Randomization in Learning Theory*

Andris Ambainis
Computer Science Division
University of California
Berkeley, CA 94720-1776, U.S.A.
e-mail: ambainis@cs.berkeley.edu

Abstract

We consider inductive inference (learning) of recursive (computable) functions. Here, the power of probabilistic algorithms increases as the probability of success decrease. The pattern of this increase depends on the exact requirements about learning algorithm. Typically, the interval of success probabilities $]0,1]$ can be split into subintervals such that the power of algorithms with the success probabilities in the same subinterval is the same. The structure of these intervals (called probability hierarchy) is simple in some cases and very complicated in some other cases.

Finite identification is one of simplest criteria of success for inductive inference. It was the first criteria for which probabilistic identification was studied. However, it also generates one of most complicated probability hierarchy and a large part of this structure remains unknown after 19 years of study (since 1979). The structure is simpler when larger probabilities of success are required. However, when the probabilities of success decrease, it becomes more complex and even small advances require large efforts.

We take an approach different from the previous work. Instead of trying to describe the probability hierarchy by explicit formulas, we study the properties of the whole hierarchy. We show several interesting results for Popperian finite identification, a special case of finite identification.

Our main result is a decision algorithm for the probability hierarchy. This algorithm takes two probabilities as the input and answers whether the learning with these two probabilities has equal power. We also determine the ordering type of the probability hierarchy. It is ϵ_0, the ordering type of all expressions possible in the first-order arithmetic. This shows how complex the probability hierarchy is and explains why it is not feasible to describe all cutpoints in the hierarchy explicitly.

1 Introduction

Understanding the process of learning has always fascinated scientists. There are several computational theories of learning. One of oldest theories is inductive inference established by Gold[10]. This theory considers the process of learning from a viewpoint of the computability theory. Unlike other theories of learning (for example, PAC-learning[30, 15]), inductive inference does not make probabilistic assumptions about the world. However, probabilistic algorithms appear in inductive inference and the study of probabilistic inductive inference creates

*This research was done at University of Latvia, supported by Latvia Science Council Grants 93.599 and 96.0282

a lot of interesting problems with elements of both computability theory and combinatorics. In this paper, we survey some of these problems.

We start with a general introduction to inductive inference. Learning can be considered as a process of gathering an information about an unknown object, processing this information and obtaining description of the unknown object. Ideally, we would like to obtain a complete description of the object. There are several things to be specified if we want to make this model precise[3, 22]. These are:

- What is the class of objects that we consider?
- What data are available? How are these data represented to a learning algorithm?
- What is the form of description that the learning algorithm outputs (Boolean formula, program, etc.)?
- When does the learning algorithm succeeds? (For example, do we require its output to match the unknown object exactly or do we allow small differences?)

In the theory of inductive inference, objects are arbitrary recursive (computable) functions (or recursively enumerable languages). The reason is that any algorithmic behavior can be represented as a recursive (computable) function and, hence, we obtain a model that includes any learning situation. Throughout this paper, we shall only consider learning of total recursive functions. (Learning of partial functions is also studied.)

The natural data about a function f are its values $f(0)$, $f(1)$, $f(2)$, $\ldots$ and the natural representation of these data is the sequence $\langle 0, f(0)\rangle$, $\langle 1, f(1)\rangle$, $\langle 2, f(2)\rangle$, $\ldots$. The most general type of description for a computable function is a program in a universal programming language. Again, any other description can be converted to this form.

Two main criteria of success are finite identification and identification in the limit. In original Gold's model[10], the identification in the limit, we allow the learning algorithm to output several programs and require that the last program output by the algorithm should be correct. This is motivated by the fact that humans learning a complex behaviour (for example, foreign language or driving),do not obtain the correct result from the first attempt.

In the finite identification, only one program is allowed and it must be correct. This is more limited model. In both of these models, we can require the program to match the function f exactly or allow some amount of differences[4]. We shall mostly consider exact finite learning. The references to work about other criteria of success are given in section 6.

This gives us the following learning model. A learning algorithm receives the values of an unknown function f in the natural order: $\langle 0, f(0)\rangle$, $\langle 1, f(1)\rangle$, $\langle 2, f(2)\rangle$, $\ldots$ and produces a program h. The algorithm succeeds on f if the program h computes f.

We will compare the classes of functions identifiable by probabilistic algorithms with different probabilities of correct answer.

2 Definitions

Next, we introduce the formal notation and definitions used in this paper. For more background information, see [25] for recursive function (computability) theory, [26, 18] for set theory and [3, 22] for inductive inference.

A *learning machine* is an algorithmic device that reads values of a function f: $f(0)$, $f(1)$, $\ldots$. Having seen finitely many values of the function it can output a conjecture. *A conjecture* is a program in some fixed acceptable programming system[19, 25]. Only one conjecture is allowed, i.e. learning machine cannot change its conjecture later.

Definition 1 *(a) A deterministic learning machine M* finitely identifies (FIN-identifies) *a function f if, receiving f as the input, it produces a program computing function f.*

(b) M FIN-*identifies a set of functions U if it* FIN-*identifies any function $f \in U$.*

(c) A set of functions U is called FIN-identifiable *if there exists a deterministic learning machine that identifies U. The collection of all* FIN-*identifiable sets is denoted* FIN.

Definition 2 *(a) A probabilistic learning machine M $\langle p\rangle$*FIN-*identifies (*FIN-*identifies with probability p) the set of functions U if, for any function $f \in U$ the probability that M FIN-identifies f is at least p.*

*(b) The collection of all $\langle p\rangle$*FIN-*identifiable sets is denoted $\langle p\rangle$*FIN.

Team identification is another idea closely related to the probabilistic identification. A team is just a finite set of learning machines $\{M_1, M_2, \ldots, M_s\}$.

Definition 3 *(a) A team M $[r,s]$*FIN-*identifies the function f if at least r of learning machines M_1, ..., M_s* FIN-*identify f.*

*(b) The collection of all $[r,s]$*FIN-*identifiable sets is denoted $[r,s]$*FIN.

It is easy to see that $[r,s]\mathrm{FIN} \subseteq \langle \frac{r}{s}\rangle\mathrm{FIN}$. (Just choose one of machines in the team uniformly at random and simulate.) In some cases, the opposite is also true and every probabilistic machine can be simulated by a team.
The main goal of research in probabilistic inductive inference is determining how $\mathrm{FIN}\langle p\rangle$ depends on the accepting probability p. Formally, it means describing the *probability hierarchy.*

Definition 4 *The probability hierarchy for* FIN *is the set of all points p such that there is $U \in \langle p\rangle\mathrm{FIN}$ but $U \notin \langle p+\epsilon\rangle\mathrm{FIN}$ for $\epsilon > 0$.*

3 Explicit results for FIN

Probabilistic FIN-identification was first studied by Freivalds[9]. He showed that any probabilistic learning machine with the probability of correct answer above 2/3 can be replaced by an equivalent deterministic machine. He also characterized machines with probabilities of correct answer between 1/2 and 2/3.

Theorem 1 *[9]*

(a) If $p > 2/3$, then $\langle p\rangle\mathrm{FIN} = \mathrm{FIN}$.

(b) $\langle 2/3\rangle FIN \neq FIN$.

(c) If $n/(2n-1) \geq p > (n+1)/(2n+1)$, then $\langle p\rangle\mathrm{FIN} = \langle n/(2n-1)\rangle\mathrm{FIN} = [n, 2n-1]\mathrm{FIN}$.

(d) $\langle (n+1)/(2n+1)\rangle\mathrm{FIN} = \langle n/(2n-1)\rangle\mathrm{FIN}$.

It also makes sense to consider probabilistic algorithms with the probability of correct answer 1/2 and below because there are infinitely many outputs and, hence, even designing algorithm that gives the correct answer with probability ϵ (for an arbitrary small fixed $\epsilon > 0$) may be nontrivial. Here, the first results were

Theorem 2 *[31, 11, 13]*

(a) There is a set of functions U such that $U \in [2,4]\mathrm{FIN}$ but $U \in [1,2]\mathrm{FIN}$.

(b) $[1,2]\mathrm{FIN} = [3,6]\mathrm{FIN} = [5,10]\mathrm{FIN} = \ldots$ and $[2,4]\mathrm{FIN} = [4,8]\mathrm{FIN} = [6,12]\mathrm{FIN} = \ldots$.

(c) $FIN\langle 1/2\rangle = [2,4]\mathrm{FIN}$.

So, at the probability 1/2, the power of a team depends not only on the ratio of programs that must succeed but also on the number of programs in the team. Probabilistic and team learning remain equivalent if we choose team size properly. Probabilities below 1/2 were analyzed by Daley, Kalyanasundaram and Velauthapillai[8, 7]. Their results are summarized below.

Theorem 3 *[8, 7] Let $p_1 = \frac{1}{2}$, $p_2 = \frac{24}{49}$, $p_3 = \frac{20}{41}$, $p_4 = \frac{18}{37}$, $p_5 = \frac{17}{35}$, $p_6 = \frac{16}{33}$, $p_7 = \frac{15}{31}$, $p_8 = \frac{44}{91}$, $p_9 = \frac{14}{29}$, $p_{10} = \frac{68}{141}$, $p_{11} = \frac{27}{56}$ and $p_m = \frac{12m-64}{25m-134}$ for $m \geq 12$. Then, for all $i \in \{1, 2, \ldots\}$*

(a) For all $x \in]p_{i+1}, p_i]$, $\langle x\rangle\mathrm{FIN} = \langle p_i\rangle\mathrm{FIN}$, and

(b) $\langle p_i\rangle\mathrm{FIN} \neq \langle p_{i+1}\rangle\mathrm{FIN}$.

However, with probabilities getting smaller, progress became more and more difficult. The full proof of Theorem 3 was more than 100 pages long. On the other hand, it only described the situation for a small interval $[\frac{12}{25}, \frac{1}{2}]$.

4 Explicit results for PFIN

One of approaches to this situation was considering Popperian FINite identification(PFIN), a restricted version of FIN. FIN allows two types of errors on functions that are not identified by a machine. These are

1. Errors of commission. The program output by a machine M produces a value different from the value of the input function.

2. Errors of omission. The program output by M does not halt on some input.

Definition 5 *[21] A learning machine M is Popperian if it does not make errors of omission (i.e., if all conjectures on all inputs are programs computing total functions).*

Definition 6 *(a) A set of functions U is PFIN-identifiable if there is a Popperian machine M that identifies U.*

(b) PFIN denotes the collection of all PFIN-identifiable sets.

Probabilistic and team PFIN-identification are introduced similarly. It is important that the requirement about learners outputting only programs computing total recursive functions is absolute, i.e.

1. All conjectures of all machines in a PFIN-team must be programs computing total recursive functions.

2. A probabilistic PFIN-machine is not allowed to output a program which does not compute total recursive function even with a very small probability.

Daley, Kalyanasundaram and Velauthapillai [6, 5] proved counterparts of Theorems 1, 2 and 3 for PFIN. The situation for probabilities greater than or equal to 1/2 was precisely the same as for FIN, only proofs became simpler. For probabilities smaller than 1/2, two sequences of points where power of probabilistic machines changed were discovered. One started at 1/2 and converged to 4/9, another started at 4/9 and converged to 3/7.
However, even for Popperian learning, things were getting more complicated as the probabilities decreased and [5] wrote that the prospects of determining all cutpoints are bleak even for the interval $[2/5, 1/2]$.

5 From specific values to general methods

Another approach was proposed in [1]. Instead of the infeasible task of finding all cutpoints explicitly, [1] focused on studying the general properties of the whole probability structure.
The first step was describing existing diagonalization constructions (i.e. constructions proving that there is $U \in \langle p\rangle$PFIN such that $U \notin \langle p+\epsilon\rangle$PFIN for $\epsilon > 0$) in a general form.

Theorem 4 *[1, 17] Let P_{PFIN} be the probability hierarchy for* PFIN *and $p_1, \ldots, p_s \in P_{PFIN}$. Let $p \in [0,1]$. If there are $q_1 \geq 0, \ldots, q_s \geq 0$ such that*

1. $q_1 + q_2 + \ldots + q_s = p$;

2. $\frac{p}{q_i+1-p} = p_i$ for $i = 1, \ldots, s$,

then $p \in P_{PFIN}$.

This led to a conjecture that P_{PFIN} is equal to the set A defined as follows.

1. $1 \in A$

2. If $p_1, p_2, \ldots, p_s \in A$ and $p \in [0,1]$ is a number such that there exist $q_1, \ldots, q_s \in [0,1]$ satisfying

   (a) $q_1 + q_2 + \ldots + q_s = p$;

   (b) $\frac{p}{q_i+1-p} = p_i$ for $i = 1, \ldots, s$,

   then $p \in A$;

Indeed, $A = P_{PFIN}$ and the first step in proving that was observing some structural properties of this set.

Definition 7 *[26, 18] A set A is well-ordered if there is no infinite strictly increasing sequence of elements of A. A set A is well-ordered in decreasing order if there is no infinite strictly increasing sequence of elements of A.*

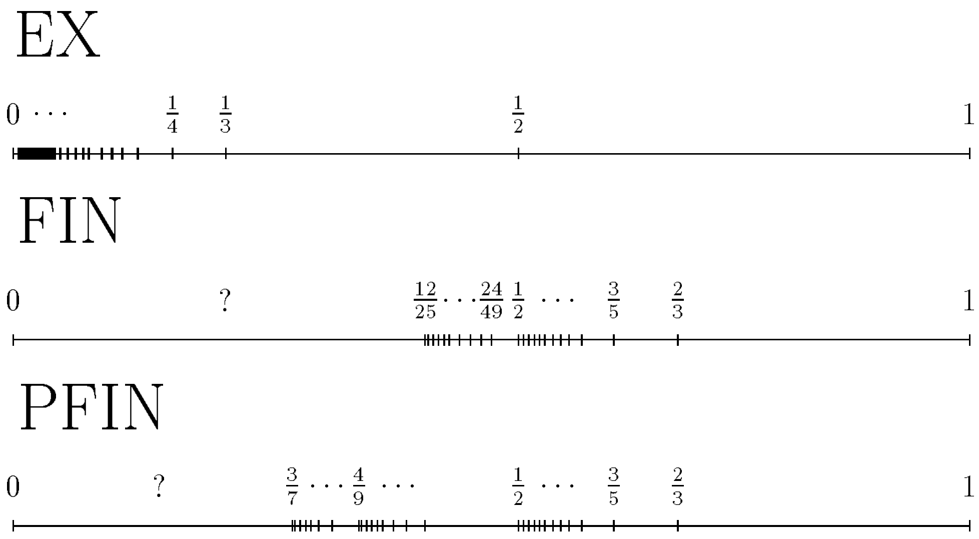

Figure 1: The probability hierarchies for EX, FIN and PFIN

Most of known probability hierarchies are well-ordered in decreasing order(see Figure 1, EX stands for learning in the limit investigated by Pitt and Smith[23, 24, 27]). It is easy to see that they all are well-ordered in decreasing order. We can also prove that the set A defined above is well-ordered.

Theorem 5 *[1] The set A is well-ordered and has a system of notation.*

A system of notations is an algorithmic description for a well-ordered set. It allows to find preceding elements, given one element. This notion was introduced by Kleene for constructive ordinals[16] and extended to sets of reals (like A) in [1]. Well-orderedness is crucial to our proof because it allows to use induction over elements of the set A. Having the system of notation is important to make this induction algorithmic. Given well-orderedness and system of notations, it is easy to construct a universal simulation algorithm.

Theorem 6 *[1] Let $p \in A$ and $p' < p$ be such that there is no $p'' \in A$ with $p' \leq p''$. Then,* $\langle p \rangle \mathrm{PFIN} = \langle p' \rangle \mathrm{PFIN}$.

Corollary 1 *[1] $A = P_{PFIN}$.*

This approach gives two other interesting results.

Theorem 7 *[1] The set P_{PFIN} is decidable, i.e. there is an algorithm that receives two probabilities p_1 and p_2 and answers whether $\langle p_1 \rangle \mathrm{PFIN} = \langle p_2 \rangle \mathrm{PFIN}$.*

Theorem 8 *[1] Let $p \in P_{PFIN}$. Then, there is an k such that $[pk, k]\mathrm{PFIN} = \langle p \rangle \mathrm{PFIN}$.*

Thus, teams of different size can have different learning power (cf. Theorem 2) but we always have the “best” team size such that team of this size can simulate any probabilistic machine (and hence, team of any other size with the same success ratio).
Finally, it is also possible to determine the precise ordering type of the probability hierarchy. The table below shows how the complexity of the ordering increases when probabilities decrease.

Interval	Ordering type of the probability hierarchy
$[\frac{1}{2}, 1]$	ω
$[\frac{4}{9}, 1]$	2ω
$[\frac{3}{7}, 1]$	3ω
$[\frac{2}{5}, 1]$	ω^2
$[\frac{3}{8}, 1]$	ω^3
$[\frac{1}{3}, 1]$	ω^ω
$[\frac{1}{4}, 1]$	ω^{ω^ω}
$[0, 1]$	ϵ_0

ω is the ordering type corresponding to a single infinite sequence (2/3, 3/5, 4/7, ...), $k\omega$ is the ordering type of a set consisting of k infinite sequences. ω^2 is the ordering type of a set consisting of infinite sequence of sequences and ω^3 is the ordering type of an infinite sequence of ω^2-type sets. ω^ω is the limit of ω, ω^2, ω^3, ... Further ordering types can be defined similarly[26, 18]. The last one, ϵ_0 is the limit of

$$\omega, \omega^\omega, \omega^{\omega^\omega}, \ldots$$

and is considered to be so big that it is hard to find any intuitive description for it[1]. This shows that the explored part of PFIN-hierarchy (the interval $[\frac{3}{7}, 1]$, the ordering type 3ω) is very simple compared to the entire hierarchy. Our result can be also considered as a partial explanation why it is unrealistic to find explicit values for all points in the probability hierarchy.

6 Conclusions and related work

A good surveys about early results in inductive inference, are [3, 22]. Since then, there has been a lot of work about probabilistic inductive inference. Most of it has been similar to section 3, describing points of probability hierarchies explicitly. Good survey papers about this work are [28, 12].

In last years, the research has gone in two directions: obtaining explicit results for new and new types of inductive inference (like language identification[14, 20]) and trying to move beyond that, to more general arguments. Research on more general arguments has concentrated on PFIN and FIN because these are most well-studied inductive inference types.

Daley and Kalyanasundaram[7] have developed an intricate machinery for obtaining explicit probability values for FIN in the interval [10/21, 1/2]. These methods may be the beginning for resolving general questions like explicit results of [5] about PFIN were the starting-point for general results described in this talk. Another piece of general work is "assymetric teams" of [2, 29].

The biggest challenge in the area is obtaining general results for unrestricted FIN. It would be good to prove more general properties about PFIN as well. (For example, how close are points of the probability hierarchy one to another?)

The probability hierarchy for probabilistic langauge learning[14] has some similarities with FIN-hierarchy and can be very interesting subject for investigation, too. However, we expect that it would be more difficult to prove general results for probabilistic language learning because it is relatively unexplored and there is less methods available for it.

[1] It is also known[26] that ϵ_0 is the ordering type of the set of all expressions possible in the first-order arithmetic but this does not look very relevant to our inductive inference result.

References

[1] A. Ambainis. Probabilistic and team PFIN-type learning: General properties. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pages 157–168. ACM Press, 1996.

[2] A. Ambainis, K. Apsītis, R. Freivalds, W. Gasarch, and C. Smith. Hierarchies of probabilistic and team FIN-learning. *Theoretical Computer Science*, 1998. To appear.

[3] D. Angluin and C. Smith. A survey of inductive inference: Theory and methods. *Computing Surveys*, 15:237–289, 1983.

[4] J. Case and C. Smith. Comparison of identification criteria for machine inductive inference. *Theoretical Computer Science*, 25:193–220, 1983.

[5] R. Daley and B. Kalyanasundaram. Use of reduction arguments in determining Popperian FIN-type learning capabilities. In K. Jantke, S. Kobayashi, E. Tomita, and T. Yokomori, editors, *Algorithmic Learning Theory: Fourth International Workshop (ALT '93)*, volume 744 of *Lecture Notes in Artificial Intelligence*, pages 173–186. Springer-Verlag, 1993.

[6] R. Daley, B. Kalyanasundaram, and M. Velauthapillai. The power of probabilism in Popperian finite learning. In *Analogical and Inductive Inference, Proceedings of the Third International Workshop*, volume 642 of *Lecture Notes in Artificial Intelligence*, pages 151–169. Springer-Verlag, 1992.

[7] Robert Daley and Bala Kalyanasundaram. FINite learning capabilities and their limits. In *Proceedings of the 10th Annual Conference on Computational Learning Theory*, pages 81–89, New York, July6–9 1997. ACM Pres.

[8] Robert Daley, Bala Kalyanasundaram, and Mahendran Velauthapillai. Breaking the probability $\frac{1}{2}$ barrier in FIN-type learning. *Journal of Computer and System Sciences*, 50(3):574–599, June 1995.

[9] R. Freivalds. Finite identification of general recursive functions by probabilistic strategies. In *Proceedings of the Conference on Fundamentals of Computation Theory*, pages 138–145. Akademie-Verlag, Berlin, 1979.

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

[11] S. Jain and A. Sharma. Language learning by a team. In M. Paterson, editor, *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, volume 443 of *Lecture Notes in Computer Science*, pages 153–166. Springer-Verlag, 1990.

[12] S. Jain and A. Sharma. On identification by teams and probabilistic machines. In K. Jantke and S. Lange, editors, *Algorithmic Learning for Knowledge-Based Systems*, volume 961 of *Lecture Notes in Artificial Intelligence*, pages 109–146. Springer-Verlag, 1995.

[13] S. Jain, A. Sharma, and M. Velauthapillai. Finite identification of function by teams with success ratio 1/2 and above. *Information and Computation*, 121:201–213, 1995.

[14] Sanjay Jain and Arun Sharma. Computational limits on team identification of languages. *Information and Computation*, 130(1):19–60, 10 October 1996.

[15] M. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory.* MIT Press, 1994.

[16] S. Kleene. Notations for ordinal numbers. *Journal of Symbolic Logic*, 3:150–155, 1938.

[17] Martin Kummer. The strength of noninclusions for teams of finite learners (extended abstract). In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory*, pages 268–277, New Brunswick, New Jersey, 12–15 July 1994. ACM Press.

[18] K. Kuratowski and A. Mostovski. *Set Theory.* North Holland, 1967.

[19] M. Machtey and P. Young. *An Introduction to the General Theory of Algorithms.* North Holland, New York, 1978.

[20] Léa Meyer. Probabilistic language learning under monotonicity constraints. *Theoretical Computer Science*, 185(1):81–128, 10 October 1997.

[21] E. Minicozzi. Some natural properties of strong identification in inductive inference. *Theoretical Computer Science*, pages 345–360, 1976.

[22] D. Osherson, M. Stob, and S. Weinstein. *Systems that Learn: An Introduction to Learning Theory for Cognitive and Computer Scientists.* MIT Press, 1986.

[23] L. Pitt. Probabilistic inductive inference. *Journal of the ACM*, 36:383–433, 1989.

[24] L. Pitt and C. Smith. Probability and plurality for aggregations of learning machines. *Information and Computation*, 77:77–92, 1988.

[25] H. Rogers. *Theory of Recursive Functions and Effective Computability.* McGraw-Hill, 1967. Reprinted by MIT Press in 1987.

[26] W. Sierpinski. *Cardinal and ordinal numbers.* PWN –Polish Scientific Publishers, 1965. Second revised edition.

[27] C. Smith. The power of pluralism for automatic program synthesis. *Journal of the ACM*, 29:1144–1165, 1982.

[28] C. Smith. Three decades of team learning. In S. Arikawa and K. Jantke, editors, *Algorithmic learning theory: Fourth International Workshop on Analogical and Inductive Inference (AII '94) and Fifth International Workshop on Algorithmic Learning Theory (ALT '94)*, volume 872 of *Lecture Notes in Artificial Intelligence*, pages 211–228. Springer-Verlag, 1994.

[29] Kalvis Apsītis. *Hierarchies of Probabilistic and Team Learning.* PhD thesis, University of Maryland, College Park, 1998.

[30] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.

[31] M. Velauthapillai. Inductive inference with bounded number of mind changes. In *Proceedings of the Second Annual Workshop on Computational Learning Theory*, pages 200–213. Morgan Kaufmann, 1989.