entropy

MDPI

Article

Entropy-Based Greedy Algorithm for Decision Trees Using Hypotheses

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

[1] Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka 72441, Saudi Arabia; mmazad@ju.edu.sa
[2] Intel Corporation, 5000 W Chandler Blvd, Chandler, AZ 85226, USA; igor.chikalov@gmail.com
[3] Computer Science Program, Dhanani School of Science and Engineering, Habib University, Karachi 75290, Pakistan; hussain.shahid@gmail.com
[4] Computer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
* Correspondence: mikhail.moshkov@kaust.edu.sa

Abstract: In this paper, we consider decision trees that use both conventional queries based on one attribute each and queries based on hypotheses of values of all attributes. Such decision trees are similar to those studied in exact learning, where membership and equivalence queries are allowed. We present greedy algorithm based on entropy for the construction of the above decision trees and discuss the results of computer experiments on various data sets and randomly generated Boolean functions.

Keywords: decision tree; hypothesis; greedy algorithm; entropy

Citation: Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M. Entropy-Based Greedy Algorithm for Decision Trees Using Hypotheses. *Entropy* **2021**, *23*, 808. https://doi.org/10.3390/e23070808

Academic Editors: Alessandra Palmigiano, Willem Conradie and Yiyu Yao

Received: 30 May 2021
Accepted: 22 June 2021
Published: 25 June 2021

1. Introduction

Decision trees are well known as means for knowledge representation, as classifiers, and as algorithms to solve various problems of combinatorial optimization, computational geometry, etc. [1–3].

Conventional decision trees have been studied in different theories, in particular, in rough set theory initiated by Pawlak [4–6] and in test theory initiated by Chegis and Yablonskii [7]. These trees use simple queries based on one attribute each.

In contrast to these theories, exact learning initiated by Angluin [8,9] studied not only membership queries that correspond to attributes from rough set theory and test theory, but also so-called equivalence queries. Relations between exact learning and probably approximately correct (PAC) learning proposed by Valiant [10] were discussed in [8].

In this paper, we add the notion of a hypothesis to the model that has been considered in rough set theory as well in test theory. This model allows us to use an analog of equivalence queries.

Let T be a decision table with n conditional attributes $f_1, \ldots, f_n$ having values from the set $\omega = \{0, 1, 2, \ldots\}$ in which rows are pairwise different and each row is labeled with a decision from ω. For a given row of T, we should recognize the decision attached to this row. To this end, we can use decision trees based on two types of queries. We can ask about the value of an attribute $f_i \in \{f_1, \ldots, f_n\}$ on the given row. We will 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 hypothesis $f_1 = \delta_1, \ldots, f_n = \delta_n$ is true, where $\delta_1, \ldots, \delta_n$ are numbers from the columns $f_1, \ldots, f_n$, respectively. Either this hypothesis will be confirmed or we will obtain a counterexample in the form $f_i = \sigma$, where $f_i \in \{f_1, \ldots, f_n\}$ and σ is a number from the column f_i different from δ_i. The considered hypothesis is called proper if $(\delta_1, \ldots, \delta_n)$ is a row of the table T.

In this paper, we consider two cost functions that characterize the time and space complexity of decision trees. We consider the depth of a decision tree as its time complexity,

which is equal to the maximum number of queries in a path from the root to a terminal node of the tree. As the space complexity of a decision tree, we consider the number of its realizable, relative to T, nodes. A node is called realizable relative to T if for a row of T and some choice of counterexamples the computation in the tree passes through this node.

Decision trees using hypotheses can be essentially more efficient than the decision trees using only attributes. Let us consider an example, the problem of computation of the conjunction $x_1 \wedge \cdots \wedge x_n$. The minimum depth of a decision tree solving this problem using the attributes $x_1, \ldots, x_n$ is equal to n. The minimum number of realizable nodes in such decision trees is equal to $2n + 1$. However, the minimum depth of a decision tree solving this problem using proper hypotheses is equal to 1: it is enough to ask only about the hypothesis $x_1 = 1, \ldots, x_n = 1$. If it is true, then the considered conjunction is equal to 1. Otherwise, it is equal to 0. The obtained decision tree contains $n + 2$ realizable nodes.

In this paper, we consider the following five types of decision trees:

1. Decision trees that use only attributes.
2. Decision trees that use only hypotheses.
3. Decision trees that use both attributes and hypotheses.
4. Decision trees that use only proper hypotheses.
5. Decision trees that use both attributes and proper hypotheses.

There are different ways to construct conventional decision trees, including algorithms that can construct optimal decision trees for medium-sized decision tables [11–15]. In particular, in [16,17], we proposed dynamic programming algorithms for the minimization of the depth and number of realizable nodes in decision trees with hypotheses. However, the most common way is to use greedy algorithms [1,18].

In this paper, we propose a greedy algorithm based on entropy that, for a given decision table and type of decision trees, constructs a decision tree of the considered type for this table. The goal of this paper is to understand which type of decision tree should be chosen if we would like to minimize the depth and which type should be chosen if we would like to minimize the number of realizable nodes. To this end, we compare the parameters of the constructed decision trees of five types for 10 decision tables from the UCI ML Repository [19]. We do the same for randomly generated Boolean functions with n variables, where $n = 3, \ldots, 6$. From the obtained experimental results, it follows that we should choose decision trees of the type 3 if we would like to minimize the depth and decision trees of the type 1 if we would like to minimize the number of realizable nodes.

The main contributions of the paper are (i) the design of the extended entropy-based greedy algorithm that can work with five types of decision trees and (ii) the understanding of which type of decision trees should be chosen when we would like to minimize the depth or the number of realizable nodes.

The rest of the paper is organized as follows. In Sections 2 and 3, we consider the main notions and in Section 4, the greedy algorithm for the decision tree construction. Section 5 contains the results of the computer experiments and Section 6, short conclusions.

2. Decision Tables

A decision table is a rectangular table T with $n \geq 1$ columns filled with numbers from the set $\omega = \{0, 1, 2, \ldots\}$ of non-negative integers. Columns of this table are labeled with the conditional attributes $f_1, \ldots, f_n$. Rows of the table are pairwise different. Each row is labeled with a number from ω that is interpreted as a decision. Rows of the table are interpreted as tuples of values of the conditional attributes.

A decision table T is called empty if it has no rows. The table T is called degenerate if it is empty or all rows of T are labeled with the same decision.

We denote $F(T) = \{f_1, \ldots, f_n\}$ and denote by $D(T)$ the set of decisions attached to the rows of T. For any conditional attribute $f_i \in F(T)$, we denote by $E(T, f_i)$ the set of values of the attribute f_i in the table T.

A system of equations over T is an arbitrary equation system of the following kind:

$$\{f_{i_1} = \delta_1, \ldots, f_{i_m} = \delta_m\},$$

where $m \in \omega$, $f_{i_1}, \ldots, f_{i_m} \in F(T)$, and $\delta_1 \in E(T, f_{i_1}), \ldots, \delta_m \in E(T, f_{i_m})$ (if $m = 0$, then the considered equation system is empty).

Let T be a nonempty table. A subtable of T is a table obtained from T by the removal of some rows. We correspond to each equation system S over T a subtable TS of the table T. If the system S is empty, then $TS = T$. Let S be nonempty and $S = \{f_{i_1} = \delta_1, \ldots, f_{i_m} = \delta_m\}$. Then TS is the subtable of the table T containing the rows from T, which in the intersection with the columns $f_{i_1}, \ldots, f_{i_m}$ have numbers $\delta_1, \ldots, \delta_m$, respectively.

3. Decision Trees

Let T be a nonempty decision table with n conditional attributes $f_1, ..., f_n$. We consider the decision trees with two types of queries. We can choose an attribute $f_i \in F(T) = \{f_1, \ldots, f_n\}$ and ask about its value. This query has the set of answers $A(f_i) = \{\{f_i = \delta\} : \delta \in E(T, f_i)\}$. We can formulate a hypothesis over T in the form of $H = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$, where $\delta_1 \in E(T, f_1), \ldots, \delta_n \in E(T, f_n)$, and ask about this hypothesis. This query has the set of answers $A(H) = \{H, \{f_1 = \sigma_1\}, \ldots, \{f_n = \sigma_n\} : \sigma_1 \in E(T, f_1) \setminus \{\delta_1\}, \ldots, \sigma_n \in E(T, f_n) \setminus \{\delta_n\}\}$. The answer H means that the hypothesis is true. Other answers are counterexamples. The hypothesis H is called proper for T if $(\delta_1, \ldots, \delta_n)$ is a row of the table T.

A decision tree over T is a marked finite directed tree with the root in which the following is true:

- Each terminal node is labeled with a number from the set $D(T) \cup \{0\}$.
- Each node, which is not terminal (such nodes are called working), is labeled with an attribute from the set $F(T)$ or with a hypothesis over T.
- If a working node is labeled with an attribute f_i from $F(T)$, then for each answer from the set $A(f_i)$, there is exactly one edge labeled with this answer, which leaves this node and there are no any other edges that leave this node.
- If a working node is labeled with a hypothesis $H = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ over T, then for each answer from the set $A(H)$, there is exactly one edge labeled with this answer, which leaves this node and there are no any other edges that leave this node.

Let Γ be a decision tree over T and v be a node of Γ. We now define an equation system $S(\Gamma, v)$ over T associated with the node v. We denote by ξ the directed path from the root of Γ to the node v. If there are no working nodes in ξ, then $S(\Gamma, v)$ is the empty system. Otherwise, $S(\Gamma, v)$ is the union of equation systems attached to the edges of the path ξ.

A decision tree Γ over T is called a decision tree for T if for any node v of Γ, the following is true:

- The node v is terminal if and only if the subtable $TS(\Gamma, v)$ is degenerate.
- If v is a terminal node and the subtable $TS(\Gamma, v)$ is empty, then the node v is labeled with the decision 0.
- If v is a terminal node and the subtable $TS(\Gamma, v)$ is nonempty, then the node v is labeled with the decision attached to all rows of $TS(\Gamma, v)$.

A complete path in Γ is an arbitrary directed path from the root to a terminal node in Γ. As the time complexity of a decision tree, we consider its depth, which is the maximum number of working nodes in a complete path in the tree, or which is the same—the maximum length of a complete path in the tree. We denote by $h(\Gamma)$ the depth of the decision tree Γ.

As the space complexity of the decision tree Γ, we consider the number of its realizable relative to T nodes. A node v of Γ is called realizable relative to T if and only if the subtable $TS(\Gamma, v)$ is nonempty. We denote by $L(T, \Gamma)$ the number of nodes in Γ that are realizable relative to T.

In this paper, we consider the following five types of decision trees:

1. Decision trees that use only attributes.
2. Decision trees that use only hypotheses.
3. Decision trees that use both attributes and hypotheses.
4. Decision trees that use only proper hypotheses.
5. Decision trees that use both attributes and proper hypotheses.

4. Greedy Algorithm Based on Entropy

Let T be a nonempty decision table with n conditional attributes $f_1, \ldots, f_n$ and Θ be a subtable of the table T. We define entropy of Θ (denoted $ent(\Theta)$) as follows. If Θ is empty, then $ent(\Theta) = 0$. Let Θ be nonempty. For any decision $t \in D(\Theta)$, we denote by $N_t(\Theta)$ the number of rows in Θ labeled with the decision t and by p_t the value $N_t(\Theta)/N(\Theta)$, where $N(\Theta)$ is the number of rows in Θ. Then,

$$ent(\Theta) = -\sum_{t \in D(T)} p_t \log_2 p_t.$$

We now define the impurity of a query for the table Θ. The impurity of the query based on an attribute $f_i \in F(T)$ (impurity of query f_i) is equal to $I(f_i, \Theta) = \max\{ent(\Theta S) : S \in A(f_i)\}$. The impurity of the query based on a hypothesis H over T (impurity of query H) is equal to $I(H, \Theta) = \max\{ent(\Theta S) : S \in A(H)\}$.

We can find by simple search among all attributes from $F(T)$ an attribute f_i with the minimum impurity $I(f_i, \Theta)$. We can also find by simple search among all proper hypotheses over T a proper hypothesis H with the minimum impurity $I(H, \Theta)$. It is not necessary to consider all hypotheses over T to find a hypothesis with the minimum impurity. For $i = 1, \ldots, n$, we denote by δ_i a number from $E(T, f_i)$ such that $ent(\Theta\{f_i = \delta_i\}) = \max\{ent(\Theta\{f_i = \sigma\}) : \sigma \in E(T, f_i)\}$. Then, the hypothesis $H = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ has the minimum impurity $I(H, \Theta)$ among all hypotheses over T.

We now describe a greedy algorithm $\mathcal{E}$ based on entropy that, for a given nonempty decision table T and $k \in \{1, \ldots, 5\}$, constructs a decision tree of the type k for the table T.

The considered algorithm is similar to standard top-down induction of decision trees [20,21]. The main peculiarity of this algorithm is step 5 in which, depending on the value of k, we choose an appropriate set of queries. Every time, the algorithm chooses a query from this set with the minimum impurity.

For a given nonempty decision table T and number $k \in \{1, \ldots, 5\}$, the algorithm 1 $\mathcal{E}$ constructs a decision tree of the type k for the table T. We denote by $h_{\mathcal{E}}^{(k)}(T)$ the depth of this decision tree and by $L_{\mathcal{E}}^{(k)}(T)$, we denote the number of realizable relative to T nodes in this tree.

Algorithm 1 $\mathcal{E}$.

Input: A nonempty decision table T and a number $k \in \{1, \ldots, 5\}$.
Output: A decision tree of the type k for the table T.

1. Construct a tree G consisting of a single node labeled with T.
2. If no node of the tree G is labeled with a table, then the algorithm ends and returns the tree G.
3. Choose a node v in G, which is labeled with a subtable Θ of the table T.
4. If Θ is degenerate, then instead of Θ, we label the node v with 0 if Θ is empty and with the decision attached to each row of Θ if Θ is nonempty.
5. If Θ is nondegenerate, then depending on k, we choose a query X (either attribute or hypothesis) in the following way:
   (a) If $k = 1$, then we find an attribute $X \in F(T)$ with the minimum impurity $I(X, \Theta)$.
   (b) If $k = 2$, then we find a hypothesis X over T with the minimum impurity $I(X, \Theta)$.
   (c) If $k = 3$, then we find an attribute $Y \in F(T)$ with the minimum impurity $I(Y, \Theta)$ and a hypothesis Z over T with the minimum impurity $I(Z, \Theta)$. Between Y and Z, we choose a query X with the minimum impurity $I(X, \Theta)$.
   (d) If $k = 4$, then we find a proper hypothesis X over T with the minimum impurity $I(X, \Theta)$.
   (e) If $k = 5$, then we find an attribute $Y \in F(T)$ with the minimum impurity $I(Y, \Theta)$ and a proper hypothesis Z over T with the minimum impurity $I(Z, \Theta)$. Between Y and Z, we choose a query X with the minimum impurity $I(X, \Theta)$.
6. Instead of Θ, we label the node v with the query X. For each answer $S \in A(X)$, we add to the tree G a node $v(S)$ and an edge $e(S)$ connecting v and $v(S)$. We label the node $v(S)$ with the subtable ΘS and label the edge $e(S)$ with the answer S. We then proceed to step 2.

5. Results of Experiments

We make experiments with 10 decision tables from the UCI ML Repository [19]. Table 1 contains the information about each of these decision tables: its name, the number of rows, and the number of conditional attributes.

Table 1. Decision tables from [19] used in experiments.

Decision Table	Number of Rows	Number of Attributes
BALANCE-SCALE	625	5
BREAST-CANCER	266	10
CARS	1728	7
HAYES-ROTH-DATA	69	5
LYMPHOGRAPHY	148	18
NURSERY	12,960	9
SOYBEAN-SMALL	47	36
SPECT-TEST	169	22
TIC-TAC-TOE	958	10
ZOO-DATA	59	17

The results of the experiments are represented in Tables 2 and 3. The first column of Table 2 contains the name of the considered decision table T. The last five columns contain values $h_{\mathcal{E}}^{(1)}(T), \ldots, h_{\mathcal{E}}^{(5)}(T)$ (minimum values for each decision table are in bold).

Table 2. Experimental results for decision tables from [19] (depth).

Decision Table T	$h_{\mathcal{E}}^{(1)}(T)$	$h_{\mathcal{E}}^{(2)}(T)$	$h_{\mathcal{E}}^{(3)}(T)$	$h_{\mathcal{E}}^{(4)}(T)$	$h_{\mathcal{E}}^{(5)}(T)$
BALANCE-SCALE	**4**	**4**	**4**	**4**	**4**
BREAST-CANCER	9	9	9	**8**	9
CARS	**6**	**6**	**6**	**6**	**6**
HAYES-ROTH-DATA	**4**	**4**	**4**	**4**	**4**
LYMPHOGRAPHY	11	11	**9**	13	10
NURSERY	**8**	**8**	**8**	**8**	**8**
SOYBEAN-SMALL	**2**	6	**2**	8	**2**
SPECT-TEST	20	**5**	**5**	14	11
TIC-TAC-TOE	**7**	8	**7**	8	**7**
ZOO-DATA	8	6	**5**	8	**5**
Average	7.9	6.7	5.9	8.1	6.6

The first column of Table 3 contains the name of the considered decision table T. The last five columns contain values $L_{\mathcal{E}}^{(1)}(T), \ldots, L_{\mathcal{E}}^{(5)}(T)$ (minimum values for each decision table are in bold).

Table 3. Experimental results for decision tables from [19] (number of realizable nodes).

Decision Table T	$L_{\mathcal{E}}^{(1)}(T)$	$L_{\mathcal{E}}^{(2)}(T)$	$L_{\mathcal{E}}^{(3)}(T)$	$L_{\mathcal{E}}^{(4)}(T)$	$L_{\mathcal{E}}^{(5)}(T)$
BALANCE-SCALE	**556**	5234	4102	5234	4102
BREAST-CANCER	**255**	446,170	304	10,3642	266
CARS	**1136**	65,624	3944	65,624	3944
HAYES-ROTH-DATA	73	**72**	**72**	367	**72**
LYMPHOGRAPHY	**123**	6,653,366	162	8,515,841	153
NURSERY	**4460**	12,790,306	14,422	12,790,306	14,422
SOYBEAN-SMALL	**7**	10,029	**7**	157,640	**7**
SPECT-TEST	**123**	6983	5495	398,926	1116
TIC-TAC-TOE	**648**	864,578	200,847	946,858	940
ZOO-DATA	**33**	2134	35	13,310	35
Average	741.4	2,084,449.6	22,939	2,299,774.8	2505.7

For $n = 3, \ldots, 6$, we randomly generate 100 Boolean functions with n variables. We represent each Boolean function with n variables as a decision table with n columns labeled with these variables and with 2^n rows that are all possible n-tuples of values of the variables. Each row is labeled with the decision that is the value of the function on the corresponding n-tuple.

For each function, using its decision table representation and the algorithm $\mathcal{E}$, we construct a decision tree of the type k computing this function, $k = 1, \ldots, 5$. For each Boolean function, each hypothesis over the decision table representing it is proper. Therefore, for each Boolean function, $h_{\mathcal{E}}^{(2)} = h_{\mathcal{E}}^{(4)}$, $h_{\mathcal{E}}^{(3)} = h_{\mathcal{E}}^{(5)}$, $L_{\mathcal{E}}^{(2)} = L_{\mathcal{E}}^{(4)}$, and $L_{\mathcal{E}}^{(3)} = L_{\mathcal{E}}^{(5)}$.

The results of the experiments are represented in Tables 4 and 5. The first column in Table 4 contains the number of variables n in the considered Boolean functions. The last five columns contain information about values $h_{\mathcal{E}}^{(1)}, \ldots, h_{\mathcal{E}}^{(5)}$ in the format ${}_{min}Avg_{max}$.

Table 4. Experimental results for Boolean functions (depth).

Number of Variables n	$h_{\mathcal{E}}^{(1)}$	$h_{\mathcal{E}}^{(2)}$	$h_{\mathcal{E}}^{(3)}$	$h_{\mathcal{E}}^{(4)}$	$h_{\mathcal{E}}^{(5)}$
3	${}_{0}2.94_{3}$	${}_{0}2.02_{3}$	${}_{0}1.86_{3}$	${}_{0}2.02_{3}$	${}_{0}1.86_{3}$
4	${}_{4}4.00_{4}$	${}_{2}3.05_{4}$	${}_{2}2.97_{3}$	${}_{2}3.05_{4}$	${}_{2}2.97_{3}$
5	${}_{5}5.00_{5}$	${}_{4}4.11_{5}$	${}_{3}3.99_{4}$	${}_{4}4.11_{5}$	${}_{3}3.99_{4}$
6	${}_{6}6.00_{6}$	${}_{5}5.09_{6}$	${}_{5}5.00_{5}$	${}_{5}5.09_{6}$	${}_{5}5.00_{5}$

The first column in Table 5 contains the number of variables n in the considered Boolean functions. The last five columns contain information about values $L_{\mathcal{E}}^{(1)}, \ldots, L_{\mathcal{E}}^{(5)}$ in the format ${}_{min}Avg_{max}$.

Table 5. Experimental results for Boolean functions (number of realizable nodes).

Number of Variables n	$L_{\mathcal{E}}^{(1)}$	$L_{\mathcal{E}}^{(2)}$	$L_{\mathcal{E}}^{(3)}$	$L_{\mathcal{E}}^{(4)}$	$L_{\mathcal{E}}^{(5)}$
3	${}_{1}9.60_{15}$	${}_{1}12.33_{22}$	${}_{1}9.61_{15}$	${}_{1}12.33_{22}$	${}_{1}9.61_{15}$
4	${}_{15}21.02_{29}$	${}_{14}44.75_{70}$	${}_{11}27.69_{58}$	${}_{14}44.75_{70}$	${}_{11}27.69_{58}$
5	${}_{31}42.54_{51}$	${}_{125}218.05_{292}$	${}_{25}70.19_{176}$	${}_{125}218.05_{292}$	${}_{25}70.19_{176}$
6	${}_{69}86.30_{101}$	${}_{649}1171.03_{1538}$	${}_{75}292.99_{807}$	${}_{649}1171.03_{1538}$	${}_{75}292.99_{807}$

From the obtained experimental results, it follows that, using hypotheses, we can decrease the depth of the constructed decision trees. However, at the same time, the number of realizable nodes usually grows. Depending on our goals, we should choose decision trees of type 3 if we would like to minimize the depth and decision trees of type 1 if we would like to minimize the number of realizable nodes.

6. Conclusions

In this paper, we studied modified decision trees that use both queries based on one attribute each and queries based on hypotheses about the values of all attributes. We proposed an entropy-based greedy algorithm for the construction of such decision trees and considered the results of computer experiments. The goal of this paper is to understand which type of decision tree should be chosen if we would like to minimize the depth and which type should be chosen if we would like to minimize the number of realizable nodes. From the obtained experimental results, it follows that we should choose decision trees of type 3 if we would like to minimize the depth and decision trees of type 1 if we would like to minimize the number of realizable nodes. The comparison of different greedy algorithms based on various uncertainty measures will be done in future papers.

Author Contributions: Conceptualization, all authors; methodology, all authors; software, I.C.; validation, I.C., M.A. and S.H.; formal analysis, all authors; investigation, M.A. and S.H.; resources, all authors; data curation, M.A. and S.H.; writing—original draft preparation, M.M.; writing—review and editing, all authors; visualization, M.A. and S.H.; supervision, I.C. and M.M.; project administration, M.M.; funding acquisition, M.M. All authors have read and agreed to the published version of the manuscript.

Funding: Research funded by King Abdullah University of Science and Technology.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data available in a publicly accessible repository that does not issue DOIs. Publicly available data sets were analyzed in this study. These data can be found here: http://archive.ics.uci.edu/ml accessed date: 12 April 2017.

Acknowledgments: Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST), including the provision of computing resources. The authors are greatly indebted to the anonymous reviewers for useful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Chapman and Hall/CRC: Boca Raton, FL, USA, 1984.
2. Moshkov, M. Time complexity of decision trees. In *Transactions on Rough Sets III*; Lecture Notes in Computer Science; Peters, J.F., Skowron, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3400, pp. 244–459.
3. Rokach, L.; Maimon, O. *Data Mining with Decision Trees—Theory and Applications*; Series in Machine Perception and Artificial Intelligence; World Scientific: Singapore, 2007; Volume 69.
4. Pawlak, Z. Rough sets. *Int. J. Parallel Program.* **1982**, *11*, 341–356. [CrossRef]
5. Pawlak, Z. *Rough Sets—Theoretical Aspects of Reasoning about Data*; Theory and Decision Library: Series D; Kluwer: Alfen am Rhein, The Netherlands, 1991; Volume 9.
6. Pawlak, Z.; Skowron, A. Rudiments of rough sets. *Inf. Sci.* **2007**, *177*, 3–27. [CrossRef]
7. Chegis, I.A.; Yablonskii, S.V. Logical methods of control of work of electric schemes. *Trudy Mat. Inst. Steklov* **1958**, *51*, 270–360. (In Russian)
8. Angluin, D. Queries and concept learning. *Mach. Learn.* **1988**, *2*, 319–342. [CrossRef]
9. Angluin, D. Queries revisited. *Theor. Comput. Sci.* **2004**, *313*, 175–194. [CrossRef]
10. Valiant, L.G. A theory of the learnable. *Commun. ACM* **1984**, *27*, 1134–1142. [CrossRef]
11. AbouEisha, H.; Amin, T.; Chikalov, I.; Hussain, S.; Moshkov, M. *Extensions of Dynamic Programming for Combinatorial Optimization and Data Mining*; Intelligent Systems Reference Library; Springer: Berlin/Heidelberg, Germany, 2019; Volume 146
12. Aglin, G.; Nijssen, S.; Schaus, P. Learning optimal decision trees using caching branch-and-bound search. In Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, 7–12 February 2020; pp. 3146–3153.
13. Alsolami, F.; Azad, M.; Chikalov, I.; Moshkov, M. *Decision and Inhibitory Trees and Rules for Decision Tables with Many-Valued Decisions*; Intelligent Systems Reference Library; Springer: Berlin/Heidelberg, Germany, 2020; Volume 156.
14. Narodytska, N.; Ignatiev, A.; Pereira, F.; Marques-Silva, J. Learning optimal decision trees with SAT. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, 13–19 July 2018; pp. 1362–1368.
15. Verwer, S.; Zhang, Y. Learning optimal classification trees using a binary linear program formulation. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, Honolulu, HI, USA, 27 January–1 February 2019; pp. 1625–1632.
16. Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M. Minimizing depth of decision trees with hypotheses (to appear). In Proceedings of the International Joint Conference on Rough Sets (IJCRS 2021), Bratislava, Slovakia, 19–24 September 2021.
17. Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M. Minimizing number of nodes in decision trees with hypotheses (to appear). In Proceedings of the 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2021), Szczecin, Poland, 8–10 September 2021.
18. Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: Burlington, MA, USA, 1993.
19. Dua, D.; Graff, C. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences. 2017. Available online: http://archive.ics.uci.edu/ml (accessed on 12 April 2017).
20. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
21. Rokach, L.; Maimon, O. Top-down induction of decision trees classifiers—A survey. *IEEE Trans. Syst. Man Cybern. Part C* **2005**, *35*, 476–487. [CrossRef]