When Decisions Are Blocked: Inhibitory Rules Induced from Tree Ensembles

Beata Zielosko[1,2][0000−0003−3788−1094], Anton Dmytrenko[1][0009−0006−7317−1179], Azimkhon Ostonov[2][0000−0001−5763−9751], and Mikhail Moshkov[2][0000−0003−0085−9483]

[1] Institute of Computer Science, University of Silesia in Katowice, Będzińska 39, 41-200 Sosnowiec, Poland
{beata.zielosko,anton.dmytrenko}@us.edu.pl
[2] Computer, Electrical and Mathematical Sciences & Engineering Division King Abdullah University of Science and Technology (KAUST) Thuwal 23955-6900, Saudi Arabia
{azimkhon.ostonov,mikhail.moshkov}@kaust.edu.sa

Abstract. Decision trees and rule-based systems are widely used in various data mining tasks. Their great advantage is the ability to describe decision-making processes in a transparent manner. Unlike standard decision rules "if-then", inhibitory rules have in their successor form: "attribute $\neq$ decision". In the paper, two types of inhibitory rules have been defined: inner and general. Inner inhibitory rules are derived from complete paths in individual decision trees (path from the root to leaf of the tree), whereas general inhibitory rules are constructed using any attributes present across the trees. This work explores the problem of extracting both inner and general inhibitory rules that are valid for the largest possible number of decision trees within a given set. We introduce a polynomial-time algorithm to solve the inner inhibitory rule optimization problem, establish that optimizing general inhibitory rules is NP-hard, and propose a heuristic approach to approximate its solution. Experimental evaluations are conducted using synthetically generated decision tree sets to compare the performance of the proposed algorithms, taking into account the number of trees for which the constructed rules are true, and their length.

Keywords: Inhibitory Rules · Inner Rules · General Rules · Decision Trees · Distributed Data.

1 Introduction

Decision trees and rule-based systems are widely used in various tasks related to classification and knowledge representation [8, 11, 14]. Their great advantage is the interpretable and structured way of modelling decision-making processes [7, 15, 16]. Decision trees represent knowledge that can be expressed in the form of 'if-then' decision rules. Each internal node of a decision tree corresponds to an attribute test, the branches represent possible test results, and the terminal

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

nodes (tree leaves) define decisions (class labels). This structure facilitates the representation of knowledge in the form of rules, making it easy to understand, analyze, and explain, even for those without expertise in a particular field.

Inhibitory rules complement this approach by explicitly modelling constraints or exceptions that prevent certain conclusions from being drawn when specific conditions are met. Unlike standard decision rules, inhibitory rules have a consequent in the form of: "attribute $\neq$ decision" and often suggest insights not captured by traditional decision rules [1, 2, 6]. Previous studies [9] demonstrated that for some information systems, ordinary (decision) rules cannot describe the whole information contained in the system. However, inhibitory rules describe the whole information for every information system. It means that inhibitory rules can express more information encoded in data than decision rules. Moreover, classifiers based on inhibitory rules often have better accuracy than classifiers based on decision rules [5].

The rationale for using inhibitory rules can be explained by highlighting their role as safeguards and a kind of filter in real-world decision support systems. While models based on ordinary decision rules suggest a positive decision during the classification process, inhibitory rules allow for the identification of conditions under which certain decisions should be suspended or reconsidered.

This study addresses the challenging problems of extracting inhibitory rules from a set of decision trees. A framework for decision rules was developed in [17]. However, existing research on pattern discovery from distributed data has not considered the use of inhibitory rules. This paper addresses this gap by proposing their first application in this context.

The proposed method can be applied in scenarios involving multiple local and distributed data sources represented as decision trees, where the goal is to extract knowledge that is important for all these sources and can be used both locally and globally. The main objective is to identify patterns that reflect knowledge common to most data sources, while also being applicable at the level of individual sources. One example of the proposed approach's application is in the field of medicine, where patient data is scattered across various departments and facilities. Integrating this data provides a more comprehensive picture of a patient's health, which supports more accurate diagnoses, ensures better treatment selection, and enables more efficient resource management.

Given a finite set S of decision trees, we investigate two types of inhibitory rules: *general inhibitory rules*, formed using any attributes occurring across the trees in S, and *inner inhibitory rules*, which are derived from complete root-to-leaf paths within individual trees. An inhibitory rule r is considered *true* for a tree $\Gamma \in S$ if an inner inhibitory rule r' exists in Γ such that the elementary conditions in r' are a subset of those in r, and both rules share the same conclusion.

This work addresses two key optimization problems: (i) identifying an inner inhibitory rule that holds for the maximum number of trees in S, and (ii) determining a general inhibitory rule that is valid for the largest subset of trees in S. These problems are particularly relevant in distributed learning environ-

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

ments, where multiple agents independently generate datasets using overlapping condition attributes and a shared decision attribute, producing separate decision trees. In this context, inhibitory rules that are valid across many trees can represent common knowledge.

We present a polynomial-time algorithm, denoted $\mathcal{A}^{inh}$, that solves the inner inhibitory rule optimization problem. We show also that optimizing general inhibitory rules is an NP-hard problem. To address this, we propose a heuristic algorithm $\mathcal{H}_1^{inh}$, which is an iterative method derived from $\mathcal{A}^{inh}$.

The main contribution of this work is a new approach to the problem of inducing inhibitory rules in a distributed data environment. The paper formalizes the problems of optimization of inner and general rules, and proposes dedicated algorithms for solving them. The effectiveness of the proposed methods was assessed in an experimental analysis on artificially generated datasets, aimed at comparing the number of induced inhibitory rules which are true for the maximum number of trees in the set. The experimental results showed that heuristic $\mathcal{H}_1^{inh}$ constructs rules that are true for a larger number of decision trees than the rules constructed by algorithm $\mathcal{A}^{inh}$.

The paper is organized as follows: Section 2 presents information related to decision trees. Section 3 contains the core definitions and preliminaries related to the considered problems. Optimization of inner and general inhibitory rules and algorithms proposed in this context is described. It was also shown that the general inhibitory optimization problem is NP-hard. Section 4 presents the experimental setup and discusses experimental results devoted to quality of algorithms $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$. Section 5 concludes the paper.

2 Decision Modelling in the Form of Trees

Decision trees are hierarchical structures used to model classification and regression problems in machine learning. In the context of classification, a decision tree recursively partitions the feature space into regions, each associated with a class label. Due to their explicit, rule-based nature, decision trees offer a significant advantage over many other classification methods: interpretability. Unlike neural networks or ensemble methods that often function as opaque models, a decision tree provides a transparent reasoning process that can be examined and understood by domain experts [4, 12].

Structurally, a decision tree consists of three types of elements. Internal nodes (including the root node) represent tests on attribute values, where each node evaluates a condition involving a single attribute. Branches emanating from an internal node correspond to the possible outcomes of the test: in binary trees, these are typically "yes" and "no" branches, while multi-way splits allow features to take multiple discrete values. Leaf nodes, also called terminal nodes, represent the final classification decision and are labeled with a class. The path from the root node to any leaf node traverses a sequence of tests on conditional attributes, and this sequence determines the outcome of classification of a particular entity being processed. The depth of a tree, defined as the length of the longest root-to-

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

leaf path, directly influences both the complexity of the model and its capacity to capture intricate decision boundaries.

Each root-to-leaf path in a decision tree corresponds to a decision rule of the form: *if* $c_1 \wedge c_2 \wedge \ldots \wedge c_n$ *then* $d = k$, where c_i are conditions on attribute values and $d = k$ denotes a decision, namely assignment to class k. These rules are mutually exclusive (any given instance satisfies exactly one path) and collectively exhaustive, covering all possible instances in the feature space. The length of a rule, measured by the number of conditions it contains, corresponds to the depth of the associated leaf in the tree. Shorter rules are generally preferred as they tend to generalize better to unseen data and are less prone to overfitting.

The construction of decision trees typically follows a top-down inductive approach, commonly referred to as Top-Down Induction of Decision Trees [12]. Starting from the root node, an algorithm selects an attribute to partition the training data, creating child nodes for each split. This process continues recursively until a stopping criterion is met, such as reaching pure leaf nodes or a maximum depth. The selection of attributes at each step is guided by heuristic measures that quantify the quality of a split. Information gain, based on entropy reduction, is employed by the ID3 and C4.5 algorithms [13], while the Gini impurity criterion is used by CART [4]. These algorithms are greedy in nature: they make locally optimal choices at each node without backtracking, which means that different attribute orderings or tie-breaking strategies can produce substantially different trees from the same training data.

This sensitivity to attribute ordering and changes in the data is a well-known limitation of decision trees. Small changes in the training set—such as the addition or removal of a few entries—can lead to significantly different tree structures [3]. Moreover, for any given dataset, multiple valid trees may exist that achieve comparable classification results. This inherent variability motivates the examination of decision rules not only within a single tree, but across a set of trees induced from the same data. By analyzing which rules appear consistently across multiple trees, one can identify more robust classification knowledge [3, 10]. Furthermore, this perspective opens the possibility of extracting rules that describe what an instance is *not* (inhibitory rules that specify class exclusion rather than class membership) providing an alternative representation of the classification knowledge embedded in decision trees.

3 Learning Inhibitory Rules from Sets of Decision Trees

This section presents the main notions related to the induction of inhibitory rules from sets of decision trees. Concepts related to general and inner inhibitory rules are defined along with proposed algorithms for defined problems of their optimization.

3.1 Main Notions

Let $\omega = \{0, 1, 2, \ldots\}$ denote the set of nonnegative integers, and let F be a set of binary attributes taking values in $\{0, 1\}$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

An *inhibitory rule* r over the attribute set F is an expression of the form:

$$(f_1 = \delta_1) \wedge \cdots \wedge (f_m = \delta_m) \rightarrow \neq t, \tag{1}$$

where $m \in \omega$, the attributes $f_1, \ldots, f_m$ are pairwise distinct elements of F, each $\delta_j \in \{0, 1\}$ for $1 \leq j \leq m$, and $t \in \omega$ denotes a decision value. The conjunction on the left-hand side is referred to as the set of elementary conditions, denoted $C(r)$, while the right-hand side expresses that the decision is different from t, written as $\neq t$, where $t(r)$ denotes the decision t.

Two inhibitory rules r_1 and r_2 are said to be *incompatible* if there exists an attribute f and a value δ such that the condition $f = \delta$ appears in $C(r_1)$, and $f = \neg\delta$ appears in $C(r_2)$, with $\neg 0 = 1$ and $\neg 1 = 0$.

A *decision tree* over the attribute set F is a labeled, directed tree with a root node, satisfying the following properties:

- Each leaf (terminal node) is labeled with a decision value from ω.
- Each internal (non-terminal) node is labeled with an attribute from F and has exactly two outgoing edges labeled 0 and 1.
- Along any root-to-leaf (complete) path, the internal node attributes are pairwise distinct.

Let S be a non-empty finite set of decision trees. Define $D(S)$ as the set of all decision values that appear at the leaves across the trees in S. For any tree $\Gamma \in S$, the number of leaves equals the number of internal nodes plus one. Each complete root-to-leaf path corresponds uniquely to a leaf.

Let ξ be a complete path in a tree Γ with m internal nodes. Denote by $d(\xi)$ the decision value at the leaf node at the end of ξ. For every $t \in D(S) \setminus \{d(\xi)\}$, we define the associated inhibitory rule $rule(\xi, t)$. If $m = 0$, then the rule $rule(\xi, t)$ has the form of eq. (2).

$$\rightarrow \neq t. \tag{2}$$

Otherwise, if the attributes $f_1, \ldots, f_m$ label the internal nodes along ξ, and $\delta_1, \ldots, \delta_m$ are the corresponding edge labels (0 or 1), then the rule $rule(\xi, t)$ has the form of eq. (3). The number m denotes the length of the rule.

$$(f_1 = \delta_1) \wedge \cdots \wedge (f_m = \delta_m) \rightarrow \neq t. \tag{3}$$

Let $\Xi(\Gamma)$ denote the set of all complete paths in Γ. The set of *inner inhibitory rules* for Γ is then defined as presented by eq. (4):

$$IIR(\Gamma) = \{rule(\xi, t) : \xi \in \Xi(\Gamma),\ t \in D(S) \setminus \{d(\xi)\}\}. \tag{4}$$

We further define:

- $IIR(S) = \bigcup_{\Gamma \in S} IIR(\Gamma)$ – the set of all inner inhibitory rules extracted from trees in S.
- $F(S)$ – the set of all attributes used in the internal nodes of trees in S.
- $GIR(S)$ – the set of all inhibitory rules over the attribute set $F(S)$ with right-hand side of the form $\neq t$ for some $t \in D(S)$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

We refer to elements of $IIR(S)$ as *inner inhibitory rules over* S, and elements of $GIR(S)$ as *general inhibitory rules over* S. Clearly, the inclusion holds, see eq. (5).

$$IIR(S) \subseteq GIR(S). \tag{5}$$

Given a tree $\Gamma \in S$ and a rule $r \in GIR(S)$, we say that r is *true* for Γ if there exists a rule $r' \in IIR(\Gamma)$ such that $t(r') = t(r)$ and $C(r') \subseteq C(r)$.

3.2 Optimization of Inner Inhibitory Rules

This section focuses on the *Inner Inhibitory Optimization Problem* (IIOP), defined as follows: Given a set of decision trees S, identify an inner inhibitory rule that holds for the maximum number of trees in S.

Let $T(S)$ denote the total number of nodes–both internal and terminal–across all trees in S. The total number of inner inhibitory rules in $IIR(S)$ is upper-bounded by $T(S)^2$. Furthermore, the number of conditions on the left-hand side of any rule in $IIR(S)$ is at most $T(S)$. As a result, the entire set $IIR(S)$ can be generated in time polynomial in $T(S)$.

We begin by presenting algorithm $\mathcal{A}_0^{inh}$, which verifies whether a given rule $r \in IIR(S)$ is valid for a specific tree $\Gamma \in S$. This algorithm operates in polynomial time relative to $T(S)$.

Algorithm $\mathcal{A}_0^{inh}$

For each rule $r' \in IIR(\Gamma)$, check whether $t(r') = t(r)$ and $C(r') \subseteq C(r)$. If such a rule r' exists, then r is considered true for Γ; otherwise, it is not.

We now introduce the main algorithm, $\mathcal{A}^{inh}$, which solves the IIOP by searching for the optimal rule within a specified subset $I \subseteq IIR(S)$. This algorithm identifies, in polynomial time with respect to $T(S)$, a rule from I that is valid for the largest number of trees in S.

Algorithm $\mathcal{A}^{inh}$

For each rule $r \in I$ and each tree $\Gamma \in S$, apply algorithm $\mathcal{A}_0^{inh}$ to determine whether r is true for Γ. After evaluating all rules, return the one that is valid for the greatest number of trees in S.

3.3 Optimization of General Inhibitory Rules

In this section, we consider the *General Inhibitory Optimization Problem* (GIOP), which aims to find, for a given set of decision trees S, a general inhibitory rule over S that is valid for the largest number of trees in S.

Let $F(S) = \{f_1, \ldots, f_m\}$ represent the set of attributes appearing in the trees of S. It can be shown that an optimal solution to GIOP always exists among the so-called *complete inhibitory rules over* S, defined as in eq. (6),

$$(f_1 = \delta_1) \wedge \cdots \wedge (f_m = \delta_m) \rightarrow \neq t, \tag{6}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

where each $\delta_i \in \{0, 1\}$ and $t \in D(S)$.

We now demonstrate that solving GIOP is an NP-hard problem. This is proven via a polynomial-time reduction from the classical NP-complete problem 3-SAT.

A Boolean formula known as 3-conjunctive normal form (3-CNF) has the structure $N(x_1, \ldots, x_n) = C_1 \wedge \cdots \wedge C_k$, where each clause C_j $(1 \leq j \leq k)$ consists of exactly three distinct literals connected by logical OR. Each literal is either a Boolean variable from $\{x_1, \ldots, x_n\}$ or its negation, and every variable from $\{x_1, \ldots, x_n\}$ appears in at least one clause.

3-SAT Problem: Given a 3-CNF formula $N(x_1, \ldots, x_n)$, determine whether there exists a truth assignment $(a_1, \ldots, a_n) \in \{0, 1\}^n$ such that $N(a_1, \ldots, a_n) = 1$.

To reduce 3-SAT to GIOP, we construct a set of decision trees $S_N = \{\Gamma_0, \Gamma_1, \ldots, \Gamma_k\}$ based on the formula $N = C_1 \wedge \cdots \wedge C_k$. Each clause $C_i = x_{t(i,1)}^{b_{i1}} \vee x_{t(i,2)}^{b_{i2}} \vee x_{t(i,3)}^{b_{i3}}$ (where $b_{ij} \in \{0, 1\}$, with $x^0 = \neg x$ and $x^1 = x$) corresponds to a decision tree Γ_i.

Note that the condition $x^b = 1$ is satisfied precisely when $x = b$. The tree Γ_0 plays an auxiliary role, while each Γ_i encodes a specific clause. Figure 1 illustrates the structure of Γ_0 and a representative clause tree Γ_i. All decision trees in S_N use decisions from the set $D(S_N) = \{0, 1\}$.

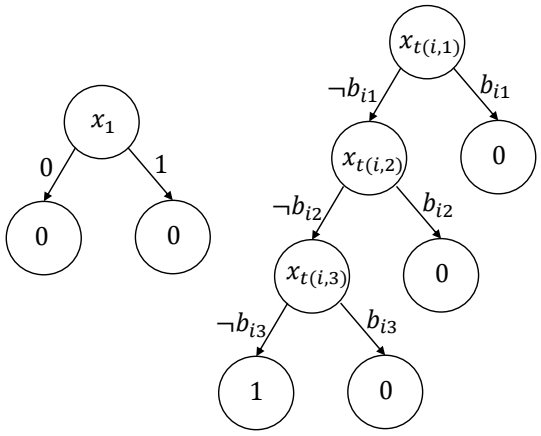

Fig. 1: Decision trees Γ_0 and Γ_i, $i = 1, \ldots, k$

We now prove the equivalence between the satisfiability of the formula N and the existence of a complete inhibitory rule over S_N that is valid for every tree in S_N.

Assume there exists a truth assignment $(a_1, \ldots, a_n) \in \{0, 1\}^n$ such that $N(a_1, \ldots, a_n) = 1$. Then each clause C_i evaluates to true under this assign-

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

ment. That is, for every $i \in \{1, \ldots, k\}$, there exists an index $p_i \in \{1, 2, 3\}$ such that $a_{t(i,p_i)} = b_{ip_i}$.

Using this assignment, we define the following complete inhibitory rule:

$$(x_1 = a_1) \wedge \cdots \wedge (x_n = a_n) \rightarrow \neq 1.$$

This rule can be verified to be valid for all trees in S_N.

Conversely, suppose there exists a complete inhibitory rule r over S_N that is true for every tree in the set. Then r must have the form:

$$(x_1 = a_1) \wedge \cdots \wedge (x_n = a_n) \rightarrow \neq 1.$$

Since r holds for each tree Γ_i, the corresponding clause C_i must be satisfied under the assignment $(a_1, \ldots, a_n)$. Hence, the entire formula N is satisfied.

This shows a polynomial-time reduction from 3-SAT to GIOP: Given a 3-CNF formula N, we construct the associated decision tree set S_N in polynomial time. Solving GIOP on S_N involves finding a general inhibitory rule that is valid for the maximum number of trees. We then verify in polynomial time whether the rule is valid for all trees.

If the considered rule is not valid for all trees, then N is unsatisfiable. If this rule holds for every tree, then it can be extended to a complete inhibitory rule over S that is valid for every tree from S_N. Thus, N is satisfiable.

Therefore, GIOP is NP-hard.

3.4 Heuristic for General Inhibitory Rule Optimization

As established earlier, the general inhibitory rule optimization problem (GIOP) is NP-hard. Consequently, this section introduces heuristic approach designed to approximate solutions to GIOP in polynomial time.

The heuristic, denoted $\mathcal{H}_1^{inh}$, extends the inner inhibitory optimization algorithm $\mathcal{A}^{inh}$ and relies on a supporting procedure $\mathcal{B}^{inh}$, described below. This method runs in time polynomial with respect to $T(S)$.

Heuristic $\mathcal{H}_1^{inh}$

For each decision value $t \in D(S)$, execute algorithm $\mathcal{B}^{inh}$ on the decision tree set S. Among all rules generated for different values of t, select the one that is valid for the largest number of trees in S.

Algorithm $\mathcal{B}^{inh}$

Step 0. Construct the initial set I_0 containing all rules from $IIR(S)$ with right-hand side $\neq t$ and a non-empty left-hand side. Initialize rule r_0 with an empty left-hand side and right-hand side $\neq t$.

Assume that step i $(i \geq 0)$ produces a rule r_i and a corresponding rule set I_i.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

Step $i+1$. Apply algorithm $\mathcal{A}^{inh}$ to the current set I_i to identify a rule ρ_{i+1} that holds for the largest number of trees in S. Define the new rule r_{i+1} with right-hand side $\neq t$ and left-hand side $C(r_{i+1}) = C(r_i) \cup C(\rho_{i+1})$.

Remove from I_i all rules that are either:

- incompatible with r_{i+1}, or
- have condition sets that are subsets of $C(r_{i+1})$.

Let I_{i+1} denote the resulting rule set.

If I_{i+1} is empty, terminate and return r_{i+1}; otherwise, proceed to the next step.

4 Experiments

This section presents a comparative experimental evaluation of the algorithm $\mathcal{A}^{inh}$ and the heuristic $\mathcal{H}_1^{inh}$, using synthetically generated sets of decision trees induced with the ID3 algorithm and entropy as attribute selection criterion. The aim is to compare the number of trees for which the constructed rules are true and their length.

4.1 Experimental Setup

We recall the main interpretation of the problem addressed in the paper. We consider t agents working on similar tasks who utilize attributes drawn from a common set of a attributes. Each agent constructs its own decision table and produces a decision tree based on it. Consequently, we obtain a collection S consisting of t decision trees. The objective is to identify rules that hold for as many trees in S as possible. Considering that the length of a rule plays a significant role in assessing its quality and interpretability, additional experiments and analyses were also conducted concerning the length of rules induced by the compared algorithms.

The study involved two groups of experiments. The key distinction between them lies in the selection of attributes in the decision tables for each agent. In the first group, all attributes from the common set are used, whereas in the second one, only 50% of attributes from the common set is selected. Denote

- $T = \{10, 20, 30, 40, 50\}$,
- $A_1 = \{10, 20, 30, 40, 50\}$,
- $A_2 = \{20, 40, 60, 80, 100\}$.

The first group of experiments

Fix $t \in T$ and $a \in A_1$. Construct a decision table with a columns labeled with the attributes $\{f_1, \ldots, f_a\}$ and $2a$ pairwise different rows chosen randomly. Rows are filled with numbers from the set $\{0, 1\}$. They are labeled with decisions chosen randomly from the set $\{1, 2, 3, 4\}$.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

Construct a decision tree for this table. Repeat the described step t times (each time, a new table is constructed). As a result, a set S_1 of t decision trees is obtained.

After repeating the whole procedure 5 times, we obtain ensembles $S_1, \ldots, S_5$ of decision trees. Specifically, this results in five ensembles with 10 trees, five with 20 trees, and so on, up to five ensembles with 50 trees.

For $i = 1, \ldots, 5$, apply the algorithm $\mathcal{A}^{inh}$ to S_i. As a result, we obtain a decision rule r_i. Let $true_i$ be the number of trees from S_i for which the rule r_i is true. Let $length_i$ be the length of this rule. For the fixed a and t, the quality of the algorithm $\mathcal{A}^{inh}$ is the number defined by eq. (7).

$$\frac{true_1 + \cdots + true_5}{5t}. \tag{7}$$

For the fixed a and t, the quality of the heuristic $\mathcal{H}_1^{inh}$ is obtained similarly. We find the quality of $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ for each $t \in T$ and $a \in A_1$.

Considering the length of the rule r_i, for the fixed a and t, the quality of the algorithm $\mathcal{A}^{inh}$ and heuristic $\mathcal{H}_1^{inh}$ in this context, is the number defined by eq. (8).

$$\frac{length_1 + \cdots + length_5}{5}. \tag{8}$$

The second group of experiments

This collection of experiments differs from the first one in the way how the decision tables are constructed. Fix $t \in T$ and $a \in A_2$. Construct a decision table with $a/2$ columns labeled with the pairwise different attributes chosen randomly from the set $\{f_1, \ldots, f_a\}$ and a pairwise different rows chosen randomly. Rows are filled with numbers from the set $\{0, 1\}$. They are labeled with decisions chosen randomly from the set $\{1, 2, 3, 4\}$. The remaining steps of the procedure are the same as in the first group of experiments.

4.2 Results of Experiments

Tables 1a and 1b present results for the first group of experiments. At the intersection of row $a \in A_1$ and column $t \in T$ we have two numbers. The top one is the quality of $\mathcal{A}^{inh}$ and the bottom one is the quality of $\mathcal{H}_1^{inh}$ for a and t. The column Avg contains average values of qualities $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ for given a. The row Avg contains average values of qualities $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ for given t. At the intersection of the row Avg and the column Avg, we have average values of qualities $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ for all a and t.

Table 1a presents the results concerning the number of trees, for which the constructed rule is true (see eq. (7), Table 1b – results related to the length of such rule (see eq. (8)).

Tables 2a and 2b present results for the second group of experiments. At the intersection of row $a \in A_1$ and column $t \in T$ we have two numbers. The top one is the quality of $\mathcal{A}^{inh}$ and the bottom one is the quality of $\mathcal{H}_1^{inh}$ for a and t.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

Table 1: Quality of algorithms $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ related to the first group of experiments.

(a) Results related to the number of trees for which the constructed rules are true.

		t					Avg
		10	20	30	40	50	
a	10	0.360	0.380	0.327	0.365	0.348	0.356
		1.000	0.910	0.893	0.865	0.848	0.903
	20	0.200	0.110	0.107	0.080	0.068	0.113
		1.000	0.920	0.867	0.845	0.860	0.898
	30	0.140	0.090	0.073	0.055	0.044	0.080
		1.000	0.950	0.913	0.885	0.860	0.922
	40	0.100	0.070	0.067	0.055	0.044	0.067
		1.000	0.960	0.913	0.870	0.872	0.923
	50	0.120	0.060	0.040	0.040	0.036	0.059
		1.000	0.990	0.947	0.905	0.876	0.944
	Avg	0.184	0.142	0.123	0.119	0.108	0.135
		1.000	0.946	0.907	0.874	0.863	0.918

(b) Results related to the length of constructed rules.

		t					Avg
		10	20	30	40	50	
a	10	5.8	6.4	6.6	7.0	7.0	6.56
		10.0	10.0	10.0	10.0	10.0	10.00
	20	5.4	5.0	6.4	6.4	6.0	5.84
		18.6	20.0	19.8	20.0	20.0	19.68
	30	5.4	5.6	5.4	5.4	5.0	5.36
		27.2	28.6	30.0	30.0	30.0	29.16
	40	5.2	5.8	5.8	6.2	5.4	5.68
		31.0	37.6	39.6	40.0	49.4	39.52
	50	5.6	5.8	6.0	5.8	6.0	5.84
		36.8	46.6	48.4	49.4	50.0	46.24
	Avg	5.48	5.72	6.04	6.16	5.88	5.86
		24.72	28.56	29.56	29.88	31.88	28.92

The column Avg contains average values of qualities $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ for given a. The row Avg contains average values of qualities $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ for given t. At the intersection of the row Avg and the column Avg, we have average values of qualities $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ for all a and t.

Table 2a presents the results concerning the number of trees, for which the constructed rule is true (see eq. (7), Table 2b – results related to the length of such rule (see eq. (8)).

All experiments were conducted on a system equipped with an ARM-based processor featuring 8 cores (6 performance cores at 3.2 GHz and 2 efficiency cores at 2.0 GHz) and 16 GB of LPDDR5 unified memory. The implementation was developed in Python 3.13.8 using Jupyter notebooks for organization of experiments and core Python code for implementing core structures and modules, with NumPy 2.3.3 for numerical computations and Pandas 2.3.2 for data manipulation, and Plotly 6.5.1 for 3D visualizations.

Based on the obtained results, it is possible to see that heuristics $\mathcal{H}_1^{inh}$ produces rules that are true for more trees than rules produced by algorithm $\mathcal{A}^{inh}$, for both groups of experiments (see Tables 1a and 2a). It can also be seen that the length of the rules induced by algorithm $\mathcal{A}^{inh}$ is significantly shorter than in the case of algorithm $\mathcal{H}_1^{inh}$.

Considering the length of the rules, it can be observed that it increases with the number of attributes and the number of trees in the case of heuristic $\mathcal{H}_1^{inh}$. Often, the length of a constructed rule reaches a value equal to or close to the number of attributes in the set. Although algorithm $\mathcal{A}^{inh}$ produces rules that are true for less number of trees in set S, it should be emphasized that they are relatively short and do not change significantly with an increase in the number

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

Table 2: Quality of algorithms $\mathcal{A}^{inh}$ and $\mathcal{H}_1^{inh}$ related to the second group of experiments.

(a) Results related to the number of trees for which the constructed rules are true.

		t					Avg
		10	20	30	40	50	
a	20	0.200	0.150	0.133	0.110	0.096	0.138
		1.000	0.980	0.913	0.870	0.868	0.926
	40	0.160	0.080	0.067	0.050	0.040	0.079
		1.000	0.980	0.933	0.915	0.888	0.943
	60	0.100	0.060	0.060	0.050	0.040	0.062
		1.000	1.000	0.960	0.925	0.908	0.959
	80	0.100	0.050	0.040	0.030	0.024	0.049
		1.000	0.990	0.987	0.930	0.920	0.965
	100	0.100	0.050	0.040	0.030	0.024	0.049
		1.000	1.000	0.987	0.960	0.928	0.975
	Avg	0.132	0.078	0.068	0.054	0.045	0.075
		1.000	0.990	0.956	0.920	0.902	0.954

(b) Results related to the length of constructed rules.

		t					Avg
		10	20	30	40	50	
a	20	5.0	5.6	5.8	6.0	6.2	5.72
		17.4	19.8	20.0	20.0	20.0	19.44
	40	5.0	5.2	6.2	5.2	5.0	5.32
		28.8	36.2	38.0	39.6	40.0	36.52
	60	4.6	5.0	5.4	5.4	5.2	5.12
		37.4	51.0	56.4	57.6	58.6	52.20
	80	5.2	5.4	5.6	5.4	5.4	5.40
		42.8	60.0	71.6	75.2	77.8	65.48
	100	5.8	5.6	6.4	5.8	6.0	5.92
		42.8	73.0	83.8	89.0	95.6	76.84
	Avg	5.12	5.36	5.88	5.56	5.56	5.50
		33.84	48.00	53.96	56.28	58.40	50.10

of trees and attributes in the set. Determining what rule length is acceptable in terms of interpretability depends on the specific application. It should also be noted that algorithm $\mathcal{A}^{inh}$ and heuristic $\mathcal{H}_1^{inh}$ are used for different optimization problems.

The figures, Fig 2 and Fig. 3, provide a visualization of the results presented in the Tables 1a and 2a, respectively.

In the case of Fig. 2 on the left-hand side, results related to the quality of algorithm $\mathcal{A}^{inh}$ are presented, on the right-hand side – results related to heuristic $\mathcal{H}_1^{inh}$, for the first group of experiments where 100% of attributes is considered during construction of decision tress.

Figure 3 presents on the left-hand side results related to the quality of algorithm $\mathcal{A}^{inh}$, on the right-hand side – results related to heuristic $\mathcal{H}_1^{inh}$, for the second group of experiments where 50% of attributes is considered during construction of decision trees.

Based on graphical representation of results on Figs. 2 and 3, it is possible to note that the results obtained by algorithm $\mathcal{A}^{inh}$ mostly demonstrate a rapid decline in quality measure as a (the number of attributes) increases as well as a much more moderate decline in quality as t (the number of trees) increases. It appears to be more exaggerated on the left hand side of Fig. 2, whereas on Fig. 3 the relation of quality to these parameters looks more uniform. As it has been mentioned above, algorithm $\mathcal{H}_1^{inh}$ produces significantly higher quality metric values which can also be derived from the scale of the Z-axis on Figs. 2 and 3. On the contrary, the obtained values represent different trends in relation of quality to number of attributes and trees, where there is an overall increase in quality measure with the increase of a and a noticeable decrease of quality as

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

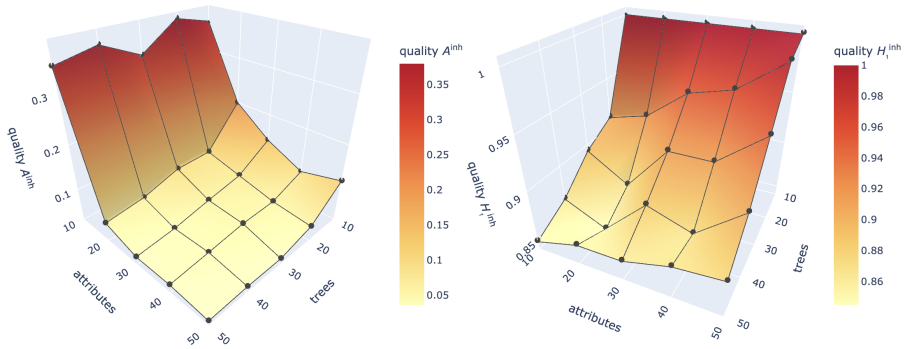

Fig. 2: The number of trees for which the constructed rules are true: for rules obtained by algorithm $\mathcal{A}^{inh}$ (the left-hand side) and heuristics $\mathcal{H}_1^{inh}$ (the right-hand side), for selection of 100% of attributes from the set.

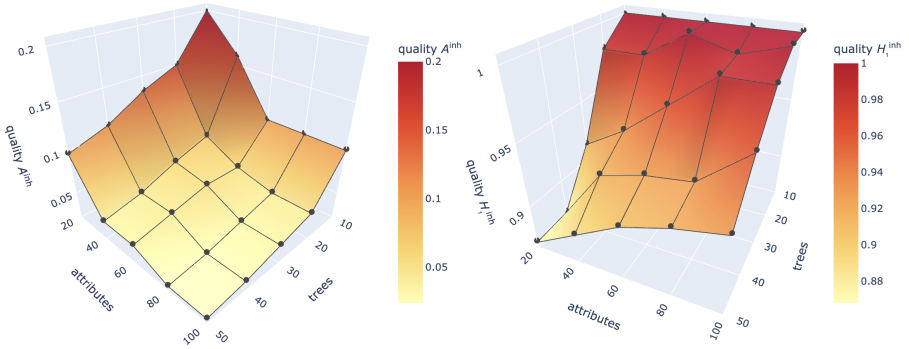

Fig. 3: The number of trees for which the constructed rules are true: for rules obtained by algorithm $\mathcal{A}^{inh}$ (the left-hand side) and heuristics $\mathcal{H}_1^{inh}$ (the right-hand side), for selection of 50% of attributes from the set.

the number of trees t increases, that can be observed on both right side graphs of Figs. 2 and 3, leading to a conclusion that algorithm $\mathcal{H}_1^{inh}$ produces results with higher number of attributes.

5 Conclusions

Inhibitory rules and decision trees play an important role in decision-making systems and data analysis because they enable transparent and interpretable representation of knowledge. Decision trees provide a hierarchical decision structure in which each path from the root to the leaf can be written as a decision rule. Inhibitory rules are particularly important in situations where is a need to clearly define the conditions under which a decision should not be made. They

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

enable the formal expression of restrictions or exceptions, which increases the precision and reliability of decision models.

The length of inhibitory rules, understood as the number of conditions in their premise, is an important factor affecting both the interpretability and quality of decision models. Shorter rules are usually easier to understand and analyze, which increases their usefulness in practical applications requiring transparency of the decision-making process. On the other hand, longer rules, although potentially more precise, can lead to excessive complexity and make it difficult to interpret the results.

In this paper, we defined and investigated the problem of learning inner and general inhibitory rules valid for the maximum number of decision trees in a given set. The novelty of the paper is related to the first application of inhibitory rules for discovering pattern from distributed data. It has been shown that the problem of optimizing general inhibitory rules is NP-hard, and a heuristic $\mathcal{H}_1^{inh}$ for inducing such rules has been proposed. In the case of the problem of optimization of inner rules, algorithm $\mathcal{A}^{inh}$ has been proposed. Both algorithms were compared, taking into account the maximum number of trees for which the induced rules are true, and their length, for different numbers of trees in the set S and different numbers of attributes. Moreover, performed experiments were related to two scenarios, where decision trees are constructed using the full set of attributes, and where only 50% of attributes from the entire set is avaiable. The results obtained show that the algorithm $\mathcal{A}^{inh}$ induces significantly shorter rules than the heuristic $\mathcal{H}_1^{inh}$ however, taking into account the number of trees for which rules are true, the heuristic $\mathcal{H}_1^{inh}$ gives better results.

The proposed approach can be applied in organizations with a distributed structure, such as hospitals, where individual units operate based on their own knowledge resources. The knowledge accumulated in each of these units is represented in the form of decision trees. The aim is to identify patterns common to a significant proportion of the units that reflect the knowledge characteristic of the entire organization.

In the future, we plan to consider the problem of learning not just a single inhibitory rule (inner or general) valid for the maximum number of decision trees in a given set, but a group of rules, each valid for a number of trees close to the maximum. The proposed methods will be generalized to handle k-valued attributes for any $k \geq 3$. Future work will include also empirical validation using real-world datasets and comparison with other approaches for inducing rules.

Acknowledgments. Research reported in this publication was supported by the Remote Working Individual Consultancy Agreement No. KAUST-2025-C0148, and King Abdullah University of Science and Technology (KAUST) and the Institute of Computer Science, University of Silesia in Katowice, Poland.

References

1. Alsolami, F., Azad, M., Chikalov, I., Moshkov, M.: Decision and Inhibitory Trees and Rules for Decision Tables with Many-valued Decisions, Intelligent Systems

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19

Reference Library, vol. 156. Springer (2020)
2. Alsolami, F., Chikalov, I., Moshkov, M., Zielosko, B.: Optimization of approximate inhibitory rules relative to number of misclassifications. In: Watada, J., Jain, L.C., Howlett, R.J., Mukai, N., Asakura, K. (eds.) 17th International Conference in Knowledge Based and Intelligent Information and Engineering Systems, KES 2013, Kitakyushu, Japan, 9-11 September 2013. Procedia Computer Science, vol. 22, pp. 295–302. Elsevier (2013)
3. Breiman, L.: Bagging predictors. Machine Learning **24**(2), 123–140 (1996)
4. Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A.: Classification and Regression Trees. Chapman and Hall/CRC (1984)
5. Delimata, P., Moshkov, M.J., Skowron, A., Suraj, Z.: Comparison of lazy classification algorithms based on deterministic and inhibitory decision rules. In: Wang, G., Li, T., Grzymala-Busse, J.W., Miao, D., Skowron, A., Yao, Y. (eds.) Rough Sets and Knowledge Technology, Third International Conference, RSKT 2008, Chengdu, China, May 17-19, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5009, pp. 55–62. Springer (2008). https://doi.org/10.1007/978-3-540-79721-0_13
6. Delimata, P., Moshkov, M.J., Skowron, A., Suraj, Z.: Inhibitory Rules in Data Analysis: A Rough Set Approach, Studies in Computational Intelligence, vol. 163. Springer (2009)
7. Kozielski, M., Sikora, M., Wawrowski, L.: Towards consistency of rule-based explainer and black box model - fusion of rule induction and xai-based feature importance. Knowl. Based Syst. **311**, 113092 (2025). https://doi.org/10.1016/J.KNOSYS.2025.113092
8. Moshkov, M.: Time complexity of decision trees. In: Peters, J.F., Skowron, A. (eds.) Trans. Rough Sets III, Lecture Notes in Computer Science, vol. 3400, pp. 244–459. Springer (2005)
9. Moshkov, M., Skowron, A., Suraj, Z.: Maximal consistent extensions of information systems relative to their theories. Inf. Sci. **178**(12), 2600–2620 (2008). https://doi.org/10.1016/J.INS.2008.01.018
10. Moshkov, M., Zielosko, B.: Combinatorial Machine Learning: A Rough Set Approach, Studies in Computational Intelligence, vol. 360. Springer (2011)
11. Pawlak, Z., Skowron, A.: Rudiments of rough sets. Inf. Sci. **177**(1), 3–27 (2007). https://doi.org/10.1016/J.INS.2006.06.003
12. Quinlan, J.R.: Induction of decision trees. Machine Learning **1**(1), 81–106 (1986)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA (1993)
14. Sakai, H., Nakata, M., Slezak, D., Watada, J.: Rule generation in rough set non-deterministic information analysis (RNIA) and some applications of the obtained rules. Appl. Soft Comput. **172**, 112842 (2025). https://doi.org/10.1016/J.ASOC.2025.112842
15. Yao, J., Cornelis, C., Wang, G., Yao, Y.: Uncertainty and three-way decision in data science. Int. J. Approx. Reason. **162**, 109024 (2023). https://doi.org/10.1016/J.IJAR.2023.109024, https://doi.org/10.1016/j.ijar.2023.109024
16. Zielosko, B.: Application of dynamic programming approach to optimization of association rules relative to coverage and length. Fundam. Informaticae **148**(1-2), 87–105 (2016). https://doi.org/10.3233/FI-2016-1424
17. Zielosko, B., Moshkov, M., Tetteh, E.T.: Optimization of inner and general rules. Inf. Sci. **719**, 122466 (2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29915-4_19