Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 22 (2013) 359 – 365

Procedia Computer Science

17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems - KES2013

Totally Optimal Decision Trees for Monotone Boolean Functions with at most Five Variables

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

Abstract

In this paper, we present the empirical results for relationships between time (depth) and space (number of nodes) complexity of decision trees computing monotone Boolean functions, with at most five variables. We use Dagger (a tool for optimization of decision trees and decision rules) to conduct experiments. We show that, for each monotone Boolean function with at most five variables, there exists a totally optimal decision tree which is optimal with respect to both depth and number of nodes.

Keywords: Totally optimal decision trees; monotone Boolean functions; number of nodes and depth of decision trees.

1. Introduction

Decision trees can be used as classifiers, a way for representing knowledge, and also as algorithms for solving different problems (see for example [1]). These different uses require optimizing decision trees for different criteria. For this purpose, we have created a software system for decision trees (as well as decision rules) called dagger—a tool based on dynamic programming which allows us to optimize decision trees (and decision rules) relative to various cost functions such as depth (length), average depth (average length), total number of nodes, and number of misclassifications sequentially [2, 3, 4, 5].

Often, during experiments with Dagger, on data from UCI ML Repository [6], we get totally optimal decision trees – simultaneously optimal relative to the depth and number of number of nodes. For example, in [7] we show that breast-cancer and house-vote datasets have totally optimal trees while there does not exist such totally optimal decision trees for the dataset lymphography. These totally optimal decision trees can be useful from the points of view of knowledge representation and efficiency of algorithms.

Studying relationship between time and space complexity of algorithms is an important topic of computational complexity theory. These relationships are considered often for non-universal computational models such as branching programs and decision trees [8, 9], where time and space complexity is independent of the length of input. The considered relationships become trivial if there exist totally optimal algorithms i.e., optimal with

E-mail address: {igor.chikalov, shahid.hussain, mikhail.moshkov}@kaust.edu.sa.

doi:10.1016/j.procs.2013.09.113

f_1	$\cdots$	f_m	d
b_{11}	$\cdots$	b_{1m}	c_1
$\vdots$	$\ddots$	$\vdots$	$\vdots$
b_{N1}	$\cdots$	b_{Nm}	c_N

Fig. 1. Decision table T with m attributes and N rows

respect both time and space complexity. To understand the phenomenon of existence of totally optimal decision trees (whether it is usual or rare), we studied monotone Boolean functions.

In this paper, we study decision trees for computation of monotone Boolean functions with n variables, $0 \leq n \leq 5$. We consider the depth and the number of nodes of decision tree as time and space complexity, respectively. For each monotone Boolean function with at most five variables, we study relationship between depth and number of nodes in decision trees computing this function. As a result, we obtain that, for each monotone Boolean function with at most five variables, there exists a totally optimal decision tree i.e., a decision tree with both minimum depth and minimum number of nodes.

This paper is organized into five sections including the Introduction. Section 2 presents some important basic notions related to decision tables/trees, cost functions, and representation of sets of decision trees for a given decision table. Section 3 describes in detail the procedure of optimization for decision trees. Main result of this paper goes into Section 4, including the plots for totally optimal decision trees for monotone Boolean functions. Section 5, concludes the paper followed by references.

2. Basic Notions

In the following, we consider notions of decision tables and decision trees in general case. Later, we will discuss the corresponding notions for monotone Boolean functions.

2.1. Decision Tables and Trees

A *decision table* is a rectangular array of values, arranged in rows and columns. The columns of a decision table are labeled with conditional attributes and rows of the table contain values of corresponding attributes. In this chapter, we consider only decision tables with discrete attributes. These tables contain neither missing values nor equal rows. Consider a decision table T depicted in Fig 1. Here $f_1, \ldots, f_m$, are names of columns (conditional attributes); $c_1, \ldots, c_N$, nonnegative integers, which are interpreted as decisions (values of the decision attribute d); b_{ij} are nonnegative integers which are interpreted as values of conditional attributes. We assume that all rows are pairwise different. We denote by $E(T)$ the set of attributes (columns of T). For $f_i \in E(T)$, we say $E(T, f_i)$ is the set of values for the column f_i.

Let $f_{i_1}, \ldots, f_{i_t} \in E(T)$ form a subset of t attributes from T and let $a_1, \ldots, a_t$ be their corresponding values, then we denote by $T(f_i, a_1) \ldots (f_i, a_t)$, the subtable of the table T, which consists of only the rows (of T) that are at the intersection of columns $f_{i_1}, \ldots, f_{i_t}$, have values $a_1, \ldots, a_t$, respectively. Such nonempty tables (including the table T) are called *separable subtables* of the table T. For a subtable Θ of the table T, we denote $R(\Theta)$, the number of unordered pairs of rows that are labeled with different decisions.

A *decision tree* Γ *over the table* T is a finite directed rooted tree in which each terminal node is labeled with a decision. Each nonterminal node is labeled with a conditional attribute, and for each nonterminal node the outgoing edges are labeled with pairwise different nonnegative integers. For each node v of Γ, we associate a subtable $T(v)$ of the table T. If v is the root node then $T(v) = T$. For every other node v of Γ, $T(v) = T(f_{i_1}, a_1) \ldots (f_{i_t}, a_t)$, where $f_{i_1}, \ldots, f_{i_t}$ are the attributes attached to the nodes in path from the root to v and $a_1, \ldots, a_t$ are values of these attributes that are attached to the edges in this path.

We say that a tree Γ is *a decision tree for* T if for any node v of Γ following conditions are satisfied:

- If $R(T(v)) = 0$ then, v is a terminal nodes, labeled with the common decision for $T(v)$,