entropy

MDPI

Article

Decision Rules Derived from Optimal Decision Trees with Hypotheses

Mohammad Azad [1], Igor Chikalov [2], Shahid Hussain [3], Mikhail Moshkov [4,*] and Beata Zielosko [5]

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 Department of Computer Science, School of Mathematics and Computer Science, Institute of Business Administration, University Road, Karachi 75270, Pakistan; shahidhussain@iba.edu.pk
4 Computer, Electrical and Mathematical Sciences & Engineering Division, King Abdullah University of Science and Technology (KAUST), Thuwal 23955-6900, Saudi Arabia
5 Institute of Computer Science, Faculty of Science and Technology, University of Silesia in Katowice, Będzińska 39, 41-200 Sosnowiec, Poland; beata.zielosko@us.edu.pl
* Correspondence: mikhail.moshkov@kaust.edu.sa

Abstract: Conventional decision trees use queries each of which is based on one attribute. In this study, we also examine decision trees that handle additional queries based on hypotheses. This kind of query is similar to the equivalence queries considered in exact learning. Earlier, we designed dynamic programming algorithms for the computation of the minimum depth and the minimum number of internal nodes in decision trees that have hypotheses. Modification of these algorithms considered in the present paper permits us to build decision trees with hypotheses that are optimal relative to the depth or relative to the number of the internal nodes. We compare the length and coverage of decision rules extracted from optimal decision trees with hypotheses and decision rules extracted from optimal conventional decision trees to choose the ones that are preferable as a tool for the representation of information. To this end, we conduct computer experiments on various decision tables from the UCI Machine Learning Repository. In addition, we also consider decision tables for randomly generated Boolean functions. The collected results show that the decision rules derived from decision trees with hypotheses in many cases are better than the rules extracted from conventional decision trees.

Keywords: decision rule; decision tree; representation of information; hypothesis

Citation: Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M.; Zielosko, B. Decision Rules Derived from Optimal Decision Trees with Hypotheses. *Entropy* **2021**, *23*, 1641. https://doi.org/10.3390/e23121641

Academic Editors: Alessandra Palmigiano, Yiyu Yao, Willem Conradie and Stanisław Drożdż

Received: 18 September 2021
Accepted: 2 December 2021
Published: 7 December 2021

1. Introduction

Decision trees are commonly used as classifiers, as an algorithmic tool for solving various problems, and a means of representing information [1–3]. They form a part of statistical learning, which refers to a vast set of tools for understanding data [4]. Conventional decision trees studied in test theory [5], rough set theory [6–8], and many other areas of computer science exploit queries based on a single attribute. In [9–12], we considered decision trees that also exploit queries based on hypotheses. Such decision trees are analogous to the tools that have been analyzed in exact learning [13–15], where both membership and equivalence queries are used.

In the present paper, we analyze decision trees with hypotheses as a means for representation of information. We design dynamic programming algorithms to optimize such trees corresponding to two cost functions. For various decision tables, we build optimal decision trees and analyze the length and coverage of decision rules extracted from the constructed trees to study which kinds of decision trees are more suitable for the representation of information.

Let us have a decision table T that contains n attributes. We can use two types of queries in the decision trees for this table. We can ask about the value of an attribute. As a

result, we obtain this value. We can choose an n-tuple of possible values of attributes and formulate a hypothesis that it is really the tuple of values of the considered attributes. As a result, we either obtain confirmation of the hypothesis or a counterexample. We call this hypothesis proper if the considered n-tuple is a row of the table T.

We studied the following five types of decision trees:

1. Using attributes.
2. Using hypotheses.
3. Using both attributes and hypotheses.
4. Using proper hypotheses.
5. Using attributes as well as proper hypotheses.

We analyzed four different cost functions for the decision trees: the depth, the number of realizable nodes, the number of realizable leaf nodes, and the number of internal nodes. We define a node as realizable relative to a given decision table if a computation can pass through this node for at least one row of the considered decision table.

Previously, we proposed a dynamic programming algorithm in [12] for each of these four cost functions. When we give a decision table and a type of decision tree to this algorithm, it returns the minimum cost of a decision tree of a given type for the given table. The results of the computer experiments show that decision trees with hypotheses can have less complexity than conventional decision trees. It means that they can be used as a means for the representation of information.

The present paper has two aims. The first aim is to construct optimal decision trees with hypotheses. We know that such trees can be used for the representation of information (especially decision trees of type 3). However, the algorithms from [12] were designed only to find the complexity of optimal trees. The algorithms for the two cost functions (the depth and the number of internal nodes) can be modified to build optimal decision trees. Unfortunately, we cannot use a similar approach to build optimal decision trees of types 2 and 3 relative to the number of realizable nodes and optimal decision trees of type 2 relative to the number of realizable leaf nodes.

The second aim is to study the length and coverage of decision rules extracted from the optimal decision trees. Decision rules can be considered one of the simplest and most understandable models for the representation of information. Deriving decision rules from decision trees is a well-known approach. We want to confirm that the decision rules derived from decision trees with hypotheses can be better than the rules derived from conventional decision trees.

For computer experiments, we chose eight decision tables from the UCI ML Repository [16] as well as 100 randomly generated Boolean functions that contain n variables ($n = 3, \ldots, 6$). We constructed optimal (relative to the depth or to the number of internal nodes) decision trees of five types for these tables. Then we analyzed the length and coverage of decision rules extracted from these trees. For a decision tree with hypotheses for some rows of the considered decision table, it can be more than one derived decision rule that covers the row. In this case, for each row we chose the best rule. The results of the computer experiments show that the decision rules derived from the decision trees with hypotheses, in many cases, are better than the ones derived from conventional decision trees.

The novelty of the paper is directly related to its two main contributions: (i) the modification of dynamic programming algorithms described in [12] such that the modified algorithms can now construct optimal decision trees of five types relative to two cost functions and (ii) the experimental confirmation that the decision rules derived from the decision trees with hypotheses can be more suitable for the representation of information than the decision rules derived from conventional decision trees.

To make the paper more understandable, we add to it slightly modified definitions and one algorithm from [12].

We present the remaining parts of the paper as follows: important notions in Sections 2 and 3, the decision tree optimization based on dynamic programming algorithms in Sections 4–6, experimental results in Section 7, and short conclusions in Section 8.

2. Decision Tables

We can define a decision table T as follows:

- It is a rectangular table that contains n ($n \geq 1$) columns.
- Its columns are tagged by conditional attributes $f_1, \ldots, f_n$.
- Its columns' values are from the set $\omega = \{0, 1, 2, \ldots\}$.
- Its rows are unique.
- Its rows are tagged by numbers from ω interpreted as decisions.
- Its rows are considered as tuples of values of the conditional attributes.

When a decision table does not have any rows, then we call it an empty table. We define a degenerate table as a decision table which is either empty or has all of its rows tagged by the same decision.

Furthermore, we consider the following notation for T:

- $F(T)$ is the set of conditional attributes, i.e., $F(T) = \{f_1, \ldots, f_n\}$.
- $D(T)$ is the set of decisions that are attached to rows.
- $E(T, f_i)$ is the set of f_i's values where $f_i \in F(T)$.
- $E(T)$ is the set of conditional attributes of T for which $|E(T, f_i)| \geq 2$.

Let $S = \{f_{i_1} = \delta_1, \ldots, f_{i_m} = \delta_m\}$ be a system of equations where $m \in \omega$, $f_{i_1}, \ldots, f_{i_m} \in F(T)$, and $\delta_1 \in E(T, f_{i_1}), \ldots, \delta_m \in E(T, f_{i_m})$ (S is empty when $m = 0$). We denote by TS the subtable of T consisting of all rows of T that have values $\delta_1, \ldots, \delta_m$ when they intersect with the columns $f_{i_1}, \ldots, f_{i_m}$. Such subtables are called separable subtables of T.

3. Decision Trees and Rules

In this section, we define notions of decision trees and rules related with a given nonempty decision table T that contains n conditional attributes $f_1, \ldots, f_n$. Let us consider the decision trees in connection with two types of queries. The first type of query is to ask the value of an attribute $f_i \in F(T) = \{f_1, \ldots, f_n\}$. The answer of this query is from the set $A(f_i) = \{\{f_i = \delta\} : \delta \in E(T, f_i)\}$. The second type of query is to ask about a hypothesis over T in the form of $H = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ where $\delta_1 \in E(T, f_1), \ldots, \delta_n \in E(T, f_n)$. The answer of this query is from the set $A(H) = \{H, \{f_1 = \sigma_1\}, \ldots, \{f_n = \sigma_n\} : \sigma_1 \in E(T, f_1) \setminus \{\delta_1\}, ..., \sigma_n \in E(T, f_n) \setminus \{\delta_n\}\}$. If the answer is H, then the hypothesis is true. Other answers are counterexamples. Note that H is a proper hypothesis for T if $(\delta_1, \ldots, \delta_n)$ is a row of the table T.

A decision tree over T is a tagged finite directed rooted tree, where the following hold:

- We label each leaf node by a number from the set $D(T) \cup \{0\}$.
- We label each internal node by a hypothesis over T or an attribute from the set $F(T)$. In both cases, there is exactly one edge leaving this node for each answer, either from the set $A(H)$ in the case of hypothesis query or from the set $A(f_i)$ in the case of attribute query, and no other edges exit from this node.

Let us consider a decision tree Γ over T. If v is a node of Γ, then we define an equation system $S(\Gamma, v)$ over T corresponding to the node v. We denote the directed path from the root of Γ to the node v as ξ. When ξ does not have any internal nodes, then $S(\Gamma, v)$ is the empty system. On the other side, if it has internal nodes, then $S(\Gamma, v)$ is the union of the systems of equation attached to the edges in ξ.

We consider a decision tree Γ over T as a decision tree for T if, for any node v of Γ, the following hold:

- When the node v is a leaf node, then the subtable $TS(\Gamma, v)$ is degenerate and vice versa.
- When v is a leaf node and the subtable $TS(\Gamma, v)$ is empty, then we label the node v by the decision 0.
- When v is a leaf node and the subtable $TS(\Gamma, v)$ is nonempty, then we label the node v by the decision attached to all rows of $TS(\Gamma, v)$.

An arbitrary directed path ξ from the root to a leaf node v in Γ is called a complete path in Γ. Denote $T(\xi) = TS(\Gamma, v)$.

The depth of a decision tree Γ is analogous to its time complexity. We denote its depth by $h(\Gamma)$, which is defined as the maximum number of internal nodes in a complete path in the tree. Similarly, the number of internal nodes in a decision tree Γ (denoted by $L_w(\Gamma)$) is analogous to its space complexity.

Let Γ be a decision tree for T, ξ be a complete path in Γ such that $T(\xi)$ is a nonempty table, and the leaf node of the path ξ be tagged with the decision d. We now define a system of equations $S(\xi)$. $S(\xi)$ is the empty system in the case of no internal nodes in ξ. Let us assume now that ξ contains at least one internal node. We now transform systems of equations attached to edges leaving internal nodes of ξ. If an edge is tagged with an equation system containing exactly one equation, then we not change this system. Let an edge e leaving a internal node v be tagged with an equation system containing more than one equation. Then v is tagged with a hypothesis H and e is tagged with the equation system H. (Note that if such a node exists, then it is the last internal node in the complete path ξ.) In this case, we remove from the equation system H attached to e all equations of the kind $f_j = \sigma$ such that $f_j \notin E(TS(\Gamma, v))$. Then, we can obtain $S(\xi)$ as the union of new equation systems corresponding to edges in the path ξ. One can show that $T(\xi) = TS(\xi)$.

We correspond to the complete path ξ the decision rule,

$$\bigwedge_{f_i=\delta \in S(\xi)} (f_i = \delta) \to d.$$

We denote this rule by $rule(\xi)$. The number of equations in the equation system $S(\xi)$ is called the length of the rule $rule(\xi)$ and is denoted $l(rule(\xi))$. The number of rows in the subtable $TS(\xi)$ is called the coverage of the rule $rule(\xi)$ and is denoted $c(rule(\xi))$.

Denote $\Xi(T, \Gamma)$ the set of complete paths ξ in Γ such that the table $T(\xi)$ is nonempty and $Rows(T)$ the set of rows of the decision table T. For a row $r \in Rows(T)$, we denote by $l(r, T, \Gamma)$ the minimum length of a rule $rule(\xi)$ such that $\xi \in \Xi(T, \Gamma)$ and r is a row of the subtable $TS(\xi)$, and we denote by $c(r, T, \Gamma)$ the maximum coverage of a rule $rule(\xi)$ such that $\xi \in \Xi(T, \Gamma)$ and r is a row of the subtable $TS(\xi)$.

We use the following notation:

$$\begin{aligned} l(T, \Gamma) &= \frac{\sum_{r \in Rows(T)} l(r, T, \Gamma)}{|Rows(T)|}, \\ c(T, \Gamma) &= \frac{\sum_{r \in Rows(T)} c(r, T, \Gamma)}{|Rows(T)|}. \end{aligned}$$

4. Construction of Directed Acyclic Graph $\Delta(T)$

Let us consider a nonempty decision table T that has n conditional attributes $f_1, \ldots, f_n$. The Algorithm 1 $\mathcal{A}_{DAG}$ is used for the construction of a directed acyclic graph (DAG) $\Delta(T)$. Consequently, this DAG is used for the construction of optimal decision trees. Some separable subtables of the table T are the nodes of this DAG. We process one node during each iteration of the algorithm. We begin by the graph consisting of unprocessed one node T and end by processing all nodes of the graph. This algorithm was described and used in [9,10,12]. It is a special version of the more general algorithm considered in [17].

Algorithm 1 $\mathcal{A}_{DAG}$ (building of DAG $\Delta(T)$).

Input: A nonempty decision table T that has n conditional attributes $f_1, \ldots, f_n$.
Output: Directed acyclic graph $\Delta(T)$.

1. Build the graph consisting of one node T that is not tagged as processed.
2. Check the processing of all nodes of the graph is completed or not. If yes, then the algorithm halts and returns the resulting graph as $\Delta(T)$. Otherwise, select a node (table) Θ which is yet unprocessed.
3. Check node Θ is degenerate or not.
   (a) If yes, then tag the node Θ as processed and move to step 2.
   (b) If no, then draw a bundle of edges from the node Θ for each $f_i \in E(\Theta)$. Let $E(\Theta, f_i) = \{a_1, \ldots, a_k\}$. Then draw k edges from Θ and attach to these edges systems of equations $\{f_i = a_1\}, \ldots, \{f_i = a_k\}$. These edges enter nodes $\Theta\{f_i = a_1\}, \ldots, \Theta\{f_i = a_k\}$, respectively. In case some of the nodes $\Theta\{f_i = a_1\}, \ldots, \Theta\{f_i = a_k\}$ are not available in the graph, then add these nodes to the graph. Tag the node Θ as processed and move to step 2.

5. Construction of Decision Trees with Minimum Depth

Let us consider a nonempty decision table T that contains n conditional attributes $f_1, \ldots, f_n$ and $k \in \{1, \ldots, 5\}$. We can use the DAG $\Delta(T)$ to construct a decision tree $\Gamma^{(k)}(T)$ of the type k with the minimum depth for the decision table T. For this purpose, we have to construct, corresponding to each vertex Θ of $\Delta(T)$, a decision tree $\Gamma^{(k)}(\Theta)$ of the type k with minimum depth for the table Θ. It is necessary not only consider subtables corresponding to the nodes of $\Delta(T)$ but also empty subtable Λ of T as well as subtables T_r containing only one row r of T, which are not nodes of $\Delta(T)$. The idea is to start with these special subtables as well as leaf nodes of $\Delta(T)$ that are degenerate separable subtables of T. In this way, we move step wise in a bottom up fashion to the table T.

Let us consider the case when Θ is a leaf node of $\Delta(T)$ or $\Theta = T_r$ for a row r of the table T, or $T = \Lambda$. If Θ is nonempty, then $\Gamma^{(k)}(\Theta)$ has only one node that is tagged by a decision which is attached to all rows of Θ. Otherwise, it is tagged with 0.

Let us consider other case when Θ is an internal node of $\Delta(T)$ and the construction of the decision tree $\Gamma^{(k)}(\Theta')$ is already completed for each child Θ' of Θ. Based on these trees, a decision tree for Θ having the minimum depth can be constructed that uses decision trees of the type k for the subtables corresponding to the children of the root. In this tree, the root can be tagged as follows:

- By an attribute from $F(T)$ (such decision tree can be designated as $\Gamma_a^{(k)}(\Theta)$).
- By a hypothesis over T (such decision tree can be designated as $\Gamma_h^{(k)}(\Theta)$).
- By a proper hypothesis over T (such decision tree can be designated as $\Gamma_p^{(k)}(\Theta)$).

The set $E(\Theta)$ is nonempty because Θ is nondegenerate. Now, three procedures for the construction of the trees $\Gamma_a^{(k)}(\Theta)$, $\Gamma_h^{(k)}(\Theta)$, and $\Gamma_p^{(k)}(\Theta)$ are described.

We now concentrate on a decision tree $\Gamma(f_i)$ for the node Θ, where the root is tagged by an attribute $f_i \in E(\Theta)$. For each $\delta \in E(T, f_i)$, there exists an edge that exits the root and enters the root of the decision tree $\Gamma^{(k)}(\Theta\{f_i = \delta\})$. We tag this edge by the equation system $\{f_i = \delta\}$. It is obvious that

$$h(\Gamma(f_i)) = 1 + \max\{h(\Gamma^{(k)}(\Theta\{f_i = \delta\})) : \delta \in E(T, f_i)\}. \quad (1)$$

One can easily prove using (1) that $\Gamma(f_i)$ is a decision tree with the minimum depth for Θ such that the root of this tree is tagged by the attribute f_i and it uses decision trees of the type k for the subtables corresponding to the children of the root.

It is obvious not to consider attributes $f_i \in F(T) \setminus E(\Theta)$. The reason is that for such f_i, we can find $\delta \in E(T, f_i)$ with $\Theta\{f_i = \delta\} = \Theta$. Therefore, we cannot construct an optimal tree for Θ based on f_i.

Construction of the tree $\Gamma_a^{(k)}(\Theta)$. We build the set $E(\Theta)$. For any $f_i \in E(\Theta)$, construct the decision tree $\Gamma(f_i)$ and choose among these trees a tree with the minimum depth. Return this tree as $\Gamma_a^{(k)}(\Theta)$.

Let us consider a hypothesis $H = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ over T. We call this hypothesis admissible for Θ and an attribute $f_i \in F(T) = \{f_1, \ldots, f_n\}$ if $\Theta\{f_i = \sigma\} \neq \Theta$ for any $\sigma \in E(T, f_i) \setminus \{\delta_i\}$. This hypothesis is not admissible for Θ and an attribute $f_i \in F(T)$ if and only if $|E(\Theta, f_i)| = 1$ and $\delta_i \notin E(\Theta, f_i)$. We call H admissible for Θ when we find that H is admissible for Θ and any attribute $f_i \in F(T)$.

We now describe a decision tree $\Gamma(H)$ for Θ. The root of this tree is tagged by an admissible hypothesis $H = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ for Θ. For any equation system $S \in A(H)$, there is an edge that exits the root of $\Gamma(H)$ and enters the root of the tree $\Gamma^{(k)}(\Theta S)$ This edge is tagged by the equation system S.

It is obvious that

$$h(\Gamma(H)) = 1 + \max\{h(\Gamma^{(k)}(\Theta S)) : S \in A(H)\}. \quad (2)$$

One can easily prove using (2) that $\Gamma(H)$ is a decision tree with the minimum depth for Θ such that the root of this tree is tagged by the hypothesis H and it uses decision trees of the type k for the subtables corresponding to the children of the root.

It is obvious not to consider hypotheses H that are not admissible for Θ. The reason is that for such H, we can find an equation system $S \in A(H)$ with $\Theta S = \Theta$. Therefore, we cannot construct an optimal decision tree for Θ based on H.

Construction of the tree $\Gamma_h^{(k)}(\Theta)$. Construct a hypothesis $H_\Theta = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ for Θ. If $f_i \in F(T) \setminus E(\Theta)$, then δ_i is the only value from $E(\Theta, f_i)$. If $f_i \in E(\Theta)$, then δ_i is minimum number from $E(\Theta, f_i)$ for which $h(\Gamma^{(k)}(\Theta\{f_i = \delta_i\})) = \max\{h(\Gamma^{(k)}(\Theta\{f_i = \sigma\})) : \sigma \in E(\Theta, f_i)\}$. Return the tree $\Gamma(H_\Theta)$ as $\Gamma_h^{(k)}(\Theta)$. Using (2), one can prove the correctness of this procedure.

Construction of the tree $\Gamma_p^{(k)}(\Theta)$. For each row $r = (\delta_1, \ldots, \delta_n)$ of the decision table T, we consider a proper hypothesis $H_r = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$. We inspect if H_r is admissible for Θ. If yes, then we construct the decision tree $\Gamma(H_r)$. We choose among the constructed trees a tree with the minimum depth. Return this tree as $\Gamma_p^{(k)}(\Theta)$.

Given input of a decision table T and $k \in \{1, \ldots, 5\}$, the following Algorithm 2 $\mathcal{C}_h$ builds for each node Θ of the DAG $\Delta(T)$ a decision tree $\Gamma^{(k)}(\Theta)$ of the type k for the table Θ having the minimum depth.

Algorithm 2 $\mathcal{C}_h$ (construction of the tree $\Gamma^{(k)}(T)$).

Input: T (a nonempty decision table), $\Delta(T)$ (the directed acyclic graph for T), and k (a natural number between 1 to 5).
Output: A decision tree $\Gamma^{(k)}(T)$.

1. Check all nodes of the DAG $\Delta(T)$ whether there is a decision tree attached to each node. If yes, then return the tree attached to the node T as $\Gamma^{(k)}(T)$ and break the algorithm. If not, select a node Θ of the graph $\Delta(T)$ that does not have an attached tree. It can be either a leaf node of $\Delta(T)$ or an internal node of $\Delta(T)$ where all children are tagged with trees.
2. If Θ is a leaf node, then attach to it the decision tree $\Gamma^{(k)}(\Theta)$ that have only a single node. This node is tagged with the decision attached to all rows of Θ. Move to step 1.
3. If Θ is not a leaf node, then do the following according to the value k:
   - When $k = 1$, construct the tree $\Gamma_a^{(1)}(\Theta)$ and attach it to Θ as the tree $\Gamma^{(1)}(\Theta)$.
   - When $k = 2$, construct the tree $\Gamma_h^{(2)}(\Theta)$ and attach it to Θ as the tree $\Gamma^{(2)}(\Theta)$.
   - When $k = 3$, construct the trees $\Gamma_a^{(3)}(\Theta)$ and $\Gamma_h^{(3)}(\Theta)$, choose between them a tree with the minimum depth and attach it to Θ as the tree $\Gamma^{(3)}(\Theta)$.
   - When $k = 4$, construct the tree $\Gamma_p^{(4)}(\Theta)$ and attach it to Θ as the tree $\Gamma^{(4)}(\Theta)$.
   - When $k = 5$, construct the trees $\Gamma_a^{(5)}(\Theta)$ and $\Gamma_p^{(5)}(\Theta)$, choose between them a tree with the minimum depth and attach it to Θ as the tree $\Gamma^{(5)}(\Theta)$.

   Move to step 1.

Let T be a decision table and $k \in \{1, \ldots, 5\}$. We use the following notation: $l_h^{(k)}(T) = l(T, \Gamma^{(k)}(T))$ and $c_h^{(k)}(T) = c(T, \Gamma^{(k)}(T))$.

6. Construction of Decision Trees Containing Minimum Number of Internal Nodes

Let us consider a nonempty decision table T that contains n conditional attributes $f_1, \ldots, f_n$ and $k \in \{1, \ldots, 5\}$. We can use the DAG $\Delta(T)$ to construct a decision tree $G^{(k)}(T)$ of the type k with the minimum number of internal nodes for the decision table T. To construct the tree $G^{(k)}(T)$, for each node Θ of the DAG $\Delta(T)$, we construct a decision tree $G^{(k)}(\Theta)$ of the type k with the minimum number of internal nodes for the decision table Θ. It is necessary to not only consider the subtables corresponding to the nodes of $\Delta(T)$ but also the empty subtable Λ of T as well as the subtables T_r containing only one row r of T which are not nodes of $\Delta(T)$. The idea is to start with these special subtables as well as leaf nodes of $\Delta(T)$ that are degenerate separable subtables of T. In this way, we move step wise in a bottom up fashion to the table T.

Let us consider the case when Θ is a leaf node of $\Delta(T)$ or $\Theta = T_r$ for a row r of the table T, or $T = \Lambda$. If Θ is nonempty, then the decision tree $G^{(k)}(\Theta)$ has only one node that is tagged by a decision which is attached to all rows of Θ. Otherwise, it is tagged with 0.

Let us consider another case when Θ is an internal node of $\Delta(T)$ such that the construction of the decision tree $G^{(k)}(\Theta')$ is already completed for each child Θ' of Θ. Based on these trees, a decision tree containing the minimum number of internal nodes for Θ can be constructed that uses decision trees of the type k for the subtables corresponding to the children of the root. In this tree, the root can be tagged as follows:

- By an attribute from $F(T)$ (such decision tree can be designated as $G_a^{(k)}(\Theta)$).
- By a hypothesis over T (such decision tree can be designated as $G_h^{(k)}(\Theta)$).
- By a proper hypothesis over T (such decision tree can be designated as $G_p^{(k)}(\Theta)$).

The set $E(\Theta)$ is nonempty because Θ is nondegenerate. Now, three procedures for the construction of the trees $G_a^{(k)}(\Theta)$, $G_h^{(k)}(\Theta)$, and $G_p^{(k)}(\Theta)$ are described.

We now concentrate on a decision tree $G(f_i)$ for the node Θ where the root is tagged by an attribute $f_i \in E(\Theta)$. For each $\delta \in E(T, f_i)$, there is an edge that exits the root and enters the root of the decision tree $G^{(k)}(\Theta\{f_i = \delta\})$. We tag this edge by the equation system $\{f_i = \delta\}$. It is obvious that

$$L_w(G(f_i)) = 1 + \sum_{\delta \in E(T,f_i)} L_w(G^{(k)}(\Theta\{f_i = \delta\})). \quad (3)$$

One can easily prove using (3) that $G(f_i)$ is a decision tree with the minimum number of internal nodes for Θ such that the root of the tree is tagged by the attribute f_i and it uses decision trees of the type k for the subtables corresponding to the children of the root.

It is obvious not to consider attributes $f_i \in F(T) \setminus E(\Theta)$. The reason is that for such f_i, we can find $\delta \in E(T, f_i)$ with $\Theta\{f_i = \delta\} = \Theta$. Therefore, we cannot construct an optimal tree for Θ based on f_i.

Construction of the tree $G_a^{(k)}(\Theta)$. We build the set of attributes $E(\Theta)$. For any $f_i \in E(\Theta)$, construct the decision tree $G(f_i)$ and choose among these trees a tree with the minimum number of internal nodes. We return this tree as $G_a^{(k)}(\Theta)$.

We now describe a decision tree $G(H)$ for Θ. The root of this tree is tagged by an admissible hypothesis $H = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ for Θ. For any equation system $S \in A(H)$, there is an edge that exits the root of $G(H)$ and enters the root of the tree $G^{(k)}(\Theta S)$. This edge is tagged by the equation system S. It is obvious that

$$L_w(G(H)) = 1 + \sum_{S \in A(H)} L_w(G^{(k)}(\Theta S)). \quad (4)$$

One can easily prove using (4) that $G(H)$ is a decision tree with the minimum number of internal nodes for Θ such that the root of the tree is tagged by the hypothesis H and it uses decision trees of the type k for the subtables corresponding to the children of the root.

It is obvious not to consider hypotheses H that are not admissible for Θ. The reason is that for such H, we can find an equation system $S \in A(H)$ with $\Theta S = \Theta$. Therefore, we cannot construct an optimal decision tree for Θ based on such H.

Construction of the tree $G_h^{(k)}(\Theta)$. We construct a hypothesis $H_\Theta = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$ for Θ. If $f_i \notin E(\Theta)$, then δ_i is the only value in $E(\Theta, f_i)$. Let $f_i \in E(\Theta)$. Then δ_i is the minimum number from $E(\Theta, f_i)$ such that $L_w(G^{(k)}(\Theta\{f_i = \delta_i\})) = \max\{L_w(G^{(k)}(\Theta\{f_i = \sigma\})) : \sigma \in E(\Theta, f_i)\}$. Obviously, H_Θ is admissible for Θ. Return the tree $G(H_\Theta)$ as $G_h^{(k)}(\Theta)$. Using (4), one can prove the correctness of this procedure.

Construction of the tree $G_p^{(k)}(\Theta)$. For each row $r = (\delta_1, \ldots, \delta_n)$ of the decision table T, let us consider a proper hypothesis $H_r = \{f_1 = \delta_1, \ldots, f_n = \delta_n\}$. We inspect if H_r is admissible for Θ. If yes, then we construct the decision tree $G(H_r)$. We choose among the constructed trees a tree with the minimum number of internal nodes. Return this tree as $G_p^{(k)}(\Theta)$.

Given input of a decision table T and $k \in \{1, \ldots, 5\}$, the following Algorithm 3 $\mathcal{C}_{L_w}$ builds for each node Θ of the DAG $\Delta(T)$ a decision tree $G^{(k)}(\Theta)$ of the type k for the table Θ having the minimum number of internal nodes.

Algorithm 3 $\mathcal{C}_{L_w}$ (construction of the tree $G^{(k)}(T)$).

Input: T (a nonempty decision table), $\Delta(T)$ (the directed acyclic graph for T), and k (a natural number between 1 to 5).
Output: A decision tree $G^{(k)}(T)$.

1. Check all nodes of the DAG $\Delta(T)$ whether there is a decision tree attached to each node. If yes, then return the tree attached to the node T as $G^{(k)}(T)$ and break the algorithm. If not, select a node Θ of the graph $\Delta(T)$ that does not have an attached tree. It can be either a leaf node of $\Delta(T)$ or an internal node of $\Delta(T)$ where all children are tagged with trees.
2. If Θ is a leaf node, then attach to it the decision tree $G^{(k)}(\Theta)$ that have only a single node. This node is tagged with the decision attached to all rows of Θ. Move to step 1.
3. If Θ is not a leaf node, then do the following according to the value k:
   - When $k = 1$, construct the tree $G_a^{(1)}(\Theta)$ and attach it to Θ as the tree $G^{(1)}(\Theta)$.
   - When $k = 2$, construct the tree $G_h^{(2)}(\Theta)$ and attach it to Θ as the tree $G^{(2)}(\Theta)$.
   - When $k = 3$, construct the trees $G_a^{(3)}(\Theta)$ and $G_h^{(3)}(\Theta)$, choose between them a tree with the minimum number of internal nodes and attach it to Θ as the tree $G^{(3)}(\Theta)$.
   - When $k = 4$, construct the tree $G_p^{(4)}(\Theta)$ and attach it to Θ as the tree $G^{(4)}(\Theta)$.
   - When $k = 5$, construct the trees $G_a^{(5)}(\Theta)$ and $G_p^{(5)}(\Theta)$, choose between them a tree with the minimum number of internal nodes and attach it to Θ as the tree $G^{(5)}(\Theta)$.

   Move to step 1.

Let T be a decision table and $k \in \{1, \ldots, 5\}$. We use the following notation: $l_{L_w}^{(k)}(T) = l(T, G^{(k)}(T))$ and $c_{L_w}^{(k)}(T) = c(T, G^{(k)}(T))$.

7. Experimental Results and Discussion

In this section, we describe the results of the experiments. First, we accomplished the experiments on eight decision tables from the UCI ML Repository [16]. We give the description of these tables in Table 1 where we show first the name of the table (Name), then number of rows (#Rows) and the number of attributes (#Attrs). We arranged the decision tables in Table 1 based on the number of rows. For each of these tables, we built an optimal decision tree of each of five possible types for each of the two possible cost functions. From these trees, we derive decision rules and study their coverage and length.

Table 1. Description of decision tables from [16] which were used in experiments.

Name	#Rows	#Attrs
SOYBEAN-SMALL	47	36
ZOO-DATA	59	17
HAYES-ROTH-DATA	69	5
BREAST-CANCER	266	10
BALANCE-SCALE	625	5
TIC-TAC-TOE	958	10
CARS	1728	7
NURSERY	12,960	9

Next, we experimented with 100 Boolean functions having n variables ($n = 3, \ldots, 6$) which are generated randomly. Let f be such a Boolean function with n variables $x_1, \ldots, x_n$. We can map it to a decision table T_f having n attributes $x_1, \ldots, x_n$. This table has 2^n rows corresponding to all possible n-tuples of variable values. We label each row with the

decision that is the value of the function f for the considered row. The decision trees for the table T_f are interpreted as decision trees that compute the function f.

For each of tables representing the generated Boolean functions, we build an optimal decision tree of each of five possible types for each of the two possible cost functions. From these trees, we derive decision rules and study their coverage and length.

7.1. Decision Trees with Minimum Depth

The results of experiments based on eight decision tables from [16] and decision trees optimal relative to the depth 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 $l_h^{(1)}(T), \ldots, l_h^{(5)}(T)$ (minimum values for each decision table are in bold).

Table 2. Results for decision tables from [16]: length of decision rules derived from decision trees with minimum depth.

Decision Table T	$l_h^{(1)}(T)$	$l_h^{(2)}(T)$	$l_h^{(3)}(T)$	$l_h^{(4)}(T)$	$l_h^{(5)}(T)$
SOYBEAN-SMALL	1.89	**1.00**	1.89	1.55	1.89
ZOO-DATA	3.69	**1.56**	2.42	2.17	3.24
HAYES-ROTH-DATA	2.83	2.22	**2.16**	2.32	2.35
BREAST-CANCER	3.61	**2.68**	2.71	2.70	2.78
BALANCE-SCALE	3.60	**3.20**	**3.20**	3.21	**3.20**
TIC-TAC-TOE	5.09	**3.04**	3.40	3.24	3.14
CARS	3.72	**2.44**	2.48	3.07	3.02
NURSERY	5.78	3.16	4.53	**3.12**	4.50
Average	3.78	2.41	2.85	2.67	3.01

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

Table 3. Results for decision tables from [16]: coverage of decision rules derived from decision trees with minimum depth.

Decision Table T	$c_h^{(1)}(T)$	$c_h^{(2)}(T)$	$c_h^{(3)}(T)$	$c_h^{(4)}(T)$	$c_h^{(5)}(T)$
SOYBEAN-SMALL	3.47	**12.53**	3.47	10.62	3.47
ZOO-DATA	7.88	10.78	9.80	**10.86**	6.46
HAYES-ROTH-DATA	3.46	6.20	**6.49**	5.48	5.45
BREAST-CANCER	4.98	9.30	8.36	**9.38**	6.90
BALANCE-SCALE	2.60	**4.19**	4.18	4.16	**4.19**
TIC-TAC-TOE	8.38	**66.01**	27.23	56.64	61.45
CARS	197.60	**332.76**	330.35	97.20	99.42
NURSERY	29.33	1524.04	304.71	**1530.50**	246.14
Average	32.21	245.73	86.82	215.60	54.18

The results of experiments based on Boolean functions and decision trees optimal relative to the depth are represented in Tables 4 and 5. The first column of Table 4 contains the number n of variables in the considered Boolean functions. The last five columns contain information about values $l_h^{(1)}, \ldots, l_h^{(5)}$ in the format ${}_{min}Avg_{max}$ (minimum values of Avg for each n are in bold).

The first column of Table 5 contains the number n of variables in the considered Boolean functions. The last five columns contain information about values $c_h^{(1)}, \ldots, c_h^{(5)}$ in the format ${}_{min}Avg_{max}$ (maximum values of Avg for each n are in bold).

Table 4. Results for Boolean functions: length of decision rules derived from decision trees with minimum depth.

Number of Variables n	$l_h^{(1)}$	$l_h^{(2)}$	$l_h^{(3)}$	$l_h^{(4)}$	$l_h^{(5)}$
3	$_{1.50}2.20_{2.75}$	$_{1.25}2.05_{2.63}$	$_{1.25}\mathbf{1.99}_{2.50}$	$_{1.25}2.08_{2.63}$	$_{1.25}2.01_{2.50}$
4	$_{1.88}3.18_{3.75}$	$_{1.63}2.92_{3.50}$	$_{1.63}\mathbf{2.87}_{3.50}$	$_{1.63}2.91_{3.50}$	$_{1.63}2.94_{3.50}$
5	$_{3.44}4.09_{4.63}$	$_{3.00}3.64_{4.22}$	$_{2.97}\mathbf{3.60}_{4.06}$	$_{3.13}3.66_{4.13}$	$_{3.09}3.70_{4.19}$
6	$_{4.78}5.14_{5.47}$	$_{3.98}4.36_{4.77}$	$_{3.98}4.41_{4.75}$	$_{3.97}\mathbf{4.35}_{4.70}$	$_{4.03}4.46_{4.78}$

Table 5. Results for Boolean functions: coverage of decision rules derived from decision trees with minimum depth.

Number of Variables n	$c_h^{(1)}$	$c_h^{(2)}$	$c_h^{(3)}$	$c_h^{(4)}$	$c_h^{(5)}$
3	$_{1.25}1.94_{3.00}$	$_{1.38}\mathbf{2.22}_{3.63}$	$_{1.50}2.21_{3.63}$	$_{1.38}2.14_{3.63}$	$_{1.50}2.17_{3.63}$
4	$_{1.25}1.99_{5.38}$	$_{1.50}\mathbf{2.57}_{6.44}$	$_{1.50}2.52_{6.44}$	$_{1.50}2.53_{6.44}$	$_{1.50}2.36_{6.44}$
5	$_{1.38}2.10_{3.69}$	$_{2.03}\mathbf{3.03}_{4.56}$	$_{2.06}2.98_{4.97}$	$_{2.03}2.93_{4.69}$	$_{1.81}2.76_{4.66}$
6	$_{1.59}2.03_{2.84}$	$_{2.69}\mathbf{3.55}_{4.81}$	$_{2.58}3.37_{4.64}$	$_{2.72}3.53_{4.70}$	$_{2.53}3.24_{4.69}$

7.2. Decision Trees Containing Minimum Number of Internal Nodes

We present the results based on the decision tables from [16] and decision trees optimal relative to the number of internal nodes in Tables 6 and 7. The first column of Table 6 contains the name of the considered decision table T. The last five columns contain values $l_{L_w}^{(1)}(T), \ldots, l_{L_w}^{(5)}(T)$ (minimum values for each decision table are in bold).

Table 6. Results for decision tables from [16]: length of decision rules derived from decision trees with minimum number of internal nodes.

Decision Table T	$l_{L_w}^{(1)}(T)$	$l_{L_w}^{(2)}(T)$	$l_{L_w}^{(3)}(T)$	$l_{L_w}^{(4)}(T)$	$l_{L_w}^{(5)}(T)$
SOYBEAN-SMALL	1.34	**1.00**	1.34	1.51	1.34
ZOO-DATA	3.05	**1.69**	3.05	2.39	3.05
HAYES-ROTH-DATA	2.64	**2.22**	2.61	2.23	2.61
BREAST-CANCER	4.98	**2.72**	5.30	2.73	5.27
BALANCE-SCALE	3.55	**3.20**	3.53	**3.20**	3.53
TIC-TAC-TOE	4.45	3.35	4.41	**3.15**	4.45
CARS	2.97	**2.49**	2.96	**2.49**	2.96
NURSERY	3.77	**3.19**	3.77	**3.19**	3.77
Average	3.34	2.48	3.37	2.61	3.37

The first column of Table 7 contains the name of the considered decision table T. The last five columns contain values $c_{L_w}^{(1)}(T), \ldots, c_{L_w}^{(5)}(T)$ (maximum values for each decision table are in bold).

The results of experiments based on Boolean functions and decision trees optimal relative to the number of internal nodes are represented in Tables 8 and 9. The first column of Table 8 contains the number n of variables in the considered Boolean functions. The last five columns contain information about values $l_{L_w}^{(1)}, \ldots, l_{L_w}^{(5)}$ in the format $_{min}Avg_{max}$ (minimum values of Avg for each n are in bold).

Table 7. Results for decision tables from [16]: coverage of decision rules derived from decision trees with minimum number of internal nodes.

Decision Table T	$c^{(1)}_{L_w}(T)$	$c^{(2)}_{L_w}(T)$	$c^{(3)}_{L_w}(T)$	$c^{(4)}_{L_w}(T)$	$c^{(5)}_{L_w}(T)$
SOYBEAN-SMALL	11.51	**12.53**	11.51	9.81	11.51
ZOO-DATA	**10.69**	10.68	**10.69**	10.63	**10.69**
HAYES-ROTH-DATA	3.84	**6.20**	3.87	**6.20**	3.87
BREAST-CANCER	2.73	8.96	3.05	**9.06**	3.15
BALANCE-SCALE	2.79	**4.21**	2.88	**4.21**	2.88
TIC-TAC-TOE	22.49	30.19	23.50	**56.50**	22.69
CARS	237.33	**332.46**	237.37	**332.46**	237.37
NURSERY	1471.45	**1527.95**	1471.47	**1527.95**	1471.47
Average	220.35	241.65	220.54	244.60	220.45

Table 8. Results for Boolean functions: length of decision rules derived from decision trees with minimum number of internal nodes.

Number of Variables n	$l^{(1)}_{L_w}$	$l^{(2)}_{L_w}$	$l^{(3)}_{L_w}$	$l^{(4)}_{L_w}$	$l^{(5)}_{L_w}$
3	$_{1.50}2.07_{2.75}$	$_{1.25}2.06_{2.63}$	$_{1.25}\mathbf{1.94}_{2.50}$	$_{1.25}2.06_{2.63}$	$_{1.25}\mathbf{1.94}_{2.50}$
4	$_{1.88}2.90_{3.50}$	$_{1.63}2.94_{3.50}$	$_{1.81}\mathbf{2.79}_{3.50}$	$_{1.63}2.94_{3.50}$	$_{1.81}\mathbf{2.79}_{3.50}$
5	$_{3.13}3.75_{4.19}$	$_{3.00}3.77_{4.31}$	$_{3.13}\mathbf{3.69}_{4.25}$	$_{3.00}3.77_{4.31}$	$_{3.13}\mathbf{3.69}_{4.25}$
6	$_{4.28}4.69_{5.06}$	$_{4.13}4.69_{5.19}$	$_{4.25}\mathbf{4.61}_{4.98}$	$_{4.13}4.69_{5.19}$	$_{4.25}\mathbf{4.61}_{4.98}$

The first column of Table 9 contains the number of variables n in the considered Boolean functions. The last five columns contain information about values $c^{(1)}_{L_w}, \ldots, c^{(5)}_{L_w}$ in the format $_{min}Avg_{max}$ (maximum values of Avg for each n are in bold).

Table 9. Results for Boolean functions: coverage of decision rules derived from decision trees with minimum number of internal nodes.

Number of Variables n	$c^{(1)}_{L_w}$	$c^{(2)}_{L_w}$	$c^{(3)}_{L_w}$	$c^{(4)}_{L_w}$	$c^{(5)}_{L_w}$
3	$_{1.25}2.14_{3.00}$	$_{1.38}2.22_{3.63}$	$_{1.50}\mathbf{2.27}_{3.63}$	$_{1.38}2.22_{3.63}$	$_{1.50}\mathbf{2.27}_{3.63}$
4	$_{1.63}2.51_{5.38}$	$_{1.50}2.56_{6.44}$	$_{1.50}\mathbf{2.63}_{5.44}$	$_{1.50}2.56_{6.44}$	$_{1.50}\mathbf{2.63}_{5.44}$
5	$_{1.88}2.79_{4.19}$	$_{1.94}\mathbf{2.91}_{4.59}$	$_{1.75}2.82_{4.34}$	$_{1.94}\mathbf{2.91}_{4.59}$	$_{1.75}2.82_{4.34}$
6	$_{2.16}2.88_{4.09}$	$_{2.09}\mathbf{3.16}_{4.58}$	$_{2.27}2.99_{4.11}$	$_{2.09}\mathbf{3.16}_{4.58}$	$_{2.27}2.99_{4.11}$

7.3. Analysis of Experimental Results

The experimental results show that the decision rules derived from decision trees with hypotheses in many cases are better than the ones derived from conventional decision trees. In particular, in the case of decision trees with the minimum depth, for each row in Tables 2–5, the results for type 2 decision trees are better than the results for type 1 decision trees. In the case of decision trees with a minimum number of internal nodes, for each row of Tables 6–9 (with the exception of the row ZOO-DATA in Table 7) there is a number $k \in \{2, \ldots, 5\}$ such that the results for type k decision trees are superior compared to the results for type 1 decision trees.

Note that for the decision trees with the minimum depth, for each decision table from [16] considered in this paper, the best results related to the length and the coverage among decision trees of types $\{2, \ldots, 5\}$ are close to the optimal ones obtained in [18] with the help of dynamic programming algorithms for the construction of optimal decision rules. Results for the decision trees of the type 1 are, generally, far from the optimal ones.

From the obtained experimental results, it follows that the decision rules derived from optimal decision trees with hypotheses are more preferable as a tool for the representation of information than the decision rules derived from optimal conventional decision trees.

8. Conclusions

In this paper, we studied modified decision trees that use two types of queries. We constructed optimal trees relative to two cost functions for a number of known datasets from the UCI Machine Learning Repository and randomly generated Boolean functions, and compared the length as well as coverage of decision rules extracted from the constructed decision trees. The experimental results confirmed that the decision rules derived from the decision trees with hypotheses in many cases are better than the ones derived from conventional decision trees.

Author Contributions: Conceptualization, all authors; methodology, all authors; software, I.C.; validation, I.C., M.A., S.H. and B.Z.; formal analysis, all authors; investigation, B.Z.; resources, all authors; data curation, M.A., S.H. and B.Z.; writing—original draft preparation, M.M.; writing—review and editing, all authors; visualization, B.Z.; 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 are available in a publicly accessible repository that does not issue DOIs. Publicly available datasets were analyzed in this study. These data can be found here: http://archive.ics.uci.edu/ml (accessed on 12 April 2017).

Acknowledgments: Research reported in this publication was supported by King Abdullah University of Science and Technology (KAUST). 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 *Trans. Rough Sets III*; Peters, J.F., Skowron, A., Eds.; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2005; Volume 3400, pp. 244–459.
3. Rokach, L.; Maimon, O. Data Mining with Decision Trees—Theory and Applications. In *Series in Machine Perception and Artificial Intelligence*; World Scientific: Singapore, 2007; Volume 69.
4. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R*; Springer: New York, NY, USA, 2021.
5. 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)
6. Pawlak, Z. Rough sets. *Int. J. Parallel Program.* **1982**, *11*, 341–356. [CrossRef]
7. Pawlak, Z. Rough Sets—Theoretical Aspects of Reasoning about Data. In *Theory and Decision Library: Series D*; Kluwer: Dordrecht, The Netherlands, 1991; Volume 9.
8. Pawlak, Z.; Skowron, A. Rudiments of rough sets. *Inf. Sci.* **2007**, *177*, 3–27. [CrossRef]
9. Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M. Minimizing depth of decision trees with hypotheses. In *Lecture Notes in Computer Science, Proceedings of the Rough Sets—International Joint Conference, IJCRS 2021, Bratislava, Slovakia, 19–24 September 2021*; Ramanna, S., Cornelis, C., Ciucci, D., Eds.; Springer: Cham, Switzerland, 2021; Volume 12872, pp. 123–133.
10. Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M. Minimizing number of nodes in decision trees with hypotheses. In *Procedia Computer Science, Proceedings of the 25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, KES 2021, Szczecin, Poland, 8–10 September 2021*; Watrobski, J., Salabun, W., Toro, C., Zanni-Merk, C., Howlett, R.J., Jain, L.C., Eds.; Elsevier: Amsterdam, The Netherlands, 2021; Volume 192, pp. 232–240.
11. Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M. Entropy-based greedy algorithm for decision trees using hypotheses. *Entropy* **2021**, *23*, 808. [CrossRef] [PubMed]
12. Azad, M.; Chikalov, I.; Hussain, S.; Moshkov, M. Optimization of decision trees with hypotheses for knowledge representation. *Electronics* **2021**, *10*, 1580. [CrossRef]
13. Angluin, D. Learning regular sets from queries and counterexamples. *Inf. Comput.* **1987**, *75*, 87–106. [CrossRef]
14. Angluin, D. Queries and concept learning. *Mach. Learn.* **1988**, *2*, 319–342. [CrossRef]
15. Angluin, D. Queries revisited. *Theor. Comput. Sci.* **2004**, *313*, 175–194. [CrossRef]

16. 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).
17. AbouEisha, H.; Amin, T.; Chikalov, I.; Hussain, S.; Moshkov, M. Extensions of Dynamic Programming for Combinatorial Optimization and Data Mining. In *Intelligent Systems Reference Library*; Springer: Cham, Switzerland, 2019; Volume 146.
18. Amin, T.; Chikalov, I.; Moshkov, M.; Zielosko, B. Dynamic programming approach for exact decision rule optimization. In *Rough Sets and Intelligent Systems—Professor Zdzisław Pawlak in Memoriam—Volume 1*; Skowron, A., Suraj, Z., Eds.; Intelligent Systems Reference Library; Springer: Berlin/Heidelberg, Germany, 2013; Volume 42, pp. 211–228.