The Minimum Description Length Principle for Pattern Mining
A Survey

[v4.9]

Esther Galbrun
School of Computing, University of Eastern Finland
esther.galbrun@uef.fi

01/05/2022

This is about the Minimum Description Length (MDL) principle applied to pattern mining. The length of this description is kept to the minimum.

Mining patterns is a core task in data analysis and, beyond issues of efficient enumeration, the selection of patterns constitutes a major challenge. The MDL principle, a model selection method grounded in information theory, has been applied to pattern mining with the aim to obtain compact high-quality sets of patterns. After giving an outline of relevant concepts from information theory and coding, as well as of work on the theory behind the MDL and similar principles, we review MDL-based methods for mining various types of data and patterns. Finally, we open a discussion on some issues regarding these methods, and highlight currently active related data analysis problems.

1 Introduction

Our aim is to review the development of pattern mining methods based on and inspired from the Minimum Description Length (MDL) principle. Although this is an unrealistic goal, we strive for completeness in our coverage of these methods.

Before we go any further, let us explain more precisely what we consider, for the present purpose, to constitute *patterns*. Patterns are about repetitions. We adopt the point of view that patterns express the repeated presence in the data of particular items, attribute values or other discrete properties. We divide them into two main categories.

Itemsets are strict conjunctive patterns that require the occurrence of all involved items for the pattern to be considered as occurring in a transaction. For a given dataset, it is thus straightforward to determine where an itemset occurs and where it does not, that is, to compute the pattern's support. Vice versa, when it is known that an itemset occurs in a given transaction, no further information is necessary, as it implies that all items must be present. These concepts naturally extend beyond transactional data. However, such occurrence requirements are fairly strict, and it can be useful to consider more relaxed patterns. In particular, one might use disjunctions, allowing patterns to express a choice between involved items or attributes. Given a dataset, one can then still straightforwardly determine where a pattern holds. On the other hand, knowing that the pattern occurs no longer unambiguously provides information about which item or attribute is present. More or less additional information is needed, depending for instance on whether it is an inclusive or exclusive disjunction. We refer to such patterns that express the presence of a specific substructure in the data as *substructure patterns*.

As an alternative way to relax occurrence requirements, patterns might express that selected items or properties are typically present but not all need always occur, for instance by means of density thresholds. In that case, which data instances belong to the support of the pattern, or vice-versa where each item or property holds, needs to be specified explicitly as it is not directly implied. Rather than the occurrence of a specific substructure, it is then the homogeneity of repetitions within the area delimited by the selected instances and attributes that is of interest. Because such patterns can be seen as delineating homogeneous rectangles in the data, we refer to them as *block patterns*.

Furthermore, one is typically looking for a collection of patterns and might impose constraints on the overlap between them, that is, require that the patterns involve disjoint sets of attributes, characterise disjoint sets of instances, or both. In particular, the patterns might be required to form a partition of the data, dividing all instances and all attributes into disjoint subsets. Such a partitioning requirement is incompatible with a strict occurrence requirement in practice, in the sense that it is not in general possible to identify a collection of substructure patterns that forms a partition of the data. On the other hand, block patterns might be required to form a partition of the data, corresponding roughly to biclustering approaches, or they might be allowed to overlap, as in tiling approaches.

In summary, we adopt a rather broad definition of what constitute patterns, from itemsets to biclusters over discrete data. However, we stop short of considering clusters more in general as patterns. Clustering constitutes another important field of data mining beside—and partially overlapping with—pattern mining. There, the goal is to organise data instances into groups such that instances within the same group are similar to each other and dissimilar to instances in other groups. Clustering often handles continuous data, typically relying on a concept of distance. Here, we focus on formalisms and methods that are by nature more discrete.

The reader is expected to be familiar with common pattern mining tasks and techniques, but not necessarily with concepts from information theory and coding, of which we therefore give an overview in Section 2, before presenting the two main encoding strategies that use respectively substructure and block patterns. Background work is covered in Section 3. We start with the theory behind the MDL principle and similar principles. Then, we go over a few examples of uses of the principle in the adjacent fields of machine learning and natural language processing. We end with an overview of methods that involve practical compression as a data mining tool or that consider the problem of selecting itemsets. In Sections 4–7, we turn to the review of MDL-based methods for pattern mining proper. The methods are grouped first by the type of data, then by the type of patterns considered, as outlined in Figure 1. Our focus is on the various encodings designed for these different types of data and patterns, rather than on algorithmic issues related to searching the patterns. In Section 4, we start with one of the thickest branches, stemming from the Krimp algorithm for mining itemsets from transactional data. We continue with itemsets and with other types of patterns for tabular data in Section 5, followed by graphs and temporal data in Sections 6 and 7, respectively. Finally, we consider some discussion points, in Section 8, before highlighting related problems that have recently attracted research interest, in Section 9.

Sections contain lists of references ordered by publication year, to provide an better overview of the development in the corresponding sub-field. To keep things simple, even though sometimes related to different sub-fields, each publication is assigned to a single sub-field, that to which it is considered most relevant. For ease of reference, a complete bibliography ordered alphabetically is included at the end.In addition, the main characteristics and bibliographic details of publications from Sections 4–7 have been collected into a searchable table.[1]

2 The basics in a nutshell

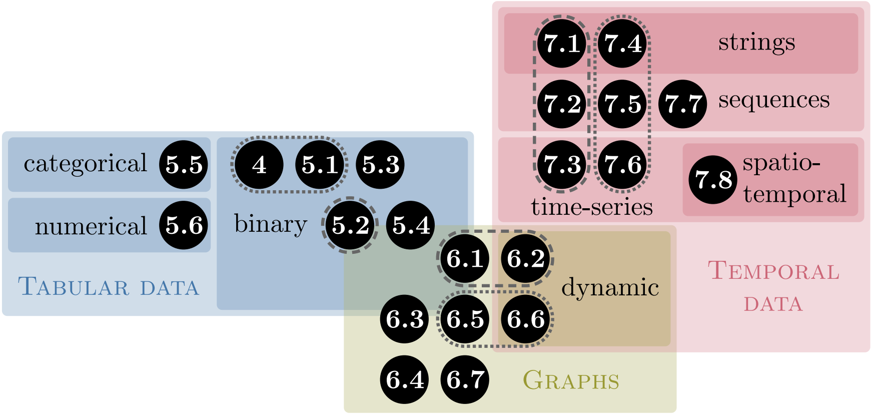
The Minimum Description Length (MDL) principle is a model selection method grounded in information theory. It can be seen as a practical variant of the Kolmogorov complexity, according to which the complexity of an object, for instance a dataset, is the length of the shortest computer program that outputs it. The idea is that regularities in the object can be exploited to make the program more concise. For example, the string that consists of a thousand repetitions of the word `baobab` can be produced with the following short program: `for i in range(1000): print('baobab')`. To output instead a string of the same length where the three letters appear in a completely random manner, we have no choice but to embed the string in the program, resulting in a much longer program. The second string is thus considered to be more complex than the first.

However, the Kolmogorov complexity cannot be computed in general. The MDL principle makes it practical by considering more restricted description methods rather than a universal programming language. Specifically, when working with the MDL principle, a class of models of the data is considered and the model that allows to most concisely represent the data is selected.

2.1 Encoding data

Clearly, this is akin to data encoding, where the aim is to map the input object to a sequence, referred to as its *description*, from which the original object can then be reconstructed. In practical scenarios of data encoding, the aim might be to efficiently and reliably either store the object on a storage medium or transmit it over a communication channel. The system of rules that dictates how the object is converted into the description and back is called a *code*. The processes of mapping the object to its description and of reconstructing it are called *encoding* and *decoding*, respectively. The considered storage or channel is typically binary, meaning that the object

[1] See `http://bib_table.html`.



4 Mining itemsets with Krimp & Co

5.1 More itemsets
5.2 Blocks in binary data
5.3 Formal Concept Analysis (FCA)
5.4 Boolean Matrix Factorisation (BMF)
5.5 Categorical data
5.6 Numerical data

6.1 Grouping nodes into blocks
6.2 Grouping nodes into blocks in dynamic graphs
6.3 Finding hyberbolic communities
6.4 The Map Equation
6.5 Identifying substructures
6.6 Identifying substructures in dynamic graphs
6.7 Finding pathways between nodes

- - - *block patterns*
······ *substructure patterns*

7.1 Finding haplotype blocks
7.2 Segmenting sequences
7.3 Segmenting timeseries
7.4 Mining substrings
7.5 Mining episodes from sequences
7.6 Mining motifs from timeseries
7.7 Mining periodic patterns
7.8 Trajectories

Figure 1: Organisation of Sections 4–7. MDL-based methods for pattern mining are grouped first by the type of data (tabular, graph or temporal), then by the type of patterns and strategies considered (blocks or substructures). Numbers refer to sections. The three main data types and their subtypes are represented by coloured shapes. Simple unlabelled graphs can be represented as binary matrices. Thus, some methods designed for binary data can be applied to them, and vice versa, some graph-mining methods in effect process binary data. The corresponding sections are therefore represented as lying at the intersection between binary and graph data. Dashed and dotted lines are used to group methods associated to the two main strategies (cf. Section 2.3). *Block-based* strategies are used to mine block patterns (also tiles and segments) that group together elements that are similarly distributed. On the other hand, *dictionary-based* strategies are used to mine substructure patterns (also motifs and episodes) that capture specific arrangements and co-occurrences between elements.

is mapped to a binary sequence, i.e. over the alphabet $\{0, 1\}$, whose length is hence measured in bits. There has been a lot of studies about communication through noisy channels, that is, when errors might be introduced into the transmitted sequence, and how to recover from it, but this is not of interest to us. Instead of noise-resistant codes, we focus purely on data compression, on obtaining compact descriptions. In general, data compression can be either lossless or lossy, depending whether the source object can be reconstructed exactly or only approximately.

Typically, encoding an object means mapping its elementary parts to binary codewords and concatenating them. Care must be taken to ensure that the resulting bit-string is decodable, that is, that it can be broken down back into the original codewords from which the parts can be recovered. For instance, imagine the data consist of the outcome of five throws of a pair of dice, i.e. a list of five integers in the interval $[1, 12]$. If we simply turn the values into their binary representations and concatenate them, we might not be able to reconstruct the list of integers. For example, there is no way to tell whether `1110110110` stands for `11 10 1 10 110`, i.e. $\langle 3, 2, 1, 2, 6\rangle$, or for `1 110 1 101 10`, i.e. $\langle 1, 6, 1, 5, 2\rangle$. To avoid this kind of confusion, we want our code to be such that there is a single unambiguous way to split an encoded string into codewords. One strategy is to use separators. For instance, we might represent each integer by as many `1`s, separated by `0`s, so that $\langle 3, 2, 1, 2, 6\rangle$ becomes `111011010110111111 0`.

More in general, this is where the *prefix-free property* becomes very useful. A *prefix-free code* (also confusingly known as prefix code or instantaneous code) is such that no extension of a codeword can itself be a codeword.

Fixed-length codes (a.k.a. *uniform codes*), that assign distinct codewords of the same length to every symbol in the input alphabet, clearly satisfy the *prefix-free property.* For an input alphabet consisting of n distinct symbols, the codewords must be of length $\lceil \log_2(n) \rceil$. Such a code minimises the worst-case codeword length, that is, the longest codeword is as short as possible. With such a code, every symbol is worth the same. Therefore it is a good option for pointing out an element among canonically ordered options under a uniform distribution, without a priori bias. For example, the sequence `baeaecdaeeccbc` over the five-letter alphabet $\langle \texttt{a}, \texttt{b}, \texttt{c}, \texttt{d}, \texttt{e} \rangle$ might be encoded in 42 bits, as

$$\texttt{001 000 100 000 100 010 011 000 100 100 010 010 001 010}\ .$$

When symbols are not uniformly distributed, using codewords of *different lengths* can result in codes that are more efficient on average. There are only so many short codewords available, so one needs to choose wisely what they are used for. Intuitively, symbols that are more frequent should be assigned shorter codewords. The *Kraft–McMillan inequality* (also known simply as Kraft inequality) gives a necessary and sufficient condition for the existence of a prefix-free code. Specifically, it states that, for a finite input alphabet $\mathcal{A}$, the codeword lengths for any prefix-free code C must satisfy

$$\sum_{x \in \mathcal{A}} 2^{-L_C(x)} \leq 1\ ,$$

where $L_C(x)$ denotes the length of the codeword assigned by C to symbol x. Vice versa, given codeword lengths satisfying this inequality, there exists a prefix-free code with these codeword lengths. Furthermore, if P is a probability distribution over a discrete input alphabet $\mathcal{A}$, there exists a prefix-free code C such that for all $x \in \mathcal{A}$, $L_C(x) = \lceil -\log_2\big(P(x)\big) \rceil$. A pair of related techniques to construct such a varying-length prefix-free code is commonly referred to as ***Shannon–Fano code***. Moreover, given an input alphabet $\mathcal{A}$ where each symbol x_i has an associated weight w_i that might represent, in particular, its frequency of occurrence, a code C is optimal if for any other code C' we have

$$\sum_{x_i \in \mathcal{A}} w_i L_C(x_i) \leq \sum_{x_i \in \mathcal{A}} w_i L_{C'}(x_i)\ .$$

Huffman's algorithm is an ingenious simple algorithm allowing to construct an optimal prefix-free code. Considering the example sequence `baeaecdaeeccbc` again, Huffman's algorithm would assign shorter codewords to the more frequent letters `a`, `c` and `e`, while ensuring that the prefix-free property is satisfied, allowing for instance to encode the sequence in just 31 bits, as

$$\texttt{011 00 11 00 11 10 010 00 11 11 10 10 011 10}\ .$$

Note that to use such a code, the alphabet and the associated probability distribution must be shared by the sender and the receiver. Indeed, it is not enough that the receiver is able to recover the transmitted codewords, the receiver must also know which symbol is represented by each codeword. In order to transmit a sequence of symbols, a simple way to proceed is therefore to first transmit the information about the distribution, then to transmit the actual sequence using Shannon–Fano coding, resulting in a ***two-part code***. For example, we would first need to transmit the occurrence counts of the five letters, or equivalently their assigned codewords, according to some agreed protocol, before sending the actual encoded 31-bit sequence.

Relying on a sequential prediction, or ***prequential***, strategy provides an alternative for coding that does not require the probability distribution to be known a priori. Simply put, the idea is to start with some initial probability distribution over the alphabet, e.g. uniform, and at each step encode the next element using a prefix-free code based on the current distribution, then update the probability distribution to account for this occurrence. These predictive plug-in codes have useful properties. In particular, the total code length does not depend on the order in which the elements are encoded, and is within a constant factor of the optimal.

Let us consider the sequence `baeaecdaeeccbc` as an example once more. We might start with occurrence counts initialised to one for all five letters, and run Huffman's algorithm to assign them codewords accordingly. Having agreed on this protocol, the sender and the receiver obtain the same initial codewords, without the sender first having to explicitly share the information about the distribution. The sender first transmits the codeword for letter `b`, which the receiver can correctly decode using the initial code. Both sides increment their occurrence count of `b` by one, and update their codewords by running Huffman's algorithm again. Note that at this point, `b` has the highest occurrence count and will thus be assigned a short codeword (2 bits). Next, the sender transmits the new codeword for `a`, which the receiver is able to correctly decode. The occurrence counts are incremented and the codewords updated, on both sides in parallel. And so on, until the entire sequence has been encoded. Crucially, as they apply the same updating protocol in parallel, relying only on information shared so far, sender and receiver maintain identical codes at every step. Later in the process, the occurrence counts of the letters approach their overall frequencies and the lengths of the assigned codewords converge towards those obtained using Shannon–Fano coding with prior knowledge of the distribution.

More recent advances in MDL and model selection theory introduced Bayesian and normalised maximum likelihood (NML) codes. Like prequential codes, both are ***one-part codes***. In contrast to *crude codes*, such *refined codes* remove the need to explicitly encode the parameters of the distribution, thereby avoiding the associated bias, and have useful properties, including optimality guarantees. In particular, the term *universal* is commonly used to refer to a code that, to put it simply, performs essentially as well as the best-fitting code for the input, for any possible input. This use of *universal* should not be confused with another common use, referring to codes for representing integers, which we present next. On the downside, refined one-part codes are not easily explained in terms of practical encoding and are not always computationally feasible.

Finally, a ***universal code for integers*** is a prefix-free code that maps non-negative integers to binary codewords. Such a code can be used to encode positive integers when the upper-bound cannot be determined a priori. *Elias codes*, which come in three variants—namely Elias gamma code, Elias delta code, and Elias omega code—are universal codes commonly used for this purpose. For instance, an integer $x \geq 1$ is encoded using the Elias gamma code as $\lfloor \log_2(x) \rfloor$ zeros followed by the binary representation of x. These codes penalise large values, since larger integers are assigned longer codewords.

2.2 Applying the MDL principle in pattern mining

It is important to note that when applying the MDL principle, **compression is used as a tool to compare models**, rather than as an end in itself. In other words, we do not care about actual descriptions, only about their lengths. Furthermore, we do not care about the absolute magnitude of the description achieved with a particular model as much as compared to those achieved with other candidate models. This has a few important consequences.

First, as the code does not need to be usable in practice, the requirement of integer code lengths can be lifted, allowing for finer comparisons. In particular, this means that for a discrete input alphabet $\mathcal{A}$ with probability distribution P, the most reasonable choice for assigning codewords is a prefix-free code C such that for all $x \in \mathcal{A}$, $L_C(x) = -\log_2\big(P(x)\big)$. This is often referred to as *Shannon–Fano coding*, although no underlying practical code is actually required, or even considered. Note that the entropy of P corresponds to the expected number of bits needed to encode in this way an outcome generated by P.

Second, elements that are necessary in order to make the encoding system work but are common to all candidate models might be left out, since they play no role in the comparison. Third, to ensure a fair comparison, only lossless codes are considered. Indeed, comparing models on the basis of their associated description lengths would be meaningless if we are unable to disentangle the savings resulting from a better ability to exploit the structure of the data, versus from increased distortion in the reconstruction. In some cases, this simply means that corrections that must be applied to the decoded data in order to achieve perfect reconstruction are considered as part of the description.

Though one-part codes might be used for components, at the highest level most proposed approaches use the *crude* ***two-part MDL***, which requires to first explicitly describe the model M, then describe the data using that model, rather than *refined one-part MDL*, which corresponds to using the entire model class $\mathcal{M}$ to describe the data. That is because the aim is not just to know how much the data can be compressed, but how that compression is achieved, by identifying the associated model. The overall description length is the sum of the lengths of the two parts, so the best model $M \in \mathcal{M}$ to explain the data D is the one that minimises $L(M, D) = L(M) + L(D \mid M)$, where L denotes description lengths in bits, as above. The two parts can be understood as capturing respectively the **complexity of the model** and the **fit of the model to the data**, and the MDL principle as a way to strike a balance between the two.

One can also view this from **the perspective of probabilistic modeling**. Consider a family of probability distributions parameterised by some set Θ, that is, $\mathcal{M} = \{p_\theta, \theta \in \Theta\}$, where each p_θ assigns probabilities to datasets. In addition, consider a distribution π over the elements of $\mathcal{M}$. In this context, in accordance with the Bayesian framework, the best explanation for a dataset D is the element of $\mathcal{M}$ minimising

$$-\log \pi(\theta) - \log p_\theta(D)\,,$$

where the two terms are the prior and the negative log likelihood of the data, respectively. When using a uniform prior, this means selecting the maximum likelihood estimator. Since codes can be associated to probability distributions, we can see the connection to the two-part MDL formulation above, where the term representing the description length of the model, $L(M)$, corresponds to the prior, and the term representing the description length of the data given the model, $L(D \mid M)$, corresponds to the likelihood.

The *Bayesian Information Criterion (BIC)* is a closely related model selection method that aims to find a balance between the fit of the model, measured in terms of the likelihood function, and the complexity of the

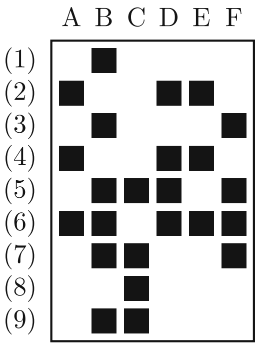


Figure 2: A toy binary dataset, with six columns (i.e. items) denoted A–F, nine rows (i.e. transactions) denoted (1)–(9), containing twenty-four ones (i.e. positive entries or item occurrences) represented as black squares.

model, measured in terms of the number of parameters k. Denoting as n the number of data points in D, it can be written as $k \log n - 2 \log p_\theta(D)$.

When applying the MDL principle to a pattern mining task, the models considered consist of patterns, capturing structure and regularities in the data. Depending on the type of data and the application at hand, one must decide what kind of patterns are relevant, i.e. ***(i)* a pattern language must be chosen**. The model class $\mathcal{M}$ then consists of all possible subsets of patterns from the chosen pattern language. Note that we generally use the term *model* to refer to a specific collection of patterns and the single corresponding probability distribution over the data, whereas from the statistical modeling perspective, *model* typically refers to a family of probability distributions. Next, ***(ii)* an encoding scheme must be designed**, i.e. a mechanism must be engineered to encode patterns of the chosen type and to encode the data by means of such patterns. Finally, ***(iii)* a search algorithm must be devised** to explore the space of patterns and identify the best set of patterns with respect to the encoding, i.e. the set of patterns that results in the shortest description length.

2.3 Dictionary-based vs. block-based strategies

The crude two-part MDL requires to explicitly specify the model (the $L(M)$ part). This means designing an ad-hoc encoding scheme to describe the patterns. This is both a blessing and a curse, because it gives leeway to introduce some bias, i.e. penalise properties of patterns that are deemed less interesting or useful than others by making them more costly in terms of code length. But it can involve some difficult design choices and lead to accusations of “putting your fingers on the scale”.

When considering substructure patterns, encoding the data using the model (the $L(D \mid M)$ part) can be fairly straightforward, simply replacing occurrences of the patterns in the data by their associated codewords. Knowing how many times each pattern X is used in the description of the data D with model M, denoted as $\text{usage}_{D,M}(X)$, X can be assigned a codeword of length

$$L(X) = -\log_2 \big(\frac{\text{usage}_{D,M}(X)}{\sum_{Y \in M} \text{usage}_{D,M}(Y)}\big)$$

using Shannon–Fano coding. The design choice of how to cover the data with a given set of patterns, dealing with possible overlaps between patterns in M, determines where each pattern is used, defining $\text{usage}_{D,M}(X)$. The requirement that the encoding be lossless means that elementary patterns, e.g. singleton itemsets, must be included in the model, to ensure complete coverage. In this case, encoding the model (the $L(M)$ part) consists in providing the mapping between patterns and codewords, typically referred to as the code table. That is, for each pattern in turn, describing it and indicating its associated codeword. Such a *code table* or ***dictionary-based* strategy**, which corresponds to frequent pattern mining approaches, is a common way to proceed.

An alternative strategy is to divide the data into blocks, that might or might not be allowed to overlap, each of which is associated with a specific probability distribution over the possible values and should be as homogeneous as possible. The values within each block are then encoded using a code optimised for the corresponding distribution. In that case, encoding the model consists in indicating the limits of each block and the associated probability distribution. Such a ***block-based* strategy** corresponds to segmentation, biclustering and tiling approaches.

These two main strategies, *dictionary-based* and *block-based*, that use respectively substructure and block patterns, are an important distinguishing factor that we use to categorise approaches, as depicted in Figure 1 with dotted and dashed lines, respectively. We further illustrate and contrast these strategies on a toy binary dataset shown in Figure 2. The dataset has six columns (i.e. items) denoted A–F, nine rows (i.e. transactions) denoted (1)–(9), and contains twenty-four ones (i.e. positive entries or item occurrences) represented as black squares. In particular, the approaches delineated here to illustrate the two strategies are based on the work of Siebes, Vreeken, and van Leeuwen (2006) and Smets and Vreeken (2012) (cf. Section 4), on one hand, and of Chakrabarti et al. (2004) (cf. Section 5.2), on the other hand.

Examples of **modeling the toy binary dataset using a dictionary-based approach** are provided in Figure 3. The simplest model is considered first *(i)*, which contains as its patterns all singleton itemsets from the dataset and is often referred to as the *standard code table (ST)*. A non-trivial code table is considered next *(ii)*. In both cases, the model is represented as a code table associating patterns, here itemsets, to codewords. Encoding the model means encoding each pair. On one hand, itemsets are specified by listing the items they contain. This uses Shannon–Fano coding over the alphabet of items associated to their frequency of occurrence in the data, which is assumed to be shared knowledge. On the other hand, codewords are assigned using Shannon–Fano coding over the alphabet of itemsets included in the code table, associated to their usage.

In Figure 3, the prefix-free codewords assigned to items and to itemsets are represented by coloured blocks in shades of green and blue, respectively. The width of a block is proportional to the length of the represented codeword. For instance, the first row of the non-trivial code table (bottom-left quadrant of Figure 3) specifies the first itemset in the code table, in this case ADE, listing the three items that constitute it, A, D and E, using the items' codewords (green blocks), then provides the codeword assigned to ADE by the code based on the usage of itemsets selected in this model (blue block).

Note that for the standard code table, these two prefix-free codes are the same because the standard code table consists of all singleton items and their usage is precisely their frequency of occurrence in the data. Therefore, all codewords in the top half of Figure 3 are represented as green blocks. For the same reason, the length of the codeword associated to item x by the first code, which does not depend on the code table, is denoted as $L_{ST}(x)$, whereas the length of the codeword associated to itemset X by the second code, which is different for different code tables, is denoted as $L_{CT}(X)$.

Then, encoding the data using the model simply means replacing itemset occurrences by the corresponding codewords.

In summary, the overall description length can be computed as

$$\begin{aligned} L(M, D) &= \qquad\qquad L(M) \qquad\qquad + \qquad L(D \mid M) \\ &= \sum_{X \in M} \underbrace{\sum_{x \in X} L_{ST}(x)}_{\text{itemset}} + \underbrace{L_{CT}(X)}_{\text{codeword}} + \underbrace{\sum_{X \in M} \text{usage}(X) \cdot L_{CT}(X)}_{\text{listing itemsets occurrences}} . \end{aligned} \tag{1}$$

In this example, the standard code table and the non-trivial code table yield an overall description length of 92.530 bits and 63.333 bits, respectively. By identifying items that occur frequently together, the latter results in a shorter description length, and one can therefore conclude that it constitutes a better model for the dataset, according to the MDL criterion.

A similar approach can be built by replacing static Shannon–Fano coding with dynamic prequential coding. In that case, encoding the model only requires to list the itemsets, i.e. only the first column of the code table is needed. Then, the data is encoded by enumerating the itemset occurrences using prequential coding (cf. Section 2.1). In practise, the process of updating the codewords does not actually need to be carried through. Luckily, the description length of the data can be calculated with a (not so simple) formula that does not depend on the order in which the itemsets are transmitted (see Budhathoki and Vreeken, 2015).

Examples of **modeling the toy binary dataset using a block-based approach** are provided in Figure 4. The simplest model is considered first *(i)*, which consists of a single block. A non-trivial model dividing the dataset into six blocks is considered next *(ii)*. The approach illustrated here requires the patterns to form a partition of the data. Therefore, any considered model consists of a set of non-overlapping blocks covering the entire dataset and, more specifically, is such that the rows and the columns are divided into disjoint groups. This requirement means that each row and each column belongs to exactly one group, a fact that can be exploited when designing the encoding, as explained below. Each block is associated to a specific probability distribution over the values occurring within it. In this case, the values are zero and one, corresponding to a Bernoulli distribution. Encoding the model means specifying the groups of rows and of columns that define the blocks, and the probability associated to each one. In this approach, all blocks are induced by a unique partition of the rows and of the columns, which can be specified by defining an order over the rows (resp. columns) and indicating the number of rows (resp. columns) in each group. Finally, the number of ones in each block is transmitted, allowing to compute the corresponding probability. This is achieved through a combination of fixed-length and universal codes.

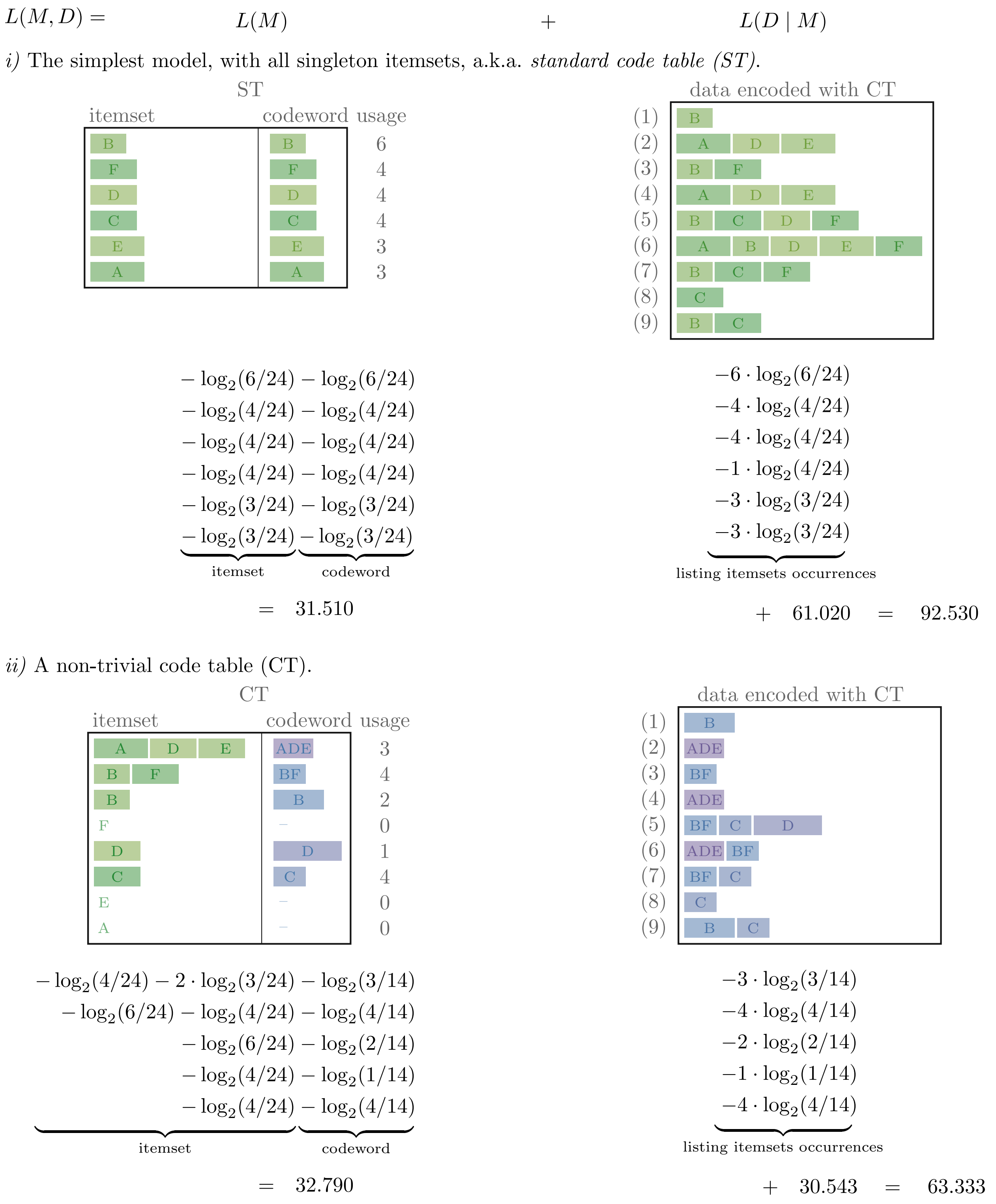
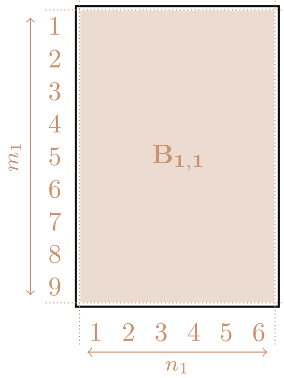
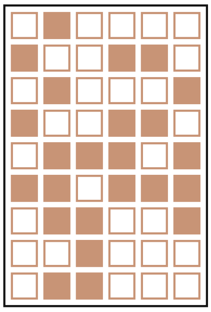


Figure 3: Dictionary-based strategy, examples on the toy binary dataset of Figure 2. The model M consists of a code table associating patterns, here itemsets, to codewords (left). The data D is encoded using the model by replacing occurrences of the itemsets by the associated codewords (right). Coloured blocks represent prefix-free codewords assigned to items (green) and itemsets (blue), their width is proportional to the code length.

$$L(M,D) = \qquad L(M) \qquad + \qquad L(D \mid M)$$

i) The simplest model, with a single block.





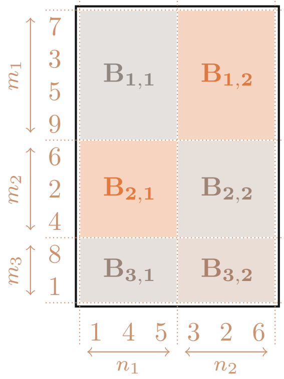
$$\underbrace{\mathrm{L}_{\mathbb{N}}(9)}_{m=9\text{ rows}} + \underbrace{9\cdot\log_2(9)}_{<1,2,3,4,5,6,7,8,9>} + \underbrace{\mathrm{L}_{\mathbb{N}}(6)}_{n=6\text{ cols}} + \underbrace{6\cdot\log_2(6)}_{<1,2,3,4,5,6>}$$

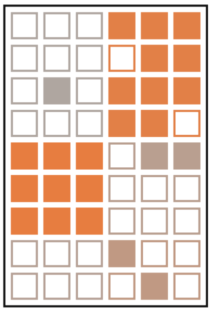
$$+ \underbrace{\mathrm{L}_{\mathbb{N}}(1)}_{k=1\text{ row group}} + \underbrace{0}_{(<9>)} + \underbrace{\mathrm{L}_{\mathbb{N}}(1)}_{l=1\text{ col group}} + \underbrace{0}_{(<6>)}$$

$$+\underbrace{\log_2(9\cdot 6+1)}_{24\text{ ones in }B_{1,1}} \qquad \underbrace{-30\cdot\log_2(30/54) - 24\cdot\log_2(24/54)}_{B_{1,1}:<0,1,0,\ldots,0,1,1,0,0,0>}$$

$$= \quad 64.686 \qquad + \quad 53.518 \quad = \quad 118.204$$

ii) A non-trivial model partitioning the dataset into six blocks.





$$\underbrace{-11\cdot\log_2(11/12) - 1\cdot\log_2(1/12)}_{B_{1,1}:<0,0,0,0\ldots0,0,0,0>}$$

$$\underbrace{-2\cdot\log_2(2/12) - 10\cdot\log_2(10/12)}_{B_{1,2}:<1,1,1,0\ldots1,1,1,0>}$$

$$\underbrace{\mathrm{L}_{\mathbb{N}}(9)}_{m=9\text{ rows}} + \underbrace{9\cdot\log_2(9)}_{<7,3,5,9,6,2,4,8,1>} + \underbrace{\mathrm{L}_{\mathbb{N}}(6)}_{n=6\text{ cols}} + \underbrace{6\cdot\log_2(6)}_{<1,4,5,3,2,6>} \qquad \underbrace{-0\cdot\log_2(0/9) - 9\cdot\log_2(9/9)}_{B_{2,1}:<1,1,1,1,1,1,1,1,1>}$$

$$+ \underbrace{\mathrm{L}_{\mathbb{N}}(3)}_{k=3\text{ row groups}} \underbrace{+\log_2(7)+\log_2(4)}_{<4,3(,2)>} + \underbrace{\mathrm{L}_{\mathbb{N}}(2)}_{l=2\text{ col groups}} \underbrace{+\log_2(5)}_{<3(,3)>} \qquad \underbrace{-7\cdot\log_2(7/9) - 2\cdot\log_2(2/9)}_{B_{2,2}:<0,1,1,0,0,0,0,0,0>}$$

$$+\underbrace{\log_2(4\cdot 3+1)}_{1\text{ one in }B_{1,1}} + \underbrace{\log_2(4\cdot 3+1)}_{10\text{ ones in }B_{1,2}} + \underbrace{\log_2(3\cdot 3+1)}_{9\text{ ones in }B_{2,1}} \qquad \underbrace{-6\cdot\log_2(6/6) - 0\cdot\log_2(0/6)}_{B_{3,1}:<0,0,0,0,0,0>}$$

$$+\underbrace{\log_2(3\cdot 3+1)}_{2\text{ ones in }B_{2,2}} + \underbrace{\log_2(2\cdot 3+1)}_{0\text{ one in }B_{3,1}} + \underbrace{\log_2(2\cdot 3+1)}_{2\text{ ones in }B_{3,2}} \qquad \underbrace{-4\cdot\log_2(4/6) - 2\cdot\log_2(2/6)}_{B_{3,2}:<1,0,0,0,1,0>}$$

$$= \quad 88.279 \qquad + \quad 25.154 \quad = \quad 113.433$$

Figure 4: Block-based strategy, examples on the toy binary dataset of Figure 2. The model M partitions the dataset D into blocks, each associated to a specific probability distribution over the values (left), which is used to encode the entries (right). More intense shades of orange represent higher probabilities of ones within the corresponding block.

Next, we delve deeper into the details of the encoding scheme, to illustrate the choices that are involved in its design. We let

- m and n denote respectively the number of rows and columns in the dataset,
- k and l denote respectively the number of row and column groups,
- m_i and n_j denote respectively the number of rows and columns in block $B_{i,j}$,
- $\gamma_v(B_{i,j})$ denote the number of entries in block $B_{i,j}$ equal to v, and
- $\mathrm{L}_{\mathbb{N}}(x)$ denote the MDL optimal universal code length for integer x.[2]

The formula for computing the overall description length is provided in Equation 2. The number of rows and the number of columns are transmitted using universal coding, since these values are not bounded a priori, and the order of rows (resp. of columns) is then specified by listing row (resp. column) identifiers in turn, using a fixed-length code (cf. part (a) of Equation 2). In fact, this part of the encoding is independent of the model and has a constant length for a given dataset. Therefore, it does not impact the comparison and can be ignored. The number of row groups k (resp. column groups l) could be transmitted using a fixed-length code $\log_2(m)$ (resp. $\log_2(n)$), since there cannot be more groups than there are rows (resp. columns). However, universal coding is used instead, to favour partitions with fewer groups. Then, assuming that the numbers of rows and of columns in the groups are sorted and transmitted by decreasing order, upper bounds m_i^* and n_j^* on m_i and n_j, respectively, can be derived given shared knowledge since already transmitted values constrain the remaining ones: $m_i^* = \big(\sum_{t=i}^{k} m_t\big) - k + i$, for $i = 1, \dots k-1$ and $n_j^* = \big(\sum_{t=j}^{l} n_t\big) - l + j$, for $j = 1, \dots l-1$.

These upper bounds are used to transmit the numbers of rows and of columns in the groups with fixed-length codes (cf. part (b) of Equation 2). Also the number of ones in each block can be transmitted using a fixed-length code, since it takes value between zero and the number of entries in the block (cf. part (c) of Equation 2).

The data is then encoded using the model. This is done by listing the entries in each block with a prefix-free code adjusted to the probability distribution within the block (cf. part (d) of Equation 2).

In summary, the overall description length can be computed as

$$
\begin{aligned}
L(M,D) &= \qquad\qquad L(M) \qquad\qquad + \qquad\qquad L(D \mid M) \\
&= \underbrace{\mathrm{L}_{\mathbb{N}}(m)}_{\text{nb. rows}} + \underbrace{m \cdot \log_2(m)}_{\text{ordered rows}} + \underbrace{\mathrm{L}_{\mathbb{N}}(n)}_{\text{nb. cols}} + \underbrace{n \cdot \log_2(n)}_{\text{ordered cols}} \qquad \Big\}\ (a) \\
&\quad + \underbrace{\mathrm{L}_{\mathbb{N}}(k)}_{\text{nb. row groups}} + \underbrace{\sum_{i=1}^{k-1} \log_2(m_i^*)}_{\text{nb. rows in each group}} \\
&\quad + \underbrace{\mathrm{L}_{\mathbb{N}}(l)}_{\text{nb. col groups}} + \underbrace{\sum_{j=1}^{l-1} \log_2(n_j^*)}_{\text{nb. cols in each group}} \qquad \Bigg\}\ (b) \\
&\quad + \underbrace{\sum_{i=1}^{k} \sum_{j=1}^{l} \log_2(m_i \cdot n_j + 1)}_{\text{nb. ones in each block}} \qquad + \qquad \Big\}\ (c) \\
(d)\ \Big\{ &\qquad \underbrace{\sum_{i=1}^{k} \sum_{j=1}^{l} \sum_{v \in \{0,1\}} -\gamma_v(B_{i,j}) \cdot \log_2\big(\frac{\gamma_v(B_{i,j})}{m_i \cdot n_j}\big)}_{\text{listing entries in each block}} \ .
\end{aligned}
\tag{2}
$$

In Figure 4, shades of orange are used to represent probability distributions within the blocks, with more intense shades representing higher probabilities of ones. In this example, the single-block model and the six-block model yield an overall description length of 118.204 bits and 113.433 bits, respectively. By partitioning the dataset into blocks that are particularly dense or particularly sparse, the latter results in a shorter description length, and one can therefore conclude that it constitutes a better model for the dataset, according to the MDL criterion.

Removing the partition constraint and allowing overlaps between the block patterns would require to modify the encoding since we could no longer assume that each row and each column belongs to a single group.

[2] $\mathrm{L}_{\mathbb{N}}(x)$ is defined for $x > 1$ as $\mathrm{L}_{\mathbb{N}}(x) = \log_2(x) + \log_2 \log_2(x) + \dots + \log_2(c_0)$, with the value of c_0 set so as to satisfy the Kraft inequality. Specifically, we let $c_0 = 2.86507$.

2.4 Algorithms

The MDL principle provides a basis for designing a score for patterns, but no way to actually find the best collection of patterns with respect to that score. The space of candidate patterns, and even more so the space of candidate pattern sets, is typically extremely large, if not infinite, and rarely possesses any useful structure. Hence, exhaustive search is generally infeasible, heuristic search algorithms are employed, and one must be satisfied with finding a good set of patterns rather than an optimal one. Mainly, algorithms can be divided between *(i)* approaches that generate a large collection of patterns then (iteratively) select a small set out of it, and *(ii)* approaches that generate candidates on-the-fly, typically in a levelwise manner, possibly in an anytime fashion. The former approaches are typically less efficient, since they generate many more candidates than necessary, but constitute a useful basis for building a proof-of-concept. Because recomputing costs following the addition or removal of a candidate pattern is often prohibitively expensive, an important component of the heuristic search algorithms consists in efficiently and accurately bounding these costs.

With a dictionary-based strategy, the simplest model consists of all separate basic elements. The search for candidates can thus start from this model and progressively combine elements. Vice versa, with a block-based strategy, the simplest model consists of a single block covering the entire dataset. The search for candidates can thus start from this model and progressively split elements. Intuitively, the two main strategies lend themselves respectively to bottom-up and top-down iterative exploration algorithms.

In summary, to apply the MDL principle to a pattern mining task, one might proceed in three main steps. First, define the pattern language, deciding what constitutes a structure of interest given the data and application considered. Second, define a suitable encoding scheme, designing a system to encode the patterns and to encode the data using such patterns. Third, design a search algorithm, allowing to identify in the data a collection of patterns that yields a good compression under the chosen encoding scheme.

Our main focus here is on the second step, the design of an encoding scheme for the different types of patterns, which is the core of the MDL methodology and its distinctive ingredient. For the most part, we do not discuss search algorithms and are not concerned by issues of performance, which are more generic aspects of pattern mining methodologies.

3 Theoretical and conceptual background

The MDL principle is maybe the most popular among several similar principles rooted in information theory. The 1948 article by Shannon is widely seen as the birth certificate of information theory. The textbooks by Stone (2013) and by Cover and Thomas (2012) provide respectively an accessible introduction and a more detailed account of information theory and related concepts. The textbook of M. Li and Vitányi (2019), on the other hand, focuses on Kolmogorov complexity. The tutorial by Csiszár and Shields (2004) covers applications of information theory in statistics and discusses the MDL principle in its last chapter.

Shannon, Claude E. (1948). “A Mathematical Theory of Communication”. In: *Bell System Technical Journal* 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

Csiszár, Imre and Paul C. Shields (2004). “Information Theory and Statistics: A Tutorial”. In: *Foundations and Trends in Communications and Information Theory* 1.4, pp. 417–528. DOI: 10.1561/0100000004.

Cover, Thomas M. and Joy A. Thomas (2012). *Elements of information theory.* John Wiley & Sons.

Stone, James V. (2013). *Information Theory: A Tutorial Introduction.* Sebtel Press.

— (2018). *Information Theory: A Tutorial Introduction.* arXiv: 1802.05968.

Li, Ming and Paul Vitányi (2019). *An Introduction to Kolmogorov Complexity and Its Applications.* 4th ed. Springer. DOI: 10.1007/978-3-030-11298-1.

3.1 The Minimum Description Length (MDL) principle

The introduction of the Minimum Description Length principle can be dated back to the seminal paper by Rissanen in 1978. Works collected in (Grünwald, Myung, and Pitt, 2005) present the theoretical foundations of the principle, as well as later advances and applications. The textbook by Grünwald (2007) is often regarded as the major reference about the MDL principle.

More recently, Grünwald and Roos (2019) present the MDL principle from the perspective of probabilistic modeling, without resorting to information theory. They review recent theoretical developments, which allow to see MDL as a generalisation of both penalised likelihood and Bayesian approaches. Vitányi and M. Li (2000) (also

M. Li and Vitányi, 1995; Vitányi and M. Li, 1999) draw also parallels between the Bayesian framework and the MDL principle.

Rissanen, Jorma (1978). "Modeling by shortest data description". In: *Automatica* 14.5, pp. 465–471. DOI: 10.1016/0005-1098(78)90005-5.

— (1983). "A Universal Prior for Integers and Estimation by Minimum Description Length". In: *The Annals of Statistics* 11.2, pp. 416–431.

— (1989). *Stochastic complexity in statistical inquiry.* World Scientific.

Li, Ming and Paul Vitányi (1995). "Computational machine learning in theory and praxis". In: *Computer Science Today: Recent Trends and Developments.* Ed. by Jan van Leeuwen. Springer, pp. 518–535. DOI: 10.1007/BFb0015264.

Barron, Andrew, Jorma Rissanen, and Bin Yu (1998). "The minimum description length principle in coding and modeling". In: *IEEE Transactions on Information Theory* 44.6, pp. 2743–2760. DOI: 10.1109/18.720554.

Vitányi, Paul and Ming Li (1999). *Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity.* arXiv: cs/9901014.

— (2000). "Minimum description length induction, Bayesianism, and Kolmogorov complexity". In: *IEEE Transactions on Information Theory* 46.2, pp. 446–464. DOI: 10.1109/18.825807.

Hansen, Mark H. and Bin Yu (2001). "Model Selection and the Principle of Minimum Description Length". In: *Journal of the American Statistical Association* 96.454, pp. 746–774. DOI: 10.1198/016214501753168398.

Lee, Thomas C. M. (2001). "An Introduction to Coding Theory and the Two-Part Minimum Description Length Principle". In: *International Statistical Review* 69.2, pp. 169–183. DOI: 10.1111/j.1751-5823.2001.tb00455.x.

Grünwald, Peter D. (2004). *A Tutorial Introduction to the Minimum Description Length Principle.* arXiv: math/0406077.

Grünwald, Peter D., Jay Injae Myung, and Mark A. Pitt (2005). *Advances in Minimum Description Length: Theory and applications.* Neural Information Processing. The MIT Press, p. 372.

Rissanen, Jorma (2005). *An Introduction to the MDL Principle.* Technical report. Helsinki Institute for Information Technology (HIIT).

Grünwald, Peter D. (2007). *The Minimum Description Length Principle.* MIT Press.

Rissanen, Jorma (2007). *Information and Complexity in Statistical Modeling.* Information Science and Statistics. Springer, pp. 97–102. DOI: 10.1007/978-0-387-68812-1.

Rooij, Steven de and Peter D. Grünwald (2011). "Luckiness and Regret in Minimum Description Length Inference". In: *Philosophy of Statistics.* Ed. by Prasanta S. Bandyopadhyay and Malcolm R. Forster. Vol. 7. Handbook of the Philosophy of Science. North-Holland, pp. 865–900. DOI: 10.1016/B978-0-444-51862-0.50029-0.

Roos, Teemu (2016). "Minimum Description Length Principle". In: *Encyclopedia of Machine Learning and Data Mining.* Ed. by Claude Sammut and Geoffrey I. Webb. Springer, pp. 1–4. DOI: 10.1007/978-1-4899-7502-7_894-1.

Grünwald, Peter D. and Teemu Roos (2019). "Minimum description length revisited". In: *International Journal of Mathematics for Industry*, p. 1930001. DOI: 10.1142/S2661335219300018.

Vreeken, Jilles and Kenji Yamanishi (2019). "Modern MDL meets Data Mining Insights, Theory, and Practice [Tutorial]". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'19.* ACM, pp. 3229–3230. DOI: 10.1145/3292500.3332284.

3.2 Other information-theoretic principles for model selection

Other minimisation principles based on information theory, and often closely related to Kolmogorov complexity or its extensions (see e.g. Vereshchagin and Vitányi, 2004), were being developed around the same time as the MDL principle. Chaitin (1975) suggested to refer to this emergent field bringing together Shannon's information theory and Turing's computational complexity as Algorithmic Information Theory (AIT).

In 1964, Solomonoff proposed the concept of algorithmic probability and a theory of inductive inference. This theory inspired the development of the MDL principle, as well as, independently, the development of the minimal representation criterion by Segen and Sanderson in 1979 (also Segen, 1989; Segen, 1990).

The MDL principle was also partly inspired from the Minimum Message Length (MML) principle, introduced by Wallace and Boulton in 1968 (also Wallace, 2005). The two principles have some important conceptual differences but often lead to similar results. They are discussed and compared by Lanterman (2001), together with other related approaches. Several applications and discussions of the principles are presented as contributions to the 1996 conference on Information, Statistics and Induction in Science (Dowe, Korb, and Oliver, 1996).

More recently, Tishby, Pereira, and Bialek (2000) introduced the Information Bottleneck (IB) method (see also Slonim, 2002), arguing that it more appropriately focuses on *relevant* information than previously formulated principles.

Solomonoff, R. J. (1964a). "A formal theory of inductive inference. Part I". In: *Information and Control* 7.1, pp. 1–22. DOI: 10.1016/S0019-9958(64)90223-2.

— (1964b). "A formal theory of inductive inference. Part II". In: *Information and Control* 7.2, pp. 224–254. DOI: 10.1016/S0019-9958(64)90131-7.

Wallace, Christopher S. and D. M. Boulton (1968). "An Information Measure for Classification". In: *The Computer Journal* 11.2, pp. 185–194. DOI: 10.1093/comjnl/11.2.185.

Chaitin, Gregory J. (1975). "A Theory of Program Size Formally Identical to Information Theory". In: *Journal of the ACM* 22.3, pp. 329–340. DOI: 10.1145/321892.321894.

Segen, Jakub and A. C. Sanderson (1979). "A minimal representation criterion for clustering". In: *Proceedings of the 12th Annual Symposium on the Interface: Computer Science and Statistics*, pp. 332–334.

Segen, Jakub (1989). "Incremental Clustering by Minimizing Representation Length". In: *Proceedings of the Sixth International Workshop on Machine Learning.* Morgan Kaufmann, pp. 400–403. DOI: 10.1016/B978-1-55860-036-2.50101-6.

— (1990). "Graph clustering and model learning by data compression". In: *Proceedings of the Seventh International Conference on Machine Learning, ICML'90.* Morgan Kaufmann, pp. 93–101.

Dowe, David L., Kevin B. Korb, and Jonathan J. Oliver (1996). *Proceedings of the Conference on Information, Statistics and Induction in Science, ISIS'96.* World Scientific. DOI: 10.1142/9789814530637.

Tishby, Naftali, Fernando C Pereira, and William Bialek (2000). *The Information Bottleneck Method.* arXiv: physics/0004057.

Lanterman, Aaron D. (2001). "Schwarz, Wallace, and Rissanen: Intertwining Themes in Theories of Model Selection". In: *International Statistical Review* 69.2, pp. 185–212. DOI: 10.1111/j.1751-5823.2001.tb00456.x.

Slonim, Noam (2002). "The information bottleneck: Theory and applications". PhD thesis. The Hebrew University.

Vereshchagin, Nikolai and Paul Vitányi (2004). "Kolmogorov's structure functions and model selection". In: *IEEE Transactions on Information Theory* 50.12, pp. 3265–3290. DOI: 10.1109/TIT.2004.838346.

Wallace, Christopher S. (2005). *Statistical and inductive inference by minimum message length.* Springer.

3.3 General considerations on simplicity, parsimony, and modeling

Several authors have contributed to the discussion on conceptual issues related to complexity in modeling.

Among them, Davis (1996), and later Robinet, Lemaire, and Gordon (2011), examine issues of model selection and parsimony, in relation to human cognition. Rathmanner and Hutter (2011) propose a rather broad but not overly technical discussion of induction in general and Solomonoff's induction in particular, as well as several associated topics. Lattimore and Hutter (2013) discuss algorithmic information theory in the context of the *no free lunch theorem*, which, simply put, posits that an algorithm that performs well on particular problems must pay for it with reduced performance on other problems, and *Occam's razor*, a general rule favouring simplicity. Domingos (1999) exposes misconceptions and misuses of Occam's razor in model selection.

Bloem, Rooij, and Adriaans (2015) (also Bloem, 2016) suggest to use *sophistication* as an umbrella term for various concepts related to the amount of information contained in data and discuss different approaches that have been proposed for measuring it.

Fürnkranz, Kliegr, and Paulheim (2020) present an insightful investigation into the perception of rule-based models by analysts, highlighting that shorter is not always better.

Davis, Mark (1996). *The Predictive Paradigm – Compression and Model Bias in Human Cognition.* Technical report.

Domingos, Pedro (1999). "The Role of Occam's Razor in Knowledge Discovery". In: *Data Mining and Knowledge Discovery* 3.4, pp. 409–425. DOI: 10.1023/A:1009868929893.

Rathmanner, Samuel and Marcus Hutter (2011). "A Philosophical Treatise of Universal Induction". In: *Entropy* 13.6, pp. 1076–1136. DOI: 10.3390/e13061076.

Robinet, Vivien, Benoît Lemaire, and Mirta B. Gordon (2011). "MDLChunker: A MDL-Based Cognitive Model of Inductive Learning". In: *Cognitive Science* 35.7, pp. 1352–1389. DOI: 10.1111/j.1551-6709.2011.01188.x.

Lattimore, Tor and Marcus Hutter (2013). "No Free Lunch versus Occam's Razor in Supervised Learning". In: *Proceedings of the Ray Solomonoff 85th Memorial Conference, Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence.* Springer, pp. 223–235. DOI: 10.1007/978-3-642-44958-1_17.

Bloem, Peter, Steven de Rooij, and Pieter Adriaans (2015). "Two Problems for Sophistication". In: *Proceedings of the 26th International Conference on Algorithmic Learning Theory, ALT'15*, pp. 379–394. DOI: 10.1007/978-3-319-24486-0_25.

Bloem, Peter (2016). “Single sample statistics: Exercises in learning from just one example”. PhD thesis. Universiteit van Amsterdam.

Fürnkranz, Johannes, Tomáš Kliegr, and Heiko Paulheim (2020). “On cognitive preferences and the plausibility of rule-based models”. In: *Machine Learning* 109.4, pp. 853–898. DOI: 10.1007/s10994-019-05856-5.

3.4 Compression and Data Mining (DM)

Various approaches using practical data compression as a tool for data mining have been proposed and discussed.

For instance, Keogh, Lonardi, and Ratanamahatana (2004) (also Keogh, Lonardi, Ratanamahatana, et al., 2007) present the Compression-based Dissimilarity Measure (CDM), that evaluates the relative gain when compressing two strings concatenated rather than separately. Inspired from Kolmogorov complexity, the measure is used while mining timeseries, to fight against large numbers of parameters in the algorithms. Similarly, Cilibrasi and Vitányi (2005) define a *Normalized Compression Distance*, which they then use for clustering.

Simovici (2013) (also Simovici et al., 2015) proposes to evaluate the presence of patterns using practical data compression. The aim is to detect whether patterns are present, not to find them.

From a more conjectural perspective, Faloutsos and Megalooikonomou (2007) argue that the strong connection between data mining, compression and Kolmogorov complexity means there is little hope for a unifying theory of data mining.

The term *compression* as used by Chandola and Kumar (2007) is a metaphor rather than a practical tool. The proposed approach for mining patterns from tables of categorical attributes relies on a pair of ad-hoc scores that intuitively measure how much the considered collection of patterns allows to compact the data, and how much information is lost, respectively.

Keogh, Eamonn, Stefano Lonardi, and Chotirat Ann Ratanamahatana (2004). “Towards parameter-free data mining”. In: *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’04*. ACM, pp. 206–215. DOI: 10.1145/1014052.1014077.

Cilibrasi, Rudi and Paul Vitányi (2005). “Clustering by compression”. In: *IEEE Transactions on Information Theory* 51.4, pp. 1523–1545. DOI: 10.1109/TIT.2005.844059.

Chandola, Varun and Vipin Kumar (2007). “Summarization – compressing data into an informative representation”. In: *Knowledge and Information Systems* 12.3, pp. 355–378. DOI: 10.1007/s10115-006-0039-1.

Faloutsos, Christos and Vasileios Megalooikonomou (2007). “On data mining, compression, and Kolmogorov complexity”. In: *Data Mining and Knowledge Discovery* 15.1, pp. 3–20. DOI: 10.1007/s10618-006-0057-3.

Keogh, Eamonn, Stefano Lonardi, Chotirat Ann Ratanamahatana, et al. (2007). “Compression-based data mining of sequential data”. In: *Data Mining and Knowledge Discovery* 14.1, pp. 99–129. DOI: 10.1007/s10618-006-0049-3.

Simovici, Dan A. (2013). “Minability through Compression”. In: *Proceedings of the 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC’13*, pp. 32–36. DOI: 10.1109/SYNASC.2013.11.

Simovici, Dan A. et al. (2015). “Compression and data mining”. In: *Proceedings of the 2015 International Conference on Computing, Networking and Communications, ICNC’15*, pp. 551–555. DOI: 10.1109/ICCNC.2015.7069404.

3.5 Some early uses of MDL in Machine Learning (ML)

The MDL principle has been applied early on in machine learning. Examples include evaluating hypotheses in inductive logic programming (Muggleton, Srinivasan, and Bain, 1992), learning Bayesian networks (Suzuki, 1993), decision trees (Quinlan and Rivest, 1989; Wallace and Patrick, 1993; Mehta, Rissanen, and Agrawal, 1995; Robnik-Šikonja and Kononenko, 1998), rules (Pfahringer, 1995a), or other related models (Quinlan, 1994; Kilpeläinen, Mannila, and E. Ukkonen, 1995), as well as feature engineering (Derthick, 1990; Derthick, 1991; Pfahringer, 1995b), supervised discretisation (Fayyad and Irani, 1993), signal smoothing (Pednault, 1989) and segmentation (Merhav, 1993; Shamir, Costello, and Merhav, 1999).

Pednault, Edwin P. D. (1989). “Some experiments in applying inductive inference principles to surface reconstruction”. In: *Proceedings of the 11th international joint conference on Artificial intelligence, IJCAI’89*. Morgan Kaufmann, pp. 1603–1609.

Quinlan, J. Ross and Ronald L. Rivest (1989). “Inferring decision trees using the minimum description length principle”. In: *Information and Computation* 80.3, pp. 227–248. DOI: 10.1016/0890-5401(89)90010-2.

Derthick, Mark (1990). *The minimum description length principle applied to feature learning and analogical mapping.* Technical report. MCC.

— (1991). "A minimal encoding approach to feature discovery". In: *Proceedings of the Ninth National conference on Artificial intelligence, AAAI'91.* Association for the Advancement of Artificial Intelligence, pp. 565–571.

Muggleton, Stephen, Ashwin Srinivasan, and Michael Bain (1992). "Compression, Significance and Accuracy". In: *Proceedings of the Ninth International Conference on Machine Learning, ICML'92.* Morgan Kaufmann, pp. 338–347. DOI: 10.1016/B978-1-55860-247-2.50048-6.

Fayyad, Usama M. and Keki B. Irani (1993). "Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning". In: *Machine Learning*.

Merhav, N. (1993). "On the minimum description length principle for sources with piecewise constant parameters". In: *IEEE Transactions on Information Theory* 39.6, pp. 1962–1967. DOI: 10.1109/18.265504.

Suzuki, Joe (1993). "A Construction of Bayesian Networks from Databases Based on an MDL Principle". In: *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence, UAI'93*, pp. 266–273. DOI: 10.1016/B978-1-4832-1451-1.50037-8.

Wallace, Christopher S. and Jon D. Patrick (1993). "Coding Decision Trees". In: *Machine Language* 11.1, pp. 7–22. DOI: 10.1023/A:1022646101185.

Quinlan, J. Ross (1994). "The Minimum Description Length Principle and Categorical Theories". In: *Proceedings of the Eleventh International Conference on Machine Learning, ICML'94.* Morgan Kaufmann, pp. 233–241. DOI: 10.1016/B978-1-55860-335-6.50036-2.

Kilpeläinen, Pekka, Heikki Mannila, and Esko Ukkonen (1995). "MDL learning of unions of simple pattern languages from positive examples". In: *Proceedings of the 2nd European Conference on Computational Learning Theory, EuroCOLT'95.* Springer, pp. 252–260. DOI: 10.1007/3-540-59119-2_182.

Mehta, Manish, Jorma Rissanen, and Rakesh Agrawal (1995). "MDL-based decision tree pruning". In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining, KDD'95.* Association for the Advancement of Artificial Intelligence, pp. 216–221.

Pfahringer, Bernhard (1995a). "A new MDL measure for robust rule induction". In: *Proceedings of the 8th European Conference on Machine Learning, ECML'95.* Springer, pp. 331–334. DOI: 10.1007/3-540-59286-5_80.

— (1995b). "Compression based feature subset selection". In: *Proceedings of the Workshop on Data Engineering for Inductive Learning @IJCAI'95*.

Robnik-Šikonja, Marko and Igor Kononenko (1998). "Pruning regression trees with MDL". In: *Proceedings of the 13th European conference on artificial intelligence, ECAI'98*.

Shamir, Gill I., Daniel J. Costello, and N. Merhav (1999). "Asymptotically optimal low complexity sequential lossless coding for regular piecewise stationary memoryless sources". In: *Proceedings of the 1999 IEEE Information Theory and Communications Workshop.* IEEE Computer Society, pp. 72–74. DOI: 10.1109/ITCOM.1999.781413.

3.6 Some uses of MDL in Natural Language Processing (NLP)

The MDL principle is also used in Natural Language Processing, in text and speech processing, in particular for clustering tasks (H. Li and Abe, 1997; H. Li and Abe, 1998b), for segmentation at the level of phrases (Kit, 1998), words (de Marcken, 1995; Kit and Wilks, 1999; Argamon et al., 2004) or morphemes (Brent, Murthy, and Lundberg, 1995; Creutz and Lagus, 2002), and to extract other types of text patterns (H. Li and Abe, 1998a; K. Wu et al., 2010). Some of the works are closely related to the development of methods for mining strings, and for analysing sequential data more generally (cf. Section 7).

Brent, Michael R., Sreerama K. Murthy, and Andrew Lundberg (1995). "Discovering morphemic suffixes: A case study in MDL induction". In: *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, AISTATS'95.* Springer, pp. 3–12. DOI: 10.1007/978-1-4612-2404-4_1.

de Marcken, Carl (1995). *The Unsupervised Acquisition of a Lexicon from Continuous Speech.* arXiv: cmp-lg/9512002.

Li, Hang and Naoki Abe (1997). "Clustering Words with the MDL Principle". In: *Journal of Natural Language Processing* 4.2, pp. 71–88. DOI: 10.5715/jnlp.4.2_71.

Kit, Chunyu (1998). "A Goodness Measure for Phrase Learning via Compression with the MDL Principle". In: *Proceedings of the 1998 European Summer School in Logic, Language and Information, ESSLLI'98, student session*, pp. 175–187.

Li, Hang and Naoki Abe (1998a). "Generalizing case frames using a thesaurus and the MDL principle". In: *Computational Linguistics* 24.2, pp. 217–244.

— (1998b). "Word clustering and disambiguation based on co-occurrence data". In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, ACL'98/COLING'98.* Association for Computational Linguistics, pp. 749–755. DOI: 10.3115/980691.980693.

Kit, Chunyu and Yorick Wilks (1999). “Unsupervised Learning of Word Boundary with Description Length Gain”. In: *Proceedings of the 1999 Workshop on Computational Natural Language Learning, CoNLL’99, Held in cooperation with EACL’99*.

Creutz, Mathias and Krista Lagus (2002). “Unsupervised discovery of morphemes”. In: *Proceedings of the ACL 2002 workshop on Morphological and phonological learning, MPL’02*. Association for Computational Linguistics, pp. 21–30. DOI: 10.3115/1118647.1118650.

Argamon, Shlomo et al. (2004). “Efficient Unsupervised Recursive Word Segmentation Using Minimum Description Length”. In: *Proceedings of the 20th International Conference on Computational Linguistics, COLING’04*. Association for Computational Linguistics, pp. 1058–1064.

Wu, Ke et al. (2010). “Unsupervised text pattern learning using minimum description length”. In: *Proceedings of the 4th International Universal Communication Symposium, IUCS’10*, pp. 161–166. DOI: 10.1109/IUCS.2010.5666227.

3.7 Mining and selecting itemsets

Mining frequent patterns is a core task in data mining, and itemsets are probably the most elementary and best studied type of pattern. Soon after the introduction of the frequent itemset mining task (Agrawal, Imieliński, and Swami, 1993), it became obvious that beyond issues of efficiency (Agrawal and Srikant, 1994; Mannila, Toivonen, and Verkamo, 1994), the problem of selecting patterns constituted a major challenge to tackle, lest the analysts drown under the deluge of extracted patterns.

Various properties and measures have been introduced to select itemsets (Webb and Vreeken, 2013; Geng and Hamilton, 2006). They include identifying representative itemsets (Bastide et al., 2000), using user-defined constraints to filter itemsets (De Raedt and Zimmermann, 2007; Soulet et al., 2011; Guns, Nijssen, and De Raedt, 2013), considering dependencies between itemsets (Jaroszewicz and Simovici, 2004; X. Yan et al., 2005; Tatti and Heikinheimo, 2008; Mampaey, 2010) and trying to evaluate the statistical significance of itemsets (Webb, 2007; Gionis et al., 2007; Tatti, 2010; Hämäläinen and Webb, 2018), also looking into alternative approaches to explore the search space (Boley, Lucchese, et al., 2011; Guns, Nijssen, and De Raedt, 2011). Initially focused on individual itemsets, approaches were later introduced to evaluate itemsets collectively, trying for instance to identify and remove redundancy (Gallo, De Bie, and Cristianini, 2007), also in an iterative manner (Hanhijärvi et al., 2009; Boley, Mampaey, et al., 2013). The goal hence moved from mining collections of good patterns to mining good collections of patterns, which is also the main objective when applying the MDL principle.

Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami (1993). “Mining association rules between sets of items in large databases”. In: *ACM SIGMOD Record* 22.2, pp. 207–216. DOI: 10.1145/170036.170072.

Agrawal, Rakesh and Ramakrishnan Srikant (1994). “Fast Algorithms for Mining Association Rules”. In: *Proceedings of 20th International Conference on Very Large Data Bases, VLDB’94*. Morgan Kaufmann, pp. 487–499.

Mannila, Heikki, Hannu Toivonen, and A Inkeri Verkamo (1994). “Efficient Algorithms for Discovering Association Rules”. In: *Proceedings of the KDD Workshop*. Association for the Advancement of Artificial Intelligence, pp. 181–192.

Silverstein, Craig, Sergey Brin, and Rajeev Motwani (1998). “Beyond Market Baskets: Generalizing Association Rules to Dependence Rules”. In: *Data Mining and Knowledge Discovery* 2.1, pp. 39–68. DOI: 10.1023/A:1009713703947.

Bastide, Yves et al. (2000). “Mining Minimal Non-redundant Association Rules Using Frequent Closed Itemsets”. In: *Proceedings of the First International Conference on Computational Logic, CL’00*. Springer, pp. 972–986.

Pavlov, Dmitry, Heikki Mannila, and Padhraic Smyth (2003). “Beyond independence: Probabilistic models for query approximation on binary transaction data”. In: *IEEE Transactions on Knowledge and Data Engineering* 15.6, pp. 1409–1421.

Jaroszewicz, Szymon and Dan A. Simovici (2004). “Interestingness of frequent itemsets using Bayesian networks as background knowledge”. In: *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’04*. ACM, pp. 178–186. DOI: 10.1145/1014052.1014074.

Yan, Xifeng et al. (2005). “Summarizing itemset patterns: a profile-based approach”. In: *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’05*. ACM, pp. 314–323. DOI: 10.1145/1081870.1081907.

Geng, Liqiang and Howard J. Hamilton (2006). “Interestingness measures for data mining: A survey”. In: *ACM Computing Surveys* 38.3. DOI: 10.1145/1132960.1132963.

Wang, Chao and Srinivasan Parthasarathy (2006). “Summarizing Itemset Patterns Using Probabilistic Models”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’19*. ACM, pp. 730–735. DOI: 10.1145/1150402.1150495.

De Raedt, Luc and Albrecht Zimmermann (2007). “Constraint-Based Pattern Set Mining”. In: *Proceedings of the 2007 SIAM International Conference on Data Mining, SDM'07*. SIAM, pp. 237–248. DOI: 10.1137/1.9781611972771.22.

Gallo, Arianna, Tijl De Bie, and Nello Cristianini (2007). “MINI: Mining Informative Non-redundant Itemsets”. In: *Proceedings of the European Conference on Knowledge Discovery in Databases, PKDD'07*. Springer, pp. 438–445. DOI: 10.1007/978-3-540-74976-9_44.

Gionis, Aristides et al. (2007). “Assessing data mining results via swap randomization”. In: *ACM Transactions on Knowledge Discovery from Data* 1.3, 14–es. DOI: 10.1145/1297332.1297338.

Webb, Geoffrey I. (2007). “Discovering Significant Patterns”. In: *Machine Learning* 68.1, pp. 1–33. DOI: 10.1007/s10994-007-5006-x.

Tatti, Nikolaj and Hannes Heikinheimo (2008). “Decomposable Families of Itemsets”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'08*, pp. 472–487. DOI: 10.1007/978-3-540-87481-2_31.

Hanhijärvi, Sami et al. (2009). “Tell Me Something I Don't Know: Randomization Strategies for Iterative Data Mining”. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'09*. ACM, pp. 379–388. DOI: 10.1145/1557019.1557065.

Mampaey, Michael (2010). “Mining Non-redundant Information-Theoretic Dependencies between Itemsets”. In: *Proceedings of the 12th International Conference on Data Warehousing and Knowledge Discovery, DaWaK'10*. Springer, pp. 130–141. DOI: 10.1007/978-3-642-15105-7_11.

Tatti, Nikolaj (2010). “Probably the best itemsets”. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'10*. ACM, pp. 293–302. DOI: 10.1145/1835804.1835843.

Boley, Mario, Claudio Lucchese, et al. (2011). “Direct local pattern sampling by efficient two-step random procedures”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11*. ACM, pp. 582–590. DOI: 10.1145/2020408.2020500.

Guns, Tias, Siegfried Nijssen, and Luc De Raedt (2011). “Itemset mining: A constraint programming perspective”. In: *Artificial Intelligence* 175.12, pp. 1951–1983.

Soulet, Arnaud et al. (2011). “Mining Dominant Patterns in the Sky”. In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM'11*. IEEE Computer Society, pp. 655–664. DOI: 10.1109/ICDM.2011.100.

Boley, Mario, Michael Mampaey, et al. (2013). “One Click Mining: Interactive Local Pattern Discovery Through Implicit Preference and Performance Learning”. In: *Proceedings of the Workshop on Interactive Data Exploration and Analytics, IDEA @KDD'13*. ACM, pp. 27–35. DOI: 10.1145/2501511.2501517.

Guns, Tias, Siegfried Nijssen, and Luc De Raedt (2013). “k-Pattern Set Mining under Constraints”. In: *IEEE Transactions on Knowledge and Data Engineering* 25.2, pp. 402–418. DOI: 10.1109/TKDE.2011.204.

Webb, Geoffrey I. and Jilles Vreeken (2013). “Efficient Discovery of the Most Interesting Associations”. In: *ACM Transactions on Knowledge Discovery from Data* 8.3, 15:1–15:31. DOI: 10.1145/2601433.

Hämäläinen, Wilhelmiina and Geoffrey I. Webb (2018). “A tutorial on statistically sound pattern discovery”. In: *Data Mining and Knowledge Discovery*. DOI: 10.1007/s10618-018-0590-x.

4 Mining itemsets with Krimp & Co.

A transactional database consists of a collection of sets, called transactions, over a universe of items. The prototypical example for this type of data comes from market-basket analysis, which is also where some of the terminology is borrowed from. Alternatively, a transactional database can be represented as a binary matrix. Frequent itemset mining, that is, finding items that frequently co-occur in a transactional database, is a central task in data mining (cf. Section 3.7).

4.1 Krimp

The introduction by Siebes, Vreeken, and van Leeuwen in 2006 of a MDL-based algorithm for mining and selecting small but high-quality collections of itemsets sparked a productive line of research, including algorithmic improvements, adaptations to different tasks, and various applications of the original algorithm. The algorithm, soon dubbed KRIMP (van Leeuwen, Vreeken, and Siebes, 2006), Dutch for “to shrink”, is a prime example of a *dictionary-based* strategy (cf. Section 2.3), illustrated in Figure 3.

Through an evaluation on a classification task, van Leeuwen, Vreeken, and Siebes (2006) show that the selected itemsets are representative. Specifically, considering a labelled training dataset, KRIMP is applied separately on the transactions associated to each class to mine a code table. A given test transaction can then be encoded using each of the code tables, and assigned to the class that corresponds to the shortest code length.

Siebes, Arno, Jilles Vreeken, and Matthijs van Leeuwen (2006). “Item Sets that Compress”. In: *Proceedings of the 2006 SIAM International Conference on Data Mining, SDM’06*. SIAM.

van Leeuwen, Matthijs, Jilles Vreeken, and Arno Siebes (2006). “Compression Picks Item Sets That Matter”. In: *Proceedings of the European Conference on Knowledge Discovery in Databases, PKDD’06*. Springer, pp. 585–592. DOI: 10.1007/11871637_59.

— (2009). “Identifying the components”. In: *Data Mining and Knowledge Discovery* 19.2, pp. 176–193. DOI: 10.1007/s10618-009-0137-2.

Vreeken, Jilles (2009). “Making Pattern Mining Useful”. PhD thesis. Universiteit Utrecht.

van Leeuwen, Matthijs (2010). “Patterns that Matter”. PhD thesis. Universiteit Utrecht.

Vreeken, Jilles, Matthijs van Leeuwen, and Arno Siebes (2011). “Krimp: Mining Itemsets that Compress”. In: *Data Mining and Knowledge Discovery* 23.1, pp. 169–214.

Siebes, Arno (2012). “Queries for Data Analysis”. In: *Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis, IDA’12*. Springer, pp. 7–22.

— (2014). “MDL in Pattern Mining: A Brief Introduction to Krimp”. In: *Proceedings of the international conference on Formal Concept Analysis, FCA’14*. Springer, pp. 37–43. DOI: 10.1007/978-3-319-07248-7_3.

van Leeuwen, Matthijs and Jilles Vreeken (2014). “Mining and Using Sets of Patterns through Compression”. In: *Frequent Pattern Mining*. Springer, pp. 165–198. DOI: 10.1007/978-3-319-07821-2_8.

4.2 Algorithmic improvements

Works building on KRIMP include several algorithmic improvements. In particular, the SLIM algorithm of Smets and Vreeken (2012) modifies KRIMP and greedily generates candidates by merging patterns, instead of evaluating candidates from a pre-mined list. The example presented in Figure 3*(ii)* (cf. Section 2.3) was actually obtained by running the SLIM algorithm on the toy dataset of Figure 2. Hess, Piatkowski, and Morik (2014) propose a data structure similar to the FP-tree to facilitate the recomputation of usages when the KRIMP code table is updated, making the mining algorithm more efficient. Sampson and Berthold (2014) apply widening, i.e. diversification of the search, to KRIMP.

Smets, Koen and Jilles Vreeken (2012). “Slim: Directly Mining Descriptive Patterns”. In: *Proceedings of the 2012 SIAM International Conference on Data Mining, SDM’12*. SIAM, pp. 236–247.

Hess, Sibylle, Nico Piatkowski, and Katharina Morik (2014). “SHrimp: Descriptive Patterns in a Tree”. In: *Proceedings of the LWA (Lernen, Wissen, Adaption) 2014 Workshops: KDML, IR, FGWM*.

Sampson, Oliver and Michael R. Berthold (2014). “Widened KRIMP: Better Performance through Diverse Parallelism”. In: *Proceedings of the 13th International Symposium on Advances in Intelligent Data Analysis, IDA’14*. Springer, pp. 276–285. DOI: 10.1007/978-3-319-12571-8_24.

4.3 Finding differences and anomalies

The analysis of differences between databases and the detection of anomalies are derivative tasks that have attracted particular attention. Vreeken, van Leeuwen, and Siebes (2007a) use KRIMP to measure differences between two databases, by comparing the length of the description of a database obtained with a code table induced on that database versus one induced on the other database. The coverage of individual transactions by the selected itemsets, and the specific code tables obtained are also compared. The DIFFNORM algorithm introduced by Budhathoki and Vreeken (2015) aims to encode multiple databases at once without redundancy, and allows to investigate the differences and similarities between the databases by inspecting the obtained code table. As a major contribution, Budhathoki and Vreeken (2015) improve the encoding by replacing the Shannon–Fano code used in the original KRIMP by a prequential plug-in code (cf. Section 2.1).

Smets and Vreeken (2011) use KRIMP for outlier detection by looking at how many bits are needed to encode a transaction. If this number is much larger than expected, the transaction is declared anomalous. In addition, the encodings of transactions can be scrutinised to obtain further insight into how they depart from the rest. Akoglu, Tong, Vreeken, et al. (2012) design an algorithm that detects as anomalies transactions having a high encoding cost. Their proposed algorithm mines multiple code tables, rather than a single one in KRIMP, and handles categorical data. Bertens, Vreeken, and Siebes (2015) (also Bertens, Vreeken, and Siebes, 2017) propose a method to detect anomalous co-occurrences based on the KRIMP/SLIM code tables.

Vreeken, Jilles, Matthijs van Leeuwen, and Arno Siebes (2007a). “Characterising the difference”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’07*. ACM, pp. 765–774. DOI: 10.1145/1281192.1281274.

Smets, Koen and Jilles Vreeken (2011). “The Odd One Out: Identifying and Characterising Anomalies”. In: *Proceedings of the 2011 SIAM International Conference on Data Mining, SDM’11.* SIAM, pp. 804–815. DOI: 10.1137/1.9781611972818.69.

Akoglu, Leman, Hanghang Tong, Jilles Vreeken, et al. (2012). “Fast and reliable anomaly detection in categorical data”. In: *Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM’12.* ACM, pp. 415–424. DOI: 10.1145/2396761.2396816.

Bertens, Roel, Jilles Vreeken, and Arno Siebes (2015). *Beauty and Brains: Detecting Anomalous Pattern Co-Occurrences.* arXiv: 1512.07048.

Budhathoki, Kailash and Jilles Vreeken (2015). “The Difference and the Norm – Characterising Similarities and Differences Between Databases”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’15.* Vol. 9285. Springer, pp. 206–223. DOI: 10.1007/978-3-319-23525-7_13.

Bertens, Roel, Jilles Vreeken, and Arno Siebes (2017). “Efficiently Discovering Unexpected Pattern-Co-Occurrences”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining, SDM’17.* SIAM, pp. 126–134. DOI: 10.1137/1.9781611974973.15.

4.4 Mining rule sets

Going beyond itemsets, a closely related task is to mine rules. van Leeuwen and Galbrun (2015) propose to mine association rules across a two-view transactional dataset, such that one view can be reconstructed from the other, and vice versa. Fischer and Vreeken (2019) instead consider a unique set of items and aim to mine associations rules that allow to reconstruct the dataset, enabling corrections in order to increase the robustness of the results. They then apply the approach to learn rules about activation patterns in neural networks (Fischer, Oláh, and Vreeken, 2021).

Aoga et al. (2018) present a method to encode a binary label associated to each transaction, using the original transactions and a list of rules, each associated to a probability that the target variable holds true. Proença and van Leeuwen (2020b) (also Proença and van Leeuwen, 2020a) consider a similar task, but with multiple classes and targeted towards predictive rather than descriptive rules, then looking for rules that capture deviating groups of transactions, i.e. dealing with the subgroup discovery task (Proença, Grünwald, et al., 2020; Proença, Grünwald, et al., 2021). Beside binary attributes, i.e. items, Proença, Grünwald, et al. (2020) also consider nominal and numerical attributes, and aim to predict a single numerical target, modeled using normal distributions, instead of a binary target. Proença, Bäck, and van Leeuwen (2021) further extend the approach to more complex nominal and numerical targets by resorting to normalised maximum likelihood (NML) and Bayesian codes, respectively.

Whereas the former two output a set of rules, these latter methods return a list of rules, such that at most one rule, the first valid rule encountered in the list, applies to any given transaction. In all cases, the dataset, or part of it, is assumed to be given and the aim is to reconstruct a target variable, or the rest of the dataset.

van Leeuwen, Matthijs and Esther Galbrun (2015). “Association Discovery in Two-View Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.12, pp. 3190–3202. DOI: 10.1109/TKDE.2015.2453159.

Aoga, John O. R. et al. (2018). “Finding Probabilistic Rule Lists using the Minimum Description Length Principle”. In: *Proceedings of the International Conference on Discovery Science, DS’18.* Springer, pp. 66–82. DOI: 10.1007/978-3-030-01771-2_5.

Fischer, Jonas and Jilles Vreeken (2019). “Sets of Robust Rules, and How to Find Them”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’19.* ACM, pp. 38–54. DOI: 10.1007/978-3-030-46150-8_3.

Proença, Hugo M., Peter D. Grünwald, et al. (2020). “Discovering outstanding subgroup lists for numeric targets using MDL”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’20.*

Proença, Hugo M. and Matthijs van Leeuwen (2020a). “Interpretable multiclass classification by MDL-based rule lists”. In: *Information Sciences* 512, pp. 1372–1393. DOI: 10.1016/j.ins.2019.10.050.

— (2020b). *Interpretable multiclass classification by MDL-based rule lists.* arXiv: 1905.00328.

Fischer, Jonas, Anna Oláh, and Jilles Vreeken (2021). “What’s in the Box? Explaining Neural Networks with Robust Rules”. In: *Proceedings of the 38th International Conference on Machine Learning, ICML’21.*

Proença, Hugo M., Thomas Bäck, and Matthijs van Leeuwen (2021). *Robust subgroup discovery.* arXiv: 2103.13686.

Proença, Hugo M., Peter D. Grünwald, et al. (2021). *Discovering outstanding subgroup lists for numeric targets using MDL.* arXiv: 2006.09186.

4.5 Other adaptations

Further work also includes extending the Krimp approach to more expressive patterns, such as patterns mined from relational databases (Koopman and Siebes, 2008; Koopman and Siebes, 2009), and using Krimp for derivative tasks.

van Leeuwen and Siebes (2008) use Krimp to detect changes in the distribution of items in a streaming setting. If a code table induced from an earlier part of the stream no longer provides good compression as compared to a code table induced from a more recent part of the stream, it is a signal that the distribution has changed. Bonchi, van Leeuwen, and A. Ukkonen (2011) extend the approach to the probabilistic setting, where the occurrence of an item in a transaction is associated to a probability, aiming to find a collection of itemsets that compress the data well in expectation.

Given an original database, Vreeken, van Leeuwen, and Siebes (2007b) use Krimp to generate a synthetic database similar to the original one, for use in the context of privacy-preserving data mining. The code table is induced on the original database, itemsets are then sampled from it and combined to generate synthetic transactions. Vreeken and Siebes (2008) use Krimp for data completion. In a way, this turns the MDL principle on its head. Starting from an incomplete database, rather than looking for the patterns that compress the data best, the proposed approach looks for the data that is best compressed by the patterns and considers that to be the best completion. Instead of a single code table, Siebes and Kersten (2011) look for a collection of code tables, that capture the structure of the dataset at different levels of granularity. Representing a categorical dataset as transactional data by mapping each value of an attribute to a distinct item implies that each transaction contains exactly one item for each categorical attribute, and hence that all transactions have the same length. Siebes and Kersten (2012) consider the problem of smoothing out the small scale structure from such datasets. That is, they aim to replace entries in the data so that its large scale structure is maintained but it can be compressed better.

Vreeken, Jilles, Matthijs van Leeuwen, and Arno Siebes (2007b). "Preserving Privacy through Data Generation". In: *Proceedings of the 7th IEEE International Conference on Data Mining, ICDM'07.* IEEE Computer Society, pp. 685–690. doi: 10.1109/ICDM.2007.25.

Koopman, Arne and Arno Siebes (2008). "Discovering Relational Item Sets Efficiently". In: *Proceedings of the 2008 SIAM International Conference on Data Mining, SDM'08.* SIAM, pp. 108–119. doi: 10.1137/1.9781611972788.10.

van Leeuwen, Matthijs and Arno Siebes (2008). "StreamKrimp: Detecting Change in Data Streams". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'08.* Springer, pp. 672–687. doi: 10.1007/978-3-540-87479-9_62.

Vreeken, Jilles and Arno Siebes (2008). "Filling in the Blanks – Krimp Minimisation for Missing Data". In: *Proceedings of the 8th IEEE International Conference on Data Mining, ICDM'08.* IEEE Computer Society, pp. 1067–1072. doi: 10.1109/ICDM.2008.40.

Koopman, Arne and Arno Siebes (2009). "Characteristic Relational Patterns". In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'09.* ACM, pp. 437–446. doi: 10.1145/1557019.1557071.

Bonchi, Francesco, Matthijs van Leeuwen, and Antti Ukkonen (2011). "Characterizing Uncertain Data using Compression". In: *Proceedings of the 2011 SIAM International Conference on Data Mining, SDM'11.* SIAM, pp. 534–545.

Siebes, Arno and René Kersten (2011). "A Structure Function for Transaction Data". In: *Proceedings of the 2011 SIAM International Conference on Data Mining, SDM'11.* SIAM, pp. 558–569. doi: 10.1137/1.9781611972818.48.

— (2012). "Smoothing Categorical Data". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'12.* Springer, pp. 42–57. doi: 10.1007/978-3-642-33460-3_8.

4.6 Applications

The Krimp algorithm has also been employed to tackle problems in different domains, including clustering tagged media (van Leeuwen, Bonchi, et al., 2009), summarising text (Vanetik and Litvak, 2017; Vanetik and Litvak, 2018), detecting malware (Asadi and Varadharajan, 2019a; Asadi and Varadharajan, 2019b) and analysing the Semantic Web (Bobed et al., 2019).

van Leeuwen, Matthijs, Francesco Bonchi, et al. (2009). "Compressing tags to find interesting media groups". In: *Proceedings of the 18th ACM conference on Information and knowledge management, CIKM'09.* ACM, pp. 1147–1156. doi: 10.1145/1645953.1646099.

Vanetik, Natalia and Marina Litvak (2017). “Query-based summarization using MDL principle”. In: *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres @ACL'17*, pp. 22–31.

— (2018). “DRIM: MDL-Based Approach for Fast Diverse Summarization”. In: *Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI'18*, pp. 660–663. DOI: `10.1109/WI.2018.00-17`.

Asadi, Behzad and Vijay Varadharajan (2019a). *An MDL-Based Classifier for Transactional Datasets with Application in Malware Detection.* arXiv: `1910.03751`.

— (2019b). *Towards a Robust Classifier: An MDL-Based Method for Generating Adversarial Examples.* arXiv: `1912.05945`.

Bobed, Carlos et al. (2019). “Data-driven Assessment of Structural Evolution of RDF Graphs”. In: *Semantic Web – Interoperability, Usability, Applicability.*

5 Tabular data (continued)

Transactional data can be seen as a binary matrix or table, and is hence a form of tabular data. Data tables might also contain categorical or real-valued attributes, which can be either turned into binary attributes as a pre-processing or handled directly with dedicated methods.

5.1 More itemsets

Beside Krimp and algorithms inspired from it, different approaches have been proposed to mine itemsets from binary matrices using the MDL principle.

Heikinheimo et al. (2009) focus on finding collections of low-entropy itemsets, typically even more compact than those obtained with Krimp. On the other hand, the main objective of Mampaey and Vreeken (2010) is to provide a summary of the dataset in the form of a partitioning of the items. For each partition, codewords are associated to the different combinations of the items that comprise the partition, and used to encode the corresponding subset of the data. In the Pack algorithm, proposed by Tatti and Vreeken (2008), the code representing an item is made dependent on the presence/absence of other items in the same transaction. This can be represented as decision trees whose intermediate nodes are items and leaves contain code tables for other items. Fischer and Vreeken (2020) introduce a rich pattern language that can capture both co-occurrences and mutual exclusivity of items.

Mampaey, Vreeken, and Tatti (2012) present a variant of their MTV itemset mining algorithm (cf. Section 9.2) where the MDL principle is used to choose the collection of itemsets that yields the best model, as an alternative to the Bayesian Information Criterion (BIC).

Tatti, Nikolaj and Jilles Vreeken (2008). “Finding Good Itemsets by Packing Data”. In: *Proceedings of the 8th IEEE International Conference on Data Mining, ICDM'08.* IEEE Computer Society, pp. 588–597. DOI: `10.1109/ICDM.2008.39`.

Heikinheimo, Hannes et al. (2009). “Low-Entropy Set Selection”. In: *Proceedings of the 2009 SIAM International Conference on Data Mining, SDM'09.* SIAM, pp. 569–580. DOI: `10.1137/1.9781611972795.49`.

Mampaey, Michael and Jilles Vreeken (2010). “Summarising Data by Clustering Items”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'10*, pp. 321–336. DOI: `10.1007/978-3-642-15883-4_21`.

Mampaey, Michael, Jilles Vreeken, and Nikolaj Tatti (2012). “Summarizing Data Succinctly with the Most Informative Itemsets”. In: *ACM Transactions on Knowledge Discovery from Data* 6.4, 16:1–16:42. DOI: `10.1145/2382577.2382580`.

Fischer, Jonas and Jilles Vreeken (2020). “Discovering Succinct Pattern Sets Expressing Co-Occurrence and Mutual Exclusivity”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'19.* ACM.

5.2 Blocks in binary data

A different type of patterns that can be mined from binary matrices consists of blocks defined by a collection of rows and columns with a homogeneous density of ones, sometimes known as *tiles* or as *biclusters*, constituting the basis of *block-based* strategies (cf. Section 2.3). Chakrabarti et al. (2004) propose a method to partition the rows and columns of the matrix into subsets that define homogeneous blocks. The example presented in Figure 4 follows this approach. Tatti and Vreeken (2012a) introduce the Stijl algorithm to mine a hierarchy of tiles from a binary matrix. The adjectives *geometric* or *spatial* typically refer to cases where the order of the rows and columns of the data matrix is meaningful (e.g. Papadimitriou, Gionis, et al., 2005; Faas and van Leeuwen, 2020).

Chakrabarti, Deepayan et al. (2004). "Fully Automatic Cross-associations". In: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'04*. ACM, pp. 79–88. DOI: 10.1145/1014052.1014064.

Papadimitriou, Spiros, Aristides Gionis, et al. (2005). "Parameter-Free Spatial Data Mining Using MDL". In: *Proceedings of the 5th IEEE International Conference on Data Mining, ICDM'05*. IEEE Computer Society, pp. 346–353. DOI: 10.1109/ICDM.2005.117.

Tatti, Nikolaj and Jilles Vreeken (2012a). "Discovering Descriptive Tile Trees". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'12*. Springer, pp. 9–24. DOI: 10.1007/978-3-642-33460-3_6.

Faas, Micky and Matthijs van Leeuwen (2020). "Vouw: Geometric Pattern Mining Using the MDL Principle". In: *Proceedings of the 18th International Symposium on Advances in Intelligent Data Analysis, IDA'20*. Springer, pp. 158–170. DOI: 10.1007/978-3-030-44584-3_13.

5.3 Formal Concept Analysis (FCA)

The field of Formal Concept Analysis focuses on basically the same problem as itemset mining, but from a slightly different perspective and with its own formalism. Some steps have been taken towards applying and further developing MDL-based methods within this framework (Otaki and Yamamoto, 2015; Yurov and Ignatov, 2017). In particular, as a topic in her doctoral dissertation, Makhalova (2021) extensively studied MDL-based itemset mining methods from the perspective of FCA (also Makhalova, Kuznetsov, and Napoli, 2018a; Makhalova, Kuznetsov, and Napoli, 2018b; Makhalova, Kuznetsov, and Napoli, 2019b; Makhalova, Kuznetsov, and Napoli, 2021).

Otaki, Keisuke and Akihiro Yamamoto (2015). "Edit Operations on Lattices for MDL-based Pattern Summarization". In: *Proceedings of the International Workshop on Formal Concept Analysis and Applications @ICFCA'15*.

Yurov, Maxim and Dmitry I. Ignatov (2017). "Turning Krimp into a Triclustering Technique on Sets of Attribute-Condition Pairs that Compress". In: *Proceedings of the International Joint Conference on Rough Sets, IJCRS'17*. Springer, pp. 558–569. DOI: 10.1007/978-3-319-60840-2_40.

Makhalova, Tatiana, Sergei O. Kuznetsov, and Amedeo Napoli (2018a). "A First Study on What MDL Can Do for FCA". In: *Proceedings of the Fifteen International Conference on Concept Lattices and Their Applications, CLA'18*, pp. 25–36.

— (2018b). "MDL for FCA: Is There a Place for Background Knowledge?" In: *Proceedings of the 6th International Workshop "What can FCA do for Artificial Intelligence?" @ IJCAI/ECAI'18*. Vol. 2149. CEUR Workshop Proceedings, pp. 45–56. URL: http://ceur-ws.org/Vol-2149/paper5.pdf.

— (2019b). "On Coupling FCA and MDL in Pattern Mining". In: *Proceedings of the international conference on Formal Concept Analysis, FCA'19*. Springer, pp. 332–340. DOI: 10.1007/978-3-030-21462-3_23.

Makhalova, Tatiana (2021). "Contributions to pattern set mining : from complex datasets to significant and useful pattern sets". PhD thesis. Université de Lorraine. URL: https://hal.univ-lorraine.fr/tel-03342124.

Makhalova, Tatiana, Sergei O. Kuznetsov, and Amedeo Napoli (2021). "Likely-Occurring Itemsets for Pattern Mining". In: *Proceedings of the 6th International Workshop "What can FCA do for Artificial Intelligence?" @ IJCAI'21*. Vol. 2972. CEUR Workshop Proceedings, pp. 39–50. URL: http://ceur-ws.org/Vol-2972/paper4.pdf.

5.4 Boolean Matrix Factorisation (BMF)

Factorising a matrix consists in identifying two matrices, the factors, so that the original matrix can be reconstructed as their product. One challenge is to find the right balance between the complexity of the decomposition (generally measured by the size of the factors) and the accuracy of the reconstruction, which is where the MDL principle comes in handy (Miettinen and Vreeken, 2011; Miettinen and Vreeken, 2014; Hess, Morik, and Piatkowski, 2017). Whereas the other approaches permit reconstruction errors in both values, Makhalova and Trnecka (2019) do not allow factors to cover zero entries, considering so-called "from-below" factorisations (also Makhalova and Trnecka, 2021). Lucchese, Orlando, and Perego (2014) (also Lucchese, Orlando, and Perego, 2010b; Lucchese, Orlando, and Perego, 2010a) propose a MDL-based score to compare factors of a fixed size.

An example of a factorisation of the toy binary dataset from Figure 2 is provided in Figure 5. The model consists of a pair of factor matrices whereas the data is encoded as a mask of corrections. To reconstruct the

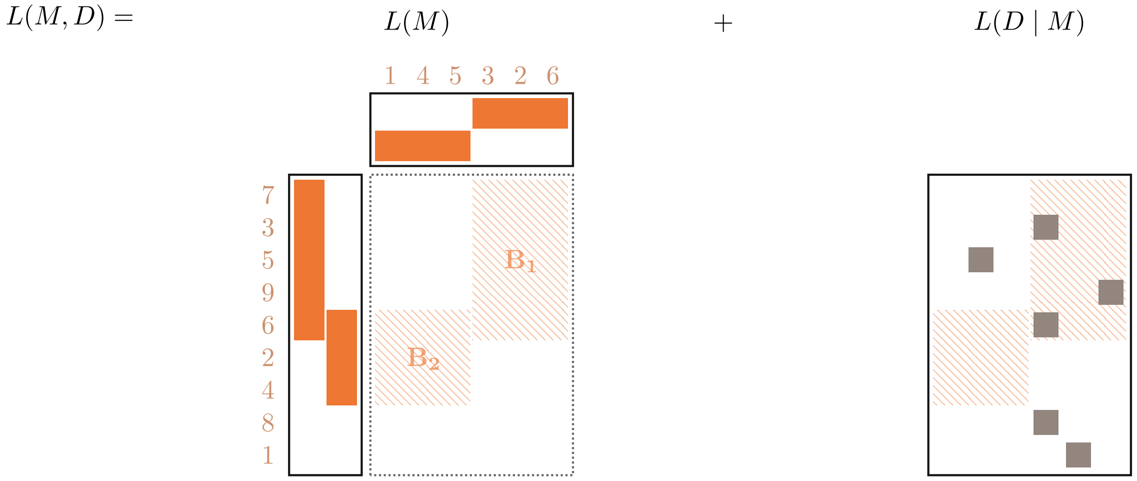


Figure 5: Boolean Matrix Factorisation, example on the toy binary dataset of Figure 2. The model M consists of a pair of factor matrices (left) and the data D is encoded as a mask of corrections (right). To reconstruct the original data, first the Boolean matrix product of the factors is computed, resulting in the matrix with two fully dense blocks B_1 and B_2. Then, the mask of corrections is applied to this matrix, that is, the value of the cells indicated in the mask (grey squares) are flipped. Note that the blocks drawn with the hatch pattern depict the intermediate step of the reconstruction and are not actually encoded as such.

original data, the Boolean matrix product of the factors is computed and the mask of corrections is applied to the resulting matrix.

When applied to a Boolean matrix, factorisation shares some similarities with itemset mining, as it aims to identify items that occur (or do not occur) frequently together. That is, the two factor matrices can be interpreted as specifying itemsets and indicators of occurrence, respectively. On the other hand, the factors can also be interpreted as specifying possibly overlapping fully dense blocks. A mask applied to the reconstructed data provides a global error correction mechanism. Thus, Boolean matrix factorisation can be seen as a hybrid approach.

Lucchese, Claudio, Salvatore Orlando, and Raffaele Perego (2010a). “A generative pattern model for mining binary datasets”. In: *Proceedings of the 2010 ACM Symposium on Applied Computing, SAC’10*. ACM, pp. 1109–1110. DOI: `10.1145/1774088.1774320`.

— (2010b). “Mining Top-K Patterns from Binary Datasets in presence of Noise”. In: *Proceedings of the 2007 SIAM International Conference on Data Mining, SDM’07*. SIAM, pp. 165–176. DOI: `10.1137/1.9781611972801.15`.

Miettinen, Pauli and Jilles Vreeken (2011). “Model order selection for boolean matrix factorization”. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’11*. ACM, pp. 51–59. DOI: `10.1145/2020408.2020424`.

Lucchese, Claudio, Salvatore Orlando, and Raffaele Perego (2014). “A Unifying Framework for Mining Approximate Top-k Binary Patterns”. In: *IEEE Transactions on Knowledge and Data Engineering* 26.12, pp. 2900–2913. DOI: `10.1109/TKDE.2013.181`.

Miettinen, Pauli and Jilles Vreeken (2014). “MDL4BMF: Minimum Description Length for Boolean Matrix Factorization”. In: *ACM Transactions on Knowledge Discovery from Data* 8.4, 18:1–18:31. DOI: `10.1145/2601437`.

Hess, Sibylle, Katharina Morik, and Nico Piatkowski (2017). “The PRIMPING routine – Tiling through proximal alternating linearized minimization”. In: *Data Mining and Knowledge Discovery* 31.4, pp. 1090–1131. DOI: `10.1007/s10618-017-0508-z`.

Makhalova, Tatiana and Martin Trnecka (2019). *From-Below Boolean Matrix Factorization Algorithm Based on MDL*. arXiv: `1901.09567`.

— (2021). “From-below Boolean matrix factorization algorithm based on MDL”. In: *Advances in Data Analysis and Classification* 15.1, pp. 37–56. DOI: `10.1007/s11634-019-00383-6`.

5.5 Categorical data

One way to mine datasets involving categorical attributes is to binarise them and then apply, for instance, an itemset mining algorithm. However, binarisation entails a loss of information and dedicated methods can hence offer a better alternative.

Mampaey and Vreeken (2013) introduce an approach to detect correlated attributes, i.e. such that the different categories occur in a coordinated manner, whereas X. He, Feng, Konte, et al. (2014) present a subspace clustering method, i.e. look not only for groups of attributes, but also corresponding groups of rows where coordinated behaviour occurs.

Mampaey, Michael and Jilles Vreeken (2013). “Summarizing categorical data by clustering attributes”. In: *Data Mining and Knowledge Discovery* 26.1, pp. 130–173. DOI: 10.1007/s10618-011-0246-6.

He, Xiao, Jing Feng, Bettina Konte, et al. (2014). “Relevant overlapping subspace clusters on categorical data”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’14*. ACM, pp. 213–222. DOI: 10.1145/2623330.2623652.

5.6 Numerical data

A numerical data table containing m columns can be seen as a collection of points in a m-dimensional space. In pattern mining, numerical data is often handled by applying discretisation, which requires to partition the data into coherent blocks. While it can be seen as a data exploration task in its own right, providing an overview of the dataset and the distribution of values, discretisation often constitutes a pre-processing task to allow applying algorithms that can handle only discrete input data. Yet, choosing good parameters for the discretisation can be difficult, and its quality can have a major impact on later processing. Unsupervised discretisation, where no side information is available, is in contrast to supervised discretisation, that takes into account class labels and often precedes a machine learning task. Here we focus on the former.

Kontkanen and Myllymäki (2007) propose a histogram density estimation method that relies on the MDL principle, formalised using the normalised maximum likelihood (NML) distribution. This method is employed by Kameya (2011) to discretise time-series seen as a collection of two-dimensional time–measurement data points, and extended by Yang, Baratchi, and van Leeuwen (2020) to two-dimensional numerical data, more in general. Along similar lines, Nguyen, Müller, et al. (2014) aim to automatically identify a high-quality discretisation that preserves the interactions between attributes. Witteveen et al. (2014) extend the Kraft inequality (cf. Section 2.1) to numerical data and introduce an approach to find hyperintervals, i.e. multidimensional blocks.

Makhalova, Kuznetsov, and Napoli (2019a) consider the problem of mining interesting hyperrectangles from discretised numerical data, and aim to design an encoding that accommodates overlaps between patterns (Makhalova, Kuznetsov, and Napoli, 2020; Makhalova, Kuznetsov, and Napoli, 2022).

Lakshmanan et al. (2002) formalise mining OLAP data, i.e. multidimensional datasets, as a problem of finding a cover in a multidimensional array containing positive, negative and neutral cells. The aim is then to find the most compact set of hyperrectangles that covers all positive cells, none of the negative cells, and no more than a chosen number of the neutral cells. The score is presented as a generalised MDL due to the tolerance on neutral cells. However, coverings are evaluated by simply counting cells, which does not actually adhere with the principle, generalised or otherwise.

Lakshmanan, Laks V. S. et al. (2002). “The generalized MDL approach for summarization”. In: *Proceedings of the 28th international conference on Very Large Data Bases, VLDB’02*. VLDB Endowment, pp. 766–777.

Kontkanen, Petri and Petri Myllymäki (2007). “MDL Histogram Density Estimation”. In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS’07*, pp. 219–226.

Kameya, Yoshitaka (2011). “Time Series Discretization via MDL-Based Histogram Density Estimation”. In: *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI’11*. IEEE Computer Society, pp. 732–739. DOI: 10.1109/ICTAI.2011.115.

Nguyen, Hoang-Vu, Emmanuel Müller, et al. (2014). “Unsupervised interaction-preserving discretization of multivariate data”. In: *Data Mining and Knowledge Discovery* 28.5, pp. 1366–1397. DOI: 10.1007/s10618-014-0350-5.

Witteveen, Jouke et al. (2014). “RealKrimp – Finding Hyperintervals that Compress with MDL for Real-Valued Data”. In: *Proceedings of the 13th International Symposium on Advances in Intelligent Data Analysis, IDA’14*. Springer, pp. 368–379. DOI: 10.1007/978-3-319-12571-8_32.

Makhalova, Tatiana, Sergei O. Kuznetsov, and Amedeo Napoli (2019a). “Numerical Pattern Mining Through Compression”. In: *Proceedings of the Data Compression Conference, DCC’19*, pp. 112–121. DOI: 10.1109/DCC.2019.00019.

— (2020). *Mint: MDL-based approach for Mining INTeresting Numerical Pattern Sets.* arXiv: 2011.14843.

Yang, Lincen, Mitra Baratchi, and Matthijs van Leeuwen (2020). *Unsupervised Discretization by Two-dimensional MDL-based Histogram.* arXiv: 2006.01893.

Makhalova, Tatiana, Sergei O. Kuznetsov, and Amedeo Napoli (2022). “Mint: MDL-based approach for Mining INTeresting Numerical Pattern Sets”. In: *Data Mining and Knowledge Discovery* 36.1, pp. 108–145. DOI: 10.1007/s10618-021-00799-9.

6 Graphs

In this section, we consider approaches for mining graphs. At their simplest, graphs are undirected and unlabelled, but they can also come with directed edges, with node or edge labels, or be dynamic, that is, time-evolving. The main tasks consist in identifying nodes that have similar connection patterns to group them into homogeneous blocks and in finding recurrent connection substructures. These correspond respectively to *block-based* and *dictionary-based* strategies (cf. Section 2.3).

For illustrative purposes, we consider a toy graph, shown in Figure 6, and delineate approaches that follow either strategy. The example shown in Figure 7 illustrates the *block-based* strategy and follows the work of Chakrabarti (2004) (cf. Section 6.1), whereas the example shown in Figure 8 illustrates the *dictionary-based* strategy and follows the work of Bariatti, Cellier, and Ferré (2020b) (cf. Section 6.5).

Looking at the corresponding adjacency matrix, a simple unlabelled graph can be represented as a binary table. Approaches from Sections 4 and 5 can thus readily be used for mining graphs. On one hand, the problem of grouping nodes into blocks that constitute particularly dense subgraphs, or communities, is closely related to identifying particularly dense tiles in a binary matrix. On the other hand, approaches that follow a *dictionary-based* strategy and aim to identify substructures in the graphs share similarities with their counterparts for binary tabular data. However, it is not enough to simply replace the subgraph patterns by their assigned codewords. The information about how the subgraphs are connected also needs to be encoded, requiring more complex encoding schemes.

The survey by Liu, Safavi, Dighe, et al. (2018) covers graph summarisation methods in general, whereas Feng (2015) presents various information-theoretic graph mining methods, both of which include MDL-based methods for analysing graphs as a subset.

Feng, Jing (2015). “Information-theoretic graph mining”. PhD thesis. Ludwig-Maximilians-Universität München.

Liu, Yike, Tara Safavi, Abhilash Dighe, et al. (2018). “Graph Summarization Methods and Applications: A Survey”. In: *ACM Computing Surveys* 51.3, 62:1–62:34. DOI: 10.1145/3186727.

6.1 Grouping nodes into blocks

Rosvall and Bergstrom (2007) present an information-theoretic framework to identify community structure in networks by grouping nodes and propose to use the MDL principle to automatically select the number of groups in which to arrange the nodes.

Chakrabarti (2004) proposes to compress the adjacency matrix of a graph by grouping nodes into homogeneous blocks (see Figure 7), with a top-down procedure to search for a good partition. Navlakha, Rastogi, and Shrivastava (2008) similarly propose to build graph summaries by grouping nodes into supernodes, but with a bottom-up search procedure. A superedge linking two supernodes represents edges between all pairs of elementary nodes from either supernodes (hence a supernode with a loop represents a clique). When reconstructing the original graph, after expanding the supernodes and superedges, some corrections must be performed, to add and remove spurious edges. Navlakha, Rastogi, and Shrivastava (2008) let the cost of encoding a graph equal the number of superedges and edge corrections, ignoring the cost of the assignment of nodes to supernodes.

Khan, Nawaz, and Y.-K. Lee (2015b) (also Khan, 2015; Khan, Nawaz, and Y.-K. Lee, 2015a) work with essentially the same encoding, using locality-sensitive hashing (LSH) to identify candidates for merger, additionally considering node labels (Khan, Nawaz, and Y.-K. Lee, 2014). Akoglu, Tong, Meeder, et al. (2012) also aim at grouping nodes while taking into account node attributes.

A similar block summary approach for bipartite graphs is proposed by Feng, X. He, Konte, et al. (2012), with a more complete encoding. Papadimitriou, Sun, et al. (2008) also focus on bipartite graphs, but the obtained blocks are arranged into a hierarchy, while J. He et al. (2009) consider k-partite graphs.

In order to compress the adjacency matrix of an input graph more efficiently, X. He, Feng, and Plant (2011) look for nodes with similar connection patterns, corresponding to similar rows in the matrix, and encode the differences, possibly in a recursive manner. The approach is used to spot nodes with unusual connections patterns, that do not lend themselves to grouping.

LeFevre and Terzi (2010) propose a supernode summary involving superedge weights that represent the probability that an edge exists for each pair of nodes in the incident supernodes. In one variant of the problem, the MDL principle is used to choose the number k of supernodes that strikes the best balance between model complexity (k) and fit to the data (reconstruction error). K. Lee et al. (2020) also consider summarising graphs by grouping nodes together, but fix a maximum length for the description of the model, i.e. the hypernodes, and look for the

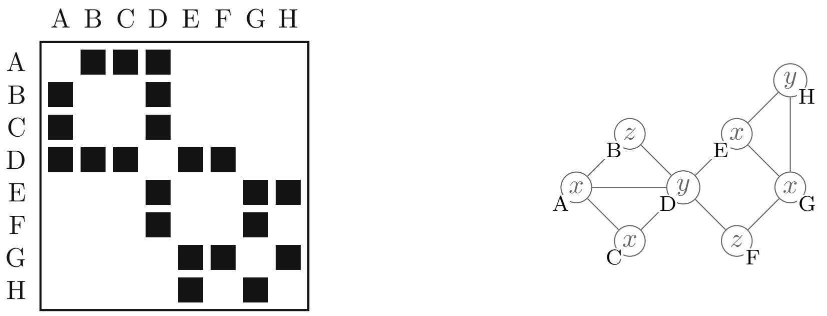


Figure 6: A toy graph, with eight nodes denoted A–H carrying labels from $\{x, y, z\}$ and connected by undirected edges (right). The adjacency matrix (left) captures the structure of the graph, ignoring the labels.

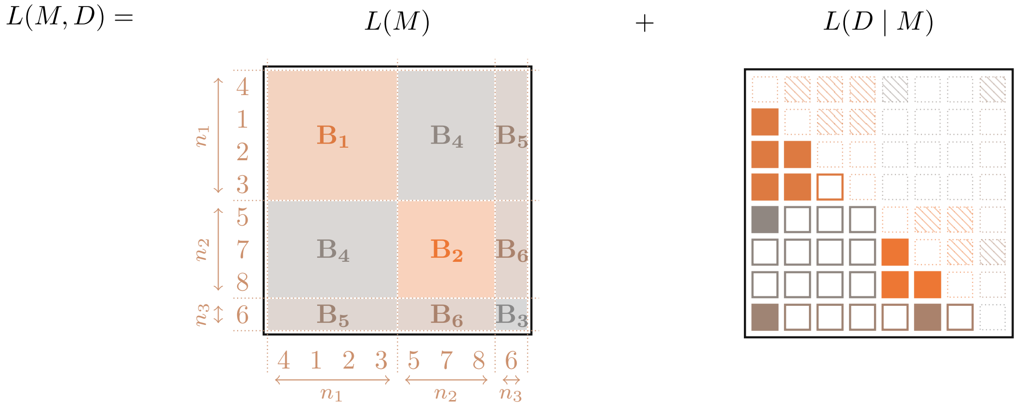


Figure 7: Block-based strategy, example on the toy graph of Figure 6. Ignoring labels and considering its adjacency matrix, the graph can be encoded in a very similar way as a binary tabular dataset (see Section 2.3 and Figure 4). Furthermore, since the graph is undirected, and its matrix hence symmetric, it is enough to encode the lower triangular part of the matrix (depicted with solid lines and colour fill), from which the upper triangular part (depicted with dotted lines and hatch pattern) can be reconstructed. More intense shades of orange represent higher probabilities of ones within the corresponding block.

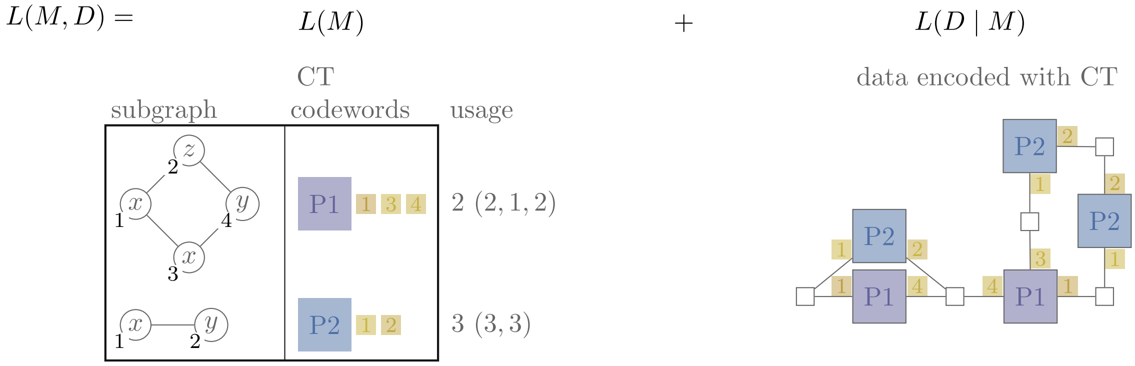


Figure 8: Dictionary-based strategy, example on the toy graph of Figure 6. The idea is similar to the one for binary tabular datasets (see Section 2.3 and Figure 3). However, it is not enough to simply replace the subgraph patterns by their assigned codewords (depicted as blue blocks), the information about how the subgraphs are connected also needs to be encoded. Here, this is done through *ports* associated to the patterns and their assigned codewords (depicted as tan blocks).

summary that minimises the reconstruction error, measured as the length of the description of edge corrections.

Plant, Biedermann, and Böhm (2020) use a MDL-inspired score to learn graph embeddings. That is, the aim is to project the nodes into a multi-dimensional space, so that the structure of the graph is preserved as much as possible and, more specifically, such that connected nodes are placed close to each other. Therefore, the distance between any pair of nodes in the embedding is used to compute a probability that the nodes are connected, which, in turn is used to encode the presence or absence of the corresponding edge. Then, the quality of an embedding can be measured by how much it allows to compress the adjacency matrix.

Chakrabarti, Deepayan (2004). "AutoPart: Parameter-Free Graph Partitioning and Outlier Detection". In: *Proceedings of the European Conference on Knowledge Discovery in Databases, PKDD'04*. Springer, pp. 112–124. DOI: 10.1007/978-3-540-30116-5_13.

Rosvall, Martin and Carl T. Bergstrom (2007). "An information-theoretic framework for resolving community structure in complex networks". In: *Proceedings of the National Academy of Sciences* 104.18, pp. 7327–7331. DOI: 10.1073/pnas.0611034104.

Navlakha, Saket, Rajeev Rastogi, and Nisheeth Shrivastava (2008). "Graph Summarization with Bounded Error". In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD'08*. ACM, pp. 419–432. DOI: 10.1145/1376616.1376661.

Papadimitriou, Spiros, Jimeng Sun, et al. (2008). "Hierarchical, Parameter-Free Community Discovery". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'08*. Springer, pp. 170–187. DOI: 10.1007/978-3-540-87481-2_12.

He, Jingrui et al. (2009). "PaCK: Scalable parameter-free clustering on K-partite graphs". In: *Proceedings of the 2006 SIAM International Conference on Data Mining, SDM'09*. SIAM, pp. 1278–1287.

LeFevre, Kristen and Evimaria Terzi (2010). "GraSS: Graph Structure Summarization". In: *Proceedings of the 2010 SIAM International Conference on Data Mining, SDM'10*. SIAM, pp. 454–465. DOI: 10.1137/1.9781611972801.40.

He, Xiao, Jing Feng, and Claudia Plant (2011). "Automatically Spotting Information-Rich Nodes in Graphs". In: *Proceedings of the 11th IEEE International Conference on Data Mining Workshops, ICDMW'11*. IEEE Computer Society, pp. 941–948. DOI: 10.1109/ICDMW.2011.37.

Akoglu, Leman, Hanghang Tong, Brendan Meeder, et al. (2012). "PICS: Parameter-free Identification of Cohesive Subgroups in Large Attributed Graphs". In: *Proceedings of the 2012 SIAM International Conference on Data Mining, SDM'12*. SIAM, pp. 439–450. DOI: 10.1137/1.9781611972825.38.

Feng, Jing, Xiao He, Bettina Konte, et al. (2012). "Summarization-based mining bipartite graphs". In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'12*. ACM, pp. 1249–1257. DOI: 10.1145/2339530.2339725.

Khan, Kifayat Ullah, Waqas Nawaz, and Young-Koo Lee (2014). "Set-Based Unified Approach for Attributed Graph Summarization". In: *Proceedings of the 4th IEEE International Conference on Big Data and Cloud Computing, BDCloud'14*. IEEE Computer Society, pp. 378–385. DOI: 10.1109/BDCloud.2014.108.

Khan, Kifayat Ullah (2015). "Set-based approach for lossless graph summarization using Locality Sensitive Hashing". In: *Proceedings of the 31st IEEE International Conference on Data Engineering Workshops, ICDEW'15*. IEEE Computer Society, pp. 255–259. DOI: 10.1109/ICDEW.2015.7129586.

Khan, Kifayat Ullah, Waqas Nawaz, and Young-Koo Lee (2015a). "Lossless graph summarization using dense subgraphs discovery". In: *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, IMCOM'15*. ACM, pp. 1–7. DOI: 10.1145/2701126.2701157.

— (2015b). "Set-based approximate approach for lossless graph summarization". In: *Computing* 97.12, pp. 1185–1207. DOI: 10.1007/s00607-015-0454-9.

Lee, Kyuhan et al. (2020). "SSumM: Sparse Summarization of Massive Graphs". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'20*. ACM, pp. 144–154. DOI: 10.1145/3394486.3403057.

Plant, Claudia, Sonja Biedermann, and Christian Böhm (2020). "Data Compression as a Comprehensive Framework for Graph Drawing and Representation Learning". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'20*. ACM, pp. 1212–1222. DOI: 10.1145/3394486.3403174.

6.2 Grouping nodes into blocks in dynamic graphs

Sun et al. (2007) introduce a block summary approach for dynamic graphs, extended to multiple dimensions or contexts by Jiang, Faloutsos, and Han (2016). Araujo, Papadimitriou, et al. (2014) also propose a block summary approach for dynamic graphs, which they later extend to multiple dimensions or contexts as represented by qualitative labels on the edges (Araujo, Günnemann, Papadimitriou, et al., 2016).

Sun, Jimeng et al. (2007). "GraphScope: parameter-free mining of large time-evolving graphs". In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'07*. ACM, pp. 687–696. DOI: 10.1145/1281192.1281266.

Araujo, Miguel, Spiros Papadimitriou, et al. (2014). "Com2: Fast Automatic Discovery of Temporal ('Comet') Communities". In: *Proceedings of 18th Pacific-Asia Conference on the Advances in Knowledge Discovery and Data Mining, PAKDD'14*. Springer, pp. 271–283. DOI: 10.1007/978-3-319-06605-9_23.

Araujo, Miguel, Stephan Günnemann, Spiros Papadimitriou, et al. (2016). "Discovery of "comet" communities in temporal and labeled graphs COM^2". In: *Knowledge and Information Systems* 46.3, pp. 657–677. DOI: 10.1007/s10115-015-0847-2.

Jiang, Meng, Christos Faloutsos, and Jiawei Han (2016). "CatchTartan: Representing and Summarizing Dynamic Multicontextual Behaviors". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'16*. ACM, pp. 945–954. DOI: 10.1145/2939672.2939749.

6.3 Finding hyberbolic communities

Instead of looking for blocks of uniform density and motivated by the observation that node degrees in real-world networks often follow a power-law distribution, Araujo, Günnemann, Mateos, et al. (2014) propose the model of *hyperbolic communities*. The name refers to the fact that when nodes in such communities are ordered by degree, edges in the adjacency matrix mostly end up below a hyperbola.

Kang and Faloutsos (2011) (also Lim, Kang, and Faloutsos, 2014) decompose the input graph into hubs and spokes, with superhubs connecting the hubs recursively, and introduce a cost to evaluate how well the decomposition allows to compress the graph. This type of decomposition is proposed as an alternative to a decomposition into cliques, referred to as "cavemen communities".

Kang, U and Christos Faloutsos (2011). "Beyond 'Caveman Communities': Hubs and Spokes for Graph Compression and Mining". In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM'11*. IEEE Computer Society, pp. 300–309. DOI: 10.1109/ICDM.2011.26.

Araujo, Miguel, Stephan Günnemann, Gonzalo Mateos, et al. (2014). "Beyond Blocks: Hyperbolic Community Detection". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'14*. Springer, pp. 50–65. DOI: 10.1007/978-3-662-44848-9_4.

Lim, Yongsub, U Kang, and Christos Faloutsos (2014). "SlashBurn: Graph Compression and Mining beyond Caveman Communities". In: *IEEE Transactions on Knowledge and Data Engineering* 26.12, pp. 3077–3089. DOI: 10.1109/TKDE.2014.2320716.

6.4 The Map Equation

Rosvall and Bergstrom (2008) propose a method to reveal the important connectivity structure of weighted directed graphs. The approach assigns codes to nodes in such a way that random walks over the graph can be described succinctly. Furthermore, nodes are partitioned into modules so that the codes for nodes are unique within each module but can be reused between modules. A walk over the graph can then be described using a combination of codes indicating transitions between modules and lists of the successive nodes encountered within each module. The resulting two-level summary of the graph maps its main structures and the connections between and within them, and the approach is therefore referred to as the *Map Equation* (Rosvall, Axelsson, and Bergstrom, 2009) or the *Infomap* algorithm.[3]

Later, refinements and extensions of the method were proposed, to study changes in the connectivity structure over time (Rosvall and Bergstrom, 2010), reveal multi-level hierarchical connectivity structure (Rosvall and Bergstrom, 2011), support overlaps between modules (Viamontes Esquivel and Rosvall, 2011), among others (Bohlin et al., 2014; De Domenico et al., 2015; Edler, Bohlin, and Rosvall, 2017; Emmons and Mucha, 2019; Calatayud et al., 2019). In particular, the method has found application in the analysis of ecological communities (Edler, Guedes, et al., 2017; Blanco et al., 2021; Rojas et al., 2021).

Rosvall, Martin and Carl T. Bergstrom (2008). "Maps of random walks on complex networks reveal community structure". In: *Proceedings of the National Academy of Sciences* 105.4, pp. 1118–1123. DOI: 10.1073/pnas.0706851105.

Rosvall, Martin, D. Axelsson, and Carl T. Bergstrom (2009). "The map equation". In: *The European Physical Journal Special Topics* 178.1, pp. 13–23. DOI: 10.1140/epjst/e2010-01179-1.

[3] https://www.mapequation.org/

Rosvall, Martin and Carl T. Bergstrom (2010). “Mapping Change in Large Networks”. In: *PLoS ONE* 5.1, pp. 1–7. DOI: 10.1371/journal.pone.0008694.

— (2011). “Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems”. In: *PLOS ONE* 6.4, e18209. DOI: 10.1371/journal.pone.0018209.

Viamontes Esquivel, Alcides and Martin Rosvall (2011). “Compression of Flow Can Reveal Overlapping-Module Organization in Networks”. In: *Physical Review X* 1.2, p. 021025. DOI: 10.1103/PhysRevX.1.021025.

Bohlin, Ludvig et al. (2014). “Community Detection and Visualization of Networks with the Map Equation Framework”. In: *Measuring Scholarly Impact: Methods and Practice.* Ed. by Ying Ding, Ronald Rousseau, and Dietmar Wolfram. Springer International Publishing, pp. 3–34.

De Domenico, Manlio et al. (2015). “Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems”. In: *Physical Review X* 5.1, p. 11027. DOI: 10.1103/PhysRevX.5.011027.

Edler, Daniel, Ludvig Bohlin, and Martin Rosvall (2017). “Mapping Higher-Order Network Flows in Memory and Multilayer Networks with Infomap”. In: *Algorithms* 10.4, p. 112. DOI: 10.3390/a10040112.

Edler, Daniel, Thaís Guedes, et al. (2017). “Infomap Bioregions: Interactive Mapping of Biogeographical Regions from Species Distributions”. In: *Systematic Biology* 66.2, pp. 197–204. DOI: 10.1093/sysbio/syw087.

Calatayud, Joaquín et al. (2019). “Exploring the solution landscape enables more reliable network community detection”. In: *Physical Review E* 100.5, p. 052308. DOI: 10.1103/PhysRevE.100.052308.

Emmons, Scott and Peter J. Mucha (2019). “Map equation with metadata: Varying the role of attributes in community detection”. In: *Physical Review E* 100.2, p. 022301. DOI: 10.1103/PhysRevE.100.022301.

Blanco, Fernando et al. (2021). “Punctuated ecological equilibrium in mammal communities over evolutionary time scales”. In: *Science* 372.6539, pp. 300–303. DOI: 10.1126/science.abd5110.

Rojas, Alexis et al. (2021). “A multiscale view of the Phanerozoic fossil record reveals the three major biotic transitions”. In: *Communications Biology* 4.1, pp. 1–8. DOI: 10.1038/s42003-021-01805-y.

6.5 Identifying substructures

Cook and Holder (1994) (also Ketkar, Holder, and Cook, 2005) propose the SUBDUE algorithm to mine substructures from graphs, possibly with labels, using the MDL principle. A substructure of the graph can be encoded and its occurrences in the graph be replaced by a single node. This can be done recursively, generating a hierarchical summary of the original graph. There are two shortcomings to the approach. First, replacing the substructure by a single node does not preserve the complete information about the connections to neighbours. Second, the matching of substructures is done in an approximate way, with an arbitrary fixed cost, rather than a proper encoding of the reconstruction errors (using the MDL principle for this evaluation is left for future work). Substructures are scored individually rather than in combination. The SUBDUE algorithm is used by Jonyer, Holder, and Cook (2004) for the induction of context-free grammars, and by Bloem (2013) in comparative experiments against practical data compression with the GZIP algorithm.

Bloem and Rooij (2018) (also Bloem and Rooij, 2020) propose to use the MDL principle when evaluating the statistical significance of the presence of substructures in a graph.

The VOG algorithm presented by Koutra et al. (2014) (also Koutra et al., 2015) allows to decompose the graph into basic primitives such as cliques, stars, and chains, which can overlap on nodes (but not on edges). Error corrections are then applied, to add and remove spurious edges. This can be seen as a global use of primitives. Liu, Shah, and Koutra (2015) (also Liu, Safavi, and Shah, 2016) use the MDL principle to compare the ability of VOG and graph clustering methods to generate graph summaries. Liu, Safavi, Shah, and Koutra (2018) build on VOG and address some of the shortcomings, such as the bias towards star structures, the inability to exploit edge overlaps, and the dependency on candidate order. Goebl et al. (2016) introduce a similar approach, with some of the same primitives, but prohibiting overlaps with the aim to make visualisation and interpretation easier. The approach presented by Bariatti, Cellier, and Ferré (2020b) removes the limitation to a predefined set of primitives and considers labelled graphs (see Figure 8). The authors later upgraded the approach by generating candidates on-the-fly, thereby providing an anytime mining algorithm (Bariatti, Cellier, and Ferré, 2021), and proposed a visualisation tool for the obtained graph patterns (Bariatti, Cellier, and Ferré, 2020a). This work forms the basis of a doctoral dissertation (Bariatti, 2021).

The approach proposed by Coupette and Vreeken (2021) aims to highlight similarities and differences between graphs, and is akin in spirit to the DIFFNORM algorithm for transactional data (cf. Section 4.3). It looks for a common model consisting of basic primitives, like those used in VOG, such that each graph can be reconstructed based on these primitives adjusted through parameters specific to the graph, as well as additional structures, where necessary.

Feng, X. He, Hubig, et al. (2013) exploit basic structures of graphs, like stars and triangles, to save on the encoding of the adjacency matrix. Such primitives assign a probability to the existence of an edge, which is used to encode it. Which primitive applies is determined in part based on structure information available so far, i.e. previously decoded. This can be seen as a local use of primitives.

Belth et al. (2020) learn relational rules which can be used to summarise knowledge graphs, involving typed edges and nodes.

Cook, Diane J. and Lawrence B. Holder (1994). "Substructure Discovery Using Minimum Description Length and Background Knowledge". In: *Journal of Artificial Intelligence Research* 1.1, pp. 231–255.

Jonyer, Istvan, Lawrence B. Holder, and Diane J. Cook (2004). "Mdl-based context-free graph grammar induction and applications". In: *International Journal on Artificial Intelligence Tools* 13.1, pp. 65–79. DOI: 10.1142/S0218213004001429.

Ketkar, Nikhil S., Lawrence B. Holder, and Diane J. Cook (2005). "Subdue: compression-based frequent pattern discovery in graph data". In: *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations, OSDM'05.* ACM, pp. 71–76. DOI: 10.1145/1133905.1133915.

Bloem, Peter (2013). "Compression-based inference on graph data". In: *Proceedings of the 22nd annual Belgian-Dutch Conference on Machine Learning, BENELEARN'13*.

Feng, Jing, Xiao He, Nina Hubig, et al. (2013). "Compression-Based Graph Mining Exploiting Structure Primitives". In: *Proceedings of the 13th IEEE International Conference on Data Mining, ICDM'13.* IEEE Computer Society, pp. 181–190. DOI: 10.1109/ICDM.2013.56.

Koutra, Danai et al. (2014). "VOG: Summarizing and Understanding Large Graphs". In: *Proceedings of the 2014 SIAM International Conference on Data Mining, SDM'14.* SIAM, pp. 91–99. DOI: 10.1137/1.9781611973440.11.

— (2015). "Summarizing and understanding large graphs". In: *Statistical Analysis and Data Mining* 8.3, pp. 183–202.

Liu, Yike, Neil Shah, and Danai Koutra (2015). *An Empirical Comparison of the Summarization Power of Graph Clustering Methods.* arXiv: 1511.06820.

Goebl, Sebastian et al. (2016). "MeGS: Partitioning Meaningful Subgraph Structures Using Minimum Description Length". In: *Proceedings of the 16th IEEE International Conference on Data Mining, ICDM'16.* IEEE Computer Society, pp. 889–894. DOI: 10.1109/ICDM.2016.0108.

Liu, Yike, Tara Safavi, and Neil Shah (2016). "Reducing Million-Node Graphs to a Few Structural Patterns: A Unified Approach". In: *Proceedings of the 12th International Workshop on Mining and Learning with Graphs, MLG @KDD'16*, p. 8.

Bloem, Peter and Steven de Rooij (2018). *A tutorial on MDL hypothesis testing for graph analysis.* arXiv: 1810.13163.

Liu, Yike, Tara Safavi, Neil Shah, and Danai Koutra (2018). "Reducing large graphs to small supergraphs: a unified approach". In: *Social Network Analysis and Mining* 8.1, p. 17. DOI: 10.1007/s13278-018-0491-4.

Bariatti, Francesco, Peggy Cellier, and Sébastien Ferré (2020a). "GraphMDL Visualizer: Interactive Visualization of Graph Patterns". In: *Proceedings of the Graph Embedding and Mining Workshop GEM@ECML/PKDD'20.* URL: https://hal.inria.fr/hal-03142207.

— (2020b). "GraphMDL: Graph Pattern Selection Based on Minimum Description Length". In: *Proceedings of the 18th International Symposium on Advances in Intelligent Data Analysis, IDA'20.* Springer, pp. 54–66. DOI: 10.1007/978-3-030-44584-3_5.

Belth, Caleb et al. (2020). "What is Normal, What is Strange, and What is Missing in a Knowledge Graph: Unified Characterization via Inductive Summarization". In: *Proceedings of The Web Conference, WWW'20.* ACM, pp. 1115–1126. DOI: 10.1145/3366423.3380189.

Bloem, Peter and Steven de Rooij (2020). "Large-scale network motif analysis using compression". In: *Data Mining and Knowledge Discovery* 34.5, pp. 1421–1453. DOI: 10.1007/s10618-020-00691-y.

Bariatti, Francesco (2021). "Mining Tractable Sets of Graph Patterns with the Minimum Description Length Principle". PhD thesis. Université de Rennes 1. URL: https://hal.inria.fr/tel-03523742.

Bariatti, Francesco, Peggy Cellier, and Sébastien Ferré (2021). "GraphMDL+: interleaving the generation and MDL-based selection of graph patterns". In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing, SAC'21.* ACM, pp. 355–363. DOI: 10.1145/3412841.3441917.

Coupette, Corinna and Jilles Vreeken (2021). "Graph Similarity Description: How Are These Graphs Similar?" In: *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'21.* ACM.

6.6 Identifying substructures in dynamic graphs

Shah, Koutra, Zou, et al. (2015) (also Shah, Koutra, Jin, et al., 2017) extend the VoG approach to dynamic graphs. More specifically, they incorporate the temporal aspect of substructures appearing only at given time steps, across a range of contiguous time steps, periodically, or in a flickering fashion. Therefore, in addition to decomposing the graph into basic structures, one needs to indicate when these structures appear. The Mango algorithm by Saran and Vreeken (2019) also looks for predefined structures in a dynamic graph, aiming more specifically at tracking their evolution through time.

Shah, Neil, Danai Koutra, Tianmin Zou, et al. (2015). “TimeCrunch: Interpretable Dynamic Graph Summarization”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’15*. ACM, pp. 1055–1064. DOI: 10.1145/2783258.2783321.

Shah, Neil, Danai Koutra, Lisa Jin, et al. (2017). “On Summarizing Large-Scale Dynamic Graphs”. In: *IEEE Data Engineering Bulletin* 40.3, pp. 75–88.

Saran, Divyam and Jilles Vreeken (2019). *Summarizing Dynamic Graphs using MDL*. Technical report. Saarland University.

6.7 Finding pathways between nodes

Given a large graph, Akoglu, Chau, et al. (2013) consider the problem of identifying a set of marked nodes. This can be done by listing the node identifiers or by navigating between nodes. The latter strategy requires to choose between the limited number of neighbours of each traversed node, rather than among all possible nodes in the graph, potentially leading to shorter descriptions. In particular, the problem formulated by Akoglu, Chau, et al. (2013) consists in finding the best collection of trees spanning the marked nodes in the graph. The graph as such is not encoded, it is regarded as shared knowledge.

Prakash, Vreeken, and Faloutsos (2014) similarly assume shared knowledge of the graph. The aim is then to transmit the starting points and spread of an epidemic through the graph over a sequence of time steps, assuming a “susceptible–infected” (SI) epidemic model.

Akoglu, Leman, Duen Horng Chau, et al. (2013). “Mining Connection Pathways for Marked Nodes in Large Graphs”. In: *Proceedings of the 2013 SIAM International Conference on Data Mining, SDM’13*. SIAM, pp. 37–45. DOI: 10.1137/1.9781611972832.5.

Prakash, B. Aditya, Jilles Vreeken, and Christos Faloutsos (2014). “Efficiently spotting the starting points of an epidemic in a large graph”. In: *Knowledge and Information Systems* 38.1, pp. 35–59. DOI: 10.1007/s10115-013-0671-5.

7 Temporal data

In this section, we look at data where the attribute values come as a sequence, i.e. in a specific order. In particular, this order might correspond to time, in which case the data is called *temporal.* In some cases only the order matters, whereas in other cases absolute positions are associated to the values, such as timestamps in temporal data. In addition to time, spatial dimension(s) might be associated to the values, resulting in *spatio-temporal* data. The terms *sequential data* and *sequence* are sometimes used to refer more narrowly to sequences of discrete attributes or items, which are typically called *events.* On the other hand, the term *timeseries* is generally used to refer to real-valued attributes sampled at regular or irregular time intervals. Text and genetic data (such as DNA or RNA sequences) fall into the former category. More specifically, such data generally comes in the form of *strings*, that is, as sequences of characters that represent occurrences of single items where the order is meaningful, not the positions. The data might consist of a single long sequence or of a database of multiple, typically shorter, sequences.

As with other types of data, most of the work on mining sequential data can be divided into two main tasks, namely segmentation and frequent pattern mining, corresponding to *block-based* and *dictionary-based* strategies, respectively (cf. Section 2.3). In segmentation problems (a.k.a. change point detection), the aim is to divide the input data into homogeneous blocks or segments, each associated to specific occurrence probabilities of the different events. On the other hand, in frequent pattern mining, the aim is to find recurrent substructures, which are commonly referred to as *episodes* and *motifs* when considering sequences and timeseries, respectively.

For illustrative purposes, we consider a toy sequence, shown in Figure 9, and delineate approaches that follow either strategy. The example shown in Figure 10 illustrates the *block-based* strategy and follows the work of Kiernan and Terzi (2008) (cf. Section 7.2), whereas the example shown in Figure 11 illustrates the *dictionary-based* strategy and follows the work of Tatti and Vreeken (2012b) (cf. Section 7.5). While similar to their counterparts for binary tabular data, approaches for temporal data must account for the order that the special dimension of time imposes on the occurrences.

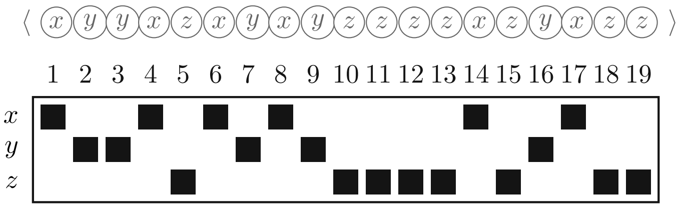


Figure 9: A toy sequence, with nineteen consecutive occurrences over three distinct events x, y and z (top), also represented as a binary matrix (bottom) indicating which event (row) occurs at a given time step (column).

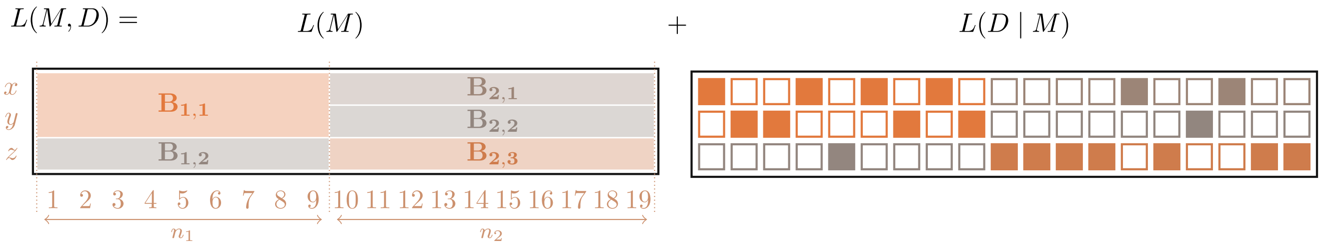


Figure 10: Block-based strategy, example on the toy sequence of Figure 9. The sequence can be encoded in a very similar way as a binary tabular dataset (see Section 2.3 and Figure 4). In this case, however, the order of the columns corresponds to time and is therefore fixed, but the events (i.e. rows) can be arranged into different groups in the different time segments (i.e. column groups). More intense shades of orange represent higher probabilities of ones within the corresponding block.

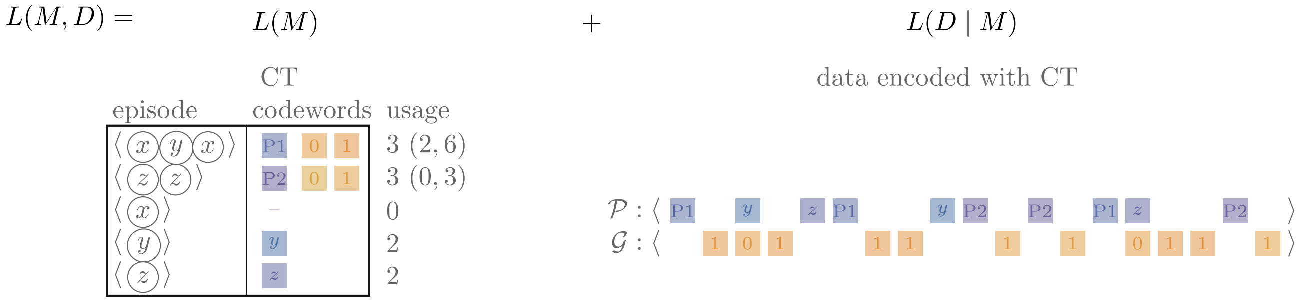


Figure 11: Dictionary-based strategy, example on the toy sequence of Figure 9. The idea is similar to the one for binary tabular datasets (see Section 2.3 and Figure 3). Episode patterns are assigned codewords (depicted as blue blocks). The original sequence is reconstructed by reading codewords from the *pattern stream* ($\mathcal{P}$) and inserting the events of the corresponding episode into the sequence, in order. In addition, each multi-event episode in the code table is associated to an additional pair of codewords (depicted as bronze blocks). These codewords, read from the *gap stream* ($\mathcal{G}$), indicate whether to insert the next element from the current episode (1) or to leave a gap where to insert the next episode (0).

7.1 Finding haplotype blocks

A haplotype, or haploid genotype, is a group of alleles of different genes on a single chromosome, which are closely linked and typically inherited as a unit. Several works have been dedicated to the problem of finding haplotype block boundaries, i.e. identifying block structure in genetic sequences. This requires jointly partitioning multiple aligned strings.

Koivisto, Mikko et al. (2002). “An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries”. In: *Proceedings of the 2003 Pacific Symposium on Biocomputing, PSB’03*. World Scientific, pp. 502–513. DOI: 10.1142/9789812776303_0047.

Anderson, Eric C. and John Novembre (2003). “Finding Haplotype Block Boundaries by Using the Minimum-Description-Length Principle”. In: *American Journal of Human Genetics* 73.2, pp. 336–354.

Greenspan, Gideon and Dan Geiger (2003). “Model-based inference of haplotype block variation”. In: *Proceedings of the seventh annual international conference on Research in computational molecular biology, RECOMB’03*. ACM, pp. 131–137. DOI: 10.1145/640075.640092.

Mannila, Heikki, M. Koivisto, et al. (2003). “Minimum Description Length Block Finder, a Method to Identify Haplotype Blocks and to Compare the Strength of Block Boundaries”. In: *The American Journal of Human Genetics* 73.1, pp. 86–94. DOI: 10.1086/376438.

Greenspan, Gideon and Dan Geiger (2004). “Model-Based Inference of Haplotype Block Variation”. In: *Journal of Computational Biology* 11.2, pp. 493–504. DOI: 10.1089/1066527041410300.

7.2 Segmenting sequences

Several approaches have also been developed for segmenting event sequences more in general.

The method introduced by Kiernan and Terzi (2008) partitions a sequence into time segments, then partitions the events of each segment into groups (see Figure 10). The proposed algorithm is then extended to allow overlaps and gaps between segments (Kiernan and Terzi, 2009a) and a tool to visualise the obtained segmentation is proposed (Kiernan and Terzi, 2009b). P. Wang et al. (2010) further aim to model dependencies between segments.

The algorithm proposed by Lam, Kiseleva, et al. (2014) partitions the alphabet into subsets, then separately encodes the sequence projected on each subset of symbols, as well as a sequence that maps each position to the corresponding subset. Chen, Amiri, and Prakash (2018) adopt a generic point of view on sequence segmentation, considering that the input data can be either univariate or multivariate, consist of categorical or real-valued variables, with no assumption on the underlying distribution. Gautrais et al. (2020) aim to segment a sequence in such a way that each segment contains a collection of recurrent “signature” events. In particular, they consider retail data and apply the approach to the sequences of transactions of individual customers, in order to analyse their shopping behaviour.

Kiernan, Jerry and Evimaria Terzi (2008). “Constructing Comprehensive Summaries of Large Event Sequences”. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’08*. ACM, pp. 417–425. DOI: 10.1145/1401890.1401943.

— (2009a). “Constructing Comprehensive Summaries of Large Event Sequences”. In: *ACM Transactions on Knowledge Discovery from Data* 3.4, 21:1–21:31. DOI: 10.1145/1631162.1631169.

— (2009b). “EventSummarizer: A Tool for Summarizing Large Event Sequences”. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, EDBT’09*. ACM, pp. 1136–1139. DOI: 10.1145/1516360.1516497.

Wang, Peng et al. (2010). “An algorithmic approach to event summarization”. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD’10*. ACM, pp. 183–194. DOI: 10.1145/1807167.1807189.

Lam, Hoang Thanh, Julia Kiseleva, et al. (2014). “Decomposing a sequence into independent subsequences using compression algorithms”. In: *Proceedings of the Workshop on Interactive Data Exploration and Analytic, IDEA @KDD’14*, pp. 67–75.

Chen, Liangzhe, Sorour E. Amiri, and B. Aditya Prakash (2018). “Automatic Segmentation of Data Sequences”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI’18*. Association for the Advancement of Artificial Intelligence.

Gautrais, Clément et al. (2020). “Widening for MDL-Based Retail Signature Discovery”. In: *Proceedings of the 18th International Symposium on Advances in Intelligent Data Analysis, IDA’20*. Springer, pp. 197–209. DOI: 10.1007/978-3-030-44584-3_16.

7.3 Segmenting timeseries

Hu et al. (2011) (also Hu et al., 2013; Hu et al., 2015) propose an approach to represent a timeseries with an Adaptive Piecewise Constant Approximation (APCA) and define the intrinsic cardinality of the data to be the number of distinct constant values in the approximation. In this approach, a timeseries is encoded by specifying the end-points and value of each segments, then listing reconstruction errors. Vespier et al. (2012) also decompose timeseries into segments, but consider components at multiple time-scales, modeled as piecewise constant or polynomial functions.

Matsubara, Sakurai, and Faloutsos (2014) consider the problem of segmenting multivariate timeseries. A timeseries is modeled using a multi-level hidden Markov model (HMM), where high-level states represent regimes that contain lower level states. D. Wu et al. (2020) also consider a problem of timeseries segmentation, and model each segment with a Markov chain.

Rakthanmanon et al. (2011) (also Rakthanmanon et al., 2012) consider the problem of clustering sequential data, in particular such as arises from discretised timeseries where each distinct numerical value is mapped to a distinct symbol. The authors argue that not every value is of interest, because sequences tend to contain meaningless transitions, and that the MDL principle can help identify the segments of interest. Begum et al. (2013) (also Begum et al., 2014) use the clusters obtained with this approach for the task of semi-supervised classification.

Hu, Bing et al. (2011). "Discovering the Intrinsic Cardinality and Dimensionality of Time Series Using MDL". In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM'11*. IEEE Computer Society, pp. 1086–1091. DOI: 10.1109/ICDM.2011.54.

Rakthanmanon, Thanawin et al. (2011). "Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data". In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM'11*. IEEE Computer Society, pp. 547–556. DOI: 10.1109/ICDM.2011.146.

— (2012). "MDL-based time series clustering". In: *Knowledge and Information Systems* 33.2, pp. 371–399. DOI: 10.1007/s10115-012-0508-7.

Vespier, Ugo et al. (2012). "MDL-Based Analysis of Time Series at Multiple Time-Scales". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'12*. Springer, pp. 371–386. DOI: 10.1007/978-3-642-33486-3_24.

Begum, Nurjahan et al. (2013). "Towards a minimum description length based stopping criterion for semi-supervised time series classification". In: *Proceedings of the 14th IEEE International Conference on Information Reuse Integration, IRI'13*. IEEE Computer Society, pp. 333–340. DOI: 10.1109/IRI.2013.6642490.

Hu, Bing et al. (2013). "Towards Discovering the Intrinsic Cardinality and Dimensionality of Time Series Using MDL". In: *Proceedings of the Ray Solomonoff 85th Memorial Conference, Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*. Springer, pp. 184–197. DOI: 10.1007/978-3-642-44958-1_14.

Begum, Nurjahan et al. (2014). "A Minimum Description Length Technique for Semi-Supervised Time Series Classification". In: *Integration of Reusable Systems*. Advances in Intelligent Systems and Computing, pp. 171–192. DOI: 10.1007/978-3-319-04717-1_8.

Matsubara, Yasuko, Yasushi Sakurai, and Christos Faloutsos (2014). "AutoPlait: automatic mining of co-evolving time sequences". In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD'14*. ACM, pp. 193–204. ISBN: 978-1-4503-2376-5. DOI: 10.1145/2588555.2588556.

Hu, Bing et al. (2015). "Using the minimum description length to discover the intrinsic cardinality and dimensionality of time series". In: *Data Mining and Knowledge Discovery* 29.2, pp. 358–399. DOI: 10.1007/s10618-014-0345-2.

Wu, Daoping et al. (2020). "Modeling Piece-Wise Stationary Time Series". In: *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'20*. IEEE Computer Society, pp. 3817–3821. DOI: 10.1109/ICASSP40776.2020.9053470.

7.4 Mining substrings

The algorithm devised by Evans, Markham, et al. (2006) (also Markham et al., 2009) searches for the best set of substrings to encode an input string according to the proposed Optimal Symbol Compression Ratio (OSCR) (Evans, Saulnier, and Bush, 2003). The algorithm, which has been applied primarily to analyse genetic sequences (Evans, Kourtidis, et al., 2007), is iterative, at each step picking the substring that compresses most and replacing it by a temporary code. Selected substrings can be recursive, in the sense that they contain previously selected substrings. In the end, the selected substrings are assigned codes using Huffman coding.

Evans, Scott, Gary Saulnier, and Stephen F Bush (2003). “A New Universal Two Part Code for Estimation of String Kolmogorov Complexity and Algorithmic Minimum Sufficient Statistic”. In: *Proceedings of the DIMACS Workshop on Complexity and Inference*.

Evans, Scott, T. Stephen Markham, et al. (2006). “An Improved Minimum Description Length Learning Algorithm for Nucleotide Sequence Analysis”. In: *Proceedings of the 2006 Fortieth Asilomar Conference on Signals, Systems and Computers, ACSSC’06*, pp. 1843–1850. DOI: 10.1109/ACSSC.2006.355081.

Evans, Scott, Antonis Kourtidis, et al. (2007). “MicroRNA Target Detection and Analysis for Genes Related to Breast Cancer Using MDLcompress”. In: *EURASIP Journal on Bioinformatics and Systems Biology* 2007.1, p. 43670. DOI: 10.1186/1687-4153-2007-43670.

Markham, T. Stephen et al. (2009). “Implementation of an Incremental MDL-Based Two Part Compression Algorithm for Model Inference”. In: *Proceedings of the 2009 Data Compression Conference, DCC’09*, pp. 322–331. DOI: 10.1109/DCC.2009.66.

7.5 Mining episodes from sequences

Lam, Mörchen, et al. (2012) propose to encode timestamped sequences with absolute positioning. That is, the positions of covered occurrences are listed separately for each singleton event or selected subsequence. A fixed-length code is used, so all elements (event or position) cost the same and, in particular, occurrences can appear arbitrarily far apart with no penalty. Follow-up work (Lam, Mörchen, et al., 2014) focuses on strings. The proposed algorithms have the same names (SeqKrimp and GoKrimp), but use a different encoding mechanism. Specifically, having constructed a dictionary mapping subsequences to codewords, each match of a selected subsequence is replaced by its associated codeword, followed by Elias codes indicating the gaps between occurrences of the successive events of the subsequence. For a given subsequence, a subroutine is proposed to find the matches with minimum gap cost. Lam, Calders, et al. (2013) consider a similar problem in a streaming setting. The proposed encoding points back to the previous occurrence of the subsequence, with a flag to indicate when an extended subsequence should be recorded as new, that is, added to the dictionary.

The Sqs (“squeeze”) algorithm of Tatti and Vreeken (2012b) follows a dictionary-based strategy and is similar to Krimp but for sequences (see Figure 11). Each selected subsequence, or episode, is assigned a codeword representing it, as well as a pair of codewords representing gap (move to next position) and fill (insert event) operations. Gaps are allowed but not interleaving. In other words, gaps must be filled by singletons. This work is then extended in multiple ways, to take into account an ontology over the events, resulting in algorithm Nemo (Grosse and Vreeken, 2017) and by adding support for rich interleaving and choice of events in patterns, resulting in algorithm Squish (Bhattacharyya and Vreeken, 2017).

After focusing on the analysis of seismic data, aiming to cluster and compare seismograms represented as multiple aligned sequences (Bertens and Siebes, 2014), Bertens, Vreeken, and Siebes (2016a) consider multivariate event sequences more in general and propose algorithm Ditto, which can be seen as an extension of Sqs to handle multivariate patterns. The work constitutes the basis of a dissertation focused on detecting anomalies and mining multivariate event sequences, also in combination, i.e. employing Ditto to detect anomalies in such sequences (Bertens, 2017, Chapter 7). Hinrichs and Vreeken (2017) use compression and the Sqs algorithm to analyse the similarities between sequence databases in terms of occurring sequential patterns, focusing mostly on text data. The proposed algorithm, called SqsNorm, provides for sequential data the type of analysis that DiffNorm allows for transactional data (cf. Section 4.3).

The approach proposed by Wiegand, Klakow, and Vreeken (2021) is clearly related to Sqs and Squish but aims to summarise entire complex event sequences, rather than capturing fragmentary behaviour. The models considered resemble Petri nets or finite state machines and specify conditional transitions between events. The data is then represented as a succession of instructions that, when fed through the model, allow to reconstruct the original event log. The authors later present a similar model called event-flow graph (Wiegand, Klakow, and Vreeken, 2022). Instead of pattern nodes, this model involves rules defined over attribute vectors associated to the sequences.

Cüppers and Vreeken (2020) aim to identify sequential patterns that reliably predict the impending occurrence of an event of interest. In other words, they look for a set of rules, but in sequential rather than transactional data (cf. Section 4.4). Along similar lines, Bourrand et al. (2021a) (also Bourrand et al., 2021b) aim to discover a compact set of sequential rules from a single long event sequence.

Fowkes and Sutton (2016) propose a generative probabilistic model of sequence databases. The authors discuss the connection between probabilistic modeling and description length, and compare their proposed algorithm, which is not based on the MDL principle, to Sqs and GoKrimp.

Ibrahim, S. Sastry, and P. S. Sastry (2016) consider sequences with timestamps and patterns that consist of subsequences with fixed inter-event times. The data is encoded by listing the patterns, along with their occurrences. More specifically, for each subsequence, the events and inter-event times are specified, as well as the timestamp of the first event of each occurrence. Mitra and P. S. Sastry (2019) propose a generative statistical model (a hidden Markov model, HMM) as justification for this encoding. Y. Yan et al. (2018) look for patterns in a stream of sequential data where each event is associated to a timestamp, using a sliding window and maximum gap constraint.

Lam, Hoang Thanh, Fabian Mörchen, et al. (2012). “Mining Compressing Sequential Patterns.” In: *Proceedings of the 2012 SIAM International Conference on Data Mining, SDM’12*. SIAM, pp. 319–330. DOI: 10.1137/1.9781611972825.28.

Tatti, Nikolaj and Jilles Vreeken (2012b). “The Long and the Short of it: Summarising Event Sequences with Serial Episodes”. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’12*. ACM, pp. 462–470.

Lam, Hoang Thanh, Toon Calders, et al. (2013). “Zips: Mining Compressing Sequential Patterns in Streams”. In: *Proceedings of the Workshop on Interactive Data Exploration and Analytics, IDEA @KDD’13*. ACM, pp. 54–62. DOI: 10.1145/2501511.2501520.

Bertens, Roel and Arno Siebes (2014). “Characterising Seismic Data”. In: *Proceedings of the 2014 SIAM International Conference on Data Mining, SDM’14*. SIAM, pp. 884–892. DOI: 10.1137/1.9781611973440.101.

Lam, Hoang Thanh, Fabian Mörchen, et al. (2014). “Mining Compressing Sequential Patterns”. In: *Statistical Analysis and Data Mining* 7.1, pp. 34–52. DOI: 10.1002/sam.11192.

Bertens, Roel, Jilles Vreeken, and Arno Siebes (2016a). “Keeping it Short and Simple: Summarising Complex Event Sequences with Multivariate Patterns”. In: *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’16*. ACM.

— (2016b). *Keeping it Short and Simple: Summarising Complex Event Sequences with Multivariate Patterns.* arXiv: 1512.07056.

Fowkes, Jaroslav and Charles Sutton (2016). “A Subsequence Interleaving Model for Sequential Pattern Mining”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’16*. ACM, pp. 835–844.

Ibrahim, A., Shivakumar Sastry, and P. S. Sastry (2016). “Discovering compressing serial episodes from event sequences”. In: *Knowledge and Information Systems* 47.2, pp. 405–432. DOI: 10.1007/s10115-015-0854-3.

Bertens, Roel (2017). “Insight in Information : from Abstract to Anomaly”. PhD thesis. Universiteit Utrecht.

Bhattacharyya, Apratim and Jilles Vreeken (2017). “Efficiently Summarising Event Sequences with Rich Interleaving Patterns”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining, SDM’17*. SIAM.

Grosse, Kathrin and Jilles Vreeken (2017). “Summarising Event Sequences using Serial Episodes and an Ontology”. In: *Proceedings of the Workshop on Interactions between Data Mining and Natural Language Processing @ECML/PKDD’17*.

Hinrichs, Frauke and Jilles Vreeken (2017). “Characterising the Difference and the Norm between Sequence Databases”. In: *Proceedings of the Workshop on Interactions between Data Mining and Natural Language Processing @ECML/PKDD’17*.

Yan, Yizhou et al. (2018). “SWIFT: Mining Representative Patterns from Large Event Streams”. In: *Proc. VLDB Endow.* 12.3, pp. 265–277. DOI: 10.14778/3291264.3291271.

Mitra, Soumyajit and P. S. Sastry (2019). *Summarizing Event Sequences with Serial Episodes: A Statistical Model and an Application.* arXiv: 1904.00516.

Cüppers, Joscha and Jilles Vreeken (2020). “Just Wait For It. . . Mining Sequential Patterns with Reliable Prediction Delays”. In: *Proceedings of the 20th IEEE International Conference on Data Mining, ICDM’20*. IEEE Computer Society.

Bourrand, Erwan et al. (2021a). “Discovering Useful Compact Sets of Sequential Rules in a Long Sequence”. In: *Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence, ICTAI’21*. IEEE Computer Society, pp. 1295–1299. DOI: 10.1109/ICTAI52525.2021.00204.

— (2021b). *Discovering Useful Compact Sets of Sequential Rules in a Long Sequence.* arXiv: 2109.07519.

Wiegand, Boris, Dietrich Klakow, and Jilles Vreeken (2021). “Mining Easily Understandable Models from Complex Event Logs”. In: *Proceedings of the 2021 SIAM International Conference on Data Mining, SDM’21*. SIAM, pp. 244–252. DOI: 10.1137/1.9781611976700.28.

— (2022). “Mining Interpretable Data-to-Sequence Generators”. In: *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI’22*. Association for the Advancement of Artificial Intelligence.

7.6 Mining motifs from timeseries

Tanaka and Uehara (2003) (also Tanaka, Iwamoto, and Uehara, 2005) look for motifs in timeseries, representing discretised values as integers using a fixed-length code.

Shokoohi-Yekta et al. (2015) consider the problem of extracting rules from timeseries, aiming to match shapes rather than the precise values. The proposed score is used to evaluate the consequent of candidate rules, allowing to compare consequents of different lengths. The score evaluates the compression gain resulting from specifying the motif once and then listing the errors for each occurrence, instead of listing the actual values for each occurrence. This is applied to evaluate candidates individually, not as a set.

Tanaka, Yoshiki and Kuniaki Uehara (2003). "Discover motifs in multi-dimensional time-series using the principal component analysis and the MDL principle". In: *Proceedings of the 3rd international conference on Machine learning and data mining in pattern recognition, MLDM'03.* Springer, pp. 252–265.

Tanaka, Yoshiki, Kazuhisa Iwamoto, and Kuniaki Uehara (2005). "Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle". In: *Machine Learning* 58.2, pp. 269–300. DOI: 10.1007/s10994-005-5829-2.

Shokoohi-Yekta, Mohammad et al. (2015). "Discovery of Meaningful Rules in Time Series". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'15.* ACM, pp. 1085–1094. DOI: 10.1145/2783258.2783306.

7.7 Mining periodic patterns

Exploiting regularities not only about what happens, that is, finding coordinated event occurrences, but also about when it happens, that is, finding consistent inter-occurrence time intervals, can allow to further compress the data.

In the context of "smart homes" and health monitoring, Heierman, Youngblood, and Cook (2004) look for periodically repeating events or sets of events in a sequence, with a MDL criterion to identify interesting candidates, which are then used to automatically construct a Markov model (HPOMDP) (also Heierman and Cook, 2003; Das and Cook, 2004; Youngblood et al., 2005). The work of Rashidi and Cook (2013) shares the same context and goal, further accounting for discontinuities in the repetitions and variations in the order of the events.

Galbrun et al. (2018) introduce patterns involving nested periodic recurrences of different events and a method for constructing them by combining simple cycles into increasingly complex and expressive patterns.

Heierman, Edwin O. and Diane J. Cook (2003). "Improving home automation by discovering regularly occurring device usage patterns". In: *Proceedings of the 3rd IEEE International Conference on Data Mining, ICDM'03.* IEEE Computer Society, pp. 537–540. DOI: 10.1109/ICDM.2003.1250971.

Das, Sajal K. and Diane J. Cook (2004). "Health Monitoring in an Agent-Based Smart Home". In: *Proceedings of the International Conference on Smart Homes and Health Telematics, ICOST'04.* IOS Press, pp. 3–14.

Heierman, Edwin O., G. Michael Youngblood, and Diane J. Cook (2004). "Mining temporal sequences to discover interesting patterns". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'19.* ACM.

Youngblood, G. Michael et al. (2005). "Automated HPOMDP Construction through Data-mining Techniques in the Intelligent Environment Domain". In: *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference, FLAIRS'05.*

Rashidi, Parisa and Diane J. Cook (2013). "COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems". In: *ACM Transactions on Intelligent Systems and Technology* 4.4, 64:1–64:20. DOI: 10.1145/2508037.2508045.

Galbrun, Esther et al. (2018). "Mining Periodic Patterns with a MDL Criterion". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'18*, pp. 535–551.

7.8 Trajectories

Phan et al. (2013) consider data that represent a collection of moving objects, each associated at each timestamp with a geospatial position. As a pre-processing step, the objects must be clustered based on proximity, separately for each timestamp. An object is allowed to belong to several clusters at any given timestamp. The goal is then to find a sequence of clusters (at most one per timestamp) having objects in common. The result is called a *swarm* and intended to represent objects moving together. When encoding the trajectory of an object (as a sequence of clusters) patterns can be used whole, or from/to an intermediate position.

Also considering spatio-temporal data but in a different scenario, Zhao et al. (2019) aim to mine frequent patterns in trajectories over a road network. Mapping the trajectories to the corresponding road segments effectively turns the problem into a frequent sequence mining problem. The MDL principle is used to formulate a problem of trajectory spatial compression, addressed using a dictionary-based strategy.

Phan, Nhat Hai et al. (2013). "Mining Representative Movement Patterns through Compression". In: *Advances in Knowledge Discovery and Data Mining*. Springer, pp. 314–326. DOI: 10.1007/978-3-642-37453-1_26.

Zhao, Peng et al. (2019). "CLEAN: Frequent Pattern-Based Trajectory Spatial-Temporal Compression on Road Networks". In: *Proceedings of the 20th IEEE International Conference on Mobile Data Management, MDM'19*. IEEE Computer Society, pp. 605–610. DOI: 10.1109/MDM.2019.00127.

8 Discussion

Our goal here is to open a discussion on issues relevant to MDL-based methods for pattern mining. In particular, the design of the encoding is a crucial ingredient when developing such a method. We consider different questions that might be raised by the involved choices, regarding, in particular, conformity and suitability with respect to the MDL principle. To illustrate the discussion, we point to various works listed in the previous sections.

8.1 Encoding in question

Alleged infractions to the MDL principle can be of different kinds and degrees of severity. They include cases where *(i)* the assignment of codewords ignores information theory, *(ii)* the proposed encoding is not functional due to some information missing, and *(iii)* the proposed encoding clearly cannot achieve a good compression due to the presence of unnecessary unjustified terms.

Several methods assign the same, typically unit, cost to all encoded elements (which might be items, nodes, edges, events, timestamps, etc. depending on the case, or even entire patterns), so that the description length is simply the number of encoded elements (see for instance Navlakha, Rastogi, and Shrivastava, 2008 in Section 6.1, and Phan et al., 2013; Zhao et al., 2019 in Section 7.8). Some authors motivate this choice by the need to avoid penalising large values, or to circumvent other encoding issues (see for instance Ibrahim, S. Sastry, and P. S. Sastry, 2016; Lam, Mörchen, et al., 2012 in Section 7.5). In the method proposed by Y. Yan et al. (2018) (cf. Section 7.5), for instance, the cost of a pattern is first defined to be the number of characters used to represent it, that is, the number of events plus the number of timestamps. In a second version of the method, this cost is then defined to be equal to one for all patterns, reportedly to avoid bias against patterns involving more events.

Transmitting the dataset through a binary communication channel as efficiently as possible is a thought experiment of sorts that motivates the score. If short codewords are assigned to specific elements because they are deemed more valuable and useful, then other elements will have to be assigned longer codewords, because not everything can be transmitted cheaply. One can think of it as the fundamental limits of information theory, through this compression scenario, forcing the designer of the method to make choices as to what he considers important and interesting. One might argue that using unit costs corresponds to using a fixed-length code and rescaling everything for convenience. This indeed simplifies the design of the encoding, as it avoids making decisions, and in this sense short-circuits the principle.

Small coding elements, such as for example required to delimit patterns in the code table, are often omitted. This is sometimes done deliberately, putting forth, in particular, the use of a pre-defined framework to be filled with the relevant values, which is common to all models and can therefore be ignored (see for instance Vreeken, van Leeuwen, and Siebes, 2011 in Section 4, and van Leeuwen and Galbrun, 2015 in Section 4.4), but is sometimes left unexplained and might seem accidental. In the approach proposed by Lam, Mörchen, et al. (2014) (cf. Section 7.5), it is unclear how the receiver knows where the codewords end when decoding the dictionary. On the other hand, Tanaka and Uehara (2003) (cf. Section 7.6) use a fixed-length code to encode values from a set, but the number of distinct values, which varies for different models, is not transmitted, so that the receiver cannot deduce the codeword length, and hence cannot decode the message.

More substantial pieces might also be missing. For instance, the encodings proposed by Lam, Kiseleva, et al. (2014) (cf. Section 7.2) and by Hu et al. (2011) (cf. Section 7.3) do not account for the transmission of the assignment of symbols to subsets and of the mapping of offset values to codewords for the corrections, respectively, which are needed to reconstruct the data.

Explanations about the encoding are sometimes kept at the level of intuitions, and the details provided can be insufficient to properly understand how it works (see for instance Khan, Nawaz, and Y.-K. Lee, 2015b in Section 6.1, Matsubara, Sakurai, and Faloutsos, 2014 in Section 7.3, Heierman, Youngblood, and Cook, 2004; Rashidi and Cook, 2013 in Section 7.7, and Phan et al., 2013 in Section 7.8).

Arguably, ensuring decodability would in some cases require only minor modifications of the encoding scheme, and would likely have no major impact on the results. Furthermore, how much effort should be spent ensuring that the proposed encoding works is debatable, since it will never be used in practice.

The choice of encoding can sometimes seem sub-optimal, ill-suited or introduce undesirable bias. For instance, Navlakha, Rastogi, and Shrivastava (2008) (cf. Section 6.1) list edge corrections that should be applied to the reconstructed graph, indicating for each one the sign of the correction. It seems, however that this information is unnecessary, as it can be inferred from the reconstructed graph, by checking whether the edge is present (must be deleted) or absent (must be added). Hu et al. (2011) (cf. Section 7.3) encode a list of value corrections using Huffman coding, meaning that having few distinct but recurrent error values is rewarded, not necessarily small ones. Using a universal code for the corrections would instead encourage small error values, which might be more intuitive. In any case, it is advisable to lay bare and motivate the potential biases introduced by the choice of encoding, whenever possible.

Considering a sequence, Lam, Calders, et al. (2013) (cf. Section 7.5) encode the occurrence of an event or subsequence by pointing back to the position of the first occurrence. Pointing back, instead, to the position of the last encountered occurrence would require to encode smaller values and might lead to savings. Keeping track of the order in which the patterns were last encountered and referring to the position in that list, so that repetitions of the same pattern do not fill up the list, is another alternative. There are often different ways to achieve the same purpose, not necessarily with a clear overall best choice. In addition to pointing back to previous occurrences, Lam, Calders, et al. (2013) maintain a dictionary of patterns. It is unclear whether the dictionary is actually needed for the encoding, or is primarily used to recover the encountered patterns.

What is part of the encoding of the model and what is part of the encoding of the data given the model is sometimes not entirely obvious. For example, the algorithm of Lam, Kiseleva, et al. (2014) (cf. Section 7.2) encodes a sequence by partitioning the alphabet and considering separately the subsequences over each subset of symbols. The authors present the term that corresponds to the assignment of positions to subsets as part of the encoding of the model. Debatably, it can be considered instead as part of the encoding of the data given the model, while the assignment of symbols to subsets, which is ignored, would belong to the encoding of the model. Besides, encodings often actually consist of three terms, *(i)* a description of the set of patterns (the model), *(ii)* information to reconstruct the data using these patterns, and *(iii)* a list of corrections to apply to the reconstructed data in order to recover the original data, with the latter two together representing the data given the model.

8.2 Code of choice

Prequential plug-in codes, and refined codes more in general, provide means to avoid unwanted bias arising from arbitrary choices in the encoding (cf. Section 2).

For instance, Budhathoki and Vreeken (2015) use prequential coding for the itemset occurrences in the DiffNorm algorithm (cf. Section 4.3). The choice is especially relevant in this scenario where the goal is to contrast the itemset make-up of different datasets, and not to inspect the usage of itemsets in a particular dataset. Bhattacharyya and Vreeken (2017) as well as Wiegand, Klakow, and Vreeken (2021) use prequential coding for the streams that contain information about pattern occurrences (cf. Section 7.5). Other recent works (Faas and van Leeuwen, 2020; Makhalova, Kuznetsov, and Napoli, 2020; Bloem and Rooij, 2020, cf. Sections 5.2, 5.6 and 6.5, respectively) also use prequential coding, while Bertens, Vreeken, and Siebes (2016a) and Hinrichs and Vreeken (2017) (cf. Section 7.5) both explicitly suggest upgrading the current encoding with a prequential code, as a direction for future work. Going further, Proença, Bäck, and van Leeuwen (2021) improved on their earlier work (Proença and van Leeuwen, 2020a) (cf. Section 4.4) by replacing prequential coding, which is only asymptotically optimal, with normalised maximum likelihood (NML), which is optimal for fixed sample sizes, employing similar techniques as Kontkanen and Myllymäki (2007) (cf. Section 5.6).

However, modern Bayesian and NML codes can be challenging to compute, or even downright infeasible. Furthermore, one-part codes can be less intuitive than two-part codes, and do not provide as direct an access to information about pattern usage. For instance, Mampaey and Vreeken (2010) (cf. Section 5.1) compare two encodings, with and without prequential coding, and, obtaining similar results, choose to proceed with the latter as it is more intuitive. All in all, modern refined codes have improved theoretical properties, but using them to build better methods comes with some challenges.

8.3 The letter or the spirit

Some approaches use the MDL principle to score and compare individual candidate patterns, rather than evaluating them in combination (see for instance Cook and Holder, 1994 in Section 6.5, Shokoohi-Yekta et al., 2015 in Section 7.6, as well as Heierman, Youngblood, and Cook, 2004; Rashidi and Cook, 2013 in Section 7.7).

Considering a two-view dataset, i.e. a dataset consisting of two tables, the approach proposed by van Leeuwen and Galbrun (2015) assumes knowledge of one table to encode the other, and vice versa. Arguably, this approach does not correspond to a practical encoding, like other MDL-based approaches, but also not to a realistic compression scenario, yet it serves as a reasonable motivation for the proposed score.

The proposed score might actually be entirely ad-hoc, in the sense that it does not correspond to the length of an encoding that could be used to represent the data (see for instance Makhalova, Kuznetsov, and Napoli, 2019a in Section 5.6). One might reasonably devise and justify an evaluation measure suited to the problem at hand, but labelling it as following the MDL principle is arbitrary and inappropriate, short of an explanation of how this corresponds to encoding, and can only lead to confusion.

Authors sometimes approach the topic with caution and include disclaimers stating that their proposed methods are inspired by or in the spirit of the MDL principle (see for instance Shokoohi-Yekta et al., 2015 in Section 7.6). This can be seen as a way to allow oneself to take some liberties with the principle, indeed considering it as a source of inspiration rather than as law, but also as a way to preventively fend off criticism and accusations of heresy. There is indeed a range of opinions about how closely one must conform to the MDL principle and to information theory.

8.4 Making comparisons

How to make meaningful comparisons between compression-based scores and between corresponding results requires careful consideration. For instance, one might ponder whether the compression achieved for a dataset is an indication of how much structure is present in it, or at least how much could be detected, and to what extent it can serve as a measure of the performance of the algorithm.

Does it make sense to compare the length of a dataset encoded with the proposed scheme to the original unencoded data? And is it a problem if the latter is shorter? Keeping in mind that compression is used as a tool for comparing models, rather than for practical purposes, we answer both questions in the negative. The compression achieved with the simplest model, be it the code table containing only elementary patterns such as the singleton itemsets, known as the *standard code table* in Krimp (Vreeken, van Leeuwen, and Siebes, 2011 in Section 4), for dictionary-based approaches (cf. Figure 3*(i)*) or the single-block model for block-based approaches (cf. Figure 4*(i)*), is often considered as a basis for comparison. The ratio of the compression achieved with a considered model to the compression achieved with the elementary model, known as the *compression ratio*, is then computed and used to compare different models, with lower compression ratios corresponding to better models. This is a way to normalise the scores and allow more meaningful comparisons and evaluations.

A direct comparison of the raw description lengths, in terms of numbers of bits, of the same data encoded with different methods is typically not meaningful. For instance, it does not make sense to compare the description lengths reported in Figure 3 to those reported in Figure 4. Comparing compression ratios across different methods is not really meaningful either in general. Indeed, an easy way to win this contest would be to design an artificial encoding that penalises very heavily the use of elementary patterns. If the different methods handle compatible pattern languages, comparing the compression ratios achieved when considering as model, in turn, the set of patterns selected by each method and applying either encoding can be of interest, and might shed some light on the respective biases of the methods. If the pattern languages are not compatible, then no quantitative comparison can be devised easily and great care must be taken to choose suitable encodings. Qualitative evaluations of obtained patterns are valuable, despite being subjective and domain dependent. In the end, finding a good set of interesting and interpretable patterns is what matters.

9 Beyond mining patterns with MDL

In this penultimate section, we highlight approaches that do not fall strictly within the category of MDL-based pattern mining methods, yet are clearly related, constitute recently active and fruitful research topics, and might therefore be of interest to the reader. First, we highlight studies of correlation and causality that build on algorithmic information theory in general and, for a few of them, on MDL-based pattern mining techniques more in particular. Second, we outline a framework for pattern mining that relies on a different modeling approach, namely on maximum entropy modeling.

9.1 Correlation and causality

A core data analysis problem consists in detecting the presence, measuring the strength, and inferring the direction of dependencies between variables in an observational dataset. Various methods have been proposed to discover correlated variables and infer the causal structure of a dataset (Pearl, 2009). In particular, efforts have focused on applying the tools of algorithmic information theory (cf. Section 3.2) to these questions (Janzing and Schölkopf, 2010), aiming to increase the scalability of developed methods and reduce their reliance on assumptions about the underlying probabilities and the shape of the relationship linking the variables.

Simply put, looking at how much can be saved by compressing two objects together rather than separately can be used to measure the strength of their correlation. Furthermore, given a pair of objects, comparing how well the first can be compressed given the second, and vice versa, provides an indication about the direction of causality between the objects. More formally, a central principle in causal inference states that if x causes y, it is easier to describe y using x than the other way around (Pearl, 2009). This principle can be formalised in terms of the Kolmogorov complexity (cf. Section 3). Specifically, the conditional Kolmogorov complexity of object x given object y, denoted $K(x \mid y)$, is the length of the shortest program that generates x and halts, having access to the information in y. Then, if x causes y we expect that there exists a shorter algorithm to describe y given x than the other way around, and hence $K(y \mid x) < K(x \mid y)$. However, the Kolmogorov complexity is not computable, and various practical instanciations have been proposed, including based on the cumulative and Shannon entropies (Rissanen and Wax, 1987; Vreeken, 2015) or on the MDL principle, for instance.

In particular, Budhathoki and Vreeken (2017a) propose two algorithmic correlation measures and present practical instanciations based on the MDL principle, using the Slim and Pack algorithms (cf. Sections 4.2 and 5.1, respectively). Budhathoki and Vreeken (2018c) introduce the Origo algorithm to infer the direction of causality between binary variables, also relying on the Pack algorithm (cf. Section 5.1) to instantiate the MDL score.

Several methods have been proposed to infer the direction of causality between pairs of variables using a MDL score, for discrete variables with refined MDL (Budhathoki and Vreeken, 2017b), as well as using classification and regression trees (Marx and Vreeken, 2019a) or global and local regression functions (Marx and Vreeken, 2019c). Given a collection of variables $X_1, \dots, X_m$, and Y, Kaltenpoth and Vreeken (2019) aim to tell whether the X variables jointly cause Y, or whether there is an unobserved confounding variable, the real parent, using probabilistic principal component analysis (PCA) and a MDL score. Mian, Marx, and Vreeken (2021) present a method for learning causal graphs where all edges are directed, using multivariate regression and a MDL score. Marx and Vreeken (2022) aim to elucidate the link between MDL-based estimators and the postulate of algorithmic independence of conditionals that underpins this line of approaches.

Other methods have been presented that rely not on MDL but on other information-theoretic scores such as the Shannon entropy or mutual information and aim to infer the direction of causality between pairs of variables (Kocaoglu et al., 2017; Budhathoki and Vreeken, 2018a), to detect functional dependencies between variables (Mandros, Boley, and Vreeken, 2020; Pennerath, Mandros, and Vreeken, 2020), as well as to detect correlations between subspaces (Nguyen, Mandros, and Vreeken, 2016), to discover causal rules from observational data (Budhathoki, Boley, and Vreeken, 2018) or to detect correlations between categorical variables without assumptions on the distribution (Mandros, Boley, and Vreeken, 2019). Considering temporal data and Granger causality, Budhathoki and Vreeken (2018b) aim to infer the direction of causality between two event sequences using a sequential normalised maximal likelihood (NML) score, while Hlaváčková-Schindler and Plant (2020) aim to detect causality between timeseries that follow a Poisson distribution, using graphical models and a minimum message length (MML) criterion.

Rissanen, Jorma and Mati Wax (1987). “Measures of mutual and causal dependence between two time series”. In: *IEEE Transactions on Information Theory* 33.4, pp. 598–601. DOI: 10.1109/TIT.1987.1057325.

Pearl, Judea (2009). *Causality*. Cambridge university press.

Janzing, Dominik and Bernhard Schölkopf (2010). “Causal Inference Using the Algorithmic Markov Condition”. In: *IEEE Transactions on Information Theory* 56.10, pp. 5168–5194. DOI: 10.1109/TIT.2010.2060095.

Vreeken, Jilles (2015). “Causal Inference by Direction of Information”. In: *Proceedings of the 2015 SIAM International Conference on Data Mining, SDM'15*. SIAM, pp. 909–917. DOI: 10.1137/1.9781611974010.102.

Nguyen, Hoang-Vu, Panagiotis Mandros, and Jilles Vreeken (2016). “Universal Dependency Analysis”. In: *Proceedings of the 2016 SIAM International Conference on Data Mining, SDM'16*. SIAM, pp. 792–800. DOI: 10.1137/1.9781611974348.89.

Budhathoki, Kailash and Jilles Vreeken (2017a). “Correlation by Compression”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining, SDM'17*. SIAM, pp. 525–533. DOI: 10.1137/1.9781611974973.59.

— (2017b). “MDL for Causal Inference on Discrete Data”. In: *Proceedings of the 17th IEEE International Conference on Data Mining, ICDM'17*. IEEE Computer Society, pp. 751–756. DOI: 10.1109/ICDM.2017.87.

Kocaoglu, Murat et al. (2017). “Entropic Causal Inference”. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17.* Association for the Advancement of Artificial Intelligence.

Budhathoki, Kailash, Mario Boley, and Jilles Vreeken (2018). “Rule discovery for exploratory causal reasoning”. In: *Proceedings of the Workshop on Causal Learning @NeurIPS'18.* Neural Information Processing Systems Foundation Inc.

Budhathoki, Kailash and Jilles Vreeken (2018a). “Accurate Causal Inference on Discrete Data”. In: *Proceedings of the 18th IEEE International Conference on Data Mining, ICDM'18.* IEEE Computer Society, pp. 881–886. DOI: 10.1109/ICDM.2018.00105.

— (2018b). “Causal Inference on Event Sequences”. In: *Proceedings of the 2018 SIAM International Conference on Data Mining, SDM'18.* SIAM.

— (2018c). “Origo: causal inference by compression”. In: *Knowledge and Information Systems* 56.2, pp. 285–307. DOI: 10.1007/s10115-017-1130-5.

Kaltenpoth, David and Jilles Vreeken (2019). “We Are Not Your Real Parents: Telling Causal from Confounded using MDL”. In: *Proceedings of the 2007 SIAM International Conference on Data Mining, SDM'19.* SIAM, pp. 199–207. DOI: 10.1137/1.9781611975673.23.

Mandros, Panagiotis, Mario Boley, and Jilles Vreeken (2019). “Discovering Reliable Correlations in Categorical Data”. In: *Proceedings of the 19th IEEE International Conference on Data Mining, ICDM'19.* IEEE Computer Society, pp. 1252–1257. DOI: 10.1109/ICDM.2019.00156.

Marx, Alexander and Jilles Vreeken (2019a). “Causal Inference on Multivariate and Mixed-Type Data”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'19*, pp. 655–671.

— (2019c). “Telling cause from effect by local and global regression”. In: *Knowledge and Information Systems* 60.3, pp. 1277–1305. DOI: 10.1007/s10115-018-1286-7.

Hlaváčková-Schindler, Kateřina and Claudia Plant (2020). “Poisson Graphical Granger Causality by Minimum Message Length”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'20.*

Mandros, Panagiotis, Mario Boley, and Jilles Vreeken (2020). “Discovering dependencies with reliable mutual information”. In: *Knowledge and Information Systems* 62.11, pp. 4223–4253. DOI: 10.1007/s10115-020-01494-9.

Pennerath, Frédéric, Panagiotis Mandros, and Jilles Vreeken (2020). “Discovering Approximate Functional Dependencies using Smoothed Mutual Information”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'20.* ACM.

Mian, Osman, Alexander Marx, and Jilles Vreeken (2021). “Discovering Fully Oriented Causal Networks”. In: *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI'21.* Association for the Advancement of Artificial Intelligence.

Marx, Alexander and Jilles Vreeken (2022). “Formally Justifying MDL-based Inference of Cause and Effect”. In: *Proceedings of the AAAI Workshop on Information Theoretic Causal Inference and Discovery, ITCI'22.*

9.2 Maximum entropy modeling

Considering the core task of mining itemsets and association rules, it became quickly obvious that finding items that frequently co-occur is not enough, and that one needs to consider statistical dependencies between the items (see for instance Silverstein, Brin, and Motwani, 1998 in Section 3.7). In particular, more insight can be obtained by estimating the expected frequency of items co-occurrence and comparing it to the observed frequency. Various probabilistic models can be used for the estimation (Pavlov, Mannila, and Smyth, 2003), including the maximum entropy distribution.

Several approaches are proposed that rely on the maximum entropy distribution to estimate the occurrence frequency of itemsets based on their subsets and contrast this estimate to filter them (Meo, 2000; Jaroszewicz and Simovici, 2002; C. Wang and Parthasarathy, 2006; Tatti, 2008). Statistics of the dataset, such as marginal counts, can also be used to constrain the model (Tatti and Mampaey, 2010).

Going beyond local models, Mampaey, Tatti, and Vreeken (2011) define maximum entropy models for the dataset, which allows them to iteratively select a collection of itemsets that summarises the data well. They implement their approach as the MTV algorithm (see also Section 5.1). Dalleiger and Vreeken (2020a) look for a collection of patterns and partition of the transactions into components, such that patterns might be relevant only to a subset of components. The actual mining is done by alternating between two algorithms; DISC refines the assignment of transactions to components given a collection of patterns, whereas DESC discovers patterns given a partitioning of the data. The latter is essentially a improved variant of the MTV algorithm as it optimises the same score but can additionally deal with different data components.

Simply put, maximum entropy modeling for pattern mining works as follows. Given some properties of the dataset, a probability distribution is computed over datasets possessing these properties in expectation. The maximum entropy distribution is chosen because this distribution makes no additional assumptions beyond the considered properties and is therefore the least biased. The probability of observing each of the different candidate patterns under this distribution, that is, the probability that the pattern occurs in a dataset with the considered properties, is then evaluated. The lower this probability, the more unexpected and surprising the pattern is considered to be, and hence the more interesting it is deemed. Selected patterns can be seen as discovered properties of the dataset. They can be incorporated as constraints and the probability distribution updated, thereby supporting an iterative, potentially interactive, mining process.

Whereas when following the MDL principle we aim to describe the whole dataset as compactly as possible, the goal when using maximum entropy models is to select the most informative patterns. Typically, selecting all non-redundant patterns that convey information would still produce a large output. Therefore, a criterion must be used to decide when to stop, putting in balance the information content of the patterns and the model complexity.

The constraints imposed on the distribution might capture measured properties of the dataset at hand, but might also reflect the expertise and (possibly incorrect) assumptions of the analyst with respect to the data. The evaluation of the patterns is thus designed to take into account the current experience and understanding of the analyst, albeit in a limited manner. For this reason, the resulting interestingness measure is often called "subjective".

De Bie, Kontonasios, and Spyropoulou (2010) (also De Bie, 2011; De Bie, 2013) introduce a framework for data mining based on maximum entropy modeling, sometimes referred to as the *FORSIED* framework, for *Formalising Subjective Interestingness*. They start with the task of mining tiles from a binary database considering assumptions on the row and column marginals (Kontonasios and Bie, 2010), then derive models for different types of assumptions (Kontonasios and De Bie, 2012), data and patterns, such as real-valued tabular data (Kontonasios, Vreeken, and De Bie, 2013) and various kinds of subgraphs (van Leeuwen, De Bie, et al., 2016; Adriaens, Lijffijt, and De Bie, 2019; Deng et al., 2020; Kapoor, Saxena, and van Leeuwen, 2020; Kapoor, Saxena, and van Leeuwen, 2021), also in a visual interactive exploratory setting (Puolamäki et al., 2020).

An intuitive difference between the two families of approaches is that, following the MDL principle, what is most frequent, most expected, results in the most efficient compression. Instead, in maximum entropy modeling, what is most unexpected, deviates most from assumptions, is generally considered most interesting. However, going too far in either direction can be dangerous. Conforming too much to expectations can lead to rather boring results, while very unexpected results can be startling and difficult to interpret.

Choosing the type of patterns of interest and designing the encoding allows to incorporate background information by favouring some patterns over others, yet this is somewhat implicit, indirect and static. Maximum entropy approaches instead require to model assumptions about the data more explicitly. They tend to be fairly computationally intensive, though much less so than randomisation approaches (see for instance Hanhijärvi et al., 2009 in Section 3.7), that need to explicitly generate, and possibly mine, a large number of randomised copies of the dataset to achieve comparably precise evaluation. Yet, as with randomisation approaches, formulating anything but simple assumptions about the distribution can be difficult. On the other hand, unlike MDL-based approaches, most methods relying on the maximum entropy distribution naturally allow for updates, incorporating feedback, and support interactive analysis. That is, in theory, maximum entropy models allow to incorporate diverse background constraints, in a flexible and potentially interactive manner. However, in practise, this is limited by the fact that constraints can quickly render the optimisation unfeasible. As a step towards alleviating this limitation, Dalleiger and Vreeken (2020b) propose an algorithm that dynamically factorises the joint distribution in order to effectively and efficiently approximate the maximum entropy distribution.

Meo, Rosa (2000). "Theory of dependence values". In: *ACM Transactions on Database Systems* 25.3, pp. 380–406. DOI: 10.1145/363951.363956.

Jaroszewicz, Szymon and Dan A. Simovici (2002). "Pruning Redundant Association Rules using Maximum Entropy Principle". In: *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD'02*. Vol. 2336. Springer, pp. 135–147.

Tatti, Nikolaj (2008). "Maximum entropy based significance of itemsets". In: *Knowledge and Information Systems* 17.1, pp. 57–77. DOI: 10.1007/s10115-008-0128-4.

De Bie, Tijl, Kleanthis-Nikolaos Kontonasios, and Eirini Spyropoulou (2010). "A framework for mining interesting pattern sets". In: *SIGKDD Explorations (and Proceedings of the ACM SIGKDD Workshop on Useful Patterns, UP'10)* 12.2, pp. 92–100.

Kontonasios, Kleanthis-Nikolaos and Tijl De Bie (2010). "An Information-Theoretic Approach to Finding Informative Noisy Tiles in Binary Databases". In: *Proceedings of the 2010 SIAM International Conference on Data Mining, SDM'10*. SIAM, pp. 153–164.

Tatti, Nikolaj and Michael Mampaey (2010). "Using background knowledge to rank itemsets". In: *Data Mining and Knowledge Discovery* 21.2, pp. 293–309. DOI: 10.1007/s10618-010-0188-4.

De Bie, Tijl (2011). “An Information Theoretic Framework for Data Mining”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’11.* ACM, pp. 564–572.

Mampaey, Michael, Nikolaj Tatti, and Jilles Vreeken (2011). “Tell me what I need to know: succinctly summarizing data with itemsets”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’11.* ACM, pp. 573–581. DOI: 10.1145/2020408.2020499.

Kontonasios, Kleanthis-Nikolaos and Tijl De Bie (2012). “Formalizing Complex Prior Information to Quantify Subjective Interestingness of Frequent Pattern Sets”. In: *Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis, IDA’12.* Springer, pp. 161–171.

De Bie, Tijl (2013). “Subjective Interestingness in Exploratory Data Mining”. In: *Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis, IDA’12.* Springer, pp. 19–31. DOI: 10.1007/978-3-642-41398-8_3.

Kontonasios, Kleanthis-Nikolaos, Jilles Vreeken, and Tijl De Bie (2013). “Maximum Entropy Models for Iteratively Identifying Subjectively Interesting Structure in Real-Valued Data”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’13.* Springer, pp. 256–271.

van Leeuwen, Matthijs, Tijl De Bie, et al. (2016). “Subjective interestingness of subgraph patterns”. In: *Machine Learning* 105.1, pp. 41–75. DOI: 10.1007/s10994-015-5539-3.

Adriaens, Florian, Jefrey Lijffijt, and Tijl De Bie (2019). “Subjectively interesting connecting trees and forests”. In: *Data Mining and Knowledge Discovery* 33.4, pp. 1088–1124. DOI: 10.1007/s10618-019-00627-1.

Dalleiger, Sebastian and Jilles Vreeken (2020a). “Explainable Data Decompositions”. In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI’20.* Vol. 34. Association for the Advancement of Artificial Intelligence, pp. 3709–3716. DOI: 10.1609/aaai.v34i04.5780.

— (2020b). “The Relaxed Maximum Entropy Distribution and its Application to Pattern Discovery”. In: *Proceedings of the 20th IEEE International Conference on Data Mining, ICDM’20.* IEEE Computer Society.

Deng, Junning et al. (2020). “Explainable Subgraphs with Surprising Densities: A Subgroup Discovery Approach”. In: *Proceedings of the 2020 SIAM International Conference on Data Mining, SDM’20.* SIAM, pp. 586–594. DOI: 10.1137/1.9781611976236.66.

Kapoor, Sarang, Dhish Kumar Saxena, and Matthijs van Leeuwen (2020). “Discovering subjectively interesting multigraph patterns”. In: *Machine Learning* 109.8, pp. 1669–1696. DOI: 10.1007/s10994-020-05873-9.

Puolamäki, Kai et al. (2020). “Interactive visual data exploration with subjective feedback: an information-theoretic approach”. In: *Data Mining and Knowledge Discovery* 34.1, pp. 21–49. DOI: 10.1007/s10618-019-00655-x.

Kapoor, Sarang, Dhish Kumar Saxena, and Matthijs van Leeuwen (2021). “Online summarization of dynamic graphs using subjective interestingness for sequential data”. In: *Data Mining and Knowledge Discovery* 35.1, pp. 88–126. DOI: 10.1007/s10618-020-00714-8.

10 Conclusion

After giving an outline of relevant concepts from information theory and coding, and an aperçu of related theoretical and conceptual contributions, we reviewed MDL-based methods for mining various types of data and patterns. In particular, we focused on aspects related to the design of an encoding scheme, rather than on algorithmic issues for instance, since the former constitutes the most distinctive ingredient of MDL methodologies, but also a major stumbling block and source of contention. We pointed out two main strategies that underpin the majority of approaches and that can be used to categorise them. Namely, we distinguished dictionary-based approaches from block-based approaches. Then, we considered some discussion points pertaining to the use of MDL in pattern mining, and highlighted related problems that constitute promising directions for future research. Indeed, there is still room for further development in mining patterns with MDL-inspired methods, and beyond.

Acknowledgments

The author is grateful to Peggy Cellier for her feedback during the preparation and revisions of the manuscript, to Hugo M. Proença and Jilles Vreeken for their comments on the first version of this document, and to anonymous reviewers for their comments on later versions of this document. The contents of this survey reflect the understanding of the author, any mistakes and misinterpretations are her own.

Constructive comments as well as pointers to missing related works are most welcome and will be considered to prepare a revision of this survey.

References

Adriaens, Florian, Jefrey Lijffijt, and Tijl De Bie (2019). “Subjectively interesting connecting trees and forests”. In: *Data Mining and Knowledge Discovery* 33.4, pp. 1088–1124. DOI: 10.1007/s10618-019-00627-1 (Section 9.2).

Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami (1993). “Mining association rules between sets of items in large databases”. In: *ACM SIGMOD Record* 22.2, pp. 207–216. DOI: 10.1145/170036.170072 (Section 3.7).

Agrawal, Rakesh and Ramakrishnan Srikant (1994). “Fast Algorithms for Mining Association Rules”. In: *Proceedings of 20th International Conference on Very Large Data Bases, VLDB’94*. Morgan Kaufmann, pp. 487–499 (Section 3.7).

Akoglu, Leman, Duen Horng Chau, et al. (2013). “Mining Connection Pathways for Marked Nodes in Large Graphs”. In: *Proceedings of the 2013 SIAM International Conference on Data Mining, SDM’13*. SIAM, pp. 37–45. DOI: 10.1137/1.9781611972832.5 (Section 6.7).

Akoglu, Leman, Hanghang Tong, Brendan Meeder, et al. (2012). “PICS: Parameter-free Identification of Cohesive Subgroups in Large Attributed Graphs”. In: *Proceedings of the 2012 SIAM International Conference on Data Mining, SDM’12*. SIAM, pp. 439–450. DOI: 10.1137/1.9781611972825.38 (Section 6.1).

Akoglu, Leman, Hanghang Tong, Jilles Vreeken, et al. (2012). “Fast and reliable anomaly detection in categorical data”. In: *Proceedings of the 21st ACM international conference on Information and knowledge management, CIKM’12*. ACM, pp. 415–424. DOI: 10.1145/2396761.2396816 (Section 4.3).

Anderson, Eric C. and John Novembre (2003). “Finding Haplotype Block Boundaries by Using the Minimum-Description-Length Principle”. In: *American Journal of Human Genetics* 73.2, pp. 336–354 (Section 7.1).

Aoga, John O. R. et al. (2018). “Finding Probabilistic Rule Lists using the Minimum Description Length Principle”. In: *Proceedings of the International Conference on Discovery Science, DS’18*. Springer, pp. 66–82. DOI: 10.1007/978-3-030-01771-2_5 (Section 4.4).

Araujo, Miguel, Stephan Günnemann, Gonzalo Mateos, et al. (2014). “Beyond Blocks: Hyperbolic Community Detection”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’14*. Springer, pp. 50–65. DOI: 10.1007/978-3-662-44848-9_4 (Section 6.3).

Araujo, Miguel, Stephan Günnemann, Spiros Papadimitriou, et al. (2016). “Discovery of “comet” communities in temporal and labeled graphs COM^2”. In: *Knowledge and Information Systems* 46.3, pp. 657–677. DOI: 10.1007/s10115-015-0847-2 (Section 6.2).

Araujo, Miguel, Spiros Papadimitriou, et al. (2014). “Com2: Fast Automatic Discovery of Temporal (‘Comet’) Communities”. In: *Proceedings of 18th Pacific-Asia Conference on the Advances in Knowledge Discovery and Data Mining, PAKDD’14*. Springer, pp. 271–283. DOI: 10.1007/978-3-319-06605-9_23 (Section 6.2).

Argamon, Shlomo et al. (2004). “Efficient Unsupervised Recursive Word Segmentation Using Minimum Description Length”. In: *Proceedings of the 20th International Conference on Computational Linguistics, COLING’04*. Association for Computational Linguistics, pp. 1058–1064 (Section 3.6).

Asadi, Behzad and Vijay Varadharajan (2019a). *An MDL-Based Classifier for Transactional Datasets with Application in Malware Detection.* arXiv: 1910.03751 (Section 4.6).

— (2019b). *Towards a Robust Classifier: An MDL-Based Method for Generating Adversarial Examples.* arXiv: 1912.05945 (Section 4.6).

Bariatti, Francesco (2021). “Mining Tractable Sets of Graph Patterns with the Minimum Description Length Principle”. PhD thesis. Université de Rennes 1. URL: https://hal.inria.fr/tel-03523742 (Section 6.5).

Bariatti, Francesco, Peggy Cellier, and Sébastien Ferré (2020a). “GraphMDL Visualizer: Interactive Visualization of Graph Patterns”. In: *Proceedings of the Graph Embedding and Mining Workshop GEM@ECML/PKDD’20*. URL: https://hal.inria.fr/hal-03142207 (Section 6.5).

— (2020b). “GraphMDL: Graph Pattern Selection Based on Minimum Description Length”. In: *Proceedings of the 18th International Symposium on Advances in Intelligent Data Analysis, IDA’20*. Springer, pp. 54–66. DOI: 10.1007/978-3-030-44584-3_5 (Section 6.5).

— (2021). “GraphMDL+: interleaving the generation and MDL-based selection of graph patterns”. In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing, SAC’21*. ACM, pp. 355–363. DOI: 10.1145/3412841.3441917 (Section 6.5).

Barron, Andrew, Jorma Rissanen, and Bin Yu (1998). “The minimum description length principle in coding and modeling”. In: *IEEE Transactions on Information Theory* 44.6, pp. 2743–2760. DOI: 10.1109/18.720554 (Section 3.1).

Bastide, Yves et al. (2000). “Mining Minimal Non-redundant Association Rules Using Frequent Closed Itemsets”. In: *Proceedings of the First International Conference on Computational Logic, CL’00*. Springer, pp. 972–986 (Section 3.7).

Begum, Nurjahan et al. (2013). “Towards a minimum description length based stopping criterion for semi-supervised time series classification”. In: *Proceedings of the 14th IEEE International Conference on Information Reuse Integration, IRI’13*. IEEE Computer Society, pp. 333–340. DOI: 10.1109/IRI.2013.6642490 (Section 7.3).

Begum, Nurjahan et al. (2014). “A Minimum Description Length Technique for Semi-Supervised Time Series Classification”. In: *Integration of Reusable Systems.* Advances in Intelligent Systems and Computing, pp. 171–192. DOI: 10.1007/978-3-319-04717-1_8 (Section 7.3).

Belth, Caleb et al. (2020). “What is Normal, What is Strange, and What is Missing in a Knowledge Graph: Unified Characterization via Inductive Summarization”. In: *Proceedings of The Web Conference, WWW’20.* ACM, pp. 1115–1126. DOI: 10.1145/3366423.3380189 (Section 6.5).

Bertens, Roel (2017). “Insight in Information : from Abstract to Anomaly”. PhD thesis. Universiteit Utrecht (Section 7.5).

Bertens, Roel and Arno Siebes (2014). “Characterising Seismic Data”. In: *Proceedings of the 2014 SIAM International Conference on Data Mining, SDM’14.* SIAM, pp. 884–892. DOI: 10.1137/1.9781611973440.101 (Section 7.5).

Bertens, Roel, Jilles Vreeken, and Arno Siebes (2015). *Beauty and Brains: Detecting Anomalous Pattern Co-Occurrences.* arXiv: 1512.07048 (Section 4.3).

— (2016a). “Keeping it Short and Simple: Summarising Complex Event Sequences with Multivariate Patterns”. In: *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’16.* ACM (Section 7.5).

— (2016b). *Keeping it Short and Simple: Summarising Complex Event Sequences with Multivariate Patterns.* arXiv: 1512.07056 (Section 7.5).

— (2017). “Efficiently Discovering Unexpected Pattern-Co-Occurrences”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining, SDM’17.* SIAM, pp. 126–134. DOI: 10.1137/1.9781611974973.15 (Section 4.3).

Bhattacharyya, Apratim and Jilles Vreeken (2017). “Efficiently Summarising Event Sequences with Rich Interleaving Patterns”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining, SDM’17.* SIAM (Section 7.5).

Blanco, Fernando et al. (2021). “Punctuated ecological equilibrium in mammal communities over evolutionary time scales”. In: *Science* 372.6539, pp. 300–303. DOI: 10.1126/science.abd5110 (Section 6.4).

Bloem, Peter (2013). “Compression-based inference on graph data”. In: *Proceedings of the 22nd annual Belgian-Dutch Conference on Machine Learning, BENELEARN’13* (Section 6.5).

— (2016). “Single sample statistics: Exercises in learning from just one example”. PhD thesis. Universiteit van Amsterdam (Section 3.3).

Bloem, Peter and Steven de Rooij (2018). *A tutorial on MDL hypothesis testing for graph analysis.* arXiv: 1810.13163 (Section 6.5).

— (2020). “Large-scale network motif analysis using compression”. In: *Data Mining and Knowledge Discovery* 34.5, pp. 1421–1453. DOI: 10.1007/s10618-020-00691-y (Section 6.5).

Bloem, Peter, Steven de Rooij, and Pieter Adriaans (2015). “Two Problems for Sophistication”. In: *Proceedings of the 26th International Conference on Algorithmic Learning Theory, ALT’15*, pp. 379–394. DOI: 10.1007/978-3-319-24486-0_25 (Section 3.3).

Bobed, Carlos et al. (2019). “Data-driven Assessment of Structural Evolution of RDF Graphs”. In: *Semantic Web – Interoperability, Usability, Applicability* (Section 4.6).

Bohlin, Ludvig et al. (2014). “Community Detection and Visualization of Networks with the Map Equation Framework”. In: *Measuring Scholarly Impact: Methods and Practice.* Ed. by Ying Ding, Ronald Rousseau, and Dietmar Wolfram. Springer International Publishing, pp. 3–34 (Section 6.4).

Boley, Mario, Claudio Lucchese, et al. (2011). “Direct local pattern sampling by efficient two-step random procedures”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’11.* ACM, pp. 582–590. DOI: 10.1145/2020408.2020500 (Section 3.7).

Boley, Mario, Michael Mampaey, et al. (2013). “One Click Mining: Interactive Local Pattern Discovery Through Implicit Preference and Performance Learning”. In: *Proceedings of the Workshop on Interactive Data Exploration and Analytics, IDEA @KDD’13.* ACM, pp. 27–35. DOI: 10.1145/2501511.2501517 (Section 3.7).

Bonchi, Francesco, Matthijs van Leeuwen, and Antti Ukkonen (2011). “Characterizing Uncertain Data using Compression”. In: *Proceedings of the 2011 SIAM International Conference on Data Mining, SDM’11.* SIAM, pp. 534–545 (Section 4.5).

Bourrand, Erwan et al. (2021a). “Discovering Useful Compact Sets of Sequential Rules in a Long Sequence”. In: *Proceedings of the 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence, ICTAI’21.* IEEE Computer Society, pp. 1295–1299. DOI: 10.1109/ICTAI52525.2021.00204 (Section 7.5).

— (2021b). *Discovering Useful Compact Sets of Sequential Rules in a Long Sequence.* arXiv: 2109.07519 (Section 7.5).

Brent, Michael R., Sreerama K. Murthy, and Andrew Lundberg (1995). “Discovering morphemic suffixes: A case study in MDL induction”. In: *Proceedings of the Fifth International Workshop on Artificial Intelligence and Statistics, AISTATS’95.* Springer, pp. 3–12. DOI: 10.1007/978-1-4612-2404-4_1 (Section 3.6).

Budhathoki, Kailash, Mario Boley, and Jilles Vreeken (2018). “Rule discovery for exploratory causal reasoning”. In: *Proceedings of the Workshop on Causal Learning @NeurIPS’18.* Neural Information Processing Systems Foundation Inc. (Section 9.1).

Budhathoki, Kailash and Jilles Vreeken (2015). “The Difference and the Norm – Characterising Similarities and Differences Between Databases”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’15*. Vol. 9285. Springer, pp. 206–223. DOI: 10.1007/978-3-319-23525-7_13 (Section 4.3).

— (2017a). “Correlation by Compression”. In: *Proceedings of the 2017 SIAM International Conference on Data Mining, SDM’17*. SIAM, pp. 525–533. DOI: 10.1137/1.9781611974973.59 (Section 9.1).

— (2017b). “MDL for Causal Inference on Discrete Data”. In: *Proceedings of the 17th IEEE International Conference on Data Mining, ICDM’17*. IEEE Computer Society, pp. 751–756. DOI: 10.1109/ICDM.2017.87 (Section 9.1).

— (2018a). “Accurate Causal Inference on Discrete Data”. In: *Proceedings of the 18th IEEE International Conference on Data Mining, ICDM’18*. IEEE Computer Society, pp. 881–886. DOI: 10.1109/ICDM.2018.00105 (Section 9.1).

— (2018b). “Causal Inference on Event Sequences”. In: *Proceedings of the 2018 SIAM International Conference on Data Mining, SDM’18*. SIAM (Section 9.1).

— (2018c). “Origo: causal inference by compression”. In: *Knowledge and Information Systems* 56.2, pp. 285–307. DOI: 10.1007/s10115-017-1130-5 (Section 9.1).

Calatayud, Joaquín et al. (2019). “Exploring the solution landscape enables more reliable network community detection”. In: *Physical Review E* 100.5, p. 052308. DOI: 10.1103/PhysRevE.100.052308 (Section 6.4).

Chaitin, Gregory J. (1975). “A Theory of Program Size Formally Identical to Information Theory”. In: *Journal of the ACM* 22.3, pp. 329–340. DOI: 10.1145/321892.321894 (Section 3.2).

Chakrabarti, Deepayan (2004). “AutoPart: Parameter-Free Graph Partitioning and Outlier Detection”. In: *Proceedings of the European Conference on Knowledge Discovery in Databases, PKDD’04*. Springer, pp. 112–124. DOI: 10.1007/978-3-540-30116-5_13 (Section 6.1).

Chakrabarti, Deepayan et al. (2004). “Fully Automatic Cross-associations”. In: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’04*. ACM, pp. 79–88. DOI: 10.1145/1014052.1014064 (Section 5.2).

Chandola, Varun and Vipin Kumar (2007). “Summarization – compressing data into an informative representation”. In: *Knowledge and Information Systems* 12.3, pp. 355–378. DOI: 10.1007/s10115-006-0039-1 (Section 3.4).

Chen, Liangzhe, Sorour E. Amiri, and B. Aditya Prakash (2018). “Automatic Segmentation of Data Sequences”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI’18*. Association for the Advancement of Artificial Intelligence (Section 7.2).

Cilibrasi, Rudi and Paul Vitányi (2005). “Clustering by compression”. In: *IEEE Transactions on Information Theory* 51.4, pp. 1523–1545. DOI: 10.1109/TIT.2005.844059 (Section 3.4).

Cook, Diane J. and Lawrence B. Holder (1994). “Substructure Discovery Using Minimum Description Length and Background Knowledge”. In: *Journal of Artificial Intelligence Research* 1.1, pp. 231–255 (Section 6.5).

Coupette, Corinna and Jilles Vreeken (2021). “Graph Similarity Description: How Are These Graphs Similar?” In: *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’21*. ACM (Section 6.5).

Cover, Thomas M. and Joy A. Thomas (2012). *Elements of information theory*. John Wiley & Sons (Section 3).

Creutz, Mathias and Krista Lagus (2002). “Unsupervised discovery of morphemes”. In: *Proceedings of the ACL 2002 workshop on Morphological and phonological learning, MPL’02*. Association for Computational Linguistics, pp. 21–30. DOI: 10.3115/1118647.1118650 (Section 3.6).

Csiszár, Imre and Paul C. Shields (2004). “Information Theory and Statistics: A Tutorial”. In: *Foundations and Trends in Communications and Information Theory* 1.4, pp. 417–528. DOI: 10.1561/0100000004 (Section 3).

Cüppers, Joscha and Jilles Vreeken (2020). “Just Wait For It... Mining Sequential Patterns with Reliable Prediction Delays”. In: *Proceedings of the 20th IEEE International Conference on Data Mining, ICDM’20*. IEEE Computer Society (Section 7.5).

Dalleiger, Sebastian and Jilles Vreeken (2020a). “Explainable Data Decompositions”. In: *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI’20*. Vol. 34. Association for the Advancement of Artificial Intelligence, pp. 3709–3716. DOI: 10.1609/aaai.v34i04.5780 (Section 9.2).

— (2020b). “The Relaxed Maximum Entropy Distribution and its Application to Pattern Discovery”. In: *Proceedings of the 20th IEEE International Conference on Data Mining, ICDM’20*. IEEE Computer Society (Section 9.2).

Das, Sajal K. and Diane J. Cook (2004). “Health Monitoring in an Agent-Based Smart Home”. In: *Proceedings of the International Conference on Smart Homes and Health Telematics, ICOST’04*. IOS Press, pp. 3–14 (Section 7.7).

Davis, Mark (1996). *The Predictive Paradigm – Compression and Model Bias in Human Cognition*. Technical report (Section 3.3).

De Bie, Tijl (2011). “An Information Theoretic Framework for Data Mining”. In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’11*. ACM, pp. 564–572 (Section 9.2).

De Bie, Tijl (2013). “Subjective Interestingness in Exploratory Data Mining”. In: *Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis, IDA'12*. Springer, pp. 19–31. DOI: 10.1007/978-3-642-41398-8_3 (Section 9.2).

De Bie, Tijl, Kleanthis-Nikolaos Kontonasios, and Eirini Spyropoulou (2010). “A framework for mining interesting pattern sets”. In: *SIGKDD Explorations (and Proceedings of the ACM SIGKDD Workshop on Useful Patterns, UP'10)* 12.2, pp. 92–100 (Section 9.2).

De Domenico, Manlio et al. (2015). “Identifying Modular Flows on Multilayer Networks Reveals Highly Overlapping Organization in Interconnected Systems”. In: *Physical Review X* 5.1, p. 11027. DOI: 10.1103/PhysRevX.5.011027 (Section 6.4).

de Marcken, Carl (1995). *The Unsupervised Acquisition of a Lexicon from Continuous Speech.* arXiv: cmp-lg/9512002 (Section 3.6).

De Raedt, Luc and Albrecht Zimmermann (2007). “Constraint-Based Pattern Set Mining”. In: *Proceedings of the 2007 SIAM International Conference on Data Mining, SDM'07.* SIAM, pp. 237–248. DOI: 10.1137/1.9781611972771.22 (Section 3.7).

Deng, Junning et al. (2020). “Explainable Subgraphs with Surprising Densities: A Subgroup Discovery Approach”. In: *Proceedings of the 2020 SIAM International Conference on Data Mining, SDM'20.* SIAM, pp. 586–594. DOI: 10.1137/1.9781611976236.66 (Section 9.2).

Derthick, Mark (1990). *The minimum description length principle applied to feature learning and analogical mapping.* Technical report. MCC (Section 3.5).

— (1991). “A minimal encoding approach to feature discovery”. In: *Proceedings of the Ninth National conference on Artificial intelligence, AAAI'91.* Association for the Advancement of Artificial Intelligence, pp. 565–571 (Section 3.5).

Domingos, Pedro (1999). “The Role of Occam's Razor in Knowledge Discovery”. In: *Data Mining and Knowledge Discovery* 3.4, pp. 409–425. DOI: 10.1023/A:1009868929893 (Section 3.3).

Dowe, David L., Kevin B. Korb, and Jonathan J. Oliver (1996). *Proceedings of the Conference on Information, Statistics and Induction in Science, ISIS'96.* World Scientific. DOI: 10.1142/9789814530637 (Section 3.2).

Edler, Daniel, Ludvig Bohlin, and Martin Rosvall (2017). “Mapping Higher-Order Network Flows in Memory and Multilayer Networks with Infomap”. In: *Algorithms* 10.4, p. 112. DOI: 10.3390/a10040112 (Section 6.4).

Edler, Daniel, Thaís Guedes, et al. (2017). “Infomap Bioregions: Interactive Mapping of Biogeographical Regions from Species Distributions”. In: *Systematic Biology* 66.2, pp. 197–204. DOI: 10.1093/sysbio/syw087 (Section 6.4).

Emmons, Scott and Peter J. Mucha (2019). “Map equation with metadata: Varying the role of attributes in community detection”. In: *Physical Review E* 100.2, p. 022301. DOI: 10.1103/PhysRevE.100.022301 (Section 6.4).

Evans, Scott, Antonis Kourtidis, et al. (2007). “MicroRNA Target Detection and Analysis for Genes Related to Breast Cancer Using MDLcompress”. In: *EURASIP Journal on Bioinformatics and Systems Biology* 2007.1, p. 43670. DOI: 10.1186/1687-4153-2007-43670 (Section 7.4).

Evans, Scott, T. Stephen Markham, et al. (2006). “An Improved Minimum Description Length Learning Algorithm for Nucleotide Sequence Analysis”. In: *Proceedings of the 2006 Fortieth Asilomar Conference on Signals, Systems and Computers, ACSSC'06*, pp. 1843–1850. DOI: 10.1109/ACSSC.2006.355081 (Section 7.4).

Evans, Scott, Gary Saulnier, and Stephen F Bush (2003). “A New Universal Two Part Code for Estimation of String Kolmogorov Complexity and Algorithmic Minimum Sufficient Statistic”. In: *Proceedings of the DIMACS Workshop on Complexity and Inference* (Section 7.4).

Faas, Micky and Matthijs van Leeuwen (2020). “Vouw: Geometric Pattern Mining Using the MDL Principle”. In: *Proceedings of the 18th International Symposium on Advances in Intelligent Data Analysis, IDA'20.* Springer, pp. 158–170. DOI: 10.1007/978-3-030-44584-3_13 (Section 5.2).

Faloutsos, Christos and Vasileios Megalooikonomou (2007). “On data mining, compression, and Kolmogorov complexity”. In: *Data Mining and Knowledge Discovery* 15.1, pp. 3–20. DOI: 10.1007/s10618-006-0057-3 (Section 3.4).

Fayyad, Usama M. and Keki B. Irani (1993). “Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning”. In: *Machine Learning* (Section 3.5).

Feng, Jing (2015). “Information-theoretic graph mining”. PhD thesis. Ludwig-Maximilians-Universität München (Section 6).

Feng, Jing, Xiao He, Nina Hubig, et al. (2013). “Compression-Based Graph Mining Exploiting Structure Primitives”. In: *Proceedings of the 13th IEEE International Conference on Data Mining, ICDM'13.* IEEE Computer Society, pp. 181–190. DOI: 10.1109/ICDM.2013.56 (Section 6.5).

Feng, Jing, Xiao He, Bettina Konte, et al. (2012). “Summarization-based mining bipartite graphs”. In: *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'12.* ACM, pp. 1249–1257. DOI: 10.1145/2339530.2339725 (Section 6.1).

Fischer, Jonas, Anna Oláh, and Jilles Vreeken (2021). “What's in the Box? Explaining Neural Networks with Robust Rules”. In: *Proceedings of the 38th International Conference on Machine Learning, ICML'21* (Section 4.4).

Fischer, Jonas and Jilles Vreeken (2019). “Sets of Robust Rules, and How to Find Them”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’19*. ACM, pp. 38–54. DOI: 10.1007/978-3-030-46150-8_3 (Section 4.4).

— (2020). “Discovering Succinct Pattern Sets Expressing Co-Occurrence and Mutual Exclusivity”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’19*. ACM (Section 5.1).

Fowkes, Jaroslav and Charles Sutton (2016). “A Subsequence Interleaving Model for Sequential Pattern Mining”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’16*. ACM, pp. 835–844 (Section 7.5).

Fürnkranz, Johannes, Tomáš Kliegr, and Heiko Paulheim (2020). “On cognitive preferences and the plausibility of rule-based models”. In: *Machine Learning* 109.4, pp. 853–898. DOI: 10.1007/s10994-019-05856-5 (Section 3.3).

Galbrun, Esther et al. (2018). “Mining Periodic Patterns with a MDL Criterion”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’18*, pp. 535–551 (Section 7.7).

Gallo, Arianna, Tijl De Bie, and Nello Cristianini (2007). “MINI: Mining Informative Non-redundant Itemsets”. In: *Proceedings of the European Conference on Knowledge Discovery in Databases, PKDD’07*. Springer, pp. 438–445. DOI: 10.1007/978-3-540-74976-9_44 (Section 3.7).

Gautrais, Clément et al. (2020). “Widening for MDL-Based Retail Signature Discovery”. In: *Proceedings of the 18th International Symposium on Advances in Intelligent Data Analysis, IDA’20*. Springer, pp. 197–209. DOI: 10.1007/978-3-030-44584-3_16 (Section 7.2).

Geng, Liqiang and Howard J. Hamilton (2006). “Interestingness measures for data mining: A survey”. In: *ACM Computing Surveys* 38.3. DOI: 10.1145/1132960.1132963 (Section 3.7).

Gionis, Aristides et al. (2007). “Assessing data mining results via swap randomization”. In: *ACM Transactions on Knowledge Discovery from Data* 1.3, 14–es. DOI: 10.1145/1297332.1297338 (Section 3.7).

Goebl, Sebastian et al. (2016). “MeGS: Partitioning Meaningful Subgraph Structures Using Minimum Description Length”. In: *Proceedings of the 16th IEEE International Conference on Data Mining, ICDM’16*. IEEE Computer Society, pp. 889–894. DOI: 10.1109/ICDM.2016.0108 (Section 6.5).

Greenspan, Gideon and Dan Geiger (2003). “Model-based inference of haplotype block variation”. In: *Proceedings of the seventh annual international conference on Research in computational molecular biology, RECOMB’03*. ACM, pp. 131–137. DOI: 10.1145/640075.640092 (Section 7.1).

— (2004). “Model-Based Inference of Haplotype Block Variation”. In: *Journal of Computational Biology* 11.2, pp. 493–504. DOI: 10.1089/1066527041410300 (Section 7.1).

Grosse, Kathrin and Jilles Vreeken (2017). “Summarising Event Sequences using Serial Episodes and an Ontology”. In: *Proceedings of the Workshop on Interactions between Data Mining and Natural Language Processing @ECML/PKDD’17* (Section 7.5).

Grünwald, Peter D. (2004). *A Tutorial Introduction to the Minimum Description Length Principle*. arXiv: math/0406077 (Section 3.1).

— (2007). *The Minimum Description Length Principle*. MIT Press (Section 3.1).

Grünwald, Peter D., Jay Injae Myung, and Mark A. Pitt (2005). *Advances in Minimum Description Length: Theory and applications*. Neural Information Processing. The MIT Press, p. 372 (Section 3.1).

Grünwald, Peter D. and Teemu Roos (2019). “Minimum description length revisited”. In: *International Journal of Mathematics for Industry*, p. 1930001. DOI: 10.1142/S2661335219300018 (Section 3.1).

Guns, Tias, Siegfried Nijssen, and Luc De Raedt (2011). “Itemset mining: A constraint programming perspective”. In: *Artificial Intelligence* 175.12, pp. 1951–1983 (Section 3.7).

— (2013). “k-Pattern Set Mining under Constraints”. In: *IEEE Transactions on Knowledge and Data Engineering* 25.2, pp. 402–418. DOI: 10.1109/TKDE.2011.204 (Section 3.7).

Hämäläinen, Wilhelmiina and Geoffrey I. Webb (2018). “A tutorial on statistically sound pattern discovery”. In: *Data Mining and Knowledge Discovery*. DOI: 10.1007/s10618-018-0590-x (Section 3.7).

Hanhijärvi, Sami et al. (2009). “Tell Me Something I Don’t Know: Randomization Strategies for Iterative Data Mining”. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’09*. ACM, pp. 379–388. DOI: 10.1145/1557019.1557065 (Section 3.7).

Hansen, Mark H. and Bin Yu (2001). “Model Selection and the Principle of Minimum Description Length”. In: *Journal of the American Statistical Association* 96.454, pp. 746–774. DOI: 10.1198/016214501753168398 (Section 3.1).

He, Jingrui et al. (2009). “PaCK: Scalable parameter-free clustering on K-partite graphs”. In: *Proceedings of the 2006 SIAM International Conference on Data Mining, SDM’09*. SIAM, pp. 1278–1287 (Section 6.1).

He, Xiao, Jing Feng, Bettina Konte, et al. (2014). “Relevant overlapping subspace clusters on categorical data”. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’14*. ACM, pp. 213–222. DOI: 10.1145/2623330.2623652 (Section 5.5).

He, Xiao, Jing Feng, and Claudia Plant (2011). "Automatically Spotting Information-Rich Nodes in Graphs". In: *Proceedings of the 11th IEEE International Conference on Data Mining Workshops, ICDMW'11*. IEEE Computer Society, pp. 941–948. DOI: 10.1109/ICDMW.2011.37 (Section 6.1).

Heierman, Edwin O. and Diane J. Cook (2003). "Improving home automation by discovering regularly occurring device usage patterns". In: *Proceedings of the 3rd IEEE International Conference on Data Mining, ICDM'03*. IEEE Computer Society, pp. 537–540. DOI: 10.1109/ICDM.2003.1250971 (Section 7.7).

Heierman, Edwin O., G. Michael Youngblood, and Diane J. Cook (2004). "Mining temporal sequences to discover interesting patterns". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'19*. ACM (Section 7.7).

Heikinheimo, Hannes et al. (2009). "Low-Entropy Set Selection". In: *Proceedings of the 2009 SIAM International Conference on Data Mining, SDM'09*. SIAM, pp. 569–580. DOI: 10.1137/1.9781611972795.49 (Section 5.1).

Hess, Sibylle, Katharina Morik, and Nico Piatkowski (2017). "The PRIMPING routine – Tiling through proximal alternating linearized minimization". In: *Data Mining and Knowledge Discovery* 31.4, pp. 1090–1131. DOI: 10.1007/s10618-017-0508-z (Section 5.4).

Hess, Sibylle, Nico Piatkowski, and Katharina Morik (2014). "SHrimp: Descriptive Patterns in a Tree". In: *Proceedings of the LWA (Lernen, Wissen, Adaption) 2014 Workshops: KDML, IR, FGWM* (Section 4.2).

Hinrichs, Frauke and Jilles Vreeken (2017). "Characterising the Difference and the Norm between Sequence Databases". In: *Proceedings of the Workshop on Interactions between Data Mining and Natural Language Processing @ECML/PKDD'17* (Section 7.5).

Hlaváčková-Schindler, Kateřina and Claudia Plant (2020). "Poisson Graphical Granger Causality by Minimum Message Length". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'20* (Section 9.1).

Hu, Bing et al. (2011). "Discovering the Intrinsic Cardinality and Dimensionality of Time Series Using MDL". In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM'11*. IEEE Computer Society, pp. 1086–1091. DOI: 10.1109/ICDM.2011.54 (Section 7.3).

— (2013). "Towards Discovering the Intrinsic Cardinality and Dimensionality of Time Series Using MDL". In: *Proceedings of the Ray Solomonoff 85th Memorial Conference, Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*. Springer, pp. 184–197. DOI: 10.1007/978-3-642-44958-1_14 (Section 7.3).

— (2015). "Using the minimum description length to discover the intrinsic cardinality and dimensionality of time series". In: *Data Mining and Knowledge Discovery* 29.2, pp. 358–399. DOI: 10.1007/s10618-014-0345-2 (Section 7.3).

Ibrahim, A., Shivakumar Sastry, and P. S. Sastry (2016). "Discovering compressing serial episodes from event sequences". In: *Knowledge and Information Systems* 47.2, pp. 405–432. DOI: 10.1007/s10115-015-0854-3 (Section 7.5).

Janzing, Dominik and Bernhard Schölkopf (2010). "Causal Inference Using the Algorithmic Markov Condition". In: *IEEE Transactions on Information Theory* 56.10, pp. 5168–5194. DOI: 10.1109/TIT.2010.2060095 (Section 9.1).

Jaroszewicz, Szymon and Dan A. Simovici (2002). "Pruning Redundant Association Rules using Maximum Entropy Principle". In: *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, PAKDD'02*. Vol. 2336. Springer, pp. 135–147 (Section 9.2).

— (2004). "Interestingness of frequent itemsets using Bayesian networks as background knowledge". In: *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'04*. ACM, pp. 178–186. DOI: 10.1145/1014052.1014074 (Section 3.7).

Jiang, Meng, Christos Faloutsos, and Jiawei Han (2016). "CatchTartan: Representing and Summarizing Dynamic Multicontextual Behaviors". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'16*. ACM, pp. 945–954. DOI: 10.1145/2939672.2939749 (Section 6.2).

Jonyer, Istvan, Lawrence B. Holder, and Diane J. Cook (2004). "Mdl-based context-free graph grammar induction and applications". In: *International Journal on Artificial Intelligence Tools* 13.1, pp. 65–79. DOI: 10.1142/S0218213004001429 (Section 6.5).

Kaltenpoth, David and Jilles Vreeken (2019). "We Are Not Your Real Parents: Telling Causal from Confounded using MDL". In: *Proceedings of the 2007 SIAM International Conference on Data Mining, SDM'19*. SIAM, pp. 199–207. DOI: 10.1137/1.9781611975673.23 (Section 9.1).

Kameya, Yoshitaka (2011). "Time Series Discretization via MDL-Based Histogram Density Estimation". In: *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence, ICTAI'11*. IEEE Computer Society, pp. 732–739. DOI: 10.1109/ICTAI.2011.115 (Section 5.6).

Kang, U and Christos Faloutsos (2011). "Beyond 'Caveman Communities': Hubs and Spokes for Graph Compression and Mining". In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM'11*. IEEE Computer Society, pp. 300–309. DOI: 10.1109/ICDM.2011.26 (Section 6.3).

Kapoor, Sarang, Dhish Kumar Saxena, and Matthijs van Leeuwen (2020). "Discovering subjectively interesting multigraph patterns". In: *Machine Learning* 109.8, pp. 1669–1696. DOI: 10.1007/s10994-020-05873-9 (Section 9.2).

Kapoor, Sarang, Dhish Kumar Saxena, and Matthijs van Leeuwen (2021). “Online summarization of dynamic graphs using subjective interestingness for sequential data”. In: *Data Mining and Knowledge Discovery* 35.1, pp. 88–126. DOI: 10.1007/s10618-020-00714-8 (Section 9.2).

Keogh, Eamonn, Stefano Lonardi, and Chotirat Ann Ratanamahatana (2004). “Towards parameter-free data mining”. In: *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’04*. ACM, pp. 206–215. DOI: 10.1145/1014052.1014077 (Section 3.4).

Keogh, Eamonn, Stefano Lonardi, Chotirat Ann Ratanamahatana, et al. (2007). “Compression-based data mining of sequential data”. In: *Data Mining and Knowledge Discovery* 14.1, pp. 99–129. DOI: 10.1007/s10618-006-0049-3 (Section 3.4).

Ketkar, Nikhil S., Lawrence B. Holder, and Diane J. Cook (2005). “Subdue: compression-based frequent pattern discovery in graph data”. In: *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations, OSDM’05*. ACM, pp. 71–76. DOI: 10.1145/1133905.1133915 (Section 6.5).

Khan, Kifayat Ullah (2015). “Set-based approach for lossless graph summarization using Locality Sensitive Hashing”. In: *Proceedings of the 31st IEEE International Conference on Data Engineering Workshops, ICDEW’15*. IEEE Computer Society, pp. 255–259. DOI: 10.1109/ICDEW.2015.7129586 (Section 6.1).

Khan, Kifayat Ullah, Waqas Nawaz, and Young-Koo Lee (2014). “Set-Based Unified Approach for Attributed Graph Summarization”. In: *Proceedings of the 4th IEEE International Conference on Big Data and Cloud Computing, BDCloud’14*. IEEE Computer Society, pp. 378–385. DOI: 10.1109/BDCloud.2014.108 (Section 6.1).

— (2015a). “Lossless graph summarization using dense subgraphs discovery”. In: *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication, IMCOM’15*. ACM, pp. 1–7. DOI: 10.1145/2701126.2701157 (Section 6.1).

— (2015b). “Set-based approximate approach for lossless graph summarization”. In: *Computing* 97.12, pp. 1185–1207. DOI: 10.1007/s00607-015-0454-9 (Section 6.1).

Kiernan, Jerry and Evimaria Terzi (2008). “Constructing Comprehensive Summaries of Large Event Sequences”. In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’08*. ACM, pp. 417–425. DOI: 10.1145/1401890.1401943 (Section 7.2).

— (2009a). “Constructing Comprehensive Summaries of Large Event Sequences”. In: *ACM Transactions on Knowledge Discovery from Data* 3.4, 21:1–21:31. DOI: 10.1145/1631162.1631169 (Section 7.2).

— (2009b). “EventSummarizer: A Tool for Summarizing Large Event Sequences”. In: *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, EDBT’09*. ACM, pp. 1136–1139. DOI: 10.1145/1516360.1516497 (Section 7.2).

Kilpeläinen, Pekka, Heikki Mannila, and Esko Ukkonen (1995). “MDL learning of unions of simple pattern languages from positive examples”. In: *Proceedings of the 2nd European Conference on Computational Learning Theory, EuroCOLT’95*. Springer, pp. 252–260. DOI: 10.1007/3-540-59119-2_182 (Section 3.5).

Kit, Chunyu (1998). “A Goodness Measure for Phrase Learning via Compression with the MDL Principle”. In: *Proceedings of the 1998 European Summer School in Logic, Language and Information, ESSLLI’98, student session*, pp. 175–187 (Section 3.6).

Kit, Chunyu and Yorick Wilks (1999). “Unsupervised Learning of Word Boundary with Description Length Gain”. In: *Proceedings of the 1999 Workshop on Computational Natural Language Learning, CoNLL’99, Held in cooperation with EACL’99* (Section 3.6).

Kocaoglu, Murat et al. (2017). “Entropic Causal Inference”. In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI’17*. Association for the Advancement of Artificial Intelligence (Section 9.1).

Koivisto, Mikko et al. (2002). “An MDL method for finding haplotype blocks and for estimating the strength of haplotype block boundaries”. In: *Proceedings of the 2003 Pacific Symposium on Biocomputing, PSB’03*. World Scientific, pp. 502–513. DOI: 10.1142/9789812776303_0047 (Section 7.1).

Kontkanen, Petri and Petri Myllymäki (2007). “MDL Histogram Density Estimation”. In: *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS’07*, pp. 219–226 (Section 5.6).

Kontonasios, Kleanthis-Nikolaos and Tijl De Bie (2010). “An Information-Theoretic Approach to Finding Informative Noisy Tiles in Binary Databases”. In: *Proceedings of the 2010 SIAM International Conference on Data Mining, SDM’10*. SIAM, pp. 153–164 (Section 9.2).

Kontonasios, Kleanthis-Nikolaos and Tijl De Bie (2012). “Formalizing Complex Prior Information to Quantify Subjective Interestingness of Frequent Pattern Sets”. In: *Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis, IDA’12*. Springer, pp. 161–171 (Section 9.2).

Kontonasios, Kleanthis-Nikolaos, Jilles Vreeken, and Tijl De Bie (2013). “Maximum Entropy Models for Iteratively Identifying Subjectively Interesting Structure in Real-Valued Data”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’13*. Springer, pp. 256–271 (Section 9.2).

Koopman, Arne and Arno Siebes (2008). “Discovering Relational Item Sets Efficiently”. In: *Proceedings of the 2008 SIAM International Conference on Data Mining, SDM’08*. SIAM, pp. 108–119. DOI: 10.1137/1.9781611972788.10 (Section 4.5).

Koopman, Arne and Arno Siebes (2009). “Characteristic Relational Patterns”. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’09*. ACM, pp. 437–446. DOI: 10.1145/1557019.1557071 (Section 4.5).

Koutra, Danai et al. (2014). “VOG: Summarizing and Understanding Large Graphs”. In: *Proceedings of the 2014 SIAM International Conference on Data Mining, SDM’14*. SIAM, pp. 91–99. DOI: 10.1137/1.9781611973440.11 (Section 6.5).

— (2015). “Summarizing and understanding large graphs”. In: *Statistical Analysis and Data Mining* 8.3, pp. 183–202 (Section 6.5).

Lakshmanan, Laks V. S. et al. (2002). “The generalized MDL approach for summarization”. In: *Proceedings of the 28th international conference on Very Large Data Bases, VLDB’02*. VLDB Endowment, pp. 766–777 (Section 5.6).

Lam, Hoang Thanh, Toon Calders, et al. (2013). “Zips: Mining Compressing Sequential Patterns in Streams”. In: *Proceedings of the Workshop on Interactive Data Exploration and Analytics, IDEA @KDD’13*. ACM, pp. 54–62. DOI: 10.1145/2501511.2501520 (Section 7.5).

Lam, Hoang Thanh, Julia Kiseleva, et al. (2014). “Decomposing a sequence into independent subsequences using compression algorithms”. In: *Proceedings of the Workshop on Interactive Data Exploration and Analytic, IDEA @KDD’14*, pp. 67–75 (Section 7.2).

Lam, Hoang Thanh, Fabian Mörchen, et al. (2012). “Mining Compressing Sequential Patterns.” In: *Proceedings of the 2012 SIAM International Conference on Data Mining, SDM’12*. SIAM, pp. 319–330. DOI: 10.1137/1.9781611972825.28 (Section 7.5).

— (2014). “Mining Compressing Sequential Patterns”. In: *Statistical Analysis and Data Mining* 7.1, pp. 34–52. DOI: 10.1002/sam.11192 (Section 7.5).

Lanterman, Aaron D. (2001). “Schwarz, Wallace, and Rissanen: Intertwining Themes in Theories of Model Selection”. In: *International Statistical Review* 69.2, pp. 185–212. DOI: 10.1111/j.1751-5823.2001.tb00456.x (Section 3.2).

Lattimore, Tor and Marcus Hutter (2013). “No Free Lunch versus Occam’s Razor in Supervised Learning”. In: *Proceedings of the Ray Solomonoff 85th Memorial Conference, Algorithmic Probability and Friends. Bayesian Prediction and Artificial Intelligence*. Springer, pp. 223–235. DOI: 10.1007/978-3-642-44958-1_17 (Section 3.3).

Lee, Kyuhan et al. (2020). “SSumM: Sparse Summarization of Massive Graphs”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’20*. ACM, pp. 144–154. DOI: 10.1145/3394486.3403057 (Section 6.1).

Lee, Thomas C. M. (2001). “An Introduction to Coding Theory and the Two-Part Minimum Description Length Principle”. In: *International Statistical Review* 69.2, pp. 169–183. DOI: 10.1111/j.1751-5823.2001.tb00455.x (Section 3.1).

LeFevre, Kristen and Evimaria Terzi (2010). “GraSS: Graph Structure Summarization”. In: *Proceedings of the 2010 SIAM International Conference on Data Mining, SDM’10*. SIAM, pp. 454–465. DOI: 10.1137/1.9781611972801.40 (Section 6.1).

Li, Hang and Naoki Abe (1997). “Clustering Words with the MDL Principle”. In: *Journal of Natural Language Processing* 4.2, pp. 71–88. DOI: 10.5715/jnlp.4.2_71 (Section 3.6).

— (1998a). “Generalizing case frames using a thesaurus and the MDL principle”. In: *Computational Linguistics* 24.2, pp. 217–244 (Section 3.6).

— (1998b). “Word clustering and disambiguation based on co-occurrence data”. In: *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, ACL’98/COLING’98*. Association for Computational Linguistics, pp. 749–755. DOI: 10.3115/980691.980693 (Section 3.6).

Li, Ming and Paul Vitányi (1995). “Computational machine learning in theory and praxis”. In: *Computer Science Today: Recent Trends and Developments*. Ed. by Jan van Leeuwen. Springer, pp. 518–535. DOI: 10.1007/BFb0015264 (Section 3.1).

— (2019). *An Introduction to Kolmogorov Complexity and Its Applications*. 4th ed. Springer. DOI: 10.1007/978-3-030-11298-1 (Section 3).

Lim, Yongsub, U Kang, and Christos Faloutsos (2014). “SlashBurn: Graph Compression and Mining beyond Caveman Communities”. In: *IEEE Transactions on Knowledge and Data Engineering* 26.12, pp. 3077–3089. DOI: 10.1109/TKDE.2014.2320716 (Section 6.3).

Liu, Yike, Tara Safavi, Abhilash Dighe, et al. (2018). “Graph Summarization Methods and Applications: A Survey”. In: *ACM Computing Surveys* 51.3, 62:1–62:34. DOI: 10.1145/3186727 (Section 6).

Liu, Yike, Tara Safavi, and Neil Shah (2016). “Reducing Million-Node Graphs to a Few Structural Patterns: A Unified Approach”. In: *Proceedings of the 12th International Workshop on Mining and Learning with Graphs, MLG @KDD’16*, p. 8 (Section 6.5).

Liu, Yike, Tara Safavi, Neil Shah, and Danai Koutra (2018). “Reducing large graphs to small supergraphs: a unified approach”. In: *Social Network Analysis and Mining* 8.1, p. 17. DOI: 10.1007/s13278-018-0491-4 (Section 6.5).

Liu, Yike, Neil Shah, and Danai Koutra (2015). *An Empirical Comparison of the Summarization Power of Graph Clustering Methods.* arXiv: 1511.06820 (Section 6.5).

Lucchese, Claudio, Salvatore Orlando, and Raffaele Perego (2010a). "A generative pattern model for mining binary datasets". In: *Proceedings of the 2010 ACM Symposium on Applied Computing, SAC'10.* ACM, pp. 1109–1110. DOI: 10.1145/1774088.1774320 (Section 5.4).

— (2010b). "Mining Top-K Patterns from Binary Datasets in presence of Noise". In: *Proceedings of the 2007 SIAM International Conference on Data Mining, SDM'07.* SIAM, pp. 165–176. DOI: 10.1137/1.9781611972801.15 (Section 5.4).

— (2014). "A Unifying Framework for Mining Approximate Top-k Binary Patterns". In: *IEEE Transactions on Knowledge and Data Engineering* 26.12, pp. 2900–2913. DOI: 10.1109/TKDE.2013.181 (Section 5.4).

Makhalova, Tatiana (2021). "Contributions to pattern set mining : from complex datasets to significant and useful pattern sets". PhD thesis. Université de Lorraine. URL: https://hal.univ-lorraine.fr/tel-03342124 (Section 5.3).

Makhalova, Tatiana, Sergei O. Kuznetsov, and Amedeo Napoli (2018a). "A First Study on What MDL Can Do for FCA". In: *Proceedings of the Fifteen International Conference on Concept Lattices and Their Applications, CLA'18*, pp. 25–36 (Section 5.3).

— (2018b). "MDL for FCA: Is There a Place for Background Knowledge?" In: *Proceedings of the 6th International Workshop "What can FCA do for Artificial Intelligence?" @ IJCAI/ECAI'18.* Vol. 2149. CEUR Workshop Proceedings, pp. 45–56. URL: http://ceur-ws.org/Vol-2149/paper5.pdf (Section 5.3).

— (2019a). "Numerical Pattern Mining Through Compression". In: *Proceedings of the Data Compression Conference, DCC'19*, pp. 112–121. DOI: 10.1109/DCC.2019.00019 (Section 5.6).

— (2019b). "On Coupling FCA and MDL in Pattern Mining". In: *Proceedings of the international conference on Formal Concept Analysis, FCA'19.* Springer, pp. 332–340. DOI: 10.1007/978-3-030-21462-3_23 (Section 5.3).

— (2020). *Mint: MDL-based approach for Mining INTeresting Numerical Pattern Sets.* arXiv: 2011.14843 (Section 5.6).

— (2021). "Likely-Occurring Itemsets for Pattern Mining". In: *Proceedings of the 6th International Workshop "What can FCA do for Artificial Intelligence?" @ IJCAI'21.* Vol. 2972. CEUR Workshop Proceedings, pp. 39–50. URL: http://ceur-ws.org/Vol-2972/paper4.pdf (Section 5.3).

— (2022). "Mint: MDL-based approach for Mining INTeresting Numerical Pattern Sets". In: *Data Mining and Knowledge Discovery* 36.1, pp. 108–145. DOI: 10.1007/s10618-021-00799-9 (Section 5.6).

Makhalova, Tatiana and Martin Trnecka (2019). *From-Below Boolean Matrix Factorization Algorithm Based on MDL.* arXiv: 1901.09567 (Section 5.4).

— (2021). "From-below Boolean matrix factorization algorithm based on MDL". In: *Advances in Data Analysis and Classification* 15.1, pp. 37–56. DOI: 10.1007/s11634-019-00383-6 (Section 5.4).

Mampaey, Michael (2010). "Mining Non-redundant Information-Theoretic Dependencies between Itemsets". In: *Proceedings of the 12th International Conference on Data Warehousing and Knowledge Discovery, DaWaK'10.* Springer, pp. 130–141. DOI: 10.1007/978-3-642-15105-7_11 (Section 3.7).

Mampaey, Michael, Nikolaj Tatti, and Jilles Vreeken (2011). "Tell me what I need to know: succinctly summarizing data with itemsets". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'11.* ACM, pp. 573–581. DOI: 10.1145/2020408.2020499 (Section 9.2).

Mampaey, Michael and Jilles Vreeken (2010). "Summarising Data by Clustering Items". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'10*, pp. 321–336. DOI: 10.1007/978-3-642-15883-4_21 (Section 5.1).

— (2013). "Summarizing categorical data by clustering attributes". In: *Data Mining and Knowledge Discovery* 26.1, pp. 130–173. DOI: 10.1007/s10618-011-0246-6 (Section 5.5).

Mampaey, Michael, Jilles Vreeken, and Nikolaj Tatti (2012). "Summarizing Data Succinctly with the Most Informative Itemsets". In: *ACM Transactions on Knowledge Discovery from Data* 6.4, 16:1–16:42. DOI: 10.1145/2382577.2382580 (Section 5.1).

Mandros, Panagiotis, Mario Boley, and Jilles Vreeken (2019). "Discovering Reliable Correlations in Categorical Data". In: *Proceedings of the 19th IEEE International Conference on Data Mining, ICDM'19.* IEEE Computer Society, pp. 1252–1257. DOI: 10.1109/ICDM.2019.00156 (Section 9.1).

— (2020). "Discovering dependencies with reliable mutual information". In: *Knowledge and Information Systems* 62.11, pp. 4223–4253. DOI: 10.1007/s10115-020-01494-9 (Section 9.1).

Mannila, Heikki, M. Koivisto, et al. (2003). "Minimum Description Length Block Finder, a Method to Identify Haplotype Blocks and to Compare the Strength of Block Boundaries". In: *The American Journal of Human Genetics* 73.1, pp. 86–94. DOI: 10.1086/376438 (Section 7.1).

Mannila, Heikki, Hannu Toivonen, and A Inkeri Verkamo (1994). "Efficient Algorithms for Discovering Association Rules". In: *Proceedings of the KDD Workshop.* Association for the Advancement of Artificial Intelligence, pp. 181–192 (Section 3.7).

Markham, T. Stephen et al. (2009). "Implementation of an Incremental MDL-Based Two Part Compression Algorithm for Model Inference". In: *Proceedings of the 2009 Data Compression Conference, DCC'09*, pp. 322–331. DOI: 10.1109/DCC.2009.66 (Section 7.4).

Marx, Alexander and Jilles Vreeken (2019a). "Causal Inference on Multivariate and Mixed-Type Data". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'19*, pp. 655–671 (Section 9.1).

— (2019c). "Telling cause from effect by local and global regression". In: *Knowledge and Information Systems* 60.3, pp. 1277–1305. DOI: 10.1007/s10115-018-1286-7 (Section 9.1).

— (2022). "Formally Justifying MDL-based Inference of Cause and Effect". In: *Proceedings of the AAAI Workshop on Information Theoretic Causal Inference and Discovery, ITCI'22* (Section 9.1).

Matsubara, Yasuko, Yasushi Sakurai, and Christos Faloutsos (2014). "AutoPlait: automatic mining of co-evolving time sequences". In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, SIGMOD'14*. ACM, pp. 193–204. ISBN: 978-1-4503-2376-5. DOI: 10.1145/2588555.2588556 (Section 7.3).

Mehta, Manish, Jorma Rissanen, and Rakesh Agrawal (1995). "MDL-based decision tree pruning". In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining, KDD'95*. Association for the Advancement of Artificial Intelligence, pp. 216–221 (Section 3.5).

Meo, Rosa (2000). "Theory of dependence values". In: *ACM Transactions on Database Systems* 25.3, pp. 380–406. DOI: 10.1145/363951.363956 (Section 9.2).

Merhav, N. (1993). "On the minimum description length principle for sources with piecewise constant parameters". In: *IEEE Transactions on Information Theory* 39.6, pp. 1962–1967. DOI: 10.1109/18.265504 (Section 3.5).

Mian, Osman, Alexander Marx, and Jilles Vreeken (2021). "Discovering Fully Oriented Causal Networks". In: *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI'21*. Association for the Advancement of Artificial Intelligence (Section 9.1).

Miettinen, Pauli and Jilles Vreeken (2011). "Model order selection for boolean matrix factorization". In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'11*. ACM, pp. 51–59. DOI: 10.1145/2020408.2020424 (Section 5.4).

— (2014). "MDL4BMF: Minimum Description Length for Boolean Matrix Factorization". In: *ACM Transactions on Knowledge Discovery from Data* 8.4, 18:1–18:31. DOI: 10.1145/2601437 (Section 5.4).

Mitra, Soumyajit and P. S. Sastry (2019). *Summarizing Event Sequences with Serial Episodes: A Statistical Model and an Application*. arXiv: 1904.00516 (Section 7.5).

Muggleton, Stephen, Ashwin Srinivasan, and Michael Bain (1992). "Compression, Significance and Accuracy". In: *Proceedings of the Ninth International Conference on Machine Learning, ICML'92*. Morgan Kaufmann, pp. 338–347. DOI: 10.1016/B978-1-55860-247-2.50048-6 (Section 3.5).

Navlakha, Saket, Rajeev Rastogi, and Nisheeth Shrivastava (2008). "Graph Summarization with Bounded Error". In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, SIGMOD'08*. ACM, pp. 419–432. DOI: 10.1145/1376616.1376661 (Section 6.1).

Nguyen, Hoang-Vu, Panagiotis Mandros, and Jilles Vreeken (2016). "Universal Dependency Analysis". In: *Proceedings of the 2016 SIAM International Conference on Data Mining, SDM'16*. SIAM, pp. 792–800. DOI: 10.1137/1.9781611974348.89 (Section 9.1).

Nguyen, Hoang-Vu, Emmanuel Müller, et al. (2014). "Unsupervised interaction-preserving discretization of multivariate data". In: *Data Mining and Knowledge Discovery* 28.5, pp. 1366–1397. DOI: 10.1007/s10618-014-0350-5 (Section 5.6).

Otaki, Keisuke and Akihiro Yamamoto (2015). "Edit Operations on Lattices for MDL-based Pattern Summarization". In: *Proceedings of the International Workshop on Formal Concept Analysis and Applications @ICFCA'15* (Section 5.3).

Papadimitriou, Spiros, Aristides Gionis, et al. (2005). "Parameter-Free Spatial Data Mining Using MDL". In: *Proceedings of the 5th IEEE International Conference on Data Mining, ICDM'05*. IEEE Computer Society, pp. 346–353. DOI: 10.1109/ICDM.2005.117 (Section 5.2).

Papadimitriou, Spiros, Jimeng Sun, et al. (2008). "Hierarchical, Parameter-Free Community Discovery". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'08*. Springer, pp. 170–187. DOI: 10.1007/978-3-540-87481-2_12 (Section 6.1).

Pavlov, Dmitry, Heikki Mannila, and Padhraic Smyth (2003). "Beyond independence: Probabilistic models for query approximation on binary transaction data". In: *IEEE Transactions on Knowledge and Data Engineering* 15.6, pp. 1409–1421 (Section 3.7).

Pearl, Judea (2009). *Causality*. Cambridge university press (Section 9.1).

Pednault, Edwin P. D. (1989). "Some experiments in applying inductive inference principles to surface reconstruction". In: *Proceedings of the 11th international joint conference on Artificial intelligence, IJCAI'89*. Morgan Kaufmann, pp. 1603–1609 (Section 3.5).

Pennerath, Frédéric, Panagiotis Mandros, and Jilles Vreeken (2020). "Discovering Approximate Functional Dependencies using Smoothed Mutual Information". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'20*. ACM (Section 9.1).

Pfahringer, Bernhard (1995a). “A new MDL measure for robust rule induction”. In: *Proceedings of the 8th European Conference on Machine Learning, ECML’95*. Springer, pp. 331–334. DOI: 10.1007/3-540-59286-5_80 (Section 3.5).

— (1995b). “Compression based feature subset selection”. In: *Proceedings of the Workshop on Data Engineering for Inductive Learning @IJCAI’95* (Section 3.5).

Phan, Nhat Hai et al. (2013). “Mining Representative Movement Patterns through Compression”. In: *Advances in Knowledge Discovery and Data Mining*. Springer, pp. 314–326. DOI: 10.1007/978-3-642-37453-1_26 (Section 7.8).

Plant, Claudia, Sonja Biedermann, and Christian Böhm (2020). “Data Compression as a Comprehensive Framework for Graph Drawing and Representation Learning”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’20*. ACM, pp. 1212–1222. DOI: 10.1145/3394486.3403174 (Section 6.1).

Prakash, B. Aditya, Jilles Vreeken, and Christos Faloutsos (2014). “Efficiently spotting the starting points of an epidemic in a large graph”. In: *Knowledge and Information Systems* 38.1, pp. 35–59. DOI: 10.1007/s10115-013-0671-5 (Section 6.7).

Proença, Hugo M., Thomas Bäck, and Matthijs van Leeuwen (2021). *Robust subgroup discovery*. arXiv: 2103.13686 (Section 4.4).

Proença, Hugo M., Peter D. Grünwald, et al. (2020). “Discovering outstanding subgroup lists for numeric targets using MDL”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’20* (Section 4.4).

— (2021). *Discovering outstanding subgroup lists for numeric targets using MDL*. arXiv: 2006.09186 (Section 4.4).

Proença, Hugo M. and Matthijs van Leeuwen (2020a). *Interpretable multiclass classification by MDL-based rule lists*. arXiv: 1905.00328 (Section 4.4).

— (2020b). “Interpretable multiclass classification by MDL-based rule lists”. In: *Information Sciences* 512, pp. 1372–1393. DOI: 10.1016/j.ins.2019.10.050 (Section 4.4).

Puolamäki, Kai et al. (2020). “Interactive visual data exploration with subjective feedback: an information-theoretic approach”. In: *Data Mining and Knowledge Discovery* 34.1, pp. 21–49. DOI: 10.1007/s10618-019-00655-x (Section 9.2).

Quinlan, J. Ross (1994). “The Minimum Description Length Principle and Categorical Theories”. In: *Proceedings of the Eleventh International Conference on Machine Learning, ICML’94*. Morgan Kaufmann, pp. 233–241. DOI: 10.1016/B978-1-55860-335-6.50036-2 (Section 3.5).

Quinlan, J. Ross and Ronald L. Rivest (1989). “Inferring decision trees using the minimum description length principle”. In: *Information and Computation* 80.3, pp. 227–248. DOI: 10.1016/0890-5401(89)90010-2 (Section 3.5).

Rakthanmanon, Thanawin et al. (2011). “Time Series Epenthesis: Clustering Time Series Streams Requires Ignoring Some Data”. In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM’11*. IEEE Computer Society, pp. 547–556. DOI: 10.1109/ICDM.2011.146 (Section 7.3).

— (2012). “MDL-based time series clustering”. In: *Knowledge and Information Systems* 33.2, pp. 371–399. DOI: 10.1007/s10115-012-0508-7 (Section 7.3).

Rashidi, Parisa and Diane J. Cook (2013). “COM: A method for mining and monitoring human activity patterns in home-based health monitoring systems”. In: *ACM Transactions on Intelligent Systems and Technology* 4.4, 64:1–64:20. DOI: 10.1145/2508037.2508045 (Section 7.7).

Rathmanner, Samuel and Marcus Hutter (2011). “A Philosophical Treatise of Universal Induction”. In: *Entropy* 13.6, pp. 1076–1136. DOI: 10.3390/e13061076 (Section 3.3).

Rissanen, Jorma (1978). “Modeling by shortest data description”. In: *Automatica* 14.5, pp. 465–471. DOI: 10.1016/0005-1098(78)90005-5 (Section 3.1).

— (1983). “A Universal Prior for Integers and Estimation by Minimum Description Length”. In: *The Annals of Statistics* 11.2, pp. 416–431 (Section 3.1).

— (1989). *Stochastic complexity in statistical inquiry*. World Scientific (Section 3.1).

— (2005). *An Introduction to the MDL Principle*. Technical report. Helsinki Institute for Information Technology (HIIT) (Section 3.1).

— (2007). *Information and Complexity in Statistical Modeling*. Information Science and Statistics. Springer, pp. 97–102. DOI: 10.1007/978-0-387-68812-1 (Section 3.1).

Rissanen, Jorma and Mati Wax (1987). “Measures of mutual and causal dependence between two time series”. In: *IEEE Transactions on Information Theory* 33.4, pp. 598–601. DOI: 10.1109/TIT.1987.1057325 (Section 9.1).

Robinet, Vivien, Benoît Lemaire, and Mirta B. Gordon (2011). “MDLChunker: A MDL-Based Cognitive Model of Inductive Learning”. In: *Cognitive Science* 35.7, pp. 1352–1389. DOI: 10.1111/j.1551-6709.2011.01188.x (Section 3.3).

Robnik-Šikonja, Marko and Igor Kononenko (1998). “Pruning regression trees with MDL”. In: *Proceedings of the 13th European conference on artificial intelligence, ECAI’98* (Section 3.5).

Rojas, Alexis et al. (2021). "A multiscale view of the Phanerozoic fossil record reveals the three major biotic transitions". In: *Communications Biology* 4.1, pp. 1–8. DOI: 10.1038/s42003-021-01805-y (Section 6.4).

Rooij, Steven de and Peter D. Grünwald (2011). "Luckiness and Regret in Minimum Description Length Inference". In: *Philosophy of Statistics*. Ed. by Prasanta S. Bandyopadhyay and Malcolm R. Forster. Vol. 7. Handbook of the Philosophy of Science. North-Holland, pp. 865–900. DOI: 10.1016/B978-0-444-51862-0.50029-0 (Section 3.1).

Roos, Teemu (2016). "Minimum Description Length Principle". In: *Encyclopedia of Machine Learning and Data Mining*. Ed. by Claude Sammut and Geoffrey I. Webb. Springer, pp. 1–4. DOI: 10.1007/978-1-4899-7502-7_894-1 (Section 3.1).

Rosvall, Martin, D. Axelsson, and Carl T. Bergstrom (2009). "The map equation". In: *The European Physical Journal Special Topics* 178.1, pp. 13–23. DOI: 10.1140/epjst/e2010-01179-1 (Section 6.4).

Rosvall, Martin and Carl T. Bergstrom (2007). "An information-theoretic framework for resolving community structure in complex networks". In: *Proceedings of the National Academy of Sciences* 104.18, pp. 7327–7331. DOI: 10.1073/pnas.0611034104 (Section 6.1).

— (2008). "Maps of random walks on complex networks reveal community structure". In: *Proceedings of the National Academy of Sciences* 105.4, pp. 1118–1123. DOI: 10.1073/pnas.0706851105 (Section 6.4).

— (2010). "Mapping Change in Large Networks". In: *PLoS ONE* 5.1, pp. 1–7. DOI: 10.1371/journal.pone.0008694 (Section 6.4).

— (2011). "Multilevel Compression of Random Walks on Networks Reveals Hierarchical Organization in Large Integrated Systems". In: *PLOS ONE* 6.4, e18209. DOI: 10.1371/journal.pone.0018209 (Section 6.4).

Sampson, Oliver and Michael R. Berthold (2014). "Widened KRIMP: Better Performance through Diverse Parallelism". In: *Proceedings of the 13th International Symposium on Advances in Intelligent Data Analysis, IDA'14*. Springer, pp. 276–285. DOI: 10.1007/978-3-319-12571-8_24 (Section 4.2).

Saran, Divyam and Jilles Vreeken (2019). *Summarizing Dynamic Graphs using MDL*. Technical report. Saarland University (Section 6.6).

Segen, Jakub (1989). "Incremental Clustering by Minimizing Representation Length". In: *Proceedings of the Sixth International Workshop on Machine Learning*. Morgan Kaufmann, pp. 400–403. DOI: 10.1016/B978-1-55860-036-2.50101-6 (Section 3.2).

— (1990). "Graph clustering and model learning by data compression". In: *Proceedings of the Seventh International Conference on Machine Learning, ICML'90*. Morgan Kaufmann, pp. 93–101 (Section 3.2).

Segen, Jakub and A. C. Sanderson (1979). "A minimal representation criterion for clustering". In: *Proceedings of the 12th Annual Symposium on the Interface: Computer Science and Statistics*, pp. 332–334 (Section 3.2).

Shah, Neil, Danai Koutra, Lisa Jin, et al. (2017). "On Summarizing Large-Scale Dynamic Graphs". In: *IEEE Data Engineering Bulletin* 40.3, pp. 75–88 (Section 6.6).

Shah, Neil, Danai Koutra, Tianmin Zou, et al. (2015). "TimeCrunch: Interpretable Dynamic Graph Summarization". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'15*. ACM, pp. 1055–1064. DOI: 10.1145/2783258.2783321 (Section 6.6).

Shamir, Gill I., Daniel J. Costello, and N. Merhav (1999). "Asymptotically optimal low complexity sequential lossless coding for regular piecewise stationary memoryless sources". In: *Proceedings of the 1999 IEEE Information Theory and Communications Workshop*. IEEE Computer Society, pp. 72–74. DOI: 10.1109/ITCOM.1999.781413 (Section 3.5).

Shannon, Claude E. (1948). "A Mathematical Theory of Communication". In: *Bell System Technical Journal* 27.3, pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x (Section 3).

Shokoohi-Yekta, Mohammad et al. (2015). "Discovery of Meaningful Rules in Time Series". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'15*. ACM, pp. 1085–1094. DOI: 10.1145/2783258.2783306 (Section 7.6).

Siebes, Arno (2012). "Queries for Data Analysis". In: *Proceedings of the 11th International Symposium on Advances in Intelligent Data Analysis, IDA'12*. Springer, pp. 7–22 (Section 4.1).

— (2014). "MDL in Pattern Mining: A Brief Introduction to Krimp". In: *Proceedings of the international conference on Formal Concept Analysis, FCA'14*. Springer, pp. 37–43. DOI: 10.1007/978-3-319-07248-7_3 (Section 4.1).

Siebes, Arno and René Kersten (2011). "A Structure Function for Transaction Data". In: *Proceedings of the 2011 SIAM International Conference on Data Mining, SDM'11*. SIAM, pp. 558–569. DOI: 10.1137/1.9781611972818.48 (Section 4.5).

— (2012). "Smoothing Categorical Data". In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD'12*. Springer, pp. 42–57. DOI: 10.1007/978-3-642-33460-3_8 (Section 4.5).

Siebes, Arno, Jilles Vreeken, and Matthijs van Leeuwen (2006). "Item Sets that Compress". In: *Proceedings of the 2006 SIAM International Conference on Data Mining, SDM'06*. SIAM (Section 4.1).

Silverstein, Craig, Sergey Brin, and Rajeev Motwani (1998). "Beyond Market Baskets: Generalizing Association Rules to Dependence Rules". In: *Data Mining and Knowledge Discovery* 2.1, pp. 39–68. DOI: 10.1023/A:1009713703947 (Section 3.7).

Simovici, Dan A. (2013). “Minability through Compression”. In: *Proceedings of the 15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC’13*, pp. 32–36. DOI: 10.1109/SYNASC.2013.11 (Section 3.4).

Simovici, Dan A. et al. (2015). “Compression and data mining”. In: *Proceedings of the 2015 International Conference on Computing, Networking and Communications, ICNC’15*, pp. 551–555. DOI: 10.1109/ICCNC.2015.7069404 (Section 3.4).

Slonim, Noam (2002). “The information bottleneck: Theory and applications”. PhD thesis. The Hebrew University (Section 3.2).

Smets, Koen and Jilles Vreeken (2011). “The Odd One Out: Identifying and Characterising Anomalies”. In: *Proceedings of the 2011 SIAM International Conference on Data Mining, SDM’11*. SIAM, pp. 804–815. DOI: 10.1137/1.9781611972818.69 (Section 4.3).

— (2012). “Slim: Directly Mining Descriptive Patterns”. In: *Proceedings of the 2012 SIAM International Conference on Data Mining, SDM’12*. SIAM, pp. 236–247 (Section 4.2).

Solomonoff, R. J. (1964a). “A formal theory of inductive inference. Part I”. In: *Information and Control* 7.1, pp. 1–22. DOI: 10.1016/S0019-9958(64)90223-2 (Section 3.2).

— (1964b). “A formal theory of inductive inference. Part II”. In: *Information and Control* 7.2, pp. 224–254. DOI: 10.1016/S0019-9958(64)90131-7 (Section 3.2).

Soulet, Arnaud et al. (2011). “Mining Dominant Patterns in the Sky”. In: *Proceedings of the 11th IEEE International Conference on Data Mining, ICDM’11*. IEEE Computer Society, pp. 655–664. DOI: 10.1109/ICDM.2011.100 (Section 3.7).

Stone, James V. (2013). *Information Theory: A Tutorial Introduction*. Sebtel Press (Section 3).

— (2018). *Information Theory: A Tutorial Introduction*. arXiv: 1802.05968 (Section 3).

Sun, Jimeng et al. (2007). “GraphScope: parameter-free mining of large time-evolving graphs”. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’07*. ACM, pp. 687–696. DOI: 10.1145/1281192.1281266 (Section 6.2).

Suzuki, Joe (1993). “A Construction of Bayesian Networks from Databases Based on an MDL Principle”. In: *Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence, UAI’93*, pp. 266–273. DOI: 10.1016/B978-1-4832-1451-1.50037-8 (Section 3.5).

Tanaka, Yoshiki, Kazuhisa Iwamoto, and Kuniaki Uehara (2005). “Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle”. In: *Machine Learning* 58.2, pp. 269–300. DOI: 10.1007/s10994-005-5829-2 (Section 7.6).

Tanaka, Yoshiki and Kuniaki Uehara (2003). “Discover motifs in multi-dimensional time-series using the principal component analysis and the MDL principle”. In: *Proceedings of the 3rd international conference on Machine learning and data mining in pattern recognition, MLDM’03*. Springer, pp. 252–265 (Section 7.6).

Tatti, Nikolaj (2008). “Maximum entropy based significance of itemsets”. In: *Knowledge and Information Systems* 17.1, pp. 57–77. DOI: 10.1007/s10115-008-0128-4 (Section 9.2).

— (2010). “Probably the best itemsets”. In: *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD’10*. ACM, pp. 293–302. DOI: 10.1145/1835804.1835843 (Section 3.7).

Tatti, Nikolaj and Hannes Heikinheimo (2008). “Decomposable Families of Itemsets”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’08*, pp. 472–487. DOI: 10.1007/978-3-540-87481-2_31 (Section 3.7).

Tatti, Nikolaj and Michael Mampaey (2010). “Using background knowledge to rank itemsets”. In: *Data Mining and Knowledge Discovery* 21.2, pp. 293–309. DOI: 10.1007/s10618-010-0188-4 (Section 9.2).

Tatti, Nikolaj and Jilles Vreeken (2008). “Finding Good Itemsets by Packing Data”. In: *Proceedings of the 8th IEEE International Conference on Data Mining, ICDM’08*. IEEE Computer Society, pp. 588–597. DOI: 10.1109/ICDM.2008.39 (Section 5.1).

— (2012a). “Discovering Descriptive Tile Trees”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’12*. Springer, pp. 9–24. DOI: 10.1007/978-3-642-33460-3_6 (Section 5.2).

— (2012b). “The Long and the Short of it: Summarising Event Sequences with Serial Episodes”. In: *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’12*. ACM, pp. 462–470 (Section 7.5).

Tishby, Naftali, Fernando C Pereira, and William Bialek (2000). *The Information Bottleneck Method*. arXiv: physics/0004057 (Section 3.2).

van Leeuwen, Matthijs (2010). “Patterns that Matter”. PhD thesis. Universiteit Utrecht (Section 4.1).

van Leeuwen, Matthijs, Francesco Bonchi, et al. (2009). “Compressing tags to find interesting media groups”. In: *Proceedings of the 18th ACM conference on Information and knowledge management, CIKM’09*. ACM, pp. 1147–1156. DOI: 10.1145/1645953.1646099 (Section 4.6).

van Leeuwen, Matthijs, Tijl De Bie, et al. (2016). “Subjective interestingness of subgraph patterns”. In: *Machine Learning* 105.1, pp. 41–75. DOI: 10.1007/s10994-015-5539-3 (Section 9.2).

van Leeuwen, Matthijs and Esther Galbrun (2015). “Association Discovery in Two-View Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 27.12, pp. 3190–3202. DOI: 10.1109/TKDE.2015.2453159 (Section 4.4).

van Leeuwen, Matthijs and Arno Siebes (2008). “StreamKrimp: Detecting Change in Data Streams”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’08*. Springer, pp. 672–687. DOI: 10.1007/978-3-540-87479-9_62 (Section 4.5).

van Leeuwen, Matthijs and Jilles Vreeken (2014). “Mining and Using Sets of Patterns through Compression”. In: *Frequent Pattern Mining*. Springer, pp. 165–198. DOI: 10.1007/978-3-319-07821-2_8 (Section 4.1).

van Leeuwen, Matthijs, Jilles Vreeken, and Arno Siebes (2006). “Compression Picks Item Sets That Matter”. In: *Proceedings of the European Conference on Knowledge Discovery in Databases, PKDD’06*. Springer, pp. 585–592. DOI: 10.1007/11871637_59 (Section 4.1).

— (2009). “Identifying the components”. In: *Data Mining and Knowledge Discovery* 19.2, pp. 176–193. DOI: 10.1007/s10618-009-0137-2 (Section 4.1).

Vanetik, Natalia and Marina Litvak (2017). “Query-based summarization using MDL principle”. In: *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres @ACL’17*, pp. 22–31 (Section 4.6).

— (2018). “DRIM: MDL-Based Approach for Fast Diverse Summarization”. In: *Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence, WI’18*, pp. 660–663. DOI: 10.1109/WI.2018.00-17 (Section 4.6).

Vereshchagin, Nikolai and Paul Vitányi (2004). “Kolmogorov’s structure functions and model selection”. In: *IEEE Transactions on Information Theory* 50.12, pp. 3265–3290. DOI: 10.1109/TIT.2004.838346 (Section 3.2).

Vespier, Ugo et al. (2012). “MDL-Based Analysis of Time Series at Multiple Time-Scales”. In: *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML/PKDD’12*. Springer, pp. 371–386. DOI: 10.1007/978-3-642-33486-3_24 (Section 7.3).

Viamontes Esquivel, Alcides and Martin Rosvall (2011). “Compression of Flow Can Reveal Overlapping-Module Organization in Networks”. In: *Physical Review X* 1.2, p. 021025. DOI: 10.1103/PhysRevX.1.021025 (Section 6.4).

Vitányi, Paul and Ming Li (1999). *Minimum Description Length Induction, Bayesianism, and Kolmogorov Complexity*. arXiv: cs/9901014 (Section 3.1).

— (2000). “Minimum description length induction, Bayesianism, and Kolmogorov complexity”. In: *IEEE Transactions on Information Theory* 46.2, pp. 446–464. DOI: 10.1109/18.825807 (Section 3.1).

Vreeken, Jilles (2009). “Making Pattern Mining Useful”. PhD thesis. Universiteit Utrecht (Section 4.1).

— (2015). “Causal Inference by Direction of Information”. In: *Proceedings of the 2015 SIAM International Conference on Data Mining, SDM’15*. SIAM, pp. 909–917. DOI: 10.1137/1.9781611974010.102 (Section 9.1).

Vreeken, Jilles and Arno Siebes (2008). “Filling in the Blanks – Krimp Minimisation for Missing Data”. In: *Proceedings of the 8th IEEE International Conference on Data Mining, ICDM’08*. IEEE Computer Society, pp. 1067–1072. DOI: 10.1109/ICDM.2008.40 (Section 4.5).

Vreeken, Jilles, Matthijs van Leeuwen, and Arno Siebes (2007a). “Characterising the difference”. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’07*. ACM, pp. 765–774. DOI: 10.1145/1281192.1281274 (Section 4.3).

— (2007b). “Preserving Privacy through Data Generation”. In: *Proceedings of the 7th IEEE International Conference on Data Mining, ICDM’07*. IEEE Computer Society, pp. 685–690. DOI: 10.1109/ICDM.2007.25 (Section 4.5).

— (2011). “Krimp: Mining Itemsets that Compress”. In: *Data Mining and Knowledge Discovery* 23.1, pp. 169–214 (Section 4.1).

Vreeken, Jilles and Kenji Yamanishi (2019). “Modern MDL meets Data Mining Insights, Theory, and Practice [Tutorial]”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’19*. ACM, pp. 3229–3230. DOI: 10.1145/3292500.3332284 (Section 3.1).

Wallace, Christopher S. (2005). *Statistical and inductive inference by minimum message length*. Springer (Section 3.2).

Wallace, Christopher S. and D. M. Boulton (1968). “An Information Measure for Classification”. In: *The Computer Journal* 11.2, pp. 185–194. DOI: 10.1093/comjnl/11.2.185 (Section 3.2).

Wallace, Christopher S. and Jon D. Patrick (1993). “Coding Decision Trees”. In: *Machine Language* 11.1, pp. 7–22. DOI: 10.1023/A:1022646101185 (Section 3.5).

Wang, Chao and Srinivasan Parthasarathy (2006). “Summarizing Itemset Patterns Using Probabilistic Models”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD’19*. ACM, pp. 730–735. DOI: 10.1145/1150402.1150495 (Section 3.7).

Wang, Peng et al. (2010). “An algorithmic approach to event summarization”. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, SIGMOD’10*. ACM, pp. 183–194. DOI: 10.1145/1807167.1807189 (Section 7.2).

Webb, Geoffrey I. (2007). “Discovering Significant Patterns”. In: *Machine Learning* 68.1, pp. 1–33. DOI: 10.1007/s10994-007-5006-x (Section 3.7).

Webb, Geoffrey I. and Jilles Vreeken (2013). “Efficient Discovery of the Most Interesting Associations”. In: *ACM Transactions on Knowledge Discovery from Data* 8.3, 15:1–15:31. DOI: 10.1145/2601433 (Section 3.7).

Wiegand, Boris, Dietrich Klakow, and Jilles Vreeken (2021). “Mining Easily Understandable Models from Complex Event Logs”. In: *Proceedings of the 2021 SIAM International Conference on Data Mining, SDM'21*. SIAM, pp. 244–252. DOI: 10.1137/1.9781611976700.28 (Section 7.5).

— (2022). “Mining Interpretable Data-to-Sequence Generators”. In: *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI'22*. Association for the Advancement of Artificial Intelligence (Section 7.5).

Witteveen, Jouke et al. (2014). “RealKrimp – Finding Hyperintervals that Compress with MDL for Real-Valued Data”. In: *Proceedings of the 13th International Symposium on Advances in Intelligent Data Analysis, IDA'14*. Springer, pp. 368–379. DOI: 10.1007/978-3-319-12571-8_32 (Section 5.6).

Wu, Daoping et al. (2020). “Modeling Piece-Wise Stationary Time Series”. In: *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'20*. IEEE Computer Society, pp. 3817–3821. DOI: 10.1109/ICASSP40776.2020.9053470 (Section 7.3).

Wu, Ke et al. (2010). “Unsupervised text pattern learning using minimum description length”. In: *Proceedings of the 4th International Universal Communication Symposium, IUCS'10*, pp. 161–166. DOI: 10.1109/IUCS.2010.5666227 (Section 3.6).

Yan, Xifeng et al. (2005). “Summarizing itemset patterns: a profile-based approach”. In: *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'05*. ACM, pp. 314–323. DOI: 10.1145/1081870.1081907 (Section 3.7).

Yan, Yizhou et al. (2018). “SWIFT: Mining Representative Patterns from Large Event Streams”. In: *Proc. VLDB Endow.* 12.3, pp. 265–277. DOI: 10.14778/3291264.3291271 (Section 7.5).

Yang, Lincen, Mitra Baratchi, and Matthijs van Leeuwen (2020). *Unsupervised Discretization by Two-dimensional MDL-based Histogram.* arXiv: 2006.01893 (Section 5.6).

Youngblood, G. Michael et al. (2005). “Automated HPOMDP Construction through Data-mining Techniques in the Intelligent Environment Domain”. In: *Proceedings of the Eighteenth International Florida Artificial Intelligence Research Society Conference, FLAIRS'05* (Section 7.7).

Yurov, Maxim and Dmitry I. Ignatov (2017). “Turning Krimp into a Triclustering Technique on Sets of Attribute-Condition Pairs that Compress”. In: *Proceedings of the International Joint Conference on Rough Sets, IJCRS'17*. Springer, pp. 558–569. DOI: 10.1007/978-3-319-60840-2_40 (Section 5.3).

Zhao, Peng et al. (2019). “CLEAN: Frequent Pattern-Based Trajectory Spatial-Temporal Compression on Road Networks”. In: *Proceedings of the 20th IEEE International Conference on Mobile Data Management, MDM'19*. IEEE Computer Society, pp. 605–610. DOI: 10.1109/MDM.2019.00127 (Section 7.8).