Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 397 (2008) 70–76
Theoretical Computer Science
www.elsevier.com/locate/tcs

Quantum inductive inference by finite automata

R. Freivalds[a,*], R.F. Bonner[b]

[a] *Institute of Mathematics and Computer Science, University of Latvia, Raiņa bulvāris 29, Riga, Latvia*
[b] *Department of Mathematics and Physics, Mälardalen University, Västerås, Sweden*

Abstract

Freivalds and Smith [R. Freivalds, C.H. Smith Memory limited inductive inference machines, Springer Lecture Notes in Computer Science 621 (1992) 19–29] proved that probabilistic limited memory inductive inference machines can learn with probability 1 certain classes of total recursive functions, which cannot be learned by deterministic limited memory inductive inference machines. We introduce quantum limited memory inductive inference machines as quantum finite automata acting as inductive inference machines. These machines, we show, can learn classes of total recursive functions not learnable by any deterministic, nor even by probabilistic, limited memory inductive inference machines.

Keywords: Quantum computation; Automata; Inductive inference; Learning

1. Introduction

E.M. Gold, in a seminal paper [20], defined the notion of *identification in the limit*. This notion concerned learning by algorithmic devices now called Inductive Inference Machines (IIMs). See the survey Angluin and Smith [4]. An IIM inputs the graph of a total recursive function, an ordered pair at a time, and while doing so, outputs computer programs. Since we will only discuss the inference of total recursive functions, we may assume, without loss of generality, that the input is received by an IIM in its natural domain increasing order, $f(0), f(1), \ldots$. An IIM, on input from a function f will output a potentially infinite sequence of programs $p_0, p_1, \ldots$. The IIM *converges* if either the sequence is finite, say of length $n+1$, or there is a program p such that $p_i = p$ for all but finitely many i. In the former case we say that the IIM converges to p_n, and in the latter case, to p. In general, there is no effective way to tell when, and if, an IIM has converged.

Following Gold, one says that an IIM M *identifies* a function f *in the limit*, written $f \in EX(M)$, if, when M is given the graph of f as input, it converges to a program p that computes f. The terms *infer* and *learn* are used as synonyms for *identify*. If M identifies a function f, then some form of learning must have taken place, since, by the properties of convergence, only finitely much of the graph of f was known to M at the (unknown) point of convergence.

* Corresponding author.
E-mail addresses: rusinsf@latnet.lv (R. Freivalds), richard.bonner@mdh.se (R.F. Bonner).

0304-3975/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.02.023

Every total recursive function is of course identifiable by some IIM.

Each IIM will learn some set of recursive functions. The collection of all such sets, over the universe of effective algorithms viewed as IIMs, serves as a characterization of the learning power inherent in the Gold model. This collection is symbolically denoted by EX (for *explanation*), and it is rigorously defined by $EX = \{U \mid \exists M\,(U \subseteq EX(M))\}$. Gold [20] showed that the set of all total recursive functions is not in EX.

Probabilistic IIMs were introduced in Pitt [26] and studied further in Pitt and Smith [27]. Very roughly, a probabilistic IIM is an IIM that makes use of a fair coin. Write $f \in PrEX(M)\langle p\rangle$ if M learns f with probability at least p, $0 \leq p \leq 1$, and put $PrEX\langle p\rangle = \{U \mid \exists M\,(U \subseteq PrEX(M)\langle p\rangle)\}$. Pitt [26] showed that for $p > \frac{1}{2}$, $PrEX\langle p\rangle = EX$.

An IIM M identifies a function f *finitely*, written $f \in FIN(M)$, if, when M is given the graph of f as input, it outputs exactly one program that computes f, and then the machine stops. $FIN = \{U \mid \exists M\,(U \subseteq FIN(M))\}$ is then the collection of all sets of finitely identifiable functions.

Write $f \in PrFIN(M)\langle p\rangle$ if M is probabilistic, and the probability that M identifies f finitely is at least p, and put $PrFIN\langle p\rangle = \{U \mid \exists M\,(U \subseteq PrFIN(M)\langle p\rangle)\}$. Freivalds [14] showed that for $p > \frac{2}{3}$, $PrFIN\langle p\rangle = FIN$.

Valuable intuition about machine learning has been gained by working with Gold's model and its derivatives. See Arikawa and Mukouchi [7] for a discussion of this influence. In the next Section, we describe the variants of Gold's model that we then examine in a quantum setting. We note that quantum versions of other models of learning have been considered by other authors, in [30,31,21] for example, but of different nature than ours'.

2. Limited memory learning

The study of inference machines with limited memory began in Wiehagen [33], and was pursued by S. Arikawa and his students [5,6], by Wiehagen and Zeugmann [34], and by others. This research concluded that restricting the data available to the inference machine also reduces its learning potential.

Our models closely follow those of Freivalds and Smith [16], a conference paper later incorporated into a larger journal paper [17]. See also [19]. To insure an accurate accounting of the memory used by an IIM, we assume that the IIM cannot back up and reread an input after another one has been read. To avoid coding issues, the memory used is measured in bits, as opposed to integers.

Under these conventions, we write $U \subseteq LEX(M)$ iff there is a constant c such that for any $f \in U$, the machine M uses no more than c bits of memory, exclusive of the input, and $f \in EX(M)$. One formalization of this notion sees memory limited IIMs as Turing machines with input tape and work tape. The input tape is read only once (one way) and the work tape has only c bits of storage capacity. An equivalent formalization is to view memory limited IIMs as finite automata. The collection $\{U \mid \exists M\,(U \subseteq LEX(M)\}$, of all sets of functions inferable by limited memory inference machines, is denoted by LEX. It is clear that $LEX \subset EX$.

Natural numbers will serve as names for programs. The function computed by program i is denoted by ϕ_i, and we assume that $\phi_0, \phi_1, \ldots$ form an acceptable programming system [29]. To get a rough idea of the relative learning power of LEX-type inference, consider the set U_0 of total functions of finite support ($f \in U_0$ iff $f(x) = 0$ for all but finitely many x), and the set U_1 of self describing functions ($f \in U_1$ iff $\phi_{f(0)} = f$). These sets were introduced in Blum and Blum [10] and used in [15–18] to separate various classes of learnable sets of functions. It was in particular shown in [16] that $U_1 \in LEX$ but $U_0 \notin LEX$.

Limiting the memory available to a probabilistic IIM according to our conventions, gives rise to the class $PrLEX\langle p\rangle$. It was shown in [16], see also [17], that a probabilistic limited memory machine could learn with probability 1 when deterministic limited memory machines cannot learn: there is a class U of total recursive functions such that $U \in PrLEX\langle 1\rangle$ but $U \notin LEX$. A surprising statement, perhaps, if "with probability 1" is thought equivalent to "deterministic" (which it is not, of course).

By analogy with the above, it is straightforward to define classes $LFIN$ and $PrLFIN\langle p\rangle$ by imposing memory limitation on *finite* learning of recursive functions. The learning is now done by an IIM with finite memory, and it terminates after a finite number of steps. Such an IIM may be realized by a one-way finite automaton, which reads the values of an input function in the natural order of its natural argument, in finite time outputs a single program, and stops. Surprisingly, we found no published results on finite learning by limited memory inductive inference machines.

Our concept of finite memory finite learning is indeed limited. The values of a target function can be arbitrarily large, while our inference machine, being a finite automaton, can only distinguish between a finite number of integers.

In terms of its input-output behavior this means that the machine can essentially only replicate the currently read value of the target function, and, we require that it does this only once.

This limited concept of learning could easily be strengthened, we remark, by involving a stronger notion of automata, such as the notion of finite automata of Blum, Shub and Smale [12]. A BSS-automaton can process arbitrarily large integers but it cannot distinguish large numbers. It can store integers in a finite set of registers, and it can move integers from one register to another. When used as inductive inference machine, it could output target function values, which it had read at earlier moments. However, we observe, a BSS-automaton would still not always be able to learn deterministically what it can learn with probability 1; see [16,17].

3. Quantum finite automata

Quantum finite automata were introduced in Kondacs and Watrous [22]; another, weaker, definition was given in Moore and Crutchfield [23], the technical report version of which appeared the same year. Superficially, quantum automata appear rather similar to probabilistic automata, with unitary transition matrices in place of stochastic matrices. Their computational characteristics, however, are quite different; Wiesner and Crutchfield [35] discuss this point in depth. Also, the formal similarity diminishes if one is to allow mixed quantum states, as in Nayak [24] and Ambainis et al. [3].

With limited memory learning in mind, we only consider one-way automata, and settle for the basic Kondacs–Watrous and Moore–Crutchfield models, summarized in Brodsky and Pippenger [11]. We briefly recall the ground notions in the context of language recognition. A quantum finite automaton is then a tuple $M = (Q, q_0, \pi; \Sigma, \delta_{\sigma,\sigma\in\widetilde{\Sigma}})$, where $Q = \{1, 2, \ldots, q\}$ is a finite set of states; $q_0 \in Q$ is the initial state; $\pi = \{A, R, N\}$ is a partition of Q into accepting, rejecting, and non-halting states; Σ is a finite input alphabet, $\widetilde{\Sigma} = \Sigma \cup \{\#, \$\}$, the extra symbols marking the extent of a word $x \in \Sigma^*$ written $\tilde{x} = \#x\$$; and δ_σ are unitary $q \times q$ (transition) matrices, $\sigma \in \widetilde{\Sigma}$. The automaton M is *real* or *rational* if δ_σ have real or rational entries, respectively. Note, if all δ_σ are permutation matrices, the automaton is deterministic reversible.

When working as an 'acceptor' (or 'recognizer'), M receives words $\#x\$$, and, for every input $x \in \Sigma^*$, it produces two real numbers, $p_A(x)$ and $p_R(x)$, in the interval $[0, 1]$, the probabilities of acceptance, and of rejection, of the word x, respectively. M is then said to *accept* (or: *recognize*) a language $L \subset \Sigma^*$ with probability $p > 1/2$ provided $p_A(x) \geq p$ if $x \in L$, while $p_R(x) \geq p$ if $x \notin L$. The numbers $p_A(x)$ and $p_R(x)$ are arrived at as follows.

Working in the one-way mode, M reads $\tilde{x} = \sigma_0 \ldots \sigma_{s+1}$ letter by letter, from left to right, say. First, however, M sets its initial 'amplitude distribution' to the (column) vector $\xi_i^0 = 1$ if $i = q_0$ and $\xi_i^0 = 0$ otherwise. Upon reading $\sigma = \sigma_0$, M updates ξ^0 to $\xi = \delta_\sigma \xi^0$, whereby the vector $|\xi|^2$ of squared moduli $|\xi_i|^2$ is a distribution of probabilities 'for M to be in state i'.

If a Moore–Crutchfield 'measure-once' automaton, M now reads the remaining letters σ, each time updating its amplitude distribution by δ_σ. Much like in the case of probabilistic automata, only the final distribution $\delta_{\sigma_{s+1}} \cdots \cdots \delta_{\sigma_0} \xi^0$ is observed: it is measured with respect to the partition π. The numbers $p_A(x)$ and $p_R(x)$ are the probabilities of observing an accepting state, $i \in A$, and a rejecting state, $i \in R$, respectively. Or, if a Kondacs–Watrous 'measure-many' automaton, for every letter σ it reads, M follows up the unitary update by measurement with respect to π. With a probability $p_A^\sigma(x)$ it observes an accepting state, with a probability $p_R^\sigma(x)$ it observes a rejecting state; if it observes neither, the amplitude distribution has 'collapsed' to its normalized cut-off to the non-halting states N, and the next letter is read. The numbers $p_A(x)$ and $p_R(x)$ are now obtained by summing the respective probabilities $p_A^\sigma(x)$ and $p_R^\sigma(x)$ over all letters σ in the word $\tilde{x}$.

As in the case of probabilistic automata, surveyed in Condon [13], most of the research on quantum finite automata concerns language recognition with probability $p > \frac{1}{2}$ (hence with bounded error). The one-way probabilistic finite automata, it is well known since Rabin [28], recognize in this sense the same languages as their deterministic counterparts, the regular languages. The one-way quantum automata, however, are less powerful, see Brodsky and Pippenger [11] for an account: the measure-once quantum automata recognize exactly the languages recognized by the reversible deterministic finite automata; the measure-many quantum automata recognize more languages than this, but still strictly fewer than the deterministic finite automata do.

Could then 'going quantum' reduce the size of a recognizer? Well, Ambainis and Freivalds [1] did prove that for some languages the size of a recognizing quantum finite automaton can be exponentially smaller than that of any deterministic, or even probabilistic, finite automaton recognizing the language. However, there are other

languages, Nayak [24] and Ambainis et al. [3] showed, for which the smallest quantum finite automata, also in Nayak's generalized sense, are exponentially larger than their deterministic counterparts.

4. Quantum limited memory learning

We set out to consider Gold-type identification in the limit in a quantum setting, in the hope of finding a model, in which quantum learning would have advantages over classical learning; deterministic or probabilistic. However, it is easy to see that all functions computable by quantum computers are recursive, and hence, if unrestricted calculations are allowed, no advantages of quantum learning can be proved. This is why we impose (severe) memory limitations: our quantum IIMs are essentially one-way quantum finite automata. These automata, however, as we recalled above, are strictly less powerful than the deterministic finite automata as language recognizers. Does this mean that they must also be less powerful as limited memory inference machines? No.

Towards substantiating our claim, we start by sketching the workings of a finite quantum automaton M, when serving as a quantum limited memory IIM. It is only natural to admit M of the most general kind, as in Nayak [24] or Wiesner and Crutchfield [35], although, it turns out, already the simplest Moore–Crutchfield [23] model is here of interest.

Whatever its type, M works one-way, reading the consecutive values $\sigma = f(n)$ of a total recursive function f at the input, in the natural order of the argument $n = 0, 1, \ldots$. In doing this, M recognizes σ only modulo a finite partition $[\sigma]$ of the natural numbers; for example, M may recognize the first ν values $0, 1, \ldots, \nu - 1$, and "a larger integer" $L = [\sigma]$, $\sigma \geq \nu$. For every value σ it reads, M updates its amplitude distribution, or its density matrix, by a suitable transformation $\delta_{[\sigma]}$. At certain times n, the update is followed by a measurement; if a halting state is observed, the current value $f(n)$ is output, and M stops.

With the natural numbers serving as names for programs, write ϕ_i for the function computed by program i, and assume $\phi_1, \phi_2, \ldots$ forms an acceptable programming sequence. In case M's output value o computes the input function, $f = \phi_o$, the function f has been identified by M. The probability that M identifies f is then the probability of observing a halting state when $f = \phi_{f(n)}$. Write $f \in QLFIN(M)\langle p\rangle$ if M identifies f with probability at least p, $0 \leq p \leq 1$, and put $QLFIN\langle p\rangle = \{U | \exists M\,(U \subseteq QLFIN(M)\langle p\rangle)\}$. The notation is ambiguous in that it does not spell out the type of quantum automaton admitted as M, but we do not insist it should.

It is often convenient to write a total function on the natural numbers as a string of its consecutive values. Recall, for strings α, β, γ over an alphabet A, with α, β of finite length, and for natural m, n, one interprets $\alpha^m\beta^n\gamma$ in the usual sense of concatenation, with $\alpha^{p+1} = \alpha^p\alpha$, and α^0 taken as the empty string; one interprets α^∞ as the string of α's recurring indefinitely.

Let $\langle\cdot,\cdot\rangle$ denote the standard pairing for one-to-one correspondence between pairs of natural numbers and natural numbers. The mutual recursion theorem of Smullyan, see [29], asserts, for any pair g, h of total recursive functions, the existence of programs m and n, which compute the same functions as the programs $g(\langle m, n\rangle)$ and $h(\langle m, n\rangle)$, respectively, do: $\phi_m = \phi_{g(\langle m,n\rangle)}$ and $\phi_n = \phi_{h(\langle m,n\rangle)}$. We require this theorem to produce functions in the classes, which we now define. Throughout, the letter ϵ will stand for any real number in the interval $(0, 1)$.

Definition 4.1. Let V_ϵ be the set of total functions $f = 2\alpha rs2^\infty$, where α is a string of u zeros and v ones, and r, s are integers greater than 2, all so chosen that $\cos^2 u + \sin^2 v < \epsilon$ and $f = \phi_r$, or $\cos^2 v + \sin^2 u < \epsilon$ and $f = \phi_s$.

The class V_ϵ turns out not finitely learnable with limited memory classically, neither in deterministic nor probabilistic sense, but it is learnable in a quantum sense. We explain these statements in three theorems.

Theorem 4.1. $V_\epsilon \notin LFIN$.

Proof. Suppose, to the contrary, that $V_\epsilon \subset LFIN(M)$, let w be the number of states of the finite automaton M serving as our inductive inference machine, and write c for the factorial of w. When processing a sequence of consecutive zeros (or a sequence of ones) in its input, M eventually starts repeating its internal states with period not exceeding w.

Let p be the least number of initial zeros in a sequence of zeros, after reading which M starts repeating states regardless of its starting state; let q be the corresponding number of ones. M is thus unable to remember the presence or absence of any block of zeros, or of ones, the length of which is a multiple of c, and which follows a string 0^p or 1^q, respectively. M cannot therefore distinguish between input functions differing by such blocks only. It remains to show that such a pair of distinct functions does exist in V_ϵ.

The sequence of natural multiples of an irrational number is uniformly distributed modulo 1, and the number π is irrational, see e.g. Thm. 6.3 and Cor. 2.6 in [25]. It follows that for all integer $m_0, n_0, c \neq 0$, the double sequence $\cos^2(m_0 + mc) + \sin^2(n_0 + nc)$, $m, n = 1, 2, \ldots$, is dense in the interval $[0, \epsilon]$. In particular, each inequality $\cos^2(m_0 + mc) + \sin^2(n_0 + nc) < \epsilon$ has infinitely many solutions; let (m, n) be a solution for $(m_0, n_0) = (p, q)$, and let $(m', n') \neq (n, m)$ be a solution for $(m_0, n_0) = (q, p)$.

Put now $f = 20^{p+mc}1^{q+nc}rr'2^\infty$ and $f' = 20^{p+n'c}1^{q+m'c}rr'2^\infty$, choosing r and r' by Smullyan's theorem so that $f = \phi_r$ and $f' = \phi_{r'}$. The functions f and f' are distinct, both are in V_ϵ, and they differ by blocks of zeros following 0^p, or of ones following 1^q, the lengths of which are multiples of c. □

Theorem 4.2. *$V_\epsilon \notin PrLFIN\langle p\rangle$ if $p > \frac{1}{2}$.*

Proof. Suppose, to the contrary, that there is a $\delta > 0$ and a probabilistic finite automaton M, which identifies every $f \in V_\epsilon$ finitely with probability greater than $\frac{1}{2} + \delta$, and hence misidentifies f with probability less than $\frac{1}{2} - \delta$. Write the internal states of M as $1, 2, \ldots, s$.

Suppose the functions $f = 2\alpha rr'2^\infty = \phi_r$ and $f' = 2\alpha' rr'2^\infty = \phi_{r'}$ are in V_ϵ. Adapting an argument of Rabin [28], consider the probabilities ξ_i for M to enter state i after processing the fragment 2α of f, and consider the probabilities ξ'_i for M to enter state i after processing the fragment $2\alpha'$ of f'. Also, let ψ_i denote the probabilities for M to output r if M starts in state i and processes the final fragment rr' of a target function.

The probability to output r when processing f then equals $\xi_1\psi_1 + \cdots + \xi_s\psi_s$, which is greater than $\frac{1}{2} + \delta$, the value r being correct. The probability to output r when processing f' equals $\xi'_1\psi_1 + \cdots + \xi'_s\psi_s$, which is less than $\frac{1}{2} - \delta$, the value r now being incorrect. By subtraction, $(\xi_1 - \xi'_1)\psi_1 + \cdots + (\xi_s - \xi'_s)\psi_s > 2\delta$, hence $|\xi_1 - \xi'_1| + \cdots + |\xi_s - \xi'_s| > 2\delta$. Thus, if f and f' are distinguishable by M then the s-dimensional vectors of probabilities ξ and ξ' corresponding to the initial fragments $2\,\alpha$ and $2\,\alpha'$, are δ–separated in the sense at hand.

However, reasoning as in the proof of Theorem 4.1, one may for any natural c choose $\alpha = 0^{mc}1^{nc}$ and $\alpha' = 0^{n'c}1^{m'c}$, with integers $(m', n') \neq (n, m)$ arbitrarily large. Now, it is a basic fact that for every s there is a natural μ, such that for any $s \times s$ stochastic matrix δ, the sequence of powers $(\delta^\mu)^m$, $m = 1, 2, \ldots$, converges. Put $c = \mu$, let δ_σ be the stochastic matrix by which M transits when reading input value $\sigma = 0, 1, 2$, and let ξ^0 be M's initial probability distribution. Since $\xi = (\delta_1^\mu)^n(\delta_0^\mu)^m\delta_2\xi^0$ and $\xi' = (\delta_1^\mu)^{m'}(\delta_0^\mu)^{n'}\delta_2\xi^0$, it is clear that some pair $\xi \neq \xi'$ of such vectors is not δ–separated. Hence, some pair of functions $f \neq f'$ in V_ϵ is indistinguishable for M. □

Theorem 4.3. *$V_\epsilon \in QLFIN\,\langle 1 - \epsilon/2\,\rangle$.*

Proof. We construct a one-way measure-once real quantum automaton M with ten states, which finitely learns every function in V_ϵ with probability at least $1 - \epsilon/2$. Our automaton has four non-halting non-output states $\{1, 2, 3, 4\}$ (with 1 as the initial state), four halting output states $\{6, 7, 9, 10\}$ (when the current input value is output), and two additional non-halting non-output states $\{5, 8\}$ used only at the very end of the inference.

The states are thought ordered by the usual ordering of the integers, and, for brevity of account, amplitude distributions on subsets of states are identified with their extension by zero to all states. Write u for the number of zero-values of $f = 2\,\alpha\,r\,s\,2^\infty$ in the class V_ϵ, and write v for the number of its one-values.

Suppose, with f as input, we can make M behave as follows. The initial amplitude distribution is 1 at the state 1. When the first value "two" is read, the distribution becomes $\frac{1}{\sqrt{2}}(1, 1)$ at the states $\{1, 3\}$. When u values "zero" and v values "one" have been read, the distribution becomes $\frac{1}{\sqrt{2}}(\cos u, \sin u, \cos v, \sin v)$ at $\{1, 2, 3, 4\}$. When the first input "larger than two" is read, the latter distribution shifts to $\{5, 6, 7, 8\}$, and the state is measured with respect to the partition $\{\{6, 7\}, \{5, 8\}\}$. If $\{6, 7\}$ is observed, the current input value r is output. Otherwise, no output is produced, the second input "larger than two" is read, and the normalized distribution $(\cos u, \sin v)$ at $\{5, 8\}$ moves to $\{9, 10\}$, with output of the current input value s, and the termination of inference.

The value r is then output with probability $p_r = \frac{1}{2}(\cos^2 v + \sin^2 u)$, and the value s is output with probability $p_s = \frac{1}{2}(\cos^2 u + \sin^2 v)$, $p_r + p_s = 1$. If $f = 2\,\alpha\,r\,s\,2^\infty \in V_\epsilon$, the output r is correct for f iff $\cos^2 u + \sin^2 v < \epsilon$, in which case $p_s < \epsilon/2$ and $p_r = 1 - p_s > 1 - \epsilon/2$, and the output s is correct for f iff $\cos^2 v + \sin^2 u < \epsilon$, in which case $p_r < \epsilon/2$ and $p_s > 1 - \epsilon/2$. In either case, M identifies f with probability at least $1 - \epsilon/2$.

It remains to exhibit orthogonal matrices δ_σ, $\sigma = 2, 0, 1, L$, with L standing for "larger than two", which make M behave as supposed. Somewhat informally, the matrix δ_2 'turns' the amplitude distribution by the angle $\pi/4$ in the

variables {1, 3}, while the matrices δ_0 and δ_1 turn it by the angle 1 in the variables {1, 2} and {3, 4}, respectively. Thus, δ_0 and δ_1 'count' the occurrence of 0 and 1 in the input, each by modulo 2π adding on the unit angle, with δ_2 having first set both counters to zero.

To write the δ_σ formally, let, for any pair A, B of square matrices, $A \oplus B$ denote the least square matrix with A and B as sub-matrices ('blocks') along its dexter diagonal downwards, and all remaining entries zero. Write I_n for the $n \times n$ identity matrix, write ρ_θ for the matrix of counter-clockwise rotation by angle θ in the real plane, and write γ for the matrix $\rho_{\pi/4} \oplus I_1$ with the last two columns interchanged. Put now $\delta_2 = \gamma \oplus I_7$, $\delta_0 = \rho_1 \oplus I_8$, $\delta_1 = I_2 \oplus \rho_1 \oplus I_6$, and let δ_L effectuate a suitable permutation of the ten states. □

5. Conclusions

Our paper relates both to learning theory and to quantum computation. We do not pretend to have discovered new effective machine learning algorithms performed by quantum computers. We were interested in theoretical capabilities and limitations of various learning models. It turns out that there are learning problems for which quantum algorithms have advantages over classical (deterministic or probabilistic) ones. Such advantages have already been discovered in papers on quantum computation [32,1,2].

However, quantum learning differs rather much from quantum computation. While quantum finite automata can recognize only languages recognizable by deterministic finite automata [22], we have here shown that there exist classes learnable by quantum finite automata, which are not learnable by deterministic finite automata.

It may seem that our Theorem 4.2 relies on usage of quantum finite automata with very special parameters. What happens if a practical implementation of such an automaton has a slight error in the parameters? A more careful analysis shows that our quantum automata almost always have advantages over their deterministic counterparts. Indeed, for Theorem 4.2 it is only essential that the angle used in our matrices for the input symbols 0 and 1, here chosen as 1, is irrational with respect to π. This is however true for nearly all possible angles: the uniform probability to have such an angle is 1, as the probability of an angle which is not irrational with respect to π equals 0.

Acknowledgment

The first author's research was supported by Grant No. 05.1528 from the Latvian Council of Science, Contract IST-1999-11234 (QAIP) from the European Commission, and the Swedish Institute, Project ML2000.

References

[1] A. Ambainis, R. Freivalds, 1-way quantum finite automata: strengths, weaknesses and generalizations, in: Proc. 39th Annual IEEE Symposium on Foundations of Computer Science, 1998, pp. 332–341.
[2] A. Ambainis, R.F. Bonner, R. Freivalds, A. Ķikusts, Probabilities to accept languages by quantum finite automata, Springer Lecture Notes in Computer Science 1627 (1999) 174–183.
[3] A. Ambainis, A. Nayak, A. Ta-Shma, U. Vazirani, Dense quantum coding and quantum finite automata, Journal of the ACM 49 (4) (2002) 496–511.
[4] D. Angluin, C.H. Smith, Inductive inference: Theory and methods, Computing Surveys 15 (3) (1983) 237–269.
[5] S. Arikawa, M. Haraguchi, A Theory of Analogical Reasoning, Ohm-sha, 1987.
[6] S. Arikawa, T. Nishino, Computational approaches to machine learning, JIPS 32 (3) (1991-03) 217–225 (in Japanese).
[7] S. Arikawa, Y. Mukouchi, Towards a mathematical theory of machine discovery from facts, Theoretical Computer Science 137 (1995) 53–84.
[8] J. Bārzdiņš, R. Freivalds, C.H. Smith, Learning with confidence, Springer Lecture Notes in Computer Science 1045 (1996) 207–218.
[9] J. Bārzdiņš, R. Freivalds, C.H. Smith, A logic of discovery, Springer Lecture Notes in Artificial Intelligence 1532 (1998) 401–402.
[10] L. Blum, M. Blum, Toward a mathematical theory of inductive inference, Information and Control 28 (2) (1975) 125–155.
[11] A. Brodsky, N. Pippenger, Characterizations of 1-way quantum finite autiomata, SIAM Journal on Computing 31 (5) (2002) 1456–1478.
[12] L. Blum, M. Shub, S. Smale, On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines, Bulletin of the American Mathematical Society 21 (1989) 1–46.
[13] A. Condon, in: P. Pardalos, J. Reif, J. Rolim (Eds.), Bounded Error Probabilistic Finite State Automata, in: Handbook on Randomized Computing, Vol. II, Kluwer, 2001, pp. 509–532.
[14] R. Freivalds, Finite identification of general recursive functions by probabilistic strategies, in: Proc. Conf. on Algebraic, Arithmetic, and Categorical Methods in Computation Theory, Akademie-Verlag, Berlin, 1979, pp. 138–145.
[15] R. Freivalds, Inductive inference of recursive functions: Qualitative theory, Springer Lecture Notes in Computer Science 502 (1991) 77–110.
[16] R. Freivalds, C.H. Smith, Memory limited inductive inference machines, Springer Lecture Notes in Computer Science 621 (1992) 19–29.
[17] R. Freivalds, E.B. Kinber, C.H. Smith, On the impact of forgetting on learning machines, Journal of the ACM 42 (6) (1995) 1146–1168.

[18] R. Freivalds, E.B. Kinber, C.H. Smith, On the intrinsic complexity of learning, Information and Computation 123 (1) (1995) 64–71.
[19] R. Freivalds, Languages recognizable by quantum finite automata, Springer Lecture Notes in Computer Science 3845 (2006) 1–14.
[20] E.M. Gold, Language identification in the limit, Information and Control 10 (1967) 447–474.
[21] M. Hunziker, D.A. Meyer, J. Park, J. Pommersheim, M. Rothstein, The geometry of quantum learning, ArXiv e-prints. quant-ph/0309059, 2003.
[22] A. Kondacs, J. Watrous, On the power of quantum finite state automata, in: Proc. 38th IEEE Synposium on Foundations of Computer Science, 1997, pp. 66–75.
[23] C. Moore, J. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science 237 (2000) 275–306.
[24] A. Nayak, Optimal lower bounds for quantum automata and random access codes, in: Proc. 40th Annual IEEE Symposium on Foundations of Computer Science, 1999, pp. 369–376.
[25] I. Niven, Irrational Numbers, The Carus Mathematical Monographs 11, Mathematical Association of America, 1967.
[26] L. Pitt, A characterization of probabilistic inference, Journal of the ACM 36 (2) (1989) 383–433.
[27] L. Pitt, C.H. Smith, Probability and plurality for aggregations of learning machines, Information and Computation 77 (1) (1988) 77–92.
[28] M.O. Rabin, Probabilistic automata, Information and Control 6 (3) (1963) 230–245.
[29] H. Rogers Jr., Theory of Recursive Functions and Effective Computability, MIT Press, 1987.
[30] R. Servedio, Separating quantum and classical learning, in: Proc. 28th International Conference on Automata, Languages and Programming, ICALP, 2001, pp. 1065–1080.
[31] R. Servedio, S.J. Gortler, Quantum versus classical learnability, in: Proc. 16th IEEE Conference on Computational Complexity, CCC, 2001.
[32] P.W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal of Computing 26 (5) (1997) 1484–1509.
[33] R. Wiehagen, Limes–Erkennung rekursiver Funktionen durch spezielle Strategien, Journal of Information Processing and Cybernetics (EIK) 12 (1976) 93–99.
[34] R. Wiehagen, T. Zeugmann, Ignoring data may be the only way to learn efficiently, Journal of Experimental and Theoretical Artificial Intelligence 6 (1) (1994) 131–144.
[35] R. Wiesner, J.P. Crutchfield, Computation in finitary quantum processes, Santa Fe Institute Working Paper 06-09-031, 2006. Also arxiv.org/quant-ph/0608206.