Closure Structure: a Deeper Insight

Tatiana Makhalova[1], Sergei O. Kuznetsov[2], and Amedeo Napoli[1]

[1] Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France
[2] National Research University Higher School of Economics, Moscow, Russia
tatiana.makhalova@inria.fr, skuznetsov@hse.ru, amedeo.napoli@loria.fr

Abstract. Discovery of the pattern search space is an essential component of Itemset Mining. The most common approach to reduce pattern search space is to compute only frequent closed itemsets. Frequent patterns are known to be not a good choice due to omitting useful infrequent itemsets and their exponential explosion with decreasing frequency. In our previous work we proposed the closure structure that allows for computing itemsets level-by-level without any preset parameters. In this work we study experimentally some properties of the closure levels.

1 Introduction

Itemset Mining (IM) envelops a wide variety of tasks and methods related to computing and selecting itemsets. Its main challenges can be summarized in two questions: What itemsets (and how) to compute? Which of them (and how) to use? IM is usually considered as unsupervised learning, meaning that one selects itemsets based on such characteristics as coverage, diversity, interestingness by a certain measure [9], etc. However, IM also includes some supervised approaches, e.g., rule-based classifiers, where itemsets are selected based on a standard quality measure of classifiers. However, both supervised and unsupervised approaches may use the same methods for itemset computing. To date, frequency remains the major criterion for computing itemsets. The methods for computing frequent itemsets are brought together under the name Frequent Itemset Mining (FIM). The main drawback of FIM is omitting interesting and useful infrequent itemsets, while the main advantage of FIM is efficiency in the sense that any FIM-approach computes frequent itemsets and only them (because of the antimonotonicity of frequency w.r.t. the order of pattern inclusion).

Nowadays there are almost no other (anti-)monotone measures that are commonly used in IM for computing itemsets. In [4] authors propose to generate closed itemsets based on Δ-measure, which is monotone w.r.t. projections. Authors propose an efficient polynomial algorithm, however, the lack of experiential study of the quality of generated itemsets may hamper a wide use of this approach in practice.

In our previous work [11], we proposed the closure structure of concept lattice (i.e., the whole set of closed itemsets) and an algorithm for its gradual computing. The algorithm computes closed itemsets (formal concepts) by levels with polynomial delay. Each level may contain itemsets of different frequency, however, the number of frequent itemsets decreases with each new level. In [11] we presented some theoretical results and described characteristics of closed itemsets by levels based on experiments. In this work, we study further topology of the closure structure and applicability of concepts for classification by closure levels.

The paper has the following structure. In Section 2 we recall the basic notions. In Section 3 we describe the GDPM algorithm for computing closure levels and discuss its flaws. In Section 4 we present a simple model of a rule-based classifier. Section 5 contains the results of the experiments. In Section 6 we conclude and give directions of future work.

2 Basic notions

2.1 Concepts and the partial order between them

A *formal context* [7] is a triple (G, M, I), where G is called a set of objects, M is called a set of attributes and $I \subseteq G \times M$ is a relation called incidence relation, i.e., $(g, m) \in I$ if object g has attribute m. The derivation operators $(\cdot)'$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

$$A' = \{m \in M \mid \forall g \in A : gIm\}\,,\; B' = \{g \in G \mid \forall m \in B : gIm\}\,.$$

Sets $A \subseteq G$, $B \subseteq M$, such that $A = A''$ and $B = B''$, are said to be closed. For $A \subseteq G$, $B \subseteq M$, a pair (A, B) such that $A'' = B$ and $B'' = A$, is called a *formal concept*, A and B are called *extent* and *intent*, respectively. In Data Mining, an *intent* is also called a *closed itemset (or closed pattern)*.

A partial order $\leq$ is defined on the set of concepts as follows: $(A, B) \leq (C, D)$ iff $A \subseteq C\,(D \subseteq B)$, a pair (A, B) is a subconcept of (C, D), while (C, D) is a superconcept of (A, B). With respect to this partial order, the set of all formal concepts forms a complete lattice $\mathcal{L}$ called the *concept lattice* of the formal context (G, M, I).

2.2 Equivalence classes and key sets

Let B be a closed itemset. Then all subsets $D \subseteq B$, such that $D'' = B$ are called *generators* of B and the set of all generators is called the *equivalence class* of B, denoted by $Equiv(B) = \{D \mid D \subseteq B, D'' = B\}$. A subset $D \in Equiv(B)$ is a *key* [2, 13] or *minimal generator* of B if for every $E \subset D$ one has $E'' \neq D'' = B''$, i.e., every proper subset of a key is a member of the equivalence class of a smaller closed set. We denote a set of keys (*key set*) of B by $K(B)$. The set of keys is an order ideal, i.e., any subset of a key is a key [13]. The *minimum key set* $K^{min}(B) \subseteq K(B)$ is a subset of the key set that contains the keys of the minimum size, i.e., $K^{min}(B) = \{D \mid D \in K(B), |D| = min_{E \in K(B)}|E|\}$. In an equivalence class there can be several keys, but only one closed itemset, which is maximal in this equivalence class. An equivalence class is called trivial if it consists only of a closed itemset.

For the sake of simplicity, we denote attribute sets by strings of characters, e.g., abc instead of $\{a, b, c\}$.

Example. Let us consider a formal context given in Table 1. Five concepts have nontrivial equivalence classes, namely $(\{g_1\}, acf)$, $(\{g_3\}, ade)$, $(\{g_5, g_6\}, bdf)$, $(\{g_5\}, bdef)$ and $(\emptyset, abcdef)$. Among them, only bdf and $abcdef$ have the minimum key sets that differ from the key sets, i.e., $K^{min}(bdf) = \{b\}$, $K(bdf) = \{b, df\}$ and $K^{min}(abcdef) = \{ab\}$, $K(abcdef) = \{ab, adf, aef, cef\}$.

Table 1. Formal context and nontrivial equivalence classes.

	a	b	c	d	e	f
g_1	×		×			×
g_2	×		×		×	
g_3	×			×	×	
g_4			×		×	
g_5		×		×	×	×
g_6		×		×		×

C	$K^{min}(C)$	$K(C)$	$Equiv(C)$
ade	ad	ad	ad, ade
acf	af, cf	af, cf	af, cf, acf
bdf	b	b, df	b, df, bd, bf, bdf
$bdef$	be, ef	be, ef	be, ef, bde, bef, def, $bdef$
$abcdef$	ab	ab, adf, aef, cef	ab, adf, aef, cef, ..., $abcdef^*$

* The equivalence class includes all itemsets that contain a key from $K(abcdef)$.

2.3 Level-wise structure on minimum key sets

In [11] we introduced the *minimum closure structure* induced by minimum key sets. Here we recall the main notions.

Let $\mathcal{C}$ be a set of all closed itemsets and $K^{min}(B)$ be the minimum key set of a closed itemset $B \in \mathcal{C}$. We denote a function that maps a closed itemset to the size of its minimum key by *level*, i.e., $level : \mathcal{C} \to \{0, \ldots, |M|\}$, such that $level(B) = |D|$, where $D \in \mathcal{K}^{min}(B)$ is an arbitrary itemset chosen from $\mathcal{K}^{min}(B)$. The minimal structural level k is given by all minimum keys of size k, i.e., $\mathcal{K}_k^{min} = \bigcup_{\substack{B \in \mathcal{C},\\ level(B)=k}} K^{min}(B)$. We say that B belongs to the minimum structural level k if keys in $K^{min}(B)$ have size k. We denote the corresponding set of closed itemsets of level k by $\mathcal{C}_k^{min}$. More formally, $\mathcal{C}_k^{min} = \{B \mid B \in \mathcal{C}, level(B) = k\}$. We call *minimum structural complexity* of $\mathcal{C}$ the maximal number of not empty levels, $N_c^{min} = \max\{k \mid k = 1, \ldots, |M|, \mathcal{K}_k^{min} \neq \emptyset\}$.

Example. The closure structure of the concept lattice from the running example is given in Fig. 1. It consists of 3 closure levels, minimum structural complexity is equal to 3.

3 The GDPM algorithm and related issues

Efficiency is a principal parameter of the algorithms for computing closed itemsets (concepts). Apart of polynomial delay, we pay attention to other important

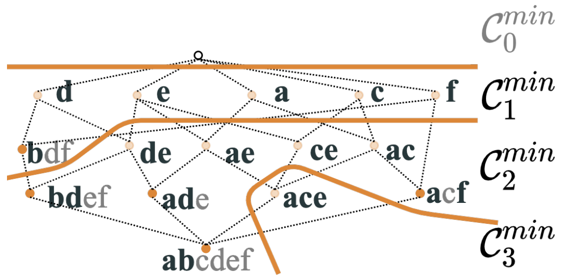

Fig. 1. The closure structure of the concept lattice of the context from Table 1 induced by the sizes of the minimum keys. The smallest lexicographically minimum keys for each concept are highlighted in boldface.

characteristics, namely (1) the strategy of computing a concept given already generated ones, and (2) the test of uniqueness of generated concepts.

Taking into account that the set of minimum keys is an order ideal, we may generate closure levels using a strategy similar to one used in the Titanic and Pascal algorithms, i.e., computing a new minimum key by merging two minimum keys, that differ in one element, from the previous level. This strategy may be non-optimal because (1) for each concept we should keep all its minimum keys, and (2) the time complexity of the key candidate generation is quadratic w.r.t. the size of the last level.

An alternative strategy is to add to each key an attribute that is not included into the corresponding intent. For this strategy it is important to use an efficient procedure for verification whether a concept is generated for the first time. We cannot use the canonicity test as it is done, for example, in CbO because a key of a concept may be lexicographically less than its minimum key.

The simplest solution to ensure the presence of only one minimum key for each concept is to use a lexicographic tree that contains all previously generated concepts. Thus, for each generated key we additionally need $O(|M|)$ time to check if a concept was generated at previous iterations.

We proposed an algorithm called GDPM (Gradual Discovery in Pattern Mining) to compute the closure structure of concept lattice by levels. Its detailed description and an example are given in [11] (referred there as CbO-Gen). Here we give its brief description.

The pseudocode of GDPM is given in Algorithm 1. The algorithm computes concepts based on the breadth-first traversal, i.e., at the level k it computes all concepts that have a minimum key of size k. Each newly generated key is obtained by computing the union of a minimum key from the previous level and an attribute that is out of the closure of the minimum key (lines 3-13). The key is added to $\mathcal{K}_k^*$ only if its closure is not in the lexicographic tree that stores all generated previously intents (lines 8-11). For each concept we store only one minimum key in $\mathcal{K}_k^*$.

Algorithm 1 GDPM

```
Require: $\mathcal{K}^*_{k-1}$, a subset of the minimum key set $\mathcal{K}^{min}_{k-1}$ of level $k-1$,
    $\mathcal{T}_{k-1}$, the lexicographic tree containing all closed itemsets $\bigcup_{i=1,\dots,k-1} \mathcal{C}^{min}_i$
Ensure: $\mathcal{K}^*_k$, a subset of the minimum key set of level $k$
$\mathcal{K}^*_k \leftarrow \emptyset$
$\mathcal{T}_k \leftarrow \mathcal{T}_{k-1}$
for all $X \in \mathcal{K}^*_{k-1}$ do
    $Y \leftarrow M \setminus X''$
    for all $m \in Y$ do
        $X^* = X \cup \{m\}$
        $S \leftarrow (X^*)''$
        if $S \notin \mathcal{T}_k$ then
            $add(\mathcal{T}_k, S)$
            $\mathcal{K}^*_k \leftarrow \mathcal{K}^*_k \cup \{X^*\}$
        end if
    end for
end for
return $\mathcal{K}^*_k$
```

4 Concepts as classifiers. Baseline classification model

In order to evaluate intents (closed itemsets) as representatives of the classes we propose to use the class labels of objects that were unavailable during computing the closure structure. Further we describe a simple concept-based classification model. This model is closely related to the JSM-method proposed by Finn, that is widely used in the FCA community [10, 3, 8].

Let (G, M, I) be a training context, and each object g belongs to one class $label(g) \in \mathcal{Y}$, where $\mathcal{Y}$ is a set of class labels. We use concepts as classifiers. Let $c = (A, B) \in \mathcal{C}$ be a formal concept, then its class is given by $class(c) = \arg\max_{y \in \mathcal{Y}}(\sum_{g \in A}[label(g) = y])$, where $[\cdot]$ is an indicator function, taking 1 if the condition in the bracket is true and 0 otherwise. An object g is classified by a set of concept classifiers $\mathcal{C}^*$ based on the weighted majority vote as follows: $classify(g, \mathcal{C}^*, w, \theta) = \arg\max_{y \in \mathcal{Y}}(\sum_{\substack{c \in \mathcal{C}^*, w(c) \geq \theta \\ class(c)=y}} w(c))$, where $w(\cdot)$ is a weight function, and θ is a weight threshold. For example, for a concept $c = (A, B)$ weight can be defined based on one of the following functions:

$$prec(c) = \frac{tp(c)}{tp(c) + fp(c)},\ recall(c) = \frac{tp(c)}{tp(c) + fn(c)},\ F1(c) = \frac{2 \cdot prec(c) \cdot recall(c)}{prec(c) + recall(c)},$$

where $tp(c) = |\{g \mid label(g) = class(c), g \in A\}|$, $fp(c) = |A| - tp(c)$, $tn = |\{g \mid label(g) \neq class(c), g \in G \setminus A\}|$, $fn = |G \setminus A| - tn$.

As a set of classifiers we use either a single level $\mathcal{C}^{min}_k$ or all concepts up to level k, i.e., $\cup_{j \leq k} \mathcal{C}^{min}_j$.

Example. Let us consider the context from Table 1. We take the class labels where objects g_1 and g_2 belong to class "+", and $g_3 - g_6$ belong to class "−".

The weights of concept classifiers are given by the extent precision, e.g., $pr(a) = pr(c) = pr(d) = pr(f) = 2/3$. The threshold is $\theta = 2/3$. The intents a and c are from "+"-class, d and f are from "−"-class. Then, to classify object g_{test} described by $acdf$ we use the intents a, c, d, f from the 1st level, and ac, acf from the 2nd level. Using the intents of the 1st level we are not able to classify g_{test} since $pr(a) + pr(c) = pr(d) + pr(f) = 4/3$. For the intents of the 2nd level we have $pr(ac) + pr(acf) = 2$ for "+" class and 0 for "−" class, thus, we classify g_{test} as "+".

The proposed model is based on all intents from a given level that meet the weight requirements. However, more sophisticated models, e.g., Classy [12] or Krimp [14], can be adapted to use the intents by closure levels instead of frequent itemsets. More proper combination of the intents may improve the classification quality.

For large datasets the closure structure can be computed for each class independently.

5 Experiments

In this section we report the results of an experimental study of the minimum closure structure, i.e., closed itemsets within levels $\mathcal{C}_k^{min}$. We use freely available datasets from the LUCS/KDD data set repository [5], their characteristics are given in Table 2.

Table 2. Description of datasets and their level structure.

name	$\|G\| \times \|M\|$	dens.	$\|\mathcal{C}\|$	level size $\|\mathcal{C}_k^{\min}\|$										
				1	2	3	4	5	6	7	8	9	10	11
hepatitis	155×50	0.36	144870	50	761	4373	14696	31240	41995	33048	14724	3570	399	14
heart-d.	303×45	0.29	25538	45	561	2696	6381	7980	5389	2037	417	32		
auto	205×129	0.19	57788	127	2695	12539	21311	15283	5042	748	43			
glass	214×40	0.23	3245	40	450	1217	1041	386	96	14	1			
pima	768×36	0.22	1625	36	284	466	451	271	96	19	2			
iris	150×32	0.50	4481	30	263	792	1279	1335	682	100				
led7	3200×28	0.50	1950	14	84	280	560	630	321	61				
ticTacToe	958×27	0.33	42711	27	324	2266	9664	16982	10648	2800				
wine	178×65	0.20	13228	65	1289	4779	5026	1791	265	13				
zoo	101×35	0.46	4569	35	377	1194	1656	1059	239	9				
breast	699×14	0.64	361	12	50	112	125	55	7					
car eval.	1728×21	0.29	7999	21	183	847	2196	3024	1728					
ecoli	327×24	0.29	425	24	138	185	67	10	1					

5.1 Concept contrast based on F1 measure

In IM apart of descriptive quality of itemsets as coverage or diversity, one is interested in assessing the quality of itemsets as representatives of the classes. The latter is closely related to emerging patterns [6, 1]. In this study we evaluate contrast of formal extents by F1 measure. As it was shown in [11], the average F1 measure usually decreases w.r.t. closure levels. However, there are some datasets with atypical behavior, where the average F1 measure increases at the last levels of the closure structure. To address the underlying causes of this behavior we study the values of F1 measure within the frequency ranges of size 0.1, i.e., $(0.0, 0.1]$, $(0.1, 0.2]$, $(0.2, 0.3]$, etc.

Fig. 2 shows the results for some datasets. Our experiments showed that usually the value of F1 measure of the concepts within a fixed frequency range remains almost the same at all levels. Thus, the average F1 at a closure level is affected by the proportion of the concepts of a certain frequency. Since the ratio of frequent (infrequent) concepts decreases (increases) with the level number, F1 measure decreases as well. Thus, we may expect increase of the average F1 measure at the last levels for datasets with a large number of frequent and "coherent" attributes and a subset of infrequent "incoherent" attributes.

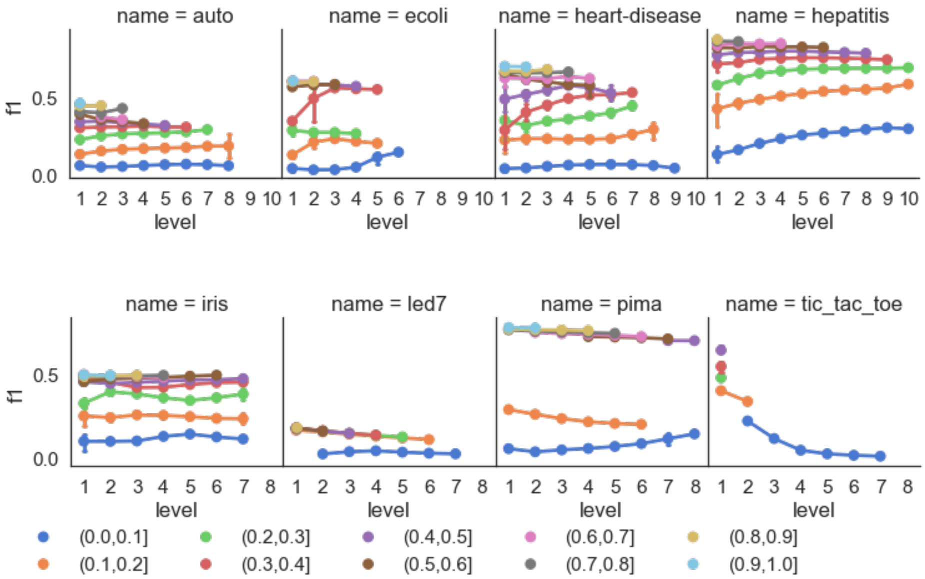

Fig. 2. The average F1 measure for 8 datasets within 10 frequency ranges.

In the previous study we showed that the size of closure levels resembles the values of binomial coefficients, i.e., the largest levels are located in the middle of the closure structure, while the first and last levels are the smallest ones. In Fig. 3 we show the size of the levels w.r.t. 10 frequency ranges. As for the whole set of concepts, for the subset of concepts of a given frequency we observe quite similar level size distributions – the largest level is on the middle, the smallest ones are located the first and last levels. The index of the largest levels is shifting – the less frequency the larger the index of the largest level.

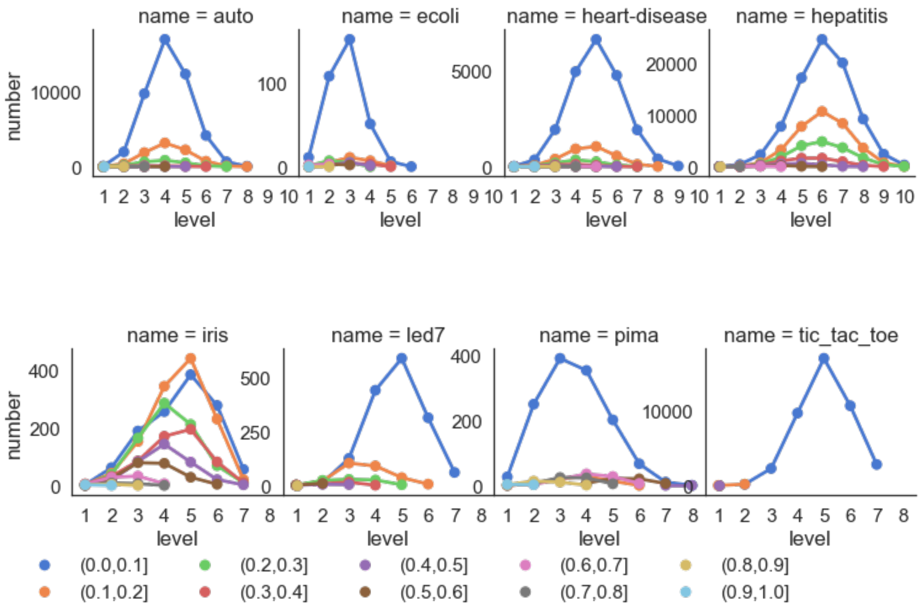

Fig. 3. The size of the concepts of a given frequency range w.r.t. level number.

5.2 Classification quality

In this section we report the average accuracy by 10-fold cross validation of the rule-based classifier described in Section 4. We use 8 folds (training set) to compute itemsets, 1 fold (test set) to select the best parameters and 1 fold (validation set) to assess the performance of the classifiers. We report the average values on the validation sets. We use both concepts from a single closure level (single level, SL) and concepts from all levels up to a given level (cumulative levels, CL) to build a classifier. As a weight function we use precision with the following threshold values: 0.0, 0.6, 0.7, 0.8 and 0.9.

The experiments showed that both SL- and CL-classifiers may achieve quite high accuracy. The average accuracy for 8 datasets is given in Fig. 4. The maximal (or close to the maximal) accuracy of CL-classifiers is achieved at the first levels and usually changes slightly when the classifier is extended by the further closure levels. For SL-classifiers the maximal (or close to the maximal) accuracy is usually achieved at one of the first levels.

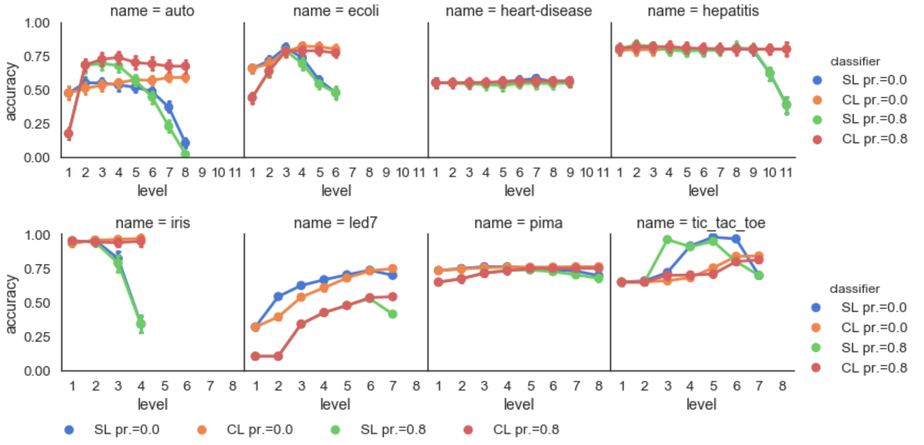

Fig. 4. Average accuracy of four classifiers: "SL/CL pr. = X" stands for a single-level (SL) or cumulative (CL) classifier where each concept has precision at least X.

We also compared the proposed classifier with the state-of-the-art classifiers from the Sklearn library. We consider SVM, Naive Bayes, and 3 tree-based models: Random Forests, CART, C5.0. We also use three sets to select the best parameters for each classifiers, i.e., the number of trees for tree-based classifiers (50 or 100), the maximum tree depth (2, 5, 10, 15) and the kernel types for SVM (polynomial, Radial basis function, sigmoid).

The average accuracy is reported in Table 3. The experiments show that even with the simplest model of classifiers based on closure level we can achieve the accuracy comparable with the one of the state-of-the-art classifiers. A more proper selection and combination of the generated concepts may provide better quality.

Based on the obtained results we may conclude that the proposed level-wise strategy allows us to generate the concepts that describe meaningful groups of objects and the intents from the first closure levels may be used as an alternative to frequent itemset.

6 Conclusion

In this paper, we study further the closure structure of concept lattice by focusing on the ability of concepts to describe meaningful subsets of objects. Our experiments show that the levels of the closure structure are a good alternative

Table 3. The average accuracy of classifiers. The best performance of the proposed classifiers (SL/CL) and their parameters (level, precision threshold) are given in the last two columns.

data	RF	CART	C5.0	SVM	NB	only the first 3 levels are considered				all levels are considered			
						CL	level, pr.	SL	level,pr.	CL	level,pr.	SL	level,pr.
auto	**0.77**	0.66	0.64	0.74	0.54	0.71	3, 0.8	0.65	2, 0.8	0.74	4, 0.9	0.65	2, 0.8
breast	**0.94**	**0.94**	**0.94**	**0.94**	0.93	**0.94**	3, 0.0	**0.94**	3, 0.6	**0.94**	3, 0.0	**0.94**	3, 0.6
car_ev.	0.96	0.97	0.97	**0.99**	0.80	0.70	3, 0.8	0.74	3, 0.0	0.86	5, 0.9	0.88	5, 0.0
ecoli	**0.87**	**0.87**	0.86	0.86	0.66	0.80	3, 0.7	0.84	3, 0.6	0.81	4, 0.7	0.84	3, 0.6
glass	**0.76**	0.69	0.66	0.72	0.45	0.65	3, 0.8	0.63	3, 0.0	0.65	4, 0.9	0.63	3, 0.0
heart-d.	0.56	0.54	0.53	0.55	0.20	0.54	1, 0.0	0.54	3, 0.6	0.55	7, 0.9	**0.57**	7, 0.0
hepatitis	0.81	0.78	0.80	**0.83**	0.53	**0.83**	2, 0.9	0.81	3, 0.9	**0.83**	2, 0.9	0.81	3, 0.9
iris	0.94	0.94	0.94	0.93	0.95	**0.96**	2, 0.6	0.95	1, 0.7	**0.96**	2, 0.6	0.95	1, 0.7
led7	0.75	0.75	0.75	**0.76**	0.75	0.55	3, 0.6	0.62	3, 0.0	0.74	7, 0.0	0.74	6, 0.7
pima	0.74	0.72	0.74	0.73	0.53	0.74	3, 0.6	**0.75**	3, 0.6	**0.75**	4, 0.6	**0.75**	4, 0.6
ticTacToe	0.98	0.95	0.94	0.98	0.68	0.97	3, 0.9	0.98	3, 0.9	**0.99**	6, 0.9	**0.99**	6, 0.6
wine	0.94	0.86	0.90	**0.95**	0.84	0.93	3, 0.9	0.90	3, 0.7	0.93	6, 0.9	0.92	4, 0.6
zoo	0.86	0.91	0.94	**0.96**	0.95	0.84	3, 0.8	0.83	3, 0.7	0.82	4, 0.8	0.83	3, 0.7
average	**0.84**	**0.81**	**0.82**	**0.84**	**0.68**	**0.78**		**0.78**		**0.81**		**0.81**	

to frequency. Each closure level is computed with polynomial delay, and the quality of itemsets decreases after the first levels.

One of the main directions of future work is to develop more efficient algorithms for computing the closure levels and study other practical applications where the proposed closure structure may provide better results than the frequency-based concept generation.

References

1. Asses, Y., Buzmakov, A., Bourquard, T., Kuznetsov, S.O., Napoli, A.: A hybrid classification approach based on FCA and emerging patterns - an application for the classification of biological inhibitors. In: Proceedings of the 9th International Conference on Concept Lattices and Their Applications. CEUR Workshop Proceedings, vol. 972, pp. 211–222 (2012)
2. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. ACM SIGKDD Explorations Newsletter **2**(2), 66–75 (2000)
3. Blinova, V., Dobrynin, D., Finn, V., Kuznetsov, S.O., Pankratova, E.: Toxicology analysis by means of the jsm-method. Bioinformatics **19**(10), 1201–1207 (2003)
4. Buzmakov, A., Kuznetsov, S., Napoli, A.: Sofia: how to make fca polynomial? In: Proceedings of the 4th International Conference on What can FCA do for Artificial Intelligence?-Volume 1430. pp. 27–34. CEUR-WS. org (2015)
5. Coenen, F.: The LUCS-KDD discretised/normalised ARM and CARM Data Library. http://www.csc.liv.ac.uk/ frans/KDD/Software/LUCS_KDD_DN,
6. Cuissart, B., Poezevara, G., Crémilleux, B., Lepailleur, A., Bureau, R.: Emerging Patterns as Structural Alerts for Computational Toxicology. In: Dong, G., Bailey,

J. (eds.) Contrast Data Mining: Concepts, Algorithms, and Applications, pp. 269–282. CRC Press (2013)

7. Ganter, B., Wille, R.: Formal concept analysis–mathematical foundations (1999)
8. Ganter, B., Grigoriev, P.A., Kuznetsov, S.O., Samokhin, M.V.: Concept-based data mining with scaled labeled graphs. In: International Conference on Conceptual Structures. pp. 94–108. Springer (2004)
9. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR) **38**(3), 9–es (2006)
10. Kuznetsov, S.O.: Interpretation on graphs and complexity characteristics of a search for specific patterns. Automatic Documentation and Mathematical Linguistics **24**(1), 37–45 (1989)
11. Makhalova, T., Kuznetsov, S.O., Napoli, A.: Gradual discovery with closure structure of a concept lattice. In: The 15th International Conference on Concept Lattices and Their Applications (2020)
12. Proença, H.M., van Leeuwen, M.: Interpretable multiclass classification by mdl-based rule lists. Information Sciences **512**, 1372–1393 (2020)
13. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with titanic. Data & knowledge engineering **42**(2), 189–222 (2002)
14. Vreeken, J., Van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery **23**(1), 169–214 (2011)