Artificial Intelligence 229 (2015) 246–257

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Learning Boolean specifications

U. Bubeck, H. Kleine Büning *

Computer Science Institute, University of Paderborn, D-33095 Paderborn, Germany

ARTICLE INFO

Article history:
Received 10 November 2014
Received in revised form 7 July 2015
Accepted 12 September 2015
Available online 16 September 2015

Keywords:
Query learning
Propositional logic

ABSTRACT

In this paper we consider an extended variant of query learning where the hidden concept is embedded in some Boolean circuit. This additional processing layer modifies query arguments and answers by fixed transformation functions which are known to the learner. For this scenario, we provide a characterization of the solution space and an ordering on it. We give a compact representation of the minimal and maximal solutions as quantified Boolean formulas and we adapt the original algorithms for exact learning of specific classes of propositional formulas.

1. Introduction

Query learning [2,10] is a classical scenario in supervised machine learning which is used to study the problem of selecting which queries to ask to learn a target concept as quickly as possible, in a scenario where information about the target concept can be obtained by making queries. In the model of *exact identification with queries* [2], the learner must figure out an unknown concept using restricted queries addressed to an oracle that constitutes the teacher. Predetermined templates for different types of queries specify what the learner can ask and how the oracle will respond. It is then up to the learner to instantiate these templates with suitable concrete questions, so that the target concept is learned with a small number of question-and-answer cycles, say, polynomially many in the size of the target in some specific representation that learner and teacher have previously agreed upon. An intuitive illustration is to think that this representation of the target is hidden in a black box. The questions to the oracle and its answers allow the learner to probe that black box.

In this paper we consider an extended variant of the exact identification model where there is an intermediary processing layer between the learner and the oracle. This additional processing layer modifies query arguments and answers by fixed transformation functions which are known to the learner. Figuratively speaking, we place the black box with the hidden target concept inside an additional white box, so that the black box can be probed only indirectly through probing the white box. We do this for the most frequently studied case of query learning in which the target concept in the black box is a Boolean function. A natural choice for the white box is then a Boolean circuit which surrounds the black box. In total, that means we have a circuit containing a node with an unknown Boolean function. Since a Boolean function can describe the computation of a Boolean circuit, this is equivalent to having a Boolean circuit that contains an unknown subcircuit (disregarding size issues of representation as formula vs. representation as circuit).

In circuit design, this type of question is a specific verification task, called *(Black Box) Combinational Equivalence Checking* [7]. A partial implementation of a circuit has its unfinished parts combined into a black box. Early design errors can be detected, if the implementation differs from its specification for all substitutions of the black box. In this case the specification of the circuit serves as the oracle. Similarly, a faulty implementation can be handled, if a fault can be restricted to a

* Corresponding author.
E-mail addresses: uwe@ub-net.de (U. Bubeck), kbcsl@uni-paderborn.de (H. Kleine Büning).

http://dx.doi.org/10.1016/j.artint.2015.09.003
0004-3702/

certain area in the implementation. Replacing this area by a black box and using the faulty implementation as the oracle, the faulty behavior might be explained. Angluin's original approach of using AI techniques, especially machine learning, has to be adapted for this more general setting of Boolean circuit design.

The learner's task is now to find a definition of the unknown component, so that substituting the learner's solution for the unknown component results in a circuit which computes for all possible inputs exactly the same outputs as the target circuit, i.e., the original circuit with revealed black box. This is analogous to exact query learning of Boolean functions [2], which requires that the learned function must return exactly the same truth values as the target function on all inputs. In this original scenario of exact query learning, the most important standard types of queries are membership queries and equivalence queries. A *membership query* asks whether a particular domain object is included in the unknown target concept. In the Boolean domain, that means asking whether a particular truth assignment makes the unknown function true. An *equivalence query* asks whether a concrete concept proposed by the learner is the same as the target concept. For Boolean functions, this holds whenever both concepts return the same valuations for all possible arguments. A negative answer to an equivalence query is always accompanied by a counterexample. In the Boolean domain, a counterexample is a truth assignment on which the proposed function and the target function disagree. Membership and equivalence queries have in common that they reason about a single or all truth assignments to the unknown function and consider the corresponding value(s) of the unknown function. We adapt this to our scenario of learning a function in a circuit as follows: instead of reasoning about truth assignments to the unknown function, we reason about truth assignments to the inputs of the whole circuit. And instead of considering the valuation of the unknown function, we consider the output of the whole circuit. That means we do not ask queries about the unknown component in isolation, but only about the whole circuit, as in the previously outlined scenario of probing the black box only indirectly through a surrounding white box.

Our paper investigates how this indirection changes the character of the original query learning problem, especially the structure of solutions and the applicability of existing learning algorithms. Clever strategies have been found for exact learning of various classes of Boolean functions with a polynomial number of membership and equivalence queries. Is it possible to adapt such strategies to our indirect query learning scenario?

These query learning strategies typically try to verify conjectures about the hidden function by making series of membership queries in which the truth assignments are chosen according to precise sequences of, e.g., bit flips or bit vector intersections. For example, if the unknown function is known to be representable as a conjunction of variable literals, and if two membership queries for identical arguments with the last bit flipped, say, $(1, 0, 1)$ and $(1, 0, 0)$, are both acknowledged with positive answers, we can deduce that the last argument cannot occur as a literal in the hidden conjunction. In the extended scenario with a circuit being placed around the unknown function, the global circuit inputs that we assign are mapped by the circuit to different actual inputs to the unknown circuit node. The problem is not only that we need to reverse this mapping in order to know which global inputs lead to the desired arguments to the unknown function. We also need to consider that the mapping from global circuit inputs to local node inputs is generally not surjective. That means certain input values might never appear at the unknown node, which allows the learned function to be defined arbitrarily for these arguments. Similarly, there can be global circuit inputs for which the actual output of the unknown node does not matter. In circuit design and optimization, the former case is called *(local) satisfiability don't care*, and the latter is called *(local) observability don't care* [6], and it is well known that this allows different possible definitions of the respective circuit node. Driven by important applications of circuit simplification or synthesis, these observations have led to the development of various techniques for computing some or all permissible functions for a node in a Boolean circuit [6,8,11]. However, these techniques do not seem to be directly applicable to our query learning problem. Obviously, their iterative computation along cut lines in the circuit is different in style from query learning. But most importantly, they do not consider the representability of the permissible functions within given classes of Boolean functions, which is the key feature of the existing query learning algorithms that we wish to adapt to the circuit scenario.

Another line of related work is a series of papers by Scholl, Becker et al. on the *partial equivalence checking problem* in circuit design [7,5], i.e., checking whether a given partial implementation of a combinational circuit can be extended to a complete design that is equivalent to a given full specification. The authors focus on detecting errors in partial implementations and not on returning implementations of black boxes. In case of a single black box, their method, which is essentially a series of different algorithms with increasing accuracy, can find all errors in the partial implementation. In a recent paper [5], Gitina et al. give a translation of the partial equivalence checking problem into a satisfiability problem for dependency quantified Boolean formulas which can be decided in Π_3^P for the special case of a single black box. Seen as a composition problem, the task in [7,5] is to find components that fit into the black boxes in a template (the surrounding white box part) such that the composition is correct with respect to the given specification. Following this approach, one could argue that a specification of the black box will be revealed implicitly. Apart from the fact that it is unknown whether a compact representation of such a specification can be given, testing candidate components against this specification is a coNP-complete problem. In our approach, we assume some knowledge about the black box to be given in form of an oracle. The question of which candidates fit into the black box of a circuit can therefore be addressed directly by a query learning approach.

The existence of multiple solutions is a significant modification of the original exact query learning problem, which always has exactly one solution. Even in the original scenario, one has to be very careful with overlapping conjectures, e.g., when a conjectured implicant contains two actual prime implicants. Our major contributions on this topic are the following: