ELSEVIER

Information Processing Letters 56 (1995) 15–17

Information
Processing
Letters

Efficient learning with equivalence queries of conjunctions of modulo functions [1]

Alberto Bertoni [2], Nicolò Cesa-Bianchi *, Guido Fiorino
DSI, Università di Milano, Via Comelico 39, 20135 Milano, Italy

Received 13 February 1995; revised 3 June 1995
Communicated by P.M.B. Vitányi

Abstract

A modulo m function takes value 0 when the weighted sum of the input variables is congruent to 0 mod m and takes value 1 otherwise. In this work we show a polynomial time algorithm for learning with equivalence queries the class of conjunctions of modulo p functions on n variables, for any fixed prime p. The result is also extended to learning with bounded adversarial noise.

Keywords: Learning with queries; Modulo functions; Analysis of algorithms

1. Introduction

For all positive integers m let Z_m be the class of residues modulo m. If n is a positive integer and w a vector in Z_m^n, we define the function $M_w : Z_m^n \to \{0,1\}$ by [3]

$$M_w(x) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } w \cdot x \equiv 0 \bmod m, \\ 1 & \text{otherwise.} \end{cases}$$

Let $\{M_w : w \in Z_m^n\}$ be the class of *modulo m functions* over Z_m^n.

The class of modulo 2 (parity) functions was shown to be polytime learnable with at most n equivalence queries in [6,7]. In [4] the same number of equivalence queries was proven sufficient for learning the class of modulo p functions for any prime p. In [5] this was further extended to arbitrary disjunctions of modulo p functions, leaving open the corresponding problem for conjunctions. In this paper we solve the open problem by showing a polynomial time algorithm for learning the class of conjunctions of modulo p functions for any fixed prime p.

We will work in the "learning with equivalence queries" framework (see [1,8]). A problem in this framework is parametrized by a pair $(\mathcal{T}, \mathcal{H})$, where $\mathcal{T}$ is the *target class* and $\mathcal{H}$ is the *hypothesis class*, both containing $\{0,1\}$-valued functions defined on a common domain X and such $\mathcal{T} \subseteq \mathcal{H}$. Learning proceeds in trials. On each trial $t = 1, 2, \ldots$ the learning algorithm generates a hypothesis $H_t \in \mathcal{H}$. Afterwards, an element $x_t \in X$ (called *counterexample*) such that $T(x_t) \neq H_t(x_t)$ for a target $T \in \mathcal{T}$ satisfying $T(x_\tau) \neq H_\tau(x_\tau)$ for all $\tau = 1, \ldots, t-1$ is

---

* Corresponding author. Email: cesabian@dsi.unimi.it.

[1] A preliminary version of this paper appeared in the dissertation "Apprendimento di unione di piani nel modello PAC" presented by the third author to obtain the Italian Laurea degree.

[2] Email: bertoni@hermes.dsi.unimi.it.

[3] For all tuples u, v we write $u \cdot v$ to denote the componentwise sum $\sum_i u_i v_i$.

0020-0190/95/$09.50
SSDI 0020-0190(95)00126-3

returned to the algorithm; otherwise, if no such counterexample exists, learning stops. Note that the generation of the hypothesis H_t can be interpreted as a query asking whether H_t is equivalent to an unknown and fixed target chosen from $\mathcal{T}$.

We say that an algorithm A for the problem $(\mathcal{T}, \mathcal{H})$ learns with at most q equivalence queries if q is the maximal length of any sequence of counterexamples that can be returned to A before learning stops.

2. Main result

With every system S of linear equations over a linear space X we associate the $\{0,1\}$-valued function T_S on X where $T_S(\boldsymbol{x}) = 1$ if and only if $\boldsymbol{x}$ is a solution of S. The linear system S can then be viewed as a representation for the $\{0,1\}$-valued function T_S.

Let $\mathcal{L}_p^n$ be the class of $\{0,1\}$-valued functions associated with linear systems on $\boldsymbol{Z}_p^n$. In [5] the class of disjunctions of modulo p functions was shown to be polynomial time learnable by means of an algorithm that learns the classes $\mathcal{L}_p^n$ for all $n \geqslant 1$ and primes p.

Theorem 1 (see [5]). *The class $\mathcal{L}_p^n$ is learnable with at most $n+1$ equivalence queries in time polynomial in n and p for all positive integers n and primes p.*

Let C_p^n be the class of arbitrary conjunctions of modulo p functions over $\boldsymbol{Z}_p^n$. In this section we prove that the same algorithm of Theorem 1, through a different reduction, can also learn C_p^n in time polynomial in n.

Let $\langle(\mathcal{H}_n, X_n)\rangle_{n\geqslant 1}$ and $\langle(\mathcal{H}'_n, X'_n)\rangle_{n\geqslant 1}$ be families where $\mathcal{H}_n$ and $\mathcal{H}'_n$ are classes of $\{0,1\}$-valued functions defined, respectively, on sets X_n and X'_n. Let $\langle(\sigma_n, \tau_n)\rangle_{n\geqslant 1}$ be a family of mappings $\sigma_n : \mathcal{H}_n \to \mathcal{H}'_n$ and $\tau_n : X_n \to X'_n$. We say that $\langle(\mathcal{H}_n, X_n)\rangle_{n\geqslant 1}$ is reducible to $\langle(\mathcal{H}'_n, X'_n)\rangle_{n\geqslant 1}$ via $\langle(\sigma_n, \tau_n)\rangle_{n\geqslant 1}$ whenever: (1) $H(x) = \sigma_n(H)(\tau_n(x))$ holds for all $H \in \mathcal{H}_n$ and all $x \in X_n$; (2) $\tau_n(x)$ as a function of n and x is computable in time polynomial in n.

To avoid notation, we will just say that $\mathcal{H}_n$ is reducible to $\mathcal{H}'_n$ via (σ_n, τ_n) whenever the domains X_n, X'_n and the parametrization with respect to n are clear from the context.

Let "$\circ$" denote functional composition. From the above definition we immediately get the following.

Fact 2. *If $\mathcal{H}'_n$ is polynomial time learnable with at most q equivalence queries and $\mathcal{H}_n$ is reducible to $\mathcal{H}'_n$ via (τ_n, σ_n), then $\mathcal{H}_n$ is polynomial time learnable using hypotheses of the form $H' \circ \tau_n$, $H' \in \mathcal{H}'_n$, and with at most q equivalence queries.*

Using standard number-theoretic tools, we will prove that C_p^n is reducible, in time polynomial in n, to $\mathcal{L}_p^{n^{p-1}}$. As, by Theorem 1, the latter class is polynomial time learnable, we get the following.

Theorem 3. *For every prime p, the class of conjunctions of modulo p functions over $\boldsymbol{Z}_p^n$ is learnable in time polynomial in n with at most $n^{p-1}+1$ equivalence queries.*

The proof is based on the classical Fermat little theorem.

Lemma 4 (Fermat little theorem). *If p is a prime number, then $a^{p-1} \equiv 1 \bmod p$ for any integer a not divisible by p.*

Theorem 3 can then be proven as follows.

Proof of Theorem 3. Fix a prime p. Choose a positive integer k and let the target T be an arbitrary conjunctions of k functions $M_{\boldsymbol{w}_1}, \dots, M_{\boldsymbol{w}_k}$ where $\boldsymbol{w}_i \in \boldsymbol{Z}_p^n$ for $i = 1, \dots, k$. Then, by definition of modulo p function, T is equivalent to the $\{0,1\}$-valued function associated with the system of k linear inequalities

$$\boldsymbol{w}_i \cdot \boldsymbol{x} \not\equiv 0 \bmod p, \quad i = 1, \dots, k \tag{1}$$

in n variables $\boldsymbol{x} = (x_1, \dots, x_n)$. Applying Fermat little theorem to each inequality we get the equivalent system of k equations

$$(\boldsymbol{w}_i \cdot \boldsymbol{x})^{p-1} \equiv 1 \bmod p, \quad i = 1, \dots, k. \tag{2}$$

Define the mapping $\boldsymbol{\lambda}_n : \boldsymbol{Z}_p^n \to \boldsymbol{Z}_p^{n^{p-1}}$ as follows. Let μ be a bijection from $\{1, \dots, n\}^{p-1}$ to $\{1, \dots, n^{p-1}\}$; then $\boldsymbol{\lambda}_n(x_1, \dots, x_n) = (y_1, \dots, y_{n^{p-1}})$ whenever $y_{\mu(k_1,\dots,k_{p-1})} = x_{k_1} x_{k_2} \cdots x_{k_{p-1}}$. Note that $\boldsymbol{\lambda}_n(\boldsymbol{x}) \cdot \boldsymbol{\lambda}_n(\boldsymbol{x}') = (\boldsymbol{x} \cdot \boldsymbol{x}')^{p-1}$ for all $\boldsymbol{x}, \boldsymbol{x}' \in \boldsymbol{Z}_p^n$. Moreover, $\boldsymbol{\lambda}_n(\boldsymbol{x})$ as a function of the variables n and $\boldsymbol{x}$ is computable in time polynomial in n. Hence

$$(\boldsymbol{w}_i \cdot \boldsymbol{x})^{p-1} = \sum_{j_1=1}^{n} \cdots \sum_{j_{p-1}=1}^{n} (w_{i,j_1} \cdots w_{i,j_{p-1}})(x_{j_1} \cdots x_{j_{p-1}}) = \boldsymbol{\lambda}_n(\boldsymbol{w}_i) \cdot \boldsymbol{\lambda}_n(\boldsymbol{x}). \tag{3}$$

This defines a σ_n (based on $\boldsymbol{\lambda}_n$) mapping each target $T = M_{\boldsymbol{w}_1} \wedge \cdots \wedge M_{\boldsymbol{w}_k}$, $k \geqslant 1$, to the function $\sigma_n(T)$ associated to the linear form

$$\boldsymbol{\lambda}_n(\boldsymbol{w}_i) \cdot \boldsymbol{\lambda}_n(\boldsymbol{x}) \equiv 1 \bmod p, \quad i = 1, \ldots, k. \tag{4}$$

By (1)–(4) we have, for such a σ_n, $T(\boldsymbol{x}) = \sigma_n(T)(\boldsymbol{\lambda}_n(\boldsymbol{x}))$ for all $\boldsymbol{x} \in \boldsymbol{Z}_p^n$ and all targets $T \in \mathcal{C}_p^n$. Thus $\mathcal{C}_p^n$ is polynomial time reducible to $\mathcal{L}_p^{n^{p-1}}$ via $(\sigma_n, \boldsymbol{\lambda}_n)$.

By Theorem 1, $\mathcal{L}_p^{n^{p-1}}$ is learnable in time polynomial in n with at most $n^{p-1} + 1$ counterexamples. Thus, by Fact 2, we conclude that $\mathcal{C}_p^n$ is polynomial time learnable with at most $n^{p-1} + 1$ equivalence queries. □

3. Learning with noise

Theorem 3 can be generalized to handle noise.

In [2] an extension of the model of learning with equivalence queries was introduced to take in account the presence of adversarial noise. In this extension learning *with noise rate r* stops at trial t if neither further counterexample $\boldsymbol{x}_t$ nor target $T \in \mathcal{T}$ can be found such that $H_t(\boldsymbol{x}_t) \neq T(\boldsymbol{x}_t)$ and $T(\boldsymbol{x}_\tau) \neq H_\tau(\boldsymbol{x}_\tau)$ for at least a fraction $1 - r$ of the trials τ, for $\tau = 1, \ldots, t-1$.

Thus, we say that an algorithm A learns the target class $\mathcal{T}$ with at most q equivalence queries and noise rate r if at most q counterexamples can be returned to A before learning with noise rate r stops.

In [3] an algorithm for learning with equivalence queries in presence of adversarial noise was presented. In particular, this algorithm is applicable for learning with noise rate $r < 1/(d+1)$ the class of all linear subspaces of any d-dimensional linear space V_d using at most $d/(1 - r(d+1))$ equivalence queries. Furthermore, this application is efficient whenever: (1) for all $\boldsymbol{x} \in V_d$ and all linear subspaces $T \subseteq V_d$, the membership of $\boldsymbol{x}$ in T can be tested in time polynomial in d; (2) for any finite set $S \subseteq V_d$, a linearly independent subset B spanning S can be computed in time polynomial in d and $|S|$.

Since, for all d, $\mathcal{L}_p^d$ represents the class of all linear subspaces of $\boldsymbol{Z}_p^{d+1}$ and, moreover, for this linear space conditions (1) and (2) above hold, Theorem 3 can be extended as follows.

Corollary 5. *For every prime p there is an algorithm that learns in time polynomial in n the class of conjunctions of modulo p functions over* $\boldsymbol{Z}_p^n$ *with at most* $N + 1/(1 - r(N+2))$ *equivalence queries and noise rate* $r < 1/(N+1)$, *where* $N = n^{p-1}$.

References

[1] D. Angluin, Queries and concept learning, *Machine Learning* **2** (1988) 319–342.

[2] P. Auer, On-line learning of rectangles in noisy environments, in: *Proc. 6th Ann. ACM Workshop on Computational Learning Theory* (ACM Press, 1993) 253–261.

[3] P. Auer and N. Cesa-Bianchi, On-line learning with malicious noise and the Closure Algorithm, in: *Proc. 5th Internat. Workshop on Algorithmic Learning Theory*, Lecture Notes on Artificial Intelligence **872** (Springer, Berlin, 1994) 229–247.

[4] A. Blum, P. Chalasani and J. Jackson, On learning embedded symmetric concepts, in: *Proc. 6th Ann. ACM Workshop on Computational Learning Theory* (ACM Press, 1993) 337–346.

[5] Z. Chen and S. Homer, On learning counting functions with queries, in: *Proc. 7th Ann. ACM Workshop on Computational Learning Theory* (ACM Press, 1994) 218–227.

[6] P. Fischer and H.U. Simon, On learning ring-sum-expansions, *SIAM J. Comput.* **21** (1992) 181–192.

[7] D.P. Helmbold, R. Sloan and M.K. Warmuth, Learning integer lattices, *SIAM J. Comput.* **21** (1992) 240–266.

[8] N. Littlestone, Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm, *Machine Learning* **2** (1988) 285–318.