Sorting by Decision Trees with Hypotheses (extended abstract)

Mohammad Azad[1], Igor Chikalov[2], Shahid Hussain[3] and Mikhail Moshkov[4]

[1]*Jouf University, Sakaka 72441, Saudi Arabia*
[2]*Intel Corporation, 5000 W Chandler Blvd, Chandler, AZ 85226, USA*
[3]*Institute of Business Administration, University Road, Karachi 75270, Pakistan*
[4]*King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia*

Abstract

In this paper, we consider decision trees that use both queries based on one attribute each and queries based on hypotheses about values of all attributes. Such decision trees are similar to ones studied in exact learning, where not only membership but also equivalence queries are allowed. For $n = 3, \ldots, 6$, we compare decision trees based on various combinations of attributes and hypotheses for sorting n pairwise different elements from linearly ordered set.

Keywords

decision tree, hypothesis, dynamic programming, sorting

1. Introduction

Decision trees are widely used in many areas of computer science, for example, test theory (initiated by Chegis and Yablonskii [1]), rough set theory (initiated by Pawlak [2, 3, 4]), and exact learning (initiated by Angluin [5, 6]). These theories are closely related: attributes from rough set theory and test theory correspond to membership queries from exact learning. Exact learning also studies equivalence queries. The notion of "minimally adequate teacher" using both membership and equivalence queries was discussed by Angluin in [7]. Relations between exact learning and PAC learning proposed by Valiant [8] were considered in [5].

In [9, 10, 11], we added the notion of a hypothesis (an analog of equivalence queries) to the model considered in both rough set theory and test theory and proposed dynamic programming algorithms for the optimization of the decision trees with hypotheses. Note that the dynamic programming algorithms for the optimization of the conventional decision trees that do not use hypotheses were proposed earlier [12].

In the present paper, we consider an application of the dynamic programming algorithms from [9, 10, 11] to the study of the problem of sorting. We compare the complexity of five types of optimal (relative to the depth and relative to the number of realizable nodes) decision trees

29th International Workshop on Concurrency, Specification and Programming (CS&P'21)
mmazad@ju.edu.sa (M. Azad); igor.chikalov@gmail.com (I. Chikalov); shahidhussain@iba.edu.pk (S. Hussain); mikhail.moshkov@kaust.edu.sa (M. Moshkov)
0000-0001-9851-1420 (M. Azad); 0000-0002-1010-6605 (I. Chikalov); 0000-0002-1698-2809 (S. Hussain); 0000-0003-0085-9483 (M. Moshkov)

based on various combinations of attributes and hypotheses for sorting n pairwise different elements from linearly ordered set, $n = 3, \dots, 6$. Results obtained for the conventional decision trees are known – see book [12]. Results obtained for the decision trees with hypotheses are completely new.

Note that in the present paper we follow [11] when discuss the notions related to the decision trees with hypotheses. Complete definitions of these notions can be found in the same paper.

2. Five Types of Decision Trees and Their Optimization

Let T be a decision table with n conditional attributes $f_1, \dots, f_n$ that have values from the set $\omega = \{0, 1, 2, \dots\}$. Rows of this table are pairwise different and each row is labeled with a decision. For a given row of T, we should recognize the decision attached to it. To this end, we will use decision trees based on two types of queries. We can ask about the value of a conditional attribute $f_i \in \{f_1, \dots, f_n\}$ on the given row. As a result, we obtain an answer of the kind $f_i = \delta$, where δ is the number in the intersection of the given row and the column f_i. We can also ask if a hypothesis $\{f_1 = \delta_1, \dots, f_n = \delta_n\}$ is true, where the numbers $\delta_1, \dots, \delta_n$ belong to the columns $f_1, \dots, f_n$, respectively. Either this hypothesis is confirmed or we obtain a counterexample of the kind $f_i = \sigma$, where $f_i \in \{f_1, \dots, f_n\}$ and σ is a number from the column f_i that is different from δ_i. We will say that this hypothesis is proper if $(\delta_1, \dots, \delta_n)$ is a row of the table T.

We study the following five types of decision trees:

1. Decision trees based on attributes only.
2. Decision trees based on hypotheses only.
3. Decision trees based on both attributes and hypotheses.
4. Decision trees based on proper hypotheses only.
5. Decision trees based on both attributes and proper hypotheses.

As time complexity of a decision tree we consider its depth, which is equal to the maximum number of queries in a path from the root to a terminal node of the tree. We consider the number of realizable relative to T nodes in a decision tree as its space complexity. A node is called realizable relative to T if the computation in the tree will pass through this node for some row and some choice of counterexamples. We use the following notation:

- $h^{(k)}(T)$ denotes the minimum depth of a decision tree of the type k for T, $k = 1, \dots, 5$.
- $L^{(k)}(T)$ denotes the minimum number of nodes realizable relative to T in a decision tree of the type k for T, $k = 1, \dots, 5$.

In [9] and [10], dynamic programming algorithms for the optimization of decision trees of all five types relative to the depth and the number of realizable nodes were proposed (see also journal extension [11] of these papers that considers additionally two cost functions: the number of realizable terminal nodes and the number of nonterminal nodes). Note that algorithms for the minimization of the depth and number of nodes for decision trees of the type 1 were considered in [12] for decision tables with one-valued decisions and in [13] for decision tables with many-valued decisions.

Table 1
Experimental results for the depth

n	$h^{(1)}(T_n)$	$h^{(2)}(T_n)$	$h^{(3)}(T_n)$	$h^{(4)}(T_n)$	$h^{(5)}(T_n)$
3	3	2	2	2	2
4	5	4	4	4	4
5	7	6	6	6	6
6	10	9	9	9	9

Table 2
Experimental results for the number of realizable nodes

n	$L^{(1)}(T_n)$	$L^{(2)}(T_n)$	$L^{(3)}(T_n)$	$L^{(4)}(T_n)$	$L^{(5)}(T_n)$
3	11	13	9	14	9
4	47	253	39	254	39
5	239	15,071	199	15,142	199
6	1,439	2,885,086	1,199	2,886,752	1,199

Dynamic programming optimization algorithms are applicable to medium-sized decision tables. These algorithms first construct a directed acyclic graph (DAG) whose nodes are some subtables of the original decision table given by conditions of the type "attribute = value". Then they pass through all the nodes of the DAG, starting with the simplest subtables, and for each subtable they find the minimum value of the considered cost function.

In the present paper, we use algorithms proposed in [9, 10, 11] to study decision trees of all five types optimal relative to the depth and relative to the number of realizable nodes for the sorting problem. Results for decision trees of the type 1 were obtained earlier [12]. Results for decision trees of the types 2–5 are new.

3. Problem of Sorting

In this paper, we study the problem of sorting n elements. Let $x_1, \dots, x_n$ be pairwise different elements from a linearly ordered set. We should find a permutation $(p_1, \dots, p_n)$ from the set P_n of all permutations of the set $\{1, \dots, n\}$ for which $x_{p_1} < \dots < x_{p_n}$. To this end, we use attributes $x_i : x_j$ such that $i, j \in \{1, \dots, n\}$, $i < j$, $x_i : x_j = 1$ if $x_i < x_j$, and $x_i : x_j = 0$ if $x_i > x_j$.

The problem of sorting n elements can be represented as a decision table T_n with $n(n-1)/2$ conditional attributes $x_i : x_j$, $i, j \in \{1, \dots, n\}$, $i < j$, and $n!$ rows corresponding to permutations from P_n. For each permutation $(p_1, \dots, p_n)$, the corresponding row of T_n is labeled with this permutation as the decision. This row is filled with values of attributes $x_i : x_j$ such that $x_i : x_j = 1$ if and only if i stays before j in the tuple $(p_1, \dots, p_n)$.

For $n = 3, \dots, 6$ and $k = 1, \dots, 5$, we find values of $h^{(k)}(T_n)$ and $L^{(k)}(T_n)$ using dynamic programming algorithms described in [9, 10, 11] – see results in Tables 1 and 2.

From the obtained experimental results it follows that the decision trees of the types 2–5 can have less depth than the decision trees of the type 1. Decision trees of the types 3 and 5 can have less number of realizable nodes than the decision trees of the type 1. Decision trees of the types 2 and 4 have too many nodes.

4. Conclusions

In this paper, we found the minimum depth and the minimum number of realizable nodes of five types of decision trees for sorting n elements, $n = 3, \ldots, 6$.

In the future, we are planning to study joint behavior of the depth and the number of nodes in such decision trees. It would be also interesting to compare the complexity of optimal decision trees of the considered five types constructed by dynamic programming algorithms and the complexity of decision trees constructed by entropy-based greedy algorithm proposed in [14].

Acknowledgments

Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST). The authors are indebted to the anonymous reviewers for interesting comments.

References

[1] I. A. Chegis, S. V. Yablonskii, Logical methods of control of work of electric schemes, Trudy Mat. Inst. Steklov (in Russian) 51 (1958) 270–360.
[2] Z. Pawlak, Rough sets, Int. J. Parallel Program. 11 (1982) 341–356.
[3] Z. Pawlak, Rough Sets - Theoretical Aspects of Reasoning about Data, volume 9 of *Theory and Decision Library: Series D*, Kluwer, 1991.
[4] Z. Pawlak, A. Skowron, Rudiments of rough sets, Inf. Sci. 177 (2007) 3–27.
[5] D. Angluin, Queries and concept learning, Mach. Learn. 2 (1988) 319–342.
[6] D. Angluin, Queries revisited, Theor. Comput. Sci. 313 (2004) 175–194.
[7] D. Angluin, Learning regular sets from queries and counterexamples, Inf. Comput. 75 (1987) 87–106.
[8] L. G. Valiant, A theory of the learnable, Commun. ACM 27 (1984) 1134–1142.
[9] M. Azad, I. Chikalov, S. Hussain, M. Moshkov, Minimizing depth of decision trees with hypotheses (to appear), in: International Joint Conference on Rough Sets (IJCRS 2021), 19–24 September 2021, Bratislava, Slovakia, 2021.
[10] M. Azad, I. Chikalov, S. Hussain, M. Moshkov, Minimizing number of nodes in decision trees with hypotheses (to appear), in: 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2021), 8–10 September 2021, Szczecin, Poland, 2021.
[11] M. Azad, I. Chikalov, S. Hussain, M. Moshkov, Optimization of decision trees with hypotheses for knowledge representation, Electronics 10 (2021) 1580. URL: https://doi.org/10.3390/electronics10131580.

[12] H. AbouEisha, T. Amin, I. Chikalov, S. Hussain, M. Moshkov, Extensions of Dynamic Programming for Combinatorial Optimization and Data Mining, volume 146 of *Intelligent Systems Reference Library*, Springer, 2019.

[13] F. Alsolami, M. Azad, I. Chikalov, M. Moshkov, Decision and Inhibitory Trees and Rules for Decision Tables with Many-valued Decisions, volume 156 of *Intelligent Systems Reference Library*, Springer, 2020.

[14] M. Azad, I. Chikalov, S. Hussain, M. Moshkov, Entropy-based greedy algorithm for decision trees using hypotheses, Entropy 23 (2021) 808. URL: https://doi.org/10.3390/e23070808.