Σοφια: how to make FCA polynomial?

Aleksey Buzmakov[1,2], Sergei Kuznetsov[2], and Amedeo Napoli[1]

[1]LORIA (CNRS – Inria NGE – Université de Lorraine), Vandœuvre-lès-Nancy, France
[2]National Research University Higher School of Economics, Moscow, Russia
aleksey.buzmakov@loria.fr, skuznetsov@hse.ru, amedeo.napoli@loria.fr

Abstract. In pattern mining, one of the most important problems is fighting exponential explosion of the set of patterns. A typical solution is generating only a part of all patterns satisfying some criteria. The most well-known criterion is support of a pattern, which has the monotonicity property allowing one to generate only frequent (highly supported) patterns. Many other useful criteria are not monotonic, which makes it difficult to generate best patterns efficiently. In this paper we introduce the notion of "generalized monotonicity" and Σοφια algorithm that allow to generate top patterns in polynomial time modulo basic operations, e.g., measure computation, for criteria that are not monotonic. This approach is applicable not only to itemsets, but to complex descriptions such as sequences, graphs, numbers or interval tuples, etc. In this paper we consider stability and Δ-measures which are not monotonic. In the experiments, we compute top best patterns w.r.t. these measures and obtain very promising results.

1 Introduction

To solve the problem of exponential explosion of patterns valid in a dataset many kinds of interestingness measures were proposed [1]. For example, pattern support, i.e., the number of objects covered by the pattern, is one of the most well-known measures of pattern quality. Among others stability of a formal concept [2] can be mentioned. Unlike support this measure is not monotonic w.r.t. the order of pattern inclusion and it is hard to generate only most interesting patterns w.r.t. these measures, so one has to find a large set of patterns and then postprocess it, choosing the best ones.

Due to the increasing importance of pattern mining, efficient approaches of finding best patterns are appearing. In [3] authors introduce an approach for efficiently searching the most interesting associations w.r.t. lift or leverage of a pattern. Another approach is searching for cosine interesting patterns [4]. The cosine interestingness of a pattern is not a monotonic measure but the authors take advantage of a conditional anti-monotonic property of cosine interestingness to efficiently mine interesting patterns. However, all of the mentioned approaches are not polynomial in the worst case.

In this paper we introduce a new algorithm Σοφια (Sofia, for Searching for Optimal Formal Intents Algorithm) for extracting top best patterns of different kinds, i.e., itemsets, string, graph patterns, etc. Σοφια algorithm is applicable to a class of measures, including classical monotonic measures, stability, δ-freeness [5], etc. For itemset mining, our algorithm can find top best patterns w.r.t. a measure from this class in polynomial time, modulo complexity of measure computation. For more complex description the time is polynomial modulo complexity of basic operations (intersecting and testing containment on descriptions, computation of a measure).

2 Preliminaries

FCA is a formalism convenient for describing models of itemset mining and knowledge discovery [6]. Here we briefly define pattern structures and the corresponding notations [7]. A *pattern structure* is a triple $\mathbb{P} = (G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a meet-semilattice of descriptions such that $(\forall X \subseteq G) \bigsqcap X \in D$ and $\delta : G \to D$ maps an object to a description. The intersection $\sqcap$ gives similarity of two descriptions.

Let us denote the derivative operators of the Galois connection between 2^G and D by $(\cdot)^\diamond$ (see [7]). A *pattern concept* of a pattern structure $(G, (D, \sqcap), \delta)$ is a pair (A, d), where $A \subseteq G$, called *pattern extent* and $d \in D$, called *pattern intent*, such that $A^\diamond = d$ and $d^\diamond = A$. The set of all pattern concepts is partially ordered w.r.t. inclusion on extents, i.e., $(A_1, d_1) \leq (A_2, d_2)$ iff $A_1 \subseteq A_2$ (or, equivalently, $d_2 \sqsubseteq d_1$), making a lattice, called pattern lattice.

For real datasets, the number of patterns can be large. In order to reduce the most interesting concepts different measures can be used. In this paper we rely on stability [2], which measures the independence of a concept intent w.r.t. randomness in data. Because of limited space we do not discuss this measure in details here. Moreover, since concept stability is hard to compute, we rely on an estimate of concept stability which can be computed in polynomial time for a single concept [8].

The approach proposed in this paper is based on projections introduced for reducing complexity of computing pattern lattices [7]. A *projection operator* $\psi : D \to D$ is an "interior operator", i.e. it is (1) monotonic ($x \sqsubseteq y \Rightarrow \psi(x) \sqsubseteq \psi(y)$), (2) contractive ($\psi(x) \sqsubseteq x$) and (3) idempotent ($\psi(\psi(x)) = \psi(x)$).

An *o-projected pattern structure* (projected pattern structure for simplicity) $\psi((G, (D, \sqcap), \delta))$ is a pattern structure $\psi(\mathbb{P}) = (G, (D_\psi, \sqcap_\psi), \psi \circ \delta)$, where $D_\psi = \psi(D) = \{d \in D \mid \exists \tilde{d} \in D : \psi(\tilde{d}) = d\}$ and $\forall x, y \in D, x \sqcap_\psi y := \psi(x \sqcap y)$ [9]. Given a projection ψ we say that the fixed set of ψ is the set of all elements from D which are mapped to themselves by ψ. The fixed set of ψ is denoted by $\psi(D) = \{d \in D \mid \psi(d) = d\}$. Any element outside of the fixed set of ψ is pruned from the description space. We say that a projection ψ_1 is *s*impler than a projection ψ_2, denoted by $\psi_1 < \psi_2$, if $\psi_1(D) \subset \psi_2(D)$, i.e., ψ_2 prunes less descriptions than ψ_1.

Our algorithm is based on this order on projections. The simpler a projection ψ is, the less patterns we can find in $\psi(\mathbb{P})$, and the less computational efforts one should take. Thus, we compute a set of patterns for a simpler projection, then we remove unpromising patterns and extend our pattern structure and the found patterns to a more detailed projection. This allows to reduce the size of patterns within a simpler projection in order to reduce the computational complexity of more detailed projection.

3 Σοφια Algorithm

3.1 Monotonicity w.r.t. a Projection

Our algorithm is based on the projection monotonicity, a new idea introduced in this paper. Many interestingness measures for patterns, e.g., stability, are not monotonic w.r.t. subsumption order on patterns, i.e., given patterns X and Y such that $X \sqsubseteq Y$, and a nonmonotonic measure $\mathcal{M}$, one does not necessarily have $\mathcal{M}(X) \geq \mathcal{M}(Y)$. For instance, support is a monotonic measure w.r.t. pattern order and it allows for efficient generation of patterns with support higher than a threshold [10]. The projection monotonicity is a generalization of standard monotonicity and allows for efficient work with a wider set of interestingness measures.

Definition 1. *Given a pattern structure* $\mathbb{P}$ *and a projection* ψ*, a measure* $\mathcal{M}$ *is called* monotonic w.r.t. the projection ψ*, if*

$$(\forall p \in \psi(\mathbb{P}))(\forall q \in \mathbb{P}, \psi(q) = p)\mathcal{M}_\psi(p) \geq \mathcal{M}(q), \tag{1}$$

where $\mathcal{M}_\psi(p)$ *is the measure* $\mathcal{M}$ *of pattern* p *computed in* $\psi(\mathbb{P})$*.*

Here, for any pattern p of a projected pattern structure we check that a preimage q of p for ψ, e.g. $p = \psi(q)$, has a measure smaller than the measure of p. It should be noticed that a measure $\mathcal{M}$ for a pattern p can yield different values if $\mathcal{M}$ is computed in $\mathbb{P}$ or in $\psi(\mathbb{P})$. Thus we use the notation $\mathcal{M}_\psi$ for the measure $\mathcal{M}$ computed in $\psi(\mathbb{P})$.

An important example is given by binary data or formal contexts (G, M, I). In this case, a projection ψ_m corresponds to the removal of an attribute $m \in M$, i.e., $\psi_m(B) = B \cap (M \setminus \{m\})$ for any $B \subseteq M$. So Definition 1 means that the interestingness of an itemset p w.r.t. a measure $\mathcal{M}$ computed in $(G, M \setminus \{m\}, I \setminus G \times \{m\})$ should be higher than the interestingness of the itemsets p and $p \cup \{m\}$ (the preimages of p for ψ_m) w.r.t. the measure $\mathcal{M}$ computed in (G, M, I). If the value of a measure for a pattern does not depend on a projection this definition is related to a classical monotonic measure. Indeed, because of contractivity of ψ ($\psi(p) \sqsubseteq p$), for any monotonic measure one has $\mathcal{M}(\psi(p)) \geq \mathcal{M}(p)$.

Thus, given a measure $\mathcal{M}$ monotonic w.r.t. a projection ψ, if p is a pattern such that $\mathcal{M}_\psi(p) < \theta$, then $\mathcal{M}(q) < \theta$ for any preimage q of p for ψ. Hence, if, given a pattern p of $\psi(\mathbb{P})$, one can find all patterns q of $\mathbb{P}$ such that $\psi(q) = p$, it is possible to find the patterns of $\psi(\mathbb{P})$ and then to filter them w.r.t. $\mathcal{M}_\psi$ and a threshold, and finally to compute the preimages of filtered patterns.

3.2 Monotonicity w.r.t. a Chain of Projections

However, given just one projection, it can be hard to efficiently discover the patterns, because the projection is either hard to compute or the number of unpromising patterns that can be pruned is not high. Hence we introduce *a chain of projections* $\psi_0 < \psi_1 < \cdots < \psi_k = \mathbb{1}$, where the whole pattern lattice for $\psi_0(\mathbb{P})$ can be easily computed and $\mathbb{1}$ is the identity projection, i.e., $(\forall x)\mathtt{1}(x) = x$. For example, to find frequent itemsets, we typically search for small itemsets and, then, extend them to larger ones. It corresponds to extension to a more detailed projection.

Let us discuss what is a chain of projections in the case of a binary context $\mathbb{K} = (G, M, I)$ with $M = \{m_1, \cdots, m_N\}$. It can be seen that any subcontext $\mathbb{K}_s = (G, N, I \cap G \times N)$, where $N \subseteq M$, corresponds to a projection ψ such that $\psi(B \subseteq M) = B \cap N$. If we put $M_i = \{m_1, \cdots, m_i\}$, then we can consider a chain of projections corresponding to the subset of attributes $M_1, M_2, \cdots, M$. The corresponding projections are properly ordered. Now we define the projection monotonicity of $\mathcal{M}$ w.r.t. a chain of projections.

Definition 2. *Given a pattern structure $\mathbb{P}$ and a chain of projections $\psi_0 < \psi_1 < \cdots < \psi_k = \mathbb{1}$, a measure $\mathcal{M}$ is called* monotonic w.r.t. the chain of projections *if $\mathcal{M}$ is monotonic w.r.t. all ψ_i for $0 \leq i \leq k$.*

3.3 Algorithms

Given a measure monotonic w.r.t. a chain of projections, if we are able to find all preimages of any element in the fixed set of ψ_i that belong to a fixed set of ψ_{i+1}, then we can find all patterns of $\mathbb{P}$ with a value of $\mathcal{M}$ higher than a given threshold θ. We call this algorithm ϑ-$\Sigma o\varphi\iota\alpha$ and its pseudocode is given in Fig. 1. In lines 11-12 we find all patterns for $\psi_0(\mathbb{P})$ satisfying the constraint that a value of $\mathcal{M}$ is higher than a threshold. Then in lines 13-15 we iteratively extend projections from smaller to bigger ones. The extension is done by constructing the set $\mathcal{P}_i$ of preimages of the set $\mathcal{P}_{i-1}$ (lines 2-5) and then by removing the patterns that do not satisfy the constraint from the set $\mathcal{P}_i$ (lines 6-9).

The algorithm is sound and complete, because first, when we compute the set of preimages of a pattern p, the pattern p is a preimage of itself ($\psi(p) = p$) and second, if we remove a pattern p from the set $\mathcal{P}$, then the value $\mathcal{M}(p) < \theta$ and, hence, the measure value of any preimage of p is less than θ by the projection chain monotonicity of $\mathcal{M}$.

The worst-case time complexity of ϑ-$\Sigma o\varphi\iota\alpha$ algorithm is

$$\begin{aligned}\mathbb{T}(\vartheta\text{-}\Sigma o\varphi\iota\alpha) = {} & \mathbb{T}(FindPatterns(\psi_0)) + \\ & + k \cdot \max_{0 < i \leq k} |\mathcal{P}_i| \cdot (\mathbb{T}(Preimages) + \mathbb{T}(\mathcal{M})),\end{aligned} \tag{2}$$

where $\mathbb{T}(X)$ is time for computing operation X. Since projection ψ_0 can be chosen to be very simple, in a typical case the complexity of $FindPatterns(\theta, \psi_0)$ can be low or even constant. The complexities of $Preimages$ and $\mathcal{M}$ depend on the

```
    Data: A pattern structure $\mathbb{P}$, a chain of projections $\Psi = \{\psi_0, \psi_1, \cdots, \psi_k\}$, a
          measure $\mathcal{M}$ monotonic for the chain $\Psi$, and a threshold $\theta$ for $\mathcal{M}$.
 1  Function ExtendProjection($i$, $\theta$, $\mathcal{P}_{i-1}$)
        Data: $i$ is the projection number to which we should extend ($0 < i \leq k$), $\theta$
              is a threshold value for $\mathcal{M}$, and $\mathcal{P}_{i-1}$ is the set of patterns for the
              projection $\psi_{i-1}$.
        Result: The set $\mathcal{P}_i$ of all patterns with the value of measure $\mathcal{M}$ higher
                than the threshold $\theta$ for $\psi_i$.
 2      $\mathcal{P}_i \longleftarrow \emptyset$;
 3      /* Put all preimages in $\psi_i(\mathbb{P})$ for any pattern $p$                    */
 4      foreach $p \in \mathcal{P}_{i-1}$ do
 5          $\mathcal{P}_i \longleftarrow \mathcal{P}_i \cup$ Preimages($i$,$p$)
 6      /* Filter patterns in $\mathcal{P}_i$ to have a value of $\mathcal{M}$ higher than $\theta$   */
 7      foreach $p \in \mathcal{P}_i$ do
 8          if $\mathcal{M}_{\psi_i}(p) \leq \theta$ then
 9              $\mathcal{P}_i \longleftarrow \mathcal{P}_i \setminus \{p\}$
10  Function Algorithm_θ-Σοφια
        Result: The set $\mathcal{P}$ of all patterns with a value of $\mathcal{M}$ higher than the
                threshold $\theta$ for $\mathbb{P}$.
11      /* Find all patterns in $\psi_0(\mathbb{P})$ with a value of $\mathcal{M}$ higher than $\theta$ */
12      $\mathcal{P} \longleftarrow$ FindPatterns($\theta, \psi_0$);
13      /* Run through out the chain $\Psi$ and find the result patterns   */
14      foreach $0 < i \leq k$ do
15          $\mathcal{P} \longleftarrow$ ExtendProjection($i, \theta, \mathcal{P}$);
```

Algorithm 1: The ϑ-Σοφια algorithm for finding patterns in $\mathbb{P}$ with a value of a measure $\mathcal{M}$ higher than a threshold θ.

measure in use and on the instantiation of the algorithm. In many cases $\max\limits_{0<i\leq k} |\mathcal{P}_i|$ can be exponential in the size of the input, because the number of patterns can be exponential. It can be a difficult task to define the threshold θ *a priori* such that the maximal cardinality of $\mathcal{P}_i$ is not higher than a given number. Thus, we introduce Σοφια algorithm, which automatically adjusts threshold θ ensuring that $\max\limits_{0<i\leq k} |\mathcal{P}_i| < L$. Here L can be considered as a constraint on the memory used by the algorithm. It can be seen from Eq. (2) that Σοφια algorithm has polynomial time complexity if $\mathcal{M}$ and *Preimages* are polynomial. In the next subsection we consider an important partial case where Σοφια has polynomial complexity.

3.4 Σοφια Algorithm for Binary Data

In this subsection we have a formal context $\mathbb{K} = (G, M, I)$ with $M = \{m_1, \cdots, m_N\}$ and we want to find itemsets $X \subseteq M$ interesting w.r.t. a measure $\mathcal{M}$. First, we instantiate a chain of projections. In the case of binary data it corresponds to the chain of contexts $\mathbb{K}_i = (G, M_i, I \cap G \times M_i)$, where $M_i = \{m_1, \cdots, m_i\}$, i.e., M_i contains the first i attributes from M. It means that $\psi_i(X) = X \cap M_i$.

Then we define how the function *Preimages* works for this kind of chains of projections. A set $X \subseteq M_{i-1}$ has two preimages in the powerset of M_i, i.e. X and $X \cup \{m_i\}$. Hence, the computation complexity of finding preimages for any itemset X is constant. For the projection ψ_0 corresponding to the context $(G, \emptyset, \emptyset)$ there is only one itemset $\emptyset$. Thus, the worst case complexity for ϑ-$\Sigma o\varphi\iota\alpha$ algorithm is

$$\mathbb{T}(\vartheta\text{-}\Sigma o\varphi\iota\alpha_{\text{binary}}) = |M| \cdot \max_{0<i\leq N} |\mathcal{P}_i| \cdot \mathbb{T}(\mathcal{M}). \tag{3}$$

In particular, the complexity of $\Sigma o\varphi\iota\alpha$ for binary data is $|M| \cdot L \cdot \mathbb{T}(\mathcal{M})$, i.e., it is polynomial modulo complexity of the measure.

3.5 $\Sigma o\varphi\iota\alpha$ Algorithm for Closed Patterns

Closed frequent itemsets are widely used as a condensed representation of all frequent itemsets since [10]. Here we show how one can adapt our algorithm for closed patterns. A closed pattern in $\psi_{i-1}(\mathbb{P})$ is not necessarily closed in $\psi_i(\mathbb{P})$. However, the extents of $\psi(\mathbb{P})$ are extents of $\mathbb{P}$ [7]. Thus, we associate the closed patterns with extents, and then work with extents instead of patterns, i.e., a pattern structure $\mathbb{P} = (G, (D, \sqcap), \delta)$ is transformed into $\mathbb{P}_C = (G, (D_C, \sqcap_C), \delta_C)$, where $D_C = 2^G$. Moreover, for all $x, y \in D_C$ we have $x \sqcap_C y = (x^\diamond \sqcap y^\diamond)^\diamond$, where diamond operator is computed in $\mathbb{P}$ and $\delta_C(g \in G) = \{g\}$. Hence, every pattern p in D_C corresponds to a closed pattern $p^\diamond$ in D.

A projection ψ of $\mathbb{P}$ induces a projection ψ_C of $\mathbb{P}_C$, given by $\psi_C(X \subseteq G) = \psi(X^\diamond)^\diamond$, where diamond is computed in $\mathbb{P}$. The function ψ_C is a projection because of the properties of $(\cdot)^\diamond$ operators and ψ mappings. We use this approach for representing closed patterns in our computer experiments.

4 Experiments and Discussion

Datasets	Decreasing order									Increasing order									Random order								
	$L = 10^3$			$L = 10^4$			$L = 10^5$			$L = 10^3$			$L = 10^4$			$L = 10^5$			$L = 10^3$			$L = 10^4$			$L = 10^5$		
	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ	t	#	θ
Mushrooms	< 1	0.99	181	2	0.87	49	39	0.89	7	1	0.99	181	6	0.87	49	38	0.89	7	< 1	0.99	181	3	0.87	49	117	0.89	7
Chess	< 1	0.997	97	2	0.92	69	17	0.94	46	1	0.88	144	4	0.24	84	38	0.68	49	< 1	0.65	103	2	0.92	69	19	0.94	46
Plants	1	1	147	14	0.96	70	146	0.94	37	3	1	147	29	0.96	70	263	0.94	37	1	1	147	14	0.96	70	143	0.94	37
Cars	< 1	0.94	19	< 1	0.61	0	< 1	0.06	0	< 1	0.86	22	< 1	0.61	0	< 1	0.06	0	< 1	0.94	19	< 1	0.6	0	< 1	0.06	0

Table 1: Evaluation results of $\Sigma o\varphi\iota\alpha$ algorithm for Δ-measure.

In the experiment we show how our algorithm in conjunction with stability estimate behaves on different datasets from UCI repository [11]. Here we should note that stability and its estimate is monotone w.r.t. any projection [12] and, thus, we can combine it with $\Sigma o\varphi\iota\alpha$. The datasets `Mushrooms`[1] and `Cars`[2] are datasets having a relatively small number of closed patterns, which can be found in some seconds, while the datasets `Chess`[3] and `Plants`[4] have a lot of closed patterns, which can be hardly found.

There are two obvious orders for adding an attribute in $\Sigma o\varphi\iota\alpha$ algorithm: the decreasing and increasing orders of attribute support. We consider also a random order of attributes allowing one to discard any bias in the order of attributes. Another point about our algorithm is that it does not ensure finding L-top best patterns. It finds no more than L patterns allowing to compute the result in polynomial time by adjusting the threshold θ of stable patterns.

Thus, in our experiment we have checked which order is better for the attributes and how many patterns we can find for a given L. Table 1 shows the results and is divided into three parts corresponding to the order in which attributes were added to the context. Then all parts are divided into three subparts corresponding to a value of $L \in \{10^3, 10^4, 10^5\}$. Hence, we have 9 experiments and for every experiment we measure the computation time in seconds (t), the ratio of found patterns to L (#) and the final θ corresponding to the found patterns. For example, in the `Mushrooms` dataset, adding the attributes in the decreasing order of their support for $L = 10000$, the total computational time is equal to 2 seconds; the algorithm found around $0.87 * L = 8700$ patterns representing all patterns with stability higher than 49.

In Table 1 we can see that our algorithm is efficient in the big and small datasets however the computational time and the number of found patterns depend on the order of attribute addition, i.e., on a projection chain. We can see that the computational time and the number of patterns for increasing order are never better than those of decreasing order and random order. Decreasing order and random order have nearly the same behavior, but in some cases the random order gives slightly worse results than the decreasing order. In fact, in the case of decreasing order we generate more patterns on earlier iterations of our algorithm, i.e., we have more chances to find an unstable pattern and filter it as earlier as possible. Since concepts are filtered earlier, we have more space for the computation, thus having smaller threshold θ and larger number of found patterns, and we should process less patterns, thus saving the computation time. We see that for the decreasing order of attributes the number of found patterns is always around or higher than $0.9 * L$, i.e., we find nearly as many patterns as the requested limit L.

[1] `https://archive.ics.uci.edu/ml/datasets/Mushroom`
[2] `https://archive.ics.uci.edu/ml/datasets/Car+Evaluation`
[3] `https://archive.ics.uci.edu/ml/datasets/Chess+(King-Rook+vs.+King-Knight)`
[4] `https://archive.ics.uci.edu/ml/datasets/Mushroom`

5 Conclusion

In this paper we have introduced a new kind of interestingness measures of patterns monotonic w.r.t. a chain of projections. Based on this monotonicity we introduce a new algorithm called $\Sigma o\varphi\iota\alpha$ that finds the top best patterns for such kind of measures in polynomial time. Our experiments justified the efficiency of our algorithms. Many directions for future work are promising. First, we should work on adaptation of $\Sigma o\varphi\iota\alpha$ for finding different kinds of patterns, e.g., itemset generators, sequences, graphs. Second, we should study the best chains of projections and the best order of attributes for $\Sigma o\varphi\iota\alpha$ algorithm. Finally, the study of new measures that can be used with $\Sigma o\varphi\iota\alpha$ is also very important.

References

1. Vreeken, J., Tatti, N.: Interesting Patterns. In Aggarwal, C.C., Han, J., eds.: Freq. Pattern Min. Springer International Publishing (2014) 105–134
2. Kuznetsov, S.O.: On stability of a formal concept. Ann. Math. Artif. Intell. **49**(1-4) (2007) 101–115
3. Webb, G.I., Vreeken, J.: Efficient Discovery of the Most Interesting Associations. ACM Trans. Knowl. Discov. from Data **8**(3) (2014) 15
4. Cao, J., Wu, Z., Wu, J.: Scaling up cosine interesting pattern discovery: A depth-first method. Inf. Sci. (Ny). **266**(0) (2014) 31–46
5. Hébert, C., Crémilleux, B.: Mining Frequent δ-Free Patterns in Large Databases. In Hoffmann, A., Motoda, H., Scheffer, T., eds.: Discov. Sci. Volume 3735 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2005) 124–136
6. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. 1st edn. Springer (1999)
7. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In Delugach, H.S., Stumme, G., eds.: Concept. Struct. Broadening Base. Volume 2120 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2001) 129–142
8. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Scalable Estimates of Concept Stability. In Sacarea, C., Glodeanu, C.V., Kaytoue, M., eds.: Form. Concept Anal. Volume 8478 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) 161–176
9. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Revisiting Pattern Structure Projections. In Baixeries, J., Sacarea, C., Ojeda-Aciego, M., eds.: Form. Concept Anal. Volume 9113 of LNAI 9113. Springer International Publishing (2015) 200–215
10. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Mining of Association Rules Using Closed Itemset Lattices. Inf. Syst. **24**(1) (1999) 25–46
11. Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences (2010)
12. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On Mining Complex Sequential Data by Means of FCA and Pattern Structures. Int. J. Gen. Syst. (2016) IN PRESS