Decision Tree Construction using Greedy Algorithms and Dynamic Programming – Comparative Study

Abdulaziz Alkhalid, Igor Chikalov, and Mikhail Moshkov

Mathematical and Computer Sciences & Engineering Division
King Abdullah University of Science and Technology
Thuwal 23955-6900, Saudi Arabia
{abdulaziz.alkhalid, igor.chikalov, mikhail.moshkov}@kaust.edu.sa

Abstract. The paper presents comparative study for different heuristics used by greedy algorithms for constructing of decision trees. We consider the problem of exact learning for decision tables with discrete attributes. We made experiments with randomly generated decision tables contain attributes with three categories {0,1,2}. Complexity of decision trees is estimated relative to several cost functions: depth, average depth, number of nodes, number of terminal nodes and number of nonterminal nodes. Costs of trees built by greedy algorithms are compared with exact minimums calculated by an algorithm based on dynamic programming. Based on the results we choose the best two greedy algorithms for each cost function.

Key words: decision tables, decision trees, greedy algorithms, dynamic programming

1 Introduction

Decision trees are widely used as a way for knowledge representation, as predictors and as algorithms for problem solving in rough set theory [12, 14], machine learning and data mining [3], test theory [4], etc. To have more understandable decision trees we need to minimize the number of nodes in a tree. To have faster decision trees we need to minimize the depth or average depth of a tree. Unfortunately, the most problems connected with decision tree optimization are NP-hard [9, 11].

The majority of approximate algorithms for decision tree optimization are based on greedy approach. These algorithms build tree in a top-down fashion, minimizing some impurity criteria at each step. There are several impurity criteria designed using theoretical-information [13], statistical [3] and combinatorial [10, 11] reasoning which can be used by greedy algorithms.

We assume that the decision tables contain only categorical attributes and free of inconsistency. Several cost functions are considered that characterize space and time complexity of decision trees: depth, average depth, and number of

CONCURRENCY, SPECIFICATION AND PROGRAMMING
M. Szczuka et al. (eds.): Proceedings of the international workshop CS&P 2011
September 28-30, Pułtusk, Poland, pp. 1-9

nodes, number of terminal nodes and number of nonterminal nodes. We use randomly generated decision tables with attributes contain values from the set {0,1,2}.

Costs of trees constructed by greedy algorithms are compared with exact minimum, calculated by an algorithm based on dynamic programming. The idea is close to algorithms described in [5, 6], but authors devised it independently and made several improvements. For example, the algorithm is capable of founding a set of optimal trees and perform sequential optimization by different criteria [1, 2, 7] (we do not consider these extensions in the paper). An effective implementation allows for applying the algorithm to decision tables containing dozens of columns (attributes) and hundreds to thousands rows (objects).

The paper is organized as follows. Section 2 introduces basic notions. Section 3 contains general schema of greedy algorithm. Section 4 describes an exact algorithm based on dynamic programming. Section 5 presents experimental setup and results of experiments. Section 6 contains conclusions.

2 Basic Notions

In this paper, we consider only decision tables with categorical attributes. These tables do not contain missing values and equal rows. A *decision table* is a rectangular table T with m columns and N rows. Columns of T are labeled with *attributes* $f_1, \ldots, f_m$. Rows of T are filled by nonnegative integers which are interpreted as values of these attributes. Rows are pairwise different, and each row is labeled with a nonnegative integer which is interpreted as the *decision*. We denote by $E(T)$ the set of attributes (columns of the table T), each of which contains different values. For $f_i \in E(T)$, let $E(T, f_i)$ be the set of values from the column f_i. We denote by $N(T)$ the number of rows in the table T.

Let $f_{i_1}, \ldots, f_{i_r} \in \{f_1, \ldots, f_m\}$ and $b_1, \ldots, b_r$ be nonnegative integers. We denote by $T(f_{i_1}, b_1) \ldots (f_{i_r}, b_r)$ the subtable of the table T, which consists of such and only such rows of T that at the intersection with columns $f_{i_1}, \ldots, f_{i_r}$ have numbers $b_1, \ldots, b_r$ respectively. Such nonempty tables (including the table T) will be called *separable subtables* of the table T.

Let rows of T be labeled with k different decisions $d_1, \ldots, d_k$. For $i = 1, \ldots, k$, let N_i be the number of rows in T labeled with the decision d_i, and $p_i = N_i/N$.

We consider four uncertainty measures for decision tables: entropy $ent(T) = -\sum_{i=1}^{k} p_i \log_2 p_i$ (we assume $0 \log_2 0 = 0$), Gini index $gini(T) = 1 - \sum_{i=1}^{k} p_i^2$, minimum misclassification error $me(T) = N - \max_{1 \le j \le k} N_j$, and the number $rt(T)$ of unordered pairs of rows in T with different decisions (note that $rt(T) = N^2 gini(T)/2$).

Let $f_i \in E(T)$ and $E(T, f_i) = \{a_1, \ldots, a_t\}$. The attribute f_i divides the table T into subtables $T_1 = T(f_i, a_1), \ldots, T_t = T(f_i, a_t)$. We now define an *impurity function* I which gives us the *impurity* $I(T, f_i)$ of this partition. Let us fix an uncertainty measure U from the set $\{ent, gini, me, rt\}$ and type of impurity function: *sum*, *max*, *weighted-sum*, or *weighted-max*. Then for the type *sum*, $I(T, f_i) = \sum_{j=1}^{t} U(T_j)$, for the type *max*, $I(T, f_i) = \max_{1 \le j \le t} U(T_j)$, for

the type *weighted-sum*, $I(T, f_i) = \sum_{j=1}^{t} U(T_j)N(T_j)/N(T)$, and for the type *weighted-max*, $I(T, f_i) = \max_{1 \le j \le t} U(T_j)N(T_j)/N(T)$. As a result, we have 16 different impurity functions.

A *decision tree* Γ *over* the table T is a finite directed tree with the root in which each terminal node is labeled with a decision. Each nonterminal node is labeled with an attribute from the set $\{f_1, \ldots, f_m\}$, and for each nonterminal node the outgoing edges are labeled with pairwise different nonnegative integers. Let v be an arbitrary node of Γ. We now define a subtable $T(v)$ of the table T. If v is the root then $T(v) = T$. Let v be a node of Γ that is not the root, nodes in the path from the root to v be labeled with attributes $f_{i_1}, \ldots, f_{i_t}$, and edges in this path be labeled with values $a_1, \ldots, a_t$ respectively. Then $T(v) = T(f_{i_1}, a_1), \ldots, (f_{i_t}, a_t)$.

Let Γ be a decision tree over T. We will say that Γ is a *decision tree for* T if any node v of Γ satisfies the following conditions:

- If $rt(T(v)) = 0$ then v is a terminal node labeled with the common decision for $T(v)$.
- Otherwise, v is labeled with an attribute $f_i \in E(T(v))$ and, if $E(T(v), f_i) = \{a_1, \ldots, a_t\}$, then t edges leave node v, and these edges are labeled with $a_1, \ldots, a_t$ respectively.

We will consider cost functions which are given in the following way: values of the considered cost function ψ, which are nonnegative numbers, are defined by induction on pairs (T, Γ), where T is a decision table and Γ is a decision tree for T. Let Γ be a decision tree that contains only one node labeled with a decision. Then $\psi(T, \Gamma) = \psi^0$ where ψ^0 is a nonnegative number. Let Γ be a decision tree in which the root is labeled with an attribute f_i, and t edges start in the root. These edges are labeled with numbers $a_1, \ldots, a_t$ and enter roots of decision trees $\Gamma_1, \ldots, \Gamma_t$. Then

$$\psi(T, \Gamma) = F(N(T), \psi(T(f_i, a_1), \Gamma_1), \ldots, \psi(T(f_i, a_t), \Gamma_t)).$$

Here $F(n, \psi_1, \psi_2, \ldots)$ is an operator which transforms the considered tuple of nonnegative numbers into a nonnegative number. Note that the number of variables $\psi_1, \psi_2, \ldots$ is not bounded from above.

The considered cost function will be called *monotone* if for any natural t, from inequalities $c_1 \le d_1, \ldots, c_t \le d_t$ the inequality $F(a, c_1, \ldots, c_t) \le F(a, d_1, \ldots, d_t)$ follows. Now we take a closer view of some monotone cost functions.

Number of nodes: $\psi(T, \Gamma)$ is the number of nodes in decision tree Γ. For this cost function, $\psi^0 = 1$ and
$F(n, \psi_1, \psi_2, \ldots, \psi_t) = 1 + \sum_{i=1}^{t} \psi_i$.

Depth: $\psi(T, \Gamma)$ is the maximum length of a path from the root to a terminal node of Γ. For this cost function, $\psi^0 = 0$ and $F(n, \psi_1, \psi_2, \ldots, \psi_t) = 1 + \max\{\psi_1, \ldots, \psi_t\}$.

Total path length: for an arbitrary row $\bar{\delta}$ of the table T, we denote by $l(\bar{\delta})$ the length of the path from the root to a terminal node v of Γ such that $\bar{\delta}$ belongs

to $T(v)$. Then $\psi(T,\Gamma) = \sum_{\bar{\delta}} l(\bar{\delta})$, where we take the sum on all rows $\bar{\delta}$ of the table T. For this cost function, $\psi^0 = 0$ and $F(n,\psi_1,\psi_2,\ldots,\psi_t) = n + \sum_{i=1}^{t}\psi_i$.

Note that the *average depth* of Γ is equal to the total path length divided by $N(T)$.

Number of nonterminal nodes: $\psi(T,\Gamma)$ is the number of nonterminal nodes in decision tree Γ. For this cost function, $\psi^0 = 0$ and $F(n,\psi_1,\psi_2,\ldots,\psi_t) = 1 + \sum_{i=1}^{t}\psi_i$.

Number of terminal nodes: $\psi(T,\Gamma)$ is the number of terminal nodes in decision tree Γ. For this cost function, $\psi^0 = 1$ and $F(n,\psi_1,\psi_2,\ldots,\psi_t) = 1+\sum_{i=1}^{t}\psi_i$.

3 Greedy Approach

Let I be an impurity function. We now describe a greedy algorithm V_I which for a given decision table T constructs a decision tree $V_I(T)$ for the table T.

Step 1. Construct a tree consisting of a single node labeled with the table T and proceed to the second step.

Suppose $t \geq 1$ steps have been made already. The tree obtained at the step t will be denoted by G.

Step $(t+1)$. If no node of the tree G is labeled with a table then we denote by $V_I(T)$ the tree G. The work of the algorithm V_I is completed.

Otherwise, we choose a node v in the tree G which is labeled with a subtable Θ of the table T. If $rt(\Theta) = 0$ then instead of Θ we mark the node v by the common decision for Θ and proceed to the step $(t+2)$. Let $rt(\Theta) > 0$. Then for each $f_i \in E(\Theta)$ we compute the value $I(T,f_i)$. We mark the node v by the attribute f_{i_0} where i_0 is the minimum $i \in \{1,\ldots,m\}$ for which $I(T,f_i)$ has the minimum value. For each $\delta \in E(\Theta,f_{i_0})$, we add to the tree G the node $v(\delta)$, mark this node by the subtable $\Theta(f_{i_0},\delta)$, draw the edge from v to $v(\delta)$, and mark this edge by δ. Proceed to the step $(t+2)$.

4 Dynamic Programming Approach

In this section, we describe a dynamic programming algorithm which for a monotone cost function ψ and decision table T finds the minimum cost (relative to the cost function ψ) of decision tree for T.

Consider an algorithm for construction of a graph $\Delta(T)$. Nodes of $\Delta(T)$ are some separable subtables of the table T. During each step we process one node and mark it with symbol *. We start with the graph that consists of one node T and finish when all nodes of the graph are processed.

Let the algorithm have already performed p steps. We now describe the step number $(p+1)$. If all nodes are processed then the work of the algorithm is finished, and the resulted graph is $\Delta(T)$. Otherwise, choose a node (table) Θ that has not been processed yet. If $rt(\Theta) = 0$, label the considered node with the common decision for Θ, mark it with symbol * and proceed to the step number $(p+2)$. Let $rt(\Theta) > 0$. For each $f_i \in E(\Theta)$, draw a bundle of edges from the node

Θ (this bundle of edges will be called f_i-*bundle*). Let $E(\Theta, f_i) = \{a_1, \ldots, a_t\}$. Then draw t edges from Θ and label these edges with pairs $(f_i, a_1), \ldots, (f_i, a_t)$ respectively. These edges enter into nodes $\Theta(f_i, a_1), \ldots, \Theta(f_i, a_t)$. If some of nodes $\Theta(f_i, a_1), \ldots, \Theta(f_i, a_t)$ do not present in the graph then add these nodes to the graph. Mark the node Θ with symbol * and proceed to the step number $(p + 2)$.

Let ψ be a monotone cost function given by the pair ψ^0, F. We now describe a procedure, which attaches a number to each node of $\Delta(T)$. We attach the number ψ^0 to each terminal node of $\Delta(T)$.

Consider a node Θ, which is not terminal, and a bundle of edges, which starts in this node. Let edges be labeled with pairs $(f_i, a_1), \ldots, (f_i, a_t)$, and edges enter to nodes
$\Theta(f_i, a_1), \ldots, \Theta(f_i, a_t)$, to which numbers $\psi_1, \ldots, \psi_t$ are attached already. Then we attach to the considered bundle the number $F(N(\Theta), \psi_1, \ldots, \psi_t)$. Among numbers attached to bundles starting in Θ we choose the minimum number and attach it to the node Θ.

We stop when a number will be attached to the node T in the graph $\Delta(T)$. One can show that this number is the minimum cost (relative to the cost function ψ) of decision tree for T.

5 Experimental Results

In this section, we consider results of 40000 experiments with randomly generated decision tables.

Each table contains 50 rows and 10 conditional attributes. The values of conditional and decision attributes are form the set {0,1,2}. We choose values 0, 1 and 2 with the same probability.

In some table there were rows that contains identical values in all columns, possibly, except the decision column. In this case, each group of identical rows was replaced with a single row with common values in all conditional columns and the most common value on the decision column.

We divide 40000 experiments into four groups with 10000 experiments in each. We study only exact decision trees, 16 greedy algorithms and five cost function : depth h, average depth h_{av}, number of nodes L, number of terminal nodes L_t, and number of nonterminal nodes L_n.

Instead of the cost of decision tree, constructed by greedy algorithm (greedy_cost), we consider relative difference of greedy_cost and min_cost:

$$\frac{\text{greedy\_cost} - \text{min\_cost}}{\text{min\_cost}}.$$

We will evaluate greedy algorithms based on this parameter. Let us remind that each impurity function is defined by its type (*sum*, *max*, *w_sum* or *w_max*) and uncertainty measure (*ent*, *gini*, *me*, or *rt*).

We consider the average value of relative differences for each subgroup with 10000 tables. Tables 1–4 show the average values of relative difference for each cost function, each heuristic and each group of experiments.

Table 1. Average value of relative difference for given cost function for the first 10000 experiments with a given greedy algorithm using randomly generated tables

	h_{av}	h	L	L_n	L_t
			max		
ent	0.138302	0.186858	0.357019	0.375511	0.376977
gini	0.139207	0.187550	0.359458	0.378020	0.379447
me	0.135144	0.074800	0.449110	0.519650	0.441295
rt	0.110908	0.020700	0.450723	0.552395	0.425130
			sum		
ent	0.123865	0.317392	0.246790	0.238360	0.279813
gini	0.125776	0.318725	0.250817	0.242099	0.284070
me	0.089036	0.183867	0.261832	0.265028	0.288510
rt	0.058667	0.055800	0.260201	0.289344	0.271940
			w_max		
ent	0.106484	0.023500	0.424265	0.513412	0.405216
gini	0.106813	0.022450	0.428339	0.520407	0.407705
me	0.109465	0.022375	0.439951	0.536621	0.416928
rt	0.113009	0.021100	0.460582	0.565543	0.433321
			w_sum		
ent	0.061044	0.147225	0.208063	0.213318	0.232353
gini	0.058550	0.162608	0.201708	0.206141	0.226325
me	0.067257	0.070250	0.267331	0.288588	0.283719
rt	0.071824	0.028325	0.321480	0.377320	0.319250

Table 2. Average value of relative difference for given cost function for the second 10000 experiments with a given greedy algorithm using randomly generated tables

	h_{av}	h	L	L_n	L_t
			max		
ent	0.138712	0.185075	0.357875	0.376055	0.377325
gini	0.140094	0.189833	0.360672	0.379017	0.380087
me	0.135490	0.074650	0.450279	0.521118	0.441452
rt	0.111381	0.020467	0.452257	0.554366	0.425475
			sum		
ent	0.123562	0.316400	0.245598	0.236394	0.278312
gini	0.125225	0.317200	0.249042	0.240189	0.281641
me	0.089461	0.183208	0.263076	0.265206	0.289669
rt	0.059283	0.055242	0.261237	0.290686	0.272079
			w_max		
ent	0.106932	0.022483	0.425617	0.515296	0.405387
gini	0.107179	0.021658	0.429923	0.523032	0.407837
me	0.109910	0.021792	0.441838	0.538607	0.417903
rt	0.113321	0.020392	0.461697	0.566410	0.433607
			w_sum		
ent	0.060785	0.144725	0.208176	0.212770	0.232146
gini	0.058707	0.164792	0.202402	0.206397	0.226575
me	0.067250	0.069875	0.267388	0.288716	0.283043
rt	0.072385	0.026167	0.323450	0.380218	0.319947

Table 3. Average value of relative difference for given cost function for the third 10000 experiments with a given greedy algorithm using randomly generated tables

	h_{av}	h	L	L_n	L_t
			max		
ent	0.137617	0.185158	0.353147	0.371859	0.372711
gini	0.138883	0.188250	0.355873	0.375106	0.375185
me	0.135333	0.075883	0.448485	0.519380	0.440130
rt	0.110708	0.019942	0.449727	0.550888	0.424056
			sum		
ent	0.123628	0.314350	0.245469	0.236374	0.278650
gini	0.125314	0.316175	0.248820	0.239650	0.282114
me	0.089613	0.181408	0.262751	0.265691	0.289410
rt	0.059529	0.055175	0.261563	0.291150	0.272876
			w_max		
ent	0.106469	0.023742	0.422727	0.511061	0.403858
gini	0.106778	0.022392	0.427845	0.519391	0.407232
me	0.109522	0.021867	0.439807	0.536012	0.416775
rt	0.113163	0.019967	0.460472	0.565323	0.432915
			w_sum		
ent	0.061365	0.145375	0.208717	0.213383	0.233150
gini	0.058957	0.162283	0.201743	0.205992	0.226279
me	0.067432	0.070033	0.266903	0.288704	0.282819
rt	0.072100	0.025883	0.321725	0.377865	0.319149

Table 4. Average value of relative difference for given cost function for the fourth 10000 experiments with a given greedy algorithm using randomly generated tables

	h_{av}	h	L	L_n	L_t
			max		
ent	0.137746	0.182383	0.354142	0.371413	0.374584
gini	0.138820	0.184333	0.356370	0.374508	0.376382
me	0.135861	0.075050	0.449204	0.519590	0.441291
rt	0.110753	0.020025	0.450736	0.551287	0.425463
			sum		
ent	0.123646	0.315375	0.244816	0.235797	0.278023
gini	0.125661	0.318083	0.248975	0.239853	0.282362
me	0.091105	0.186658	0.265199	0.268307	0.291879
rt	0.059374	0.054725	0.262123	0.290910	0.273949
			w_max		
ent	0.106344	0.023408	0.423681	0.511672	0.405050
gini	0.106676	0.022667	0.428509	0.519639	0.408184
me	0.109939	0.022792	0.441653	0.537470	0.418941
rt	0.113125	0.020075	0.461237	0.565246	0.434245
			w_sum		
ent	0.061625	0.145508	0.209636	0.214445	0.234108
gini	0.059607	0.161550	0.203449	0.207444	0.228237
me	0.067749	0.070042	0.268840	0.289638	0.285424
rt	0.072147	0.027242	0.322424	0.378150	0.320088

To summarize the results, we chose the best two heuristics for each cost function. Table 5 shows the best two greedy algorithms for each cost function.

Table 5. The best two greedy algorithms for each cost function

cost function	The best two greedy algorithms
h_{av}	(sum, rt), (w_sum, $gini$)
h	(max, rt), (w_max, rt)
L	(w_sum, $gini$), (w_sum, ent)
L_n	(w_sum, $gini$), (w_sum, ent)
L_t	(w_sum, $gini$), (w_sum, ent)

6 Conclusions

The paper is devoted to the study of 16 greedy algorithms for decision tree construction. For 40000 randomly generated decision tables with attributes contains three values {0,1,2} we compare the values of depth, average depth, number of nodes, number of nonterminal nodes and number of terminal nodes, constructed by these algorithms, with the minimums found by an algorithm based on dynamic programming approach. The best two greedy algorithms were selected for each cost function.

References

1. Alkhalid, A., Chikalov, I., Moshkov, M.: On algorithm for building of optimal α-decision trees. In: Szczuka, M. et al. (eds.) RSCTC 2010, LNCS (LNAI), vol. 6086, pp. 438–445. Springer, Heidelberg (2010)
2. Alkhalid, A., Chikalov, I., Moshkov, M.: A tool for study of optimal decision trees. In: Yu, J. et al. (eds.): RSKT 2010, LNCS (LNAI), vol. 6401, pp. 353–360.Springer, Heidelberg (2010)
3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth & Brooks, 1984
4. Chegis, I.A., Yablonskii, S.V.: Logical methods of electric circuit control. Trudy MIAN SSSR 51 (1958) 270–360 (in Russian)
5. Garey, M. R.: Optimal binary identification procedures. SIAM Journal on Applied Mathematics, 23(2):173–186, 1972.
6. Martelli, A. and Montanari,U.: Optimizing decision trees through heuristically guided search. Commun. ACM, 21:1025–1039, December 1978.
7. Moshkov, M. J. and I. Chikalov. V., Consecutive optimization of decision trees concerning various complexity measures. Fundamenta Informaticae, 61(2):87–96, 2003.
8. Chikalov, I., Moshkov, M., Zelentsova, M.: On optimization of decision trees. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets IV. LNCS, vol. 3700, pp. 18–36. Springer, Heidelberg (2005)

9. Hyafil, L., Rivest, R.: Constructing optimal binary decision trees is NP-complete. Information Processing Letters, 5 (1976) 15–17
10. Moret, B. E., Thomason, R. C., Gonzalez, M.: The activity of a variable and its relation to decision trees. ACM Transactions on Programming Languages and Systems (TOPLAS), 2 (1980) 580–595
11. Moshkov, M.: Time complexity of decision trees. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rought Set III. LNCS, vol. 3400, pp. 244–459. Springer, Heidelberg (2005)
12. Pawlak, Z.: Rough Sets – Theoretical Aspects of Reasoning about Data. Kluwer Academic Publishers, Dordrecht, Boston, London, 1991
13. Quinlan, J. R.: Induction of decision trees. Mach. Learn. 1 (1986) 81–106.
14. Skowron, A., Rauszer, C.: The discernibility matrices and functions in information systems. Intelligent Decision Support. Handbook of Applications and Advances of the Rough Set Theory. Edited by R. Slowinski. Kluwer Academic Publishers, Dordrecht, Boston, London (1992) 331–362