Learnability, Turing Jump and Measure. Research Report, August 2020.

Dariusz Kalociński
Institute of Computer Science
Polish Academy of Sciences

Abstract

This report contains part of the results obtained while searching the answer to the following and related questions: given a problem A, suppose the collection of problems X such that A is learnable relative to X has non-zero Lebesgue measure; does this imply that A is learnable? The solution follows from known results and a minor modification of a classical argument showing that if the collection of problems X such that A is computable relative to X, then A is computable. Certain properties of the jump obtained along the way might be of separate interest.

Let $\phi_0, \phi_1, \ldots$ be an effective enumeration of all partial recursive Turing functionals. Self-contained expositions of Turing functionals may be found, e.g., in [Rogers, 1987] or, in a very concise but instructive form in [**?**]. Stripping away technicalities, the basic idea behind the notion of Turing functional is simple: given a natural number n and $X \subseteq \mathbb{N}$, ϕ_n^X is a partial function computed by the nth Turing machine (assuming some fixed effective coding of Turing machines) relative to the oracle X. We also consider ϕ_n^σ, for $\sigma \in 2^{<\omega}$, where σ may be treated as an incomplete oracle, since it only contains answers for finitely many values $0, 1, \ldots, |\sigma| - 1$. If the nth Turing machine, performing a computation on input x relative to an incomplete oracle σ asks about the value $\geq |\sigma|$, we assume the computation loops and thus the value $\phi_n^\sigma(x)$ is undefined. An important observation, that will be used later, is that given a recursive set of binary words T, the set $\{x : \exists \sigma \in T \phi_n^\sigma(x) \downarrow\}$ is computably enumerable (c.e.).

The Turing jump X' of the set $X \in 2^\omega$ (we identify subsets of natural numbers with their characteristic functions) is defined by:

$$X' = \{n : \phi_n^X(n) \downarrow\}.$$

In plain words, the Turing jump of X is the set of (codes) of oracle Turing machines n such that the machine n stops on input n, the computation being performed relative to the oracle X.

We say that A is Turing reducible to B (symb. $A \leq B$) if $A = \phi_n^B$, for some n, i.e. if A is computed by some oracle Turing machine relative to B. A is learnable (relative to X), if the exists a uniformly computable (relative to X)

family of sets $\{A_s\}_{s\in\mathbb{N}}$ such that $\lim_s A_s = A$, meaning that for every $x \in \mathbb{N}$ there exists s_0 such that for every $t > s_0$, $A_t(x) = A(x)$.

Let $(2^\omega, S, \mu)$ be a probability measure space where S is the σ-algebra generated by the basic open sets $[\sigma] = \{\sigma X : X \in 2^\omega\}$, for every $\sigma \in 2^{<\omega}$.

Theorem 1 (Limit Lemma, [Shoenfield, 1971])**.** *A is learnable relative to X iff $A \leq X'$.*

We recall a classical result that relates Turing reducibility with measure. The result is due to Sacks. We reconstruct the proof since its technicalities allow to perform a modification that is necessary for our purposes.

Theorem 2 ([Rogers, 1987])**.** *If $\mu\{X : A \leq X\} > 0$ then A is computable.*

Proof. Let $A \in 2^\omega$ and suppose that $\mu\{X : A \leq X\} > 0$. Observe that $\{X : A \leq X\} = \bigcup_{e\in\mathbb{N}}\{X : \phi_e^X = A\}$. Since this set has non-zero measure, by σ-additivity of μ, there exists e, henceforth fixed, such that $\mu\{X : \phi_e^X = A\} > 0$. Let $\mathcal{A} := \{X : \phi_e^X = A\}$.

There exists a sequence $T_1, T_2, \ldots$ of finite subsets of $2^{<\omega}$ such that $\mathcal{A} \subseteq [T_n]$, for $n = 1, 2, \ldots,$ and $\lim_n \mu([T_n]) = \mu(\mathcal{A})$. Choose T to be a T_n with the property that $\mu(\mathcal{A})/\mu([T_n]) > 0.5$.

We describe an algorithm for computing A. Let $x \in \mathbb{N}$ be an input number. At each step $s = 1, 2, \ldots,$ perform s steps of all computations $\phi_e^\sigma(x)$ for every $\sigma \in S_s$, where S_s is the set of all words σ extending a word from T with $|\sigma| = s$. Let $m_0 = \mu(\{\sigma \in S_s : \phi_{e,s}^\sigma(x) = 0\})$, $m_1 = \mu(\{\sigma \in S_s : \phi_{e,s}^\sigma(x) = 1\})$. If $m_0 > 0.5$, return 0, If $m_1 > 0.5$, return 1. Otherwise, go to the step $s + 1$. This ends the description of the algorithm computing A.

It suffices to observe that for each input x at some step s we finally output an answer. This is a consequence of the fact that $\mu(\mathcal{A})/\mu([T]) > 0.5$. □

Theorem 3. *If $\mu\{X : A$ is learnable in $X\} > 0$ then A is learnable.*

Proof. Assume that $\mu\{X : A$ is learnable in $X\} > 0$. Given X such that A is learnable in X, there exists a family of uniformly computable in X sets $\{A_s\}_{s\in\mathbb{N}}$ such that $\lim_s A_s = A$. Observe that this is equivalent to the existence of a Turing functional $\psi : \mathbb{N} \times \mathbb{N} \times \mathcal{P}(2^\omega) \to \{0, 1\}$ such that $\lim_s \lambda x[\psi_s^X(x, s)] = A$. Consider an enumeration $\psi_1, \psi_2, \ldots$ of all Turing functionals from $\mathbb{N}\times\mathbb{N}\times\mathcal{P}(2^\omega)$ to $\{0, 1\}$. By σ-additivity of μ and the fact that $\mu(\{X : A$ is learnable in $X\}) > 0$, there exists e, henceforth fixed, such that $\mu\{X : \lim_s \lambda x[\psi_e^X(x, s) = A\} > 0$. Let $\mathcal{A} = \{X : \lim_s \lambda x[\psi_e^X(x, s) = A\}$. Again, by a similar argument as before, let T be a finite subset of $2^{<\omega}$ such that $\mathcal{A} \subseteq [T]$ and $\mu(\mathcal{A})/\mu([T]) > 0.5$.

To show that A is learnable, it suffices to provide an algorithm which for every input x works ad infinitum and produces a sequence $v_0, v_1, \ldots$ of binary values with the property that the limit of the sequence exists and equals $A(x)$. The following paragraph describes such an algorithm.

Let $x \in \mathbb{N}$ be an input number. The algorithm works in stages $s = 0, 1, 2, \ldots.$ At stage 0, output 0. At stage $s > 0$, let $S_s = \{\sigma : \exists\tau \in T\ (\sigma \succeq \tau \wedge |\sigma| = s)\}$. Perform computations $\psi_{e,s}^\sigma(x, s)$, for all $\sigma \in S_s$. Compute $p_0 =$

$\mu(\bigcup_{\sigma \in S_s \wedge \psi_e^\sigma(x,s)=0}[\sigma])$ and $p_1 = \mu(\bigcup_{\sigma \in S_s \wedge \psi_e^\sigma(x,s)=1}[\sigma])$. If $p_0 > 0.5$, produce 0, if $p_1 > 0.5$, produce 1. Otherwise, produce the same value as at the previous stage. This ends the description of the algorithm. $\square$

Note that the converse of Theorem 2 is obvious. Below we show that the converse of Theorem 3 holds as well. We will need the following basic property of the Turing jump.

Theorem 4 ([Rogers, 1987], 13.I(e))**.** $A \leq B \implies A' \leq B'$.

Theorem 5. *If A is learnable then $\mu\{X : A$ is learnable in $X\} > 0$.*

Proof. By the Limit Lemma, the above theorem may be restated as follows: if $A \leq 0'$ then $\mu\{X : A \leq X'\} > 0$. Let $A \leq 0'$. We observe that $\{X : A \leq X'\} = 2^\omega$. Let $X \in 2^\omega$. We have $0 \leq X$, where 0 is the Turing degree of computable sets. By Theorem 4, $0' \leq X'$, and thus by transitivity of Turing reducibility, $A \leq X'$, so X in $\{X : A \leq X'\}$. Therefore $\mu\{X : A \leq X'\} = 1 > 0$. $\square$

I attempted to obtain the proof of Theorem 3 in the following way (which turned out to be a dead end but allowed me to derive certain properties of the Turing jump): if it was the case that $\mu\{X : A \leq X'\} > 0$ implies $\lambda\{X' : A \leq X'\} > 0$, then one could simply restate Sacks's proof that would entail a very strong consequence that A is not only learnable but computable. However, this way of reasoning is flawed because we have the following.

Theorem 6. *For every measurable set $\mathcal{E} \subseteq 2^\omega$, $\mu(\mathcal{E}') = 0$.*

Proof. Let $\mathcal{E} \subseteq 2^\omega$ be measurable. Observe that $\mathcal{E}' \subseteq \{B : 0' \leq B\}$. To see this, let $E \in \mathcal{E}$. We have $0 \leq E$. By Theorem 4, $0' \leq E'$. Hence, $E' \in \{B : 0' \leq B\}$. However, by Theorem 2, $\mu\{B : 0' \leq B\} = 0$. Thus, by the monotonicity of measure, $\mu(\mathcal{E}') = 0$. $\square$

References

[Rogers, 1987] Rogers, Jr., H. (1987). *Theory of Recursive Functions and Effective Computability.* MIT Press, Cambridge, MA, USA.

[Shoenfield, 1971] Shoenfield, J. R. (1971). *Degrees of Unsolvability*, volume 2 of *North-Holland Mathematics Studies.* North-Holland.