Estimation of Errors Rates for FCA-based Knowledge Discovery*

Dmitry V. Vinogradov[1,2][0000−0001−5761−4706]

[1] FRC Computer Science and Control RAS, Moscow 119333, Russia
vinogradov.d.w@gmail.com
[2] Russian State University for Humanities, Moscow 125993, Russia
http://isdwiki.rsuh.ru

Abstract. In this paper we present an approach for estimating the error rate of bitset representation of object descriptions in FCA-based knowledge discovery. Errors of this kind lead to overfitting phenomenon. The key technique used in our approach is based on the Möbius function on finite partial ordered sets, which was introduced by G.-C. Rota.

Keywords: FCA · JSM-method · Möbius function · error rates.

Introduction

'JSM-method of automatic hypotheses generation' is an approach to Knowledge Discovery that was proposed by V.K. Finn (see, [2], [3]). Initial goal was to formalize 'Inductive Logic' proposed by sir John Stuart Mill in 1848 (at the same time as Boolean Logic was proposed by John Boole) using Boolean Algebra and Many-Valued Logic. This approach has been extended to predict the target property of test examples with the help of generated causes of the property (also called 'hypotheses'). This approach is actually a Machine Learning techniques but high computational complexity (see [7]) is an obstacle to its scalability.

Later JSM-method has been extended to arbitrary (lower semi-)lattices of object descrptions, where similarity operation on object descriptions (wedge operation) satisfies usual idempotent, commutative and associative laws. In [1], [7] FCA-based techniques [4] were applied to JSM-method. FCA provides a very efficient representation for training objects by means of bitsets (fixed length strings of bits) with bit-wise multiplication as similarity operation on them.

The author [12] investigated the 'overfitting' phenomenon for JSM-method by means of so-called 'phantom' or 'accidental' hypotheses. In practice they occur when generated hypotheses are contained in descriptions of examples of the opposite sign (so-called 'counter-examples') that do not possess the target property. JSM-method rejects such hypotheses by means of the 'forbidding counter-example test' (FCET), however the remaining hypotheses can be contained in test examples and erroneously classify them as having the target property, hence causing the 'overfitting'. This phenomenon was experimentally detected

* partially supported by RFBR grant 18-29-03063mk.

by RSUH student L.A. Yakimova within her master project under supervision of the author. Another approach to the overfitting of FCA-based Machine Learning was invented and studied in [8].

The analysis of such situations reveals that FCET can reject some phantom hypotheses if the data contains errors in values of some attributes. This paper considers a possibility to estimate error rates in attribute values taking into account the lattice structures on them. Without loss of generality, we restrict ourselves to the case of single target attribute. The general case is reduced to this one by assuming the independence of errors rates of values of different attributes.

1 Basic definitions and results

1.1 Bitset Encoder Algorithm

A **(finite) context** is a triple (G, M, I) where G and M are finite sets and $I \subseteq G \times M$. The elements of G and M are called **objects** and **attributes**, respectively. As usual, we write gIm instead of $\langle g, m\rangle \in I$ to denote that object g has attribute m.

For $A \subseteq G$ and $B \subseteq M$, define

$$A' = \{m \in M | \forall g \in A(gIm)\}, \tag{1}$$

$$B' = \{g \in G | \forall m \in B(gIm)\}; \tag{2}$$

so A' is the set of attributes common to all the objects in A and B' is the set of objects possesing all the attributes in B. The maps $(\cdot)' : A \mapsto A'$ and $(\cdot)' : B \mapsto B'$ are called **derivation operators** (**polars**) of the context (G, M, I).

A **concept** of the context (G, M, I) is defined to be a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The first component A of the concept (A, B) is called the **extent** of the concept, and the second component B is called its **intent**. The set of all concepts of the context (G, M, I) is denoted by $L(G, M, I)$.

Let (G, M, I) be a context. For concepts (A_1, B_1) and (A_2, B_2) in $L(G, M, I)$ we write $(A_1, B_1) \leq (A_2, B_2)$, if $B_1 \subseteq B_2$. The relation $\leq$ is a **partial order** on $L(G, M, I)$.

Element $x \in L$ of finite lattice $\langle L, \wedge, \vee\rangle$ is called $\vee$**-irreducible**, if $x \neq \emptyset$ and for all $y, z \in L$ $y < x$ and $z < x$ imply $y \vee z < x$. Element $x \in L$ of finite lattice $\langle L, \wedge, \vee\rangle$ is called $\wedge$**-irreducible**, if $x \neq \emptyset$ and for all $y, z \in L$ $x < y$ and $x < z$ imply $x < y \wedge z$.

If subsets of attributes $\{g_i\}' \subset M$ and $\{g_j\}' \subset M$ are intents of objects $g_i \in G$ and $g_j \in G$, respectively, with corresponding bitsets, then the derivation operator on a pair of objects corresponds to bit-wise multiplication, since $\{g_i, g_j\}' = \{g_i\}' \cap \{g_j\}'$. Moreover, polars correspond to iteration of bit-wise multiplication (in arbitrary order) of corresponding bitset-represented objects and attributes, respectively. The last remark is important, since bit-wise multiplication is a basic operation of modern CPU and GPGPU. In terms of JSM-method the binary operation $\cap : 2^M \times 2^M \to 2^M$ is called 'similarity operation'. This operation defines a low-semilattice on subsets of attributes.

Descriptions of objects can combine discrete and continuous attrtibutes. Here we restrict ourselves to discrete case only, and we will consider the continuous case in another paper. The set of objects' descriptions must be a part of extended set F of 'fragments' supplied with binary 'similarity' operation $\wedge : F \times F \to F$, which is idempotent $x \wedge x = x$, commutative $x \wedge y = y \wedge x$, and associative $x \wedge (y \wedge z) = (x \wedge y) \wedge z$. Moreover, there is the minimal element $\emptyset$ satisfying $x \wedge \emptyset = \emptyset$ (so-called 'trivial fragment'). In FCA this construction is realized by means of *pattern structures* [5]. We convert a fragment into a subset of attributes in such a way that similarity operation becomes set-theoretic intersection (and bit-wise multiplication on corresponding bitsets). We reformulate Basic Theorem 1 of FCA in the following form to construct such an algorithm for encoding values of each attribute, then we form their concatenation for encoding whole object descriptions.

Theorem 1. *[4] For every finite lattice $\langle L, \wedge, \vee \rangle$ let G be a (super)set of all $\wedge$-irreducible elements and M be a (super)set of all $\vee$-irreducible elements. For $gIm \Leftrightarrow g \geq m$ the formal context (G, M, I) generates $L(G, M, I)$, which is isomorphic to the original lattice $\langle L, \wedge, \vee \rangle$.*

In [13] we used this theorem to prove correctness of the following algorithm with respect to the property that similarity operation between values corresponds to the bit-wise multiplication between their codes:

```
Data: set $V$ of values of current attribute
Result: matrix $B$ with rows as bitset codes of values
$V := topological\_sort(V)$; // topological sorting
$T := order\_matrix$; // transitive closure of cover relation
$\forall i[Del[i] = false]$; // deleted columns
for $(index = 2; index < n; ++index)$ do
    for $(indx = 1; indx < index; ++indx)$ do
        for $(ndx = 0; ndx < indx; ++ndx)$ do
            if $(T[\,][V[index]] == T[\,][V[indx]] \& T[\,][V[ndx]])$ then
                $Del[V[index]] := true$;
            end
        end
    end
end
for $(index = 2; index < n; ++index)$ do
    for $(indx = 1; indx < index; ++indx)$ do
        if $\neg Del[indx]$ then
            $\Rightarrow B[indx][index] := T[index][indx]$;
        end
    end
end
```

Algorithm 1: Encoder Algorithm

For instance, when we consider famous Mushroom Data Set [11] from Machine Learning Repository at University of California in/ Irvine, the following values of 'spore print color' are available: black (k), brown (n), buff (b), chocolate (c), green (r), orange (o), purple (u), white (w), and yellow (y). Let us concentrate on *black, brown, buff, chocolate*, and *yellow* values. The corresponding semi-lattice is shown in Fig. 1.

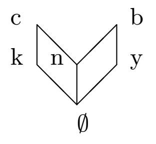

Fig. 1. A fragment of spore print color values semi-lattice

We added ∅ value to denote the absence of similarity of spore print colors between several training examples that generate some hypotheses.

The order corresponds to "to be more specific/general" relation between values. For example, *buff* is brown-yellow and *chocolate* is black-brown. Hence the similarity between mushrooms with chocolate spore print (c) and ones with buff spore print (b) has brown color (n) of spore print as common.

The algorithm has the following steps:

1. Topological sorting of elements of the semilattice.
2. In the context of order $\geq$ look for columns that coincide with bit-wise multiplication of previous ones (every such column corresponds to $\vee$-reducible element).
3. All found ($\vee$-reducible) columns are removed.
4. Rows of reduced context form bitset representations of the corresponding values.

We sort the values as $k < n < c < y < b$ in correspondence with the partial order on them. Then Theorem 1 gives a big context corresponding to $\geq$.

$values$	k	n	c	y	b
k	1	0	0	0	0
n	0	1	0	0	0
c	1	1	1	0	0
y	0	0	0	1	0
b	0	1	0	1	1

The column 'c' is equal to the product of columns 'k' and 'n' since $c = k \vee n$. Hence it is reducible. The other $\vee$-reducible element of the lattice is 'b' (again column 'b' is a product of columns 'n' and 'y').

The rows of reduced context

values	*k*	*n*	*y*
k	1	0	0
n	0	1	0
c	1	1	0
y	0	0	1
b	0	1	1

are the bitset representations of the corresponding colors of spore print.

The encoding of the whole description is a concatenation of bit-set presentation of values of its features in some fixed order. Since the bitset representations of different values of same feature have same length, the bit-wise multiplication of the whole representations reduces to the bit-wise multiplications of the corresponding parts, and the similarity between objects is given by the component-wise similarity of their intents.

1.2 Overfitting Phenomenon for JSM-method

We begin with a demonstration of the phenomenon by JSM-method's application to a school problem in geometry. The task is to learn sufficient conditions on a convex polygon to be circled and to predict this property for test examples. Hence there are two target classes: the positive one (with a possible circle around the figure) and the negative one.

The training sample contains regular triangle, rectangular triangle, square, isosceles trapezoid, and diamond (the last figure is negative, the rest contains positive training examples). The test sample contains isosceles triangle, rectangle, and deltoid. We consider the most general case of the corresponding polygon. Hence, for instance, isosceles trapezoid has bases of different sizes and differs from rectangle.

We represent each polygon by a subset of attributes from the following list:

(a) the figure is a triangle;
(b) the figure is a quadrangle;
(c) the figure has a right angle;
(d) the figure has a pair of equal length sides;
(e) all sides of the figure have same length;
(f) the figure has a pair of parallel sides;
(g) the figure has a pair of equal angles;
(h) all angles of the figure are equal.
(i) the sum of the opposite angles of the quadrangle is equal to π.

Hence, the training context $(G^+, M = \{a, b, c, d, e, f, g, h, i\}, I)$ is

training objects	*a*	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*
regular triangle	1	0	0	1	1	0	1	1	0
rectangular triangle	1	0	1	0	0	0	0	0	0
square	0	1	1	1	1	1	1	1	1
isosceles trapezoid	0	1	0	1	0	1	1	0	1

This context generates the following concepts with extents of cardinality > 1

$$\langle\{regular\ triangle, rectangular\ triangle\}, \{a\}\rangle,$$
$$\langle\{regular\ triangle, square\}, \{d, e, g, h\}\rangle,$$
$$\langle\{regular\ triangle, square, isosceles\ trapezoid\}, \{d, g\}\rangle,$$
$$\langle\{square, isosceles\ trapezoid\}, \{b, d, f, g, i\}\rangle.$$

The first concept $\langle\{a\}', \{a\}\rangle$ corresponds to the well-known geometric theorem "**Each triangle can be circumscribed by a circle**". The second one expresses the famous fact about regular polygons "**Vertices of regular polygon lie on a circle**". This concept has the form $\langle\{e, h\}', \{e, h\}''\rangle$. The fourth concept represented as $\langle\{i\}', \{i\}''\rangle$ corresponds to the well-known geometric theorem "**Every quadrangle with the sum of opposite angles equal to π can be circumscribed by a circle**". The third concept is a 'phantom' because its extent contains two types of objects: a regular_triangle with the first 'real' cause, and quadrangles with the sum of opposite angles equal to π (square and isosceles trapezoid). Its intent consists of 'accidental common' attributes. Luckily, the forbidden counter-example test (FCET) procedure rejects this concept because of the counter-example

$counter - example$	a	b	c	d	e	f	g	h	i
$diamond$	0	1	0	1	1	1	1	0	0

However, the situation is very subtle, since the 4th concept has two types of objects in its extent. However, it corresponds to the true geometric fact! We think that this concept is a 'real cause' as opposed to 'phantom concept' 3.

Test sample G^τ contains

$test\ objects$	a	b	c	d	e	f	g	h	i
$isosceles\ triangle$	1	0	0	1	0	0	1	0	0
$rectangle$	0	1	1	1	0	1	1	1	1
$deltoid$	0	1	0	1	0	0	1	0	0

JSM-method predicts the first test (isosceles triangle) positively through the 1st concept. The second test object (rectangle) is classified positively by applying the 4th concept. JSM-method might incorrectly predict the target property of the last test case if the 3rd hypothesis is not rejected. This is exactly the phenomenon of overfitting: the hypothesis is consistent with the training sample, but it leads to the incorrect classification of test examples.

A similar situation occurs in real data experiments with the use of JSM-method. For example, consdier an application of JSM-method to the study of toxicity of substituted nitrobenzenes [6]. The data was collected by pharmacologists from the Liverpool University. RSUH student Anastasia S. Oparysheva detected suspicious phenomenon when she analyzed the results of this experiment with respect to overfitting within her undergraduate project [9] under the supervision of the author.

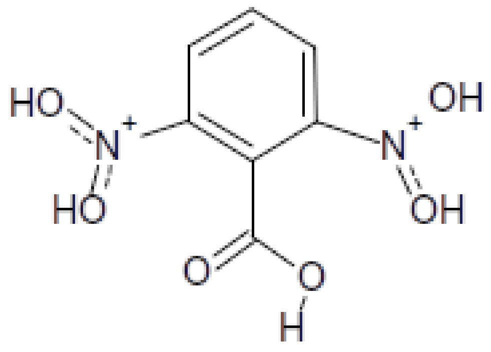

Fig. 2. Concept suspicious to be a phantom

Concept (hypothetical cause of toxicity) 28 with extent consisting 2 elements (training examples 37 and 39) has the form shown in figure 2.

However, example 37 and other 6 training examples generate alternative hypothesis 3. Similarly, example 39 contains alternative concept 41 with 16 elements extent. There is a plausible assertion that hypothesis 28 is a 'phantom' because of an accidental coincidence fragment between two training examples each of which contains some different 'real cause' (example 37 has 'real cause' 3, and example 39 contains 'real cause' 41).

This case was not unique. More than 10 percent of concepts without counter-examples exhibit the same behavior. The aim of her study was to study empirically the overfitting phenomenon, which was previously investigated theoretically by the author in [12]. We present some results from this article below.

An attribute is called **essential**, if it appears in some 'real cause'. Here 'cause' is a set of attributes. Assume for the sake of simplicity that two 'real causes' have no common attributes. Other attributes are called **accompanying**. Hence we partition set M of all attributes into three subsets: the first 'real cause', the second one, and accompanying attributes.

Term 'real cause' corresponds to a generator of intent $\{a\}$ of 1st concept and $\{i\}$ of 4th concept in our initial illustrative example. However in mathematical study below it means just a special subset of attributes other than the accompanying ones. Last attributes form building blocks for 'phantom' concepts. So 'real' in 'real causes' means nothing! It's simply initially introduced term to distinct the group of essential attributes from accompanying ones.

Assume for the sake of simplicity that counter-examples do not contain any essential attribute. Now we introduce probabilistic model to simultaneously generate accompanying attributes for a pair of training examples and m counter-examples.

Denote the number of counter-examples by m, and the number of accompanying attributes by n. It is clear that the accompanying attributes of training objects and counterexamples form a $(2 + m) \times n$ binary matrix. It contains

$N = (2+m) \cdot n$ bits. Accompanying attributes are generated by Bernoulli series of N tests.

Bernoulli series of N tests is the probability distribution on $\{0,1\}^N$ with

$$\mathbf{P}(x_1 = \delta_1, \ldots, x_N = \delta_N) = \prod_{j=1}^{N} p^{\delta_j} \cdot (1-p)^{1-\delta_j},$$

where $0 < p < 1$. The number $p > 0$ is called *success probability* $x_j = 1$ in test j.

Then we set all the attributes of first "real" cause to 1s, all attributes of the second "real" cause to 0s, and add accompanying attributes of first training example to obtain first training example itself. We generate the second training example by setting attributes of the first "real" causes to 0, and of the second one to 1. All counter-examples have 0 in positions corresponding to both "real" causes.

As result we obtain context $2 \times |M|$ and list of m counter-examples. What is the probability to generate concept with 2 element extent without any counter-example from the list?

Theorem 2. *If number n of random attributes tends to infinity, the probability of success equals to $\sqrt{\frac{a}{n}}$, and there are $m = b \cdot \sqrt{n}$ counterexamples, then the probability of an accidental formal concept with 2 elements extent and without any counterexample is $1 - e^{-a} - a \cdot e^{-a} \cdot [1 - e^{-b \cdot \sqrt{a}}]$ at limit.*

Note, that even smaller number $1 - e^{-a} - a \cdot e^{-a}$ is positive, since it coincides with the probability that the Poisson variable Y_a with mean a has value $Y_a > 1$.

Recently Lyudmila A. Yakimova, a former master student of the Russian State University for Humanities made experimental studies [15] on behavior of Machine Learning procedures based on FCA. She also detected the essential overfitting phenomenon. For example, on Mushroom Data Set [11] the JSM method generates several 'phantom' concepts. And as consequence, their use resulted in the wrong classification of toadstools as eatable mushrooms.

Another result of Yakimova's study is a higher rate of 'phantom' concepts than its estimate by the theorem. The reason is in the difference of frequency of appearance of different attributes. Moreover, Yakimova's experiments do not detect overfitting phenomenon for VKF method of Machine Learning based on FCA [14].

2 Error Rates for Values

2.1 Problem Explanation

While checking the condition of forbidding counter-examples, the similarity of some training examples can be contained in description of a counter-example. JSM-method rejects such similarities, however some suspicious hypotheses may be missed if some values of attributes were entered erroneously. Can we estimate the rate of such errors?

Consider again Figure 1. Assume that an expert mistakenly replaces buff spore print color (b) by yellow one (y) for some counter-example. Then the similarity with a mushroom with chocolate spore print has common brown color (n) and can not be included into counter-example, so the procedure saves the hypothesis. If such similarity is phantom it leads to overfitting.

It is clear that value $w \in V$ frequently replaces value $v \in V$ when $w \leq v$. The case of totally fatal mistake is ignored in this study. Denote the rate of such errors $r(v|w)$. The problem is that there does not exist a way to discover v, we see only w as a value entered by an expert. To resolve this difficulty the Möbius function from the incidence algebra is used.

2.2 Möbius functions on finite partial ordered sets

In fundamental work [10] Gian-Carlo Rota introduced the definition of Möbius function on (locally) finite partial ordered sets. It is a working tool for our approach. Below we will recall some key concepts and results of this theory.

Consider the set of real-valued functions of two variables on V with the property $f(x, y) = 0$, if $x \not\leq y$. It has the structure of an associative algebra over the real field if we define the product of such functions $h = f \cdot g$ as

$$h(x, y) = \sum_{z:x \leq z \leq y} f(x, z) \cdot g(z, y). \tag{3}$$

Addition and multiplication by constants are defined in a natural way. This structure is called **incidence algebra** of the given poset. This algebra has the **identity** element $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ otherwise, the **Kronecker delta**.

The **zeta** function $\zeta(x, y)$ is an element of incidence algebra such that $\zeta(x, y) = 1$ if $x \leq y$ and $\zeta(x, y) = 0$ otherwise. It has the inverse element $\mu(x, y)$, **Möbius function**. The proof of the next statement is trivial check.

Proposition 1. *Function defined by induction as $\mu(x, x) = 1$ and*

$$\mu(x, y) = -\sum_{z:x \leq z < y} \mu(x, z) \tag{4}$$

is the inverse element to zeta function.

Proposition 2. *Let $f(x)$ be a real-valued function, defined on (locally) finite p.o.set V. Let an element $v \in V$ exists with property that $f(x) = 0$ unless $x \leq v$. Suppose that*

$$g(x) = \sum_{y:x \leq y} f(y). \tag{5}$$

Then

$$f(u) = \sum_{z:u \leq z} \mu(u, z) g(z). \tag{6}$$

Proof. The function $g(x)$ is well-defined since it equals to $\sum\limits_{y:x\leq y\leq v} f(y)$, which is finite for a locally finite poset.

Substituting the right side of 5 into the right side of 6 and simplifying, we get

$$\sum_{z:u\leq z} \mu(u,z)g(z) = \sum_{z:u\leq z}\sum_{y:z\leq y} \mu(u,z)f(y) = \sum_{z:u\leq z}\sum_{y} \mu(u,z)\zeta(z,y)f(y).$$

Interchanging the order of summation, this becomes

$$\sum_{y} f(y) \sum_{z:u\leq z} \mu(u,z)\zeta(z,y) = \sum_{y} f(y)\delta(u,y) = f(u).$$

2.3 Algorithm

We can collect statistics for mistakenly missed phantom hypotheses h (with help of negative examples from tests sample). Let hypothesis h be included into some negative example o (either from the training or test sample) when we omit the values of the attribute under study. Such inclusions are called **pruned**.

Let us fix value x of the attribute under study. We compute the fraction $q_w(x)$ of pruned inclusions of hypotheses with value x into counter-examples with value w with respect to total of all pruned inclusions of hypotheses with value x into negative examples (either from training or tests samples).

Then we have

$$g_w(x) = \sum_{v:x\leq v} (\zeta(w,v) - \delta(w,v))r(v \mid w). \tag{7}$$

Here $\zeta(w,v)-\delta(w,v)$ determines the condition $w < v$ since there exists erroneous replacement invisible v by observable $w < v$. Summation holds because rates of different replaces are additive.

At first, we use Proposition 1 to compute Möbius functions for every attribute values lattice.

Then we compute statistics $g_w(x)$ by application of pruning inclusions.

Finally, we apply Proposition 2 to estimate errors rates as

$$r(v|w) = \sum_{z:v\leq z} \mu(v,z)q_w(z). \tag{8}$$

The omitted factor $\zeta(w,v) - \delta(w,v)$ means $w < v$.

Conclusion

We applied Möbius functions on finite posets to estimate rates of mistakenly replaces of attribute value by a smaller one that leads to overfitting in JSM-method. Experiments with Mushroom Dataset [11] demonstrate a very small (less than 0.01) rate of erroneous replacements 'buff' color of spore print by 'yellow' one.

Acknowledgments

Author would like to thank his colleagues from Dorodnicyn Computing Center of Federal Research Center "Computer Science and Control" of Russian Academy of Science and Russian State University for Humanities for long-term support and useful discussions. Author is grateful to anonymous reviewers, whose comments helped significantly improve the presentation.

References

1. Blinova, V.G. et al.: Toxicology Analysis by Means of the JSM-method. *Bioinform.* **19**(10): 1201–1207 (2003)
2. Finn, V.K.: J.S. Mill's inductive methods in artificial intelligence systems I. *Sci. Tech. Inform. Proc.* **38**, 385–402 (2011)
3. Finn, V.K.: J.S. Mill's inductive methods in artificial intelligence systems II. *Sci. Tech. Inform. Proc.* **39**, 241–261 (2012)
4. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
5. Ganter, B., Kuznetsov, S.O.: In: Stumme G. and Delugach H. (eds.) *Proc. 9th International Conference on Conceptual Structures (ICCS 2001)*, Lecture Notes in Artificial Intelligence, **2120**, 129–142 (2001)
6. Kharchevnikova, N.V. et al.: A JSM intelligent system for toxicity. Analysis of functional cumulation of chemical compounds. *Autom. Doc. Math. Linguist.* **51**, 20–26 (2017)
7. Kuznetsov, S.O.: Complexity of Learning in Concept Lattices from Positive and Negative Examples. *Discrete Applied Mathematics*, no. 142(1-3). – 111–125 (2004)
8. Makhalova, T., Kuznetsov, S.O.: On Overfitting of Classifiers Making a Lattice. In: Bertet K., Borchmann D., Cellier P., Ferré S. (eds) *Formal Concept Analysis. ICFCA 2017.* Lecture Notes in Computer Science, **10308**, 184–197 (2017)
9. Oparysheva, A.S.: Search for Accidental Hypotheses in Real Data. *Baccalaureate work (adv.: Vinogradov, D.V.)* Russian State University for Humanities, Moscow (2018) (in Russian)
10. Rota, G.C.: On the Foundations of Combinatorial Theory I. Theory of Möbius Functions. *Z. Wahrscheinlichkeitstheorie verw Gebiete* **2**, 340–368 (1964)
11. UCI Machine Learning Repository: Mushroom Data Set, https://archive.ics.uci.edu/ml/datasets/Mushroom. Last accessed 10 May 2020
12. Vinogradov, D.V.: Accidental formal concepts in the presence of counterexamples. In: *Proceedings of International Workshop on Formal Concept Analysis for Knowledge Discovery (FCA4KD 2017)*: CEUR Workshop Proceedings. **1921**, 104–112 (2017)
13. Vinogradov, D.V.: On Object Representation by Bit Strings for the VKF-Method. *Autom. Doc. Math. Linguist.* **52**, 113–116 (2018)
14. Vinogradov, D.V.: Continuous Attributes for FCA-based Machine Learning. *8th Workshop "What can FCA do for Artificial Intelligence?"* (2020)
15. Yakimova, L.A.: Experimental studies on behavior of solvers based in binary similarity operation. *Master's Degree Thesis (adv.: Vinogradov, D.V.)* Russian State University for Humanities, Moscow (2020) (in Russian)