COMPARISON OF GREEDY ALGORITHMS FOR DECISION TREE CONSTRUCTION

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

Keywords: Decision trees, Greedy algorithms, Dynamic programming.

Abstract: The paper compares different heuristics that are used by greedy algorithms for constructing of decision trees. Exact learning problem with all discrete attributes is considered that assumes absence of contradictions in the decision table. Reference decision tables are based on 24 data sets from UCI Machine Learning Repository (Frank and Asuncion, 2010). Complexity of decision trees is estimated relative to several cost functions: depth, average depth, and number of nodes. Costs of trees built by greedy algorithms are compared with exact minimums calculated by an algorithm based on dynamic programming. The results associate to each cost function a set of potentially good heuristics that minimize it.

1 INTRODUCTION

Decision trees are widely used for representing of knowledge, for prediction and as algorithms in search theory (Ahlswede and Wegener, 1979), machine learning (Breiman et al., 1984; Quinlan, 1993), fault diagnosis (Pattipati and Dontamsetty, 1992), etc. Many problems of constructing optimal decision trees are *NP*-hard (Hyafil and Rivest, 1976). Moreover, for several problem statements an approximation preserving reduction is done that guarantees absence of polynomial complexity algorithms under reasonable assumptions about complexity classes *P* and *NP* (Alekhnovich et al., 2004; Heeringa and Adler, 2005).

The majority of approximate algorithms for decision tree construction are based on greedy approach. Such algorithms build tree in a top-down fashion, minimizing some impurity criteria at each step. There are several impurity criteria designed using theoretical-information (Quinlan, 1986), statistical (Breiman et al., 1984) and combinatorial (Moret et al., 1980) reasoning. For some criteria, bounds on approximation ratio is obtained that limit deviation of tree characteristics from the minimum (Chakaravarthy et al., 2007; Heeringa and Adler, 2005; Moshkov, 2010).

The aim of our work is comparative analysis of several greedy algorithms in application to exact learning problems. 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 nodes. Since algorithm behavior depends heavily on the input data, we choose reference decision tables close to real-life problems, taking data sets mainly from UCI Machine Learning Repository (Frank and Asuncion, 2010) as a base.

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 (Garey, 1972; Martelli and Montanari, 1978), 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 (Moshkov and Chikalov, 2003) (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.

Alkhalid A., Chikalov I. and Moshkov M..
COMPARISON OF GREEDY ALGORITHMS FOR DECISION TREE CONSTRUCTION.
DOI: 10.5220/0003654904300435
In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval* (KDIR-2011), pages 430-435
ISBN: 978-989-8425-79-9

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* $f_1,\dots,f_m$. 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 $f_{i_1},\dots,f_{i_r} \in \{f_1,\dots,f_m\}$ and $b_1,\dots,b_r$ be nonnegative integers. We denote by $T(f_{i_1},b_1)\dots(f_{i_r},b_r)$ the subtable of the table T, which consists of such and only such rows of T that at the intersection with columns $f_{i_1},\dots,f_{i_r}$ have numbers $b_1,\dots,b_r$ 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 $d_1,\dots,d_k$. For $i=1,\dots,k$, 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 $E(T,f_i) = \{a_1,\dots,a_t\}$. The attribute f_i divides the table T into subtables $T_1 = T(f_i,a_1),\dots,T_t = T(f_i,a_t)$. 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 $\{f_1,\dots,f_m\}$, 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 $f_{i_1},\dots,f_{i_t}$, and edges in this path be labeled with values $a_1,\dots,a_t$ respectively. Then $T(v) = T(f_{i_1},a_1),\dots,(f_{i_t},a_t)$.

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 $E(T(v),f_i) = \{a_1,\dots,a_t\}$, then t edges leave node v, and these edges are labeled with $a_1,\dots,a_t$ 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 $a_1,\dots,a_t$ and enter roots of decision trees $\Gamma_1,\dots,\Gamma_t$. Then

$\psi(T,\Gamma) = F(N(T),\psi(T(f_i,a_1),\Gamma_1),\dots,$
$\psi(T(f_i,a_t),\Gamma_t))$. Here $F(n,\psi_1,\psi_2,\dots)$ is an operator which transforms the considered tuple of nonnegative numbers into a nonnegative number. Note that the number of variables $\psi_1,\psi_2,\dots$ is not bounded from above.

The considered cost function will be called *monotone* if for any natural t, from inequalities $c_1 \le d_1,\dots,c_t \le d_t$ the inequality $F(a,c_1,\dots,c_t) \le F(a,d_1,\dots,d_t)$ 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 $F(n,\psi_1,\psi_2,\dots,\psi_t) = 1 + \sum_{i=1}^{t} \psi_i$.

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 $F(n,\psi_1,\psi_2,\dots,\psi_t) = 1 + \max\{\psi_1,\dots,\psi_t\}$.

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 $F(n,\psi_1,\psi_2,\dots,\psi_t) = n + \sum_{i=1}^{t} \psi_i$.

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

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

Different impurity functions give us different greedy algorithms. The following experiments compare average depth, number of nodes and depth of decision trees built by these algorithms with the minimum average depth, minimum number of nodes and minimum depth calculated by the dynamic programming algorithm.

The data sets were taken from UCI Machine Learning Repository (Frank and Asuncion, 2010). Experiments using data sets which are not from UCI Machine Learning Repository give us similar results. Each data set is represented as a table containing several input columns and an output (decision) column. Some data sets contain index columns that take unique value for each row. Such columns were removed. In some tables there were rows that contain 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 input columns and the most common value in the decision column. In some tables there were missed values. Each such value was replaced with the most common value in the corresponding column.

Tables 1–3 show results of experiments with 24 data sets and three cost functions: average depth, number of nodes and depth respectively. Each row contains data set name, minimum cost of decision tree (min_cost), calculated with the dynamic programming algorithm (see column Opt), and infor-

Table 1: Results of Experiments with Average Depth.

Name	Opt	*sum*				*weighted-sum*			
		ent	*gini*	*me*	*rt*	*ent*	*gini*	*me*	*rt*
adult-stretch	1.50	0.00	0.00	1.33	0.00	0.00	0.00	1.33	0.00
agaricus-lepiota	1.52	0.54	0.54	0.01	0.00	0.00	0.00	0.00	0.30
balance-scale	3.55	0.00	0.00	0.02	0.00	0.00	0.00	0.02	0.00
breast-cancer	3.24	0.96	0.96	0.25	0.02	0.08	0.14	0.02	0.03
cars	2.95	0.04	0.04	0.26	0.28	0.00	0.00	0.36	0.49
flags	2.72	2.43	2.58	0.18	0.04	0.16	0.16	0.04	0.03
hayes-roth-data	2.62	0.01	0.01	0.01	0.00	0.01	0.01	0.00	0.00
house-votes-84	3.54	0.66	0.97	0.49	0.06	0.04	0.07	0.06	0.02
lenses	1.80	0.00	0.00	0.67	0.67	0.67	0.00	0.67	0.67
lymphography	2.67	1.66	1.66	0.26	0.06	0.17	0.17	0.05	0.04
monks-1-test	2.50	0.80	0.80	0.00	0.00	0.00	0.00	0.00	0.00
monks-1-train	2.53	0.71	0.71	0.00	0.09	0.26	0.27	0.00	0.00
monks-2-test	5.30	0.01	0.01	0.01	0.05	0.02	0.02	0.05	0.05
monks-2-train	4.11	0.14	0.14	0.10	0.02	0.06	0.06	0.04	0.04
monks-3-test	1.83	1.24	0.52	0.52	0.00	0.00	0.14	0.00	0.00
monks-3-train	2.51	0.50	0.21	0.08	0.01	0.01	0.01	0.01	0.01
nursery	3.45	0.17	0.22	0.09	0.09	0.01	0.00	0.12	0.21
poker-hand-training-true	4.09	0.60	0.60	0.14	0.01	0.01	0.01	0.01	0.01
shuttle-landing-control	2.33	0.69	0.69	0.26	0.00	0.03	0.03	0.00	0.00
soybean-small	1.34	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.41
spect-test	2.95	1.01	0.88	0.67	0.18	0.03	0.13	0.17	0.16
teeth	2.78	0.58	0.62	0.00	0.02	0.02	0.00	0.02	0.02
tic-tac-toe	4.35	0.12	0.07	0.11	0.13	0.06	0.05	0.16	0.17
zoo-data	2.29	0.69	0.69	0.07	0.04	0.04	0.04	0.04	0.05
Average		0.565	0.539	0.230	0.073	0.069	0.055	0.131	0.113

Table 2: Results of Experiments with Number of Nodes.

Name	Opt	*sum*				*weighted-sum*			
		ent	*gini*	*me*	*rt*	*ent*	*gini*	*me*	*rt*
adult-stretch	5	0.00	0.00	3.60	0.00	0.00	0.00	3.60	0.00
agaricus-lepiota	21	0.57	0.57	0.62	0.38	0.38	0.38	0.38	3.00
balance-scale	501	0.00	0.00	0.07	0.00	0.00	0.00	0.07	0.00
breast-cancer	161	0.45	0.45	0.37	0.32	0.25	0.25	0.29	0.41
cars	396	0.10	0.10	0.38	0.21	0.03	0.03	0.70	1.35
flags	97	0.80	1.01	0.34	0.62	0.25	0.25	0.59	0.61
hayes-roth-data	52	0.06	0.06	0.06	0.02	0.06	0.06	0.02	0.02
house-votes-84	45	1.02	1.38	1.02	0.36	0.18	0.27	0.31	0.27
lenses	8	0.00	0.00	0.88	0.88	0.88	0.00	0.88	0.88
lymphography	53	0.89	0.89	0.53	0.66	0.43	0.43	0.55	0.77
monks-1-test	37	3.51	3.51	0.11	0.11	0.11	0.11	0.11	0.11
monks-1-train	36	1.86	1.86	0.11	0.67	1.28	1.39	0.11	0.11
monks-2-test	403	0.00	0.00	0.09	0.58	0.19	0.19	0.58	0.58
monks-2-train	129	0.16	0.16	0.43	0.35	0.23	0.23	0.34	0.44
monks-3-test	17	3.65	1.71	2.71	0.00	0.00	0.12	0.00	0.00
monks-3-train	38	0.82	0.37	0.18	0.18	0.11	0.11	0.18	0.18
nursery	1066	0.58	1.11	0.96	0.95	0.09	0.02	1.35	1.37
poker-hand-training-true	18832	0.36	0.36	0.23	0.19	0.18	0.17	0.18	0.20
shuttle-landing-control	15	0.13	0.13	0.00	0.00	0.00	0.00	0.00	0.00
soybean-small	6	0.17	0.17	0.17	0.17	0.17	0.17	0.17	2.83
spect-test	29	1.72	1.45	1.66	0.83	0.14	0.34	0.69	0.76
teeth	35	0.09	0.09	0.00	0.03	0.03	0.00	0.03	0.03
tic-tac-toe	244	0.68	0.41	0.48	1.00	0.41	0.32	1.05	1.09
zoo-data	17	0.59	0.59	0.35	0.35	0.35	0.35	0.35	0.47
Average		0.758	0.682	0.639	0.368	0.239	0.216	0.522	0.645

Table 3: Results of Experiments with Depth.

Name	Opt	*max*	*weighted-max*				*weighted-sum*			
		rt	*ent*	*gini*	*me*	*rt*	*ent*	*gini*	*me*	*rt*
adult-stretch	2	1.00	1.00	1.00	1.00	1.00	0.00	0.00	1.00	0.00
agaricus-lepiota	3	0.00	0.00	0.00	0.00	0.33	0.33	0.33	0.33	0.33
balance-scale	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
breast-cancer	6	0.00	0.00	0.00	0.00	0.00	0.17	0.33	0.00	0.00
cars	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
flags	4	0.25	0.25	0.25	0.25	0.25	0.75	0.75	0.25	0.25
hayes-roth-data	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
house-votes-84	6	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33	0.33
lenses	3	0.33	0.33	0.33	0.33	0.33	0.33	0.00	0.33	0.33
lymphography	4	0.25	0.25	0.25	0.25	0.25	0.50	0.50	0.25	0.25
monks-1-test	3	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
monks-1-train	3	0.00	0.00	0.00	0.00	0.00	1.00	1.00	0.00	0.00
monks-2-test	6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
monks-2-train	5	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20	0.20
monks-3-test	3	0.33	0.33	0.33	0.33	0.33	0.00	0.00	0.00	0.00
monks-3-train	4	0.00	0.00	0.00	0.00	0.00	0.25	0.25	0.00	0.00
nursery	8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
poker-hand-training-true	5	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00
shuttle-landing-control	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
soybean-small	2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
spect-test	8	0.13	0.13	0.13	0.13	0.13	0.13	0.50	0.25	0.13
teeth	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
tic-tac-toe	6	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17	0.17
zoo-data	4	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average		0.125	0.125	0.125	0.125	0.139	0.173	0.190	0.130	0.083

mation about cost of decision trees built by each of the considered greedy algorithms. Instead of the cost of decision tree, constructed by greedy algorithm (greedy_cost), we consider relative difference of greedy_cost and min_cost:

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

The last line shows average relative difference of greedy_cost and 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*, *weighted-sum* or *weighted-max*) and uncertainty measure (*ent*, *gini*, *me*, or *rt*).

Considering average depth, we noticed that the type *sum* dominates *max*, i.e. it has less value of average relative difference between greedy_cost and min_cost for each uncertainty measure. Similarly, the type *weighted-sum* dominates *weighted-max*. Table 1 presents results for two best types: *sum* and *weighted-sum*. One can see that the two best impurity functions are given by combinations of *weighted-sum* with Gini index (the criterion used by CART (Breiman et al., 1984)) and entropy (the criterion used by ID3 (Quinlan, 1986)).

Analysis of experiments for the number of nodes lead us to the same results. The type *sum* dominates *max* and *weighted-sum* dominates *weighted-max*. Table 2 presents results for two best types: *sum* and *weighted-sum*. One can see that along with the average depth, the two best impurity functions are given by combinations of *weighted-sum* with Gini index and entropy.

Experiments with depth lead to different ranking of impurity functions. The type *weighted-sum* dominates *sum*, and the *weighted-max* dominates *max* by each uncertainty measure with the exception of *rt*. Table 3 shows results for the best functions: combinations of *weighted-max* and *weighted-sum* with all four uncertainty measures, and the combination of *max* and *rt*.

One can see that the best impurity function is given by the combination of *weighted-sum* with *rt*. The following combinations give us similar results: *sum* and *rt*, *weighted-max* and *ent*, *weighted-max* and *gini*, *weighted-max* and *me*, and *max* and *rt*. The greedy algorithm based on the last combination is known to be close (from the point of view of accuracy) to the best approximate polynomial algorithms for minimization of decision tree depth under some assumptions about the class *NP* (Moshkov, 2005).

6 CONCLUSIONS

The paper is devoted to the study of 16 greedy algorithms for decision tree construction. For 24 data sets from UCI ML Repository (Frank and Asuncion, 2010) we compare average depth, number of nodes and depth of decision trees, constructed by these algorithms, with minimum average depth, minimum number of nodes and minimum depth found by an algorithm based on dynamic programming approach. The obtained results show that for the average depth and number of nodes the greedy algorithms used by CART (Breiman et al., 1984) and by ID3 (Quinlan, 1986) are the best among the considered greedy algorithms. However, for the minimization of depth, we probably should use some other greedy algorithms.

ACKNOWLEDGEMENTS

This work was supported by KAUST Baseline Research Fund.

REFERENCES

Ahlswede, R. and Wegener, I. (1979). *Suchprobleme*. Teubner Verlag, Stuttgart.

Alekhnovich, M., Braverman, M., Feldman, V., Klivans, A. R., and Pitassi, T. (2004). Learnability and automatizability. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 621–630, Washington, DC, USA. IEEE Computer Society.

Breiman, L. et al. (1984). *Classification and Regression Trees*. Chapman & Hall, New York.

Chakaravarthy, V. T., Pandit, V., Roy, S., Awasthi, P., and Mohania, M. (2007). Decision trees for entity identification: approximation algorithms and hardness results. In *Proceedings of the twenty-sixth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '07, pages 53–62, New York, NY, USA. ACM.

Frank, A. and Asuncion, A. (2010). UCI machine learning repository.

Garey, M. R. (1972). Optimal binary identification procedures. *SIAM Journal on Applied Mathematics*, 23(2):173–186.

Heeringa, B. and Adler, M. (2005). Approximating optimal binary decision trees. Technical Report 05-25, University of Massachusetts, Amherst.

Hyafil, L. and Rivest, R. (1976). Constructing optimal binary decision trees is np-complete. *Information Processing Letters*, 5:15–17.

Martelli, A. and Montanari, U. (1978). Optimizing decision trees through heuristically guided search. *Commun. ACM*, 21:1025–1039.

Moret, B. E., , and R. C. Gonzalez, M. T. (1980). The activity of a variable and its relation to decision trees. *ACM Trans. Program. Lang. Syst.*, 2:580–595.

Moshkov, M. J. (2005). Time complexity of decision trees. *T. Rough Sets*, 3400:244–459.

Moshkov, M. J. (2010). Greedy algorithm with weights for decision tree construction. *Fundam. Inform.*, 104(3):285–292.

Moshkov, M. J. and Chikalov, I. V. (2003). Consecutive optimization of decision trees concerning various complexity measures. *Fundamenta Informaticae*, 61(2):87–96.

Pattipati, K. R. and Dontamsetty, M. (1992). On a generalized test sequencing problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(2):392–396.

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1:81–106.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning (Morgan Kaufmann Series in Machine Learning)*. Morgan Kaufmann, 1 edition.