MDL for FCA: is there a place for background knowledge?

Tatiana Makhalova, Sergei Kuznetsov, Amedeo Napoli

MDL for FCA: is there a place for background knowledge?

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

[1] National Research University Higher School of Economics,
3 Kochnovsky Proezd, Moscow, Russia
[2] LORIA, (CNRS – Inria – U. of Lorraine), BP 239
Vandœuvre-lès-Nancy, France
tpmakhalova@hse.ru, skuznetsov@hse.ru, amedeo.napoil@loria.fr

Abstract. The Minimal Description Length (MDL) principle is a powerful and well founded approach, which has been successfully applied in a wide range of Data Mining tasks. In this paper we address the problem of pattern mining with MDL. We discuss how constraints – background knowledge on interestingness of patterns – can be embedded into MDL and argue the benefits of MDL over a simple selection of patterns based on measures.

1 Introduction

Formal Concept Analysis (FCA) is a formalism that can be applied to Knowledge Discovery and Data Mining. It is used commonly for solving a wide range of tasks: from pattern mining to design of ontologies.

Even controlled application of FCA in practice may result in exponentially large output, which entails additional steps aimed at reducing / selecting a small subset of concepts. The reduction of the number of formal concepts may be done during pre-/postprocessing stages.

In this paper we propose to combine two of the most common concept filtering approaches: the Minimal Description Length principle (MDL) [2, 6, 7, 12] and measure-based selection [8]. This combination tries to take the advantages of both methods and reduces the drawbacks of each one.

The idea of MDL is to select a subset of patterns that ensures the best compression of data. It has been embedded into FCA in a number of ways: for defining how many factors to use in Boolean matrix factorization (BMF) [3,10,11] or to get more diverse itemsets in frequent pattern mining (FIM) [1, 9, 13] or to select triclusters [14]. Being threshold-free, MDL provides a succinct non-redundant set of concepts. However, it has some shortcomings. Since the length minimisation is at least NP-complete, the implementation of MDL is based on heuristics. The selected itemsets cannot be interpreted easily by experts.

Unlike MDL, the selection of itemsets based of values of some measure is easy to interpret. A measure reflects the assumption on interestingness of patterns. Selecting the best itemsets w.r.t. the chosen measure one obtains patterns

with the desired characteristics. However, this approach requires threshold and returns a lot of similar patterns.

In this paper we use the Krimp algorithm as an implementation of MDL principle to improve measure-based selection. Krimp is based on greedy covering of data by a set of patterns (subsets of attributes) called candidate set. The patterns in a candidate set are ordered w.r.t. the pattern length and its frequency. We propose to use different interestingness measures to order candidates. This modification allows for embedding background knowledge, i.e., our assumptions on interestingness. The aim of the ordering w.r.t. different measures is to improve measure-based pattern selection rather than to compress data the best. Using a preferable ordering one gets a good compression as well as only those patterns that satisfy defined constraints.

The rest of the paper has the following structure. In Section 2 we briefly recall the main notions of FCA. In Section 3 we describe the MDL principle and discuss how interestingness measures can be used within MDL. The benefits of our approach are discussed in Section 4, where we compare MDL-based with threshold-based measure selection. Section 5 gives the conclusion and discuss the direction of future work.

2 Formal Concept Analysis: Basic Notions

Here we briefly recall FCA terminology [5]. A formal context is a triple (G, M, I), where $G = \{g_1, g_2, ..., g_n\}$ is called a set objects, $M = \{m_1, m_2, \ldots, m_k\}$ is called a set attributes and $I \subseteq G \times M$ is a relation called incidence relation, i.e. $(g, m) \in I$ if the object g has the attribute m. The derivation operators $(\cdot)'$ are defined for $A \subseteq G$ and $B \subseteq M$ as follows:

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

A' is the set of attributes common to all objects of A and B' is the set of objects sharing all attributes of B. An object g is said to contain a pattern (set of items or itemset) $B \subseteq M$ if $B \subseteq g'$. The double application of $(\cdot)'$ is a closure operator, i.e. $(\cdot)''$ is extensive, idempotent and monotone. Sets $A \subseteq G$, $B \subseteq M$, such that $A = A''$ and $B = B''$, are said to be closed.

A (formal) concept is a pair (A, B), where $A \subseteq G$, $B \subseteq M$ and $A' = B$, $B' = A$. A is called the (formal) extent and B is called the (formal) intent of the concept (A, B). A formal concept is said to cover set of objects A and set of attributes B. A partial order $\leqslant$ 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).

The number of formal concepts can grow exponentially w.r.t. the size of a formal context, i.e., the number of objects in G and attributes in M. We say that a set of patterns $\mathcal{S}$ covers objects in G if $\bigcup_{B \in \mathcal{S}} B' = G$, where $B \subseteq M$. We are interested in a small set of patterns (intents) $\mathcal{S}$ that covers all objects and most of their attributes, i.e., $\left|\bigcup_{B \in \mathcal{S}} \{gIm \mid g \in B', m \in B\}\right| \approx |I|$.

Example. Let us consider a toy example. A formal context is given in Figure 1 (1). We consider 3 sets of itemsets (intents): $\mathcal{S}_2 = \{\{abc\}, \{bcde\}, \{de\}, \{cde\}, \{ac\}\}$, $\mathcal{S}_3 = \{\{bc\}, \{de\}, \{ac\}\}$ and $\mathcal{S}_4 = \{\{c\}, \{de\}\}$. The corresponding coverings of the context are given in Figure 1 (2-4). The intensity of colors is proportional to the number of times a particular "cross" is covered by intents. In our example "crosses" are covered by intents from 0 to 4 times. We count not only the number of intents, but also the number of covered "crosses", we call this value the rate of a cover relation, or $RCR = |\text{"crosses" that covered at least once}|/|I|$. It can be seen from the covering given in Figure 1 that $\mathcal{S}_3$ (Figure 1, (3)) provides the best covering w.r.t. the number of itemsets (intents) and the rate of covered elements in the object-attribute relation.

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(1) A formal context

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(2) Covering of objects with $\mathcal{S}_2 = \{\{abc\}, \{bcde\}, \{de\}, \{cde\}, \{ac\}\}$. RCR = 1.

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(3) Covering of objects with $\mathcal{S}_3 = \{\{bc\}, \{de\}, \{ac\}\}$. RCR = 1.

	a	b	c	d	e
1	x	x	x		
2		x	x	x	x
3				x	x
4	x		x	x	x
5	x		x		

(4) Covering of objects with $\mathcal{S}_4 = \{\{c\}, \{de\}\}$. RCR = 10/15.

Fig. 1. Formal context and its coverings.

The Minimal Description Length principle allows for covering with a substantial rate of "crosses" in I by a small number of patterns. In the next section we use MDL in a more general framework, i.e., in pattern mining. Intents of formal concepts, in turn, can be considered as patterns of a special kind.

3 Minimal Description Length: Basic Notions

MDL is aimed to find a subset of patterns that compresses data the best. In our study we use Krimp [13] as a practical implementation of this principle. In Section 3.1 we give a short description of it and in Section 3.2 we discuss how background knowledge can be embedded into MDL.

3.1 MDL in Practice: the Krimp Algorithm

The input of the algorithm is a dataset and a list of patterns (that are computed on the same dataset). The patterns are ordered w.r.t. their length and frequency. The result of Krimp is a two-column code table that consists of patterns and their encoding lengths (an examples of code tables are given in Figure 2). The objective of Krimp is minimization of the function

$$L(D, CT) = L(D \mid CT) + L(CT \mid D), \tag{1}$$

where $L(D \mid CT)$ is the length of the dataset $D = \{g' \mid g \in G\}$ encoded with the code table CT and $L(CT \mid D)$ is the length of the code table CT computed w.r.t. D. The objects are encoded by disjoint patterns in a greedy manner, i.e., starting from the top elements of CT. The length of pattern B is computed using an optimal prefix code given by Shannon entropy, i.e., the length $l(B) = -\log(u(B)/U)$ is inversely proportional to the usage $u(B) = |\{t \in D \mid B \in cover(t, CT)\}|$. The usage shows how many times B is used to cover objects in D, $U = \sum_{B \in CT} u(B)$ is the total usage of itemsets. We leave the details on itemset storage out of scope of this paper and take into account the compression related to a particular choice of itemsets, i.e., we use the simplified version of the lengths:

$$L(D \mid CT) = \sum_{g \in D} \sum_{B \in cover(g, CT)} l(B) = -\sum_{B \in CT} u(B) \log \frac{u(B)}{U},$$

$$L(CT \mid D) = \sum_{B \in CT} l(B) + code(B).$$

A code table is incrementally computed. At the beginning it contains only single-attribute patterns $\{\{m\} \mid m \in M\}$. A set of patterns – candidates in the code table – are ordered w.r.t. their length (intent cardinality) and frequency (extent cardinality). At each step the best candidate is added to the code table if its usage allows for smaller encoding length, otherwise it is removed from the code table and the candidate set.

Example. Let us consider how Krimp selects patterns using the running example (the context is given in Figure 1, (1)). Here we represent the context as a set of transactions, see Figure 2, (1). The main stages are given in Figure 2, (2-4). As candidates we use intents of formal concepts with the size of intent and extent exceeding 1. We sort them first by the size of intent and then by the size of extent (in descending order). Let us consider some steps of the algorithm.

Initial state (Figure 2, (2)): the code table consists of single-attribute patterns. Usage is equal to frequency. Sets of attributes in the dataset are covered by single-attribute patterns.

First step (Figure 2, (3)): An attempt to add the top pattern from the candidate set. Pattern ac is used to cover object g_1, g_4 and g_5 (Figure 2, (3)), the usage of single attributes a and c decreases by 3. The description length (see Formula 1) is computed for the updated code table and covering. Since the inclusion of ac into the code table provides smaller description length, ac is accepted for the code table.

Further, the top patterns one by one are used to minimize the description length.

Last step (Figure 2, (4)): The last candidate bc can cover only object g_2 (since subsets bc are partially covered by other members of the code table). It is not added since its inclusion in the code table does not provide better compression (i.e. smaller description length). The subset of MDL-optimal patterns is $\{\{ac\}, \{de\}\}$.

.

Objects	Desctiption
g_1	acb
g_2	bcde
g_3	de
g_4	acde
g_5	ac

(1) A formal context from Figure 1, (1) represented as a set of transactions

CT	
IS	U
a	3
b	2
c	4
d	3
e	3

CS
ac
de
bc

Covering
(a)(b)(c)
(b)(c)(d)(e)
(d)(e)
(a)(c)(d)(e)
(a)(c)

(2) Initial state.

CT	
IS	U
a	0
b	2
c	1
d, e	3
ac	3

CS
de
bc

Covering
(ac)(b)
(b)(c)(d)(e)
(d)(e)
(ac)(d)(e)
(ac)

(3) The state after adding ac to the code table.

CT	
IS	U
a,d,e	0
b	2
c	1
ac	3
de	3

CS

Covering
(ac)(b)
(b)(c)(de)
(de)
(ac)(de)
(ac)

(4) The last step. bc is not added to the code table.

Fig. 2. The main stages of the Krimp algorithm. "Covering" tables show the dataset with covering by itemsets from the corresponding code table above the covering, $(\cdot)$ depicts an itemset that covers some attributes of an object. CT is a two-column code table, where "IS" and"U" stand for itemsets and their usage in greedy covering, respectively. "CS" is a candidate set.

3.2 MDL in Practice: Compression under Constraints

The implementation of the MDL principle is based on heuristics and allows for the solution which is close to the optimal one. In practice, there exist several ways to select subsets of patterns that have almost the same size and ensure good compression. Thus, it becomes difficult to explain why a particular subset was chosen.

More than that, by selecting a subset of patterns one is interested in patterns that have particular properties, e.g., being stable w.r.t. noise, have high probability under certain condition, etc. Despite proper interpretability, the application of interestingness measures requires a threshold and results in a redundant set of patterns (quite similar patterns). As interestingness measures of concept (A, B) we took frequency $fr(B) = |A|$, i.e. the size of extent, length $len(B) = |B|$, i.e., the size of the intent, and lift $lift(B) = \prod_{b \in B} Pr(b)/Pr(B)$, where $Pr(\cdot) = |(\cdot)'|/|G|$.

In our study we combine the measure-based selection with Krimp to get a threshold-free approach that provides a small non-redundant subset of patterns having desired properties. The modified approach works as follows. First, all patterns are sorted w.r.t. chosen interestingness measures. Then the ordered set is considered as a candidate set. The greedy covering strategy (Krimp) is applied to select the most interesting and diverse patterns. The original workflow and the adapted version that is used in the paper are given in Figure 3.

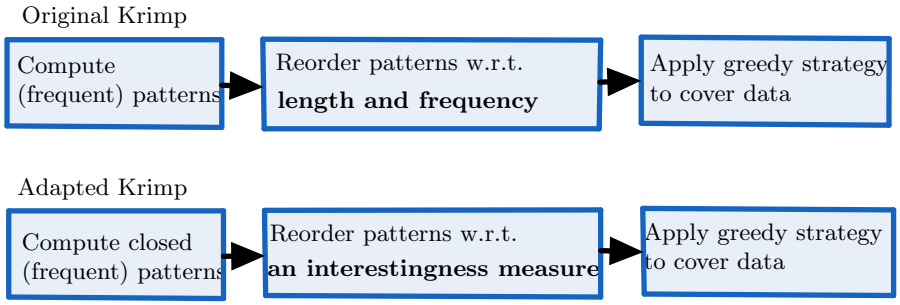


Fig. 3. The workflow for pattern mining by the original Krimp and its adapted version.

In the next section we show how the embedding of background knowledge (i.e. reordering of patterns w.r.t. interestingness measures) affects the results of pattern mining.

4 MDL in Closed Itemset Mining

In the worst case a concept lattice contains an exponential number of partially ordered intents (concepts), the application of MDL allows for the selection of a small subset of intents. Our experiments show that the application of the MDL principle allows for significant reduction in the number of patterns (up to 5% of the formal concepts, see Table 2). In the context of measure-based pattern mining, the application of MDL makes the measure-based selection threshold-free. More than that, a set of the MDL-optimal patterns has better characteristics than the top-n patterns. First of all, almost the same concepts (intents) are removed from the set of selected patterns. In our experiments we call this property "non-redundancy". For a set of patterns to be "non-redundant" means to have the following characteristics: differ from the most similar pattern in the set (i.e., distance to the 1st nearest neighbor), make shallow hierarchy by inclusion $B_1 \subset B_2 \subset \ldots \subset B_n$ (i.e., average length of the longest paths built from partially ordered itemsets) and do not have a lot of more general patterns $B_i \subset B$, $i \in [1, k]$ (the rate of patterns with children).

If we compare the sets of top-n and MDL-optimal patterns of the same size we will see, as a side effects of the "non-redundancy", that MDL-optimal patterns cover in total more data ("crosses" in a formal context) being diverse and interesting w.r.t. a given measure.

It is clear to see that MDL approach not only dispenses from predefined thresholds but also filter out similar interesting patterns and provides more comprehensive data description.

We examine the following orders of patterns: $area\_fr\_lift(B) = fr(B) \cdot lift(B)$, $area\_len\_fr(B) = len(B) \cdot fr(B)$, $area\_len\_lift(B) = len(B) \cdot lift(B)$ and sequential ordering by len and fr, len and $lift$, $lift$ and len (the patterns

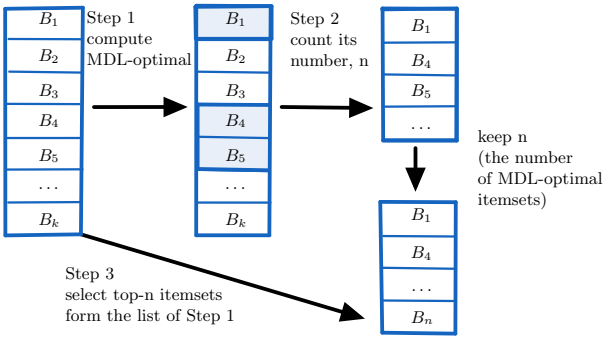


Fig. 4. The principle of computing MDL-optimal and top-n sets of patterns

are ordered by the chosen measure on Step 1 in Figure 4). An example of ordering for frequent closed itemsets (frequency is greater than 2) for the running example is given in Table 1.

Table 1. Values of interestingness measures and ordering of patterns for the running example from Figure 1, (1). An alternative ordering is given in $(\cdot)$, to select one ordering among the altervative ones additional rules are required to set.

Concepts (A, B)	fr $\lvert A\rvert$	len $\lvert B\rvert$	area (len × fr)	patterns ordered w.r.t. length and frequency **values of measures**	patterns ordered w.r.t. area_len_fr **values of measures**
({1245},{c})	4	1	4	{cde}; **3,2**	{de} ({ac}, {cde}); **6**
({234},{de})	3	2	6	{de} ({ac}); **2,3**	{ac} ({de}, {cde}); **6**
({145},{ac})	3	2	6	{ac} ({de}); **2,3**	{cde} ({de}, {ac}); **6**
({12},{bc})	2	2	4	{bc}; **2,2**	{c} ({bc}); **4**
({24},{cde})	2	3	6	{c}; **1,4**	{bc} ({c}); **4**

The discretized datasets from LUCS-KDD repository [4] were used in the study, the parameters of the datasets are given in Table 2. We split each dataset into 10 parts and in each of 10 experiments we use 9 of them as a training set and one part as a test set.

In this section we compare characteristics of MDL-optimal with top-n itemsets, patterns in both sets are ordered w.r.t. the same interestingness measure. The size of a set of top-n itemsets is equal to the size of a set of MDL-optimal patterns. The scheme of computing these sets is given in Figure 4. We compare the sets of patterns within the following properties: non-redundancy, data covering and representativeness.

Table 2. Characteristics of datasets

dataset	nmb. of obj.	nmb. of attr.	nmb. of concepts	Number of MDL-optimal					
				area fr_lift	area len_fr	area len_lift	len fr	len lift	lift len
breast	699	16	702	36.0	32.2	20.4	37.3	37.3	33.5
car	1 728	25	12 420	868.4	849.2	138.6	714.6	847.7	698.3
ecoli	336	29	690	58.8	55.9	16.4	64.9	65.6	55.9
iris	150	19	183	31.1	28.9	12.9	34.8	34.6	26.3
led7	3 200	24	3 808	108.0	118.3	64.2	108.7	108.7	130.3
pima	768	38	2 769	110.1	106.3	35.9	120.6	112.1	101.7

4.1 Non-redundancy

By redundant set of patterns we mean a set of patterns that contains a lot of similar itemsets. We measure redundancy by three parameters: average distance to the 1st nearest neighbor, average length of the longest paths built from partially ordered itemsets, and average number of itemsets that have at least one more general itemset (child).

Distance to the 1st nearest neighbor. To compute this parameter we represent patterns as binary vectors and take into account the smallest Euclidean distance between each pattern and the remaining patterns in the pattern set. The average value for a pattern set is taken as the average distance to the 1st nearest neighbor. A set containing a lot of similar patterns will have low average values, see Figure 5 (1) for an example.

As it can be seen from Figure 6 (1), the MDL principle provides much more distinctive itemsets. Top-n concepts have a lot of similar patterns, while MDL-optimal ones are pairwise distinctive (w.r.t. Euclidean distance).

Average length of the longest paths built from partially ordered itemsets. The patterns can be partially ordered by inclusion, i.e. $B_1 \subset B_2 \subset \ldots \subset B_n$, where B_n is the most specific patters and B_1 is the most general one. We call this ordered sequences paths. If $B_n \subseteq g'$ then $B_i \subseteq g'$ is guaranteed for all $i \in [1, n-1]$. Longer paths contain more patterns describing the same objects. Thus, a long path can be considered as an indicator of redundancy. In other words, these patterns characterize the same objects at different levels of abstraction and contain only a few new details w.r.t. the nearest neighbors in the path. Short paths correspond to "flat" structures with more varied patterns. An example of comparison of two tiny pattern sets is given in Figure 5, (2).

As we see in Figure 6 (b), for ordering w.r.t. *len* (see len_fr and len_lift) the MDL priciple does not provide any benefits, while its application to $area\_len\_sep$ and $area\_sep\_lift$, $lift\_len\_fr$ allows for more flattened structures, even more flattened than with *len*. It means that pattern mining with $area\_len\_sep$ and $area\_sep\_lift$, $lift\_len\_fr$ can be significantly improved by the application of MDL.

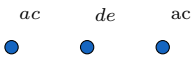
$\mathcal{S}_3$	binary representation (abcde)	nearest neighbor	Euclidean distance
bc	01100	ac	$\sqrt{2}$
de	00011	bc(ac)	2
ac	10100	bc	$\sqrt{2}$

The average distance is $(2 + 2\sqrt{2})/3$.

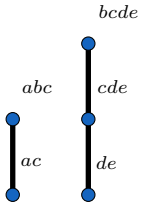
$\mathcal{S}_2$	binary representation (abcde)	nearest neighbor	Euclidean distance
bcde	01111	cde	$\sqrt{2}$
cde	00111	bcde (de)	$\sqrt{2}$
abc	11100	ac	$\sqrt{2}$
ac	10100	abc	$\sqrt{2}$
de	00011	cde	$\sqrt{2}$

The average distance is $\sqrt{2}$.

(1) Euclidean distances to the 1st nearest neighbors. The average distance for $\mathcal{S}_3$ is longer then for $\mathcal{S}_4$, thus $\mathcal{S}_3$ contains for diverse patterns.



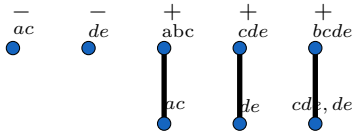
The average length
of the longest paths is 1.



The average length
of the longest paths is 2.5.

(2) The longest paths built on partially ordered patterns (by inclusion).

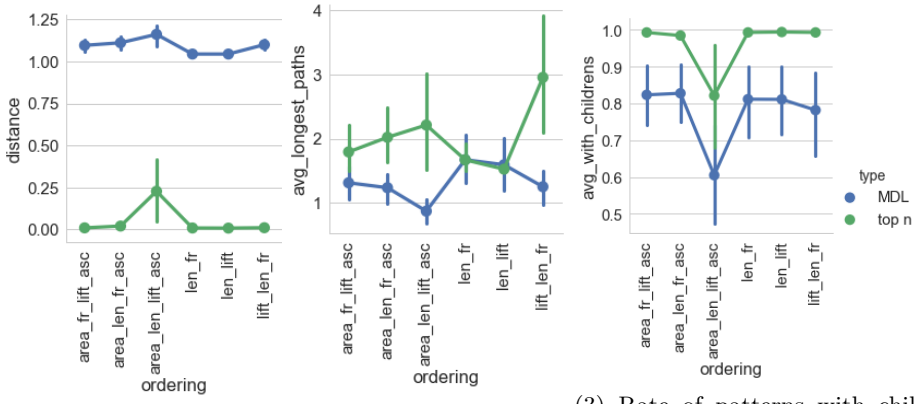
The rate of pattern with children is 0.



The rate of pattern with children is 3/5.

(3) The rate of patterns with children (i.e. more general / short patterns).

Fig. 5. Non-redundancy measures computed for patterns set given in Figure 1. The set $\mathcal{S}_3$ (column 1) is better than $\mathcal{S}_2$ (column 2) w.r.t. all the parameters: the average distance is higher, the average length of the longest paths and the rate of patterns with children are smaller.



(1) Distance to the 1NN (2) The average path lengths (3) Rate of patterns with children.

Fig. 6. Non-redundancy parameters: (1) the average distance to the 1st nearest neighbor for itemsets selected with MDL and top-n itemsets; (2) the average length of the longest paths computed on the chain of itemsets formed by inclusion of its attributes; (3) the average rate of itemsets with children. On the X-axis is different orderings of patterns, on the Y-axis is the values of the listed above non-redundancy parameters for MDL-optimal set (blue) and top-n (green) set of the same size.

Average number of itemsets with children (more general itemsets). This parameter characterizes the uniqueness of patterns in a set, absence of the second pattern $B_2 \subset B_1$ that characterizes the same subset as a more specific one. This parameter is related to the previous measure, but it indicates just an amount of itemsets having at least one more general itemset. An example of computing this parameter is given on Figure 5, (3).

The results of experiments (see Figure 6, (c)) show that the MDL principle selects more distinctive itemsets than top-n itemsets.

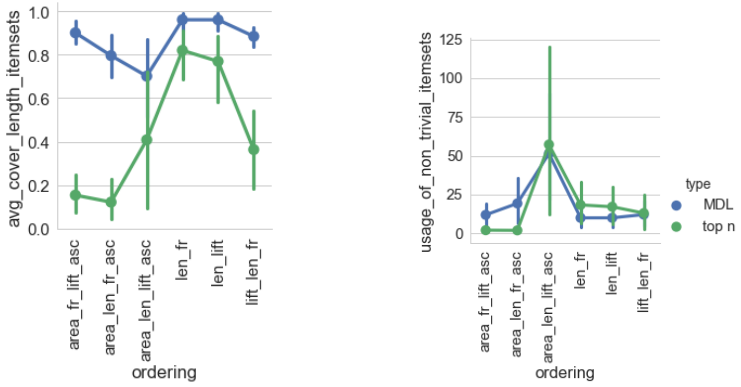
4.2 Data coverage

A subset of selected patterns can be considered as a concise representation of a dataset. Thus, it is important to know how much information is lost by compression. We measure this parameter by the rate of covered attributes. Values close to 1 correspond to the lossless compression.

The average covering rate is given in Figure 7 (1). With the same number of patterns MDL ensures better covering. For $area\_fr\_lift$, $area\_len\_fr$ and $area\_len\_lift$ MDL-optimal set covers much more data than top-n patterns.

4.3 Itemset typicality (representativeness)

In our experiments we also address typicality of patterns. In this study we measure it by the usage of patterns. To compute usage we consider the ordered



(1) Data coverage (2) Itemset typicality

Fig. 7. Pattern set parameters: (1) the average covering rate of itemsets (i.e. the rate of crosses covered by patterns); (2) the average itemset usage (reflects typicality/representativeness of patterns). On the X-axis is different orderings of patterns, on the Y-axis is the covering rate and the average itemset usage for MDL-optimal set (blue) and top-n (green) set of the same size.

patterns (in case of MDL, top patterns are those that have the shortest encoding length, for top-n they are top-patterns w.r.t. a chosen measure). The ordered patterns are used one by one to cover data. The attributes are covered only ones (disjoint covering by patterns). The number of times a patterns is used in the covering is its usage, thus the usage does not exceed the pattern frequency. For example, in Figure 2 (4), the frequency of bc is 2, but it can be used only one time to cover $(b)(c)(de)$, since in $(ac)(b)$ only b is left to cover.

It should be noted that it is not obvious which values are better. The usage serves to characterize a subset of patterns. The high values correspond to a subset of common patterns, while low values indicates that a subset contains less typical, but still interesting (w.r.t. interestingness measures) patterns.

Figure 7(2) shows the average usage for MDL-optimal and top-n patterns. The usage of MDL-optimal patterns is almost the same for different orders while the usage of top-n is dependent on ordering.

5 Conclusion

In the paper we propose a new approach to the measure-based pattern mining. It can be considered as an *"implementation of the MDL principle under constrains"* or *"embedding of background knowledge (on interestingness) into MDL"*. We took the Krimp algorithm as a basic implementation of MDL and studied a range of interestingness measures within it.

The proposed approach is a threshold-free method for the selection of a small set of patterns having desired properties. The chosen patterns are diverse and varied, they cover almost all attributes of objects.

The studied Krimp algorithm can be changed further to improve (closed) pattern mining as follows. The greedy strategy may be relaxed, i.e., overlapping patterns can be used to cover an object. Some additional mechanism may be proposed to deal with noisy data (missed values).

References

1. Aggarwal, C.C., Han, J.: Frequent pattern mining. Springer (2014)
2. Barron, A., Rissanen, J., Yu, B.: The minimum description length principle in coding and modeling. IEEE Transactions on Information Theory **44**(6), 2743–2760 (1998)
3. Belohlavek, R., Trnecka, M.: From-below approximations in boolean matrix factorization: Geometry and new algorithm. Journal of Computer and System Sciences **81**(8), 1678–1697 (2015)
4. Coenen, F.: The lucs-kdd discretised/normalised arm and carm data library. Department of Computer Science, The University of Liverpool, UK (2003), http://www.csc.liv.ac.uk/ frans/KDD/Software/LUCS_KDD_DN
5. Ganter, B., Wille, R.: Formal concept analysis: Logical foundations. Springer Verlag Berlin, RFA (1999)
6. Grünwald, P.D.: The minimum description length principle. MIT press (2007)
7. Hansen, M.H., Yu, B.: Model selection and the principle of minimum description length. Journal of the American Statistical Association **96**(454), 746–774 (2001)
8. Kuznetsov, S.O., Tatiana, M.: On interestingness measures of formal concepts. Information Sciences **442-443**, 202 – 219 (2018)
9. Li, J., Li, H., Wong, L., Pei, J., Dong, G.: Minimum description length principle: Generators are preferable to closed patterns. In: AAAI. pp. 409–414 (2006)
10. Miettinen, P., Vreeken, J.: Model order selection for boolean matrix factorization. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 51–59. ACM (2011)
11. Miettinen, P., Vreeken, J.: Mdl4bmf: Minimum description length for boolean matrix factorization. ACM Transactions on Knowledge Discovery from Data (TKDD) **8**(4), 18 (2014)
12. Siebes, A.: Mdl in pattern mining a brief introduction to krimp. In: Glodeanu, C.V., Kaytoue, M., Sacarea, C. (eds.) Formal Concept Analysis. pp. 37–43. Springer International Publishing, Cham (2014)
13. Vreeken, J., Van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery **23**(1), 169–214 (2011)
14. Yurov, M., Ignatov, D.I.: Turning krimp into a triclustering technique on sets of attribute-condition pairs that compress. In: Polkowski, L., Yao, Y., Artiemjew, P., Ciucci, D., Liu, D., Ślezak, D., Zielosko, B. (eds.) Rough Sets. pp. 558–569. Springer International Publishing, Cham (2017)