

# Recovering Noisy Contexts with Probabilistic Formal Concepts<sup>\*</sup>

Vitaliy V. Martynovich<sup>1</sup> and Evgeniy E. Vityaev<sup>1,2</sup>

<sup>1</sup> Novosibirsk State University, Novosibirsk, Russia,  
vilco@ya.ru.

<sup>2</sup> Sobolev Institute of Mathematics, Novosibirsk, Russia,  
evgenii.vityaev@math.nsc.ru,

**Abstract.** The uncertainty in the environment typically generates noisy concept alternatives and leads to an overpopulated concept lattice. From a computational point of view, a straightforward filtering of the noisy concept lattice will suffer from an exponential-size computational overkill, and from a semantical one – will face numerous ambiguities due to an overfitting. We managed to bypass the filtering problem by applying a sort of probabilistic approach. We developed a probabilistic generalization of formal concepts which seems to avoid a monstrous combinatorial complexity of a complete context lattice construction. The theoretical base for this method is described, as well as a ready-to-work noise resistant algorithm. The algorithm has been tested and showed a moderate precision and recall rate on various datasets, including a toy one presented with the presence of a 2, 3 or 5% random noise.

**Keywords:** formal concept analysis, concept lattice, inductive learning, data mining, association rules, classification task

## 1 Introduction

Formal concepts may be successfully used as classification units [1, 2]. However, reviewing the concept lattice as a plain graph with the Formal Concept Analysis (FCA) works well until data become uncertain, when lattices can become prohibitively huge even on small-sized datasets.

There are some attempts to get rid of noise in data by concepts selection or filtering. E.g. measures of the concept stability has been shown to pick out the most reliable formal concepts [4, 5]. It was demonstrated that the stability index is relevant to data mining tasks and possesses several attractive properties [9].

Nevertheless, this is still not enough for uncertain environments [4]. Noisy clones overloading makes the calculation intractable even on small datasets. Formally speaking, a zero-populated concept context superimposed with a random Bernoulli noise is expected to produce exponential-size lattices [8].

---

<sup>\*</sup> The work is financially supported by the Russian Foundation for Basic Research Grant 15-07-03410-a

Approaches based on the hypothesis-making has been analyzed: performance of a model still suffers in the practical tasks environment [7]. The closest research domains are probably connected with the fuzzy concepts analysis, like [18].

In the paper we reconsider the problem of handling a possible noise in data by means of probability and logic. The origin of the probabilistic pattern for formal concepts lies in cognitive science, where they are closely related to the "natural classes" [16]. We will focus on developing a context recovery method keeping eye on the next key capabilities:

1. The stability of a reproduced context concept lattice with respect to a possible minor noise;
2. Computational tractability, avoiding filtering the whole concept lattice;
3. Handling the prediction ambiguity problem;
4. Relationship with the theory of category formation [17, 16].

The first step has been made in [12] – a probabilistic generalization of formal concepts goes here. The next step is to equip concepts with possible attribute negations and develop a logical language of a context probabilistic reasoning. We will also prove some technical facts about a consistency of probabilistic reasoning.

## 2 Formal concept analysis foundations

This section suggests a brief overview for a formal concept analysis framework [1, 2, 13] exploited in the paper.

A dataset is represented by an attribute-value cross-table. Formally speaking,

**Definition 1.** *A formal context is a triple  $(G, M, I)$  where  $G$  and  $M$  are the sets of an arbitrary nature and  $I \subseteq G \times M$  is a binary relation.*

On the formal context (or simply context) a derivation operator  $'$  is defined:

**Definition 2.**  *$A \subseteq G, B \subseteq M$ . Then*

1.  $A' = \{m \in M \mid \forall g \in A, (g, m) \in I\}$
2.  $B' = \{g \in G \mid \forall m \in B, (g, m) \in I\}$
3. *The pair  $(A, B)$  is called a formal concept if  $A' = B$  and  $B' = A$ .*

Generally speaking, formal concepts analysis concentrates on a concept lattice arising on concept extents from the natural subset order. However, we aim to avoid considering a concept lattice and exploit the intrinsic properties of the data. The implication is a core notion.

**Definition 3.** *The implication is a pair  $(B, C)$ ,  $B, C \subseteq M$ , which we write as  $B \rightarrow C$ . The implication  $B \rightarrow C$  is true on  $K = (G, M, I)$ , if  $\forall g \in G (B \not\subseteq g' \text{ or } C \subseteq g')$ . We denote the set of all true implications as  $Imp(K)$ .*

Implications are not only the forms a conceptual bridge from FCA to logic structures, but are an essential way of reasoning within a machine logic and prediction task particularly.

**Definition 4.** For any set of implications  $L$  we construct an operator of a direct inference  $f_L$  that adds all conclusions of applicable implications:

$$f_L(X) = X \cup \{C \mid B \subseteq X, B \rightarrow C \in L\}$$

The following theorem characterizes concepts by means of fixed points.

**Theorem 1 (see [2]).** For any set  $B \subseteq M$ ,  $f_{Imp(K)}(B) = B \Leftrightarrow B'' = B$ .

The theorem application may be illustrated on a simple formal context.

$m_1$	$m_2$	$m_3$
1	1	0
1	1	1
0	1	1
0	0	0

Table 1: A very simple formal context  $K_0$

We can reformulate concept lattice construction task by means of implications and an inference operator. It can be easily found out from Table 1 that attributes  $m_1$  and  $m_3$  determine the class of an object. In fact,  $m_1$  implies  $m_2$  and so does  $m_3$ . This is written as  $m_1 \rightarrow m_2$  and  $m_3 \rightarrow m_2$ . The sets  $\{m_1, m_2\}$ ,  $\{m_2, m_3\}$  and  $\{m_1, m_2, m_3\}$  are the formal concepts, so do fixed points of the direct inference operator. For example,

$$\{m_1\} \xrightarrow{f_{Imp(K)}} \{m_1, m_2\} \xrightarrow{f_{Imp(K)}} \{m_1, m_2\}$$

Thus indeed,  $\{m_1, m_2\}$  is a fixed point and a formal concept simultaneously.

### 3 Probabilistic logic on a formal context

Let us add some noise on  $K_0$ . We also extend  $K_0$ , by adding redundant objects duplicates. It will help to keep the noise level rather low in order to make a concept recovery practically possible.

Every single altering will change a concept lattice a lot. The first context is equivalent to  $K_0$  above and has the same concept lattice. However, the second one generates a lot of side concepts, provoked by noise: the sets  $\{m_2\}$  and  $\{m_1, m_3\}$  also become formal concepts. The amount of side concepts is increasing as more noise is incoming – the dependency tends to be asymptotically exponential [8].

The stability may be obtained in various ways. The most obvious way is computing some stability index in order to evaluate, does the concept from noisy context is good enough, either does it is produced by noise [4, 5].

The other one may be based on a different subject: instead of measuring a stability of concepts, a stability of implications is measured. An essential way to do this is to exploit a likelihood of attribute implications, but we will also generalize them up to logical formulas.

$m_1$	$m_2$	$m_3$
1	1	0
1	1	1
0	1	1
1	1	0
1	1	1
0	1	1
1	1	0
1	1	1
0	1	1
0	0	0

 Table 2:  $K_0$  populated with duplicates

$m_1$	$m_2$	$m_3$
0	1	0
1	1	1
0	1	1
1	1	0
1	0	1
0	1	1
1	1	0
1	1	1
0	1	1
0	0	0

 Table 3:  $K_{noise}$ , with a little bit noise

**Definition 5.** For a formal context  $K = (G, M, I)$  we introduce classical logical definitions:

- $L_K$  is a letters set and includes any  $m \in M$  as well as their negations  $\neg m$ ;
- $\Phi_K$  is a formulas set and is defined inductively: a letter is a formula and for any  $\phi, \psi \in \Phi_K$  products of  $\phi \wedge \psi, \phi \vee \psi, \phi \rightarrow \psi, \neg \phi$  are formulas, too;

*Remark 1.* For brevity, we assume  $\bigwedge L = \bigwedge_{P \in L} P$  (or  $\bigwedge L = 1$  if  $L = \emptyset$ ). Similarly,  $\neg L = \{\neg P \mid P \in L\}$ .

For every object  $\{g\}$ , a logic model of the object  $K_g$  is defined. We say that the object  $g$  respects the formula  $\phi \in \Phi_K$ , if the formula is true for the model  $K_g$ . We will write this fact as  $g \models \phi \Leftrightarrow K_g \models \phi$ . The set  $G_\phi = \{g \in G \mid g \models \phi\}$  is called the support of  $\phi$ .

**Definition 6.** Let us consider an arbitrary probability measure  $\mu$ , i.e.  $\mu$  is a finite countably additive measure on the set  $G$ . Then the contextual probability measure is defined by the following:

$$\nu : \Phi_K \rightarrow [0, 1], \quad \nu(\phi) = \mu(\{g \mid g \models \phi\}).$$

.

The most common understanding of formula probability may be linked with a well-known confidence index for context implications:  $\text{conf}(X \rightarrow Y) = \frac{|\text{supp}(X \cup Y)|}{|\text{supp}(X)|}$ . The formula probability will express exactly the same, if we keep things simple and assume  $\mu$  to be a counting measure:  $\mu(\{g\}) = \frac{1}{|G|}$ .

For practical applications, here and further we will suppose that  $G$  is finite and does not contain any objects of a zero measure, i.e.  $\forall g \in G, \mu(\{g\}) \neq 0$ .

**Definition 7.** The set of attributes  $M$  is compatible, if  $M' \neq \emptyset$ .

The same may be expressed as  $\nu(\bigwedge M) > 0$ .

Now let us consider the set  $L = \{m_i, m\}_{i=1\dots k} \subseteq L_K$ . The formula  $m_1 \wedge m_2 \dots \wedge m_k \rightarrow m$  will look like the classical context implication  $(\{m_i\}, \{m\})$ , except when it is possible to include negations of attributes, like in this one:  $m_1 \wedge \neg m_2 \dots \wedge m_k \rightarrow \neg m$ . The concept of the implication as a formula is reified in definition of the rule:

**Definition 8.** Let  $C, H_i \in L_K$ ,  $C \notin \{H_1, H_2, \dots, H_k\}$ ,  $k \geq 0$ . Then:

1. The **rule**  $R = (H_1, H_2, \dots, H_k \rightarrow C)$  is an implication  $(H_1 \wedge H_2 \dots \wedge H_k \rightarrow C)$ ;
2. The **premise**  $R^\leftarrow$  of a rule  $R$  is a set of letters  $\{H_1, H_2, \dots, H_k\}$ ;
3. The **conclusion** is  $R^\rightarrow = C$ ;
4. If  $R_1^\leftarrow = R_2^\leftarrow$  and  $R_1^\rightarrow = R_2^\rightarrow$ , then  $R_1 = R_2$ .

**Definition 9.** The probability of the rule  $R$  is a conditional probability

$$\eta(R) = \nu(R^\rightarrow \mid R^\leftarrow) = \frac{\nu(R^\leftarrow \wedge R^\rightarrow)}{\nu(R^\leftarrow)}$$

If  $\nu(R^\leftarrow)$  is zero, the probability of the rule remains undefined.

Keeping eye on  $K_0$ , let us try to watch what is happening on  $K_{noise}$  with the implications  $m_1 \rightarrow m_3$  and  $m_2 \rightarrow m_3$ . They stopped to be contextual tautologies, but we still can stick to the corresponding rules with reasonable likelihoods:  $\eta(m_1 \rightarrow m_2) = \frac{4}{5}$  and  $\eta(m_3 \rightarrow m_2) = \frac{5}{6}$ .

The core idea of the approach is to exploit Theorem 1. An operator of a direct inference could be easily adapted to employing probabilistic rules in contrast to formal context implications.

**Definition 10.** The prediction operator  $\Pi$  on the set of the rules  $\mathcal{R}$  works as follows:

$$\Pi_{\mathcal{R}}(L) = L \cup \{C \mid \exists R \in \mathcal{R}: R^\leftarrow \subseteq L, R^\rightarrow = C\}.$$

**Definition 11.** A closure  $\bar{L}$  of the set of the letters  $L$  is the smallest fixed point of the prediction operator:  $\bar{L} = \Pi^\infty(L)$ .

## 4 Rule classes

Note, that the definition 10 accepts any set of rules. To produce a relevant and consistent set of generalized concepts, additional restrictions for this set are needed. Following the [6], we will prove the compatibility theorem and ensure the correctness property for the prediction closure operator.

**Definition 12 (subrule).**  $R_1 \sqsubset R_2$ , if  $R_1^\leftarrow \subset R_2^\leftarrow$  and  $R_1^\rightarrow = R_2^\rightarrow$ .

**Definition 13 (refinement).**  $R_1 > R_2$ , if  $R_2 \sqsubset R_1$  and  $\eta(R_1) > \eta(R_2)$ .

For example, the rule  $m_1 \rightarrow m_2$  from  $K_{noise}$  has the only unconditional subrule:  $(\emptyset \rightarrow m_2) \sqsubset (m_1 \rightarrow m_2)$ . This could not be considered as a refinement relation:  $\eta(\emptyset \rightarrow m_2) = \frac{8}{10} = \frac{4}{5} = \eta(m_1 \rightarrow m_2)$ . However,  $(m_3 \rightarrow m_2) > (\emptyset \rightarrow m_2)$  because  $\eta(\emptyset \rightarrow m_2) = \frac{8}{10} < \frac{5}{6} = \eta(m_3 \rightarrow m_2)$ .

The class  $M_1$  requires that the rules have a greater conditional probability than an unconditional probability of  $C$ , i.e. the rule is guaranteed to be useful in reasoning:

**Definition 14.**  $R \in M_1(C) \Leftrightarrow \eta(R) > \nu(R^\rightarrow), R^\rightarrow = C$ .

The class  $M_2$  requires a rule to be specific – we cannot improve probability by refining the rule:

**Definition 15.**  $R \in M_2(C) \Leftrightarrow R \in M_1(C)$  and  $[R \sqsubset \tilde{R} \Rightarrow \eta(\tilde{R}) \leq \eta(R)]$

The rule  $(m_3 \rightarrow m_2)$  could be refined up to the  $(\neg m_1 \wedge m_3 \rightarrow m_2)$  due to the inequality:  $\eta(m_3 \rightarrow m_2) = \frac{5}{6} < 1 = \frac{3}{3} = \eta(\neg m_1 \wedge m_3 \rightarrow m_2)$ . The last rule satisfies all  $M_2$  conditions, and thus  $(\neg m_1 \wedge m_3 \rightarrow m_2) \in M_2(m_2)$ .

The class  $Imp$  contains all exact implications. So does any contextual tautology:

**Definition 16.**  $R \in Imp(C) \Leftrightarrow R^\rightarrow(R) = C$  and  $\eta(R) = 1$

We also consider compound classes for entire set of letters:

**Definition 17.**  $M_1 = \bigcup_{C \in L_K} M_1(C)$

*Remark 2.*  $M_2$  and  $Imp$  are defined similarly.

All exact implications are indeed necessary to ensure a completeness property for the prediction operator. In turn, a set of rules must consist only from the  $M_2$  rules in order to obtain a consistency property. The set of the letters  $L$  is called consistent, if it does not contain an atom  $C$  and its negation  $\neg C$ .

**Definition 18.** If  $Imp \subset \mathcal{R}$ , then the set of rules  $\mathcal{R}$  is called complete.

**Definition 19.** By a system of the rules, we will call any  $\mathcal{R} \subseteq M_2$ .

## 5 Prediction consistency

**Definition 20.** The set of attributes  $M$  is consistent, if  $L \in M \Rightarrow \neg L \notin M$ .

$\Pi_{\mathcal{R}}$  must avoid inconsistent inferences [3]. The following theorem is the main theoretical result of the paper. It proves predictions to be consistent and compatible (see def. 7). For the proof and technical details, see [6].

**Theorem 2 (Compatibility).** If  $L$  is compatible, then  $\Pi_{\mathcal{R}}(L)$  is also compatible and consistent for any system of the rules  $\mathcal{R}$ .

Somewhat more difficult, but still solvable, is the question of the inconsistency of prediction closures. Let us assume  $\mathcal{R}$  to be a complete set of rules and  $\Pi_{\mathcal{R}}$  to be the corresponding prediction operator. It is important to note that the rule systems containing  $M_2$  are always complete.

**Theorem 3.** If  $L$  is incompatible, then  $\Pi_{\mathcal{R}}(L)$  is inconsistent and incompatible.

## 6 Probabilistic formal concepts

The fixed points of a prediction operator are clear to be the candidates for concept intents. What about concept extents? The principles proposed in [5, 9] give us a cue. The idea is to take all possible closure preimages attribute sets, i.e. all  $M: \Pi(M) = B$ , and compose their derivative sets together into a derived concept extent  $A$ . This will allow restoring the actual concept reference by applying the prediction operator and include all the objects of the same class into a conjoined extent.

For example, let  $K_{squares}$  be a context depicted as two disjoint squares (which are two independent formal concepts). To bring extra complexity, we also alter some entries:

G	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$	$m_7$	$m_8$
$g_1$	1	1	1	1	1	0	0	0
$g_2$	1	1	1	1	0	0	0	0
$g_3$	1	1	1	1	0	0	0	0
$g_4$	1	1	1	1	0	0	0	0
$g_5$	0	0	0	0	1	1	1	1
$g_6$	0	0	0	0	1	1	0	1
$g_7$	0	0	0	0	1	1	1	1
$g_8$	0	0	0	0	1	1	1	1

Table 4: A two-concept context  $K_{squares}$  with a minor noise

Note that the most specific rules referring to  $M_2$  are  $(m_{i=1\dots4} \rightarrow \neg m_{j=5\dots8})$ , however rule  $m_5 \rightarrow \neg m_{i=1\dots4}$  is not. There is a more specific rule for the last one:  $\eta(m_5 \wedge m_6 \rightarrow \neg m_1) = 1 > \frac{4}{5} = \eta(m_5 \wedge \neg m_1)$ . This is how noise is handled being encapsulated in probability and refinement.

To pick up the first object from  $K_{squares}$ , firstly, the prediction operator computes a closure:  $\Pi(g'_1) = \{m_1, m_2, m_3, m_4, \neg m_5, \neg m_6, \neg m_7, \neg m_8\}$ . And secondly, all objects with the same closure are composed into a concept with the extent  $\{g_1, g_2, g_3, g_4\}$ .

**Definition 21.** *By a probabilistic formal concept on  $K = (G, M, I)$  we mean any pair  $(A, B)$  which satisfies*

$$\Pi(B) = B, A = \bigcup_{C \subseteq B, \Pi(C)=B} G_C$$

Our selection is also justified by the following statement, relating probabilistic and ordinary formal concepts on the same context.

**Theorem 4 (Ordinary concepts inclusion [12]).** *Let  $K$  be a formal context.*

1. *If  $(A, B)$  is an ordinary concept on  $K$ , then there is a probabilistic concept  $(N, M)$  such that  $A \subseteq N$ , and  $B \subseteq M$ .*

2. If  $(N, M)$  is a probabilistic concept on  $K$ , then there is a set of ordinary concepts  $\mathcal{C}$ , such that

$$\forall (A, B) \in \mathcal{C} \ (\Pi(B) = M),$$

$$N = \bigcup_{(A, B) \in \mathcal{C}} A.$$

## 7 Probabilistic concepts discovery

For practical applications, a computational problem should be solved. It is still exponentially hard if we require a full  $M_2$  set enumeration.

A semantic probabilistic inference as an enumeration procedure has been described in details in [15]. The idea is to perform a kind of a greedy search combined with a branches and boundaries search on the inference tree. The last aims to obtain an  $M_2$  subset, which will be enough for the most practical tasks.

**Definition 22.**  $R$  is a probabilistic law, if for any  $\tilde{R}$ ,  $(\tilde{R} \sqsubset R) \Rightarrow (\tilde{R} < R)$ .

**Definition 23.** The rule  $\tilde{R}$  is semantically probabilistic inferred from the rule  $R$ . We write  $R \triangleright \tilde{R}$ , if  $R, \tilde{R}$  are the probabilistic laws, and  $\tilde{R} > R$ .

**Definition 24.** The probabilistic law  $R$  is the strongest, or  $R \in \text{SPL}$ , if there is no other probabilistic law  $\tilde{R}$  such that  $(\tilde{R} > R)$ .

**Proposition 1.** All strongest probabilistic laws are in  $M_2$ .

The rules extraction routine is based on exploiting a Semantic Probabilistic Inference (SPI) approach. It requires each path in the inference graph to be a sequence of semantic inferences:

**Definition 25.** SPI is a sequence of the rules  $R_0 \triangleright R_1 \triangleright R_2 \dots \triangleright R_m$ , such that  $R_0^\leftarrow = \emptyset$  and  $R_m$  is the strongest probabilistic law.

Now let us assume that some system of the rules  $\mathcal{R}$  on a context  $K$  has already been discovered by semantic probabilistic inference. The probabilistic concept definition implies the following **closure-search procedure**.

1. Set the step counter  $k = 1$  and generate the set  $C^{(0)} = \{\Pi_{\mathcal{R}}(R^\leftarrow) \mid R \in \mathcal{R}\}$ . In fact, this may be an arbitrary family of letter sets to be extended up to their probabilistic concepts closures. The set  $C^{(0)}$  is almost always redundant, but it should be enough to cover all statistically significant attribute sets;
2. On the step  $k > 1$  in case  $C^{(k)} = \emptyset$  the algorithm finishes the execution and outputs a list of detected probability concepts;
3. On the step  $k > 1$  the set  $A = \{g \in G \mid \Pi_{\mathcal{R}}(g' \cap B) = B\}$  is computed for each  $B \in C^{(k)}$ . If  $A \neq \emptyset$ , the pair  $(A, B)$  is added to the list of the found concepts. It corresponds to a join operation on the concept lattice and subsequently climbs to superordinate levels of the lattice;

4. The set  $C^{(k+1)} = \{\Pi_{\mathcal{R}}(B \cup C) \mid B, C \in C^{(k)}\} \setminus C^{(k)}$  is generated. In fact, actual prediction closures are computed on this step;
5. Let  $k := k + 1$  and go to the step 2.

The algorithm could be applied to a context recovery task as well as to a wide variety of data mining problems, such as classification and clusterisation tasks. In the final section we will focus on handling noise in a toy, a rather hard context recovery task.

## 8 An example

m0	m1	m2	m3	m4	m5	m6	m7		m0	m1	m2	m3	m4	m5	m6	m7
0	0	0	0	0	1	1	0		0	0	0	1	1	0	0	0
0	0	0	0	0	1	1	0		0	0	0	1	1	0	0	0
0	0	0	0	0	1	1	0		0	0	0	1	1	0	0	0
0	0	0	0	0	1	1	0		0	0	0	1	1	0	0	0
0	0	0	0	0	1	1	0		0	0	0	1	1	0	0	0
1	1	1	1	0	1	1	0		0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	0		0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	0		0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	0		0	0	1	1	0	0	0	0
1	1	1	1	0	1	1	0		0	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0		0	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0		0	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0		0	0	1	1	0	0	0	0
1	1	1	1	0	0	0	0		0	0	1	1	0	1	1	0
1	1	1	1	0	0	0	0		0	0	1	1	0	1	1	0
1	1	1	1	0	0	1	1		0	0	1	1	0	1	1	0
1	1	1	1	0	0	1	1		0	0	0	0	0	0	0	0
1	1	1	1	0	0	1	1		0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1		0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1		0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1		0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1		0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1		0	0	0	0	0	0	0	0
0	0	0	1	1	0	1	1		0	0	0	0	0	1	1	0
0	0	0	1	1	0	1	1		0	0	0	0	0	1	1	0
0	0	0	1	1	0	1	1		0	0	0	0	0	1	1	0
0	0	0	1	1	0	1	1		0	0	0	0	0	1	1	0
0	0	0	1	1	0	1	1		0	0	0	0	0	1	1	0

Fig. 1: Initial context

Earlier we considered the  $K_{squares}$  context very simple but illustrative. A more sophisticated example should contain more interactions between concepts, both in extent and intent components. Also more noise should be produced.

To measure some performance issues, we will set up several modifications of a single context. Modifications differ at levels of a noise and there may be a number of data duplicates, when producing more data is necessary. An initial context has been composed from rectangle blocks, easy to be recognized as formal concepts (let them be denoted as "solid" concepts).

A set of experiments was based on:

1.  $K_{exp}$  – the initial context, depicted on Fig. 1.
2.  $K_{x3}$  – similar to  $K_{exp}$ , except it contains 3 duplicates of each  $K_{exp}$  object;
3.  $K_{x3.n05} = K_{x3} +$  randomly inflicted binary noise, Bernoulli distributed with  $p = 0.05$
4.  $K_{x3.n04} = K_{x3} +$  noise,  $p = 0.04$
5.  $K_{x3.n03} = K_{x3} +$  noise,  $p = 0.03$

The primary characteristics of the datasets are presented in Table 5.

Context	$ G $	$ M $	# Concepts	# Solid	# Logical	Noise
$K_{exp}$	61	8	5 + 4 + 2	5	6 + 4	0
$K_{x3}$	183	8	5 + 4 + 2	5	6 + 4	0
$K_{x3.n05}$	183	8	5 + 4 + 2	5	6 + 4	0.05
$K_{x3.n04}$	183	8	5 + 4 + 2	5	6 + 4	0.04
$K_{x3.n03}$	183	8	5 + 4 + 2	5	6 + 4	0.03

Table 5: Experimental data summary:  $|G|$  and  $|M|$  stay for the amount of objects and attributes in a context. The number of concepts is a sum of three different types of concepts: solid concepts, which are indicated on Fig. 1 as solid sequences of ones; join-concepts, which are made of two sequences; and meet-concepts, which are produced by an intersection of two solid concepts. Note that a logical approach is a little bit specific: the model excludes meet-concepts from the consideration and accepts attributes negations, while respecting the empty concept. Thus, the column logical sums solid and join-concepts add an empty one to solid.

The first stage in executing a **closure-search procedure** is a rules extraction routine. According to the method discussed in Section 7, a computer program was implemented to perform a semantical probabilistic inference. For each context a set of rules has been obtained and has eventually been used in a **closure-search procedure**.

Context	# Rules	# Rules ( $p > 0.5$ )	# Rules ( $p > 0.9$ )
$K_{x3.n05}$	1712	1358	682
$K_{x3.n04}$	1193	967	497
$K_{x3.n03}$	1103	916	532

Table 6: Rules extraction routine summary: # Rules are a total number of discovered rules via semantic probabilistic inference. The next two values are the numbers of the rules with conditional probability thresholds of 0.5 and 0.9.

While increasing a noise level, a context becomes less and less clear and requires more and more rules for describing attribute associations. A minor noise produces an insignificant effect and affords to solve the problem almost exactly.

Following [10], we will compare an original concept lattice  $O$  with a predicted one  $E$  and measure the method performance by calculating two ratios:

$$Precision = \frac{|O \cap E|}{|E|}, \quad Recall = \frac{|O \cap E|}{|O|}$$

The experiment results are presented in Table 7. In addition to the performance indexes, the data are presented separately for the solid concepts and the join-concepts.

Context	$ O $	$ O \cap E $	$ O \setminus E $	$ E \setminus O $	Precision	Recall
$K_{x3}$	6 + 4	6 + 4	0 + 0	0 + 0	1.0 + 1.0	1.0 + 1.0
$K_{x3.n03}$	6 + 4	6 + 4	0 + 0	0 + 0	1.0 + 1.0	1.0 + 1.0
$K_{x3.n04}$	6 + 4	6 + 1	0 + 3	0 + 0	1.0 + 1.0	1.0 + 0.25
$K_{x3.n05}$	6 + 4	6 + 3	0 + 1	0 + 1	1.0 + 0.75	1.0 + 0.75

Table 7: Concepts recovery summary on the same contexts:  $|O|$  is the total number of expected concepts;  $|O \cap E|$  is the number of correctly predicted concepts;  $|O \setminus E|$  is the number of lost concepts; and  $|E \setminus O|$  is the number of incorrectly predicted concepts.

It was rather easy for a **closure-search procedure** to determine all formal concepts without noise.

However, even on noisy contexts the algorithm has been able to restore the original set of concepts with a moderate accuracy. All probabilistic concepts encountered by the algorithm may be essentially associated with the original images in ordinary concepts, while some non-primal concepts have been leaked. Nevertheless, it seems that probabilistic formal concepts perform more accurately, in comparison with a stability approach [4].

Indeed, the main advantage may not even be the method accuracy: probabilistic formal concepts are able to discover concepts on big data frames. The estimated computational complexity for SPI is  $|M|^{d+2} * |G|$ , and one for a closure-search seems to be  $|I| * |M|^c * |G|$ , where  $3 < c < 4$  (the estimation is empirical and still needs to be checked). Noise induces extra complexity but using a Pentium 4 2-core 2.4GHz computer is enough to solve a 321x26 context with 10% noise in about 10 minutes, while it takes 5 minutes to complete a 3% noise task.

## 9 Conclusion

The introduced method has been experimentally and theoretically proven to be correct and accurate. Some extra experiments have been proposed in earlier works [12, 20]. Probabilistic formal concepts are also very profitable as they may serve to construct exact concept lattices from real, noisy raw data immediately instead of performing a filtering on a overpopulated concept lattices, possibly exponentially sized. The further work includes theoretical evaluation for computational complexity as well as more sophisticated experiments on a big dataset. We are also planning to compare our results with some famous classification methods in terms of prediction accuracy and speed.

## References

1. Ganter, B.: Formal Concept Analysis: Methods, and Applications in Computer Science. TU Dresden (2003)
2. Ganter, B., Wille, R.: Formal concept analysis – Mathematical Foundations. Berlin-Heidelberg-New York, Springer (1999)
3. Carl G. Hempel: Inductive Inconsistencies. *Synthese*, 12: 439-469 (1960)
4. Kuznetsov, S. O., Makhhalova, T. P.: Concept interestingness measures: a comparative study. Proceedings of the Twelfth International Conference on Concept Lattices and Their Applications Clermont-Ferrand, France, October 13-16, 2015 Vol. 1466. CEUR Workshop Proceedings, 59-72 (2015)
5. Kuznetsov, S.O.: On Stability of a Formal Concept. *Annals of Mathematics and Artificial Intelligence*. 49, 101-115 (2007)
6. Vityaev, E.E., Martynovich, V.V.: Probabilistic Formal Concepts with Negation. *Perspectives of System Informatics*. A. Voronkov, I. Virbitskaite (Eds.), LNCS vol. 8974, pp. 385-399 (2015)
7. Prokashcheva O., Onishchenko A., Gurov S., “Classification based on formal concept analysis and biclustering: Possibilities of the approach”, *Computational mathematics and modeling*, **23(3)** (2012)
8. Emilion R., Levy G.: Size of random Galois lattices. *Discrete Applied Math. J.* 157, 2945-2957 (2009)
9. Buzmakov, A., Kuznetsov, S.O., Napoli A.: Concept Stability as a Tool for Pattern Selection. CEUR Workshop proceedings, vol. 1257, ECAI 2014, pp. 51-58 (2014)
10. L. Piskova, S. Pero, T. Horvath, S. Krajci: Mining Concepts from Incomplete Datasets Utilizing Matrix Factorization. Szathmary L., Priss U. (Eds.), CEUR Workshop proceedings, vol. 972, Proc. CLA 2012, pp. 33-44 (2012)
11. Kashnitsky, Y., Ignatov, D.I.: Can FCA-based Recommender System Suggest a Proper Classifier. CEUR Workshop proceedings 1257, ECAI 2014, pp. 17-26 (2014)
12. Vityaev, E.E., Demin, A.V., Ponomaryov, D. K.: Probabilistic Generalization of Formal Concepts. *Programming and Computer Software*. 38(5), 219-230 (2012)
13. Ganter, B., Obiedkov, S.: Implications in Triadic Formal Contexts. TU Dresden, Springer (2004)
14. Missaoui, R., Kwuida, L.: Implications in Triadic Formal Contexts. In: 9th International Conference, ICFCA 2011. Nicosia, Cyprus, Springer (2011)
15. Kovalerchuk, B., Vityaev, E.: Data Mining in Finance: Advances in Relational and Hybrid methods. Kluwer Academic Publishers (2000)
16. Rehder, B.: Categorization as causal reasoning. *Cognitive Science*, Vol. 27(5), pp. 709-748 (2003)
17. Rosch, E., Lloyd, B.B.: Principles of categorization. *Cognition and Categorization*, Lawrence Elbaum Associates (1978)
18. Quan, T.T., Hui, S.C., Cao, T.H.: A Fuzzy FCA-based Approach to Conceptual Clustering for Automatic Generation of Concept Hierarchy on Uncertainty Data. Belohlavek R., Snasel V. (Eds.): Proc. CLA 2004, CEUR Workshop proceedings, vol. 110, Proc. CLA 2004 (2004)
19. Speransky, S.O.: Logic of probability and probability logic. Novosibirsk State University, Novosibirsk, PhD thesis. (2013) (in Russian)
20. Vityaev, E.E., Neupokoev, N.V.: Formal model of perception and pattern as fix point of anticipations. In: Approaches to the thinking modeling, pp.155-172, Moscow, URSS Editorial (2014) (in Russian)