Numerical Pattern Mining Through Compression

Tatiana Makhalova, Sergei O. Kuznetsov, Amedeo Napoli

Numerical Pattern Mining Through Compression

Tatiana Makhalova[*,†], Sergei O. Kuznetsov[*], and Amedeo Napoli[†]

[*]National Research University
Higher School of Economics
3 Kochnovsky Proezd
Moscow, 125319, Russia
{tpmakhalova,skuznetsov}@hse.ru

[†]Université de Lorraine,
CNRS, Inria, LORIA
615 Rue du Jardin-Botanique
Nancy, F-54000, France
firstname.lastname@inria.fr

Abstract

Pattern Mining (PM) has a prominent place in Data Science and finds its application in a wide range of domains. To avoid the exponential explosion of patterns different methods have been proposed. They are based on assumptions on interestingness and usually return very different pattern sets. In this paper we propose to use a compression-based objective as a well-justified and robust interestingness measure. We define the description lengths for datasets and use the Minimum Description Length principle (MDL) to find patterns that ensure the best compression. Our experiments show that the application of MDL to numerical data provides a small and characteristic subsets of patterns describing data in a compact way.

1 Introduction

Pattern Mining (PM) is an essential field in Knowledge Discovery, and a lot of Machine Learning and Data Mining problems can be considered as its particular cases: from clustering and classification to association rule mining and feature selection. The presence of patterns in data makes compression valuable, since it allows for a more compact data representation. Data Mining, in turn, aims to "compress data by finding some structure in it" [1]. To date, apart from different indices for interesting assessment [2], many approaches to the compression-based mining have been proposed [3, 4, 5].

Mining through compression is related to the Minimum Description Length principle (MDL), where compression is the minimization of a description length. Recently, it was proposed to apply MDL to PM for binary and nominal data [6] and its application to closed itemset (formal concept) mining has been studied in [7, 8]. In the existing approaches, MDL is applied to nominal or binary data, while real-world data are usually more complex and very often numerical for example. In this paper we propose to apply MDL principle to numerical pattern mining.

We present a two-stage compression approach to compute numerical patterns. At the first stage we use compression techniques to generate pre-patterns, i.e., dense groups of similar objects in the complete attribute space. At the second stage we compress the pre-pattern descriptions to get patterns. To do that, we minimize an entropy-based description length defined on the intervals of attribute values.

The proposed approach is linear w.r.t. the input and output size. We offer an experimental proof that MDL can be used to find a set of diverse, non-redundant and interesting numerical patterns with short (lossless) descriptions.

The paper is organized as follows. Section 2 provides the basic notions used in this paper and gives the general view on the studied problem. We use Pattern Structures to handle the compressing data, the basics are given in Section 2.1. In Section 2.2 we discuss why the problem of numerical PM is harder than clustering and nominal or binary PM. We briefly recall the main notions of MDL in Section 2.3. In Section 3 we describe the proposed approach: Sections 3.1 and 3.2 present the pre-pattern and numerical pattern mining principles, respectively. In Section 4 we give the evaluation of the proposed approach. We focus on pattern characteristics rather than the compression rate to prove that the generated patterns not only ensure a good compression but also ease interpretation of the results and provide meaningful patterns. In Section 5 we conclude and give the directions of future work.

2 Pattern Mining: Basics

In this section we introduce the formalism used to handle pre- and patterns and discuss why the problem of numerical PM is more difficult than clustering and boolean/nominal PM. We also give an outline of MDL.

2.1 Interval Pattern Structures

When dealing with numerical patterns, we should chose how patterns will be represented, e.g., prototypes (mean and standard deviation), conjunctions of restrictions over the numerical attributes (intervals), etc. In our study we choose an interval-based pattern representation called *Interval Pattern Structures* (IPS).

Pattern Structures [9] is the generalization of Formal Concept Analysis (FCA) [10]. FCA is an applied lattice theory that relies on smart technique for enumerating the pattern search space and, generally, focus on compressed collections of closed patterns to avoid redundancy. It deals with binary data. Pattern Structures handle more complex data, e.g., numerical one, graphs, sequences, etc. Below, we briefly list the basic notion of IPS, the particular type of PS that deals with numerical data [11].

A pattern structure is defined as a triple $(G, (D, \sqcap), \delta)$, where G is a set of objects, $(D, \sqcap)$ is a complete meet-semilattice of descriptions given in an $|M|$-dimensional attribute space and mapping $\delta : G \to D$ associates each object with its description. In the IPS settings an object $g \in G$ is described by a vector of intervals $d \in D$, $d = \langle [l_i, r_i] \rangle_{i \in \{1,\dots,|M|\}}$, with $l_i, r_i \in \mathbb{R}$ and $l_i \leq r_i$. In $(D, \sqcap)$ the similarity operator $\sqcap$ is applied to object descriptions $d_1 = \langle [l_i^1, r_i^1] \rangle_{i \in \{1,\dots,|M|\}}$ and $d_2 = \langle [l_i^2, r_i^2] \rangle_{i \in \{1,\dots,|M|\}}$, $d_1, d_2 \in D$ and returns the convex hull given by $d_1 \sqcap d_2 = \langle [\min(l_i^1, l_i^2), \max(r_i^1, r_i^2)] \rangle_{i \in \{1,\dots,|M|\}}$.

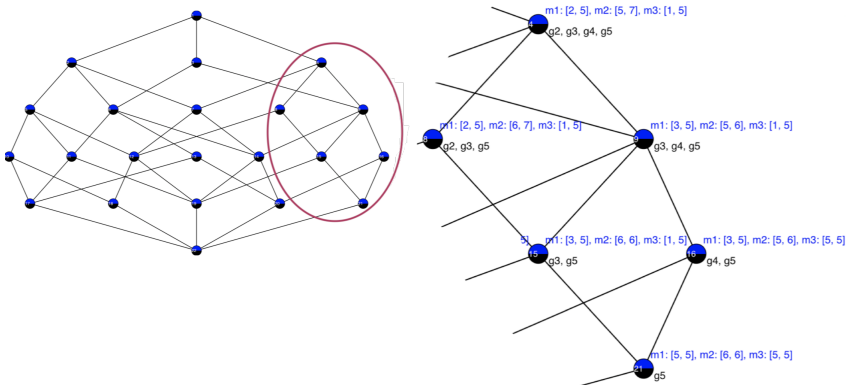
The Galois connection between $(\mathcal{P}(G), \subseteq)$ and $(D, \sqcap)$ is defined as follows:

$$A^{\square} := \sqcap_{g \in A} \delta(g) \text{ for } A \subseteq G; \; d^{\square} := \{g \in G \mid d \sqsubseteq \delta(g)\} \text{ for } d \in D,$$

where $\mathcal{P}(G)$ is a power set of objects G. $A^{\square}$ returns the description common for all objects from A and $d^{\square}$ returns the set of objects whose description subsumes d. The patterns are partially ordered w.r.t. the subsumption order $\sqsubseteq$, i.e., $\forall c, d \in D$, $c \sqsubseteq d \Leftrightarrow c \sqcap d = c$.

	m_1	m_2	m_3
g_1	1	5	1
g_2	2	7	1
g_3	3	6	1
g_4	3	5	5
g_5	5	6	5

(a) Numerical context.



(b) A pattern structure computed on a numerical context (a).

Figure 1: Numerical context and pattern concepts computed on it.

A pair (A, d) for which $A^{\square} = d$ and $d^{\square} = A$ is called pattern concept, where description d is a pattern intent and is common for all objects in A, A is a pattern extent and is a maximal set of objects that fit to description d.

In this study we differentiate *pattern intents* from *patterns*. A pattern intent contains intervals of all attributes (i.e., defined in the complete attribute set), while a pattern is not necessary to be defined in a complete attribute space. For the sake of convenience, we indicate explicitly to which attribute an interval belongs to when it is not clear.

Example. Let us consider a toy numeric context and the corresponding pattern structure given on Figure 1, (a) and (b), respectively. The most general concepts are on the top of the pattern structure, they are described by the largest intervals. The pattern intents are vectors of the minimal intervals that include all objects from the pattern extent. For example, a subset of objects $\{g_3, g_4, g_5\}$ is described by the following tuple of intervals: $\langle [3,5], [5,6], [1,5] \rangle$. A single interval $[1,2]_{m_1}$ is the descriptor of 2 objects $\{g_1, g_2\}$, the pattern intent for $\{g_1, g_2\}$ is $\langle [1,2], [5,7], [1,1] \rangle$.

As it can been seen from the example, even for a tiny dataset the number of pattern concepts is quite large. The number of patterns, i.e., fragments of pattern intents, is even much higher. Further we consider this problem from the PM perspective.

2.2 Search Space for Pattern Mining

Clustering is aimed to search subsets of similar objects, thus the search space for the problem is of the size $2^{|G|}$. In binary pattern mining the number of closed itemsets (formal concepts) is at most $min(2^{|M|}, 2^{|G|})$ [12], it grows exponentially w.r.t. the size of a formal context, i.e., the number of objects in G and attributes in M. For numerical PM, the search space even larger, it is at most $2^{|G|+|M|}$, since to compute borders of intervals we need to find subsets of objects $A \subseteq G$ (the search space is of the size $2^{|G|}$) and a pattern is a subset of intervals of $A^{\square}$ (the search space is of the size

$2^{|M|}$). Hence, the numerical PM implies the growth of the already exponential-sized search space by an order of magnitude w.r.t. clustering and boolean PM.

2.3 MDL principle

Minimum Description Length principle (MDL) reflects the widely accepted idea that the best data models provide the best data compression [13]. This belief is related to "Occam's razor" and has given rise to several theories and approaches in Computer Science, e.g., Kolmogorov complexity [14], Minimum Message Length [15], Kolmogorov's minimal sufficient statistic [16], etc.

In this study we propose to use a two-stage compression for numerical PM. At the first stage (Section 3.1) we compute pre-patterns, we use a description length that favors dense groups of similar objects. At the second stage (Section 3.2) we compute specific descriptions of the pre-patterns based on the entropy of intervals and take into account the description length of the pre-patterns (from the previous stage).

3 Numerical Pattern Mining: Two-stage Compression

In our approach we mine patterns in two stages: we compress data in the complete attribute space and then we compress the obtained description to get patterns. Let us consider in details these stages.

3.1 Pre-pattern Computing: Compression of Groups of Similar Objects

The goal of the first stage is to reduce search space for further pattern mining. Here, we compress a dataset by replacing dense groups of similar objects by their shared description, i.e., we compute those pattern concepts that ensure the best compression. The description length to be minimized is

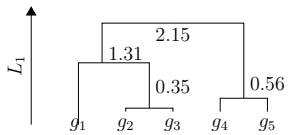
$$L_1(A) = \sum_{g_i,g_j \in A, i \neq j} \frac{|| \{g_i\}^{\square} - \{g_j\}^{\square} ||_2}{|A|^2}, \tag{1}$$

where $|| \cdot ||_2$ is the Euclidean distance, $|A|$ is the size of the pattern extent (A, d).

We use a greedy strategy to compress dataset, starting from the smallest pattern concepts, we use $^{\square\square}$ and $^{\square}$ to find closed descriptions. At each step we compute more general pattern concepts by merging those of them that provide the minimal description length. If the computed pattern concept meets the requirements on the extent size, it is accepted to be a pre-pattern.

3.2 Pattern Mining: Compression of Pre-pattern Descriptions

The computed pre-patterns, i.e., pattern concepts, are compressed to find patterns. Each pattern concept is a pair (A, d), where A is a set of objects (pattern extent) and d is a vector of $|M|$ intervals of attributes, i.e., $d = \langle [l_i, r_i] \rangle_{i \in \{1,\ldots,|M|\}}$ (pattern intent). A pattern is a subvector of d, i.e., $\langle [l_i, r_i] \rangle_{i \in B}$, $B \subseteq M$, we denote it by d_B. The support of d_B is a number of objects it describes, i.e., $sup(d) = |d^{\square}| = |A|$,



(a) The sequential compression

$(\{g_2, g_3\}, \langle [2,3],[6,7],[1,1]\rangle)$, $L_1(\{g_2, g_3\}) = 0.35$
$(\{g_1, g_2, g_3\}, \langle [1,3],[5,7],[1,1]\rangle)$, $L_1(\{g_1, g_2, g_3\}) = 1.31$,
$(\{g_4, g_5\}, \langle [3,5],[5,6],[5,5]\rangle)$, $L_1(\{g_4, g_5\}) = 0.56$.

(b) The chosen pre-patterns (pattern concepts) that meet the requirement $q_{min} < |A| < q_{max}$ and their length by Formula 1

Figure 2: The first-stage compression: the compression sequence and the results.

$sup(d_B) \geq sup(d) = |A|$. At the second stage we search for the patterns (subvectors) that minimize an entropy-based description length.

For each attribute $m \in M$ we minimize the Shannon entropy of the probability distribution of the intervals that are used to describe objects G:

$$L(I_m) = -\sum_{i=1}^{|I_m|} p_i \log_2 p_i, \tag{2}$$

where $p_i = sup([l_i, r_i]) / \sum_{j=1}^{|I_m|} sup([l_j, r_j])$ is the probability of interval $[l_i, r_i]$ and I_m is a set of intervals of attribute m.

Let us consider the principles of the pre-pattern compression. The pseudocode is given in Algorithm 1. At the beginning, the set of initial attributes $I_{init}(m)$ is comprised of the shortest intervals $[v, v]$ that describe single objects (line 2) and the set of optimal intervals $I_{opt}(m)$ for each attribute $m \in M$ is empty (line 3), pre-patterns are ordered w.r.t. their lengths computed at the previous stage by Formula 1 (line 5).

The optimal intervals $I_{opt}(m)$ are computed in a greedy manner, at each iteration a new interval d_m (that is shared description by attribute m for objects A) is examined. In case where the replacement of initial intervals $I \in I_{init}(m)$ such that $I^{\square} \subseteq A$ minimizes the description length given in Formula 2, the initial intervals are removed and a new d_m is added to the set of optimal intervals $I_{opt}(m)$. We note that the condition in line 10 is true for the initial intervals $I = [l_I, r_I]$ included in the interval $d_m = [l_d, r_d]$, i.e., $[l_I, r_I] \subseteq [l_d, r_d]$. This compression is performed for each attribute separately (lines 6 - 21).

When all the intervals of pre-patterns are examined, the optimal intervals $I_{opt}(m)$, $m \in M$ are combined together, if they describe the same objects (line 22-34).

Example. Let us consider the principle of compression-based PM on an example given in Figure 1. At the first stage we merge the objects to get pre-patterns (pattern concepts) that minimize description length given in Formula 1. The dendrogram of the sequential merging is given in Figure 2, (a). We set the thresholds $q_{min} = 1$ and $q_{max} = 5$, the pre-patterns that meet these requirements are listed in Figure 2 (b).

The second stage, i.e., the pre-pattern compression is sketched in Figure 3. The pre-patterns (A, d) are examined in the ascending order of their lengths $L_1(A)$. At each iteration we try to compress a pre-pattern (A, d) by minimizing the entropy-

Input: Set of pre-patterns $clusters = \{\langle (A, d)\rangle \mid A \subset G, q_{min} < |A| < q_{max}\}$ and their lengths $L_1(\cdot)$ (see Formula 1)
Output: Set of patterns *patterns*

```
foreach m ∈ M do
    I_init(m) ← {({g}^□)_m | g ∈ G}
    I_opt(m) ← {∅}
end
rankedClusters ← sort({⟨(A, d), L_1(A)⟩ | (A, d) ∈ clusters})
foreach (A, d) ∈ rankedClusters do
    foreach d_m ∈ d, m ∈ M do
        I_del ← {∅}
        foreach I ∈ I_init(m) do
            if I^□ ⊆ A then
                I_del = I_del ∪ I
            end
        end
        L_old = L_2(I_opt(m) ∪ I_init(m))
        L_new = L_2(I_opt(m) ∪ (I_init(m) \ I_del) ∪ {d_m})
        if L_new < L_old then
            I_init(m) ← I_init(m) \ I_del
            I_opt(m) ← I_opt(m) ∪ A^□
        end
    end
end
patterns ← {∅}
foreach m ∈ M do
    foreach I_m ∈ I_init(m) do
        patterns ← I_m
    end
end
foreach p_1 ∈ patterns do
    foreach p_2 ∈ patterns do
        if p_1 ≠ p_2 and p_1^□ = p_2^□ then
            patterns = ⟨p_1, p_2⟩ ∪ patterns \ {p_1, p_2}
        end
    end
end
return patterns
```

Algorithm 1: MDL-based pattern computing

based description length separately for each attribute m_i, $i \in \{1, 2, 3\}$. The optimal intervals (column "Intervals") and the object values covered by the optimal intervals (column "Covering") are highlighted in red. As it can be seen, at the first iteration initial intervals $[2, 2]$, $[3, 3]$ of attribute m_1 and $[6, 6]$, $[7, 7]$ of attribute m_2 are replaced by the optimal ones by $[2, 3]_{m_1}$ and $[6, 7]_{m_2}$, respectively.

When all the pre-pattern descriptions are examined, we get the following set of optimal intervals: $[2, 3]_{m_1}$, $[6, 7]_{m_2}$, $[1, 1]_{m_3}$ and $[5, 5]_{m_3}$. We only need to combine them to get patterns. We consider the set of objects they describe and if the sets coincide we combine these intervals in one pattern. In our case all the intervals describe different object sets, i.e., $[2, 3]^{\square}_{m_1} = \{g_2, g_3, g_4\}$, $[6, 7]^{\square}_{m_2} = \{g_2, g_3, g_5\}$, $[1, 1]^{\square}_{m_3} = \{g_1, g_2, g_3\}$ and $[5, 5]^{\square}_{m_3} = \{g_4, g_5\}$, thus the final set of patterns is comprised of the four single-attribute intervals.

The proposed approach can be also considered as compression-based scaling for numerical attributes, where instead of quantiles or intervals of equal lengths we use

Iteration 0, initial settings

	Covering m_1	m_2	m_3	Intervals m_1	m_2	m_3	Pre-pattern descriptions
g_1	1	5	1	[1,1]	[5,5]	[1,1]	([2,3], [6,7], [1,1])
g_2	2	7	1	[2,2]	[6,6]	[5,5]	([1,3], [5,7], [1,1])
g_3	3	6	1	[3,3]	[7,7]		([3,5], [5,6], [5,5])
g_4	3	5	5	[5,5]			
g_5	5	6	5	*initial intervals*			

Iteration 1

	Covering m_1	m_2	m_3	Intervals m_1	m_2	m_3	Pre-pattern descriptions
	1	5	1	[1,1]	[5,5]	[1,1]	**([2,3], [6,7], [1,1])**
	2	7	1	[2,3]	[6,7]	[5,5]	([1,3], [5,7], [1,1])
	3	6	1	[5,5]			([3,5], [5,6], [5,5])
	3	5	5				
	5	6	5				

Iteration 2

g_1	1	5	1	[1,1]	[5,5]	[1,1]	([2,3], [6,7], [1,1])
g_2	2	7	1	[2,3]	[6,7]	[5,5]	**([1,3], [5,7],[1,1])**
g_3	3	6	1	[5,5]			([3,5], [5,6], [5,5])
g_4	3	5	5				
g_5	5	6	5				

Iteration 3

	1	5	1	[1,1]	[5,5]	[1,1]	([2,3], [6,7],[1,1])
	2	7	1	[2,3]	[6,7]	[5,5]	([1,3], [5,7], [1,1])
	3	6	1	[5,5]			**([3,5], [5,6], [5,5])**
	3	5	5				
	5	6	5				

Figure 3: The second stage (pre-pattern compression) of PM for dataset from Figure 1. At each iteration intervals of a candidate (the line in "Pre-pattern descriptions" column highlighted in bold) are examined. The optimal intervals are highlighted in red as well as fragment of dataset described by them (columns "Intervals" and "Covering", respectively).

an entropy- and similarity-based criterion.

4 Experiments

We use real world datasets from UCI repository [17] to evaluate pattern set quality. All datasets have class labels, i.e., they are appropriate for supervised learning. The labels are used to evaluate the ability of the proposed approach to reveal "true" classes of objects. The size of datasets and the number of classes are given in Table 1, column "Parameters of datasets".

In our study we evaluate the results from the Pattern Mining perspective rather than the compression quality. We examine *object covering*, *interpretability*, *diversity* and *interestingness* of pattern sets, other measures can be found in [18].

Interpretability of patterns is a subjective measure and it cannot be evaluated formally. In this study, under interpretability we mean the ease of getting some knowledge from patterns. We characterize the interpretability by
– the number of patterns $|patterns|$, a small set is easy to examine;
– the length of patterns (number of intervals), the shortest description is easy to "explain" in words;
– the number of compressed pre-patterns, i.e., pre-patterns with a non-empty compressed description, the number close to the number of classes testifies to the meaningfulness of the compressed pre-patterns.

Covering rate shows how well a pattern set retains information from the original dataset:
– object covering rate, $\left|\left\{p^{\square} \mid p \in patterns\right\}\right| / |G|$, the value close to 1 shows that the pattern set is able to describe most of objects, i.e., we have information about every object;
– cell covering rate, $\left|\left\{(g,m) \mid g \in p^{\square}, p \in patterns, m \in p_m\right\}\right| / (|G| \cdot |M|)$, the value close to 1 shows that the pattern set is able to describe most of cells in dataset.

Diversity shows how varied patterns are:

– overlapping rate, the average number of patterns that characterize a cell in a dataset, for diverse patterns the value is close to 1.

Interestingess of patterns is a difficult characteristic to evaluate. Under interestingness we mean the ability of patterns to distinguish hidden classes, i.e., when all patterns have been computed, we examine the rate of classes of the objects that correspond to a pattern :
– contrastness, $max_{l \in L} \left|\left\{g \mid class(g) = l, g \in p^{\square}\right\}\right| / \left|\left\{g \mid g \in p^{\square}\right\}\right|$, the rate of the major class label in a pattern; we study average, minimal and maximal values among pattern set *patterns*.

Dataset name	Parameters of datasets			Parameters of mined patterns								
	# objects	# attributes	# true classes	avg # attr. in patterns	# patterns	# compressed pre-patterns	obj. cover. rate	cell cover. rate	avg cell overlapping	min contrastness	avg contrastness	max contrastness
blood	748	4	2	3,83	23	6	0,98	0,98	1,33	0,72	0,78	0,85
breast	699	9	2	4,60	23	5	0,91	0,91	1,35	0,95	0,99	1,00
connectionist	208	60	2	43,00	129	3	1,00	0,86	1,00	0,54	0,57	0,61
glass	214	9	6	5,30	53	10	0,92	0,85	1,71	0,43	0,67	1,00
iris	150	4	3	2,57	18	7	1,00	0,94	1,54	0,77	0,94	1,00
mam. mass	961	5	2	3,00	24	8	1,00	0,91	1,99	0,68	0,81	0,90
user knowledge	403	5	5	4,33	13	3	0,99	0,91	1,00	0,57	0,72	0,98
vertebral, 2c	310	6	2	4,00	24	6	0,97	0,97	1,38	0,51	0,72	0,95
wholesale, channel	440	6	2	4,11	37	9	0,89	0,78	1,64	0,80	0,92	0,99
wine	178	13	3	7,00	49	7	0,95	0,83	1,42	0,90	0,97	1,00
average	**415,15**	**10,77**	**3,31**	**7,17**	**37,00**	**6,69**	**0,95**	**0,89**	**1,46**	**0,66**	**0,79**	**0,93**

Table 1: Quality of patterns for datasets.

As it can be seen from the table, the approach provides the fruitful results. The set of patterns is usually much smaller than the size of dataset – 37 patterns for datasets with the number of objects 415 on average. That illustrates a good compression and ensures that patterns will be easily examined by experts. More than that, the patterns contain quite enough attributes to ensure reasoning on them. When the number of attributes in a dataset is more than 7, the generated patterns keep about a half of attributes, when the attribute number is small, the patterns contain more than a half of them, the variable rate of retained attributes ensures the patterns will have enough attributes to find well-interpretable attribute collocations.

It should be noted, that the number of compressed pre-patterns (the pre-patterns that include at least one pattern) is a bit bigger than to the number of "true" classes. This (with the high values of *contrastness*) is indicative of the ability to identify true classes (and subclasses) of objects.

Being quite small, the pattern set is able to describe 95% of objects on average

and cover 89% of cells. The high coverage by a small number of patterns testifies to the high-quality compression.

The generated patterns are diverse, i.e., they describe different "data fragments", the average overlapping rate is 1.46. It means that every cell in dataset, that is covered by at least one pattern, is covered only by one pattern in more than a half of the cases.

We also emphasize that the contrastness of the pattern is high – around 0.79, the last value shows that being computed without any information about classes, patterns reconstruct them.

The experiments show that the proposed approach provides a good compression and generates high-quality pattern sets with characteristics ensuring fruitful interpretation by humans.

5 Conclusion

In this paper we presented a new approach to numerical Pattern Mining that returns a small subset of patterns with short description, thus providing a compression technique. The obtained pattern sets are non-redundant, well-interpretable (i.e., small and short) and can be used to identify in an unsupervised manner true classes of objects.

The proposed approach can be adapted to handle more complex data, such as sequences, trees, graphs, etc. by changing Pattern Structure descriptions and respective $\square$-operators. This extension, as well as the relationship of the proposed approach to R-trees [19], will be the subject of further research.

Acknowledgement

The work was supported by the Russian Science Foundation under grant 17-11-01294 and performed at National Research University Higher School of Economics, Moscow, Russia.

References

[1] Heikki Mannila, "Theoretical frameworks for data mining," *ACM SIGKDD Explorations Newsletter*, vol. 1, no. 2, pp. 30–32, 2000.

[2] Sergei O. Kuznetsov and Tatiana Makhalova, "On interestingness measures of formal concepts," *Information Sciences*, vol. 442, pp. 202–219, 2018.

[3] Eamonn Keogh, Stefano Lonardi, Chotirat Ann Ratanamahatana, Li Wei, Sang-Hee Lee, and John Handley, "Compression-based data mining of sequential data," *Data Mining and Knowledge Discovery*, vol. 14, no. 1, pp. 99–129, 2007.

[4] Eamonn Keogh, Li Keogh, and John C Handley, "Compression-based data mining," in *Encyclopedia of Data Warehousing and Mining, Second Edition*, pp. 278–285. IGI Global, 2009.

[5] Li Wei, John Handley, Nathaniel Martin, Tong Sun, and Eamonn Keogh, "Clustering workflow requirements using compression dissimilarity measure," in *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*. IEEE, 2006, pp. 50–54.

[6] Jilles Vreeken, Matthijs Van Leeuwen, and Arno Siebes, "Krimp: mining itemsets that compress," *Data Mining and Knowledge Discovery*, vol. 23, no. 1, pp. 169–214, 2011.

[7] Tatiana Makhalova, Sergei O. Kuznetsov, and Amedeo Napoli, "A first study on what MDL can do for FCA," in *CLA*, 2018, pp. 25–36.

[8] Tatiana Makhalova, Sergei O. Kuznetsov, and Amedeo Napoli, "MDL for FCA: is there a place for background knowledge?," in *CEUR*, 2018, pp. 45–56.

[9] Bernhard Ganter and Sergei O. Kuznetsov, "Pattern structures and their projections," in *International Conference on Conceptual Structures*. Springer, 2001, pp. 129–142.

[10] B Ganter and R Wille, "Formal concept analysis: Logical foundations," 1999.

[11] Mehdi Kaytoue, Sergei O Kuznetsov, and Amedeo Napoli, "Revisiting numerical pattern mining with formal concept analysis," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, 2011, vol. 22, p. 1342.

[12] Sergei O. Kuznetsov, "On stability of a formal concept," *Annals of Mathematics and Artificial Intelligence*, vol. 49, no. 1-4, pp. 101–115, 2007.

[13] Andrew Barron, Jorma Rissanen, and Bin Yu, "The minimum description length principle in coding and modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, 1998.

[14] Paul MB Vitanyi and Ming Li, *An introduction to Kolmogorov complexity and its applications*, vol. 34, Springer Heidelberg, 1997.

[15] Peter D Grünwald, In Jae Myung, and Mark A Pitt, *Advances in minimum description length: Theory and applications*, MIT press, 2005.

[16] Thomas M Cover and Joy A Thomas, *Elements of information theory*, John Wiley & Sons, 2012.

[17] Dua Dheeru and Efi Karra Taniskidou, "UCI machine learning repository," 2017.

[18] Dmitry I. Ignatov, Dmitry V. Gnatyshak, Sergei O. Kuznetsov, and Boris G. Mirkin, "Triadic formal concept analysis and triclustering: searching for optimal patterns," *Machine Learning*, vol. 101, no. 1-3, pp. 271–302, 2015.

[19] Yannis Manolopoulos, Alexandros Nanopoulos, Apostolos N Papadopoulos, and Yannis Theodoridis, *R-trees: Theory and Applications*, Springer Science & Business Media, 2010.