Discrete Applied Mathematics 215 (2016) 1–13

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Totally optimal decision trees for Boolean functions

Igor Chikalov, Shahid Hussain *, Mikhail Moshkov

Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia

ARTICLE INFO

Article history:
Received 3 April 2015
Received in revised form 27 June 2016
Accepted 4 July 2016
Available online 26 July 2016

Keywords:
Boolean functions
Monotone Boolean functions
Totally optimal decision trees
Time complexity
Space complexity

ABSTRACT

We study decision trees which are totally optimal relative to different sets of complexity parameters for Boolean functions. A totally optimal tree is an optimal tree relative to each parameter from the set simultaneously. We consider the parameters characterizing both time (in the worst- and average-case) and space complexity of decision trees, i.e., depth, total path length (average depth), and number of nodes. We have created tools based on extensions of dynamic programming to study totally optimal trees. These tools are applicable to both exact and approximate decision trees, and allow us to make multi-stage optimization of decision trees relative to different parameters and to count the number of optimal trees. Based on the experimental results we have formulated the following hypotheses (and subsequently proved): for almost all Boolean functions there exist totally optimal decision trees (i) relative to the depth and number of nodes, and (ii) relative to the depth and average depth.

1. Introduction

Time and space complexity relationships for algorithms play important role in computational complexity theory. These relationships are often considered for non-universal computational models, such as branching programs or decision trees [3,20], where time and space complexity is characterized by a number and not by a function depending on the length of input. The considered relationships become trivial if there exist totally optimal algorithms—optimal with respect to both time and space complexity. Total optimality with respect to two different criteria (e.g., time and space) gives us optimal solution which can be used in many practical applications.

We study totally optimal decision trees for computing Boolean functions. We consider depth and total path length (average depth) of decision trees as time complexity in the worst- and in the average-case, respectively, and the number of nodes in decision trees as space complexity.

The research presented in this paper has four goals. First, we present all the tools (algorithms and mathematical background) to construct and optimize decision trees for Boolean functions. We perform experiments on table representation of Boolean functions, and lastly we formulate hypotheses based on the experimental results and prove them. We also present hardness results regarding optimization of decision trees corresponding to some cost functions such as depth, number of nodes, and average-depth of decision trees (see Section 4 for details.)

We have created a number of tools based on extensions of dynamic programming to study decision trees, i.e., to construct and optimize decision trees with respect to various criteria (see for example [1,12]). These tools allow us to work with both exact and approximate decision trees. We can describe the set of decision trees under consideration by a directed acyclic

* Corresponding author. Fax: +966 12 8021291.
E-mail address: shahid.hussain@kaust.edu.sa (S. Hussain).

http://dx.doi.org/10.1016/j.dam.2016.07.009

graph, make multi-stage optimization of decision trees relative to different cost functions, and count the number of optimal trees (see for example Section 5).

Dynamic programming for optimization of decision trees has been studied before by Garey [9] and others [18,22], however not in the context of multi-stage optimization. Previously, we have considered multi-stage optimization such as [2] but we did not distinguish between optimal and strictly optimal decision trees (see Section 3.3). However, the considered procedures of optimization allow us to describe the whole set of optimal trees only for number of nodes and average depth. For depth we can only obtain the set of strictly optimal decision trees for which each subtree is optimal for the corresponding subtable. Chikalov in [5] discusses in detail the average time complexity of decision trees. Moshkov in [19] presented approximate algorithms for minimization of depth of decision trees and proved bound using an approximate algorithm for set cover problem. There is another direction of research involving evaluating stochastic Boolean functions in the context of decision trees (see for example Deshpande et al. [8]). Construction of decision trees for Boolean functions in general and for special purposes is well studied. Kundakcioglu and Ünlüyurt also discuss minimum-cost decision trees (specialized and/or tree) for sequential fault diagnosis (see [17]). There are many research papers which discuss the complexity of Boolean functions in various contexts and for different parameters (see for example the surveys by Korshunov [15] and Buhrman and de Wolf [4]).

We discuss the cost functions for optimization of decision trees in detail in Section 2. It is important to note that all cost functions are increasing cost functions however, some of these cost functions are strictly increasing functions. That leads to the consideration of the two kinds of optimal trees, optimal and strictly optimal, which allow us to understand the difference between optimization (i) relative to depth, and (ii) relative to average depth and number of nodes.

We also discuss the hardness results for decision tree optimization. Particularly, we show that total Boolean functions in a decision table form have polynomial-time algorithms for construction and optimization of decision trees in terms of the size of the table (the existence of such algorithm for minimization of size of decision trees was proved earlier by David Guijarro, Víctor Lavín, and Vijay Raghavan [10]). However, for general cases of partial Boolean functions the problem of optimization remains hard. We provide the proofs of NP-hardness for the well-understood cases when optimizing decision trees for depth and number of nodes of decision trees and show the result from Hyafil and Rivest [13] about NP-hardness of minimization of average depth of decision trees for pseudo Boolean functions.

An essential part of this paper is devoted to the experimental study of total and partial Boolean functions. We begin with the study of three well-known functions: conjunction, linear function, and majority function each for n variables for n from 2 to 14 (and in some cases up to $n = 16$). We study minimum depth, average depth and number of nodes for these functions as well as the existence of totally optimal decision trees for each pair of the considered cost functions. We show that such totally optimal trees exist in all cases we have studied, and how the complexity of decision trees depends on their accuracy.

The second direction of experimental research is to study the existence of totally optimal decision trees for monotone and arbitrary total Boolean functions. In [6], we proved that for any monotone Boolean function with at most five variables, there exists a totally optimal decision tree relative to the depth and the number of nodes. In this paper, we extend this result to six variables and find a counterexample for seven variables. We obtained similar results for each possible pair of these three cost functions and all three cost functions together. We also studied randomly generated total Boolean functions for $n = 4, \ldots, 10$. The obtained experimental results allowed us to formulate the following hypotheses (which we subsequently proved in Theorems 16 and 17): for almost all total Boolean functions there exist (i) totally optimal decision trees relative to the depth and number of nodes, and (ii) totally optimal decision trees relative to the depth and average depth, respectively.

The third direction of study is devoted to the experiments with partial Boolean functions. The obtained results show that the percentage of partial Boolean function that have totally optimal trees, is decreasing quickly with the growth of the number of variables. This situation is in some sense similar to the situation with patterns in the partial Boolean functions: for each total Boolean function, for each point in Boolean cube, there is a pattern which covers this point and only the points with the same value of the function, has the minimum length of description and maximum coverage. For partial Boolean functions, such patterns do not exist in each case, and that is why it is necessary to consider Pareto optimal patterns [11].

The experimental results presented in this paper lead us to postulate two hypotheses (mentioned above) regarding the total optimality of decision trees relative to (i) depth and number of nodes, and (ii) depth and average depth. We formally treat these hypotheses and prove the consequent theorems.

This paper consists of seven sections. Section 2 explains partial Boolean functions and decision trees for these functions. Section 3 presents definitions and tools to study the decision trees. We prove some hardness results for decision tree optimization in Section 4. Section 5 is devoted to the consideration of experimental results. Proofs of hypotheses postulated with the help of experimental results go to Section 6, and Section 7 concludes the paper.

2. Partial Boolean functions and decision trees

In this section, we consider the notions connected with table representation of partial Boolean functions, and with approximate decision trees (α-decision trees).

A *partial Boolean function* $f(x_1, \ldots, x_n)$ is a partial function of the kind $f : \{0, 1\}^n \rightarrow \{0, 1\}$. We work with a *table representation of the function* f (*table* for short) which is a rectangular table T with n columns filled with numbers from the set $\{0, 1\}$. Columns of the table are labeled with variables $x_1, \ldots, x_n$. Rows of the table are pairwise different, and the set of