

Notes on Relation Between Symbolic Classifiers

Xenia Naidenova¹, Alexey Buzmakov², Vladimir Parkhomenko³, and
Alexander Schukin³

¹ Military Medical Academy, Saint-Petersburg, Russia
ksennaidd@gmail.com

² NRU Higher School of Economics, Perm, Russia
avbuzmakov@hse.ru

³ Peter the Great St. Petersburg Polytechnic University, Saint-Petersburg, Russia
parhomenko.v@gmail.com, alexander.schukin@spbstu.ru

Abstract. Symbolic classifiers allow for solving classification task and provide the reason for the classifier decision. Such classifiers were studied by a large number of researchers and known under a number of names including tests, JSM-hypotheses, version spaces, emerging patterns, proper predictors of a target class, representative sets etc. Here we consider such classifiers with restriction on counter-examples and discuss them in terms of pattern structures. We show how such classifiers are related. In particular, we discuss the equivalence between good maximally redundant tests and minimal JSM-hypotheses and between minimal representations of version spaces and good irredundant tests.

Keywords: machine learning · symbolic classifier · version spaces · JSM-method · minimal JSM-hypotheses · test theory · irredundant test · good test · representative sets · jumping emerging patterns · formal concept analysis · pattern structure · semiconcepts

1 Introduction

Symbolic classifiers are being developed more and more extensively nowadays. One of the main advantages of such classifiers is their ability to provide logical rules that can be interpreted by experts. In this paper, we consider classifiers with a restriction on counter-examples, which is widely spread. The reason of its popularity is the opportunity to control the entrance of an attribute to the premise of classification rules, in particular, if a counter-example is forbidden, some combinations of its attributes are guaranteed not to be in the premise.

These classifiers were studied by a large number of researchers and known under a number of names. Accordingly, tests [7,44,41,29], JSM-hypotheses [13], version spaces [25], emerging patterns [42,6], representative sets [43,45] (first mentioned in [3] as representative samples), proper predictors of a target class [16] are considered in this paper. Establishing links between these classifiers would allow for mutually enriching the approaches. The main motivation of the paper is to provide an appropriate framework on the basis of pattern structures.

The rest of the paper is organized as follows. Sec. 2 presents the basic definitions of Formal Concept Analysis and pattern structures to obtain a language for symbolic classifiers comparison. Sec. 3 is devoted to the interpretation of diagnostic tests, version spaces and good maximally redundant tests in the language of pattern structures. Finally, the relation between good tests, minimal JSM-hypotheses and version spaces is discussed in Sec. 4.

2 Main definitions of FCA and pattern structures

Let us consider the main ideas of FCA, a deeper review of FCA and its applications can be found in [39,37,38]. Let $(2^G, \subseteq)$ and $(2^M, \subseteq)$ be posets, where G and M are called the set of objects and the set of attributes, respectively, \subseteq is the inclusion relation. The binary relation $I \subseteq G \times M$ provides links between G and M . The triple $\mathbb{K} \stackrel{\text{def}}{=} (G, M, I)$ is called a formal context. In Formal Concept Analysis (FCA), Galois connection is a pair of derivation operators between $(2^G, \subseteq)$ and $(2^M, \subseteq)$ usually denoted with one symbol $(\cdot)'$ [15,40]. In this paper for clarity, they are denoted with two different symbols $(\cdot)^\uparrow$ and $(\cdot)^\downarrow$, since they represent different operators.

$$A^\uparrow := \{m \in M \mid (\forall g \in A)[gIm]\}; \quad (1)$$

$$B^\downarrow := \{g \in G \mid (\forall m \in B)[gIm]\}. \quad (2)$$

A pair (A, B) such that $A \subseteq G, B \subseteq M$ and $(A = B^\downarrow) \& (A^\uparrow = B)$ [40] is called a formal concept, left and right parts of which are called the concept extent and the concept intent, respectively. The seminal paper on FCA is written by R. Wille 1982 and reprinted in 2009 [40]. Some similar results were obtained in [2] however did not get much attention. Maximal rectangles in Boolean matrices [35] can also be considered as formal concepts. Retrospective analysis of more earlier papers from USSR and other countries can be seen in [19,11,33].

The superpositions of operators $(\cdot)^{\uparrow\downarrow}$ and $(\cdot)^{\downarrow\uparrow}$ form closure operators. Formulas (3–8), where $A_2, A_1, A \subseteq G$ and $B_2, B_1, B \subseteq M$, reflect the properties of Galois connection [40].

$$\text{If } A \subseteq A_1, \text{ then } A_1^\uparrow \subseteq A^\uparrow; \quad (3)$$

$$\text{if } B \subseteq B_1, \text{ then } B_1^\downarrow \subseteq B^\downarrow; \quad (4)$$

$$A \subseteq A^{\uparrow\downarrow}; \quad (5)$$

$$B \subseteq B^{\downarrow\uparrow}; \quad (6)$$

$$A^\uparrow = A^{\uparrow\downarrow\uparrow}; \quad (7)$$

$$B^\downarrow = B^{\downarrow\uparrow\downarrow}. \quad (8)$$

Let us note in Proposition 1, that the antecedents of equations 3 and 4 can satisfy \subset instead of \subseteq .

Proposition 1. *If $A_1, A \subseteq G$ and $B_1, B \subseteq M$, then*

$$\text{if } A \subset A_1, \text{ then } A_1^\uparrow \subseteq A^\uparrow; \quad (9)$$

$$\text{if } B \subset B_1, \text{ then } B_1^\downarrow \subseteq B^\downarrow. \quad (10)$$

Proof is the same as the similar proof for \subseteq case in antecedent made by R. Wille and B. Ganter in 1999. Later we will rely on the following Corollaries.

Corollary 1. *Let $A_1 \subseteq G$ and $A \subset A_1$, then*

1. $A_1^\uparrow \subset A^\uparrow$, if $\exists m [(m \in A^\uparrow) \wedge (m \notin (A_1 \setminus A)^\uparrow)]$;
2. $A_1^\uparrow = A^\uparrow$, if $\forall m [(m \in A^\uparrow) \rightarrow (m \in (A_1 \setminus A)^\uparrow)]$. □

Corollary 2. *Let $B_1 \subseteq M$ and $B \subset B_1$, then*

1. $B_1^\downarrow \subset B^\downarrow$, if $\exists g [(g \in B^\downarrow) \wedge ((B_1 \setminus B) \not\subseteq g^\uparrow)]$;
2. $B_1^\downarrow = B^\downarrow$, if $\forall g [(g \in B^\downarrow) \rightarrow (g \in (B_1 \setminus B)^\downarrow)]$. □

Formal concepts are partially ordered by inclusion of extents or intents: for $A_1, A \subseteq G$, and $B_1, B \subseteq M$ the order is given by

$$(A_1, B_1) \leq (A, B) :\Leftrightarrow A_1 \subseteq A \text{ (or dually } \Leftrightarrow B \subseteq B_1). \quad (11)$$

The set of ordered formal concepts forms the formal concept lattice $\mathfrak{B}(G, M, I)$ (Galois lattice) of an input formal context.

Although FCA deals with binary contexts, many other types of data can be processed. In particular, a many-valued context \mathbb{M} is considered in FCA. $\mathbb{M} \stackrel{\text{def}}{=} (G, M, W, J)$, where G, M, W are the set of object names, attributes and attribute values, J is the ternary relation $J \subseteq G \times M \times W$, which gives the value $w \in W$ of the attribute $m \in M$ to the object $g \in G$ (notation $(g, m, w) \in J$ or $m(g) = w$). The statements $(g, m, w) \in J$ and $(g, m, v) \in J$ lead to $w = v$. \mathbb{M} can be represented with a table, where columns and rows corresponds to M and G , respectively.

The procedure of transformation of many-valued contexts to binary contexts is called conceptual scaling [15]. The disadvantage of scaling is the essential growth of the number of attributes. Additionally, since scaled attributes are considered independently, a lattice building procedure cannot be implemented in the most efficient way.

There are related papers that avoid these disadvantages [5,31,17,12,14,34,8,4]. In particular, [17] introduced so called pattern structures (PSs) $\mathcal{P} \stackrel{\text{def}}{=} (G, (D, \sqcap), \delta)$, where G and D are the sets of objects and the set of descriptions, $\delta : G \rightarrow D$ is the mapping giving the correspondence between objects and their descriptions, (D, \sqcap) forms a semilattice. $\delta(G) \stackrel{\text{def}}{=} \{\delta(g) \mid g \in G\}$ produces the complete subsemilattice (D_δ, \sqcap) of (D, \sqcap) [17].

Elements of D are called patterns. For arbitrary patterns $c, d \in D$ the subsumption order \sqsubseteq is given by: $c \sqsubseteq d :\Leftrightarrow c \sqcap d = c$.

Derivation operators of Galois connection between $(2^G, \subseteq)$ and (D, \sqsubseteq) [17] are typically denoted by $(\cdot)^\square$ or $(\cdot)^\diamond$. As earlier, for readability it is replaced by two symbols (12) and (13) in the following way:

$$A^\diamond := \{\sqcap_{g \in A} \delta(g) \mid \forall A \subseteq G\}; \quad (12)$$

$$d^\diamond := \{g \in G \mid (\forall d \in (D, \sqcap)) [d \sqsubseteq \delta(g)]\}. \quad (13)$$

A pair (A, d) , $A \subseteq G$, $d \in (D, \sqcap)$, $(A^\diamond = d) \wedge (d^\diamond = A)$ is a pattern concept. The set of pattern concepts is ordered by extents and intents and forms the lattice $\mathfrak{B}(G, (D, \sqcap), \delta)$. For any PS \mathcal{P} , there can be several formal contexts generating an isomorphic lattice. Every such context is called representative context $\mathcal{R}(\mathcal{P})$. A possible construction procedure is discussed in Theorem 1 from [17]. Note, that the notations of intents and extents, some propositions and their corollaries, e.g. Proposition 1, Corollaries 1 and 2, from FCA are applicable to PSs, and vice versa.

Many kinds of data can be represented by PSs. Let us now show how to rely on PSs in a classification task. Let K be the set of class labels (or possible class descriptions in general). For a subset of objects we know their labels by means of a function $\varepsilon : G \rightarrow K$. Let us call $\alpha : D \rightarrow K$ the classification function or, in some sense, the recognition (forecasting) algorithm: see, please, Figure 1, where the appropriate commutative diagram is depicted. For the sake of simplicity, suppose that K divides G into two disjoint classes G_+ and G_- . If the class of an object is unknown, it can be referred to G_τ .

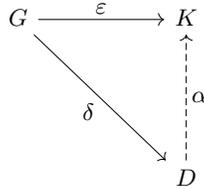


Fig. 1. Commutative diagram of classification function α

Pattern structures $\mathcal{P}_+ \stackrel{\text{def}}{=} (G_+, (D, \sqcap), \delta)$, $\mathcal{P}_- \stackrel{\text{def}}{=} (G_-, (D, \sqcap), \delta)$ and $\mathcal{P}_\tau \stackrel{\text{def}}{=} (G_\tau, (D, \sqcap), \delta)$ are called positive, negative and unknown pattern structures, respectively. Then \mathcal{P}_\pm is a classification structure $\mathcal{P}_+ \cup \mathcal{P}_-$. Let $\mathcal{E} \stackrel{\text{def}}{=} (G, (K, \sqcap), \epsilon)$ be a pattern structure of class descriptions. Let $\mathcal{L} \stackrel{\text{def}}{=} \mathcal{P}_\pm \times \mathcal{E} = (G_\pm, (D, \sqcap) \times (K, \sqcap), \beta(g))$, where $\beta(g) = \langle \delta(g), \epsilon(g) \rangle$, be a learning PS.

3 Interpretation of Diagnostic Tests, Version Spaces and Good Maximally Redundant Tests

Let us discuss application of PSs for dealing with the classification task.

3.1 Diagnostic Tests and I-tests

A notion of a diagnostic test was introduced in 1955 by Chegis and Yablonskii for checking electronic schemas [7]. In their work, the authors suggest an approach that allows for identification of the type of an electronic schema malfunction. In the data, the input and output of the schemas are paired with the type of the malfunction. Accordingly, such data do not allow for counter-examples, since a schema can be either valid or invalid. Moreover, all possible inputs are considered to be present in the data. The diagnostic test is defined as a set of the data rows that allow for identification of the malfunction.

Later, diagnostic tests were used in a meaning being very close to a special kind of functional dependencies (FDs) in the data [10,44,41]. In particular, [44] defines a diagnostic test in the following way (rewritten in terms of many-valued contexts):

Definition 1. *Given a many-valued context (G, M, W, J) , a diagnostic test is a set of attributes $B \subseteq M$ having the following property: the context $(G, B, W, J \cap G \times B \times W)$ does not contain identical rows in distinct classes.*

It should be mentioned that it is not a trivial task to express FDs in terms of PSs. Indeed, FDs subsume that there are a number of “attributes” and only some of them are considered within an FD. However, in PSs there are no attributes in general. What could be an analogue of the attributes for graphs or sequences?

A possible way for expressing FDs in terms of PSs is the following. Given a PS $(G, (D, \sqcap), \delta)$, let D be represented as $D = D_1 \times D_2 \times \dots \times D_N$, where $(D_i, \sqcap_i) \times (D_j, \sqcap_j) = (D_i \times D_j, \sqcap_{i,j})$ and $\langle a, b \rangle \sqcap_{i,j} \langle c, d \rangle = \langle a \sqcap_i c, b \sqcap_j d \rangle$. Accordingly, δ is decomposed as $\delta(g) = \langle \delta_1(g), \delta_2(g), \dots, \delta_N(g) \rangle$, where $\delta_i : G \rightarrow D_i$. Given a subset $\mathcal{D} \subseteq \{D_i\}$ let $\delta_{\mathcal{D}}(g) = \langle \delta_{j_1}(g), \delta_{j_2}(g), \dots \rangle$, where $\forall j_i [D_{j_i} \in \mathcal{D}]$. Then a diagnostic test can be expressed as follows.

Definition 2. *A diagnostic test for the learning context $\mathbb{L} \stackrel{\text{def}}{=} (G_{\pm}, (D, \sqcap) \times (K, \sqcap), \langle \delta(g), \epsilon(g) \rangle)$ is a subset $\mathcal{D} \subseteq \{D_i\}$ such that, if there are $g_1, g_2 \in G_{\pm}$ and $\delta_{\mathcal{D}}(g_1) = \delta_{\mathcal{D}}(g_2)$, then $\epsilon(g_1) = \epsilon(g_2)$.*

Very similar entities to diagnostic tests are premises of implications with the conclusion from the set of classes K . Different kinds of such premises are known as hypotheses [13], representative sets [3], tests [30]¹, jumping emerging patterns [42], and others. In order to distinguish such notions from diagnostic tests let us call them *i-tests*.

Definition 3 (i-test in terms of PSs). *h_{test} is called an i-test for \mathbb{P}_{\pm} iff*

$$h_{\text{test}} \sqsubseteq \delta(g), \exists g \in G_+, h_{\text{test}} \blacklozenge \neq \emptyset; \quad (14)$$

$$h_{\text{test}} \not\sqsubseteq \delta(g), \forall g \in G_-. \quad (15)$$

¹ In this paper, premises with their extents are considered.

The notions of diagnostic tests and i-tests are deeply interconnected. There is an algorithm that allows for reducing the tasks of finding diagnostic tests and i-tests to each other [28]. Moreover natural links between FDs and dependencies in a partition lattice are studied in [27,9,1], and relation between these three notions are shown in [33]. A related notion to i-test is semiconcept introduced by R. Wille.

3.2 Semiconcepts

Semiconcept is a relaxation of a formal concept and was introduced in 1990 by R. Wille [23].

Definition 4 (Object semiconcept, OSC). *A pair (A, A^\diamond) is called an object semiconcept.*

Definition 5 (Attribute semiconcept, ASC). *A pair (d^\diamond, d) is called an attribute semiconcept.*

The left part of ASC is closed and is called the extent of ASC, while the right part is not necessary closed and is called the generator of ASC. This relaxation of concepts allows for expressing different kinds of i-tests (representative sets, jumping emerging patterns, etc.). Similarly, the right part of an OSC is closed and is called the intent, while the left part is not necessary closed and can be called the generator. Semiconcept notation is useful for a description of i-tests inferring algorithms.

I-tests are used for solving the classification task. The classification task itself is well defined in the works of Mitchell under the name of version spaces. Let us consider them in more detail.

3.3 Version Spaces

Version spaces are proposed in 1977 [25] with a detailed description given in 1978 [26] by T. Mitchell. Although T. Mitchell works only with very specific language of classifiers in the report (this language is closed to i-tests and hypotheses for a formal context), the formalization of version spaces is very general. It allows for any language of descriptions of objects and any language of classifiers. Let us recall a particular case of version spaces, where the language describing the classifiers is (D, \sqcap) [18].

Definition 6. *A classifier α is any predicate on D , in particular, $\alpha : (D, \sqcap) \rightarrow K$.*

Defining classifiers by predicates allows for constructing a classifier as any rule that can be applied to unknown objects. Although a classifier can be any predicate, but not all classifiers are consistent with the data in hand.

Definition 7. *A classifier α is called consistent under restriction on counterexample if $\{g \in G_\pm \mid \alpha(\delta(g))\} = G_+$.*

A consistent classifier divides all possible objects, that are called instances, into two groups (or more groups in a multiclass case) of positive objects and negative objects. When the classifier is consistent, it means that all the instances from the dataset (called examples in the thesis of Mitchell) belong to the correct group. There is a number of such classifiers that are consistent with the data.

Definition 8. *A set of consistent classifiers is called a version space VS .*

In terms of examples, i.e. the objects of the dataset, there is the only possible “consistent” classifier. Indeed, there are two class labels. Accordingly there is the only one partition of the set of examples (the objects with known class labels from the dataset) in two groups with the same class label at each group. However, in terms of instances, i.e., all possible objects, the set of classifiers can be huge. These classifiers are naturally ordered by means of set inclusion operations on instances. Correspondingly, if the language of classifiers is (D, \sqsupseteq) , then the order of classifiers is opposite to the order given by \sqsubseteq of descriptions in D .

The version space can be described by the minimal and maximal boundaries of classifiers in terms of the classifier order.

Definition 9. $VS_{\max} = \{p \in VS \mid \nexists p_1 \in VS \text{ such that } p_1 \sqsubseteq p\}$

Definition 10. $VS_{\min} = \{p \in VS \mid \nexists p_1 \in VS \text{ such that } p_1 \supseteq p\}$

For convenience let us call the classifiers that are not from VS_{\min} and VS_{\max} “middle” classifiers.

Definition 11. *A middle VS is given by $VS_{\text{mid}} = VS \setminus VS_{\min} \setminus VS_{\max}$.*

Although version spaces are presented in this paper based on [25,26,18] there are some important details of interpretation of version spaces in [25,26], which were clarified in the further manuscripts by T.Mitchell (see, e.g. [24, p.31]). This interpretation is associated with the following specification of the order \sqsubseteq : in definition of VS_{\min} one shall use this order w.r.t. the extent of VS_{\max} . To clarify the order-based notation Mitchell just gives subscription g to the order \sqsubseteq . To be more careful with the abbreviations used, we will use the notation $\sqsubseteq_{VS_{\max}^{\diamond}}$. The order is discussed in subsection 4.3. Now we shall discuss JSM-hypotheses to show their relations with proper predictors and i-tests.

3.4 JSM-hypotheses

JSM-hypotheses were proposed in 1981 by F.K.Finn [13] and were formulated in 2001 by Ganter & Kuznetsov in terms of PSs.

Minimal JSM-hypotheses are proposed by S.O.Kuznetsov in [20,21]. Later this formalism is reformulated in terms of PSs [18].

Definition 12. *A minimal JSM-hypothesis h_{\min} is a hypothesis (i-test) that $\nexists h_{\text{test}} [(h_{\text{test}} \sqsubseteq h_{\min}) \wedge (h_{\text{test}} = h_{\text{test}}^{\diamond\diamond}) \wedge (h_{\min} = h_{\min}^{\diamond\diamond})]$.*

Another important notion is proper premise, which is closely related to JSM-hypothesis [18]. We give it in terms of PSs.

Definition 13. *Given a PS \mathbb{P} and $d \in D$, $d \neq \emptyset$ is a proper premise of \mathbb{P} , if $d^{\heartsuit} \neq d$ and $\nexists c \sqsubset d$ such that $c^{\heartsuit} = d^{\heartsuit}$.*

Let us denote descriptions of positive and negative objects by D_+ and D_- .

Definition 14. *For a classification structure \mathbb{P}_{\pm} , $d \in D_+$ is a proper premise for a classification pattern (say, $K = \{k_+, k_-\}$), if d is a proper premise and $k_+ \not\sqsubset d$ and $k_+ \sqsubset d^{\heartsuit}$.*

One extra notion is proposed in [18] to show a direct relation between PSs and VS (VS_{\min}).

Definition 15. *For \mathbb{P}_{\pm} , $d \in D_+$ is a proper predictor iff the following conditions are satisfied:*

- $d^{\heartsuit} \cap G_- = \emptyset$,
- $d^{\heartsuit} \cap G_+ \neq \emptyset$,
- if $d_1 \sqsubseteq d$ and $d_1 \neq d$, then $d_1^{\heartsuit} \cap G_- \neq \emptyset$.

3.5 Good Tests

The notion of good tests as FDs was developed in 1985 [28]. Later in 1992, good tests were extended the case of i-tests (the implication case) [30]. This extension is a transformation algorithm for reducing one case to another.

The algorithm reducing search for diagnostic tests (as FDs) to the search for implications is given in Fig. 2 [30]. Similar ideas can also be found in [22,1].

The input of the algorithm is a many-valued context \mathbb{M} and a labelling function ϵ . The output is two new contexts $\overline{\mathbb{K}}_+ \stackrel{\text{def}}{=} (\overline{G}_+, M, I_+)$, $\overline{\mathbb{K}}_- \stackrel{\text{def}}{=} (\overline{G}_-, M, I_-)$ such that i-tests found in $\overline{\mathbb{K}}_+$ are diagnostic tests in \mathbb{M} , and objects from $\overline{\mathbb{K}}_-$ are counter-examples.

In line 10 of this algorithm, the Boolean matrix (the classification context is obtained as $\overline{\mathbb{K}}_{\pm} \stackrel{\text{def}}{=} \overline{\mathbb{K}}_+ \cup \overline{\mathbb{K}}_-$) can be called an indiscernibility matrix [33]. After this step, if it is necessary, only maximal elements of \overline{G}_- can be stored. After running the described algorithm one can infer the initial set of diagnostic tests from the found i-tests.

Usually, the definition of a classifier is given according to the class label, however a classification structure \mathbb{P}_{\pm} is omitted for simplicity [22,32]. If an i-test is referred to positive or negative objects, then the appropriate situation will be denoted by \mathbb{P}_{\pm} or \mathbb{P}_{\mp} , respectively.

Definition 16. *An i-test (h^{\heartsuit}, h) for \mathbb{P}_{\pm} is called an i-test extended with its extents h^{\heartsuit} , or ASC-test.*

In order to define good i-tests we should first introduce irredundant i-tests. The predecessor of this notion was introduced by S. Yablonskiy in 1955 [7] under the name elementary test.

```

Input: the many-valued context  $\mathbb{M} \stackrel{\text{def}}{=} (G, M, W, J)$ , the class membership
 $\epsilon : G \rightarrow K$ 
Output: positive and negative binary contexts  $\overline{\mathbb{K}}_+ \stackrel{\text{def}}{=} (\overline{G}_+, M, I_+)$ ,
 $\overline{\mathbb{K}}_- \stackrel{\text{def}}{=} (\overline{G}_-, M, I_-)$  such that i-tests found in  $\overline{\mathbb{K}}_+$  are diagnostic
tests in  $M$ , and objects from  $\overline{\mathbb{K}}_-$  are counter-examples
1. for  $\forall g_i, g_j \in G$  do
2.   if  $i < j$  then
3.      $\overline{G} \leftarrow (g_i, g_j)$ ;
4. for  $\forall (g_i, g_j) \in \overline{G}$  do
5.   if  $m(g_i) = m(g_j)$  then
6.      $(g_i, g_j)Im$ ;
7.   if  $\epsilon(g_i) = \epsilon(g_j)$  then
8.      $\overline{G}_+ \leftarrow (g_i, g_j)$ ;
9.   else  $\overline{G}_- \leftarrow (g_i, g_j)$ ;
10.  $I_+ = I \cap (\overline{G}_+ \times M)$ ,  $I_- = I \cap (\overline{G}_- \times M)$ ;
11. for  $\forall \overline{g}_+ \in \overline{G}_+$ ,  $\forall \overline{g}_- \in \overline{G}_-$  do
12.   if  $\overline{g}_+^\uparrow \subseteq \overline{g}_-^\uparrow$  then
13.      $\overline{G}_+ \leftarrow \overline{G}_+ \setminus \overline{g}_+^\uparrow$ ;

```

Fig. 2. Algorithm DiagnosticTestsScalingAndInferring

Definition 17. An *i*-test for \mathbb{P}_\pm is called *irredundant* and denoted by h_{irr} iff there is no smaller *i*-test $h \sqsubset h_{\text{irr}}$ for \mathbb{P}_\pm .

Notions of maximally redundant and good tests on FDs are proposed by X.A.Naidenova and Y.Polegaeva in 1985 [28] and adopted to implications in [30].

Definition 18. An *i*-test for \mathbb{P}_\pm is *maximally redundant* (or *closed*) *i*-test and denoted by h_{max} iff $h_{\text{max}}^{\diamond\diamond} = h_{\text{max}}$.

Definition 19. An *i*-test for \mathbb{P}_\pm is called *good* and denoted by h_{good} , iff there is no object $g \in G_+$, $g \notin h_{\text{good}}^\diamond$ such that $(h_{\text{good}}^\diamond \cup g)^\diamond$ is an *i*-test.

An irredundant, maximally redundant or good *i*-test can be named with prefix “ASC-” if it is represented as ASC, i.e. a pair (h^\diamond, h) . Maximally redundant ASC-test is named as closed ASC-test for short.

4 Relations

4.1 Venn diagram

Let us illustrate several kinds of ASC-tests proposed in the previous section using the Venn diagram in Fig. 3. Definitions of predicates that form following sets are omitted for simplicity, for \mathbb{P}_\pm (\mathbb{P}_\mp):

- U is the set of ASC-tests $\{(h^\diamond, h) \mid h \text{ is an i-test}\}$;
- X is the set of good ASC-tests $\{(h^\diamond, h) \mid h \text{ is a good i-test}\}$;
- Y is the set of irredundant ASC-tests $\{(h^\diamond, h) \mid h \text{ is an irredundant i-test}\}$;
- Z is the set of closed ASC-tests $\{(h^\diamond, h) \mid h \text{ is an i-test, and } h = h^{\diamond\diamond}\}$.

Each set named as bold italic number is a set of ASC-tests, which can be inferred using intersection (or/and negation, subtraction) of sets U, X, Y , and Z . Each number-named set is bounded by lines in Fig. 3. Let us show, that the set **6**, i.e., the set $(Y \cap Z) \setminus X$, is \emptyset .

Proposition 2. *Any irredundant and closed ASC-test is good.*

Proof. Consider the opposite situation. Let this (A, h) exists and suppose it is in the set $(Y \cap Z) \setminus X \neq \emptyset$. Since it is not a good i-test, $\exists g$ such that $g \notin h^\diamond$, $g \in G_+$ and $(h^\diamond \cup g)^\diamond$ is an i-test. This means that $h^\diamond \subset (h^\diamond \cup g)$ and using (9), we get $(h^\diamond \cup g)^\diamond \sqsubseteq h^{\diamond\diamond}$. The situation satisfies property 1 from Corollary 1 and strict inclusion take place otherwise $(h^\diamond \cup g)^\diamond$ is not an i-test. Note that $h = h^{\diamond\diamond}$, because h is a closed set. It means that an i-test $(h^\diamond \cup g)^\diamond \sqsubset h$ exists. However h is irredundant, which means that $(h^\diamond \cup g)^\diamond$ strictly included in h does not exist. \square

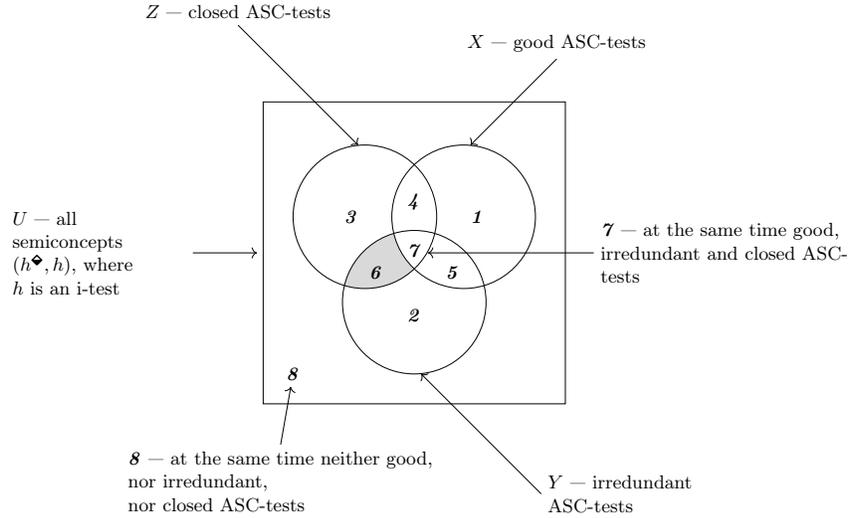


Fig. 3. Venn diagram of ASC-test types.

Example 1. Let us illustrate the kinds of ASC-tests by the toy example based on Tab. 1 from [36]. Let object descriptions be a conjunction of object properties.

Let h_i be an ASC-test, where $i = \overline{1,8}$ are indices of sets from the Venn diagram mentioned above:

- $h_1 = (g_2, (m_1 = 1 \wedge m_2 = 2 \wedge m_3 = 0))$.
- $h_2 = (g_1, (m_2 = 1 \wedge m_4 = 0))$.
- $h_3 = (g_1, \delta(g_1))$.
- $h_4 = (\{g_1, g_3\}, (m_1 = 0 \wedge m_2 = 1))$.
- $h_5 = (\{g_1, g_3\}, (m_1 = 0))$.
- $\nexists h_6$ according to Prop.2.
- $h_7 = \emptyset$, because it is not represented in the table.
- $h_8 = (g_1, (m_2 = 1 \wedge m_3 = 1 \wedge m_4 = 0))$.

Table 1. Learning structure

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_2	1	2	0	1	1
g_3	0	1	0	1	1
g_4	1	2	1	0	2
g_5	1	1	0	1	2
g_6	1	1	1	2	2

To illustrate an ASC-test from set **7** in Fig. 3, let us use data from Tab. 2. It is $(g_1, m_1 = 1 \wedge m_2 = 0)$.

Table 2. Learning structure to illustrate an ASC-test from set **7** in Fig. 3

G	m_1	m_2	Cl
g_1	1	0	1
g_2	0	1	2

4.2 Closed Good I-tests and JSM-hypotheses

The next Proposition reveals the following idea: if there exists a concept with the minimal intent, which is an i-test, then its extent is not included in any extent of other closed i-test, and vice versa. The proof can be divided into two parts.

Proposition 3. *The definitions of minimal JSM-hypotheses and closed good i-tests are equivalent.*

Proof. On the one side, for each closed good i-test being the intent of (A, h) $\nexists g \in G_+, g \not\subseteq h^\diamond$ such, that $(h^\diamond \cup g)^\diamond$ is an i-test. According to Proposition 2 this means that $\nexists g$ such that i-test $(h^\diamond \cup g)^\diamond \sqsubset h$. Take into account that $(h^\diamond \cup g)^\diamond$ is obligatory an intent, e.g. $((h^\diamond \cup g)^\diamond)^\diamond, (h^\diamond \cup g)^\diamond$. Closed i-test h , could be generated only by intersection of h^\diamond with object descriptions not included in this extent. Thus h is an extent, and h is a minimally possible intent.

On the other side, for each minimal JSM-hypotheses being the intent of (h^\diamond, h) does not exist $d \sqsubset h$ such that d is a closed i-test. Suppose, that h is not a closed good i-test, i.e. $\exists g \in G_+, g \not\subseteq h^\diamond$ such that $(h^\diamond \cup g)^\diamond$ is an i-test, and $(h^\diamond \cup g)^\diamond \sqsubset h$ according to Proposition 2. The contradiction finishes the proof. \square

4.3 Discussion about other relations

Let us recall some results about relation between representations of VS and minimal JSM-hypotheses (if there exists only one maximal representation of VS) [18]:

- the minimal JSM-hypothesis forms the maximal boundary of VS;
- the set of all proper predictors for the target class forms the minimal boundary of VS.

As a result of transitivity, a closed ASC-test is also a maximal representation of VS. However we claim, that a proper predictor for target class is not a minimal representation of VS w.r.t. implicit representation of order $\sqsubseteq_{VS_{\max}^\diamond}$ discussed in subsection 3.3 instead of \sqsubseteq from [18]. Let us show the difference in the following small example with the use of Tab. 3.

Table 3. Learning structure for counter-example modified from Tab. 1

G	m_1	m_2	m_3	m_4	K
g_1	0	1	1	0	1
g_3	0	1	0	1	1
g_5	1	1	0	1	2
g_6	1	1	1	2	2

There are the following classifiers inferred from Tab. 3:

- proper predictors for \mathcal{P}_\pm are $m_1 = 0$ and $m_4 = 0$;
- VS_{\max} is $m_1 = 0 \wedge m_2 = 1$;
- VS_{\min} is only $m_1 = 0$.

It is clear that elements of VS_{\min} are irredundant good i-tests. Let us prepare supplementary Proposition 4.

Proposition 4. *Given a good i-test $h \in D$, if $d \sqsubset h$ is an i-test, then d is a good i-test.*

Proof. Let $d \sqsubset h$ be an i-test for \mathcal{P}_+ , then $h^\diamond \sqsubseteq d^\diamond$ according to equation (10). If $h^\diamond \subset d^\diamond$, then according to property 1 of Corollary 2 there exists the object g such that $d \sqsubseteq g^\diamond$, and there exists $c \sqsubseteq h$, $d \not\sqsubseteq c$, $c \not\sqsubseteq g^\diamond$. Such object can exist only in \mathcal{P}_- and falsify h , otherwise there is a contradiction to the condition of this Proposition.

If $h^\diamond = d^\diamond$, then according to property 2 of Corollary 2 and to the definition of closed good i-test, there does not exist the object discussed above. Thus d is an i-test with the same extent as a good i-test h , i.e. also a good i-test. \square

Proposition 5. *If a version space is given and there exists only one VS_{\max} , then there can exist several elements of VS_{\min} that are irredundant good i-tests.*

Proof. Minimal JSM-hypotheses are VS_{\max} (Theorem 1, [18]). Closed good i-tests are minimal JSM-hypotheses (proved in Proposition 3). Thus by transitivity closed i-test is VS_{\max} . In Proposition 4 it is shown that all i-test subsets of VS_{\max} are exactly good i-tests (or they are not i-tests). Thus these i-tests satisfy the condition $\sqsubseteq_{VS_{\max}^\diamond}$. Let us choose within these i-tests those, that satisfy the condition of VS_{\min} . They are exactly irredundant good i-tests. \square

Let us note, that one of the last properties of Theorem 1, saying “that version space is represented by a convex set in the lattice of all pattern intents with maximal element minimal JSM-hypothesis” should be improved.

Proposition 6. *If the version space VS has only one VS_{\max} , then elements from VS_{\max} , VS_{\min} , and VS_{mid} form join-semilattice of i-tests w.r.t. \sqsupseteq with maximal element VS_{\max} .*

Proof. Note, that VS_{\max} due to equivalence with closed good i-test can exist only in the set $\mathbf{4} \cup \mathbf{7}$ of Fig. 3. VS_{\min} can exist only in the set $\mathbf{7} \cup \mathbf{5}$ according to Proposition 5. If there is the set VS_{mid} , then by the definition it consists of elements from $\mathbf{1} = Z \setminus \{\mathbf{4} \cup \mathbf{5} \cup \mathbf{7}\}$.

Because of the fact that VS_{mid} and VS_{\min} refers to one and only one VS_{\max} , all these i-tests are in equivalence relation under the object covering. This means, that for any two h, h_1 from $\overline{VS} \stackrel{\text{def}}{=} VS_{\max} \cup VS_{\min} \cup VS_{\text{mid}}$, h^\diamond is equal to h_1^\diamond , and the natural order is $\sqsubseteq_{VS_{\max}^\diamond}$. Then the join-semilattice $(\overline{VS}, \sqsubseteq_{VS_{\max}^\diamond})$ is obtained. \square

5 Conclusion

The equivalence between good maximally redundant tests (GMRTs) and minimal JSM-hypotheses and between minimal representations of version spaces and irredundant tests, that are included in GMRTs, are shown. Extracting functional dependencies from pattern structures is an important question for the future work.

References

1. Baixeries, J., Kaytoue, M., Napoli, A.: Computing functional dependencies with pattern structures. In: Szathmary, L., Priss, U. (eds.) Proceedings of The Ninth International Conference on Concept Lattices and Their Applications. CEUR Workshop Proceedings, vol. 972, pp. 175–186. CEUR-WS.org (2012)
2. Barbut, M., Monjardet, B.: *Ordre et classification: algèbre et combinatoire*. Collection Hachette université: Méthodes mathématiques des sciences de l’homme, Hachette (1970)
3. Baskakova, L., Zhuravlev, Y.: A model of recognition algorithms with representative samples and systems of supporting sets. U.S.S.R. Comput. Math. Math. Phys. 21(5), 189–199 (1981)
4. Brézellec, P., Soldano, H.: Tabata: A learning algorithm performing a bidirectional search in a reduced search space using a tabu strategy. In: Proceedings of the 13th European Conference on Artificial Intelligence. pp. 420–424 (1998)
5. Brito, P.: Order structure of symbolic assertion objects. I.E.E.E. Trans. Knowl. Data Eng. 6(5), 830–835 (1994)
6. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: A new approach to classification by means of jumping emerging patterns. In: CEUR Workshop Proceedings. vol. 939, pp. 15–22 (2012)
7. Chegis, I., Yablonskii, S.: Logical methods of electric circuit control. Trudy Mian SSSR 51, 270–360 (1958), (in Russian)
8. Cohen, W.W., Hirsh, H.: The learnability of description logics with equality constraints. Machine Learning 17(2), 169–199 (1994)
9. Cosmadakis, S.S., Kanellakis, P.C., Spyratos, N.: Partition semantics for relations. Journal of Computer and System Sciences 33(2), 203 – 233 (1986)
10. Dmitriev, A., Zhuravlev, Y.I., Krendelev, F.: On mathematical principles for classification of objects and phenomena. Discrete Analysis 7, 3–17 (1966), in Russian
11. Domenach, F., Leclerc, B.: On the roles of galois connections in classification. In: Exploratory Data Analysis in Empirical Research, pp. 31–40. Studies in Classification, Data Analysis, and Knowledge Organization, Springer (2003)
12. Ferré, S., Ridoux, O.: Introduction to logical information systems. Information Processing & Management 40(3), 383 – 419 (2004)
13. Finn, V.: On machine-oriented formalization of plausible reasoning in the style of F.Backon–J.S. Mill. Semiotika i Informatika 20, 35–101 (1983), (in Russian)
14. Ganascia, J.G.: Tdis: an algebraic formalization. In: Proceedings of the 13th International Joint Conference on Artificial Intelligence. vol. 2, pp. 1008–1015 (1993)
15. Ganter, B., Wille, R.: *Formal concept analysis: mathematical foundations*. Springer, Berlin (1999)
16. Ganter, B., Kuznetsov, S.O.: Formalizing hypotheses with concepts. In: Conceptual Structures: Logical, Linguistic, and Computational Issues: Proceedings of the 8th International Conference on Conceptual Structures. pp. 342–356 (2000)
17. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: Delugach, H.S., Stumme, G. (eds.) Conceptual Structures: Broadening the Base: Proceedings of the 9th International Conference on Conceptual Structures. pp. 129–142 (2001)
18. Ganter, B., Kuznetsov, S.: Hypotheses and version spaces. In: Conceptual Structures for Knowledge Creation and Communication, Lecture Notes in Computer Science, vol. 2746, pp. 83–95 (2003)

19. Kuznetsov, S.: Galois connections in data analysis: Contributions from the soviet era and modern russian research. In: Ganter, B., Stumme, G., Wille, R. (eds.) *Formal Concept Analysis*, LNCS, vol. 3626, pp. 196–225. Springer (2005)
20. Kuznetsov, S.: Interpretation on Graphs and Complexity Characteristics of a Search for Specific Patterns. *Nauchno-Tekhnicheskaya Informatsiya, Seriya 2 Informatsionnye protsessy i sistemy* 23(1), 23–27 (1989)
21. Kuznetsov, S.: JSM-method as a Machine Learning System. *Itogi Nauki i Tekhniki*, ser. *Informatika* 15, 17–50 (1991), (in Russian)
22. Kuznetsov, S.: Mathematical aspects of concept analysis. *Journal of Mathematical Sciences* 80(2), 1654–1698 (1996)
23. Luksch, P., Wille, R.: A Mathematical Model for Conceptual Knowledge Systems. In: Bock, H.H., Ihm, P. (eds.) *Proceedings of the 14th Annual Conference of the Gesellschaft fur Klassifikation (GfKI 1990)*. pp. 156–162 (1991)
24. Mitchell, T.M.: *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edn. (1997)
25. Mitchell, T.M.: Version spaces: A candidate elimination approach to rule learning. In: *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*. pp. 305–310. *IJCAI'77*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1977)
26. Mitchell, T.M.: *Version Spaces: An Approach to Concept Learning*. Tech. rep., Stanford University California (1978)
27. Naidenova, K.: The relational model of experimental data analysis. *Trans. of Ac. Sci. USSR, series' Technical Cybernetics'*,(4, 1982) pp. 103–119 (1982)
28. Naidenova, X.A., Polegaeva, J.G.: Model of human reasoning for deciphering forest's images and its implementation on computer. In: *Theses of papers and reports of school-seminar "Semiotic aspects of the intellectual activity formalization"*. Kutaisy, Georgia Soviet Socialist Republic (1985), (in Russian)
29. Naidenova, X.A., Polegaeva, J.G.: An Algorithm of Finding the Best Diagnostic Tests. In: Mintz, G., Lorents, E. (eds.) *The 4-th All Union Conference "Application of Mathematical Logic Methods"*. pp. 87 – 92 (1986), (in Russian)
30. Naidenova, X.: Machine Learning as a diagnostic task. In: Arefiev, I. (ed.) *Knowledge-Dialog-Solution, Materials of the Short-Term Scientific Seminar*, pp. 26 – 36. State North-West Technical University Press, St Petersburg, Russia (1992), in Russian
31. Naidenova, X.: The Data-Knowledge Transformation. In: Solovyev, V. (ed.) *Text Processing and Cognitive Technologies*, vol. 3, pp. 130–151. Pushchino (1999)
32. Naidenova, X.: Good classification tests as formal concepts. In: Domenach, F., Ignatov, D., Poelmans, J. (eds.) *Proceedings of the 10th International Conference on Formal Concept Analysis*. vol. 7278, pp. 211–226 (2012)
33. Naidenova, X.: *Machine Learning Methods for Commonsense Reasoning Processes: Interactive Models* (2010)
34. Nguifo, E.M., Njiwoua, P.: Iglue: A lattice-based constructive induction system. *Intelligent data analysis* 5(1), 73–91 (2001)
35. Norris, E.M.: Maximal rectangular relations. In: Karpiński, M. (ed.) *Fundamentals of Computation Theory: Proceedings of the 1977 International FCT-Conference, Poznań-Kórnik, Poland September 19–23, 1977*, vol. 56, pp. 476–481. Springer Berlin Heidelberg, Berlin, Heidelberg (1977)
36. Peskov, N.V.: Searching for informative fragments of object descriptions in the recognition tasks. Ph.D. thesis (2004)

37. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. *Expert Systems with Applications* 40(16), 6538 – 6560 (2013)
38. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Fuzzy and rough formal concept analysis: a survey. *International Journal of General Systems* 43(2), 105–134 (2014)
39. Poelmans, J., Kuznetsov, S.O., Ignatov, D.I., Dedene, G.: Formal concept analysis in knowledge processing: A survey on models and techniques. *Expert Systems with Applications* 40(16), 6601 – 6623 (2013)
40. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. *Lecture Notes in Computer Science* 5548, 314–339 (2009)
41. Zakrevskiy, A.: Algorithms of discrete automata synthesis. Moscow: Nauka (1971), in Russian
42. Zhang, X., Dong, G., Ramamohanarao, K.: Information-Based Classification by Aggregating Emerging Patterns. In: Leung, K., Chan, L.W., Meng, H. (eds.) *Intell. Data Eng. Autom. Learn. – IDEAL 2000. Data Mining, Financ. Eng. Intell. Agents*, LNCS, vol. 1983, pp. 48–53. Springer (2000)
43. Zhuravlev, Y., Ryazanov, V., Senko, O.: Recognition. *Mathematical methods. Software. Practical Applications*. Moscow: Fazis (2006), in Russian
44. Zhuravlev, Y.I., Nikiforov, V.V.: Recognition algorithms based on computation of estimates. *Cybernetics* 7(3), 387–400 (1971)
45. Zhuravlev, Y., Ryazanov, V., Senko, O., Biryukov, A., Vetrov, D., Dokukin, A., Kropotov, D.: The program system for intellectual data analysis, recognition and forecasting. *WSEAS Transactions on Information Science and Applications* 2(1), 55–58 (2005)