Characterizing the Learnability of Kolmogorov Easy Circuit Expressions*

José L. Balcázar
Universitat Politécnica de Catalunya¶

Harry Buhrman
Centrum voor Wiskunde en Informatica‖

December 19, 1995

Abstract

We show that Kolmogorov easy circuit expressions can be learned with membership queries in polynomial time if and only if every NE-predicate is E-solvable. Moreover we show that the previously known algorithm, that uses an oracle in NP, is optimal in some relativized world.

1 Results

The purpose of this short note is to present an exact complexity-theoretic characterization of a learnability problem: learning Kolmogorov-easy circuit expressions via membership queries.

Circuit expressions play essentially the role of circuits, without the limitation that the length of the input is always the same. Their construction is similar to that of regular expressions, with the difference that each boolean circuit is considered as a boolean expression. They were introduced by Watanabe and Gavaldà [WG94] as a simple variant of boolean circuits that allows for easier technical argumentations.

There are cases in which assumptions like low Kolmogorov complexity allow for learnability properties that do not hold, or are not known or not expected to hold,

*Supported by the EC through Neurocolt (Esprit working group 8556) and Coloret (HCM network CHRX–CT93–0415).

¶Dept. LSI, Edif. U, 08028, Barcelona, Spain. Work done while visiting the CWI. Partially supported by the Spanish Government, grant DGICYT PB92-0709.

‖P.O. Box 94070, 1090 GB Amsterdam, The Netherlands. Partially supported by the Dutch Foundation for Scientific Research (NWO) through NFI project ALADDIN under contract number NF 62-376.

without them [LV91]. In particular, the learnability of circuit expressions of low (logarithmic) resource-bounded (polynomial-time) Kolmogorov complexity has been studied in [BBH95], where the reader can find also an indication of what is known for the more general case of arbitrary circuit expressions (as well as formal definitions and preliminary notions used here).

This note improves upon a theorem from that reference, by weakening a complexity-theoretic hypothesis, achieving an exact characterization. More precisely, the following was proved in [BBH95]:

Theorem 1 *In the presence of an oracle in NP, Kolmogorov-easy circuit expressions can be learned from membership queries in polynomial time.*

It is natural to ask whether the implication can be reversed, assuming unrelativized learnability and trying to obtain the P=NP equality as a consequence. One of the consequences of our main theorem below is that such a proof should not relativize.

Specifically, our contribution here reduces the strength of the complexity-theoretic hypothesis needed to obtain learnability, weakening the presence of an NP oracle to the so-called hypothesis Q, introduced by Allender and Watanabe [AW90]. Hypothesis Q is short for the following statement, which is not known to hold: for each nondeterministic linear exponential time machine accepting an input, it is possible to compute deterministically, in linear exponential time, one accepting computation.

Note that Q trivially implies that E=NE; the converse is known to fail under relativization [IT89]. A similar statement at the NP level is well-known to characterize P=NP, via the computation of witnesses provided by the disjunctive self-reducibility of SATISFIABILITY (SAT), an NP-complete problem.

It turns out that Q has a number of beautiful and varied characterizations, in terms of various concepts like the bounded truth-table degrees of tally sets and the invertibility of certain honest computations; see [AW90]. We give a new one here: this is our main theorem.

Some preliminaries are needed for the proof, but for the time being we assume that the reader is familiar with the notation in [BBH95].Our proofs have some degree of missing details, in that specific names should be introduced for the constants in the logarithm and exponent in the polynomial when dealing with circuit expressions with logarithmically bounded polynomial-time Kolmogorov complexity. Such expressions will be called Kolmogorov-easy.

Theorem 2 *Kolmogorov-easy circuit expressions can be learned in polynomial time if and only if Q holds.*

Proof: It is shown in [BBH95] how to learn such expressions provided that the algorithm has access to a function satisfying the following specification: given a length n in unary, and Kolmogorov seeds of length $\log(n)$ for two circuit expressions, output, if it exists, a word of length at most n that is accepted by exactly one of the two expressions.

Now we prove that Q implies that such a function can be computed in polynomial time. First consider the following small change in the specification: n is to be given in binary. Then it is obvious how to program a linear exponential time nondeterministic machine to find the required word, by guessing and checking it. From hypothesis Q, a function satisfying the new specification can be computed in deterministic linear exponential time. A preprocessor that compresses the unary encoding of n into binary and then runs this exponential time (on the logarithmically shorter) input obtains, in polynomial time, the function needed in the algorithm in [BBH95].

It is worthwhile noting that the argument does not carry over beyond Kolmogorov-easy expressions. Also, we note for later reference that the algorithm may be programmed (at the cost of squaring the number of queries) so that the membership queries are nonadaptive. Actually, if an NP oracle is used instead of the Q hypothesis, it can be seen that all NP queries can be made before any membership queries.

To prove the converse, we show that learnability of Kolmogorov-easy circuit expressions entails Q. Consider any nondeterministic machine M working in linear exponential time on some input x. applying the standard construction [Coo71, Lev73] that proves the completeness of SAT for NP to M and x yields an exponentially long formula ϕ such that, first, it is satisfiable if and only if $M(x)$ accepts, and second, an accepting computation can be obtained with very small overhead from any satisfying assignment for ϕ. We describe how to find one deterministically in exponential time: note, beforehand, that the formula is completely described by the code of the machine and the input, which is logarithmically small, and therefore is itself a Kolmogorov-easy circuit expression.

So simply run the learner on 0^{2^n}. It will work for polynomial time on that length, which is thus linear exponential time on the length of the input to the original machine, and will make membership queries, which in our case is tantamount to evaluating the formula on some assignments: it takes again linear exponential time on the input length.

Consider for a moment what would happen if the formula is unsatisfiable. In this case the behavior of the learner is completely determined, since all the membership queries will be answered with NO. Thus it must output some unsatisfiable formula. Indeed, it must come up with a formula equivalent to the learned one, since there are Kolmogorov-easy formulas of the same length that are unsatisfiable.

Hence the learner, trying to learn a satisfiable formula, must find out in polynomial time (and a fortiori polynomially many queries) on the length of the formula, that the formula it is trying to learning is *not* unsatisfiable. Thus it must notice the difference somehow with unsatisfiable formulas, and this can be done only through a positively answered membership query. This means that, if the formula is satisfiable, then one of the membership queries that we find will be necessarily answered yes: the simulation of the learner stops there since the objective, a satisfying assignment for the formula, has been found. $\square$

This, in a sense, weakens the oracle needed to run a polynomial time learning algorithm, as indicated above. Our characterization gives us the means of exploring to what extent it can be still weakened: to learn without auxiliary oracle establishes Q.

What about the following intermediate possibility: the oracle is still there but the access to it is restricted? It is easy to see that the use of the oracle in the algorithm of [BBH95] is adaptive, since it is used to compute a function with the above indicated specification. We have already mentioned that one can learn with non-adaptive use of the membership queries to the target concept. Can we learn under non-adaptive access to the NP oracle? This is to be understood as that all the queries to the NP oracle are listed down before making them; adaptive or nonadaptive membership queries to the target concept can be made both before and after the NP queries.

As we saw above, polynomial time learnability establishes Q. This intermediate, non-adaptive case however turns out to establish that Q is equivalence with the equality E=NE.

Theorem 3 *Assume that a polynomial time learner exists for Kolmogorov-easy circuit expressions, which non-adaptively use of an oracle A in NP. Then E=NE implies Q (i.e. E=NE if and only if Q).*

Proof: Consider the learner that is assumed to exist, and assume it first makes a few sequential membership queries to the target, then prepares the list of parallel queries to NP, then makes more sequential membership queries and stops (so there is nonadaptiveness in the NP queries only). Call the "risky path" the one in which all membership queries of the first phase are answered NO.

The set A' of all pairs $\langle 0^n, x\rangle$ such that $x \in A'$ and x is (non-adaptively) queried to its NP oracle somewhere in the risky path of the learner on input 0^n forms a set in NP too. Moreover, all such pairs have logarithmic Kolmogorov complexity, because n and the index of x in the list of queries to the NP oracle suffice to describe them. (Non-adaptiveness is crucial here, as is the restriction to the risky path.) By standard complexity-theoretic arguments, A' can be encoded into a tally set T in NP, which is in P under the hypothesis E=NE.

Now run the learner on a specific CNF K-easy formula as in theorem 2. If a solution is found on the first phase, then stop; else, we are in the risky path and can use the set T to go on and complete the construction of a satisfying assignment, if one exists. Therefore Q holds. $\Box$

As a final comment, we note that all of the arguments used here do relativize, under the appropriate notion of relativized learning and Kolmogorov-easy circuit expressions. (Everything, including even the expressions, must be relativized, i.e. they are allowed to contain oracle gates). We get the following:

Corollary 4 *Under the appropriate notion of relativization, Kolmogorov-easy circuit expressions can be learned in polynomial time with adaptive access to an NP oracle, but cannot be learned with non-adaptive access to the oracle (and in particular under no access to the oracle).*

Proof: Impagliazzo and Tardos [IT89] have shown that there is a relativization such that E=NE but Q fails. Under such a relativization, by Theorem 3, non-adaptive access to NP does not suffice to learn Kolmogorov-easy circuit expressions. □

Acknowledgements

To Ricard Gavaldà, who kept asking questions about this and throwing to us his ideas, and to Montse Hermo for proofreading.

References

[AW90] Eric Allender and Osamu Watanabe. Kolmogorov complexity and degrees of tally sets. *Information and Computation*, 86(2):160–178, June 1990.

[BBH95] Balcázar, Buhrman, and Hermo. Learnability of kolmogorov-easy circuit expressions via queri es. In *EUROCOLT 95*, number 904 in Lecture Notes in Artificial Intelligence, pages 112–124, 1995.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proc. 3rd ACM Symposium Theory of Computing*, pages 151–158, Shaker Heights, Ohio, 1971.

[IT89] R. Impagliazzo and G. Tardos. Decision versus search problems in super-polynomial time. In *Proc. 30th IEEE Symposium on Foundations of Computer Science*, pages 222–227, 1989.

[Lev73] L. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9:115–116, 1973. in Russian.

[LV91] M. Li and P.M.B. Vitányi. Learning simple concepts under simple distributions. *SIAM Journal on Computing*, 20(5):911–935, 1991.

[WG94] O. Watanabe and R. Gavalda. Structural analysis of polynomial time query learnability. *Mathematical Systems Theory*, 27:231–256, 1994.