form the following

**Conjecture:** For all \(a \in N\), for all integers \(n \geq 1\), and for all \(p \in R\): if \(1/(n+1) < p \leq 1/n\), then \(BC^a_{\text{prob}}(p) = BC^a_{\text{freq}}(p) = BC^a_{\text{team}}(n)\).

6. Nondeterministic Inference Strategies: An Observation

A team of deterministic IIMs may be thought of as a single nondeterministic machine which is restricted to choosing from among \(n\) deterministic strategies. We consider unrestricted nondeterministic IIMs, and observe that this model is too powerful to be interesting. We then consider a different type of restriction, that of "reliability" [2,9], and determine
that reliable nondeterministic IIMs are no more powerful than deterministic IIMs.

A nondeterministic IIM (NIIM) is simply a deterministic IIM with a 0-1 oracle. The NIIM may query the oracle for a "nondeterministic bit" which it then receives on a special tape. Thus there is essentially no difference between a probabilistic IIM and an NIIM, except that in the latter case, the oracle isn't a coin, and there is no associated notion of probability. For a particular NIIM \( N \), and a function \( f \) as input, there is a corresponding computation tree \( T_{N,f} \), defined as for probabilistic IIMs.

The NIIM \( N \) EX-(BC-) identifies the function \( f \) iff there exists a path in \( T_{N,f} \) which corresponds to a single deterministic EX-(BC-) identification of \( f \). It is immediately clear that there is a single NIIM \( N \) which EX-(and hence BC-) identifies \( T \), the class of all total recursive functions (in fact \( N \) identifies every partial recursive function): \( N \) uses its oracle to nondeterministically generate an integer, and then guesses that integer as a program index for the function to be identified. Thus unrestricted nondeterminism is too powerful a model to be of interest.

For EX-identification, a natural restriction for IIMs is that of reliability. (Reliability is not a meaningful notion for BC-identification.) An IIM \( M \) is reliable iff for all \( f \), if \( M \) converges to some guess on input \( f \), then the guess is a program index for \( f \). Then \( M \) reliably EX-identifies \( U \) if \( M \) EX-identifies \( U \), and \( M \) is reliable. Thus we may assume that whenever \( M \) converges, its answer is correct. Reliable inference strategies have been studied in [2,9].

We are surprised to find that

**Theorem 6.1:** If \( U \) is EX-identifiable by a RNII,M, then \( U \) is EX-identifiable by a deterministic IIM.

Thus "unrestricted" nondeterministic IIMs are too powerful, and reliable nondeterministic IIMs are no more powerful than deterministic ones. This supports our view that **team inference** is the most natural notion of nondeterminism for inductive inference.

### Acknowledgements

I am extremely grateful to Dana Angluin for suggesting the problem area, for supplying much encouragement, and for a countless number of very fruitful discussions, without which this paper would not have been possible. I also thank Bob Daley for pointing out Podnieks' frequency results; and Carl Smith, for reading an early abstract, and for bringing the excellent paper [3] to my attention.

### References


