



Article Paired Patterns in Logical Analysis of Data for Decision Support in Recognition

Igor S. Masich, Vadim S. Tyncheko *¹⁰, Vladimir A. Nelyub, Vladimir V. Bukhtoyarov, Sergei O. Kurashkin *¹⁰ and Aleksey S. Borodulin

Artificial Intelligence Technology Scientific and Education Center, Bauman Moscow State Technical University, 105005 Moscow, Russia

* Correspondence: vadimond@mail.ru (V.S.T.); scorpion_ser@mail.ru (S.O.K.); Tel.: +7-95-0973-0264 (V.S.T.)

Abstract: Logical analysis of data (LAD), an approach to data analysis based on Boolean functions, combinatorics, and optimization, can be considered one of the methods of interpretable machine learning. A feature of LAD is that, among many patterns, different types of patterns can be identified, for example, prime, strong, spanned, and maximum. This paper proposes a decision-support approach to recognition by sharing different types of patterns to improve the quality of recognition in terms of accuracy, interpretability, and validity. An algorithm was developed to search for pairs of strong patterns (prime and spanned) with the same coverage as the training sample, having the smallest (for the prime pattern) and the largest (for the spanned pattern) number of conditions. The proposed approach leads to a decrease in the number of unrecognized observations (compared with the use of spanned patterns only) by 1.5–2 times (experimental results), to some reduction in recognition errors (compared with the use of prime patterns only) of approximately 1% (depending on the dataset) and makes it possible to assess in more detail the level of confidence of the recognition result due to a refined decision-making scheme that uses the information about the number and type of patterns covering the observation.

Keywords: interpretable machine learning; logical analysis of data; prime patterns; spanned patterns; rule-based classifiers

1. Introduction

Different machine-learning methods based on various ideas and assumptions (inductive bias) are known to solve classification problems in recognition [1,2]. The choice of method for a particular problem is determined not only by the estimated classification accuracy but also by the situation in question, as well as by the purpose for which the machine generates a recognition solution. Thus, for some tasks, black-box predictive models are applicable, but there are also those tasks that require the interpretation of the solution and the justification of the result, for example [3–5]. The most suitable method for these situations can be summarized as "interpretable machine learning" [6,7]. Such methods make it possible to build recognition and prediction systems that provide user-interpretable results [8]. These are decision-support systems for recognition that not only assign a new object to a certain class but also answer the following questions: (1) Why does the object belong to this class? (2) How confident is this recognition? (3) Which features are the most influential? (4) How far the object is from the "class boundary"? (6) Other Questions.

In this study, the logical analysis of data (LAD) methodology was used to build decision-support systems for recognition. LAD is based on finding logical expressions (patterns) in the data, which summarize many examples of the same class using Boolean functions [9–11]. A decision rule for recognition was generated from a set of patterns [12].

A broad overview of the main achievements and applications of LAD can be found in [13,14]. The recent advances in the theory and practice of the logical analysis of data are described in [14].



Citation: Masich, I.S.; Tyncheko, V.S.; Nelyub, V.A.; Bukhtoyarov, V.V.; Kurashkin, S.O.; Borodulin, A.S. Paired Patterns in Logical Analysis of Data for Decision Support in Recognition. *Computation* **2022**, *10*, 185. https://doi.org/10.3390/ computation10100185

Academic Editor: Demos T. Tsahalis

Received: 20 September 2022 Accepted: 11 October 2022 Published: 12 October 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). The further development of LAD may be due to its faster processing as a result of its ability to process large volumes of data, as well as its increasing interpretability.

LAD, in its original form, is a rather laborious computational procedure [15], which may limit its practical application for the analysis of large volumes of data. However, there are ways to accelerate the LAD process. In [16,17], a technique based on ensembles and the merging of LAD models obtained from subsamples of data was proposed to accelerate the LAD process.

One of the most interesting directions in the development of LAD is the construction of a compact classifier. This requires the selection of those features with combinatorial effects [18,19]. It is also important to select the most informative and significant patterns [20,21] and to form a LAD model (decision rule) [22].

A feature of LAD is that, among many patterns (logical rules), different types of patterns can be identified, for example, prime, strong, spanned, and maximum [23–25]. The use of particular types of patterns makes it possible to place the right emphasis when building a decision-support system: making the rules simpler or more selective, paying attention to reducing recognition errors, or decreasing the proportion of unrecognized cases.

This paper proposes a decision-support approach to recognition by sharing different types of patterns to improve the quality of recognition in terms of accuracy, interpretability, and validity.

2. Patterns in LAD

Consider the problem of recognizing the objects described by binary features and dividing them into two classes: $K = K^+ \cup K^- \subset B_2^n$, where $B_2^n = \{0,1\}^n B_2^n = B_2 \cdot B_2 \cdot \ldots \cdot B_2$. The classes do not intersect: $K^+ \cap K^- = \emptyset$.

An observation $X \in K$ is described by a binary vector $X = (x_1, x_2, ..., x_n)$ and can be represented as a point in the hypercube of the binary feature space B_2^n . The observations of class K^+ will be called the positive sampling points of K, and the observations of K^- will be referred to as negative sampling points.

Consider a subset of points from B_2^n , in which some variables are fixed and identical, and others take an arbitrary value Equation (1) [26]:

$$T = \left\{ x \in B_2^n | x_i = 1 \text{ for } \forall i \in A \text{ and } x_i = 0 \text{ for } \forall j \in B \right\},\tag{1}$$

for some subsets of $A, B \subseteq \{1, 2, ..., n\}, A \cap B =, \emptyset$. This set can also be defined as a Boolean function that takes the true value for the elements of the set $t(x) = (\bigwedge_{i \in A} x_i) \land (\bigwedge_{i \in B} \overline{x}_i)$.

The set of points *x* for which t(x) = 1 denotes S(t). S(t) is a subcube in the Boolean hypercube B_2^n . The number of points in the subcube is $2^{(n-|A|-|B|)}$.

A binary variable x_i or its negation \overline{x}_i in a term is called literal. The notation x_i^{α} denotes x_i , if $\alpha = 1$, and \overline{x}_i , if $\alpha = 0$. Thus, a term is a conjunction of different literals that do not contain some variable and its negation at the same time. We denote the set of literals in term *t* as *Lit*(*t*).

Consider that the term t covers the point $a \in B_2^n$; if t(a) = 1, then this point belongs to the corresponding subcube.

The basic concept of the logical analysis of data is the notion of a pattern. A positive pattern is a subcube of an entire hypercube that intersects with K^+ and does not intersect with K^- [27]. Negative patterns have similar definitions.

In other words, pattern P is a term that covers at least one observation of a class and does not cover any observation of another class. That is, the pattern corresponds to a subcube that has a non-empty intersection with one of the sets (K^+ or K^-) and an empty intersection with another set (K^- or K^+). Pattern P, which does not intersect with K^- , will be referred to as positive, and pattern P', which does not intersect with K^+ , will be called negative.

More formally [26], term C is called a positive (negative) dataset pattern (K^+ , K^-) if

• C (w) = 0 for every
$$w \in K^-$$
 ($w \in K^+$), and;

The set of observations covered by pattern P is denoted as Cov(P). Patterns are elementary blocks used for constructing logical decision functions.

2.1. Example 1

Consider the binary dataset presented in Table 1. In this table, *a*, *b*, *c*, *d*, and *e* are positive observations, and *f*, *g*, *h*, *i* and *j* are negative observations. For example, it is possible to verify that $x_1x_3x_4$ is a positive pattern, and $\overline{x}_1\overline{x}_3x_4$ is negative.

		x ₁	x ₂	x ₃	x ₄	x ₅
	а	1	0	1	1	0
	b	0	1	0	0	1
K^+	с	1	1	1	1	1
	d	1	1	0	1	1
	e	1	1	1	1	0
	f	1	0	0	1	0
	g	1	0	0	0	0
K-	ň	1	1	0	0	0
	i	0	0	0	1	0
	j	0	0	0	1	1

Table 1. Binary data (*K*⁺, *K*⁻).

As the terms are geometrically interpreted as the subcubes of an *n*-dimensional cube $\{0,1\}^n$, positive (negative) patterns correspond to those subcubes that intersect set K^+ (K^-) but do not intersect set K^- (K^+).

Consider Example 1 again. The term $C = \overline{x}_1 \overline{x}_4 x_5$ has a positive pattern. The set of points for which *C* takes a value of 1, that is, the points for which $x_1 = 0$, $x_4 = 0$, $x_5 = 1$, is subcube $Q = \{(00001), (00101), (01001), (01101)\}$.

As the properties of positive and negative patterns are completely symmetric, without any loss of generality, we focus on positive patterns and refer to positive patterns simply as patterns.

As patterns play a central role in LAD, different types of patterns (e.g., prime, spanned, maximum) were studied, algorithms were developed to enumerate them [25,28], and their relative effectiveness was analyzed [29,30].

Unfortunately, there is no single and unambiguous criterion for comparing patterns. Different data may have different requirements for the quality and features of the formed patterns. In accordance with [18], three partial-order relations—simplicity, selectivity, and evidence—as well as their possible combinations are used to assess the quality of pure (homogeneous, without covering observations of other classes) patterns.

The simplicity (or compactness) relationship is often used to compare patterns, including those produced by different learning algorithms. Pattern P_1 is preferred to P_2 with respect to *simplicity* (denoted as $P_1 \succ P_2$), if $Lit(P_1) \subseteq Lit(P_2)$.

Pattern *P* is *prime* if after removing any literal from Lit(P), a term that is not a (pure) pattern (i.e., it covers the observations of another class) is formed. Evidently, the optimality of a pattern with respect to simplicity is identical to the statement that this pattern is prime.

2.2. Example 2

The prime pattern Equation (2) can be specified in the binary dataset shown in Table 1.

$$x_3, \overline{x}_1 \overline{x}_4, \overline{x}_4 x_5, x_2 x_4, x_1 x_5, \overline{x}_1 x_2, x_2 x_5,$$
(2)

In contrast, the pattern x_3x_4 gives an example of a non-prime pattern.

The search for simpler patterns requires consideration. First, such patterns are better interpretable and understandable for a person who uses them to make a decision. Second,

simpler patterns are often considered to have better generalizability, and their use leads to better recognition accuracy. However, this claim is controversial, and, as will be considered later, reducing simplicity can lead to higher accuracy.

The use of simple, short patterns reduces the number of incorrectly recognized positive observations (false negatives) but can also increase the number of incorrectly recognized negative observations (false positives). A natural way to reduce the number of false positives is to form more selective patterns, which is achieved by reducing the size of the subcube defining the pattern.

Pattern P_1 is preferred to P_2 with respect to *selectivity* (denoted as $P_1 \succ_{\Sigma} P_2$), if $S(P_1) \supseteq S(P_2)$.

It should be noted that the two relationships discussed earlier are opposed to each other, that is, $Lit(P_1) \subseteq Lit(P_2) \Leftrightarrow S(P_1) \supseteq S(P_2)$.

The maximum pattern in relation to selectivity is a *minterm*, that is, a pattern that covers a single positive observation. The use of this relationship by itself is naturally ineffective because minterms do not have any generalizing power. However, the selectivity relationship is extremely useful in conjunction with other relationships, as will be discussed later.

Another useful relation based on the coverage Cov(P) of pattern P is the set of positive observations of the training sample $X \in K^+$, satisfying the conditions of the pattern P(X) = 1. There is no doubt that patterns with larger coverage have higher generalizability. The observations of the training sample covered by the pattern are evidence that this pattern is applicable in decision making.

However, the following points should be noted: Although the relation $|Cov(P_1)| > |Cov(P_2)|$ can be interpreted as meaning that pattern P_1 is more representative than P_2 , it considers only the number of elements in the two sets $Cov(P_1)$ and $Cov(P_2)$. However, replacing the mentioned comparison of the number of elements in these two sets with a stronger relation, which considers the elements of these sets, makes it possible to consider the individual observations covered by these two patterns. The observations in Cov(P) can be considered as a "body of evidence" confirming pattern *P*.

Pattern P_1 is preferred to P_2 with respect to *evidence* (denoted as $P_1 \succ_{e} P_2$) if it is $Cov(P_1) \supseteq Cov(P_2)$. Those patterns that are maximal in the relation of evidence are called strong; that is, pattern P is strong if there is no pattern P' such that $Cov(P') \supset Cov(P)$.

2.3. *Example* 3

The following strong pattern Equation (3) can be identified in the binary dataset presented in Table 1.

$$x_3, x_3x_4, x_1x_3, x_1x_3x_4, x_2x_4, x_1x_2x_4, x_2x_5, \tag{3}$$

You can see that, for example, pattern x_1x_5 is not strong, because $Cov(x_2x_5) = \{b, c, d\} \supset \{c, d\} = Cov(x_1x_5)$.

It is important to note that the relationships in question are not completely independent. Thus, the relations of simplicity and selectivity are opposite. Moreover, we note the following dependencies in Equations (4) and (5):

$$P_1 \succ_{\sigma} P_2 \Rightarrow P_1 \succ_{\varepsilon} P_2, \tag{4}$$

$$P_1 \succ P_2 \Rightarrow P_2 \succ P_1, \tag{5}$$

As each of the presented relations expresses different aspects of pattern preference, it appears reasonable to use different combinations, as noted in [24].

The new relations that can be obtained by applying their combinations (intersection and lexicographic refinement) are as follows:

The patterns that are maximal in their intersections $\Sigma \wedge \varepsilon$ are called *spanned* patterns. The patterns that are maximal in lexicographic refinement $\varepsilon | \sigma$ are called *strong prime* patterns. The patterns that are maximal in lexicographic refinement are called *strongly spanned* patterns.

Among all the types of patterns obtained in accordance with the relations and their combinations considered earlier, the most useful for identifying informative patterns and using them to support decision making in recognition appears to be the following: prime, strong prime, and strongly spanned patterns.

Table 2 shows some examples highlighting the existence of patterns with different combinations of the properties described above.

	Pattern Properties		- Examples of Pattorns
Strong	Prime	Spanned	- Examples of Fatterns
-	+	-	$\overline{x}_1\overline{x}_4, \overline{x}_4x_5, x_1x_5, x_1\overline{x}_2$
+	+	-	$x_3, x_2 x_4$
+	-	+	$x_1 x_3 x_4, x_1 x_2 x_4$
+	+	+	$x_2 x_4 x_5$

Table 2. Patterns for binary data, presented in Table 1.

3. Searching for Maximum Strong Patterns

Patterns are the building blocks for the formation of the recognition solver function. In most situations, except for the simplest cases, a single pattern is not sufficient to construct a solver function [27,31,32], and a set of different patterns $\prod^{k} = \{P_{1}^{k}, \ldots, P_{qk}^{k}\}$, which together cover all or almost all the training observations of some class *k* (approximate the class domain), is required. Finding a set of patterns is a key problem in LAD.

Different approaches can be used to find a set of patterns, particularly enumeration algorithms, which implement a pattern search as an optimization problem. The original version of LAD [12] used an enumeration algorithm to search for prime patterns. In [23], an enumeration algorithm was proposed to find spanned patterns. These algorithms are time-consuming, especially when processing large volumes of data; thus, their practical use is limited.

In [24], the algorithms for transforming a random pattern into a pattern with certain properties (prime, spanned, and strong) were given. However, using them to convert an arbitrary pattern into a prime or a strong pattern does not lead to a pattern with maximum coverage.

In [30], the optimization problem aimed at searching for patterns with maximum coverage of training observations of some classes was considered, provided that the coverage of the observations of other classes was unacceptable. A set of patterns requires diversity to cover all the training observations of some classes. The diversity of the resulting patterns in this approach is achieved by relying on the feature values of the specific objects.

Consider the observation $a \in K^+$. The regularity P^a covers observation a. Those variables that are fixed in P^a are equal to the corresponding values of the object features a [13].

Based on [30], we consider an a-pattern as a pattern covering observation a. A maximum a-pattern is an a-pattern P with maximum coverage, that is, with the maximum number of positive observations covered by P (if a is positive) or with a maximum number of negative observations covered by P (if a is negative).

Consider the problem of finding a maximum regularity P^a , that is, a term that, in addition to observation *a*, covers as many positive observations as possible without negative ones.

To define regularity P^a , the binary variables $Y = (y_1, y_2, ..., y_n)$ are introduced Equation (6):

$$y_j = \begin{cases} 1, i-\text{th attribute is fixed in } P^a, \\ 0, \text{ if not.} \end{cases}$$
(6)

Some points $b \in K^+$ will be covered by regularity P^a only if $y_i = 0$ for all i, for which $b_i \neq a_i$. In contrast, some points $c \in K^-$ will not be covered by regularity $P^a y_i = 1$ for at least one variable i, for which $c_i \neq a_i$.

Thus, the problem of finding a maximum pattern can be described as the problem of finding such values $Y = (y_1, y_2, ..., y_n)$, in which the resulting regularity P^a covers as many points as possible $b \in K^+$ and does not cover a single point $c \in K^-$ Equations (7) and (8) [30]:

$$\sum_{b \in K^{+}} \prod_{\substack{i = 1 \\ b_i \neq a_i}}^{n} (1 - y_i) \to \max,$$

$$\sum_{\substack{i = 1 \\ c_i \neq a_i}}^{n} y_i \ge 1 \text{ for all } c \in K^{-}.$$
(8)

This problem is a conditional pseudo-Boolean optimization problem, that is, the problem in which the target function and the left parts of the constraints are pseudo-Boolean functions that are the real functions of the Boolean variables. The target and constraint functions in this problem are unimodal and monotonic, respectively.

To search for the maximum negative regularities, the problem is formulated in a similar manner.

It is important to note that any point in $Y = (y_1, y_2, ..., y_n)$ corresponds to a subcube in the Boolean feature space.

 $X = (x_1, x_2, ..., x_n)$, which includes basic observations. At $Y \in O_k(Y^1)$ (i.e., Y differs from Y^1 3 by the value of k coordinates), where $Y^1 = (1, 1, ..., 1)$, the number of points of this subcube is equal to 2^k .

Objective Equation (1) is nonlinear. Bonates, Hammer, and Kogan [30] considered reducing problems Equations (1) and (2) to integer linear programming (ILP) problems. However, as a result, the dimensionality of the problem greatly increases, and they refuse the practical application of this approach and resort to heuristic algorithms, particularly the greedy algorithm, which allows for finding an approximate solution to the problem.

3.1. Greedy Algorithm 1: Increasing Patterns to Maximum Prime Patterns

For a given positive pattern *P*, covering *a*, this heuristic converts *P* into a positive, prime pattern by sequentially removing the literals from *P*. At each step, the removal of a literal is considered advantageous if the resulting pattern is "closer" to the set of positive observations not covered by it than to the set of negative observations.

To refine the criterion for choosing the best pattern at each step, the "divergence" between observation *b* and pattern *P* is introduced as the number of *P* literals whose values in *b* are equal to zero (these conditions are not satisfied for a given observation). We denote the divergence between a positive pattern *P* and the set of positive observations not covered by it by $d^+(P)$. Similarly, $d^-(P)$ denotes the divergence between *P* and a set of negative observations. The computational experiments carried out in [30] showed that the ratio $(d^+(P))/(d^-(P))$ is a good criterion for choosing a deletable literal at each step.

This heuristic makes it possible to find the prime pattern for the underlying observation *a*. However, it should be noted that an approximate solution to the problem using a greedy algorithm does not guarantee a strong pattern.

3.2. Greedy Algorithm 2: Increasing Patterns to Maximum Strong Patterns

This heuristic extends the current positive pattern P, covering a, by choosing the next observation to be included in Cov(P), i.e., in the set of positive observations covered by P. For a non-empty subset S of positive observations, denote by [S] the convex hull of S, i.e., the smallest subcube containing S. The heuristic chooses a positive observation b, not yet

covered by *P*, such that " $[Cov(P) \cup \{b\}]$ ", is a positive pattern with the maximum number of literals.

The considered problems with Equations (1) and (2) aim to generate the pure patterns for which constraint Equation (2) on the non-coverage of observations of the opposite class is strictly satisfied. However, this leads to overtraining in certain tasks. In such cases, constraint Equation (2) can be weakened [20], resulting in partial (non-uniform) patterns. This increases the generalizability of individual patterns and reduces the effects of overtraining. The following approach is applicable to both pure (homogeneous) and partial (heterogeneous) patterns.

4. Decision Making on a Set of Patterns

Suppose that several positive $\Pi^+ = \{P_1^+, \dots, P_{q_+}^+\}$ and negative $\Pi^- = \{P_1^-, \dots, P_{q_-}^-\}$ patterns are found. According to the logical analysis of the data, the following rule is used to determine whether a recognizable observation belongs to one of the classes (Figure 1):

- If an observation is only covered by positive patterns, it is considered positive;
- If an observation is only covered by negative patterns, it is considered negative;
- If an observation is subject to the condition t of the patterns of one class and f of the other, then the class of observation is determined by voting, for example, as the result of the difference t/T f/F, where T and F are the numbers of patterns of these classes;
- If an observation is not covered by any pattern, it is considered to be unrecognized.



Figure 1. Explanatory diagram for a decision rule.

Thus, the entire feature space is divided into the following areas: unambiguous areas, which are covered by patterns of only one class; a conflict area, where points are covered by patterns of different classes (in this case, class membership is determined by pattern voting); and an area not covered by any pattern (observations of this area cannot be recognized).

The LAD methodology, as described above, makes it possible to identify the patterns of different types. The use of different patterns has several significant features. The most influential factor seems to be the opposition between the prime and spanned patterns. The influence of the pattern type on the recognition results is summarized in Table 3.

Table 3. Influence of pattern types on recognition.

-	Prime Patterns	Spanned Patterns
Recognition result	Fewer unrecognized observations	Lower recognition error
Interpretability	Shorter rules	Greater confidence in the recognition result

Prime patterns are simpler and consist of fewer conditions than other patterns. The use of prime patterns reduces the number of unrecognized observations. The use of spanned patterns produces classifiers with better generalizability.

This study proposes an approach based on the joint use of two types of patterns, namely, the construction and use of patterns in pairs: spanned and prime. This makes it

possible to combine the advantages of these two types of patterns. Certainly, a strongly spanned pattern and a strong prime pattern are preferable.

If a strong prime pattern and its corresponding strongly spanned pattern (which differs from the prime pattern in the presence of additional literal) are taken, the following expressions can be written with respect to them Equations (9)–(11):

$$Cov(P^{spanned}) = Cov(P^{prime}), \tag{9}$$

$$S(P^{spanned}) \subseteq S(P^{prime}), \tag{10}$$

$$Lit(P^{prime}) \subseteq Lit(P^{spanned}).$$
 (11)

where *P*^{spanned} is a strongly spanned pattern, and *P*^{prime} is a strong prime pattern.

In Figure 2, some examples of pattern pairs are shown.



Figure 2. Examples of pattern pairs.

Thus, the spanned patterns are more reliable. The prime patterns are simpler and involve more observations. This increases the interpretability of recognition and makes decision making more reasonable (Figure 3).



Figure 3. Diagram explaining decision making when using pattern pairs.

Decoding areas:

- 1—Coverage by spanned patterns of the same class;
- 2—Coverage only by prime patterns of the same class;
- 3—Coverage by spanned patterns of different classes;
- 4—Coverage only by prime patterns of different classes;
- 5—Coverage by spanned patterns of one class and prime patterns of another class;
- 6—No pattern coverage.

The proposed approach makes it possible to assess the level of reliability of the recognition result in more detail through a refined decision-making scheme using the information on the number and type of patterns covering the observation.

When using only one type of pattern (or without considering the type of pattern) to make a classification decision for an observation, the situation could be assigned to one of the following levels, in descending order of confidence in the recognition result:

- Level 1: implementing patterns of the same class;
- Level 2: patterns of the two classes → voting;
- Level 3: no satisfying patterns.

The number of these levels increases when two types of patterns, prime (PP) and spanned (SP), are used.

- Level 1: spanned patterns (SP) of the same class;
- Level 2: only prime patterns (PP) of one class;
- Level 3: SP of one class and only PP of another class;
- Level 4: SP of two classes \rightarrow voting;
- Level 5: only PP (of two classes) \rightarrow voting;
- Level 6: no satisfying patterns.

When making a decision based on the patterns of one type (prime or spanned), there are four possible options (Table 4).

Table 4. Classification by patterns of one type.

Implomentin	a the Pulse	"—" Rules			
Implementing the Kules		Positive	Negative		
"+" rules	positive negative	voting "—" observed	"+" observed unrecognized		

When making a decision based on two types of patterns (prime and spanned), the number of possible choices increases (Table 5).

Table 5. Two types of pattern classification.

Implementi	ng tha Dulas	"—" Rules					
Implementi	lig the Kules	Spanned Prime Neg		Negative			
	spanned	voting	+	+			
"+" rules	prime	—	voting	+			
	negative	—	_	unrecognized			

Consider the sets of pattern pairs consisting of a prime pattern and its corresponding spanned pattern (with additional literal ($Lit(P^{prime}) \subseteq Lit(P^{spanned})$) and the same coverage ($Cov(P^{spanned}) = Cov(P^{prime})$).

Despite the fact that the coverage (on training observations) of these two patterns in each pair is the same, the subcube obtained by the prime pattern can be more extensive $(S(P^{spanned}) \subseteq S(P^{prime}))$. Consequently, the area obtained by combining the prime patterns of the same class includes (and may be wider than) the area formed by combining corresponding covering patterns.

Consider a classification based only on prime patterns. In the set of control observations, we select a subset of observations that are covered by the patterns of both classes. Voting was used to confirm these observations. The experiments showed that most recognition errors occurred in these observations. Now, we consider the classification of this subset of observations by the patterns of the two types. The considered subset of observations in this case can be divided into four groups according to the combinations of the pattern types that cover them:

Spanned (and prime) positive patterns and only prime (without spanned) negative patterns;

- Spanned (and prime) negative patterns and only prime (without spanned) positive patterns;
- Spanned (and prime) patterns of both classes;
- Prime (without spanned) patterns of both classes.

In view of the fact that spanned patterns are, by definition, more "selective" than prime patterns, the observations from the first group should be classified as positive and the observations from the second group as negative, thus reducing the uncertainty that exists when using prime patterns alone (compare the tables in Figure 4).

Implementing the rules			"-" rules				
		positive		negative			
"+" rules	positive	voting			"+" observed		
	negative	"_	"observed		unrecognized		
			↓				
Imple	Implementing the rules		"—" rules				
			spanned	P	orime	negative	
"+" rules	spanne	d	voting	+	-	+	
	prime		-	V	oting	+	
	negativ	e	-	-		unrecognized	

Figure 4. Use of two types of patterns compared with using prime patterns only.

Thus, the use of two types of patterns leads to a higher recognition accuracy than the use of only prime patterns.

Now, consider classification by covering only the patterns. In the set of control observations, we identify a subset of observations that are not covered by any pattern. These observations remain unclear. To this subset of observations, apply the prime patterns, which have more coverage (corresponding to larger subcubes), as shown above. In this case, some of the observations not covered by the spanned patterns are covered by the prime patterns. The considered subset of observations in this case can be divided into four groups according to the combinations of the classes of regularities that cover them:

- Only positive (prime) patterns;
- Only negative (prime) patterns;
- Positive and negative (prime) patterns;
- No coverage.

The observations from the first two groups were clearly recognized as positive and negative observations, respectively. Voting should be applied to the observations of the third group. Only the observations of the fourth group remain unrecognized (see the tables in Figure 5).

Implementing the rules			"-" rules				
F		positive			negative		
"+" rules	positive	voting			"+" observed		
	negative	"_"	"observed		unrecognized		
Imple	ementing th rules	ie	ł		"—" rule	25	
			spanned prime no		negative		
"+" rules	spanne	d	voting		F	+	
	prime		-	1	voting	+	
	negativ	e	-	-	-	unrecognized	

Figure 5. Use of two types of patterns compared with using spanned patterns only.

Thus, the use of two types of patterns results in fewer unrecognized observations than the use of spanned patterns only.

5. Algorithms for Finding a Pair of Patterns

The following algorithm is proposed to find a pair of patterns: First, a prime pattern is identified by solving the optimization problem using a greedy algorithm to find the prime pattern. The prime pattern is then uniquely converted into the corresponding spanned pattern.

Algorithm for Finding a Pair of Patterns

- Find the prime pattern *P*_{prime} by solving problems Equations (1) and (2);
- Determine the set of observations S that are covered *P*_{prime}: *S* = *Cov*(*P*_{prime});
- Find the corresponding P_{prime} spanned pattern $P_{spanned} = \prod_{i \in I} x_i^{\alpha_i}$ (convex hull of set *S*), where *i* is the set of all indices *i*, for which the *i*-th components of all vectors $X \cap S$ have the same value.

Thus, after executing this algorithm for some observations a, the output has a prime pattern P_{prime} as a conjunction of literals $t^{prime}(x) = (\bigwedge_{i \in A} x_i) \land (\bigwedge_{j \in B} \overline{x}_j)$ for some subsets $A, B \subseteq \{1, 2, ..., n\}$, and a corresponding spanned pattern $P_{spanned}$ as a conjunction of $t^{spanned}(x) = t^{prime}(x) \land t^{add}(x)$, where $t^{add}(x)$ is a conjunction of additional literals, the presence of which distinguishes the spanned pattern from the prime. The coverage of these patterns is the same as for the training observations. However, owing to different descriptions (additional literals) and volumes, the coverage of the test observations can differ.

The problem with this method is that finding a strong prime pattern requires an exact solution to the optimization problem. An approximate solution will only provide a prime

pattern that may not provide the best coverage. The solution to this problem is to use a better optimization algorithm to find an exact solution.

6. Results and Discussion

The testing was conducted using publicly available data from the UCI repository. The most popular datasets [33,34] were selected for testing in order to be able to compare the results with the results of other studies on LAD [12,30,35]: Wisconsin Breast Cancer (699 observations, 9 categorical attributes) [36,37]; Heart Disease (303 observations, 14 heterogeneous attributes); Australian Credit (690 observations, 15 heterogeneous attributes); Boston Housing (506 observations, 14 heterogeneous attributes); and Congressional Voting (435 observations, 16 categorical attributes).

Comparisons of LAD results with other classification methods, including rule-based classifiers, have already been performed and have shown the competitiveness of LAD with other methods [29,38]. In this study, the effectiveness of the proposed approach was tested.

The binarization of categorical and numerical attributes was carried out in accordance with the traditional method used in LAD [27,39,40]. The search for positive and negative patterns (strongly spanned and prime) was performed using the greedy algorithms described above. Each training observation was used as the base observation for the pattern generation. The test observations were classified by voting for positive and negative patterns. A set of pairs of patterns was generated using the proposed algorithm to search for a pair of patterns. The classification was performed by voting, considering the type of covering patterns, as described in the previous section.

Each dataset was randomly divided into two datasets: training (50%) and testing (50%). The average values for the 20 random partitions are listed in Table 6.

-	Observation	Classification	Breast Cancer	Heart	Credit	Housing	Voting
		True	91.3	75.6	83.7	83.4	95.2
	pos	False	5.8	19.4	11.4	14.2	3.7
		Not recognized	2.9	5.0	4.9	2.4	1.1
		True	97.4	79.9	84.0	84.9	95.8
Strong prime	neg	False	2.4	16.4	12.3	11.9	3.6
		Not recognized	0.2	3.7	3.7	3.2	0.6
		True	95.3	77.9	83.9	84.2	95.4
	all	False	3.6	17.8	11.9	13.0	3.7
		Not recognized	1.1	4.3	4.2	2.8	0.9
		True	91.8	73.4	81.8	82.3	94.4
	pos	False	4.1	18.7	10.4	12.6	3.7
		Not recognized	4.1	7.9	7.8	5.1	1.9
Strongly		True	96.8	78.1	82.7	83.4	96.4
snannad	neg	False	2.1	14.6	11.8	11.1	3.0
spanned	8	Not recognized	1.1	7.3	5.5	5.5	0.6
		True	95.0	75.9	82.4	82.8	95.2
	all	False	2.9	16.5	11.1	11.9	3.4
		Not recognized	2.1	7.6	6.5	5.3	1.4
		True	93.5	75.7	84.4	84.2	95.2
	pos	False	4.6	18.7	10.7	13.4	3.7
		Not recognized	2.9	5.6	4.9	2.4	1.1
Pairs of		True	97.7	81.1	84.5	85.7	96.4
1 dils 01	neg	False	2.1	15.2	11.8	11.1	3.0
patients		Not recognized	0.2	3.7	3.7	3.2	0.6
		True	95.9	78.9	84.5	84.9	95.7
	all	False	3.0	16.8	11.3	12.3	3.4
		Not recognized	1.1	4.3	4.2	2.8	0.9

Table 6. The average values for 20 random partitions.

One of the considered real problems is the problem of diagnosing breast cancer from a sample collected in Wisconsin (Wisconsin Breast Cancer) [34].

The sample contained information on 699 cases. Each case was described by 11 variables: variable 1—identification number; significant variables 2–10 describe the quantitative signs of tissues (sample code number, clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, mitoses), expressed as integers from 1 to 10; variable 11 is the target, indicating the class of neoplasm (benign or malignant). The sample contained 458 observations of the negative class (benign) and 241 observations of the positive class (malignant).

Since data are numeric (integer) values, their binarization is necessary, that is, the transition to new binary attributes. Cut-point-based binarization was used. Based on the original attribute x, new binary attributes x_t can be introduced as follows Equation (12):

$$x_t = \begin{cases} 1, & \text{if } x \ge t, \\ 0, & \text{if } x < t, \end{cases}$$
(12)

where *t* is a cut point.

As a result of the binarization procedure, 72 binary attributes were obtained from 9 initial attributes, on the basis of which the patterns were generated. When visualizing patterns, binary attributes are converted back to numerical values: the presence of the literal " x_t " in the pattern corresponds to the condition of exceeding the cut point t, and the presence of the literal "negation of x_t " in the pattern corresponds to the condition of not exceeding the cut point t.

The application of the proposed approach is aimed at obtaining pairs of patterns, where each pair consists of a prime and a spanned pattern. First, a prime pattern is generated, and then a spanned pattern is built from it. The resulting spanned pattern differs from the prime one by the presence of additional literals, that is, the additional clarifying conditions for this problem. In some cases, the spanned pattern may coincide with the prime.

Some examples of pairs of patterns for the problem under consideration are given below (the given positive patterns have coverage of at least 30%, and the negative ones of at least 50%). The main conditions (corresponding to the prime pattern) are highlighted in bold. The additional conditions (corresponding to the spanned pattern) are written in normal font: [prime conditions] [additional spanned conditions].

[**Clump Thickness > 8**] [Uniformity of Cell Size > 3] \geq malignant;

[Clump Thickness > 6] [Uniformity of Cell Shape > 4] [Bare Nuclei > 2] [Marginal Adhesion > 2] ≥ malignant;

[**Clump Thickness > 8**] [Bare Nuclei > 1] ≥ malignant;

[Marginal Adhesion > 5] [Bare Nuclei > 4] [Uniformity of Cell Shape > 4] \geq malignant; [Uniformity of Cell Size \leq 1] [Normal Nucleoli \leq 2] [Bare Nuclei \leq 1] \geq benign;

[Uniformity of Cell Shape \leq 2] [Bare Nuclei \leq 2] [Uniformity of Cell Size \leq 1] [Marginal Adhesion \leq 1] \geq benign.

The classification of objects was carried out on the basis of the obtained patterns according to the procedure described in Section 4, and the results are shown in Table 6. In the future, it is planned to test the proposed approach on other, more specific machine learning problems, as well as to expand the class of problems to be solved, for example, its application to unsupervised learning problems.

7. Conclusions

Searching for patterns is a key part of LAD. Although a pattern is simply a conjunction of a number of literals, it is possible to identify patterns with certain properties among the entire set of patterns: prime, spanned, and strong. The distinctive feature of LAD is that the pattern generation process can be controlled, and patterns with the desired properties can be obtained. This study proposes an approach that extends the advantages of LAD. A study of the peculiarities of using different types of patterns revealed that the use of prime patterns reduces the number of unrecognized observations, whereas the use of spanned patterns reduces recognition errors. A new approach for decision support in recognition was developed by combining the use of two types of patterns: prime and spanned. This result aims to extend the capabilities of LAD as a promising method for interpretable machine learning. From the point of view of interpretability, paired patterns can be considered not as two different rules, but as two variants of one rule: one version is simpler, and the other is stricter, clarifies, and uses additional conditions.

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

Funding: The studies were carried out within the program of the Russian Federation of strategic academic leadership "Priority-2030" aimed at supporting the development programs of educational institutions of higher education, the scientific project PRIOR/SN/NU/22/SP5/16 "Building intelligent networks, determining their structure and architecture, operation parameters in order to increase productivity systems and bandwidth of data transmission channels using trusted artificial intelligence technologies that provide self-learning, self-adaptation and optimal reconfiguration of intelligent systems for processing large heterogeneous data".

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

- Dhall, D.; Kaur, R.; Juneja, M. Machine Learning: A Review of the Algorithms and Its Applications. *Lect. Notes Electr. Eng.* 2020, 597, 47–63. [CrossRef]
- 2. Wolberg, W. UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set. Available online: https://archive. ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original) (accessed on 27 June 2022).
- 3. Udousoro, I.C. Machine Learning: A Review. Semicond. Sci. Inf. Devices 2020, 2, 5–14. [CrossRef]
- 4. Ledesma, S.; Ibarra-Manzano, M.A.; Cabal-Yepez, E.; Almanza-Ojeda, D.L.; Avina-Cervantes, J.G. Analysis of Data Sets with Learning Conflicts for Machine Learning. *IEEE Access* 2018, *6*, 45062–45070. [CrossRef]
- Halbouni, A.; Gunawan, T.S.; Habaebi, M.H.; Halbouni, M.; Kartiwi, M.; Ahmad, R. Machine Learning and Deep Learning Approaches for CyberSecurity: A Review. *IEEE Access* 2022, 10, 19572–19585. [CrossRef]
- Letham, B.; Rudin, C.; McCormick, T.H.; Madigan, D. Interpretable Classifiers Using Rules and Bayesian Analysis: Building a Better Stroke Prediction Model. Ann. Appl. Stat. 2015, 9, 1350–1371. [CrossRef]
- Prentzas, N.; Nicolaides, A.; Kyriacou, E.; Kakas, A.; Pattichis, C. Integrating Machine Learning with Symbolic Reasoning to Build an Explainable Ai Model for Stroke Prediction. In Proceedings of the 2019 IEEE 19th International Conference on Bioinformatics and Bioengineering, BIBE 2019, Athens, Greece, 28–30 October 2019; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA; pp. 817–821.
- Rudin, C.; Shaposhnik, Y. Globally-Consistent Rule-Based Summary-Explanations for Machine Learning Models: Application to Credit-Risk Evaluation. SSRN Electron. J. 2019, 4, 1–19. [CrossRef]
- Crama, Y.; Hammer, P.L.; Ibaraki, T. Cause-Effect Relationships and Partially Defined Boolean Functions. Ann. Oper. Res. 1988, 16, 299–325. [CrossRef]
- 10. Niu, D.; Liu, L.; Lu, S. Augmenting Negation Normal Form with Irrelevant Variables. IEEE Access 2019, 7, 91360–91366. [CrossRef]
- 11. Muselli, M.; Ferrari, E. Coupling Logical Analysis of Data and Shadow Clustering for Partially Defined Positive Boolean Function Reconstruction. *IEEE Trans. Knowl. Data Eng.* 2011, 23, 37–50. [CrossRef]
- 12. Boros, E.; Hammer, P.L.; Ibaraki, T.; Kogan, A.; Mayoraz, E.; Muchnik, I. An Implementation of Logical Analysis of Data. *IEEE Trans. Knowl. Data Eng.* 2000, *12*, 292–306. [CrossRef]

- Alexe, G.; Alexe, S.; Bonates, T.O.; Kogan, A. Logical Analysis of Data—The Vision of Peter L. Hammer. Ann. Math. Artif. Intell. 2007, 49, 265–312. [CrossRef]
- Chikalov, I.; Lozin, V.; Lozina, I.; Moshkov, M.; Nguyen, H.S.; Skowron, A.; Zielosko, B. Three Approaches to Data Analysis: Test Theory, Rough Sets and Logical Analysis of Data, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 41, ISBN 9783642286667.
- 15. Lancia, G.; Serafini, P. Computational Complexity and Ilp Models for Pattern Problems in the Logical Analysis of Data. *Algorithms* **2021**, *14*, 235. [CrossRef]
- 16. Elfar, O.; Yacout, S.; Osman, H. Accelerating Logical Analysis of Data Using an Ensemble-Based Technique. *Eng. Lett.* **2021**, *29*, 1616–1625.
- Zhou, B.; Shang, L.; Song, X.; Wang, J.; Xu, J. Logical Causal Model of Power System Fault Alarm and Its Application. In Proceedings of the 2020 IEEE/IAS Industrial and Commercial Power System Asia, I and CPS Asia 2020, Weihai, China, 13–15 July 2020; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA; pp. 964–969.
- Yan, K.; Miao, D.; Guo, C.; Huang, C. Efficient Feature Selection for Logical Analysis of Large-Scale Multi-Class Datasets. J. Comb. Optim. 2021, 42, 1–23. [CrossRef]
- Bertolazzi, P.; Felici, G.; Festa, P.; Lancia, G. Logic Classification and Feature Selection for Biomedical Data. *Comput. Math. Appl.* 2008, 55, 889–899. [CrossRef]
- Han, J.; Kim, N.; Yum, B.J.; Jeong, M.K. Pattern Selection Approaches for the Logical Analysis of Data Considering the Outliers and the Coverage of a Pattern. *Expert Syst. Appl.* 2011, 38, 13857–13862. [CrossRef]
- Subasi, M.M.; Ávila, J.F. A New Approach to Select Significant Patterns in Logical Analysis of Data. *Rutcor Res. Rep.* 2012, 1–20. [CrossRef]
- 22. Kuzmich, R.; Stupina, A.; Korpacheva, L.; Ezhemanskaja, S.; Rouiga, I. The Modified Method of Logical Analysis Used for Solving Classification Problems. *Informatica* 2018, 29, 467–486. [CrossRef]
- 23. Alexe, G.; Hammer, P.L. Spanned Patterns for the Logical Analysis of Data. Discret. Appl. Math. 2006, 154, 1039–1049. [CrossRef]
- 24. Hammer, P.L.; Kogan, A.; Simeone, B.; Szedmák, S. Pareto-Optimal Patterns in Logical Analysis of Data. *Discret. Appl. Math.* 2004, 144, 79–102. [CrossRef]
- Guo, C.; Ryoo, H.S. On Pareto-Optimal Boolean Logical Patterns for Numerical Data. *Appl. Math. Comput.* 2021, 403, 126153. [CrossRef]
- 26. Boros, E.; Crama, Y.; Hammer, P.L.; Ibaraki, T.; Kogan, A.; Makino, K. Logical Analysis of Data: Classification with Justification. *Ann. Oper. Res.* **2011**, *188*, 33–61. [CrossRef]
- 27. Hammer, P.L.; Bonates, T.O. Logical Analysis of Data-An Overview: From Combinatorial Optimization to Medical Applications. *Ann. Oper. Res.* 2006, 148, 203–225. [CrossRef]
- 28. Lejeune, M.; Lozin, V.; Lozina, I.; Ragab, A.; Yacout, S. Recent Advances in the Theory and Practice of Logical Analysis of Data. *Eur. J. Oper. Res.* **2019**, 275, 1–15. [CrossRef]
- 29. Alexe, G.; Alexe, S.; Hammer, P.L.; Kogan, A. Comprehensive vs. Comprehensible Classifiers in Logical Analysis of Data. *Discret. Appl. Math.* **2008**, *156*, 870–882. [CrossRef]
- 30. Bonates, T.O.; Hammer, P.L.; Kogan, A. Maximum Patterns in Datasets. Discret. Appl. Math. 2008, 156, 846–861. [CrossRef]
- An, A.; Cercone, N. Rule Quality Measures for Rule Induction Systems: Description and Evaluation. *Comput. Intell.* 2001, 17, 409–424. [CrossRef]
- 32. Chou, C.A.; Bonates, T.O.; Lee, C.; Chaovalitwongse, W.A. Multi-Pattern Generation Framework for Logical Analysis of Data. Ann. Oper. Res. 2017, 249, 329–349. [CrossRef]
- Bain, T.C.; Avila-Herrera, J.F.; Subasi, E.; Subasi, M.M. Logical Analysis of Multiclass Data with Relaxed Patterns. Ann. Oper. Res. 2020, 287, 11–35. [CrossRef]
- Dua, D.; Graff, C. UCI Machine Learning Repository. Available online: https://archive.ics.uci.edu/ml/index.php (accessed on 10 June 2022).
- 35. Xin, B.; Chen, L.; Chen, J.; Ishibuchi, H.; Hirota, K.; Liu, B. Interactive Multiobjective Optimization: A Review of the State-of-the-Art. *IEEE Access* 2018, *6*, 41256–41279. [CrossRef]
- 36. Emmanuel Gbenga, D.; Christopher, N.; Comfort Yetunde, D.; Author, C. Performance Comparison of Machine Learning Techniques for Breast Cancer Detection. *Nova* 2017, *6*, 1–8. [CrossRef]
- Rui, S. Breast Cancer Wisconsin (Original) Data Set (Analysis with Statsframe ULTRA). Available online: https://www. researchgate.net/publication/337304299_Breast_Cancer_Wisconsin_Original_Data_Set_analysis_with_Statsframe_ULTRA (accessed on 10 June 2022).
- Elfar, O.; Montréal, P.; Yacout, S.; Osman, H. Merging Logical Analysis of Data Models. In Proceedings of the International Conference on Industrial Engineering and Operations Management, Toronto, Canada, 23 October 2019; International Journal of Industrial Engineering and Operations Management Inc.; Emerald Publishing: Bingley, UK; pp. 183–194.
- Anthony, M.; Ratsaby, J. Robust Cutpoints in the Logical Analysis of Numerical Data. *Discret. Appl. Math.* 2012, 160, 355–364.
 [CrossRef]
- 40. Boros, E.; Hammer, P.L.; Ibaraki, T.; Kogan, A. Logical Analysis of Numerical Data. Math. Program. 1997, 79, 163–190. [CrossRef]