Theoretical Computer Science 733 (2018) 4–43

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Exact learning from an honest teacher that answers membership queries

Nader H. Bshouty

Technion, Haifa, Israel

ARTICLE INFO

Article history:
Available online 2 May 2018

Keywords:
Exact Learning
Membership queries

ABSTRACT

Given a teacher that holds a function $f : X \to R$ from some class of functions C. The teacher can receive from the learner an element d in the domain X (a query) and returns the value of the function in d, $f(d) \in R$. The learner goal is to find f with a minimum number of queries, optimal time complexity, and optimal resources.

In this survey, we present some of the results known from the literature, different techniques used, some new problems, and open problems.

1. Introduction

Robert Dorfman's paper in 1943 introduced the field of Group Testing [112]. The motivation arose during the Second World War when the United States Public Health Service and the Selective service embarked upon a large scale project. The objective was to weed out all syphilitic men called up for induction. However, syphilis testing back then was expensive and testing every soldier individually would have been very cost heavy and inefficient. A basic breakdown of a test is: Draw sample from a given individual, perform required tests and determine the presence or absence of syphilis. Suppose we have n soldiers. Then this method of testing leads to n tests. Our goal is to achieve effective testing in a scenario where it does not make sense to test $100,000$ people to get (say) 10 positives. The feasibility of a more effective testing scheme hinges on the following property. We can combine blood samples and test a combined sample together to check if at least one soldier has syphilis [280].

Let S be the set of the n soldiers and let $I \subseteq S$ be the set of the sick soldiers. Suppose we know that the number of sick soldiers, $|I|$, is bounded by some integer d. If T is the set of soldiers for which their blood samples is combined, then the test is positive if and only if $I \cap T$ is not empty. Thus, we can regard the set of sick soldiers I as a Boolean function $f_I : 2^S \to \{0,1\}$ and the answer of the test "Is $I \cap T$ is not empty" as $f_I(T) = 1$ if and only if $I \cap T \neq \emptyset$. The goal is to identify the function f_I (and therefore the sick soldiers) from a minimal number of substitutions (tests) and optimal time. We can also identify the set of soldiers with the set $[n] := \{1, 2, \dots, n\}$ and regard each test as an assignment $a \in \{0,1\}^n$, where $a_i = 1$ if and only if the ith soldier blood is in the test. Then the set $S = \{0,1\}^n$ is the set of all possible tests. The set of sick soldiers $I \subseteq [n]$ corresponds to a Boolean function $f'_I : S \to \{0,1\}$ where $f'_I(x_1, \dots, x_n) = \bigvee_{i \in I} x_i$ and $\vee$ is the Boolean or (disjunction). So this problem is also equivalent to the problem of identifying, a hidden Boolean conjunction of up to d variables, with a minimal number of substitutions and optimal time.

E-mail address: bshouty@cs.technion.ac.il.

https://doi.org/10.1016/j.tcs.2018.04.034
0304-3975/

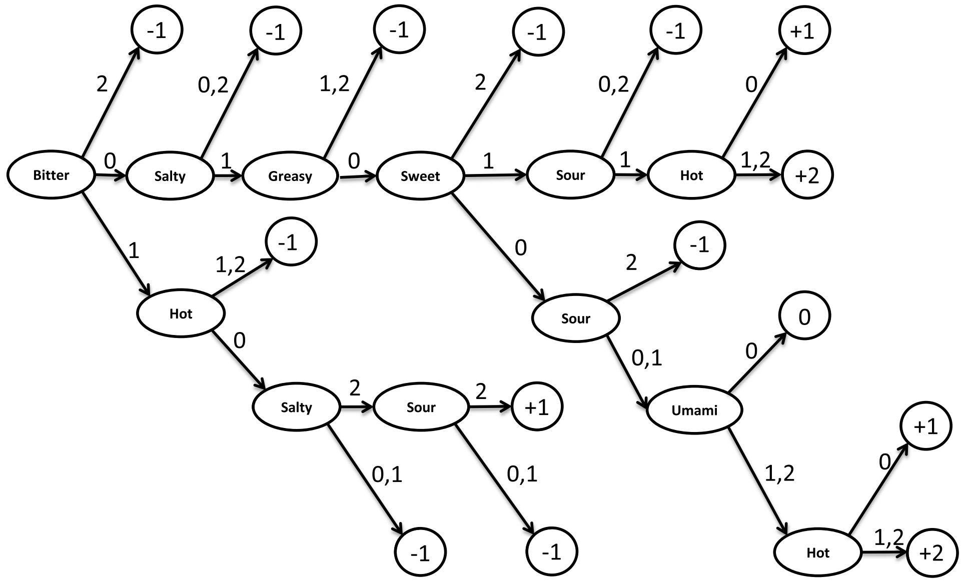

Fig. 1. A decision tree of the tastes preference of some customer. On the edges, $2, 1, 0$ is the degree of flavor, "very", "little" and "no", respectively. In the leaves, -1 stands for "not delicious", 0 for "OK", $+1$ for "delicious' and $+2$ for "very delicious".

Another interesting problem is the problem of learning decision tree with a minimal number of queries. Let's say one has a restaurant and she wants to learn each customer tastes preference in food. For every customer, she offers a sample of a meal that was never ordered by the customer before and then receives some feedback. The customer tastes preference depends on some attributes of the food. For example, "sweet", "sour", "salty", "umami", "bitter", "greasy", "hot" etc. Those are the attributes. The goal is to learn (find out) the customer tastes preference from a minimal number of samples. Each sample can be regarded as a set of attributes. The customer tastes preference is the objective function. This function depends on the attributes, and the value of the function is the customer feedback. In many cases, the target function can be described as a decision tree. See the example in Fig. 1.

In the following subsection, we give a framework to the above problems and many other similar problems.

1.1. The learning model

Let the *domain* (*instance space*) be the set $X_n \in \{X_j\}_{j\ge1}$ and the *range* be the set $R_n \in \{R_j\}_{j\ge1}$. Let C_n be a *class of representations of functions* $f: X_n \to R_n$ (*target class, concept class*). Given a *teacher* (*black box, opponent player, responder*) that holds a (*target*) function (*concept*) f from the class C_n. The *learner* (*player, questioner*) can ask the teacher *membership queries* (for Boolean functions. i.e. $R_n = \{0, 1\}$) or *substitution queries* (for non-Boolean functions), i.e., it can send the teacher an element d of the domain X_n and the teacher returns $f(d)$. The learner knows $\{C_j, X_j, R_j\}_{j\ge1}$. Our (the *learner*) ultimate goal is to write an *(exact) learning algorithm* that learns $C = \cup_{j\ge1} C_j$ with a minimum number of queries and optimal resources. That is,

1. **Input:** The learning algorithm receives the input n and has access to an *oracle* MQ_f that answers membership/substitution queries for the target function $f \in C_n$.
2. **Query complexity:** It asks the teacher a minimum number of membership/substitution queries.
3. **Exact learning:** It either learns (finds, outputs) $g \in C_n$ such that g is logically equivalent to f, $g = f$ (*proper learning*) or learns $h \in H_n \supseteq C_n$ such that $h = f$ (*non-proper learning from* H_n).
4. **Resources Complexity:** It runs in linear/polynomial/optimal time complexity, optimal space complexity, an optimal number of random bits or/and other optimal resources.

The following decision problems are also considered in the literature

1. **Equivalent test**: Given two teachers that have two functions from C_n each. Test whether the two functions are equivalent.
2. **Identity test from** H_n: Given a teacher that has a function f from C_n. Given a function $h \in H_n$. Test whether $f = h$.
3. **Zero test**: Given a teacher that has a function f from C_n. Test whether $f = 0$.