Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Non-adaptive learning of a hidden hypergraph

Hasan Abasi [a,*], Nader H. Bshouty [a], Hanna Mazzawi [b]

[a] *Department of Computer Science, Technion, Haifa, 32000, Israel*
[b] *Google, Mountain View, United States*

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
Hypergraphs
Monotone DNF
Cover Free Family
Perfect Hash Family
Non-adaptive learning
Group Testing

ABSTRACT

We give a new deterministic algorithm that non-adaptively learns a hidden hypergraph from edge-detecting queries. All previous non-adaptive algorithms either run in exponential time or have non-optimal query complexity. We give the first polynomial time non-adaptive learning algorithm for learning hypergraphs that asks an almost optimal number of queries.

1. Introduction

Let $\mathcal{G}_{s,r}$ be a set of all labeled Sperner hypergraphs[1] of rank at most r (the maximum size of an edge $e \subseteq V$ in the hypergraph) on the set of vertices $V = \{1, 2, \ldots, n\}$ with at most s edges. Given a hidden hypergraph $G \in \mathcal{G}_{s,r}$, we need to identify it by asking *edge-detecting queries*. An edge-detecting query $Q_G(S)$, for $S \subseteq V$ is: Does S contain at least one edge of G?

In adaptive algorithms, the queries can depend on the answers to the previous queries while in the non-adaptive algorithms the queries are independent and therefore, can be asked in parallel. Our objective is to learn *non-adaptively* the hypergraph G by asking as few queries as possible.

This problem has many applications in chemical reactions, molecular biology, and genome sequencing, where deterministic non-adaptive algorithms are most desirable. In chemical reactions, we are given a set of chemicals, some of which react and some which do not. When multiple chemicals are combined in one test tube, a reaction is detectable if and only if at least one set of the chemicals in the tube reacts. The goal is to identify which sets react using as few experiments as possible. The time needed to compute which experiments to do is a secondary consideration, though it is polynomial for the algorithms we present. See [2,4–7,13,15,17–22,26–28,30,33] for more details and many other applications in molecular biology.

In all of the above applications, the rank of the hypergraph and the number of edges are much smaller than the number of vertices n. Therefore, all previous results assume that $n \geq (\max(r, s))^2$. Our results in this paper are true for all values of r, s and n.

The above hypergraph learning problem is equivalent to the problem of exact learning of a monotone DNF with at most s monomials (monotone terms), where each monomial contains at most r variables (s-term r-MDNF) from membership queries [1,7]. In this paper, we will use the latter terminology rather than the hypergraph one.

* Corresponding author.
E-mail addresses: hassan@cs.technion.ac.il (H. Abasi), bshouty@cs.technion.ac.il (N.H. Bshouty).

[1] A hypergraph is called Sperner hypergraph if no edge is a subset of another. We will show that if it is not Sperner hypergraph, then learning is not possible.

https://doi.org/10.1016/j.tcs.2017.11.019
0304-3975/

Please cite this article in press as: H. Abasi et al., Non-adaptive learning of a hidden hypergraph, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2017.11.019

The non-adaptive learnability of s-term r-MDNF was studied in [12,20,21,25,27,28,33]. Torney, [33], first introduced the problem and gave some applications in molecular biology. Gao et al., [25], gave the first explicit non-adaptive learning algorithm for s-term r-MDNF. They showed that this class can be learned using a $(n,(s,r))$-cover-free family ($(n,(s,r))$-CFF). This family is a set $A \subseteq \{0,1\}^n$ of assignments such that for every distinct $i_1,\dots,i_s,j_1,\dots,j_r \in \{1,\dots,n\}$ there is an $a \in A$ such that $a_{i_1}=\cdots=a_{i_s}=0$ and $a_{j_1}=\cdots=a_{j_r}=1$. Given such a set, the "folklore algorithm" simply takes all the monomials M of size at most r that satisfy $(\forall a\in A)(M(a)=1 \Rightarrow f(a)=1)$. The disjunction of all such monomials is equivalent to the target function. Assuming a set of $(n,(s,r))$-CFF of size N can be constructed in time T, this algorithm learns s-term r-MDNF with N membership queries in time $O(N\binom{n}{r}+T)$. Notice that, regardless of the time complexity of constructing the $(n,(s,r))$-CFF, the folklore algorithm runs in time at least $n^{\Theta(r)}$, which is non-polynomial in n for non-constant r.

In [9,21], it is shown that for $n \ge r+s$, any set $A \subset \{0,1\}^n$ that non-adaptively learns s-term r-MDNF is an $(n,(s-1,r))$-CFF. We also show that the set A is an $(n,(s,r-1))$-CFF. Therefore, the minimum size of a set that is $(n,(s-1,r))$-CFF and $(n,(s,r-1))$-CFF is a lower bound for the number of queries (and therefore, a lower bound for the time complexity) of any non-adaptively learning algorithm for s-term r-MDNF. It is known, [31], that any $(n,(s,r))$-CFF, $n \ge (\max(r,s))^2$, must have a size of at least $\Omega(N(s,r)\log n)$ where

$$N(s,r)=\frac{s+r}{\log\binom{s+r}{r}}\binom{s+r}{r}. \tag{1}$$

Therefore, for $n \ge (\max(r,s))^2$, any non-adaptive algorithm for learning s-term r-DNF must ask at least $\max(N(s-1,r), N(s,r-1))\log n = \Omega(N(s,r)\log n)$ membership queries and runs in at least $\Omega(N(s,r)n\log n)$ time.[2]

In complexity theory, a polynomial time algorithm is an algorithm that runs in polynomial time in the input size. In the exact learning model, the time complexity of learning a class C is, at least, n times the minimum membership query complexity which can be exponential in the target function size. Therefore, polynomial time learning algorithm for the class s-term r-MDNF is an algorithm that asks membership queries and runs in polynomial time in n and the minimum number of membership queries for learning the class, i.e., in $poly(N(s,r),n)$.

Before we proceed, to avoid confusion and misinterpretation of details, we will assume throughout this introduction that $n \ge (\max(r,s))^2$. In the paper, we will study the problem for any n, r and s.

To improve the query complexity of the folklore algorithm, many tried to construct $(n,(s,r))$-CFF with optimal size in polynomial time. That is, time $poly(n,N(s,r))$. Gao et al., [25], constructed an $(n,(s,r))$-CFF of size $S=(2s\log n/\log(s\log n))^{r+1}$ in time $\tilde{O}(S)$. It follows from [32] that an $(n,(s,r))$-CFF of size $O\left((sr)^{\log^* n}\log n\right)$ can be constructed in polynomial time. Almost optimal constructions of size $N(s,r)^{1+o(1)}\log n$ for $(n,(s,r))$-CFF that runs in polynomial time were given in [10–12,24]. Those constructions give almost optimal query complexity for the folklore algorithm, but still, the running time is non-polynomial for non-constant r.

Chin et al. claim in [20] that they have a polynomial time algorithm that constructs an $(n,(s,r))$-CFF of optimal size. Their analysis is misleading.[3] The size is indeed optimal but the time complexity of the construction is $O(\binom{n}{r+s})$. However, as we mentioned above, even if an $(n,(s,r))$-CFF can be constructed in polynomial time, the folklore learning algorithm still takes non-polynomial time.

Macula et al., [27,28], gave the first polynomial time randomized non-adaptive learning algorithm. They gave several randomized non-adaptive algorithms that are not optimal in the number of queries but use a different learning algorithm that runs in polynomial time. They show that for every s-term r-MDNF f and every monomial M in f there is an assignment a in a $(n,(s-1,r))$-CFF A such that $M(a)=1$ and all the other monomials of f are zero on a. To learn this monomial, they compose every assignment in A with a set of assignments that learns one monomial.

We first use the algorithm of Macula et al., [27,28], combined with the deterministic constructions of $(n,(r,s))$-CFF in [10–12,24] to change their algorithm to a deterministic non-adaptive algorithm and show that it asks $N(s,r)^{1+o(1)}\log^2 n$ queries and runs in polynomial time. The query complexity of this algorithm is almost optimal in s and r but quadratic in $\log n$. We then use a new technique, similar to the one in [16], that changes any non-adaptive learning algorithm that asks $Q(r,s,n)$ queries and runs in polynomial time to a non-adaptive learning algorithm that asks $(rs)^2\cdot Q(r,s,(rs)^2)\log n$ queries and runs in polynomial time. This reduction gives a non-adaptive learning algorithm that asks $N(s,r)^{1+o(1)}\log n$ queries and runs in $n\log n \cdot poly(N(s,r))$ time. Notice that the time complexity of this algorithm is almost linear in n compared to the folklore algorithm that runs in non-polynomial time $n^{\Theta(r)}$. Our algorithm has an almost optimal query complexity and time complexity.

The following table summarizes the results mentioned above for non-adaptive learning algorithms for s-term r-MDNF. In this table, we assume that $n \ge (\max(r,s))^2$ and $r=\omega(1)$. The result in line 4 in the table follows from the algorithm in [27,28], our analysis in this paper and the construction in [12]. In this paper, we give almost optimal algorithms for all values of n, r and s.

[2] The factor of n in the lower bound of the time complexity comes from the length n of the queries.

[3] Some parts of their construction can indeed be performed in polynomial time, but not the whole construction.

Please cite this article in press as: H. Abasi et al., Non-adaptive learning of a hidden hypergraph, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2017.11.019