Workshop Notes

Eighth International Workshop

"What can FCA do for Artificial Intelligence?"

FCA4AI 2020

24th European Conference on Artificial Intelligence

ECAI 2020

August 29 2020

Santiago de Compostela, Spain

Editors

Sergei O. Kuznetsov (NRU HSE Moscow)

Amedeo Napoli (LORIA Nancy)

Sebastian Rudolph (TU Dresden)

http://fca4ai.hse.ru/2020/

ECAI DIGITAL - 2020

Preface

The seven preceding editions of the FCA4AI Workshop showed that many researchers working in Artificial Intelligence are deeply interested by a well-founded method for classification and data mining such as Formal Concept Analysis (see `https://conceptanalysis.wordpress.com/fca/`). FCA4AI was co-located with ECAI 2012 (Montpellier), IJCAI 2013 (Beijing), ECAI 2014 (Prague), IJCAI 2015 (Buenos Aires), ECAI 2016 (The Hague), IJCAI/ECAI 2018 (Stockholm), and IJCAI 2019 (Macao). The workshop has now a quite long history and all the proceedings are available as CEUR proceedings (see `http://ceur-ws.org/`, volumes 939, 1058, 1257, 1430, 1703, 2149, and 2529). This year, the workshop has again attracted many researchers from many countries working on actual and important topics related to FCA, showing the diversity and the richness of the relations between FCA and AI.

Formal Concept Analysis (FCA) is a mathematically well-founded theory aimed at data analysis and classification. FCA allows one to build a concept lattice and a system of dependencies (implications and association rules) which can be used for many Artificial Intelligence needs, e.g. knowledge discovery, learning, knowledge representation, reasoning, ontology engineering, as well as information retrieval and text processing. Recent years have been witnessing increased scientific activity around FCA, in particular a strand of work emerged that is aimed at extending the possibilities of FCA w.r.t. knowledge processing, such as work on pattern structures, relational context analysis, and triadic analysis. These extensions are aimed at allowing FCA to deal with more complex data, both from the data analysis and knowledge discovery points of view. Actually these investigations provide new possibilities for AI practitioners within the framework of FCA. Accordingly, we are interested in the following issues:

- How can FCA support AI activities such as knowledge processing, i.e. knowledge discovery, knowledge representation and reasoning, learning, i.e. clustering, pattern and data mining, natural language processing, and information retrieval (non exhaustive list).
- How can FCA be extended in order to help Artificial Intelligence researchers to solve new and complex problems in their domains.

The workshop is dedicated to discussion of such issues. First of all we would like to thank all the authors for their contributions and all the PC members for their reviews and precious collaboration. This year, 24 papers were submitted and 14 were accepted for presentation at the workshop, out of which 6 short papers. The papers submitted to the workshop were carefully peer-reviewed by three members of the program committee. Finally, the order of the papers in the proceedings (see page 5) follows the program order (see `http://fca4ai.hse.ru/2020/`).

The Workshop Chairs

Sergei O. Kuznetsov
National Research University Higher School of Economics, Moscow, Russia

Amedeo Napoli
Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France

Sebastian Rudolph
Technische Universität Dresden, Germany

Program Committee

Mehwish Alam (AIFB Institute, FIZ KIT Karlsruhe, Germany)

Gabriela Arevalo (Universidad Austral, Buenos Aires, Argentina)

Jaume Baixeries (UPC Barcelona, Catalunya)

Karell Bertet (L3I, Université de La Rochelle, France)

Aleksey Buzmakov (National Research University HSE Perm, Russia)

Victor Codocedo (UFTSM Santiago de Chile, Chile)

MiguelCouceiro (LORIA, Nancy France)

Diana Cristea (Babes-Bolyai University, Cluj-Napoca, Romania)

Mathieu D'Aquin (National University of Ireland Galway, Ireland)

Florent Domenach (Akita International University, Japan)

Elizaveta Goncharova (NRU Higher School of Economics, Moscow, Russia)

Tom Hanika (University of Kassel, Germany)

Marianne Huchard (LIRMM/Université de Montpellier, France)

Dmitry I. Ignatov (National Research University HSE Moscow, Russia)

Dmitry Ilvovsky (NRU Higher School of Economics, Moscow, Russia)

Mehdi Kaytoue (Infologic, Lyon, France)

Jan Konecny (Palacky University, Olomouc, Czech Republic)

Ondrej Kridlo (University of P.J. Safarik in Kosice, Slovakia)

Francesco Kriegel (Technische Universität Dresden, Germany)

Leonard Kwuida (Bern University of Applied Sciences, Switzerland)

Florence Le Ber (ENGEES/Université de Strasbourg, France)

Tatiana Makhalova (National Research University HSE Moscow, Russia, and Inria LORIA, Nancy, France)

Nizar Messai (Université François Rabelais Tours, France)

Rokia Missaoui (UQO Ottawa, Canada)

Sergei A. Obiedkov (NRU Higher School of Economics, Moscow, Russia)

Jean-Marc Petit (Université de Lyon, INSA Lyon, France)

Uta Priss (Ostfalia University, Wolfenbüttel, Germany)

Christian Sacarea (Babes-Bolyai University, Cluj-Napoca, Romania)

Henry Soldano (Laboratoire d'Informatique de Paris Nord, Paris, France)

Renato Vimieiro (Universidade Federal de Minas Gerais, Belo Horizonte, Brazil)

Contents

Embedding Formal Contexts Using Unordered Composition*

Esteban Marquer, Ajinkya Kulkarni, and Miguel Couceiro

Université de Lorraine, CNRS, Inria N.G.E., LORIA, F-54000 Nancy, France
{esteban.marquer,ajinkya.kulkarni,miguel.couceiro}@inria.fr

Abstract. Despite their simplicity, formal contexts possess a complex latent structure that can be exploited by formal context analysis (FCA). In this paper, we address the problem of representing formal contexts using neural embeddings in order to facilitate knowledge discovery tasks. We propose *Bag of Attributes* (BoA), a dataset agnostic approach to capture the latent structure of formal contexts into embeddings. Our approach exploits the relation between objects and attributes to generate representations in the same embedding space. Our preliminary experiments on attribute clustering on the SPECT heart dataset, and on co-authorship prediction on the ICFCA dataset, show the feasibility of BoA with promising results.

Keywords: Formal Concept Analysis · Vector Space Embedding · Neural Networks · Complex Data · Link Prediction · Clustering.

1 Introduction

In recent years there has been an increasing interest in approaches to combine formal knowledge and artificial neural networks (NN). As mentioned in [1, p. 6], these approaches have a "generally hierarchical organization", with the "lowest-level network [taking] raw data as input and [producing] a model of the dataset". These networks represent data as real-valued vectors called *embeddings*.

Formal concept analysis (FCA) is another powerful tool for understanding complex data. Replicating its mechanisms using NNs could help processing complex and large datasets [5,17] by tackling FCA's scalability issues. Following this idea, we want to reproduce the general extraction process of FCA with NN architectures. This asks for a general embedding framework for contexts capable of handling data of arbitrary dimensions while encoding much of the contextual information. To our knowledge, there are only a few approaches in this direction [5,10,17,20]. Dürrschnabel *et al.* [5] propose FCA2VEC to embed formal contexts by encoding FCA's closure operators. It has three main components: *attribute2vec* and *object2vec*, (both based on *word2vec* [19]), and *closure2vec* that relies on a distance between closures of sets of attributes. In fact, *closure2vec*

* We would like to thank the Inria Project Lab (IPL) HyAIAI ("Hybrid Approaches for Interpretable AI") for funding the research internship of E. Marquer.

is based on [20] that showed how to encode closure operators with simple feed-forward NNs. An advantage of this framework is that it produces embeddings of low dimensions (2 and 3) to facilitate interpretability. Yet FCA2VEC has several limitations: the embedding models need to be trained on each formal context (without guaranties of generalization) and the embeddings for objects and attributes are not defined in the same embedding space, which can be problematic when processing objects and attributes together.

To overcome these limitations, we propose *Bag of Attributes* (BoA) for providing a shared embedding space for objects and attributes by composing object embeddings from attribute embeddings. It is based on unordered composition and *long short-term memory neural network* (LSTM). Also, it predicts the number of concepts and a new measure of attribute similarity, called *co-intent similarity*, based on metric learning. This novel approach differs from the existing ones as it is agnostic to the data. Indeed, the model can accommodate any number of objects and attributes and it generalizes on real-world formal contexts despite being trained on randomly generated ones. We also explore the advantages and limits of our approach and provide experimental results on attribute clustering on the SPECT heart dataset[1], and on co-authorship prediction on the ICFCA dataset[2]. The comparison of BoA with FCA2VEC shows competitive performances on both tasks.

The paper is organized as follows. In Section 2, we briefly recall some basic background on FCA and NN. The architecture of BoA is defined in Section 3, whereas the corresponding training process and datasets are presented in Section 4. We discuss the impact of datasets characteristics on the performance in Section 5. Section 6 describes our experiments on attribute clustering and co-authorship prediction using real-world datasets[3]. Finally, we discuss potential developments for future work in Section 7.

2 Preliminaries And Basic Background

In this section we briefly recall some basic background on FCA (Subsection 2.1) and deep learning (Subsection 2.2 and 2.3), and define the new *co-intent similarity* for attributes. For further details on FCA see, *e.g.*, [8,12], and on NN architectures see, *e.g.*, [11].

2.1 Formal Concept Analysis

A *formal context* is a triple $\langle A, O, \mathbf{I} \rangle$, where A is a finite set of attributes, O is a finite set of objects, and $\mathbf{I} \subseteq A \times O$ is an incidence relation between A and O. A formal context can be represented by binary table C with objects as rows C_o and

[1] `https://archive.ics.uci.edu/ml/datasets/SPECT+Heart`

[2] `https://github.com/tomhanika/conexp-clj/tree/dev/testing-data/icfca_community`

[3] Compared to graph neural network approaches, FCA2VEC shows an improvement of at least 5% on all metrics for link prediction.

attributes as columns C_a, for $o \in O$ and $a \in A$. The entry of C corresponding to o and a is defined by $C_{o,a} = 1$ if $(o, a) \in \mathbf{I}$, and 0 otherwise.

It is well known [8] that every formal context $\langle A, O, \mathbf{I} \rangle$ induces a Galois connection between objects and attributes: for $X \subseteq O$ and $Y \subseteq A$, defined by: $X' = \{y \in A \mid (x, y) \in \mathbf{I} \text{ for all } x \in X\}$ and $Y' = \{x \in O \mid (x, y) \in \mathbf{I} \text{ for all } y \in Y\}$. A *formal concept* is then a pair (X, Y) such that $X' = Y$ and $Y' = X$, called respectively the *intent* and the *extent*. It should be noticed that both X and Y are closed sets, i.e., $X = X''$ and $Y = Y''$. The set of all formal concepts can be ordered by inclusion of the extents or, dually, by the reversed inclusion of the intents. We denote by $I \subseteq 2^A$ the set of intents and $E \subseteq 2^O$ the set of extents.

2.2 Auto-Encoders, Embeddings And Metric Learning

Auto-encoders are a class of deep learning models composed of *(i)* an encoder, that takes some x as an input and produces a latent representation z, and *(ii)* a decoder, that takes z as an input and reconstructs $\hat{x}$ a prediction of x. The model learns to compress x into z by training the the model to match x and $\hat{x}$. The training objective matching x and $\hat{x}$ is the *reconstruction loss*. Auto-encoders are one of the methods to generate representation of data as vectors. In that case z is called the *embedding* of x, and the real-valued space in which z is defined is called the *embedding space*.

Unlike traditional auto-encoders, *variational auto-encoders* (VAEs) [14] encode a distribution for each value of z instead of the value itself. In practice, for each component of z, the encoder produces two values: a mean μ and a standard deviation σ. When training the model, z is sampled from the normal distribution defined by μ and σ. Finally, the distribution defined by μ and σ is normalized by adding the *Kullback–Leibler (KL) divergence* loss term. To make this process differentiable and be able to train the model, a method called *reparametrization* [14] is used. VAEs are known to provide better generalization capabilities and are easier to use to decode arbitrary embeddings, compared to classic auto-encoders. This property is useful for generation since we can train a model generating embeddings and then decode them with a pre-trained VAE. In fact, VAEs are used in a wide variety of applications to improve the quality of embedding spaces, *e.g.*, for image [14], for speech [16] and for graph generation [15].

Metric learning [18] is a training process used ensure that embedding spaces have the properties of metric spaces. To achieve this, a loss is used to reduce the distance between the embeddings of equal elements and increase the distance between embeddings of different elements. Multiple losses can achieve this, such as the *pairwise loss* and the *triplet loss*. Triplet loss considers the embeddings of three elements: an input x_1, some x_2 judged equal to x_1 and some y different from x_1. In some approaches [16] a predictor (typically a *multi-layer perceptron* (MLP)) is used to predict a distance between the embeddings instead of applying a standard distance directly on the embeddings. It is possible to learn distances on different properties of the embedded elements, by splitting the embedding into segments and learn a different distance on each one [16]. Metric learning is usually used to approximate actual distances. However, this process can be

applied to learn other kinds of measures not fitting the definition of a distance, which is what we do in this paper.

We need both "equal" and "different" attributes to use metric learning losses on attribute embeddings. Nonetheless, even if we consider equivalent attributes (*i.e.* with the same extent) as "equal", they are usually rare within a given context. We define co-intent similarity to compare attributes and avoid this issue. Given two attributes a_1 and a_2 we define their *pairwise co-intent similarity* as:

$$\text{co-intent}(a_1, a_2) = \begin{cases} 1 & \text{if } |\{i \in I | a_1 \in i\}| + |\{i \in I | a_2 \in i\}| = 0 \\ \dfrac{2 \times |\{i \in I | a_1 \in i, a_2 \in i\}|}{|\{i \in I | a_1 \in i\}| + |\{i \in I | a_2 \in i\}|} & \text{otherwise.} \end{cases} \tag{1}$$

In other words, it is the ratio of intents containing both attributes over the intents containing a_1 or a_2[4]. In cases where no intent contain the attributes (both attributes are empty or padding columns), the similarity is set to 1. Co-intent similarity ranges from 0, for attributes never appearing in the same intents, to 1, for attributes always appearing together or for identical attributes.

2.3 Unordered Composition

By *unordered composition functions* we mean operations that do not take into account the order of the input elements and that can accommodate any number of input elements. Typical examples are the componentwise min, max, and average (also called respectively min-, max-, and average-pooling). Unordered composition-based models have proven their effectiveness in a variety of tasks, for instance, sentence embedding [13], sentiment classification [4] and feature classification [9]. On the one hand, this family of methods allows inputs of varying sizes to be processed at a relatively low computational cost, by opposition to recurrent models like LSTM [11]. On the other hand, we lose the information related to the order of the input elements.

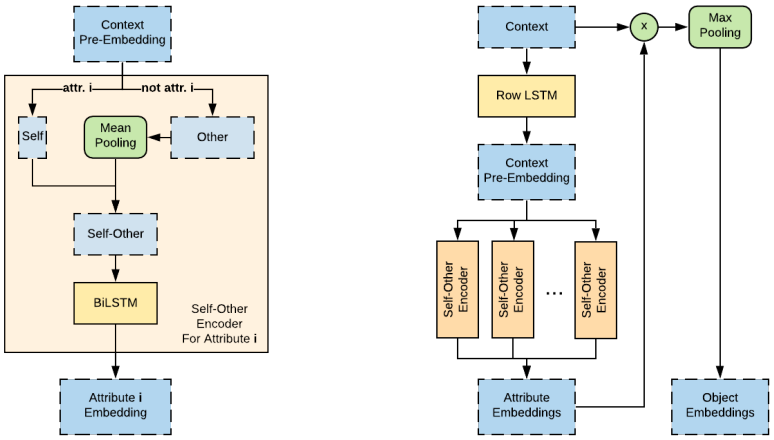
3 Proposed approach

In this section we first define the proposed approach and explain the objectives used to train the model. The training process is detailed in Section 4.

3.1 Bag of Attributes

Bag of Attributes (BoA) takes a formal context as input, and produces embeddings for its attributes. Then, object embeddings are computed using the embeddings of the attributes and the formal context. BoA has four main components: a pre-embedding generator, an attribute encoder called *self-other encoder* to compute the attribute embedding, an object encoder and a decoder. It considers the

[4] Observe that this is essentially the Jaccard index on the set of intents.



(a) Self-other attribute encoder for an attribute. (b) BoA encoder architecture.

Fig. 1: Schematic representation of the BoA architecture. Blue blocks correspond to tensors, the orange to neural components and green blocks to non-neural computations. Arrows joining blocks represent concatenation of tensors.

attributes as an unordered set to produce the object and attribute embeddings. The name is inspired by *Bag of Words* (BoW) [19]. The structure of the encoder is schematized in Figure 1. The decoder itself is an MLP predicting if an object has an attribute or not (1 or 0, respectively). Its input is the concatenation of the object and the attribute embeddings. A sigmoid function applied on the output ensures it is in $[0, 1]$. BoA is trained as a VAE on formal contexts, so a μ and σ vector is produced for each attribute. The sampling of the attribute embeddings is done before the generation of object embeddings.

The order of the attributes in the dataset does not matter for FCA, therefore in BoA each attribute is processed in a similar manner to capture this property. Each attribute is compared to all the other attributes, for each object of the dataset. In practice, the column of an attribute (*self*) is compared to an unordered composition (average-pooling) of all the other attributes (*other*). *Self* and *other* are then processed by a *bidirectional LSTM* (BLSTM) [11], with the object dimension as the sequence dimension. The last hidden state of the BLSTM is processed with a feed-forward layer into an embedding that represents the attribute. The structure of the attribute encoder is presented on Figure 1a. Finally, the object embeddings are computed by applying max-pooling on the embeddings of the attributes present in the object's intent. We apply a LSTM on each row of C before the *self-other encoder*, as it produces different embeddings for each attribute despite the same input, to allow the model to determine which attribute is involved by avoiding the use unordered composition directly on C.

3.2 Training Objective

We train BoA using KL divergence on the attribute embeddings exclusively as the sampling happens before the computation of the object embeddings. We use the *binary cross entropy* loss for reconstruction because the model predicts between two classes (1 and 0). On top of that, we use metric learning with the new *co-intent similarity* (see Equation 1) and the number of concept [8], with *mean square error* (MSE) as the loss function. Predicting the number of concepts from the context without actually computing the intents, helps when generating the set of concepts using neural models. Indeed, knowing how many elements to generate beforehand facilitates the generation process. We use MLPs to predict the co-intent similarity and number of concepts. For co-intent similarity between two attributes a_1 and a_2, the input is the concatenated embeddings of a_1 and a_2 and a sigmoid output function is added to ensure the predicted similarity is in $[0, 1]$. We apply a max-pooling over the attribute embeddings before predicting the number of concepts, which corresponds to a *deep averaging network* (DAN) [13].

4 Training Setup

In this section we present our training process (Subsection 4.1), dataset (Subsection 4.2) and we describe the data augmentation process (Subsection 4.3).

4.1 Training Process

We train BoA in two phases of 5000 epochs each. In the first phase, we apply the reconstruction loss and the KL divergence only. Then, we gradually introduce the prediction of the co-intent similarity and of the number of concepts. When using metric learning with multiple distances, a common approach is to split the embedding space and to learn one distance per sub-part of the embedding space [16]. We apply the same principle and use 50% of the embedding space to predict the co-intent similarity and 25% for the number of concepts. The exact embedding dimension of BoA is 128, with a pre-embedding of size of 64. The LSTM and the BLSTM have two layers each. The decoder MLP has four layers, and the MLPs used for distance prediction both have two layers. We use a *rectified linear unit* activation function between all the layers of the model.

4.2 Training Data

The dataset used for training the BoA model is composed of 6000 randomly generated formal contexts split into training and validation, and the corresponding intents computed using the Coron system[5]. To generate a context of $|O|$ objects and $|A|$ attributes we sample $|O| \times |A|$ values from a Poisson distribution and apply a threshold of 0.3. Values under the threshold correspond to 1 in the context, which leads to a density around 0.3. Note that the random generation

[5] `http://coron.loria.fr/site/index.php`

Table 1: Descriptive statistics on the dataset of randomly generated contexts.

	Dataset	# Object	# Attribute	# Concept	Context density
Mean	train	12.83 ± 6.11	12.98 ± 6.03	77.93 ± 78.39	0.329 ± 0.057
$\pm$ std.	test	12.83 ± 6.13	12.97 ± 6.04	78.12 ± 77.27	0.332 ± 0.057
Range	train	1 to 20	2 to 20	1 to 401	0 to 0.56
	test	2 to 20	3 to 20	2 to 401	0 to 0.49

process may result in empty rows and columns, which will be dropped by Coron if they are at the extremities of the context. For this reason, the actual size of the generated context may be smaller than the requested one. We generate a training set of 5000 contexts and a test set of 1000 samples. For the training phase, a development set of 10% of the training set is randomly sampled from the training set. For each set, we generate different sizes of contexts, 20% of each: 5×5, 10×10, 10×20, 20×10 and 20×20 contexts ($|O| \times |A|$). The statistics of the generated datasets are reported in Table 1.

4.3 Data Augmentation

We rely on plain random generation for the formal contexts, and not on more involved generation processes as discussed in [7,6], so the random data is biased. We introduce a simple way to compensate some of those biases while improving the generalization capability of the model. We implement the following data augmentation pipeline: *(i)* duplicating of objects and attributes, *(ii)* inverting the value of entries and *(iii)* shuffling objects and attributes. With this process, we simulate identical (duplication) and nearly identical (duplication + drop) objects or attributes that appear in real-world datasets.

Objects and attributes have a probability p of being duplicated. If duplicated, they have the same probability p of being duplicated again. From this definition, the number of copies of an object (or attribute) follow a geometric law with a probability of success p. Consequently, the exact number of objects and attributes actually seen during training do not match the ones reported in Table 1. Nonetheless, the duplication follows a geometric law so we can estimate the number amount of object and attribute seen as $number/(1-p)$. Inverting some randomly selected values in the formal context is our adaptation of dropout, a common technique in deep learning. The shuffling after duplication avoids model's reliance on order of the objects and the attributes. We set the duplication probability to $p = 0.1$ and the drop probability to 0.01. In this setting, the estimated average object and attribute numbers are respectively 14.25 and 14.42, for both the training and development sets.

When co-intent similarity is used, duplication and shuffling are reproduced on the intents. However, drops in a formal context alter the corresponding lattice, so they are not applied at all when using data from the lattice. This precaution avoids making the model insensitive to small variations in the input.

4.4 Issues With KL Divergence

When adding the KL divergence to the prototype of BoA (initially a simple auto-encoder) the performance of the model was greatly impaired. The analysis of the predictions revealed the model was going for "low hanging fruits" and ignored the embeddings themselves, as described in [3]. To solve this issue we apply *annealing* [3] and multiply the KL divergence by a lambda that we set to 10^{-3}. This reduces the impact of the KL divergence on the training and allows the model to learn some features before the KL divergence comes into effect. However, it reduces the benefits we get from using a VAE.

5 Exploring The Limits Of BoA

We now explore the limits of BoA *w.r.t.* input data. All experiments are performed on randomly generated data to control of the evaluation process.

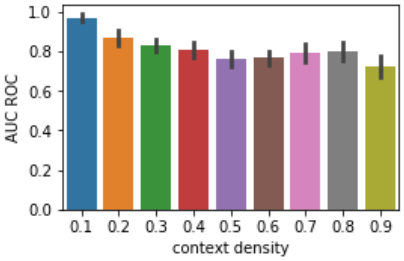
5.1 Reconstruction Performance

To assess the reconstruction performance of the BoA auto-encoder, we use the *area under the ROC curve* (AUC ROC). It allows to determine whether the BoA has good predictive capacity and, similarly to the F1 measure, gives a general account of performance. To determine if the results are significantly different, we use Student t-test on means. The results are presented in Figure 2.

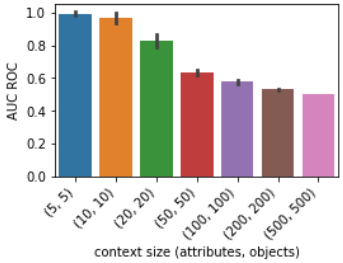
We first evaluate the impact of the density on the reconstruction by comparing the performance on random contexts with densities from 0.1 to 0.9. We use 100 samples per density with a fixed size of 20 objects and attributes. Student's t-test show significant differences between the performance with the various densities: all the p-values are under 0.01 except between 0.4 and 0.8 (0.24), 0.5 and 0.6 (0.39), and 0.7 and 0.8 (0.19). However, the model performance stays overall stable across the densities, while slightly better with smaller densities. We suspect this tendency is due to the composition process of the object embeddings: the higher the density, the more attributes are present for an object, so more attribute embeddings are involved in the composition of the object embeddings, making it more complex to decode.

We also examine the effect of the size on the AUC ROC. Square random contexts ($|O| = |A|$) of sizes in $\{5, 10, 20, 50, 100, 200, 500\}$ and a fixed density of 0.3 are used for this experiment, with 20 samples per size. The model performs very well for seen data sizes with a slight drop to 0.83 for 20 objects and attributes. As expected, the performance drops when manipulating larger contexts. For 50 objects and attributes (2.5 times the maximum seen size), the AUC ROC is above 0.63, but from 100 objects and attributes onward, it drops under 0.6. Finally, with 500 objects and attributes (25 times the largest seen data size and 4 times the embedding size), the reconstruction AUC ROC falls to 0.50 in average. This is the limit of reconstruction performance with the current training process.

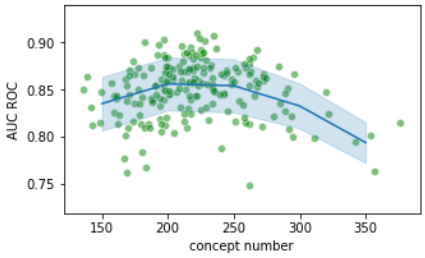
Finally, we examine the impact of the number of concepts on the performance of the model. We use contexts of fixed size (20 objects and attributes) from the



(a) Impact of the density, from 0.1 to 0.9, 100 samples per density.



(b) Impact of the size, from 5 to 500 objects and attributes, 20 samples per size.



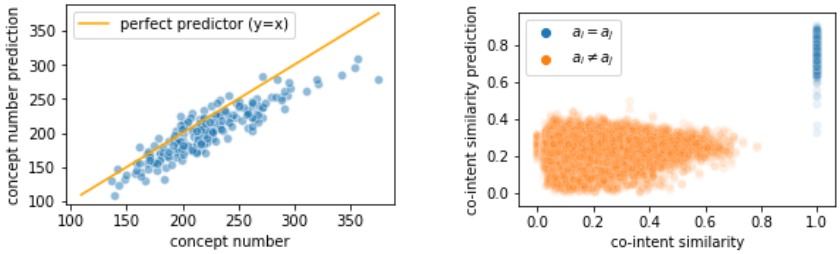
(c) Impact of the concept number, for 200 sample with 20 objects and attributes. The blue line is the general tendency when rounding the concept number to 50.

Fig. 2: Reconstruction performance on random contexts. The error bars and the shaded area correspond to the standard deviation.

test set, totaling 200 contexts. We consider the concept number as an indicator of the variety of attributes and objects in the context. Indeed, if the concept number is high for a given size of context, we can expect the context to be close to the clarified context (context with no equivalent objects or attributes). This implies a lower amount of duplicate objects and attributes. In addition, we can expect the model to have a harder time encoding and decoding irregular contexts than repetitive ones. Consequently, the drop of the AUC ROC for higher concept numbers is not surprising. However, we also observe a lower performance around 150 concepts. This second decrease requires further investigation.

5.2 Metric Learning Performance

We evaluate the performance of BoA for the co-intent similarity and number of concepts' prediction, by computing the attribute embeddings and applying the predictors trained together with the BoA model. We use the 200 contexts of 20 objects and attributes from the test set. The prediction results are reported Figure 3. The Pearson correlation coefficient is 0.9 between the actual concept number and the prediction, indicating a strong correlation. We can notice the tendency of the model to under-evaluate the concept number. Even though BoA



(a) Concept number. (b) Co-intent similarity.

Fig. 3: Predicted pseudo-metrics against the actual values, for the 200 samples with 20 objects and attributes from the test set.

manages to differentiate a_i and a_j, when $a_i = a_j$, the predictions in the other cases are not clear: for similarities between 0 and 0.8, they seem randomly picked between 0 and 0.4. The analysis of the training process reveals a small difference of the MSE between the first and the last training epochs: from 0.17 to 0.05.

6 Experiments On Real-World Datasets[6]

To evaluate the performance of BoA on real-world datasets, we follow the empirical setting of [5]: we reproduce their link prediction and attribute clustering tasks, used to evaluate object2vec (o2v) and attribute2vec (a2v), respectively. We use the same ICFCA dataset as [5] for link prediction. For attribute clustering however, we use SPECT heart[7] as it is smaller than *wiki44k* [5], with dimensions closer to the training data: 68 objects and 23 attributes. We train the CBoW and SG variants of FCA2VEC models using the same settings as in [5], with 20 random iterations of each model and an embedding size of 3. To obtain comparable results, we reduce the embeddings produced by BoA to 3 dimensions by applying two standard dimensionality reduction techniques: *principal component analysis* (PCA) and *t-distributed stochastic neighbor embedding* (TSNE). We use Student t-test on means to determine if the results are significant.

We report the link prediction performance in Table 2. The three BoA variants show a significantly different performance from o2v SG, with all the p-values lower than 0.005. We found that the classifier based on BoA, the one with the best F1 score, systematically answers positive. Additionally, we fail to reproduce the performance of [5] (F1 score of 0.69 for o2v CBoW, 0.66 for o2v SG) Finally, the ICFCA context is very sparse: it has a density of 0.003 on the train and 0.005 on the test set. Due to this, the task may not be representative of the

[6] Experiments were carried out using the Grid'5000 testbed, supported by a scientific interest group hosted by Inria and including CNRS, RENATER and several Universities as well as other organizations (see `https://www.grid5000.fr`).

[7] `https://archive.ics.uci.edu/ml/datasets/SPECT+Heart`

Table 2: Performance on the link prediction task (mean ± std.).

Model	Precision	Recall	F1
o2v CBoW	0.63 ± 0.05	0.46 ± 0.05	0.53 ± 0.05
o2v SG	0.70 ± 0.04	0.49 ± 0.03	0.57 ± 0.02
BoA PCA 3d	0.65	0.42	0.51
BoA TSNE 3d	0.58	0.67	0.62
BoA	0.50	1.00	0.67

Table 3: Performance on the attribute clustering task with 2, 5 and 10 clusters.

Model	k = 2	k = 5	k = 10
a2v CBoW	0.66 ± 0.00	0.14 ± 0.02	0.063 ± 0.013
a2v SG	0.35 ± 0.13	0.11 ± 0.03	0.042 ± 0.010
BoA TSNE 3d	0.30	0.29	0.044
BoA PCA 3d	0.70	0.22	0.051
BoA	0.70	0.22	0.051

performance of the object embeddings on most datasets. These results hint that the task needs to be adapted to get proper insight on the object embedding performance. The attribute clustering performance is reported in Table 3. In this experiment, we find that the CBoW variant performs significantly better than the SG (all t-test p-values under 0.0005). This is the opposite of the result found by [5] for attribute clustering. However, this result may be due to using a different dataset. Interestingly, the BoA PCA variant performs equally to the full BoA. The performance of BoA (and BoA PCA) is significantly better than a2v CBoW for 2 and 5 clusters (p-values under 10^{-14}). For 10 clusters however, a2v CBoW performs significantly better (p-value under 0.001). The model improves the performance of a2v CBoW by 4% for 2 clusters and 8% for 5 clusters.

7 Conclusion And Future Work

We introduced the *co-intent similarity* for attributes and proposed BoA, a generalized embedding framework for formal contexts that integrates several FCA aspects. Our framework is data agnostic and scales to real-world datasets such as the SPECT heart dataset. It is also robust *w.r.t.* variations in the density and the concept number of formal contexts. The experimental results are encouraging, as our general approach achieves performance similar to FCA2VEC, a dataset specific one. In addition to being data agnostic, BoA constitutes a promising alternative to FCA2VEC since it uses a single embedding space for all contexts. Moreover, the asymptotic time complexity of embedding a formal context through BoA is $\theta(|O| \times |A|^2)$. In comparison, applying a linear embedding (the most basic embedding) or an LSTM embedding model (like our pre-embedding) to each entry has a complexity of $\theta(|O| \times |A|)$.

As future work we aim to tackle two active issues in the FCA community: the random generation of contexts and the scalability of concept lattices. Random

context generation introduces several biases as it does not match the distribution of real-world formal contexts (see *e.g.*, the "stegosaurus effect" discussed in [2]). It could be beneficial to use more accurate generation algorithms than our current algorithm, like those discussed in [6,7]. Nonetheless, it is encouraging to see that BoA achieves acceptable performance when trained on simply generated contexts of relatively small size (up to 20 objects and attributes). Since this preliminary work enables the use of decoders in generation processes, using NNs seems a feasible direction for concept lattices construction. These are some directions of current ongoing work.

References

1. Besold, T.R., *et al.*: Neural-Symbolic Learning and Reasoning: A Survey and Interpretation. CoRR **1711.03902** (2017)
2. Borchmann, D., Hanika, T.: Some experimental results on randomly generating formal contexts. In: CLA. vol. 1624, pp. 57–69 (2016)
3. Bowman, S.R., *et al.*: Generating sentences from a continuous space pp. 10–21 (2016)
4. Chen, X., *et al.*: Adversarial deep averaging networks for cross-lingual sentiment classification. CoRR **1606.01614** (2016)
5. Dürrschnabel, D., *et al.*: Fca2vec: Embedding techniques for formal concept analysis. CoRR **1911.11496** (2019)
6. Felde, M., *et al.*: Formal context generation using dirichlet distributions. In: Graph-Based Representation and Reasoning. pp. 57–71. Cham (2019)
7. Ganter, B.: Random extents and random closure systems. In: CLA (2011)
8. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer (1999)
9. Gardner, A., *et al.*: Classifying unordered feature sets with convolutional deep averaging networks. CoRR **1709.03019** (2017)
10. Gaume, B.e.: Clustering bipartite graphs in terms of approximate formal concepts and sub-contexts. IJCIS **vol. 6**, 1125–1142 (2013)
11. Graves, A., *et al.*: Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural networks **18**(5-6), 602–610 (2005)
12. Ignatov, D.I.: Introduction to formal concept analysis and its applications in information retrieval and related fields. CoRR **1703.02819** (2017)
13. Iyyer, M., *et al.*: Deep unordered composition rivals syntactic methods for text classification. In: 53rd ACL and 7th IJCNLP. pp. 1681–1691 (2015)
14. Kingma, D., Welling, M.: auto-encoding variational bayes (2013)
15. Kipf, T.N., *et al.*: Variational Graph Auto-Encoders. CoRR **1611.07308** (2016)
16. Kulkarni, A., *et al.*: Deep Variational Metric Learning For Transfer Of Expressivity In Multispeaker Text To Speech (2020), to appear in SLSP 2020
17. Kuznetsov, S.O., *et al.*: On neural network architecture based on concept lattices. In: Foundations of Intelligent Systems. pp. 653–663. Cham (2017)
18. Lin, X., *et al.*: Deep variational metric learning. In: Computer Vision – ECCV. pp. 714–729. Cham (2018)
19. Mikolov, T., *et al.*: Efficient Estimation of Word Representations in Vector Space. pp. 1–12 (2013)
20. Rudolph, S.: Encoding closure operators into neural networks. In: NeSy. pp. 40–45. CEUR-WS. org (2007)

Improving User's Experience in Navigating Concept Lattices: An Approach Based on Virtual Reality

Christian Săcărea and Raul-Robert Zavaczki

Babeş-Bolyai University Cluj-Napoca
csacarea@cs.ubbcluj.ro, zavaczki.raul@gmail.com

Abstract. Concept lattices are an elegant and effective tool for knowledge representation. In the last decades, there have been many advances in FCA theory building, offering a deep understanding of the theoretical foundations. There are many very efficient algorithms to compute concept lattices and FCA has been extended and generalized (pattern structures, triadic, n-adic FCA, etc.). Many projects have been conducted and there is an impressive list of FCA related software[1]. Due to technological advances, we have now new and interesting possibilities for representing concept lattices. This paper's aim is to discuss how virtual reality (VR) can improve user's experience in navigating concept lattices. Using VR rooms, we can have a totally new 3D experience of a concept lattice, we can freely move through that structure, exploring it from literally many points of view, and can communicate in a multi-player setting with other users, fulfilling R. Wille's dream of conceptual landscapes of knowledge.

1 Introduction

FCA has been designed from its very beginning, as a mathematical theory aiming to offer an effective reasoning support for scientists, practitioners and users in their attempt to analyse data. The key word was *restructuring* and its seminal paper [9] was also programmatic. R. Wille mentioned that "... the connections of the theory to its surroundings are getting weaker and weaker, with the result that the theory and even many of its parts become more isolated." Ten years later, FCA developed to "a set-theoretical model for concepts and conceptual hierarchies, allowing the mathematical study of the representation, inference, acquisition, and communication of conceptual knowledge" [10]. This development led then to *conceptual landscapes of knowledge* [8], a paradigm which allows a unifying model to various tasks of knowledge processing. For 20 years, conceptual structures have been implemented in various software systems but the representation of knowledge tended to be two-dimensional, due to inherent technological limitations.

For improving effectiveness and also the user's experience in navigating concept lattices, modern technological advances makes possible to represent conceptual knowledge in 3D. As real-life landscapes are three dimensional, conceptual

[1] https://conceptanalysis.wordpress.com/fca-software/

landscapes of knowledge should be also represented in a 3D environment, which comes with a couple of challenges and need for innovation. In this paper, we present our approach, own views and ideas of a translation of 2D representation of concept lattices in 3D, followed by some technical solutions of the implementation.

2 Related Work

At the best of our knowledge, there is no well established available tool to represent concept lattices and ToscanaJ[2] related features in a VR environment. Nevertheless, there is a plethora of software tools and implementations developed by various groups of our FCA community, which are targeting several problems and approaches. Without any claim of being comprehensive, we briefly present some of these contributions.

`LatViz` [1] proposes improvements over existing tools, as well as new functionalities such as visualization of Pattern Structures, concept annotations, intuitive visualization of implications. `ConExp`[3] is meanwhile a classic among FCA software tools, while `ConExp-FX`[4], and `ConExp-NG`[5] are reimplementations of the former. The latter one is part of `FCA Tools`, a collection of FCA related software tools[6]. `FCA Tools Bundle` [4] is a growing collection of FCA tools, including dyadic, triadic, analogical complexes, local navigation in triadic data sets, and a method for narrowing down the set of concepts using a like-dislike feature based on ASP.

Nevertheless, at this moment, there is no commercial software implementing FCA methods, and, paradoxically, exactly those FCA varieties having the most potential for real life applications are neglected: many-valued contexts and temporal FCA. Many-valued contexts are handled by the ToscanaJ suite [2], and an attempt to implement scaling features is done in `FCA Tools Bundle`[7] [4]. Besides scale building (which is done using Elba), the ToscanaJ suite includes also conceptual landscapes navigation capabilities, by defining a browsing scenario and then perform navigation [8]. Unfortunately, ToscanaJ has not been updated for a long time and there is a need to implement a more modern version.

As times evolve and new technologies emerge, new opportunities arise allowing innovation upon existing technologies by combining these technologies with some research fields in a way that improves the experience that an user has, or flattening the learning curve of otherwise rather difficult fields/technologies.

Virtual Reality (VR) is not at all new, but recent advances make it particularly suitable for collaborative learning, for improving conceptual knowledge [7], or for innovative communication in health care [5].

[2] http://toscanaj.sourceforge.net/
[3] http://conexp.sourceforge.net/
[4] https://github.com/francesco-kriegel/conexp-fx
[5] https://github.com/fcatools/conexp-ng
[6] https://github.com/fcatools
[7] fca-tools-bundle.com

3 Setting the stage

With the development of new technologies and game engines[8], the modern graphic capabilities of these technologies increased dramatically and it lies at hand to include them in new FCA software tools. Good practices in FCA data analysis projects showed that decoupling all technical aspects related to data preparation, scale building, conceptual schemata, etc, from the actual graphical navigation experience in conceptual landscapes is beneficient. This is especially important for users that have almost no technical background. Especially for applications of FCA in data analysis projects, in order to make the users comfortable enough to make the concept analysis by themselves and not be overwhelmed by the various tools and various concepts surrounding Formal Concept Analysis, we need to decouple the creation with the exploration of the lattices. This decoupling allows the first one to be done by someone that knows the concepts, possibly a Data Scientist, while allowing the user or researcher to focus on the latter one. This approach is not new, and it has been already implemented in the `ToscanaJ` management suite. On the other hand, throwing the user with a VR headset on in a room with a concept lattice in front of him will not do the trick though, because there are two factors that come into play – one comes if the user never experienced Virtual Reality before, which is already a pretty overwhelming experience, and the other one comes if he has never seen a concept lattice before. The first one is resolved by the gaming feel such environment gives the user, and the other one should be resolved by presenting the user a really basic tutorial, in which is stated plainly what the objects and attributes are, and how can you read the relationship between them.

At this point of early stage setting, the aim is gamifying the FCA experience. For this, we need an intuitive and captivating technological environment where an intuitive tool, specifically designed for non FCA experts can be used.

4 Stage Design

We propose an approach that allows the user to explore and navigate through concept lattices in a Virtual Reality environment.

For this, we have developed a tool that is designed specifically for people that had no connection to FCA before, or even mathematics. This is a process that has been proven to be a real challenge as every person is different and what may work for someone may not work for someone else. To overcome this problem, we revised every functionality and user experience aspect (controls, navigation, visualization) based on the feedback we received from various sources. These sources varied from students that had no experience with FCA trying the project in their free time, to university staff and senior FCA researchers.

Because feedback came from such many sources, it gave us a good overview over the overall aspect of the application, the common ground of these people, and future improvements.

[8] https://unity3d.com/

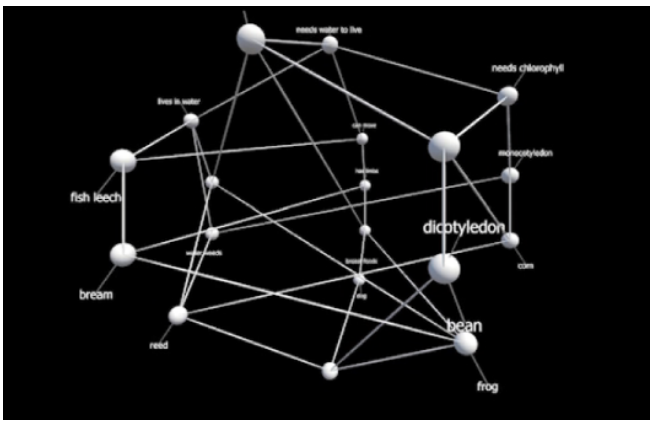


Fig. 1: 3D view of a concept lattice: Live in water data set

5 Technologies

The technologies used are Unity3D, for a compatible working space that currently supports VR Headsets, meaning that it is a cross platform (and by thus it supports HTC Vive, Oculus,etc.), and SteamVR 2D Plugin, the actual plugin that helps us with VR I/O operations.

5.1 Unity

Unity is a game engine developed by Unity Technologies. It is cross-platform, and it is mainly used to develop games, simulations, and can be also used to develop Virtual Reality and Augmented Reality applications. Unity offers a scripting API over the engine in C#, allowing the developer only to focus on the developing of the application, the engine handling the Virtual Reality rendering itself and the loading of 3D models.

5.2 HTC Vive

HTC Vive is a Virtual Reality Headset developed by HTC and Valve. As a tracking system it uses two lighthouses, which are base stations emitting pulsed IR lasers, allowing it to use a room scale technology that allows the user to move in 3D space and use motion-tracked handheld controllers to interact with the environment

5.3 SteamVR

SteamVR is a plugin for Unity which allows Virtual Reality application developers to target one single API that all the popular Virtual Reality headsets can connect to. This allows the developer to develop and compile the application one

time, rather than doing it for every supported headset separately. It handles the input from the controllers, and the motion-tracking of the VR Headset and of the controllers and provides an interaction framework based on laser-pointing.

5.4 HoverUI Kit

HoverUI Kit is a framework that allows the developer to create Virtual Reality interfaces, based on the mechanism of hovering.

6 3D visualization of conceptual landscapes

6.1 Concept lattices

Concept lattices are represented in 3D by using a circular cone like view of the nodes which are at the same depth in the lattice. Concept lattices are computed with the NextClosure algorithm. Our application allows the parsing a formal context file exported from Concept Explorer, computing its formal concepts and representing it with a 3D concept lattice, as seen in Figure 1. This 3D concept lattice is calculated with the rules of a 2D concept lattice, but given an extra dimension we can extrapolate the z-coordinate of each node, making the nodes and connections more distinguishable from each other.

6.2 Many-valued contexts

The conceptual structure of a many-valued context is encoded in its conceptual schema, as it is usually done in the ToscanaJ system. This conceptual schema is a collection of conceptual scales which is then connected to a database in order to permit conceptual navigation. The same idea is transferred now in the VR environment. Many-valued contexts are scaled (either using ToscanaJ or FCA Tools Bundle) and the resulting concept lattice is represented in 3D in an VR environment. Elba from the ToscanaJ suite does not arrange the nodes automatically, resulting a diagram which is difficult to read and time consuming to rearrange, see Figure 2.

This problem is resolved by our VR application because we can extrapolate the nodes to use the extra z-dimension, leaving them aligned nicely for easy and clean visualization. The same list of strings as in Figure 2 was represented in a browsing scenario in our application, and the result can be seen in Figure 3.

6.3 Temporal Concept Analysis

Temporal FCA deals with data with a temporal layer and has as main aim visualization of this temporal dimension in a conceptual hierarchy. Life-tracks are visual representations of temporal modifications in conceptual landscapes. For that, we need to identify in the many-valued context the attribute representing the time. For this, we have to scale the time-representing attribute, resulting a

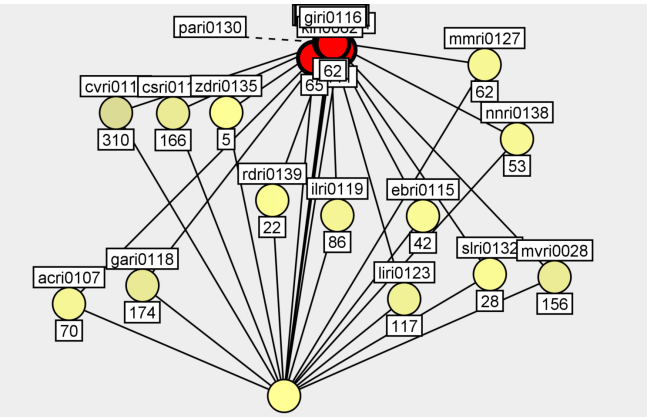


Fig. 2: Nominal scale used to represent a list of strings in ToscanaJ

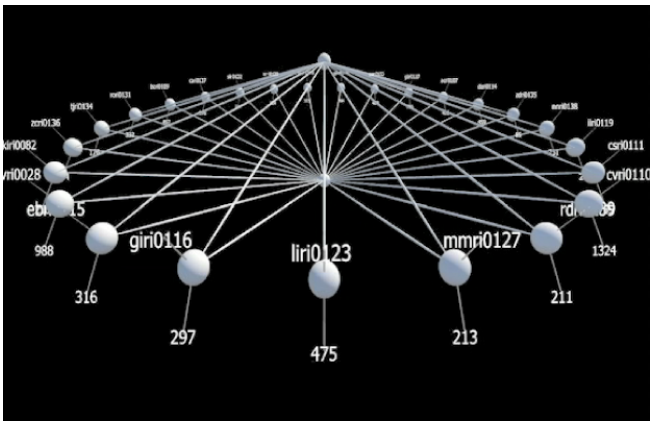


Fig. 3: Nominal Scale used to represent a list of strings in the application

temporal scale, which can be used to determine a subject's concept with respect to the temporal scale and a scale we want to search the position in.

An example of a life-track in a lattice can be seen in Figure 4, for which we took a database provided by our university [3] representing a log file of sites accessed by the students in one semester. The searched scale is representing the different materials provided by the instructor for the ninth week. The week in which the subject was can be read as the object, for example he did not visit any of these sites on week 1, 2, 3, 7, 10, 11 and 13. We can remark, for example, that in the fourteenth week the student accessed all of these sites, probably because the exam was coming soon.

Analyzing the life-tracks of more students, allows you to draw different conclusions about each student, for example seeing how stressed is the particular student in the exam-period, compared to a week from the teaching-period, or even to another students.

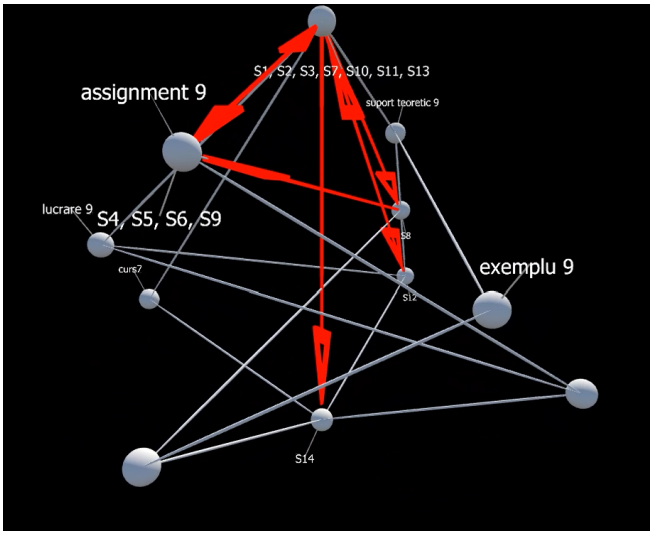


Fig. 4: Life-track lattice representing the sites accessed by a student

7 Exploration and navigation

The Input System of the application is based on the Command Pattern, which is illustrated in Figure 5.

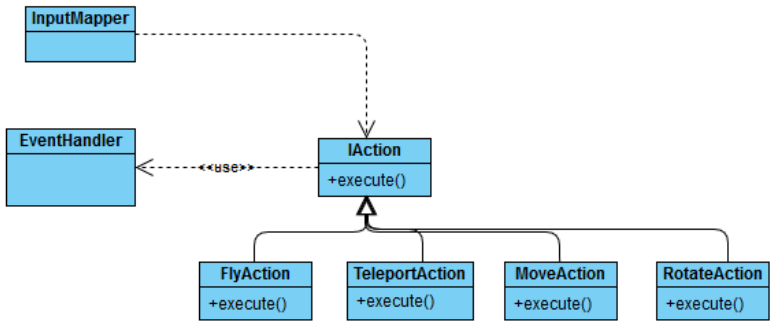


Fig. 5: Class Diagram of the Actions

Every controller has one action assigned to it, allowing the user to switch between actions using a menu built using HoverUI Kit. The menu (see Figure 6) uses the hover mechanism, which makes the whole interaction with menus for the user very intuitive.

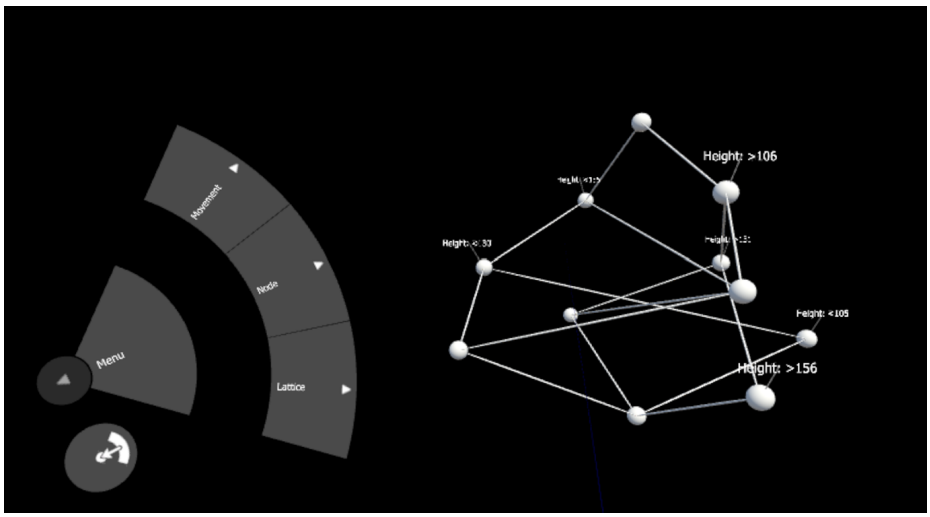


Fig. 6: Action Selection Menu

The actions are categorized, each being put under a category that describes what the action is affecting. The user can quickly switch off hand actions by using the menu selecting button. As seen in Figure 6, the following categories are available:

- Movement
  - Fly Action: When triggered, it moves the user into the direction that the controller is facing, see Figure 7.
  - Teleport Action: When triggered, it creates a pointer that the user can move around. When released it teleports the user to the pointed node.
  - Fly Speed: Slider that allows the user to choose the speed of the Fly Action.
- Node Move Action: When triggered, the selected node is following the position change of the trigger controller.
- Lattice Rotate Action: When triggered, it rotates the lattice around its center, with respect to the position of the trigger controller.
- Multi-player mode: A new multi-player mode was introduced, that leverages the sockets TCP connections to help users share insights and data in real-time. To be able to locate another user, a new service was introduced, where a user can introduce the IP of another user, basically allowing the users to connect to one another as a server-client. When a user is in the virtual room, with the new menu explained below, it has the option to make his virtual room public – allowing any other friend to find him and to connect to the room. The configuration of this service is required if the user wants to use the multi-player extension – if it's not configured, the option of Create or Join a server will be greyed out and if pressed an error notification is shown stating that the user has not configured the service.
  The real-time information sharing is achieved by allowing the users to speak with each other with the press of a button. Visually, real-time information

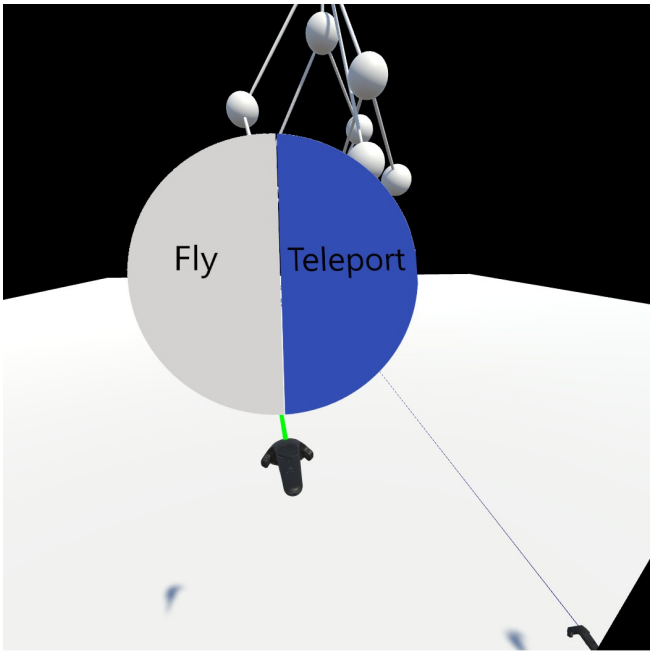


Fig. 7: Actions for a 3D concept lattice

sharing is achieved by allowing one user to highlight a node – which makes the node highlighted for every other user. Another way is to save one or more nodes in a list, which every user can access. While browsing life-track or many-valued browsing-scenarios, two users can visualize different lattices at one time, and if the users access one interesting node from the table, it will take the user to the same browsing scenario as when the original user has saved the node.

– Gesture-based input system: The old input system [6] was lacking what seems essential for any virtual reality application – the flexibility of the controls. For this reason, it was rethought as a gesture-based system, replacing the old Action Menu that the user had on each controller with gestures that the user can do with both controllers.
The new control system works as follows:
  - If only one controller back-trigger button is pressed, the user will rotate the lattice in the direction of the movement of his hand.
  - If both of the controllers have their back-trigger button pressed:
    * If the controllers move in the same direction, it will move the anchor at which the camera looks.
    * If the controllers move in opposite directions, if the controllers are moving away from each other it will zoom-out, else it will zoom-in.

  The old selection system was kept, allowing the users to point with a laser at a node they want to select and select it using the grip button, see Figure

8. If a user wants to move a node, while a node is selected the user can hold the touch-pad button pressed and the nodes position will mimic the changes of positions of the controller. The arrow size of the life-tracks was pushed into a new menu option.

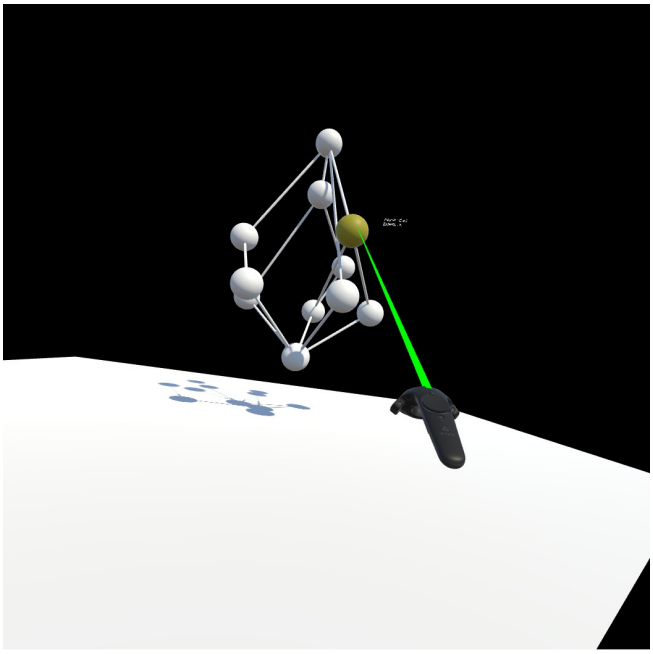


Fig. 8: Point and action

Another important characteristic of a virtual-reality environment is the feeling that the users really is in another room than he is physically - it is important to preserve that feeling by making the virtual environment react to users actions. For this, new haptic responses were added, and so if a user would do a valid action the controllers will now vibrate.

– Menus: New menus were introduced to help users to navigate through the application smoothly. A new main-menu was introduced that allows the user to select whether he wants to navigate through lattices. A new settings menu was added, allowing the user to select widgets that the user wants to see while in the virtual room - as for now, the user can select whether wants to see if the server is public or not. A new in-game menu was added, allowing the user to go back to the main-menu if it is already in a virtual room, or to make the server public, or to connect to a friends server (if the multi-player service was configured).

8 Conclusions

In this paper, we have described a design and implementation plan of a Virtual Reality tool that can handle formal and many-valued contexts, as well as contexts with a temporal layer. An early prototype has been presented at an FCA Tools workshop organized at the ICFCA 2019 [6]. Since then, the application has been improved constantly. A new multi-player feature was added, in order to support communication in VR rooms, the input system has been modified, as well as new user interfaces and menus.

This approach faced the constant challenge of representing order diagrams in 3D, and with the meanwhile classical problem of a "nice" concept lattice representation.

As future work, we plan to investigate new visualization algorithms. For instance, the use of genetic algorithms can be greatly improve the visualization of the lattice. We also intend to extend our research towards a so-called Navigation Assistant: Defining what the user is looking for, an AI assistant could be helping the user finding the desired node.

References

1. Alam, M., Le, T.N.N., Napoli, A.: Steps towards interactive formal concept analysis with latviz. In: Kuznetsov, S.O., Napoli, A., Rudolph, S. (eds.) Proceedings of the 5th International Workshop "What can FCA do for Artificial Intelligence"? co-located with the European Conference on Artificial Intelligence, FCA4AI@ECAI 2016, The Hague, the Netherlands, August 30, 2016. CEUR Workshop Proceedings, vol. 1703, pp. 51–62. CEUR-WS.org (2016)
2. Becker, P., Correia, J.H.: The ToscanaJ Suite for Implementing Conceptual Information Systems. In: Ganter, B., Stumme, G., Wille, R. (eds.) Formal Concept Analysis, Foundations and Applications. LNCS, vol. 3626, pp. 324–348. Springer (2005)
3. Dragos, S.: PULSE extended. In: Perry, M., Sasaki, H., Ehmann, M., Ortiz, G., Dini, O. (eds.) Fourth International Conference on Internet and Web Applications and Services, ICIW 2009, 24-28 May 2009, Venice/Mestre, Italy. pp. 510–515. IEEE Computer Society (2009), https://doi.org/10.1109/ICIW.2009.82
4. Kis, L.L., Sacarea, C., Troanca, D.: FCA tools bundle - A tool that enables dyadic and triadic conceptual navigation. In: Proceedings of the 5th International Workshop "What can FCA do for Artificial Intelligence"? co-located with the European Conference on Artificial Intelligence, FCA4AI@ECAI 2016, The Hague, the Netherlands, August 30, 2016. pp. 42–50 (2016)
5. S., P.: Employing immersive virtual environments for innovative experiments in health care communication. Patient Educ Couns.
6. Sacarea, C., Sotropa, D., Zavaczki, R.: Toscana goes 3d: Using VR to explore life tracks. In: Cristea, D., Ber, F.L., Missaoui, R., Kwuida, L., Sertkaya, B. (eds.) Supplementary Proceedings of ICFCA 2019 Conference and Workshops, Frankfurt, Germany, June 25-28, 2019. CEUR Workshop Proceedings, vol. 2378, pp. 82–87. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2378/shortAT10.pdf

7. Stevens SM, Goldsmith TE, S.K.e.a.: Virtual reality training improves students' knowledge structures of medical concepts. Stud Health Technol Inform. 111, 519–525 (2005)
8. Wille, R.: Conceptual Landscapes of Knowledge: A Pragmatic Paradigm for Knowledge Processing, pp. 344–356. Springer Berlin Heidelberg, Berlin, Heidelberg (1999)
9. Wille, R.: Restructuring lattice theory: An approach based on hierarchies of concepts. In: Rival, I. (ed.) Ordered Sets, NATO Advanced Study Institutes Series, vol. 83, pp. 445–470. Springer Netherlands (1982)
10. Wille, R.: Concept lattices and conceptual knowledge systems. Computers and Mathematics with Applications 23(6), 493 – 515 (1992)

Dihypergraph decomposition: application to closure system representations ⋆

Lhouari Nourine and Simon Vilmin

LIMOS, Université Clermont Auvergne, Aubière, France
simon.vilmin@ext.uca.fr, lhouari.nourine@uca.fr

Abstract. Closure systems and their representations are essential in numerous fields of computer science. Among representations, dihypergraphs (or attribute implications) and meet-irreducible elements (reduced context) are widely used in the literature. Translating between the two representations is known to be harder than hypergraph dualization, a well-known open problem. In this paper we are interested in enumerating the meet-irreducible elements of a closure system from a dihypergraph. To do so, we use a partitioning operation of a dihypergraph which gives a recursive characterization of its meet-irreducible elements. From this result, we deduce an algorithm which computes meet-irreducible elements in a divide-and-conquer way and puts the light on the major role of dualization in closure systems. Using hypergraph dualization, this strategy can be applied in output quasi-polynomial time to particular classes of dihypergraphs, improving at the same time previous results on ranked convex geometries.

Keywords: Dihypergraphs · Decomposition · Closure systems · Meet-irreducible elements

1 Introduction

Closure systems play a major role in several areas of computer science and mathematics such as database [9, 18, 19], Horn logic [16], lattice theory [6, 7] or Formal Concept Analysis (FCA) [12] where they are known as concept lattice.

Due to the exponential size of a closure system, several compact representations have been studied over the last decades [12, 14, 16, 20]. Among all possible representations, there are two prominent candidates: *implicational bases* and *meet-irreducible elements*. The former consists in set a of rules $B \to h$ over the ground set where B is the body and h the head of the rule. A rule depicts a causality relation between the elements of B and h, i.e., whenever a set contains B, it must also contain h. As several implicational bases can represent the same closure system, numerous bases with "good" properties have been studied. Among them, the Duquenne-Guigues base [13] being minimum or the canonical direct base [5] are worth mentioning. Like closure systems, implicational bases are ubiquitous in computer science. They appear for instance as Horn theories in propositional logic [16], attribute implications in FCA [12], functional dependencies in databases theory [9, 18] and they are conveniently expressed by

⋆ The second author is funded by the CNRS, France, ProFan project.

directed hypergraphs (*dihypergraphs* for short) [1,11] where an implication $B \rightarrow h$ corresponds to an arc (B, h). A nice survey on the topic can be found in [23].

The second representation for a closure system is a (minimum) subset of its elements from which it can be reconstructed. These elements are known as *meet-irreducible elements* [7]. In Horn logic, they are known as the characteristic models [16]. In FCA, they are written as a binary relation: the context [12]. They appear in the Armstrong relation [19] in database theory.

The problem of translating between these representations has been widely studied in the literature [2, 4, 8, 16, 19, 23]. Even though the two directions of the translation are equivalent [16], computing meet-irreducible elements from a set of implications has been less studied. This problem can be equivalently reformulated in FCA terms as follows: given a set of attribute implications, find an associated (reduced) context. Algorithms for this problem are used in databases to build relations satisfying a set of functional dependencies [19]. Furthermore, some tasks such as a abduction [16] are easier with meet-irreducible elements than implications. On the negative side, it has been shown in [16] that this problem is harder than enumerating minimal transversals of a hypergraph, also known as *hypergraph dualization* for which the best algorithm runs in output quasi-polynomial time [10]. Furthermore, Kavvadias et al. [15] have shown that enumerating maximal meet-irreducible elements cannot be done in output-polynomial time unless $\mathsf{P} = \mathsf{NP}$. On the positive side, exponential time algorithms have been given in [19, 22]. More recently, output quasi-polynomial time algorithms have been given for some classes of closure systems [4, 8].

In this paper we seek to push further the understanding of this problem, based on previous works such as [8, 17]. We use a hierarchical decomposition method introduced in [21] for dihypergraphs representing implicational bases. To achieve this decomposition we use a restricted version of a *split*, a partitioning operation of the ground set [21]. We call this restriction an *acyclic split*. An acyclic split of $\mathcal{H}$ is a bipartition of its ground set V into two non-trivial parts $\mathrm{V}_1, \mathrm{V}_2$ such that any arc (B, h) (i.e., any implication) is either fully contained in one of the two parts or the body B is in V_1 while the head h is in V_2. Intuitively, $\mathcal{H}$ is divided in three subhypergraphs, $\mathcal{H}[\mathrm{V}_1], \mathcal{H}[\mathrm{V}_2]$ and a *bipartite dihypergraph* $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ which models interactions from V_1 to V_2. Clearly, some dihypergraphs do not admit such splits. An acyclic split yields a decomposition of the underlying closure system into *projections* (or *traces*) and provide a recursive characterization of its meet-irreducible elements. Therefore, we propose an algorithm which compute meet-irreducible elements of a dihypergraph from a hierarchical decomposition using acyclic splits.

The paper is presented as follows. In Section 2 we recall definitions about directed hypergraphs and closure systems. Section 3 introduces acyclic split of a dihypergraph $\mathcal{H}$ and presents an example to illustrate our contribution. In Section 4 we study the construction of the underlying closure system and we give a characterization of its meet-irreducible elements. This characterization suggests a recursive algorithm which computes meet-irreducible elements of $\mathcal{H}$ in a divide-and-conquer way with acyclic splits, discussed in Section 5. We obtain

new classes of dihypergraphs for which computing meet-irreducible elements can be done in output quasi-polynomial time using hypergraph dualization, thus generalizing recent works on ranked convex geometries [8].

2 Preliminaries

All the objects considered in this paper are finite. If V is a set, 2^{V} denotes its powerset. For $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, \ldots, n\}$. Sometimes we will denote by $x_1 \ldots x_n$ the set $\{x_1, \ldots, x_n\}$.

We begin with notions on lattices and closure systems [6,7]. A *closure system* on V is a set system $\mathcal{F} \subseteq 2^{\mathrm{V}}$ such that $\mathrm{V} \in \mathcal{F}$ and for any $F_1, F_2 \in \mathcal{F}$, $F_1 \cap F_2 \in \mathcal{F}$. An element F of $\mathcal{F}$ is called a *closed set*. The number of closed sets in $\mathcal{F}$ represents its size, written $|\mathcal{F}|$. When ordered by set-inclusion, $(\mathcal{F}, \subseteq)$ is a *lattice*. Let $F \in \mathcal{F}$. The *ideal* of F, denoted $\downarrow F$ is the collection of closed sets of $\mathcal{F}$ included in F, namely $\downarrow F = \{F' \in \mathcal{F} \mid F' \subseteq F\}$. The *filter* $\uparrow F$ is defined dually. For a subset $\mathcal{B}$ of $\mathcal{F}$, we put $\downarrow \mathcal{B} = \bigcup_{F \in \mathcal{B}} \downarrow F$ and dually $\uparrow \mathcal{B} = \bigcup_{F \in \mathcal{B}} \uparrow F$. Let $F_1, F_2 \in \mathcal{F}$. We say that F_1 and F_2 are *incomparable* if $F_1 \not\subseteq F_2$ and $F_2 \not\subseteq F_1$. Assume $F_1 \subseteq F_2$. Then F_2 is a *cover* of F_1, written $F_1 \prec F_2$, if for any other $F' \in \mathcal{F}$, $F_1 \subseteq F' \subseteq F_2$ implies $F_1 = F'$ or $F_2 = F'$. A closed set M of $\mathcal{F}$ is a *meet-irreducible element* if for any $F_1, F_2 \in \mathcal{F}$, $M = F_1 \cap F_2$ implies $M = F_1$ or $M = F_2$. The ground set V is not a meet-irreducible element. Equivalently, M is a meet-irreducible element of $\mathcal{F}$ if and only if it has a unique cover. The set of meet-irreducible elements of $\mathcal{F}$ is written $\mathcal{M}(\mathcal{F})$ or simply $\mathcal{M}$ when clear from the context. A subset $\mathcal{B}$ of $\mathcal{F}$ is an *antichain* if elements of $\mathcal{B}$ are pairwise incomparable. Let $U \subseteq \mathrm{V}$. The *trace* (or *projection*) of $\mathcal{F}$ on U, denoted $\mathcal{F}\colon U$, is obtained by intersecting each closed set of $\mathcal{F}$ with U, i.e., $\mathcal{F}\colon U = \{F \cap U \mid F \in \mathcal{F}\}$. If $\mathcal{F}' \subseteq \mathcal{F}$ is a closure system, it is a *meet-sublattice* of $\mathcal{F}$. Let $\mathcal{F}_1, \mathcal{F}_2$ be two closure systems on disjoint $\mathrm{V}_1, \mathrm{V}_2$ respectively. The *direct product* of $\mathcal{F}_1$ and $\mathcal{F}_2$, denoted $\mathcal{F}_1 \times \mathcal{F}_2$, is given by $\mathcal{F}_1 \times \mathcal{F}_2 = \{F_1 \cup F_2 \mid F_1 \in \mathcal{F}_1, F_2 \in \mathcal{F}_2\}$.

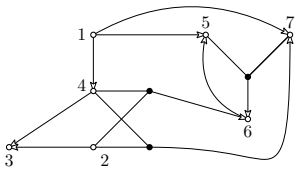
In this paper, we suppose that implicational bases are given as directed hypergraphs. Directed hypergraphs are a convenient representation for attribute implications of FCA, Horn clauses, functional dependencies [1,11,23]. We mainly refer to papers [1, 11] for definitions of dihypergraphs. A *(directed) hypergraph* (*dihypergraph* for short) $\mathcal{H}$ is a pair $(\mathrm{V}(\mathcal{H}), \mathcal{E}(\mathcal{H}))$ where $\mathrm{V}(\mathcal{H})$ is its set of vertices, and $\mathcal{E}(\mathcal{H}) = \{e_1, \ldots, e_n\}$, $n \in \mathbb{N}$, its set of *arcs*. An arc $e \in \mathcal{E}(\mathcal{H})$ is a pair $(B(e), h(e))$, where $B(e)$ is a non-empty subset of V called the *body* of e and $h(e) \in \mathrm{V} \setminus B$ called the *head* of e. When clear from the context, we write V, $\mathcal{E}$ and (B, h) instead of $\mathrm{V}(\mathcal{H})$, $\mathcal{E}(\mathcal{H})$ and $(B(e), h(e))$ respectively. An arc $e = (B, h)$ is written as the set $e = B \cup \{h\}$ when no confusion can arise. Whenever a body B is reduced to a single vertex b, we shall write (b, h) instead of $(\{b\}, h)$ for clarity. In this case, the arc (b, h) is called a *unit arc*. A dihypergraph where all edges are unit is a digraph. Let $\mathcal{H} = (\mathrm{V}, \mathcal{E})$ be a dihypergraph and $U \subseteq \mathrm{V}$. The subhypergraph $\mathcal{H}[U]$ *induced* by U is the pair $(U, \mathcal{E}(\mathcal{H}[U]))$ where $\mathcal{E}(\mathcal{H}[U])$ is the set of arcs of $\mathcal{E}$ contained in U, namely $\mathcal{E}(\mathcal{H}[U]) = \{e \in \mathcal{E} \mid e \subseteq U\}$. A *bipartite dihypergraph* is a dihypergraph in which the ground set can be partitioned into two parts $(\mathrm{V}_1, \mathrm{V}_2)$ such that for any $(B, h) \in \mathcal{E}$, $B \subseteq \mathrm{V}_1$ or $B \subseteq \mathrm{V}_2$. We denote a bipartite dihypergraph by $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$. A *split* [21] of a dihypergraph

$\mathcal{H}$ is a non-trivial bipartition (V_1, V_2) of V such that for any arc (B, h) of $\mathcal{H}$, either $B \subseteq V_1$ or $B \subseteq V_2$. A split (V_1, V_2) partitions $\mathcal{H}$ into three arc disjoint subhypergraphs $\mathcal{H}[V_1], \mathcal{H}[V_2]$ and a bipartite dihypergraph $\mathcal{H}[V_1, V_2]$.

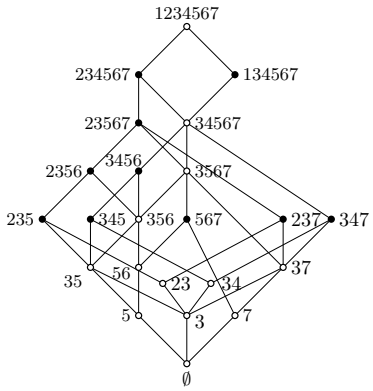
The closure system associated to a dihypergraph $\mathcal{H}$ is obtained with the *forward chaining algorithm*. It starts from a subset X of V and constructs a chain $X = X_0 \subseteq X_1 \subseteq \cdots \subseteq X_k = X^{\mathcal{H}}$ such that for any $i = 1, \ldots, k$ we have $X_i = X_{i-1} \cup \{h \mid \exists (B, h) \in \mathcal{E} \text{ s.t. } B \subseteq X_{i-1}\}$. The operation $(.)^{\mathcal{H}}$ is a *closure operator*, that is for any $X, Y \subseteq V$, we have $X \subseteq X^{\mathcal{H}}$, $X \subseteq Y \implies X^{\mathcal{H}} \subseteq Y^{\mathcal{H}}$ and $(X^{\mathcal{H}})^{\mathcal{H}} = X^{\mathcal{H}}$. A set X is closed if $X = X^{\mathcal{H}}$. Note that X is closed for $\mathcal{H}$ if and only if for any arc $(B, h) \in \mathcal{E}$, $B \subseteq X$ implies $h \in X$. We say that X *satisfies* an arc (B, h) if $B \subseteq X \implies h \in X$. The collection $\mathcal{F}(\mathcal{H}) = \{X^{\mathcal{H}} \mid X \subseteq V\}$ of closed sets of $\mathcal{H}$ is a closure system. For clarity, we may write $\mathcal{F}$ instead of $\mathcal{F}(\mathcal{H})$. Our definition of a dihypergraph implies $\emptyset \in \mathcal{F}$, without loss of generality.

3 Acyclic split and illustration on an example

In this section we introduce acyclic splits and we illustrate our approach to compute meet-irreducible elements from a dihypergraph on a toy example. Let $V = [7]$ and $\mathcal{H} = (V, \{(2,3), (4,3), (6,5), (57,6), (24,6), (24,7), (1,4), (1,5), (1,7)\})$. It is represented in Figure 1 (a). To represent an arc (B, h) with $|B| \geq 2$ we use a black vertex connecting every elements of B from which starts an arrow towards h. The closure system $\mathcal{F}$ associated to $\mathcal{H}$ is given in Figure 1 (b).



(a) A dihypergraph $\mathcal{H}$



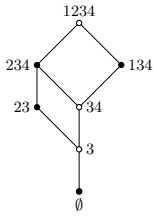
(b) The closure system $\mathcal{F}$ associated to $\mathcal{H}$, where meet-irreducible elements are darkened.

Fig. 1: The dihypergraph $\mathcal{H}$ and its closure system $\mathcal{F}$

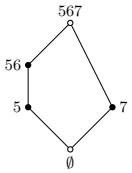
The idea is to split $\mathcal{H}$ into three subhypergraphs $\mathcal{H}[V_1], \mathcal{H}[V_2]$ and $\mathcal{H}[V_1, V_2]$ as in [21]. However we use a restricted version of a split we call an *acyclic split*. A split is acyclic if for any arc (B, h) of $\mathcal{H}[V_1, V_2]$, $B \subseteq V_1$ and $h \in V_2$. A dihypergraph which does not have any acyclic split is *indecomposable*. A maximum subhypergraph of $\mathcal{H}$ which has no acyclic split is a *c-factor* (cyclic

factor) of $\mathcal{H}$. If a c-factor $\mathcal{H}'$ of $\mathcal{H}$ is reduced to a vertex, i.e., $\mathcal{H}' = (\{x\}, \emptyset)$, it is a *singleton c-factor* of $\mathcal{H}$.

For instance in $\mathcal{H}$, the bipartition $\mathrm{V}_1 = \{1, 2, 3\}$, $\mathrm{V}_2 = \{4, 5, 6, 7\}$ is not a split because the body of $(24, 6)$ has elements from both V_1 and V_2. If we fix $\mathrm{V}_1 = \{1, 2, 4, 6\}$ and $\mathrm{V}_2 = \{3, 5, 7\}$, then the bipartition is a split but not acyclic since the arc $(6, 5)$ goes from V_1 to V_2 and $(57, 6)$ from V_2 to V_1. An acyclic split is $\mathrm{V}_1 = \{1, 2, 3, 4\}$ and $\mathrm{V}_2 = \{5, 6, 7\}$. It induces the three subhypergraphs $\mathcal{H}[\mathrm{V}_1] = (\mathrm{V}_1, \{(4, 3), (1, 4), (2, 3)\})$, $\mathcal{H}[\mathrm{V}_2] = (\mathrm{V}_2, \{(6, 5), (57, 6)\}$ and $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2] = (\mathrm{V}, \{(24, 6), (24, 7), (1, 5), (1, 7)\})$. Observe that $\mathcal{H}[\mathrm{V}_2]$ is indecomposable: the unique split of V_1 is $\mathrm{V}'_1 = \{5, 7\}$ and $\mathrm{V}'_2 = \{6\}$, which is not acyclic. Hence, $\mathcal{H}[\mathrm{V}_2]$ is a c-factor of $\mathcal{H}$. Closure systems $\mathcal{F}_1$, $\mathcal{F}_2$ of $\mathcal{H}[\mathrm{V}_1]$ and $\mathcal{H}[\mathrm{V}_2]$ are given in Figure 2. Note that $\mathcal{F}$ is a meet-sublattice of $\mathcal{F}_1 \times \mathcal{F}_2$.



(a) $\mathcal{F}_1$, the closure lattice of $\mathcal{H}[\mathrm{V}_1]$



(b) $\mathcal{F}_2$, the closure lattice of $\mathcal{H}[\mathrm{V}_2]$

Fig. 2: Closure lattices of $\mathcal{H}[\mathrm{V}_1]$ and $\mathcal{H}[\mathrm{V}_2]$, meet-irreducible are darkened.

The splitting operation provides a partition of $\mathcal{M}(\mathcal{F})$ into two classes. The first class contains meet-irreducible elements of $\mathcal{F}_1$ to which we added V_2. This is the case for example of 234567 and 567, which are the meet-irreducible elements 234 and $\emptyset$ of $\mathcal{F}_1$. The second class contains meet-irreducible which are inclusion-wise maximal closed sets of $\mathcal{F}$ whose trace on V_2 is meet-irreducible in $\mathcal{F}_2$. For instance, 235 and 345 are inclusion-wise maximal closed sets of $\mathcal{F}$ whose intersection with V_2 rise 5, a meet-irreducible element of $\mathcal{F}_2$.

Thus, every meet-irreducible element of $\mathcal{F}$ belongs to exactly one of these two classes. Observe that any other $F \in \mathcal{F}$ cannot be part of $\mathcal{M}(\mathcal{F})$. As $\mathcal{F} \subseteq \mathcal{F}_1 \times \mathcal{F}_2$, every $M \in \mathcal{M}$ arise from the combination of some $F_1 \in \mathcal{F}_1$ and $F_2 \in \mathcal{F}_2$. Let $F \in \mathcal{F}$ be outside of those two class. If $\mathrm{V}_2 \subseteq F$, then $F \cap \mathrm{V}_1$ cannot be meet-irreducible in $\mathcal{F}_1$. In this case, covers of $F \cap \mathrm{V}_1$ in $\mathcal{F}_1$ can be used to produce distinct covers of F in $\mathcal{F}$. If however $\mathrm{V}_2 \not\subseteq F$, then covers of $F \cap \mathrm{V}_2$ in $\mathcal{F}_2$ yield covers of F in $\mathcal{F}$. In the case where $F \cap \mathrm{V}_2$ is meet-irreducible in $\mathcal{F}_2$, there will be a closed set F_1 in $\mathcal{F}_1$ such that $F \cap \mathrm{V}_1 \subset F_1$ and $F_1 \cup (F \cap \mathrm{V}_2)$ will be closed in $\mathcal{F}$ by assumption. This can be used to find another cover of F in $\mathcal{F}$.

This characterization suggests to recursively find meet-irreducible elements of $\mathcal{F}$. If $\mathcal{H}$ is indecomposable, we computes $\mathcal{M}$ with known algorithms [4, 19]. Otherwise, we find an acyclic split $(\mathrm{V}_1, \mathrm{V}_2)$ and recursively applies on $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$. Then, we compute $\mathcal{M}$ using $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$, $\mathcal{M}_1$ and $\mathcal{M}_2$. In Figure 3, we give the trace of a decomposition for $\mathcal{H}$ using acyclic splits. This strategy is particularly interesting for cases where c-factors of $\mathcal{H}$ are all of the form $(\{x\}, \emptyset)$ for $x \in \mathrm{V}$, since the unique meet-irreducible element in this case is $\emptyset$.

Thus, the steps we will follow are the following. Given a dihypergraph $\mathcal{H}$ and its closure system $\mathcal{F}$, we will study the construction of $\mathcal{F}$ with respect to an acyclic split. This will lead us to a characterization of $\mathcal{M}$. Recursively applying this characterization, we will get an algorithm to compute $\mathcal{M}$ from $\mathcal{H}$.

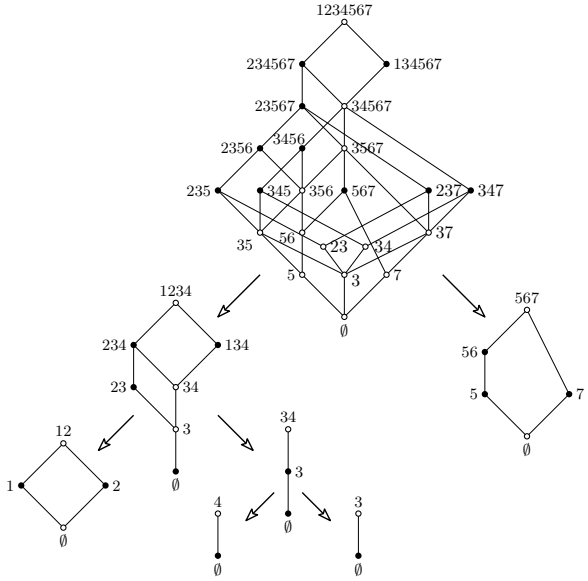


Fig. 3: Hierarchical decomposition of $\mathcal{F}$ using acyclic splits. Meet-irreducible elements are darkened.

4 The closure system induced by an acyclic split

In this section, we show the construction of a closure system with respect to an acyclic split. We give a characterization of its closed sets and meet-irreducible elements $\mathcal{M}$. Let $\mathcal{H} = (\mathrm{V}, \mathcal{E})$ be a dihypergraph and $(\mathrm{V}_1, \mathrm{V}_2)$ an acyclic split of $\mathcal{H}$. Let $\mathcal{F}_1, \mathcal{F}_2$ be the closure systems associated to $\mathcal{H}[\mathrm{V}_1]$ and $\mathcal{H}[\mathrm{V}_2]$ respectively. Similarly, $\mathcal{M}_1, \mathcal{M}_2$ are their meet-irreducible elements. We show how to construct $\mathcal{F}$ from $\mathcal{F}_1, \mathcal{F}_2$ and $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$. We begin with the following theorem from [21]:

Theorem 1 (Theorem 3 of [21]). *Let $(\mathrm{V}_1, \mathrm{V}_2)$ be a split of $\mathcal{H}$, $\mathcal{F}_1$ and $\mathcal{F}_2$ the closure systems corresponding to $\mathcal{H}[\mathrm{V}_1]$ and $\mathcal{H}[\mathrm{V}_2]$ respectively. Then,*

1. *If $F \in \mathcal{F}_\mathcal{H}$ then $F_i = F \cap \mathrm{V}_i \in \mathcal{F}_i, i = \{1, 2\}$. Moreover, $\mathcal{F}_\mathcal{H} \subseteq \mathcal{F}_1 \times \mathcal{F}_2$.*
2. *If $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ has no arc then $\mathcal{F}_\mathcal{H} = \mathcal{F}_1 \times \mathcal{F}_2$.*
3. *If $B \subseteq \mathrm{V}_1$ for any arc (B, h) of $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$, then $\mathcal{F}_\mathcal{H} : \mathrm{V}_i = \mathcal{F}_i$ for $i \in \{1, 2\}$.*
4. *If $B \subseteq \mathrm{V}_2$ for any arc (B, h) of $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$, then $\mathcal{F}_\mathcal{H} : \mathrm{V}_i = \mathcal{F}_i$ for $i \in \{1, 2\}$.*

The first item states that $\mathcal{F}$ is a meet-sublattice of $\mathcal{F}$. From item *2* we can derive a characterization of meet-irreducible elements of the direct product $\mathcal{F}_1 \times \mathcal{F}_2$. This result has already been formulated in lattice theory, for instance in [7]. We reprove it in our framework for self-containment.

Proposition 1. *Let $\mathcal{H}$ be a dihypergraph an $(\mathrm{V}_1, \mathrm{V}_2)$ an acyclic split of $\mathcal{H}$ where $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ has no arcs. Then $\mathcal{M} = \{M_1 \cup \mathrm{V}_2 \mid M_1 \in \mathcal{M}_1\} \cup \{M_2 \cup \mathrm{V}_1 \mid M_2 \in \mathcal{M}_2\}$.*

Proof. Let $M \in \mathcal{M}$, $i \in \{1, 2\}$ and $M_i = M \cap \mathrm{V}_i$. As $M \neq \mathrm{V}$, $\mathrm{V}_i \not\subseteq M$ for at least one of $i \in \{1, 2\}$. Suppose it holds for V_1 and V_2. Then, there exists $M_i' \in \mathcal{F}_i$, such that $M_i \prec M_i'$ in $\mathcal{F}_i$. However, by Theorem 1, $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$. Hence $M_1 \cup M_2'$ and $M_1' \cup M_2$ belong to $\mathcal{F}$. Furthermore they are incomparable and we have $M \prec M_1 \cup M_2'$ and $M \prec M_1' \cup M_2$ which contradicts $M \in \mathcal{M}$. Therefore, either $\mathrm{V}_1 \subseteq M$ or $\mathrm{V}_2 \subseteq M$. Assume without loss of generality that $\mathrm{V}_1 \subseteq M$. Let M'' be the unique cover of M in $\mathcal{F}$. Then, $\mathrm{V}_1 \subseteq M''$ and it follows that $M_2 \prec M'' \cap \mathrm{V}_2$ in $\mathcal{F}_2$. As M'' is the unique cover of M in $\mathcal{F}$, we conclude that $M'' \cap \mathrm{V}_2$ is the unique cover of M_2 in $\mathcal{F}_2$ and $M_2 \in \mathcal{F}_2$.

Let $M_1 \in \mathcal{M}_1$ and consider $M_1 \cup \mathrm{V}_2 \in \mathcal{F}_2$. Let M_1' be the unique cover of M_1 in $\mathcal{F}_1$. As $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$ by Theorem 1, we have that $M_1 \cup \mathrm{V}_2 \prec M_1' \cup \mathrm{V}_2$ is in $\mathcal{F}$. Let F be any closed set such that $M_1 \cup \mathrm{V}_2 \subset F$. We have $F \cap \mathrm{V}_2 = \mathrm{V}_2$ and hence $M_1 \subset F \cap \mathrm{V}_1$. Since $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2$, we get $F \cap \mathrm{V}_1 \in \mathcal{F}_1$. As $M_1 \prec M_1'$ in $\mathcal{F}_1$ and $M_1 \in \mathcal{M}_1$, we conclude that $M_1' \subseteq F \cap \mathrm{V}_1$ and hence that $M_1' \cap \mathrm{V}_2 \subseteq F$. Therefore, $M_1 \cup \mathrm{V}_2 \in \mathcal{M}$. Similarly we obtain $M_2 \cup \mathrm{V}_1 \in \mathcal{M}$, for $M_2 \in \mathcal{M}_2$. □

Item *3* of Theorem 1 considers the case where the split is acyclic (as item *4*). In particular, the proof of Theorem 1 shows that $\mathcal{F}_2 \subseteq \mathcal{F}$ in this case. Since $\mathcal{F}$ is a meet-sublattice of $\mathcal{F}_1 \times \mathcal{F}_2$, and both $\mathcal{F}_1$, $\mathcal{F}_2$ appear as traces of $\mathcal{F}$, we have that $|\mathcal{F}| \geq |\mathcal{F}_1|$ and $|\mathcal{F}| \geq |\mathcal{F}_2|$. As $\mathcal{F}_2 \subseteq \mathcal{F}$, for each $F_2 \in \mathcal{F}_2$, it may exists several closed sets of $\mathcal{F}_1$ which extend F_2 to another element of $\mathcal{F}$.

Definition 1. *Let $\mathcal{H}$ be a dihypergraph with acyclic split $(\mathrm{V}_1, \mathrm{V}_2)$. Let $F_1 \in \mathcal{F}_1$, $F_2 \in \mathcal{F}_2$. We say that $F_1 \cup F_2$ is an* extension *of F_2 if it belongs to $\mathcal{F}$. We denote by $\mathsf{Ext}(F_2)$ the set of extensions of F_2, namely $\mathsf{Ext}(F_2) = \{F \in \mathcal{F} \mid F \cap \mathrm{V}_2 = F_2\}$.*

We denote by $\mathsf{Ext}(F_2)$: V_1 the set of closed sets of $\mathcal{F}_1$ which make extensions of F_2. Hence, any closed set F of $\mathcal{F}$ can be seen as the extension of some $F_2 \in \mathcal{F}_2$ so that $\mathcal{F}$ results from the union of extensions of closed sets in $\mathcal{F}_2$:

$$\mathcal{F} = \bigcup_{F_2 \in \mathcal{F}_2} \mathsf{Ext}(F_2)$$

Extensions of $F_2 \in \mathcal{F}_2$ can be characterized using $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ as follows.

Lemma 1. *Let $F_1 \in \mathcal{F}_1$, $F_2 \in \mathcal{F}_2$. Then $F_1 \cup F_2$ is an extension of F_2 if and only if for any arc (B, h) in $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$, $B \subseteq F_1$ implies $h \in F_2$.*

Proof. We begin with the only if part. Let F_1 be a closed set of $\mathcal{F}_1$ such that $F_1 \cup F_2$ is an extension of F_2 and let $(B, h) \in \mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$. If $B \subseteq F_1$, then it must be that $h \in F_2$ since otherwise we would contradict $F_1 \cup F_2 \in \mathcal{F}$.

We move to the if part. Let F_1 be a closed set of $\mathcal{F}_1$ and F_2 a closed set of $\mathcal{F}_2$ such that for any arc $(B, h) \in \mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$, $B \subseteq F_1$ implies $h \in F_2$. We have to show that $F_1 \cup F_2$ is closed. Let (B, h) be an arc of $\mathcal{H}$. As $(\mathrm{V}_1, \mathrm{V}_2)$ is an acyclic split of V, we have two cases for (B, h): either (B, h) is in $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ or it is not. In the second case, assume it is in $\mathcal{H}[\mathrm{V}_1]$. As $B \subseteq F_1 \cup F_2$, we have

$B \subseteq F_1$. Furthermore, F_1 is closed for $\mathcal{H}[\mathrm{V}_1]$. Hence, $h \in F_1 \subseteq F_1 \cup F_2$. The same reasoning can be applied if (B, h) is in $\mathcal{H}[\mathrm{V}_2]$. Now assume (B, h) is in $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$. We have that $B \subseteq \mathrm{V}_1$ by definition of an acyclic split. In particular we have $B \subseteq F_1$ which entails $h \in F_2$ by assumption. In any case, $F_1 \cup F_2$ already contains h for any arc (B, h) such that $B \subseteq F_1 \cup F_2$ and $F_1 \cup F_2$ is closed. □

Observe that for the particular case $F_2 = \mathrm{V}_2$, we have $\mathsf{Ext}(\mathrm{V}_2)\colon \mathrm{V}_1 = \mathcal{F}_1$ because any arc (B, h) of $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ satisfies $h \in \mathrm{V}_2$. A consequence of Lemma 1 is that the extension is hereditary, as stated by the following lemma.

Lemma 2. *Let $F_1 \in \mathcal{F}_1$, $F_2 \in \mathcal{F}_2$. If $F_1 \cup F_2$ is an extension of F_2, then for any closed set F_1' of $\mathcal{F}_1$ such that $F_1' \subseteq F_1$, $F_1' \cup F_2$ is also an extension of F_2.*

Proof. Let $F_1 \in \mathcal{F}_1, F_2 \in \mathcal{F}_2$ such that $F_1 \cup F_2 \in \mathcal{F}$. Let $F_1' \in \mathcal{F}_1$ such that $F_1' \subseteq F_1$. As $F_1 \cup F_2$ is an extension of F_2, for any arc (B, h) of $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ such that $B \subseteq F_1$, we have $h \in F_2$ by Lemma 1. Since $F_1' \subseteq F_1$, this condition holds in particular for any arc (B, h) of $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ such that $B \subseteq F_1' \subseteq F_1$. Applying Lemma 1, we have that $F_1' \cup F_2$ is closed. □

As $\mathcal{F}$ is a meet-sublattice of $\mathcal{F}_1 \times \mathcal{F}_2$ by Theorem 1, it follows from Lemma 2 that for any $F_2 \in \mathcal{F}_2$, the set $\mathsf{Ext}(F_2)\colon \mathrm{V}_1$ is an ideal of $\mathcal{F}_1$. Thus, it is uniquely determined by its maximal elements. They are inclusion-wise maximal closed sets of $\mathcal{F}_1$ satisfying the condition of Lemma 1.

Example 1. We consider the introductory example $\mathcal{H}$ and the acyclic split $\mathrm{V}_1 = \{1, 2, 3, 4\}$ and $\mathrm{V}_2 = \{5, 6, 7\}$. We have for instance $\mathsf{Ext}(7) = \{7, 37, 237, 347\}$ which corresponds to the ideal $\{\emptyset, 3, 23, 24\}$ of $\mathcal{F}_1$ illustrated on the left of Figure 2 representing $\mathcal{F}_1$.

Now we are interested in the characterization of meet-irreducible elements $\mathcal{M}$ of $\mathcal{F}$. The strategy is to identify for each $F_2 \in \mathcal{F}_2$, which closed sets of $\mathsf{Ext}(F_2)$ are meet-irreducible elements of $\mathcal{F}$.

Proposition 2. *Let $F = F_1 \cup F_2 \in \mathcal{F}$. Let $F_2' \in \mathcal{F}_2$ such that $F_2 \prec F_2'$. Then $F_2' \cup F_1$ is closed in $\mathcal{F}$ and $F \prec F_2' \cup F_1$ in $\mathcal{F}$.*

Proof. Let $F = F_1 \cup F_2 \in \mathcal{F}$. Let $F_2' \in \mathcal{F}_2$ such that $F_2 \prec F_2'$. As $F_1 \cup F_2$ is an extension of F_2, for every arc (B, h) of $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ such that $B \subseteq F_1$, we have $h \in F_2 \subseteq F_2'$ by Lemma 1. Therefore, $F_1 \cup F_2'$ is an extension of F_2'.

Now we show that $F_1 \cup F_2'$ is a cover of F. Let $F'' \in \mathcal{F}$ such that $F \subseteq F'' \subseteq F_1 \cup F_2'$. As $F \cap \mathrm{V}_1 = F_1 = (F_1 \cup F_2') \cap \mathrm{V}_1$, we have that $F'' \cap \mathrm{V}_1 = F_1$. Recall from Theorem 1 that $\mathcal{F} \subseteq \mathcal{F}_1 \times \mathcal{F}_2$. Therefore $F'' \cap \mathrm{V}_2$ is a closed set of $\mathcal{F}_2$ and $F_2 \subseteq F'' \cap \mathrm{V}_2 \subseteq F_2'$. As $F_2 \prec F_2'$ in $\mathcal{F}_2$, we have either $F_2 = F'' \cap \mathrm{V}_2$ or $F_2' = F'' \cap \mathrm{V}_2$. Consequently, $F'' = F$ or $F'' = F_1 \cup F_2'$ which entails $F \prec F_1 \cup F_2'$ in $\mathcal{F}$, concluding the proof. □

A consequence of this proposition is that for any $F_2, F_2' \in \mathcal{F}_2$ such that $F_2 \subseteq F_2'$, one has $\mathsf{Ext}(F_2)\colon \mathrm{V}_1 \subseteq \mathsf{Ext}(F_2')\colon \mathrm{V}_1$. In particular, if $F_2 \prec F_2'$ in $\mathcal{F}_2$, then each extension F of F_2 is covered by the unique extension F' of F_2' such that $F \cap \mathrm{V}_1 = F' \cap \mathrm{V}_1$. This leads us to the following lemmas.

Lemma 3. *Let $F_2 \in \mathcal{F}_2, F_2 \neq \mathrm{V}_2$ and $F_1 \in \mathcal{F}_1$ such that $F_1 \cup F_2$ is a non-maximal extension of F_2. Then $F_1 \cup F_2 \notin \mathcal{M}$.*

Proof. Let $F_2 \in \mathcal{F}_2, F_2 \neq \mathrm{V}_2$ and $F_1 \in \mathcal{F}_1$ such that $F_1 \cup F_2$ is a non-maximal extension of F_2. As $F_2 \neq \mathrm{V}_2$, there exists at least one closed set $F_2' \in \mathcal{F}_2$ such that $F_2 \prec F_2'$. By Proposition 2 we have that $F_1 \cup F_2 \prec F_1 \cup F_2'$ in $\mathcal{F}$. Furthermore, $F_1 \cup F_2$ is not a maximal extension of F_2. Therefore, there exists a closed set F_1' in $\mathcal{F}_1$ such that $F_1 \prec F_1'$ and $F_1' \cup F_2 \in \mathcal{F}$. As $\mathcal{F} \subseteq \mathcal{F}_1 \times \mathcal{F}_2$ by Theorem 1 and extension is hereditary by Lemma 2, it follows that $F_1 \cup F_2 \prec F_1' \cup F_2$ in $\mathcal{F}$ with $F_1 \cup F_2' \neq F_1' \cup F_2$. Therefore $F_1 \cup F_2$ is not a meet-irreducible element of $\mathcal{F}$. □

Lemma 4. *Let $F_2 \in \mathcal{F}_2$ such that $F_2 \neq \mathrm{V}_2$ and $F_2 \notin \mathcal{M}_2$. Then $F \notin \mathcal{M}$ for any $F \in \mathsf{Ext}(F_2)$.*

Proof. Let $F_2 \in \mathcal{F}_2$ such that $F_2 \neq \mathrm{V}_2$ and $F_2 \notin \mathcal{M}_2$. Let $F \in \mathsf{Ext}(F_2)$ and $F_1 = F \cap \mathrm{V}_1$. As $F_2 \notin \mathcal{M}_2$, it has at least two covers F_2', F_2'' in $\mathcal{F}_2$. By Proposition 2, it follows that both $F_2' \cup F_1$ and $F_2'' \cup F_1$ are covers of F in $\mathcal{F}$. Hence $F \notin \mathcal{M}$. □

These lemmas suggest that meet-irreducible elements of $\mathcal{F}$ arise from maximal extensions of meet-irreducible elements of $\mathcal{F}_2$. They might also come from meet-irreducible extensions of V_2 since $\mathsf{Ext}(\mathrm{V}_2)\colon \mathrm{V}_1 = \mathcal{F}_1$. As V_2 has no cover in $\mathcal{F}_2$, Proposition 2 cannot apply. These ideas are proved in the following theorem which characterize meet-irreducible elements $\mathcal{M}$ of $\mathcal{F}$.

Theorem 2. *Let $\mathcal{H} = (\mathrm{V}, \mathcal{E})$ be a dihypergraph with an acyclic split $(\mathrm{V}_1, \mathrm{V}_2)$. Meet-irreducible elements $\mathcal{M}$ of $\mathcal{F}$ are given by the following equality:*

$$\mathcal{M} = \{M_1 \cup \mathrm{V}_2 \mid M_1 \in \mathcal{M}_1\} \cup \{F \in \mathsf{max}_{\subseteq}(\mathsf{Ext}(M_2)) \mid M_2 \in \mathcal{M}_2\}$$

Proof. First we show that $\{M_1 \cup \mathrm{V}_2 \mid M_1 \in \mathcal{M}_1\} \subseteq \mathcal{M}$. Let $M_1 \in \mathcal{M}_1$. By Lemma 1, we have that $M_1 \cup \mathrm{V}_2 \in \mathcal{F}$, as $h \in \mathrm{V}_2$ for any (B, h) in $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$. Let F', F'' be two covers of $M_1 \cup \mathrm{V}_2$ in $\mathcal{F}$. First, observe that F' and F'' differ from $M_1 \cup \mathrm{V}_2$ only in V_1 as they both contain V_2. By Theorem 1, $\mathcal{F} \subseteq \mathcal{F}_1 \times \mathcal{F}_2$, so $F' \cap \mathrm{V}_1$ and $F'' \cap \mathrm{V}_1$ are closed sets of $\mathcal{F}_1$. Furthermore $\mathsf{Ext}(\mathrm{V}_2)\colon \mathrm{V}_1 = \mathcal{F}_1$ by Lemmas 2 and 1. Therefore, both $F' \cap \mathrm{V}_1$ and $F'' \cap \mathrm{V}_1$ cover M_1 in $\mathcal{F}_1$. Since M_1 is a meet-irreducible element of $\mathcal{F}_1$, we conclude that $F' = F''$ and $M_1 \cup \mathrm{V}_2 \in \mathcal{M}$.

Next, we prove that $\{F \in \mathsf{max}_{\subseteq}(\mathsf{Ext}(M_2)) \mid M_2 \in \mathcal{M}_2\} \subseteq \mathcal{M}$. Let $M_2 \in \mathcal{M}_2$ and $F \in \mathsf{max}_{\subseteq}(\mathsf{Ext}(M_2))$ with $F = F_1 \cup M_2$. Since $M_2 \in \mathcal{F}_2$, it has a unique cover M_2' in $\mathcal{F}_2$. By Proposition 2, we get $F \prec M_2' \cup F_1$ in $\mathcal{F}$. Let $F'' \in \mathcal{F}$ such that $F \subset F''$. Recall that $\mathcal{F} \subseteq \mathcal{F}_1 \times \mathcal{F}_2$ by Theorem 1, so that $F'' \cap \mathrm{V}_1 \in \mathcal{F}_1$ and $F'' \cap \mathrm{V}_2 \in \mathcal{F}_2$. Furthermore, $F \in \mathsf{max}_{\subseteq}(\mathsf{Ext}(M_2))$, therefore $F \subset F''$ implies that $M_2 \subset F'' \cap \mathrm{V}_2$ and hence that $M_2' \subseteq F'' \cap \mathrm{V}_2$ as $M_2 \in \mathcal{F}_2$. Since $F_1 \subseteq F'' \cap \mathrm{V}_1$, we get $F \prec M_2' \cup F_1 \subseteq F''$ and $F \in \mathcal{M}$ as it has a unique cover.

Now we prove the other side of the equation. Let $M \in \mathcal{M}$. As $\mathcal{F} \subseteq \mathcal{F}_1 \times \mathcal{F}_2$, $M \cap \mathrm{V}_2 \in \mathcal{F}_2$ and we can distinguish two cases. Either $M \cap \mathrm{V}_2 = \mathrm{V}_2$ or $M \cap \mathrm{V}_2 \subset \mathrm{V}_2$. Let us study the first case and let $M_1 = M \cap \mathrm{V}_1$. Let M' be the unique cover of M in $\mathcal{F}$. We show that $M_1' = M' \cap \mathrm{V}_1$ is the unique cover of M_1 in $\mathcal{F}_1$. By Theorem 1 and Lemma 2, we have that $M_1 \prec M_1'$ in $\mathcal{F}_1$. Let F_1 be any closed set of $\mathcal{F}_1$ with $M_1 \subset F_1$. Recall that $\mathsf{Ext}(\mathrm{V}_2)\colon \mathrm{V}_1 = \mathcal{F}_1$ by Lemmas 1 and 2. Hence

$F_1 \cup \mathrm{V}_2$ is closed and $M \subseteq F_1 \cup \mathrm{V}_2$. As $M \in M$, we also deduce $M' \subseteq F_1 \cup \mathrm{V}_2$. Therefore, $M'_1 \subseteq F_1$, and M'_1 must be the unique cover of M_1 in $\mathcal{F}_1$. So, $M_1 \in \mathcal{M}_1$ and for any $M \in \mathcal{M}$ such that $\mathrm{V}_2 \subseteq M$, we have $M \in \{M_1 \cup \mathrm{V}_2 \mid M_1 \in \mathcal{M}_1\}$.

Now assume that $M \cap \mathrm{V}_2 \subset \mathrm{V}_2$. Let $M_1 = M \cap \mathrm{V}_1$ and $M_2 = M \cap \mathrm{V}_2$. Then by contrapositive of Lemma 3 we have that $M \in \mathsf{max}_{\subseteq}(\mathsf{Ext}(M_2))$ as $M_2 \neq \mathrm{V}_2$. Similarly we get $M_2 \in \mathcal{M}_2$ by Lemma 4. □

This theorem hints a strategy to compute meet-irreducible elements in a recursive manner, using a hierarchical decomposition of $\mathcal{H}$ with acyclic splits, as proposed in the next section.

5 Recursive application of acyclic splits

In this section, we discuss an algorithm to compute $\mathcal{M}$ from a dihypergraph $\mathcal{H}$ based on Theorem 2. First, note that we have both $|\mathcal{M}| \geq |\mathcal{M}_1|$ and $|\mathcal{M}| \geq |\mathcal{M}_2|$. Furthermore, each $M \in \mathcal{M}$ arise from a unique element of $M' \in \mathcal{M}_1 \cup \mathcal{M}_2$, and each $M' \in \mathcal{M}_1 \cup \mathcal{M}_2$ is used to construct at least one new meet-irreducible element $M \in \mathcal{M}$. Therefore, we deduce an algorithm whose output is precisely $\mathcal{M}$, where each $M \in \mathcal{M}$ is given only once. Furthermore, the space needed to store intermediate solutions is bounded by the size of the output $\mathcal{M}$ which prevents an exponential blow up during the execution. The algorithm proceeds as follows. For c-factors of $\mathcal{H}$, we use algorithms such as in [19] to compute $\mathcal{M}$. When c-factors are singletons, the unique meet-irreducible to find is $\emptyset$ and hence no call to other algorithm is required. Otherwise, we find an acyclic split $(\mathrm{V}_1, \mathrm{V}_2)$ of $\mathcal{H}$ and we recursively call the algorithm on $\mathcal{H}[\mathrm{V}_1]$ and $\mathcal{H}[\mathrm{V}_2]$. Then, we compute $\mathcal{M}$ using $\mathcal{M}_1, \mathcal{M}_2$ and Theorem 2.

Computing $\mathcal{M}$ from $\mathcal{M}_1$, $\mathcal{M}_2$ requires to find maximal extensions of every meet-irreducible element $M_2 \in \mathcal{M}_2$. We will show that finding maximal extensions of a closed set is equivalent to a *dualization problem* in closure systems. First, we state the extension problem:

Problem: Find Maximal Extensions with Acyclic Split (FMEAS)
Input: A triple $\mathcal{H}[\mathrm{V}_1]$, $\mathcal{H}[\mathrm{V}_2]$, $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ given by an acyclic split of a dihypergraph $\mathcal{H}$, meet-irreducible elements $\mathcal{M}_1, \mathcal{M}_2$, and a closed set F_2 of $\mathcal{H}[\mathrm{V}_2]$.
Output: The maximal extensions of F_2 in $\mathcal{F}$, i.e., $\mathsf{max}_{\subseteq}(\mathsf{Ext}(F_2))$.

Let $\mathcal{B}^+$, $\mathcal{B}^-$ be two antichains of $\mathcal{F}$. The dualization in lattices asks if two antichains $\mathcal{B}^-$, $\mathcal{B}^+$ are *dual* in $\mathcal{F}$, that is if

$$\downarrow \mathcal{B}^+ \cup \uparrow \mathcal{B}^- = \mathcal{F} \text{ and } \uparrow \mathcal{B}^- \cap \downarrow \mathcal{B}^+ = \emptyset.$$

Note that $\mathcal{B}^-$ and $\mathcal{B}^+$ are dual if either $\mathcal{B}^+ = \mathsf{max}_{\subseteq}\{F \in \mathcal{F} \mid F \notin \uparrow \mathcal{B}^-\}$ or $\mathcal{B}^- = \mathsf{min}_{\subseteq}\{F \in \mathcal{F} \mid F \notin \downarrow \mathcal{B}^+\}$. If $\mathcal{F}$ is given, the question can be answered in polynomial time. In our case however, $\mathcal{F}$ is implicitly given by $\mathcal{M}$ and $\mathcal{H}$. More precisely we use the next generation problem:

Problem: Dualization with Dihypergraph and meet-irreducible (DMDual)

Input: A dihypergraph $\mathcal{H} = (\mathrm{V}, \mathcal{E})$, the meet-irreducible elements $\mathcal{M}$ of $\mathcal{F}$, and an antichain $\mathcal{B}^-$ of $\mathcal{F}$.
Output: The dual antichain $\mathcal{B}^+$ of $\mathcal{B}^-$.

This problem has been introduced in [3] in its decision version, where authors show that it is not harder than finding a (minimum) dihypergraph from a set of meet-irreducible elements. In general however, the problem is open. When $\mathcal{H}$ has no arcs, DMDual is equivalent to hypergraph dualization as there are $|\,\mathrm{V}\,|$ meet-irreducible elements which can easily be computed by taking $\mathrm{V}\setminus\{x\}$ for any $x \in \mathrm{V}$. This latter problem can be solved in output quasi-polynomial time using the algorithm of Fredman and Khachiyan [10].

We show that FMEAS and DMDual are equivalent under polynomial reduction. First, we relate maximal extensions of a closed set with dualization. Let $F_2 \in \mathcal{F}_2$. Recall that $\mathsf{Ext}(F_2)\colon \mathrm{V}_1$ is an ideal of $\mathcal{F}_1$. Hence, the antichain $\mathsf{max}_{\subseteq}(\mathsf{Ext}(F_2)\colon \mathrm{V}_1)$ has a dual antichain $\mathcal{B}^-(F_2)$ in $\mathcal{F}_1$, i.e., $\mathcal{B}^-(F_2) = \mathsf{min}_{\subseteq}\{F_1 \in \mathcal{F}_1 \mid F_1 \notin \mathsf{Ext}(F_2)\colon \mathrm{V}_1\}$.

Proposition 3. *Let $F_2 \in \mathcal{F}_2$, and $F_1 \in \mathcal{F}_1$. Then, $F_1 \in \mathcal{B}^-(F_2)$ if and only if $F_1 \in \mathsf{min}_{\subseteq}\{B^{\mathcal{H}[\mathrm{V}_1]} \mid (B,h) \in \mathcal{H}[\mathrm{V}_1,\mathrm{V}_2], h \notin F_2\}$.*

Proof. We show the if part. Let $F_1 \in \mathsf{min}_{\subseteq}\{B^{\mathcal{H}[\mathrm{V}_1]} \mid (B,h) \in \mathcal{H}[\mathrm{V}_1,\mathrm{V}_2], h \notin F_2\}$. We show that for any closed set $F_1' \subseteq F_1$ in $\mathcal{F}_1$, F_1' contributes to an extension of F_2. It is sufficient to show this property to the case where $F_1' \prec F_1$ as $\mathsf{Ext}(F_2)\colon \mathrm{V}_1$ is an ideal of $\mathcal{F}_1$. Hence consider a closed set F_1' in $\mathcal{F}_1$ such that $F_1' \prec F_1$. Note that such F_1' exists since $\emptyset \in \mathcal{F}_1$ and no arc (B,h) in $\mathcal{H}$ has $B = \emptyset$ so that $\emptyset \subset B^{\mathcal{H}[\mathrm{V}_1]}$ for any arc (B,h) of $\mathcal{H}[\mathrm{V}_1,\mathrm{V}_2]$ such that $h \notin F_2$. Then, by construction of F_1', for any (B,h) in $\mathcal{H}[\mathrm{V}_1,\mathrm{V}_2]$ such that $h \notin F_2$, we have $B^{\mathcal{H}[\mathrm{V}_1]} \not\subseteq F_1'$. As $(.)^{\mathcal{H}[\mathrm{V}_1]}$ is a closure operator, it is monotone and $B^{\mathcal{H}[\mathrm{V}_1]} \not\subseteq {F_1'}^{\mathcal{H}[\mathrm{V}_1]} = F_1'$ entails $B \not\subseteq F_1'$ for any such arc (B,h). Therefore $F_1' \in \mathsf{Ext}(F_2)\colon \mathrm{V}_1$ and $F_1 \in \mathcal{B}^-(F_2)$.

We prove the only if part. We use contrapositive. Assume $F_1 \notin \mathsf{min}_{\subseteq}\{B^{\mathcal{H}[\mathrm{V}_1]} \mid (B,h) \in \mathcal{H}[\mathrm{V}_1,\mathrm{V}_2], h \notin F_2\}$. Then we have two cases. First, for any arc (B,h) in $\mathcal{H}[\mathrm{V}_1,\mathrm{V}_2]$ such that $h \notin F_2$, $B^{\mathcal{H}[\mathrm{V}_1]} \not\subseteq F_1$. As $(.)^{\mathcal{H}[\mathrm{V}_1]}$ is a closure operator, it is monotone, and since F_1 is closed in $\mathcal{F}_1$, we have $B \not\subseteq F_1$ and $F_1 \in \mathsf{Ext}(F_2)\colon \mathrm{V}_1$ by Lemma 1. Hence $F_1 \notin \mathcal{B}^-(F_2)$. In the second case, there is an arc (B,h) with $h \notin F_2$ in $\mathcal{H}[\mathrm{V}_1,\mathrm{V}_2]$ such that $B^{\mathcal{H}[\mathrm{V}_1]} \subseteq F_1$ which implies $F_1 \notin \mathsf{Ext}(F_2)\colon \mathrm{V}_1$. If $B^{\mathcal{H}[\mathrm{V}_1]} \subset F_1$, then clearly $F_1 \notin \mathcal{B}^-(F_2)$ as $B^{\mathcal{H}[\mathrm{V}_1]} \in \mathcal{F}_1$ and $B^{\mathcal{H}[\mathrm{V}_1]} \notin \mathsf{Ext}(F_2)\colon \mathrm{V}_1$. Hence, assume that $F = B^{\mathcal{H}[\mathrm{V}_1]}$. Since $F_1 \notin \mathsf{min}_{\subseteq}\{B^{\mathcal{H}[\mathrm{V}_1]} \mid (B,h) \in \mathcal{H}[\mathrm{V}_1,\mathrm{V}_2], h \notin F_2\}$ by hypothesis, there exists another arc $(B',h') \in \mathcal{E}(\mathcal{H}[\mathrm{V}_1,\mathrm{V}_2])$ such that $h \notin F_2$ and ${B'}^{\mathcal{H}[\mathrm{V}_1]} \subset F_1$. Hence ${B'}^{\mathcal{H}[\mathrm{V}_1]} \notin \mathsf{Ext}(F_2)\colon \mathrm{V}_1$ and $F_1 \notin \mathcal{B}^-(F_2)$ as it is not an inclusion-wise minimum closed set which does not belong to $\mathsf{Ext}(F_2)\colon \mathrm{V}_1$. □

Observe that for any $F_2 \in \mathcal{F}_2$, $\mathcal{B}^-(F_2)$ can easily be computed using $\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2]$ and Lemma 1. Therefore we prove the following theorem.

Theorem 3. FMEAS *and* DMDual *are polynomially equivalent.*

Proof. First we show that DMDual is harder than FMEAS. Let $\mathcal{H} = (\mathrm{V}, \mathcal{E})$ be a dihypergraph, and $(\mathcal{H}[\mathrm{V}_1], \mathcal{H}[\mathrm{V}_2], \mathcal{H}[\mathrm{V}_1, \mathrm{V}_2], \mathcal{M}_1, \mathcal{M}_2, F_2)$ be an instance of FMEAS. By Proposition 3, finding $\mathsf{max}_{\subseteq}(\mathsf{Ext}(F_2))$ amounts to find the dual antichain of $\mathcal{B}^-(F_2) = \mathsf{min}_{\subseteq}\{B^{\mathcal{H}[\mathrm{V}_1]} \mid (B, h) \in \mathcal{H}[\mathrm{V}_1, \mathrm{V}_2], h \notin F_2\}$ in $\mathcal{F}_1$. Note that $\mathcal{B}^-(F_2)$ can be computed in polynomial time in the size of $\mathcal{H}[\mathrm{V}_1]$ and $|\mathcal{B}^-(F_2)| \leq |\mathcal{E}(\mathcal{H}[\mathrm{V}_1, \mathrm{V}_2])|$. Therefore, the instance of FMEAS reduces to the instance $(\mathcal{H}[\mathrm{V}_1], \mathcal{M}_1, \mathcal{B}^-(F_2))$ of DMDual.

Now we show that FMEAS is harder than DMDual. Let $(\mathcal{H}, \mathcal{M}, \mathcal{B}^-)$ be an instance of DMDual. Let z be a new gadget vertex and consider the bipartite dihypergraph $\mathcal{H}[\mathrm{V}, \{z\}] = (\mathrm{V} \cup \{z\}, \{(B, z) \mid B \in \mathcal{B}^-\})$. Let $\mathcal{H}_{new} = \mathcal{H} \cup \mathcal{H}[\mathrm{V}, \{z\}]$. Clearly, $\mathcal{H}_{new}$ has an acyclic split $(\mathrm{V}, \{z\})$ such that $\mathcal{H}_{new}[\mathrm{V}] = \mathcal{H}$, $\mathcal{H}_{new}[\{z\}] = (\{z\}, \emptyset)$ and $\mathcal{H}_{new}[\mathrm{V}, \{z\}] = \mathcal{H}[\mathrm{V}, \{z\}]$. The closure system associated to $\mathcal{H}_{new}[\{z\}]$ has only 2 elements: its unique meet-irreducible element $\emptyset$ and $\{z\}$. We obtain an instance FMEAS where the input is $\mathcal{H}$, $\mathcal{H}_{new}[\{z\}]$, $\mathcal{H}[\mathrm{V}, \{z\}]$, $\mathcal{M}$, $\{\emptyset\}$ and where the closed set of interest is $\emptyset$. Moreover this reduction is polynomial in the size of $(\mathcal{H}, \mathcal{M}, \mathcal{B}^-)$ as we create a unique new element and $|\mathcal{B}^-|$ arcs. According to Proposition 3, maximal extensions of $\emptyset$ are given by the antichain dual to $\mathcal{B}^-(\emptyset) = \mathsf{min}_{\subseteq}\{B^{\mathcal{H}} \mid (B, z) \in \mathcal{H}[\mathrm{V}, \{z\}]\}$. However, we have $\mathcal{B}^-(\emptyset) = \mathcal{B}^-$, so that maximal extensions of $\emptyset$ are precisely elements of the dual antichain $\mathcal{B}^+$ of $\mathcal{B}^-$. □

We can deduce a class of dihypergraphs where our strategy can be applied to obtain meet-irreducible elements in output quasi-polynomial time. Let us assume that $\mathcal{H}$ can be decomposed as follows. Its c-factors are singletons. If $\mathcal{H}$ is not itself a singleton, it has an acyclic split $(\mathrm{V}_1, \mathrm{V}_2)$ with $\mathcal{H}[\mathrm{V}_1] = (\mathrm{V}_1, \emptyset)$. Hence, DMDual reduces to hypergraph dualization and can be solved in output-quasi polynomial time using the algorithm of [10]. Recursively applying hypergraph dualization, we get $\mathcal{M}$ for $\mathcal{H}$ in output-quasi polynomial time. This class of dihypergraph generalizes ranked convex geometries of [8].

The closure system represented by a dihypergraph $\mathcal{H}$ is a ranked convex geometry if there exists a full partition $\mathrm{V}_1, \ldots, \mathrm{V}_n$, of V such that $\mathcal{H}[\mathrm{V}_i] = (\mathrm{V}_i, \emptyset)$ for any $1 \leq i \leq n$ and for any arc (B, h) in $\mathcal{H}$ there is a $j < k$ such that $B \subseteq \mathrm{V}_j$ and $h \in \mathrm{V}_{j+1}$. All c-factors of $\mathcal{H}$ are singletons. Choosing the acyclic split $(\mathrm{V}_i, \bigcup_{j=i+1}^{n} \mathrm{V}_j)$ at the i-th step of the algorithm yields a decomposition which satisfies conditions of the previous paragraph.

6 Conclusion

In this paper we investigated the problem of finding meet-irreducible elements of a closure system represented by a dihypergraph. In general, the complexity of this problem is unknown and harder than hypergraph dualization. Using a partitioning operation called an acyclic split on the dihypergraph, we gave a characterization of its associated meet-irreducible elements. Acyclic splits lead to a recursive algorithm to find meet-irreducible elements from a dihypergraph.

With our algorithm, we reach new classes of dihypergraphs for which meet-irreducible elements can now be computed in output quasi-polynomial time. In particular, we improve previous results on ranked convex geometries [8].

Acknowledgment Authors are thankful to reviewers for their helpful remarks.

References

[1] Ausiello, G., and Laura, L. Directed hypergraphs: Introduction and fundamental algorithms—a survey. *Theoretical Computer Science 658* (2017), 293–306.

[2] Babin, M. A., and Kuznetsov, S. O. Computing premises of a minimal cover of functional dependencies is intractable. *Discrete Applied Mathematics 161*, 6 (2013), 742–749.

[3] Babin, M. A., and Kuznetsov, S. O. Dualization in lattices given by ordered sets of irreducibles. *Theoretical Computer Science 658* (2017), 316–326.

[4] Beaudou, L., Mary, A., and Nourine, L. Algorithms for k-meet-semidistributive lattices. *Theoretical Computer Science 658* (2017), 391–398.

[5] Bertet, K., and Monjardet, B. The multiple facets of the canonical direct unit implicational basis. *Theoretical Computer Science 411*, 22-24 (2010), 2155–2166.

[6] Birkhoff, G. *Lattice theory*, vol. 25. American Mathematical Soc., 1940.

[7] Davey, B. A., and Priestley, H. A. *Introduction to lattices and order*. Cambridge university press, 2002.

[8] Defrain, O., Nourine, L., and Vilmin, S. Translating between the representations of a ranked convex geometry. *arXiv preprint arXiv:1907.09433* (2019).

[9] Demetrovics, J., Libkin, L., and Muchnik, I. B. Functional dependencies in relational databases: a lattice point of view. *Discrete Applied Mathematics 40*, 2 (1992), 155–185.

[10] Fredman, M. L., and Khachiyan, L. On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms 21*, 3 (1996), 618–628.

[11] Gallo, G., Longo, G., Pallottino, S., and Nguyen, S. Directed hypergraphs and applications. *Discrete applied mathematics 42*, 2-3 (1993), 177–201.

[12] Ganter, B., and Wille, R. *Formal concept analysis: mathematical foundations*. Springer Science & Business Media, 2012.

[13] Guigues, J.-L., and Duquenne, V. Familles minimales d'implications informatives résultant d'un tableau de données binaires. *Mathématiques et Sciences humaines 95* (1986), 5–18.

[14] Habib, M., and Nourine, L. Representation of lattices via set-colored posets. *Discrete Applied Mathematics 249* (2018), 64–73.

[15] Kavvadias, D. J., Sideri, M., and Stavropoulos, E. C. Generating all maximal models of a boolean expression. *Information Processing Letters 74*, 3-4 (2000), 157–162.

[16] Khardon, R. Translating between horn representations and their characteristic models. *Journal of Artificial Intelligence Research 3* (1995), 349–372.
[17] Libkin, L. Direct product decompositions of lattices, closures and relation schemes. *Discrete Mathematics 112*, 1-3 (1993), 119–138.
[18] Maier, D. Minimum covers in relational database model. *Journal of the ACM (JACM) 27*, 4 (1980), 664–674.
[19] Mannila, H., and Räihä, K.-J. *The design of relational databases*. Addison-Wesley Longman Publishing Co., Inc., 1992.
[20] Markowsky, G. Primes, irreducibles and extremal lattices. *Order 9*, 3 (1992), 265–290.
[21] Nourine, L., and Vilmin, S. Hierarchical decompositions of dihypergraphs. *arXiv preprint arXiv:2006.11831* (2020).
[22] Wild, M. Computations with finite closure systems and implications. In *International Computing and Combinatorics Conference* (1995), Springer, pp. 111–120.
[23] Wild, M. The joy of implications, aka pure horn formulas: mainly a survey. *Theoretical Computer Science 658* (2017), 264–292.

Closure Structure: a Deeper Insight

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

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

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

1 Introduction

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

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

In our previous work [11], we proposed the closure structure of concept lattice (i.e., the whole set of closed itemsets) and an algorithm for its gradual computing. The algorithm computes closed itemsets (formal concepts) by levels with

polynomial delay. Each level may contain itemsets of different frequency, however, the number of frequent itemsets decreases with each new level. In [11] we presented some theoretical results and described characteristics of closed itemsets by levels based on experiments. In this work, we study further topology of the closure structure and applicability of concepts for classification by closure levels.

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

2 Basic notions

2.1 Concepts and the partial order between them

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

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

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

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

2.2 Equivalence classes and key sets

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

is maximal in this equivalence class. An equivalence class is called trivial if it consists only of a closed itemset.

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

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

Table 1. Formal context and nontrivial equivalence classes.

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

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

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

2.3 Level-wise structure on minimum key sets

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

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

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

3 The GDPM algorithm and related issues

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

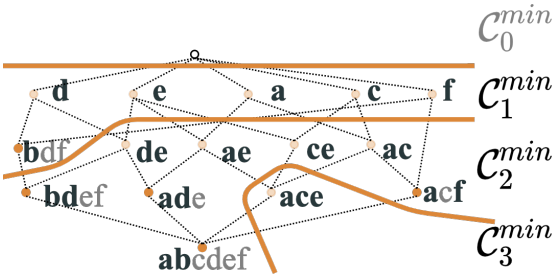


Fig. 1. The closure structure of the concept lattice of the context from Table 1 induced by the sizes of the minimum keys. The smallest lexicographically minimum keys for each concept are highlighted in boldface.

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

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

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

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

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

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

Algorithm 1 GDPM

Require: $\mathcal{K}^*_{k-1}$, a subset of the minimum key set $\mathcal{K}^{min}_{k-1}$ of level $k-1$,
$\mathcal{T}_{k-1}$, the lexicographic tree containing all closed itemsets $\bigcup_{i=1,\dots,k-1} \mathcal{C}^{min}_i$
Ensure: $\mathcal{K}^*_k$, a subset of the minimum key set of level k

$\mathcal{K}^*_k \leftarrow \emptyset$
$\mathcal{T}_k \leftarrow \mathcal{T}_{k-1}$
for all $X \in \mathcal{K}^*_{k-1}$ **do**
  $Y \leftarrow M \setminus X''$
  for all $m \in Y$ **do**
    $X^* = X \cup \{m\}$
    $S \leftarrow (X^*)''$
    if $S \notin \mathcal{T}_k$ **then**
      $add(\mathcal{T}_k, S)$
      $\mathcal{K}^*_k \leftarrow \mathcal{K}^*_k \cup \{X^*\}$
    end if
  end for
end for
return $\mathcal{K}^*_k$

4 Concepts as classifiers. Baseline classification model

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

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

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

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

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

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

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

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

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

5 Experiments

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

Table 2. Description of datasets and their level structure.

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

5.1 Concept contrast based on F1 measure

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

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

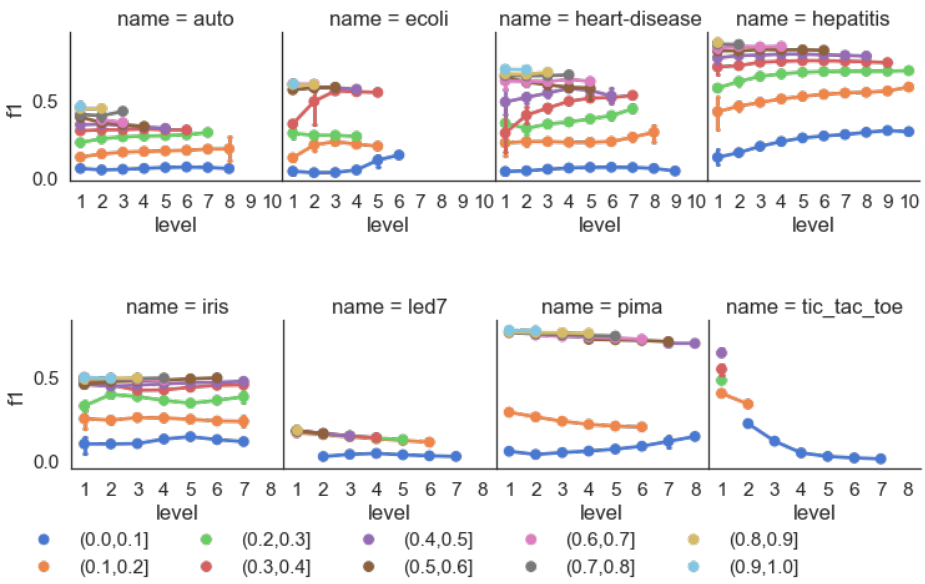


Fig. 2. The average F1 measure for 8 datasets within 10 frequency ranges.

In the previous study we showed that the size of closure levels resembles the values of binomial coefficients, i.e., the largest levels are located in the middle

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

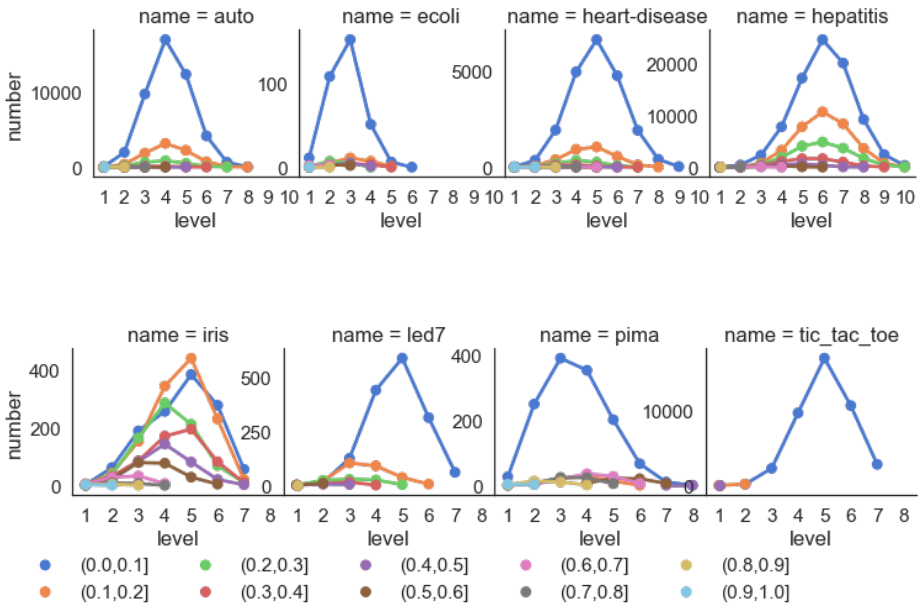


Fig. 3. The size of the concepts of a given frequency range w.r.t. level number.

5.2 Classification quality

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

The experiments showed that both SL- and CL-classifiers may achieve quite high accuracy. The average accuracy for 8 datasets is given in Fig. 4. The maximal (or close to the maximal) accuracy of CL-classifiers is achieved at the first

levels and usually changes slightly when the classifier is extended by the further closure levels. For SL-classifiers the maximal (or close to the maximal) accuracy is usually achieved at one of the first levels.

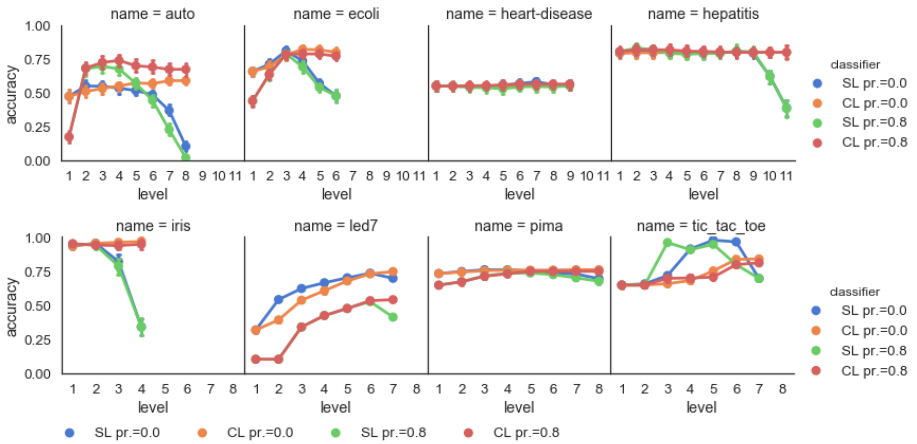


Fig. 4. Average accuracy of four classifiers: "SL/CL pr. = X" stands for a single-level (SL) or cumulative (CL) classifier where each concept has precision at least X.

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

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

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

6 Conclusion

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

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

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

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

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

References

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

J. (eds.) Contrast Data Mining: Concepts, Algorithms, and Applications, pp. 269–282. CRC Press (2013)
7. Ganter, B., Wille, R.: Formal concept analysis–mathematical foundations (1999)
8. Ganter, B., Grigoriev, P.A., Kuznetsov, S.O., Samokhin, M.V.: Concept-based data mining with scaled labeled graphs. In: International Conference on Conceptual Structures. pp. 94–108. Springer (2004)
9. Geng, L., Hamilton, H.J.: Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR) **38**(3), 9–es (2006)
10. Kuznetsov, S.O.: Interpretation on graphs and complexity characteristics of a search for specific patterns. Automatic Documentation and Mathematical Linguistics **24**(1), 37–45 (1989)
11. Makhalova, T., Kuznetsov, S.O., Napoli, A.: Gradual discovery with closure structure of a concept lattice. In: The 15th International Conference on Concept Lattices and Their Applications (2020)
12. Proença, H.M., van Leeuwen, M.: Interpretable multiclass classification by mdl-based rule lists. Information Sciences **512**, 1372–1393 (2020)
13. Stumme, G., Taouil, R., Bastide, Y., Pasquier, N., Lakhal, L.: Computing iceberg concept lattices with titanic. Data & knowledge engineering **42**(2), 189–222 (2002)
14. Vreeken, J., Van Leeuwen, M., Siebes, A.: Krimp: mining itemsets that compress. Data Mining and Knowledge Discovery **23**(1), 169–214 (2011)

Towards Polynomial Subgroup Discovery by means of FCA*

Aleksey Buzmakov[0000−0002−9317−8785]

National Research University Higher School of Economics, Russia
avbuzmakov@hse.ru

Abstract. The goal of subgroup discovery is to find groups of objects that are significantly different than "average" object w.r.t. some supervised information. It is a computational intensive procedure that traverses a large searching space corresponding to the set of formal concepts. It was recently found that a part of formal concepts, called stable concepts, can be found in polynomial time. Accordingly, in this paper a new algorithm, called `SD-SOFIA`, is presented. `SD-SOFIA` fits subgroup discovery process in the framework of stable concept search. The proposed algorithm is evaluated on a dataset from UCI repository. It is shown that its practical computational complexity is polynomial.

Keywords: Subgroup Discovery · Stable Concepts · Supervised Learning · Exploratory Data Analysis · Algorithms.

Introduction

Subgroup discovery is supervised data mining technique allowing for finding groups of objects that express unexpected behavior w.r.t. some supervised information [1]. For example, class labels of objects is an example of such supervised information. Subgroup discovery not only enumerates the objects with unexpected behavior but also describes them in a human readable form providing a way for understanding the connection between the supervised information and the description space. Such understanding is important for analysis of the real process generating the supervised information.

Formal concept analysis (FCA) [11] is a well-established mathematical formalism suitable for description of subgroup discovery process. Indeed, extents of formal concepts are subgroups, their intents are subgroup descriptions. Then every concept can be evaluated by means of a quality function that relates object class labels[1] and the concept interest. Then the task of the subgroup discovery in these terms is to find the concept with the best value of the quality function.

One of the challenges in subgroup discovery is the exponentially large searching space of concepts that entails two consequences. First, it is computationally hard to find the best subgroups. Second, since the searching space is large, it is

* The reported study was funded by RFBR, project number 19-31-37001

[1] For simplicity only classification task is considered.

likely that some its elements are associated with high values of quality function just by chance, i.e., spurious findings are possible. There are a number of approaches that deals with these consequences. In particular, the searching space can be limited, e.g., it can be stated that only concepts with few attributes constitute the searching space. Another possible way is to rely on a certain heuristic such that it is unlikely to miss the best subgroups [14, 5]. Finally, some approaches allow gradually refining the searching space in such a way that their computation can be stopped at any moment and the best found result is reported [2].

The problem of spurious findings can be solved by controlling the statistical significance of the examined subgroups [15]. Such approaches can be naturally combined with the approaches that limit the searching space. By contrast, combing statistically significant subgroup discovery with heuristic methods is hard.

Accordingly, in this paper we discuss a limitation of the searching space that is based on stability of a formal concept. This choice is motivated by two facts. First, stability is a meaningful concept selection method. Indeed, stability is the probability of a concept to be preserved after random deletion of objects from the dataset. Thus, if just a small modification of the object set is enough for removing a concept, then probably this concept can be removed from the searching space. Second, it was recently shown that stability threshold can be adjusted on the fly, allowing for polynomial time computational procedure [8, 6]. Consequently, combining this procedure with subgroup discovery give an opportunity for controling the subgroup discovery computation time and allowing for statistically significant subgroup discovery.

Accordingly, the main contribution of this paper is the algorithm combining the subgroup discovery process and the stability threshold adjustment allowing for practical polynomial time complexity subgroup discovery with mathematical guarantees for the resulting subgroup.

The rest of the paper is organized as follows. Section 1 provides some basic definitions, then in Section 2 the task is defined. Later the algorithms proposed in this paper are discussed in Section 3. Finally, before concluding the paper the algorithms are evaluated an a dataset from the UCI repository [9].

1 Definitions

Subgroup discovery task statement depends (1) on the searching space and (2) on the quality function. The searching space is defined by means of FCA, while quality function is considered to be known and is not discussed in this paper.

1.1 Formal concept analysis

For simplicity the dataset is described as a formal context (G, M, I), where G is the set of objects, M is the set of attributes, and $I \subseteq G \times M$ is a relation between. The sets of objects and attributes are connected by means of a derivation

operator[2]:

$$A^{\uparrow} = \{B \subseteq M \mid \forall g \in A\,((g,m) \in I)\},\ \text{for } A \subseteq G, \tag{1}$$

$$B^{\downarrow} = \{A \subseteq G \mid \forall m \in B\,((g,m) \in I)\},\ \text{for } B \subseteq M. \tag{2}$$

A formal concept is a pair (A, B), where $A \subseteq G$ is callede extent and $B \subseteq M$ is called intent, such that $A = B^{\downarrow}$ and $B = A^{\uparrow}$. The set of concepts is ordered w.r.t. the order on extents (or dually inverse order on intents). This order is a lattice called concept lattice.

1.2 Projections

The algorithm presented in Section 3 is based on the notion of projections. Originally, it was introduced within the framework of pattern structures [10]. However, for the sake of simplicity, it is presented here in terms of standard FCA.

Definition 1. *Projection $\psi : 2^M \to 2^M$ is a function sutisfying:*

- *idempotency, i.e., $\psi(\psi(X)) = \psi(X)$;*
- *monotonicity, i.e., $X \subset Y \longrightarrow \psi(X) \subseteq \psi(Y)$;*
- *contractivity, i.e., $X \supseteq \psi(X)$.*

In this paper, a special kind of projections is considered corresponding to removal of certain attributes. In particular, $\psi_Y(X) = X \cap Y$, i.e., it removes all attributes outside of Y. It can be seen, that the ψ_Y is a projection. The larger the set Y the more attributes are preserved after the projection.

As it will be discussed later the algorithm starts from the most simple projections, removing many attributes, and then iteratively adds attributes one by one updating the result. A similar approach was used in [2] where numerical intervals were iteratively updated.

1.3 Subgroup Discovery

From the subgroup discovery (SD) point of view the concept lattice is the searching space. The extent of a formal concept is the subgroup and the intent of a formal context is the description of this subgroup. Given a quality function $Q : 2^G \to \mathbb{R}$, the goal of SD is to find the formal concept with extent maximizing the quality function Q.

Let us consider a standard quality function for the classification task [13]. Let class labels of objects are given by a function $\texttt{class} : G \to \{0, 1\}$, where $\{0, 1\}$ is the set of available class labels. Given a set of objects A, the weighted relative accuracy can be expressed as

$$Q_1(A) = \frac{|A|}{|G|} \cdot \left(\frac{|\{g \in A \mid \texttt{class}(g) = 1\}|}{|A|} - \frac{|\{g \in G \mid \texttt{class}(g) = 1\}|}{|G|} \right).$$

[2] The operators $(\cdot)^{\downarrow}$ and $(\cdot)^{\uparrow}$ are used instead of the standard $(\cdot)'$, since it makes the reading more clear w.r.t. the operator argument.

This quality function express a trade-off between the size of the subgroup $\frac{|A|}{|G|}$ and the improvement in the "purity" of class labels within the concept w.r.t. the average "purity".

Although it is possible to iterate over all formal concepts and compute their quality, it is not efficient. Thus, a typical exhaustive subgroup discovery procedure is implemented as a branch-and-bound search. It is based on the following two ingredients [3]:

- A refinement operator $\mathtt{r} : 2^G \to 2^{2^G}$, that is monotone, i.e., given $B_0 \subseteq M$, $(\forall B \in \mathtt{r}(B_0))B \subseteq B_0$. The refinement operator organizes the procedure of generating new "more specific" candidate subgroups based on already found ones.
- An optimistic estimator $\overline{Q} : 2^G \to \mathbb{R}$ of the subgroup quality function Q. The optimistic estimator $\overline{Q}(A)$ is the upper bound of the quality function Q on any subset of A, i.e., $(\forall S \subseteq A)\overline{Q}(A) \geq Q(S)$.

Given these two components and the quality of the already found best subgroup q_{best} the discovering procedure is as follows. For any found subgroup A with description $B_0 = A^{\uparrow}$ one can verify if any subset of A is a potentially interesting subgroup, i.e., $\overline{Q}(A) > q_{best}$. If not the subgroup A can be ignored, otherwise it is refined by means of the refinement operator $\mathtt{r}$. The refined subgroups with description $B \in r(B_0)$ are evaluated by means of the subgroup quality function Q and if any subgroup is better then the already found one, the best subgroup is updated.

1.4 Formal Concept Stability and Δ-stability

The branch and bound computational efficiency is still limited if the dataset is large. The further improvement of the efficiency can be made by modification of the searching space. One way is to consider only "stable" concepts as the searching space.

Stability of a formal concept [12] measures how strong the concept depend on the dataset. If removal of random objects from the dataset is likely to remove a concept then the concept is not stable.

It was shown that, given a context and a concept, the computation of concept stability is #P-complete [12]. Accordingly, in [7] bounds on stability of a concept was discussed, in particular, stability is bounded as follows:

$$1 - \sum_{d \in \mathrm{DD}(c)} \frac{1}{2^{\Delta(c,d)}} \leq \mathrm{Stab}(c) \leq 1 - \max_{d \in \mathrm{DD}(c)} \frac{1}{2^{\Delta(c,d)}}, \tag{3}$$

where $\mathrm{DD}(c)$ is the set of all direct descendants of a concept c in the lattice and $\Delta(c,d)$ is the size of the set-difference between extent of c and extent of d, i.e. $\Delta(c,d) = |\,\mathrm{Ext}(c) \setminus \mathrm{Ext}(d)|$. It can be seen that stability in 3 is bounded tightly if $\Delta(c) = \min\limits_{d \in \mathrm{DD}(c)} \Delta(c,d)$ is high. Moreover, the higher the stability is, the tighter

the bounds. Thus, in order to identify the most stable concepts one can switch to $\Delta(c)$, called Δ-measure, which is polynomially computable.

Recently, it was also shown, that given a Δ-measure threshold θ, it is possible to directly find formal concepts with Δ-measure higher than θ [8]. Moreover, introducing a special adjustment procedure, one is able to extract concepts with the highest value of Δ-measure in polynomial time [6]. Accordingly, in this paper this approach is translated to subgroup discovery task.

2 Task Statement

The goal of this paper is to present an algorithm for subgroup discovery task modifying the searching space in such a way that its practical computational complexity is polynomial.

More formally, given a formal context $\mathbb{K} = (G, M, I)$ and a quality function Q of a formal concept, the goal is an algorithm that finds the threshold θ of Δ-measure and the corresponding best Δ-stable concept $\mathcal{C}$, w.r.t. Q, such that the practical computational complexity is polynomial w.r.t. $\mathbb{K}$, $\mathbb{T}(Q)$, and the memory usage limit L, where $\mathbb{T}(Q)$ is the computational complexity of Q, and L is the maximal number of potential concepts to be extended at the same time.

3 Algorithms

3.1 Algorithm ϑ-Σοφια

Let us first remind the original algorithm Σοφια [8, 6]. It is based on the following observations.

Proposition 1 (Projection antimonotonicity). *Given a context $\mathbb{K} = (G, M, I)$, for any projection $\psi : 2^M \to 2^M$ and a concept $\mathcal{C} = (A, B)$ it is valid that*

$$\Delta_{(G,M,I)}(\mathcal{C}) \leq \Delta_{(G,\psi(M),I)}\left((\psi(B)^{\downarrow}, \psi(B))\right). \tag{4}$$

It should be noticed, that Δ-measure depends on the structure of the concept lattice, i.e., Δ-measure should be computed when the lattice is fixed. Accordingly, in (4) every Δ-measure is indexed with the corresponding formal context. It also should be noticed that $(\psi(B)^{\downarrow}, \psi(B))$ is indeed a formal concept in $(G, \psi(M), I)$ as it is shown earlier [10].

This property (4) gives a way for direct search for Δ-stable concepts. Indeed, a concept $\mathcal{C} = (A, B)$ is not Δ-stable in $(G, \psi(M), I)$ for some ψ, then any preimages of B cannot be the intents of Δ-stable concepts in (G, M, I), i.e., any concept with intent $\hat{B}$, such that $\psi(\hat{B}) = B$ is not Δ-stable. Thus, if one is able to find Δ-stable concepts in $(G, \psi(M), I)$, then only the preimages of these concepts w.r.t. ψ can be Δ-stable in (G, M, I).

However, how can one efficiently find Δ-stable concepts in $(G, \psi(M), I)$? Since $(G, \psi(M), I)$ is a context it is possible to consider a projection of it. It forms a chain of projections and Δ-stable concepts are first found in the most

Data: A context $\mathbb{K} = (G, M, I)$, a chain of projections $\Psi = \{\psi_0, \psi_1, \cdots, \psi_k\}$, and a threshold θ for Δ-measure.

Function `ExtendProjection`(i, θ, $\mathcal{P}_{i-1}$)

  Data: Projection number i, a threshold value θ, and the set of concepts $\mathcal{P}_{i-1}$ for the projection ψ_{i-1}.

  Result: The set $\mathcal{P}_i$ of all concepts with the value of Δ-measure higher than the threshold θ for the projection ψ_i.

  $\mathcal{P}_i \longleftarrow \emptyset$;

  `/* Put all preimages in` $\psi_i(\mathbb{K})$ `for any concept` p `*/`

  foreach $p \in \mathcal{P}_{i-1}$ **do**

    $\mathcal{P}_i \longleftarrow \mathcal{P}_i \cup \text{Preimages}(i, p)$

  `/* Filter concepts in` $\mathcal{P}_i$ `to have a value of` Δ `higher than` θ `*/`

  foreach $p \in \mathcal{P}_i$ **do**

    if $\Delta_{\psi_i}(p) \leq \theta$ **then**

      $\mathcal{P}_i \longleftarrow \mathcal{P}_i \setminus \{p\}$

Function *ϑ-Σοφια*

  Result: The set $\mathcal{P}$ of all concepts with a value of Δ higher than θ.

  $\mathcal{P} \longleftarrow \left(\emptyset^{\downarrow}, \emptyset\right)$;

  `/* Run through out the chain` Ψ `*/`

  foreach $0 < i \leq |M|$ **do**

    $\mathcal{P} \longleftarrow$ `ExtendProjection`$(i, \theta, \mathcal{P})$;

Algorithm 1: The Θ-Σοφια algorithm for finding concepts in $\mathbb{K}$ with a value of a Δ-measure higher than a threshold θ.

general projections removing all attributes, then the next projections removes all but one attribute, the next one removes all but two attributes, on so on.

This procedure is shown in Algorithm 1 and is called ϑ-Σοφια algorithm. The computational complexity of this algorithm is

$$O\left(|M| \cdot \max_{0<i\leq|M|} |\mathcal{P}_i| \cdot (\mathbb{T}(\text{Preimages}) + \mathbb{T}(\Delta))\right).$$

It become polynomial if the threshold θ is adjusted in such a way that $|\mathcal{P}_i| < L$ for any i and a predefined memory limit L.

3.2 Algorithm `SD-SOFIA`

Usage of Algorithm 1 is limited for subgroup discovery, since it finds preimages of all concepts from projection i to projection $i+1$ simultaneously. By contrast, the most commonly used strategy in subgroup discovery is expansion of the most promising concept [4]. This allows for earlier finding of concepts with high quality Q improving the efficiency of branch cutting.

It should be noticed that in Algorithm 1 finding preimages of a concept does not depend on other concepts and, thus, concepts that are stored in $\mathcal{P}$ can correspond to different projections. In this case, one is able to choose the order of concept expansion and, in particular, it can be used for expanding first

```
1  Function FindBestConcept()
2  |  queue.Push ((G, ⊥ | 0));                    /* Projection number 0 */
3  |  while not queue.isEmpty() do
4  |  |  c ← queue.PopTheMostPromissing ();
5  |  |  {c_i} ← Preimages(c);
6  |  |  foreach cc ∈ {c_i} do
7  |  |  |  if Proj(cc) = |M| then
8  |  |  |  |  best.Register(cc);
9  |  |  |  |  next;
10 |  |  |  if Δ_{Proj(cc)}(cc) < θ then
11 |  |  |  |  next;
12 |  |  |  if not best.IsPromissing(cc) then
13 |  |  |  |  next;
14 |  |  |  queue.Push(cc);
15 |  |  if queue.Size() > L then
16 |  |  |  θ ← AdjustThld(queue, θ);
```

Algorithm 2: The SD-SOFIA algorithm identifying the Δ-measure threshold θ and the corresponding best concept w.r.t. a quality function Q. The algorithm ensures the polynomial computational complexity.

the most promising concepts w.r.t. subgroup discovery task. This procedure is shown in Algorithm 2. All concepts are stored in a queue. The queue can contain concepts from different projections, thus, a concept is denoted as $(A, B \mid i)$, where A and B are the extent and the intent of the concept correspondingly, and i is the projection it is computed in. The corresponding elements of a concept $c = (A, B \mid i)$ can be extracted by means of functions Ext, Int, and Proj for the extent, the intent, and the projection number of c correspondingly.

Let us first fix some order on attributes M. Let M_i be the first i attributes from M. Then, this algorithm relies on the following chain of projections $\Psi = \langle \psi_0, \psi_1, \ldots, \psi_{|M|} \rangle$, where $\psi_i(X) = X \cap M_i$, i.e., it removes all attributes but the first i attributes from M. In line 2, Algorithm SD-SOFIA initializes the queue with the only available concept in projection 0. This concept is $(G, \emptyset)$. Indeed, since M_0 contain no attribute, the only available intent is $\emptyset$.

Then, while queue is not empty the concepts are extracted one by one. For subgroup discovery task the concepts are extracted (line 4) w.r.t. their potential, i.e., the value of the optimistic estimate $\overline{Q}$ of the quality function Q. Then in line 5 the preimages of the most promissing concept $c = (A, B \mid i)$ are computed. Since projection ψ_{i+1} preserves one more attribute than projection ψ_i, there are only two possible preimages: $c_1 = (A, B^{\downarrow\uparrow} \mid i + 1)$ and $c_2 = ((B \cup \{i\})^{\downarrow}, (B \cup \{i\})^{\downarrow\uparrow} \mid i + 1)$. The preimages c_1 and c_2 can coincide and, thus, in this case only one of them should be considered.

Then every preimage cc of the concept c is processed in lines 7–14. First in lines 7–9 it is verified that cc is already in the last projection $\psi_{|M|}$. Only in this case the final Δ-measure value for this concept is known. Thus, only in this moment it is possible decide if this concept can be reported as the best concept.

```
1 Function AdjustThld(queue, θ0)
2 |   queue ← {c ∈ queue | best.IsPromissing(c)};
3 |   if queue.Size < |L| then
4 |   |   return θ0;
5 |   return min {θ > θ0 | L ≥ |{c ∈ queue | Δ(c) > θ}|};
```

Algorithm 3: Adjustment of minimal threshold for Δ-measure.

If yes, in line 8 the quality of cc is checked and if it is high the concept is saved as potentially best concept.

In lines 10-11 the concept cc is checked. If the value of its Δ-measure is smaller than the threshold θ, it is not Δ-stable concept and, thus, its preimages cannot be Δ-stable either.

Then, in lines 12–13 the concept cc is verified if its preimages can potentially be reported as the best concepts, i.e., that the optimimstic estimate for the quality function on its subconcepts (concepts with smaller but comparable extents) is large enough. If yes, the concept cc is pushed to the `queue` in line 14 for further processing.

Finally, in lines 15–16 the threshold θ is adjusted in such a way that ensures that the size of the `queue` is smaller than the memory limit L (the parameter of the algorithm). Let us discuss the adjustment in more details.

3.3 Adjustment of Δ threshold

The algorithm for adjusting the threshold is shown in Algorithm 3. First, in line 2 the algorithm removes all patterns that cannot generate concepts with high quality. Then, if necessary it increases the threshold θ in such a way, that the number of concepts with high Δ-measure is less then L.

This adjustment is polynomial, however, the computational complexity of the whole procedure is no more polynomial. Indeed, if the concept with the maximal projection number is expanded, then this procedure become a depth first order FCA algorithm, that requires only $O(|M|)$ memory to store concepts and thus if $L > |M|$ the adjustment procedure is never run and, thus, Algorithm 2 can iterate over all concepts. However, if one always selects the concept with the minimal projection number, then this procedure becomes the same as in Algorithm 1 with the polynomial complexity. Indeed, in this case one can consider the expansions of all concepts with the minimal projection number as one expansion and it become exactly Σοφια algorithm. Similarly, if only expansion of the concepts with small projection number is allowed ,i.e. if the projection number is in the interval $[i_{\min}, i_{\min} + k]$, where $i_{\min}$ is the minimal projection number, gives an intermediate worst-case computational complexity with a multiplier of 2^k. For small k it is small and algorithm can be considered polynomial.

The similar effect on computational complexity can be achieved by always expanding concepts with the largest extent. Such kind of concept expansion corresponds to subgroup discovery procedure. Indeed, the larger the extent, the

better the quality Q can be potentially attained on its subsets. Thus, there is a hope, that in practice Algorithm 2 can behave as an algorithm with polynomial computational complexity for subgroup discovery task.

Before proceeding to practical evaluation we should discuss how the best concepts should be registered, i.e. lines 8 and 12 of Algorithm 2.

3.4 Best concept registration

The most simple concept registration procedure can just verify that the quality of the input concept is larger than the quality of the currently known best concept. If the new concept is better, then it substitutes the previous best concept. This procedure has a problem when the Δ-measure threshold θ is adjusted. Indeed, let $\theta = 1$ and the best concept found so far be c_{best}, let $\Delta(c_{best}) = 1$. *If on the next step the stability threshold is adjusted, what should happen to* c_{best}*?*

If it is just preserved, then the result would not correspond to the final projection. Thus, this would invalidate any procedure that compute the statistical significance of the found subgroup [15]. Consequently, a special mechanism is needed allowing for defining the best stable concept found so far for any Δ-threshold ϑ higher than the current threshold. Consequently, any concept that is not dominated either by Δ-measure or by SD quality Q should be registered. Indeed, if for two concepts $\Delta(c_1) \leq \Delta(c_2)$ and $Q(c_1) \leq Q(c_2)$, i.e., c_1 is dominated by c_2, the concept c_1 cannot be the best SD-concept for any θ. By contrast, if $\Delta(c_1) < \Delta(c_2)$ and $Q(c_1) > Q(c_2)$, then for $\theta \leq \Delta(c_1)$ the concept c_1 is Δ-stable and since $Q(c_1) > Q(c_2)$ the concept c_1 is better than c_2 and can be reported as the best concept, however if $\Delta(c_1) < \theta \leq \Delta(c_2)$, then c_1 is no more Δ-stable and thus it cannot be reported as the best concept, but c_2 is still can be reported. Thus, all undominated concepts should be preserved. Thus, Algorithm 4 describes how one should register the best concepts.

In line 2 of Algorithm 4 an element of the `best` concept storage is found. This element is such that $\texttt{best}[i-1] < \Delta(cc) \leq \texttt{best}[i]$ (the set `best` of the best concepts is ordered w.r.t. Δ), i.e., $\texttt{best}[i]$ dominates cc w.r.t. Δ. Thus, if $Q(cc) < Q(\texttt{best}[i])$ the concept cc should not be registered (lines 3–4). Simlarly, in lines 14–15 if the potential of cc is small then it is not promising.

If the concept cc is not dominated by $\texttt{best}[i]$ it is inserted into `best` in lines 5–8. Finally, the quality of cc can be so high that it dominates some concepts from `best` (the concepts $\texttt{best}[j]$ for $j < i$ are already dominated w.r.t. the Δ-measure, thus, they should be compared by means of the quality function). The operations `best.FindInsertPositions`, `best.Insert`, and `best.Remove` are standard operations and can be efficiently implemented by means of red-black trees and other standard approaches.

4 Evaluation

Let us now check how the proposed approach behaves on real data. The approach is evaluated on `chess` dataset from the UCI machine learning repository [9].

```
1  Function best.Register(cc)
2  |  i ← best.FindInsertPosition(Δ(cc));
3  |  if Q(best[i]) ≥ Q(cc) then
4  |  |  return;
5  |  if Δ(best[i]) == Δ(cc) then
6  |  |  best[i] ← cc;
7  |  else
8  |  |  best.Insert(i,cc);
9  |  i ← i − 1;
10 |  while Q(best[i]) < Q(cc) do
11 |  |  best.Remove(i);
12 |  |  i ← i − 1;
13 Function best.IsPromissing(cc)
14 |  i ← best.FindInsertPosition(Δ_Proj(cc)(cc));
15 |  return Q(best[i]) < Q̄(cc);
```

Algorithm 4: Registration of undominated concepts w.r.t. Δ-measure and SD quality Q.

The dataset contains 3196 of objects, corresponding to positions in chess with the supervised information ('the whites win' and 'the whites do not win'). The dataset contains a huge number of concepts and thus it is a good candidate for computational efficiency test. For this task we run the subgroup discovery task for the classification quality function from [13] discussed in Section 1.3.

4.1 Computational time and the dataset size

In the first experiment the size of the dataset is varied from 200 objects to the whole dataset of 3196 objects. For every size a random sample from the original dataset is taken. The memory limit L is also varied from 100 to 100000. The result is shown in Figure 1a. Since it is interesting to see the difference w.r.t. large interval of possible dataset sizes it is given in the log-log scale. However, in order to judge the computational complexity of the approach the real values should be converted to relative values. In particular the time is measured in computational time needed for computing the smallest dataset with the same L. For example, for $L = 1000$ the computation time for the right most point (the whole dataset) is shown to be equal to 2.5, which means that it is in $2^{2.5} = 5.6$ times larger than the time needed for the smallest dataset for the same L. Then the algrithm can be considered linear or better if it is below the function $y = x$.

As it can be seen from the plot for all values of L the computational behaviour of the approach is not worser than linear time complexity. We can also note that the slope of all curves is never larger than 1, which also proves that it behaves linearly w.r.t. the dataset size.

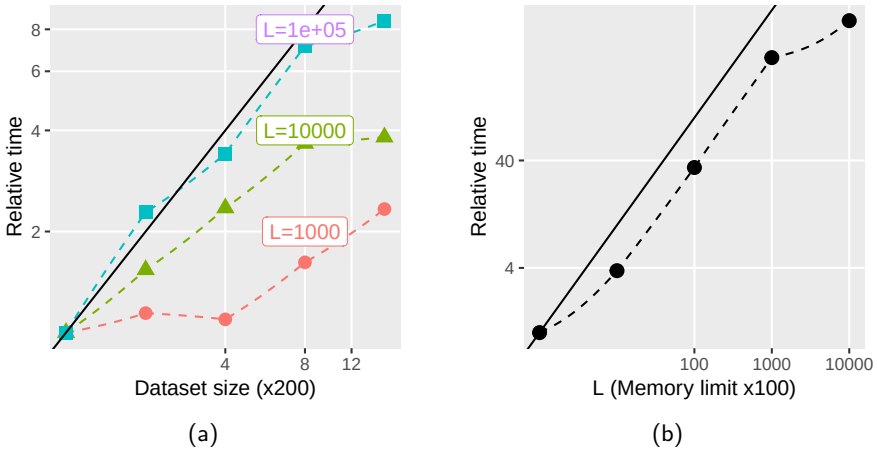


Fig. 1. Computation time w.r.t. dataset size and memory limit

4.2 Computational time and the memory limit

In the second experiment the whole dataset is taken and the memory limit is varying form 100 to 10^6. The result is shown in Figure 1b. As before it is given in the log-log space. As we can see the slope of this curve is also never larger than 1, i.e., the practical computational complexity of this approach is linear.

In both experiments we can see that the slope is much smaller than 1 till a certain moment. It could be explained by the fact that the memory limit is too large and is not totally used. Indeed, when the datasets are small in the first experiments, the total number of concepts is also small, and, thus, the availble memory is too large. However, when the size of the dataset is increased, the total number of concepts is also increased and then the memory limit L become to be important. Similarly, when in the second experiment the memory limit is large ($L = 10^6$), then the memory limit become to be less important.

4.3 Quality of the found concepts

Let us now briefly discuss how the quality of the found concepts depends on different memory limits on the whole dataset. Setting the memory limit to 100, allows finding the best concept with Δ measure of 106, with the extent size of 1002 and the quality of 0.121. By contrast, for $L = 10^6$, the Δ of the found concept is 3, the extent size is 1326 and the quality is 0.157. We see that if we can wait, a better concept can be found, however if the time is limited small L threshold allows finding interesting subgroups much faster.

Conclusion

In this paper a new approach to subgroup discovery was proposed. Although it is not possible to prove polynomial complexity of the algorithm, in practice

for subgroup discovery task it shows polynomial behaviour, which is extremly important for processing of large datasets.

References

1. Atzmueller, M.: Subgroup discovery. Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery **5**(1), 35–49 (2015)
2. Belfodil, A., Belfodil, A., Kaytoue, M.: Anytime subgroup discovery in numerical domains with guarantees. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 500–516. Springer (2018)
3. Boley, M., Goldsmith, B.R., Ghiringhelli, L.M., Vreeken, J.: Identifying consistent statements about numerical data with dispersion-corrected subgroup discovery. Data Mining and Knowledge Discovery **31**(5), 1391–1418 (2017)
4. Boley, M., Grosskreutz, H.: Non-redundant Subgroup Discovery Using a Closure System. In: Buntine, W., Grobelnik, M., Mladenić, D., Shawe-Taylor, J. (eds.) Machine Learning and Knowledge Discovery in Databases. pp. 179–194. Springer, Berlin, Heidelberg (2009)
5. Bosc, G., Boulicaut, J.F., Raissi, C., Kaytoue, M.: Anytime discovery of a diverse set of patterns with monte carlo tree search. Data mining and knowledge discovery **32**(3), 604–650 (2018)
6. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Efficient Mining of Subsample-Stable Graph Patterns. In: 2017 IEEE International Conference on Data Mining (ICDM). pp. 757–762. New Orlean, LA, USA (2017)
7. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Scalable Estimates of Concept Stability. In: Sacarea, C., Glodeanu, C.V., Kaytoue, M. (eds.) Formal Concept Analysis, LNCS, vol. 8478, pp. 161–176. Springer (2014)
8. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: Fast Generation of Best Interval Patterns for Nonmonotonic Constraints. In: Appice, A., Rodrigues, P.P., Santos Costa, V., Gama, J., Jorge, A., Soares, C. (eds.) Machine Learning and Knowledge Discovery in Databases, LNCS, vol. 9285, pp. 157–172. Springer (2015)
9. Frank, A., Asuncion, A.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. University of California, Irvine, School of Information and Computer Sciences (2010)
10. Ganter, B., Kuznetsov, S.O.: Pattern Structures and Their Projections. In: Delugach, H.S., Stumme, G. (eds.) Conceptual Structures: Broadening the Base, LNCS, vol. 2120, pp. 129–142. Springer (2001)
11. Ganter, B., Wille, R.: Formal Concept Analysis: Mathematical Foundations. Springer, 1st edn. (1999)
12. Kuznetsov, S.O.: On stability of a formal concept. Annals of Mathematics and Artificial Intelligence **49**(1-4), 101–115 (2007)
13. Lavrač, N., Kavšek, B., Flach, P., Todorovski, L.: Subgroup Discovery with CN2-SD. The Journal of Machine Learning Research **5**, 153–188 (2004)
14. Li, G., Zaki, M.J.: Sampling frequent and minimal boolean patterns: theory and application in classification. Data mining and knowledge discovery **30**(1), 181–225 (2016)
15. Pellegrina, L., Riondato, M., Vandin, F.: SPuManTE: Significant Pattern Mining with Unconditional Testing. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining-KDD. vol. 19 (2019)

Representation of Knowledge Using Different Structures of Concepts

Dmitry Palchunov[1,2], Gulnara Yakhyaeva[2]

[1] Sobolev Institute of Mathematics, Novosibirsk, Russian Federation
palch@math.nsc.ru
[2] Novosibirsk State University, Novosibirsk, Russian Federation
gul_nara@mail.ru

Abstract. The paper is devoted to the problem of knowledge representation using concepts of different levels of generality. Model-theoretical methods are being developed for translating data and knowledge presented in the language of low-level concepts into knowledge presented using concepts of a high level of generality. The formalization of knowledge is carried out in the terms of FCA, which allows us to move from low-level concepts to more general concepts and, as a result, generate new, more general knowledge about the domain.

Keywords: Knowledge Representation, FCA, Precedent Model, Boolean-valued Model, Semantic Domain Model

1 Introduction

Knowledge representation is a central part of the development of intelligent systems. Today, there are several different methodologies for representing knowledge: frames, semantic networks, production systems, etc. [1].

One of the most developed and popular methodologies are the logical knowledge representation. Logical systems are highly expressive. Representation of the data array in the form of an algebraic system makes it possible to work with two levels of information consideration: the level of the initial data on which the analysis is performed, and the level of knowledge – generalized laws formulated in the form of sentences of first-order predicate logic. Consideration of a set of different situations (precedents) of the given domain as a class of algebraic systems allows us to work with statistical knowledge about the domain [2].

On the other hand, FCA is a powerful tool for presenting and processing knowledge [3-5]. FCA methods are widely used for ontology engineering [6], machine learning [7], semantic web [8] and so on. The connection between FCA and the theory of axiomatizable classes of algebraic systems was shown in [9]. There was described a method for constructing a formal context for the case model (using the notion of a Boolean-valued model) and a method of transitioning to an object-clarified context [10, 11].

Recently, much attention has been paid to approaches to concept mining simplification. For example, in [12], the concept indices were studied and their applications for evaluating interestingness measures of formal concepts.

In this paper we consider various levels of knowledge representation, which are presented using concepts of varying degrees of generality. For example, knowledge of a lower level of generality is:

– symptoms of diseases, test results;

– specific numerical data on exchange rates, oil prices;

Knowledge of the high level of generality is, for example:

– patients' diseases, syndromes, complications of diseases;

– economic forecasts, expectations of stability or instability, currency crises, etc.

Knowledge containing statements of different levels of generality is formalized in the form of algebraic systems of different signatures. For these systems, formal contexts are constructed that describe them. The properties of lattices of constructed formal contexts are studied.

2 Semantic Domain Model

We start the formalization with a finite set $\mathbb{E} = \{E_1, \dots, E_n\}$ of domain precedents. We have a set (signature) σ of low-level concepts of this domain. Each domain precedent $E_i \in \mathbb{E}$ is formalized as a model $E = \langle A, \sigma \rangle$ [13, 14].

Next, we will bring together knowledge about of all precedents from the class $\mathbb{E}$. For this we need to enrich the signature σ with a set of constants $\boldsymbol{C_A} = \{c_a \mid a \in A\}$, i.e. to take $\sigma_A = \sigma \cup \boldsymbol{C_A}$. Now we may consider the set $S_a(\sigma_A)$ of all atomic sentences of the signature σ_A as a formalization of low-level concepts of the object domain and the set $S(\sigma_A)$ of all sentences of the signature σ_A as a formalization of all possible concepts of the domain.

Definition 1 [15]. *Ordered triple* $\mathfrak{A}_{\mathbb{E}} \leftrightharpoons \langle A, \sigma_A, \tau \rangle$ *is called* ***Precedent Model*** *generated by the set of the precedents* $\mathbb{E}$*, if for any sentences* $\varphi(c_{a_1}, \dots, c_{a_n}) \in S(\sigma_c)$ *we have*

$$\tau\left(\varphi\left(c_{a_1}, \dots, c_{a_n}\right)\right) = \{E \in \mathbb{E} \mid E \vDash \varphi(a_1, \dots, a_n)\}.$$

The definition of ***Boolean-valued Model*** one can find in [15].

It was shown in [15] that for any Precedent Model it is possible to construct a Boolean-valued Model isomorphic to it.

When there is a formalization of individual precedents of the given domain, it is often necessary to describe low-level concepts. So, for example, in the subject domain of computer security [16] we formalized the knowledge obtained from texts in natural language. Of these texts, it was often possible to single out very small (in volume) concepts. For example, in some precedent it was said about an attack using the *Randex virus*, in another precedent the *CMJ virus* was used, and in the third - *MrKlunky*. For each of these viruses it will be difficult to identify any regularities, since the ratio of the number of attack precedents, where there was a mention of this

particular virus to the total number of precedents, is very small. Thus the evalution μ on the predicates formalizing these concepts will be very close to zero.

On the other hand, all these viruses have many common characteristics, so it's reasonable to combine them into one, more general concept "*TSR Viruses*" and to study the properties of this new concept. Note that each such concept is expressible in the signature σ through some formula.

So, we select the set of formulas $F \subseteq F(\sigma)$ and enrich the signatureσ with the set of new predicts $\boldsymbol{P}_F = \{P_\varphi \mid \varphi \in F\}$. Let $\sigma_F = \sigma_A \cup \boldsymbol{P}_F$. Next, we extend the Boolean-valued model $\mathfrak{A}_\mathbb{B}$ to the signature σ_F, i.e. put

$$\mathfrak{A}'_\mathbb{B} = \mathfrak{A}_\mathbb{B} \restriction \sigma_F,$$

where the estimation $\tau' : S(\sigma_F) \to \mathbb{B}$ is redefined as follows. For each $P_\varphi(x_1, \dots, x_n) \in \boldsymbol{P_F}$ and for any elements$a_1, \dots, a_n \in A$ we have

$$\tau'\left(P_\varphi(c_{a_1}, \dots, c_{a_n})\right) = \tau\left(\varphi(c_{a_1}, \dots, c_{a_n})\right).$$

Now we can remove low-level concepts from the signature σ_F leaving only concepts of a higher level, i.e. put $\sigma^* = \boldsymbol{C_A} \cup \boldsymbol{P_F}$. The model $\mathfrak{A}''_\mathbb{B} = \mathfrak{A}'_\mathbb{B} \restriction \sigma^*$ is a formalization of the subject domain at a higher level.

Let $\mathfrak{A}_\mathbb{B}$ be an atomic Boolean-valued model. Denote

$$At(\mathbb{B}) = \{a \in \mathbb{B} \mid a \text{ is an atom}\}.$$

Consider the formal context

$$\boldsymbol{K}(\mathfrak{A}_\mathbb{B}) = \left(At(\mathbb{B}),\ S_a(\sigma_\mathrm{A}),\ I_\tau\right),$$

where

$$a\ I_\tau\ \varphi \Leftrightarrow\ a \le \tau(\varphi).$$

We say that the formal context $\boldsymbol{K}(\mathfrak{A}_\mathbb{B})$ describing the Boolean-valued model $\mathfrak{A}_\mathbb{B}$ [17].

In the next section, we will consider various generation algorithms of the model $\mathfrak{A}''_\mathbb{B}$ from the model $\mathfrak{A}_\mathbb{B}$ and show how the formal contexts describing them will change.

3 Formal contexts representing higher-level concepts

When forming a set of $\boldsymbol{P}_F$ of higher-level concepts, first of all, we consider the possibility of obtaining these concepts directly from the formal context itself that describes this Boolean-valued model.

Definition 2. *Consider the formal context* $\boldsymbol{K} = (G, M, I)$. *Denote by* $\widetilde{M}$ *the set of contents of all concepts of the context* $\boldsymbol{K}$, *i.e.*

$$\widetilde{M} = \left\{B \subseteq M \middle| B^{\downarrow\uparrow} = B\right\}.$$

Consider the formal context $\widetilde{\boldsymbol{K}} = (G, \widetilde{M}, \widetilde{I})$, *where for any object* $g \in G$ *and for any* $B \in \widetilde{M}$ *we have*

$$g \widetilde{I}\, B\ \Leftrightarrow\ g \in B^{\downarrow}\ \textit{in the context}\ \boldsymbol{K}.$$

Proposition 1. *The concept lattices generated by the formal contexts* $\boldsymbol{K} = (G, M, I)$ and $\tilde{\boldsymbol{K}} = (G, \tilde{M}, \tilde{I})$ are isomorphic, i.e. $\underline{\mathfrak{B}}(\boldsymbol{K}) \cong \underline{\mathfrak{B}}(\tilde{\boldsymbol{K}})$.

Thus, this Proposition shows that in order to move to the meta level it is not enough to have only "internal" information contained in a formal context. So, for example, in [16] it was proposed to obtain "external" information through cauterization of the set of objects' properties and, when moving to the meta-level, consider generalized properties that characterize different clusters.

We describe this approach in the terms of FCA [18].

Let an equivalence relation $\sim$ be defined on the set M. We will denote by $[m]_{\sim}$ the equivalence class generated by the element m and denote by $M/_{\sim}$ the factor-set.

Definition 3. Consider the formal context $\boldsymbol{K} = (G, M, I)$ *and the equivalence relation* $\sim$ *defined on the set* M.

1. *By* $\boldsymbol{K}_{\wedge} = (G, M/_{\sim}, I_{\wedge})$ *we denote the formal context in which*

$$g\ I_{\wedge}\ [m]_{\sim} \ \Leftrightarrow\ \forall n \in [m]_{\sim}\ \ gIn.$$

2. *By* $\boldsymbol{K}_{\vee} = (G, M/_{\sim}, I_{\vee})$ *we denote the formal context in which*

$$g\ I_{\vee}\ [m]_{\sim} \ \Leftrightarrow\ \exists n \in [m]_{\sim}\ \ gIn.$$

Proposition 2. *Consider the formal contexts* $\boldsymbol{K} = (G, M, I)$ *and* $\boldsymbol{K}_{\wedge} = (G, M/_{\sim}, I_{\wedge})$. *Then the lattice* $\underline{\mathfrak{B}}(\boldsymbol{K}_{\wedge})$ *is a sublattice of the lattice* $\underline{\mathfrak{B}}(\boldsymbol{K})$.

Theorem 1. *Consider the formal contexts* $\boldsymbol{K} = (G, M, I)$ *and* $\boldsymbol{K}_{\vee} = (G, M/_{\sim}, I_{\vee})$. *We define a map* $h\colon \underline{\mathfrak{B}}(\boldsymbol{K}) \to \underline{\mathfrak{B}}(\boldsymbol{K}_{\vee})$ *as follows:* $h\big((A, B)\big) = \big(A^{\uparrow\downarrow}, A^{\uparrow}\big)$. *Then for any* $(A_1, B_1), (A_2, B_2) \in \underline{\mathfrak{B}}(\boldsymbol{K})$ *we have:*

1. $(A_1, B_1) \leq (A_2, B_2) \Rightarrow h\big((A_1, B_1)\big) \leq h\big((A_2, B_2)\big)$;
2. $h\big((A_1, B_1)\big) \cap h\big((A_2, B_2)\big) = h\big((A_1, B_1) \cap (A_2, B_2)\big)$;
3. $h\big((A_1, B_1)\big) \cup h\big((A_2, B_2)\big) \leq h\big((A_1, B_1) \cap (A_2, B_2)\big)$.

Thus, it follows from the Theorem 1 that the map $h\colon \underline{\mathfrak{B}}(\boldsymbol{K}) \to \underline{\mathfrak{B}}(\boldsymbol{K}_{\vee})$ is a homomorphism of the lower semilattices.

Corollary 1. A map $h\colon \underline{\mathfrak{B}}(\boldsymbol{K}) \to \underline{\mathfrak{B}}(\boldsymbol{K}_{\vee})$ is homomorphism of the lattices if and only if for any $A_1, A_2 \in G$ follow condition is satisfied:

$$(A_1 = A_1'' \ and\ A_2 = A_2'') \Rightarrow A_1 \cup A_2 = (A_1 \cup A_2)''.$$

Note that in the general case, the homomorphism h may not possess the properties of injectivity and surjectivity. This means that when moving from the context $\boldsymbol{K}$ to the context $\boldsymbol{K}_{\vee}$, on the one hand, some concepts will be combined into more general concepts, and on the other hand, new concepts will appear. On the other hand, when moving from the context $\boldsymbol{K}$ to the context $\boldsymbol{K}_{\wedge}$, some concepts will be "forgotten" only.

The results obtained in the paper may be applied to the development of ontological [19-22] and semantic [23-28] technologies.

References

1. Archana, P., Sarika, J. **:** Formalisms of Representing Knowledge**.** Procedia Computer Science**,** Volume 125, 2018, Pages 542-549
2. Palchunov, D.E., Tishkovsky, D.E., Tishkovskaya, S.V., Yakhyaeva G.E. Combining logical and statistical rule reasoning and verification for medical applications \\ Proceedings - 2017 International Multi-Conference on Engineering, Computer and Information Sciences, SIBIRCON 2017 , 18-22 September 2017, Novosibirsk, Russia.
3. Kuznetsov, S.O., Poelmans, J.: Knowledge representation and processing with formal concept analysis. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 3 (3), 200-215 (2013)
4. Poelmans, J., Kuznetsov, S.O., Ignatov, D.I., Dedene, G.: Formal concept analysis in knowledge processing: A survey on models and techniques. Source of the Document Expert Systems with Applications, 40 (16), 6601-6623 (2013)
5. Kuznetsov SO, Obiedkov SA, Roth C. Reducing the representation complexity of lattice-based taxonomies. In: Priss U, Polovina S, Hill R, eds. Proceedings of the ICCS 2007, LNAI 4604; July 22–27; Sheffield, UK. Berlin, Heidelberg: Springer; 2007, 241–254.
6. Makhalova, T., Kuznetsov, S.O., Napoli, A.: MDL for FCA: is there a place for background knowledge? CEUR, pp. 45–56 (2018).
7. Maddouri M. Towards a machine learning approach based on incremental concept formation. Intell Data Anal 2004, 8:267–280.
8. Sertkaya B. A survey on how description logic ontologies benefit from FCA. Proceedings of CLA; 2010, 2–21.
9. Palchunov, D.E.: Lattices of Relatively Axiomatizable Classes. In: Kuznetsov, S.O., Schmidt, S. (eds.) ICFCA 2007, LNAI, vol. 4390, pp. 221–239. Springer, Heidelberg (2007).
10. Palchunov D., Yakhyaeva G., Dolgusheva E. Conceptual Methods for Identifying Needs of Mobile Network Subscribers \\ CEUR Workshop Proceedings 1624, c. 147-160
11. Palchunov D., Yakhyaeva G. Application of Boolean-valued models and FCA for the development of ontological model \\ CEUR Workshop Proceedings, 1921, c. 77-87
12. Kuznetsov S., Makhalova T. On interestingness measures of formal concepts // *Information Sciences*. 2018. No. 442–443. P. 202-219.
13. C.C.Chang, H.J.Keisler. Model theory. North-Holland Publishing Company, New York, 1973.
14. Yu.L.Ershov, E.A.Palutin. Mathematical logic. Walter de Gruyter, Berlin and New York, 1989.
15. D. Pal'chunov and G.Yakhyaeva, "Fuzzy logics and fuzzy model theory," Algebra and Logic, vol. 54, no. 1, pp. 74-80, 2015.
16. Yakhyaeva G.E. Application of Boolean Valued and Fuzzy Model Theory for Knowledge Base Development \\ 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia, 2019, pp. 0868-0871.
17. Palchunov D.E., Yakhyaeva G.E.: Integration of Fuzzy Model Theory and FCA for Big Data Mining. In: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), pp. 0961–0966. IEEE Press (2019).
18. B.Ganter, R.Wille. Formal Concept Analysis. Mathematical foundations. Springer-Verlag, Berlin, Heidelberg (1999).
19. Allemang D., Hendler J. Semantic Web for the Working Ontologist. Morgan Kaufmann, (2008).

20. Handbook on Ontologies, S. Staab, R. Studer (eds), 2nd edition, Springer-Verlag 2009, pp. 509 – 528.
21. Shen H., Hu J., Zhao J., Dong J. – Ontology-based Modeling of Emergency Incidents and Crisis Management // Department of Management Information Systems, Shanghai Jiao Tong University, Shanghai, 2010.
22. Gutiererz F., Dou D., Fickas S., and Griffiths G. – Online Reasoning for Ontology-Based Error Detection in Text // Computer and Information Science Department University of Oregon, 2014.
23. Parreiras F.S. Semantic Web and Model-Driven Engineering. Wiley-IEEE Press, 2012.
24. Szeredi P., Lukácsy G., Benkő T. The Semantic Web Explained: The Technology and Mathematics behind Web 3.0. Cambridge University Press, 2014.
25. Bernard E., Introduction to Semantic Web Rule Language – SWRL. – // Aix-Marseille Université (AMU) Polytech-Marseille, Nov. 2017.
26. Gumirov V.S., Matyukov P.Y., Palchunov D.E. Semantic Domain-specific Languages // In: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), pp. 0955–0960. IEEE Press (2019).
27. Galieva A.G., Palchunov D.E. Logical Methods for Smart Contract Development // In: 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), pp. 0881–0885. IEEE Press (2019).
28. Galitsky B., Developing Enterprise Chatbots: Learning Linguistic Structures. // Springer, 2019.

The study of the relationship between publications in social networks communities via formal concept analysis

Kristina Pakhomova and Alina Belova

Siberian Federal University, Kraskoyarsk, Russia
kpahomova@yandex.ru, alina_belova94@mail.ru

Abstract. Nowadays the users generate considerable amount of information in the Internet. Therefore, they have a deal with the issue of how to retrieve the required information perhaps much more relevant than he/she supposed. Moreover, the mix of different data sources usually muddles users in the case of searching for the instructive information. The authors of this paper will introduce the approach based on text processing and formal concept analysis in order to structure the information from a variety of sources, particularly, social media communities. Additionally, they will clarify the relations between community posts with the same topic, where these relations become a recommendation tool for the user's decision making. In conclusion, the authors will build a diagram that will be a convenient visualization tool in an effort to structure information that was obtained from various sources.

Keywords: Formal Concept Analysis · Semantic Analysis · Social Network · Data Mining · Community topic.

1 Introduction

Due to a rapidly increasing amount of data which is generated by users of the Internet, how to deal with this data becomes the most important issue. For instance, in 2019, 4.39 billion people were registered in services provided by social networks, which is 366 million (9%) more than in January 2018 (information provided by 'We Are Social agency' and 'Hootsuite' service [9]). Social networking services contain information that includes a variety of data, for instance, text and media. Obviously, text information is publications in social networking communities, each of those are described by a wide range of topics. In this case, users need to quickly filter and analyze a large amount of information. Currently, the filtering may be carried out in two ways: automatically, by using data mining methods, and manually by the user. However, the relevant result of the user request is an extremely important issue, but also the speed of request and visualization of the result has value. According to the wide range of data mining methods, the authors will explain a mathematical approach - formal concept analysis (FCA) that satisfies the above criteria.

Current research that based on the analysis of social network data is presented in the following works [5,6,7,8]. It is worth noting that the most significant work is the one [8], in which scientists deal with social media data via FCA.

This paper includes four-section: an introduction, approach explanation, experiment computation, and conclusion. Firstly, the authors will explain briefly the theory of FCA and basics semantic analysis methods.Secondly, the authors propose a solution based on methods of semantic data analysis and FCA. Finally, they will explain an experiment implementation based on the social network dataset and also they will propose to visualize the results of computation by using concepts lattice as a diagram, in which each circle will be marked in a certain color.

2 Computation approach

2.1 Brief review about Semantic Analysis

In order to study the text data ,in particular, the meaning of the text obviously the authors deal with semantic analysis [11,12,13]. According to its theory in general, the text should be subjected to the basic text manipulation methods such as tokenize, lemmatize, and etc. We apply semantic analysis in order to compute the set of keywords that will be related to the posts with the common topic. Where set of posts $G = \{g_1, g_2, ..., g_n\}$, $N = 0, ..., n$, then the set of words and symbols is $W = \{w_1, w_2, ..w_p\}$, $P = 0, ..., p$ when the word includes in text $W \subseteq G$ for $g \in G$ and $w \in W$. We deal with tokenize, lemmatizer, removing stop words and parts of speech other than nouns in order to compute keywords. So the final set of keywords is M which satisfy $M \subseteq W$, where $M = \{m_1, m_2, ..., m_l\}$, $L = 0, ..., l$.

2.2 Brief review of Formal Concept Analysis

FCA was invented by Germany mathematic Rudolf Wille in 1981 [2]. According to Wille paper, after 1996 Wille and Ganter clarify the theory of formal concept [3].

According to Wille and Ganter paper that introduce to formal context is (G, M, I), where G - the set of objects, M - set of attributes, binary relationship between objects and attributes explains $I \subseteq G \times M$ for $g \in G$, $m \in M$ when gIm, if object g has an attribute m.

Formal concept is pair $(A, B) : A \subseteq G$, $B \subseteq M$ satisfy the Galois connection between $(2^G, \subseteq)$ and $(2^M, \subseteq)$, $A^{'} = B$ and $B^{'} = A$. Where A and B - extent and intent of formal concept (A, B). The ordered set of all formal concept forms is called the concept lattice $\underline{\mathrm{B}}\ (G, M, I)$ [1].

2.3 Concept lattice building

In a previous subsection, the authors have defined the main mathematical method founded on formal concept computation. Therefore, the ordered concepts build the concept lattice, which is visualised by diagram, Hasse. Every set of formal concepts has a great common subconcept as supremum. Its extent consists of those objects that are common to all extents of the set. Every set of formal concepts has a least common superconcept, the intent of which comprises all attributes which all objects of that set of concepts have. Additionally, ordered this way the set should satisfy the axioms defining a lattice, there are commutative, associative, absorption laws. Thus, a complete lattice is an ordered poset in which all subsets have both a supremum and an infimum [4]. This diagram consists of main elements such as circles are set of formal concepts, lines explain the relation between formal concepts and labels. Notably, an attribute can be reached from an object via an ascending path according to subconcept-superconcept hierarchy. Additionally, it satisfies if and only if the object has the attribute.

Each post includes specific metadata that describe users' personal interest in the particular post, they are likes and reposts. Concerning of those measures we want to compute for every concept the average values of likes and reposts, and after we are going to clustering the ordered set of concepts by using k-means (KMeans) [14] . This also will assist to fix the color of each circle of the diagram in order to visualize the clusters of formal concepts.

3 Computation experiment

The authors have investigated the social network dataset concerning the common topic that takes place in the social network communities. Thus, this dataset includes information about Id post, the text of the post, value of users' attitudes, and value of reposts (see Tab. 1). It was obtained from 'Vk' social network, It community, which text of the posts in Russian and English [10].

Table 1. Experimental dataset

Id Post	Text	Value of likes	Value of reposts
1297322	$g1$	6	0
1297419	$g2$	41	5

In order to compute keywords,the authors used semantic analysis which was explained before. We computed the formal context according to set of key-words will be presented as attributes of a formal context and the set of objects - a number of posts (see Tab. 2).

Table 2. Example of formal context

Id post	microsoft	httpamp	remote	...	javascript
1297322	1	1	0	...	0
...	...	...	...	...	...
1297419	0	1	0	...	1

According to FCA theory the set of formal concepts was computed. Although the set of posts includes no more than 100 items, the number of obtained formal concepts is quite huge,approximately 800 items. Moreover, for each formal concept the average of users' likes and average of reposts were computed (see Tab. 3).

Table 3. Example of formal concepts

Extent	Intent	color like	color repost
1297430, 1297330, 1297313, 1297235, 1296857	javascript, httpamp	blue	green
1296857, 1296779	javascript, help	blue	green
...	...	...	...
1297419, 1297266	javascript, developer, company	green	red
1297419, 1296779	javascript, developer,middle	blue	green
1297266, 1296779	javascript, developer, senior	green	red

However, the great number of ordered formal concepts build a huge diagram, for this reason, the authors have explained only the set of concepts and their clustering. According to Tab. 3 in which an object $'javascript'$ takes place by following the next set of formal concepts. We deal with diagram in order to concentrate on users preferences,for instance, the user who takes an interest in hiking a job or career development and he \ she takes an interest on $'javascript'$ may be recommended the next set of posts: 1297419, 1297266, 1296779 and etc. This argument makes sense due to lattice lines properties that explain the relation between formal concepts.

By using the measures (likes and reposts) we computed three concept clusters, where centroids according to likes are $(color\ cluster : value) : (red : 108.333, blue : 55.3985, green : 24.227)$, and reposts are $(green : 9.1345, red : 3.254)$. Additionally, each circle of the diagram has its own color that explains a cluster number. This opportunity allows users to visualize their searching and to rank posts according to measures. For instance, the user concentrates on $'javascript'$ therefore high priority has 1296779 and after 1296779,1297419 as stated by post likes. However, measure repost supports a few users so its values explain only two clusters but another hand this measure makes more sense than likes measure in the opinion of an issue of ranking.

4 Conclusions and future work

This paper has discussed the approach which manipulates the social network dataset by using FCA, particularly, it assists the user of social media to deal with communities' posts. The authors take into account the specific framework of a dataset that is satisfied with a variety of social services. Moreover, the authors concentrated on such criteria as accessible, high quality, immediate, and relevant information that can be provided to the user by his/her request. Although this approach partially satisfies this number of criteria, so it will be tried to improve in the future by using FCA advantages.

References

1. Ferré,S.,Huchard, M.,Kaytoue,M., Kuznetsov, S. O.,Napoli A.: Formal Concept Analysis: From Knowledge Discovery to Knowledge Processing. A Guided Tour of Artificial Intelligence Research, pp. 411–445, Springer Nature Switzerland (2020).
2. Wille, R.: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. Ordered Sets, pp. 445–470, Springer Netherlands (1982).
3. Ganter, B., Wille, R.: Formal Concept Analysis. Berlin.Heidelberg: Springer-Verlag (1999).
4. Davey, B.A., Priestley, H.A.: Introduction to Lattices and Order. Cambridge: Cambridge University Press (2002).
5. Ignatov, D.I., Kuznetsov, S.O.: Concept-based Recommendations for Internet Advertisement. Palacky University, vol. 433, pp. 157–166, Olomouc (2008).
6. Medina, J., Pakhomova, K., Ramirez-Poussa, E.: Recommendation Solution for a Locate-Based Social Network via Formal Concept Analysis. Trends in Mathematics and Computational Intelligence. Studies in Computational Intelligence, vol. 796, pp. 131–138, Springer, Cham (2019).
7. Cordero, P. et al.: Knowledge discovery in social networks by using a logic-based treatment of implications. Knowledge-Based Syst, Elsevier B.V., vol. 87, pp. 16–25, (2015).
8. Missaoui, R., Kuznetsov, S., Obiedkov, S.: Formal Concept Analysis of Social Networks. Springer International Publishing (2017).
9. LNCS Homepage, https://www.web-canape.ru/business/vsya-statistika-interneta-na-2019-god-v-mire-i-v-rossii/. Last accessed 19 Jun 2020
10. LNCS Homepage, https://vk.com/habr. Last accessed 19 Jun 2020
11. Bird,S.,Klein, E.,Loper, E.: Natural Language Processing with Python. O'Reilly Media, Inc.,Gravenstein Highway North, Sebastopol (2009).
12. Manning,C.,Schütze,H.: Foundations of Statistical Natural Language Processing. MIT Press. Cambridge (1999).
13. Russell,S.,Norvig,P.: Artificial Intelligence: A Modern Approach, 3rd Edition. Prentice Hall (2010).
14. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In Proc. 5th Berkeley Symp. on Math. Statistics and Probability, pp 281—297,(1967).

Patterns via Clustering as a Data Mining Tool

Lars Lumpe[1] and Stefan E. Schmidt[2]

Institut für Algebra,
Technische Universität Dresden
larslumpe@gmail.com[1], midt1@msn.com[2]

Abstract. Research shows that pattern structures are a useful tool for analyzing complex data. In our paper, we present a new general framework of using pattern structures as a data mining tool and as an application of the new framework we show a way to handle a classification problem of red wines.

1 Introduction

Pattern structures within the framework of formal concept analysis have been introduced in [4]. Since then they have been the subject of further investigations like in [2, 9, 11, 10, 12] and have turned out to be a useful tool for analyzing various real-world applications (cf. [4–8]). In this paper, we want to present a new application. But first we will introduce a general way to construct a pattern structure. Then we are going to use several clustering algorithm to find important pattern in it. Further we will use this pattern to build a model to solve a classification problem. In particular, we will take the dataset from [3] and train an algorithm to predict the quality of red wines.

2 Preliminaries

We start with some general definitions:

Definition 1 (restriction)**.** Let $\mathbb{P} := (P, \leq_{\mathbb{P}})$ be a poset, then for every set U the poset

$$\mathbb{P} \mid U := (U, \leq_{\mathbb{P}} \cap (U \times U))$$

is called the **restriction** of $\mathbb{P}$ onto U.

If we consider a poset of patterns, it often arises as a dual of a given poset.

Definition 2 (opposite or dual poset)**.** Let $\mathbb{P} := (P, \leq_{\mathbb{P}})$ be a poset. Then we call

$$\mathbb{P}^{op} := (P, \geq_{\mathbb{P}}) \text{ with } \geq_{\mathbb{P}} := \{(q, p) \in P \times P \mid p \leq q\}$$

the **opposite or dual** of $\mathbb{P}$.

Definition 3 (Interval Poset)**.** Let $\mathbb{P} := (P, \leq_{\mathbb{P}})$ be a poset. Then we call

$$\mathrm{Int}\mathbb{P} := \{[p,q]_{\mathbb{P}} \mid p, q \in P\} \text{ with } [p,q]_{\mathbb{P}} := \{t \in P \mid p \leq_{\mathbb{P}} t \leq_{\mathbb{P}} q\}$$

the **set of all intervals** of $\mathbb{P}$, and we refer to $\mathbb{I}\mathrm{nt}\mathbb{P} := (\mathrm{Int}\mathbb{P}, \subseteq)$ as the interval poset of $\mathbb{P}$.

Remark: (i) Let $\mathbb{P}$ be a poset. Then $\mathbb{I}\mathrm{nt}\mathbb{P}$ is a lower bounded poset, that is, $\emptyset$ is the least element of $\mathbb{I}\mathrm{nt}\mathbb{P}$, provided $\mathbb{P}$ has at least 2 elements. Furthermore if $\mathbb{P}$ is a (complete) lattice than so is $\mathbb{I}\mathrm{nt}\mathbb{P}$.
(ii) If A is a set of attributes than $(\mathbb{R}^A, \leq) := (\mathbb{R}, \leq)^A$ is a lattice. With (i) it follows that $\mathbb{I}\mathrm{nt}(\mathbb{R}^A, \leq)$ is a lower bounded lattice.

Definition 4 (kernel operator)**.** A **kernel operator** on a poset $\mathbb{P} := (P, \leq)$ is a map $\gamma : P \to P$ such that for all $x, y \in P$:

$$kx \leq y \Leftrightarrow kx \leq ky \tag{1}$$

A subset ζ of P is called a **kernel system** in $\mathbb{P}$ if for every $x \in P$ the restriction of $\mathbb{P}$ onto $\{t \in \zeta \mid t \leq x\}$ has a greatest element.

Remark: A closure operator on $\mathbb{P} := (P, \leq)$ is defined as a kernel operator on $\mathbb{P}^d$, and a closure system in $\mathbb{P}$ is defined as a kernel system in $\mathbb{P}^d$.

The main definitions of FCA are based on a binary relation I between a set of so called objects G and a set of so called attributes M. However, in many real-world knowledge discovery problems, researchers have to deal with data sets that are much more complex than binary data tables. In our case, there was a set of numerical attributes, such as the amount of acetic acid, density, the amount of salt, etc., describing the quality of a red wine. To deal with this kind of data, pattern structures are a useful tool.

Definition 5 (pattern setup, pattern structure)**.** A triple $\mathcal{P} = (G, \mathbb{D}, \delta)$ is a **pattern setup** if G is a set, $\mathbb{D} = (D, \sqsubseteq)$ is a poset, and $\delta : G \to D$ is a map. In case every subset of $\delta G := \{\delta g \mid g \in G\}$ has an infimum in $\mathbb{D}$, we will refer to $\mathcal{P}$ as **pattern structure**.

For pattern structures, an important complexity reduction is often provided by so-called o-projections:

Proposition 1 (o-projection)**.** For a pattern structure $\mathcal{P} := (G, \mathbb{E}, \varepsilon)$ and a kernel operator κ on $\mathbb{E}$, the triple

$$opr(\mathcal{P}, \kappa) := (G, \mathbb{D}, \delta)$$

with $\mathbb{D} := \mathbb{E}|D$ where $D := \kappa E$ and

$$\delta : G \to D, g \mapsto \kappa(\varepsilon g)$$

is a pattern structure, called the **o-projection** of $\mathcal{P}$ via κ (see [2]).

3 Construction of a pattern structure

The following definition establishes the connection between pattern setups and pattern structures.

Definition 6 (embedded pattern structure)**.** Let $\mathbb{E}$ be a complete lattice and $\mathcal{P} := (G, \mathbb{D}, \delta)$ a pattern setup with $\mathbb{D} := \mathbb{E}|D$. Then we call

$$\mathcal{P}_e := (G, \mathbb{E}, \mathbb{D}, \overline{\delta}) \text{ with } \overline{\delta} : G \to L, g \mapsto \delta g$$

the **embedded pattern structure**. We say the pattern setup $\mathcal{P}$ is **embedded** in the pattern structure $(G, \mathbb{E}, \overline{\delta})$.

This definition shows, that it is possible to build a pattern structure from a given pattern stup. The construction below is another demonstration of how to build a pattern structure from a pattern setup.

Construction 1. Let G be a set and let $\mathbb{L} := (L, \leq)$ be a poset, then for every map $\varrho : G \to L$ the **elementary pattern structure** is given by:

$$\mathcal{P}_\varrho := (G, \mathbb{E}, \varepsilon) \text{ with } \mathbb{E} := (2^L, \supseteq) \text{ and } \varepsilon : G \to 2^L, g \mapsto \{\varrho g\}.$$

Hence, the pattern setup $(G, \mathbb{L}, \varrho)$ is embedded in the pattern structure $(G, \mathbb{E}, \varepsilon)$. In many cases this construction leads to a large set of patterns. Therefore we need the following: Let ζ be a closure system in $\mathbf{2}^L := (2^L, \subseteq)$, that is, ζ is a kernel system in $\mathbb{E}$ and let $\gamma : 2^L \to 2^L$ be the associated closure operator of ζ w.r.t. $\mathbf{2}^L$. Thus, γ is a kernel operator on $\mathbb{E}$. Then $(G, \mathbb{D}, \delta)$ is a pattern structure for $\mathbb{D} := \mathbb{E}|\zeta$ and

$$\delta : G \to D, g \mapsto \gamma\{g\}.$$

Indeed the map

$$\psi : 2^L \to \zeta, X \mapsto \gamma X$$

is a residual map from $\mathbb{E}$ to $\mathbb{D}$ with $\delta = \psi \circ \varepsilon$.
By the above proposition, $(G, \mathbb{D}, \delta)$ is a pattern structure, since

$$opr(\mathcal{P}_\varrho, \gamma) = (G, \mathbb{D}, \delta)$$

is the o-projection of $\mathcal{P}_\varrho$ via γ.

4 Connection to Data Mining and to our Dataset

In this section we describe how we use the construction 1 to handle a typical data mining classification problem. We train a model to predict classes of red wines. But first we give an insight to our data.

4.1 Red Wine Dataset

To apply our previous results on a public data set we choose the red wine data set from [3]. There are 1599 examples of wines, described by 11 numerical attributes. The input includes objective tests (e.g. fixed acidity, sulphates, PH values, residual sugar, chlorides, density, alcohol...) and the output is based on sensory data (median of at least 3 evaluations made by wine experts). Each expert graded the wine quality between 0 (very bad) and 10 (very excellent). For our purpose, we established a binary distinction where every wine with a quality score above 5 is classified "good" and all below as "bad". This led to a set of 855 positive and 744 negative examples. We split the data into a training set (75% of the examples) and a test set (25% of the examples).

4.2 Describing the Proceeding

Many data mining relevant data sets (like the red wine dataset) can be described via an evaluation matrix:

Definition 7 (evaluation map, evaluation setup)**.** Let G be a finite set, M a set of attributes and $\mathbb{W}_m := (W_m, \leq_m)$ a complete lattice for every attribute $m \in M$. Further, let

$$W := \prod_{m \in M} W_m \text{ and } \mathbb{W} := \prod_{m \in M} \mathbb{W}_m.$$

Then, a map

$$\alpha : G \to W, g \mapsto \prod_{m \in M} \{\alpha_m g\}$$

such that

$$\alpha_m : G \to W_m, g \mapsto w_m$$

is called **evaluation map**. We call $\mathfrak{E} := (G, M, \mathbb{W}, \alpha)$ an **evaluation setup**.

Example 1. *In the wine data set in [3] we can interpret the wines as a set G, the describing attributes as the set M and $\mathbb{W}_m$ as the numerical range of attribute m with the natural order.*

In the above example the the evaluation map

$$\alpha : G \times M \to W$$

assigns to every wine bottle the values of all attributes $m \in M$. This map is a good starting point for an elementary pattern structure with

$$L := \prod_{m \in M} W_m \text{ and } \mathbb{L} := \prod_{m \in M} \mathbb{W}_m.$$

Thus, $\mathbb{E} := (2^L, \supseteq)$ is the dually ordered power set of vectors with values of the attributes, which describe the wine. On $\mathbb{E}$ we installed the following kernel operator

$$\gamma : 2^L \to 2^L, X \mapsto [inf_{\mathbb{L}} X, sup_{\mathbb{L}} X]_{\mathbb{L}},$$

This leads to the o-projection of the elementary pattern structure $\mathcal{P}_\varrho$ via γ, that is,

$$(G, \mathbb{D}, \delta) := opr(\mathcal{P}_\varrho, \gamma).$$

As a matter of fact,

$$\mathbb{D} = (D, \supseteq) \text{ with } D = \text{Int}\mathbb{L}$$

is the dual interval lattice of $\mathbb{L}$ and the map δ is given by

$$\delta : G \to D,\ g \mapsto \{\varrho g\}.$$

Often the dual power set lattice $\mathbb{E}$ is too large for applications. Therefore, we concentrate on relevant patterns in $\mathbb{D}$, that is, in the dual interval lattice of $\mathbb{E}$.

To identify important patterns in $\mathbb{E}$ for the red wine classification, we looked at the positive examples of the training set and combined the results of different clustering algorithms implemented in python. In particular, we used a *k-means* algorithm, *k-medoids* algorithm (with metrices Mahalanobis, Euclidean and correlation), a *Gausian Mixture Model* and a *Bayesian Gaussian Mixture Model* to cluster the good red wines. Furthermore we interprete the leaves of decision trees (with Gini Impurity and entropy as splitting measure) as cluster of wines to find important patterns in $\mathbb{E}$ for our case. The same cluster algorithm can lead to different output clusters; this is a result of the different metrics, which were used to measure the distance and the randomly choosen starting points of the algorithms. Hence we tried every algorithm 5 times. For every attempt we used different specifications. The number of clusters for the *k-medoids*, *k-means*, *Gausian Mixture Model* and *Gausian Mixture Model* algorithm is set randomly between 2 and 50. For the decision trees we set the number of examples in a leaf to at least 100. This leads to more than 700 clusters in $\mathbb{E}$.
Via the kernel operator on $\mathbb{E}$:

$$\gamma : 2^L \to 2^L, X \mapsto [inf_{\mathbb{L}} X, sup_{\mathbb{L}} X]_{\mathbb{L}}$$

we get patterns in $\mathbb{D}$ of the clusters in $\mathbb{E}$. Since γ is a closure operator on the power set $\mathbf{2}^L := (2^L, \subseteq)$ we can think of the patterns as closures of clusters.
On the next step we eliminated all clusters with less than 100 wines. Then we looked at the ratio of good examples (wines with a scoring of 5 or better) and all examples (good and bad) in the patterns and took the five patterns with the best ratio. These patterns are listed below. The range of the attributes is printed in red. For a better interpretability we scaled every attribute to the range $[0, 1]$.

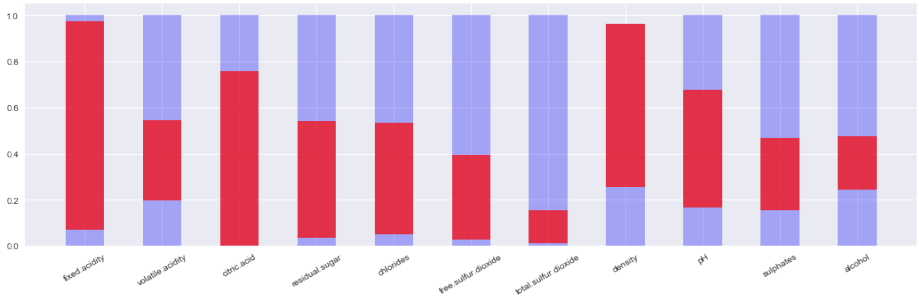


Fig. 1. Interval 1: decision tree (entropy), 108 wines, 108 good and 0 bad

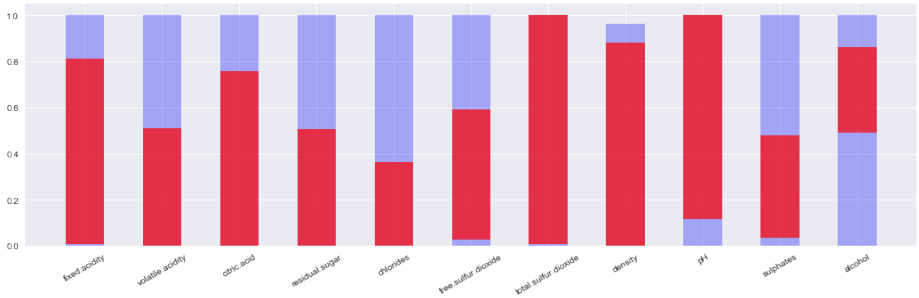


Fig. 2. Interval 2: decision tree (entropy), 158 wines, 158 good and 0 bad

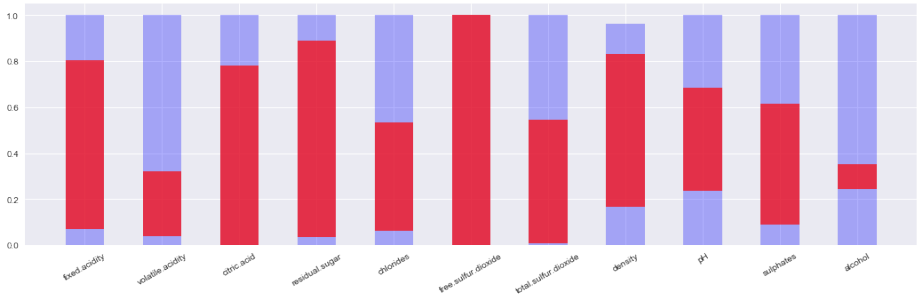


Fig. 3. Interval 3: decision tree (gini), 128 wines, 127 good and 1 bad

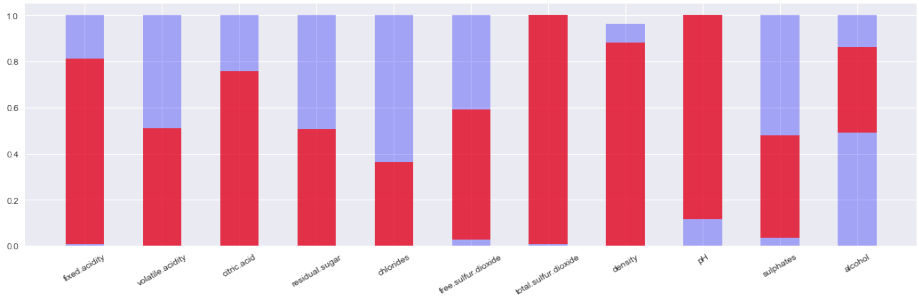


Fig. 4. Interval 4: k-medoids (mahalanobis), 163 wines, 160 good and 3 bad

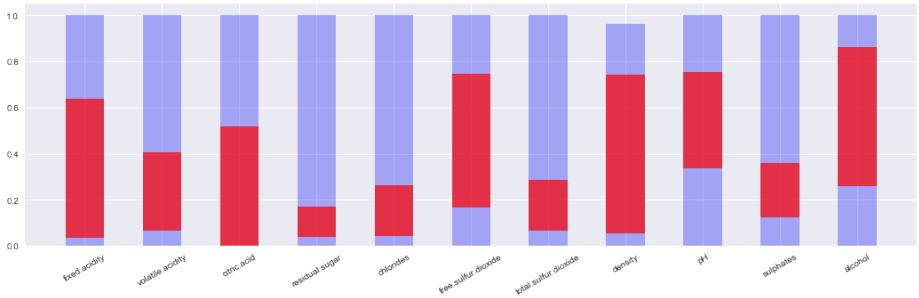


Fig. 5. Interval 5: k-medoids (mahalanobis), 216 wines, 209 good and 7 bad

Combining these 5 intervals to predict the classes of the test set leads to the following confusion matrix:

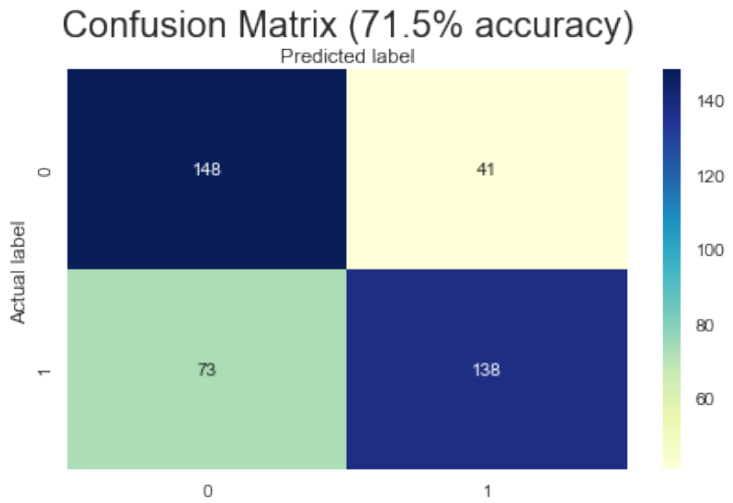


Fig. 6. Combination of the 5 intervals

The following table presents a comparison of our method to other algorithms.

Our method is easy to interprete and leads to the second best precision of all listed algorithms, but the recall value is the worst under all methods. More patterns would probably lead to a better recall, but likely worsen the precision. Further investigations are needed to find the best collections of patterns for different usecases (e.g. maximize accuracy). The here presented proceeding is just an example of building a model from our framework. Hopefully, further investigations show, that it is possible to create stronger models with our framework.

Model	Accuracy	Presicion	Recall	F1-score
cluster pattern	**71,5**%	**77,1**%	**65,5**%	**70,7**%
K-nearest neighbors	74,2%	73,0%	81,0%	76,9%
Support Vector	73,5%	75,6%	73,4%	74,5%
Naive Bayes	70,0%	69,2%	77,8%	73,2%
logistic regression	73,2%	73,6%	76,8%	75,2%
Random Forest	78,0%	80,0%	77,7%	78,8%

Table 1. Comparison of different classifier algorithms

5 Conclusion

We introduced a new general framework for the application of pattern structures. Then we gave an example how this general framework can be used to predict the quality of red wines. In the presented way the pattern structures can be a useful tool in analysing data. As shown here they are capable to give good predictions and the good interpretability makes them even more powerful.

References

1. T.S. Blyth, M.F.Janowitz (1972), Residuation Theory, Pergamon Press, pp. 1-382.
2. A. Buzmakov, S. O. Kuznetsov, A. Napoli (2015) , Revisiting Pattern Structure Projections. Formal Concept Analysis. Lecture Notes in Artificial Intelligence (Springer), Vol. 9113, pp 200-215.
3. P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis (2009), Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems 47(4), pp. 547-553.
4. B. Ganter, S. O. Kuznetsov (2001), Pattern Structures and Their Projections. Proc. 9th Int. Conf. on Conceptual Structures, ICCS'01, G. Stumme and H. Delugach (Eds.). Lecture Notes in Artificial Intelligence (Springer), Vol. 2120, pp. 129-142.
5. T. B. Kaiser, S. E. Schmidt (2011), Some remarks on the relation between annotated ordered sets and pattern structures. Pattern Recognition and Machine Intelligence. Lecture Notes in Computer Science (Springer), Vol. 6744, pp 43-48.
6. M. Kaytoue, S. O. Kuznetsov, A. Napoli, S. Duplessis (2011), Mining gene expression data with pattern structures in formal concept analysis. Information Sciences (Elsevier), Vol.181, pp. 1989-2001.
7. S. O. Kuznetsov (2009), Pattern structures for analyzing complex data. In H. Sakai et al. (Eds.). Proceedings of the 12th international conference on rough sets, fuzzy sets, data mining and granular computing (RSFDGrC'09). Lecture Notes in Artificial Intelligence (Springer), Vol. 5908, pp. 33-44.
8. S. O. Kuznetsov (2013), Scalable Knowledge Discovery in Complex Data with Pattern Structures. In: P. Maji, A. Ghosh, M.N. Murty, K. Ghosh, S.K. Pal, (Eds.). Proc. 5th International Conference Pattern Recognition and Machine Intelligence (PReMI'2013). Lecture Notes in Computer Science (Springer), Vol. 8251, pp. 30-41.
9. L. Lumpe and S. E. Schmidt (2015), A Note on Pattern Structures and Their Projections. Formal Concept Analysis. Lecture Notes in Artificial Intelligence (Springer), Vol. 9113, pp 145-150.

10. L. Lumpe, S. E. Schmidt (2016), Morphisms Between Pattern Structures and Their Impact on Concept Lattices, FCA4AI@ ECAI 2016, pp 25-34.
11. L. Lumpe, S. E. Schmidt (2015), Pattern Structures and Their Morphisms. CLA 2015, pp 171-179.
12. L. Lumpe, S. E. Schmidt (2016), Viewing Morphisms Between Pattern Structures via Their Concept Lattices and via Their Representations, International Symposium on Methodologies for Intelligent Systems 2017, pp. 597-608.
13. H. S. Park, C. H. Jun (2009), A simple and fast algorithm for K-medoids clustering. Expert systems with applications 36(2), pp. 3336-3341.

Interval-based sequence mining using FCA and the NextPriorityConcept algorithm

Salah Eddine Boukhetta, Jérémy Richard, Christophe Demko, and Karell Bertet

Laboratory L3i, La Rochelle University, La Rochelle, France

Abstract. In this paper, we are interested in sequential data analysis using `GALACTIC`, a new library based on Formal Concept Analysis (FCA) for calculating a concept lattice from heterogeneous and complex data. Inspired by the pattern structure theory, data in `GALACTIC` are described by predicates according to their types and a system of plugins allows an easy integration of new characteristics and new descriptions. We present new ways to analyse interval-based sequences, where items persist in time. Here we address the question of mining relevant sequential patterns, describing a set of sequences, by maximal common subsequences, or shortest supersequences. Experimentation on two real sequential datasets shows the effectiveness of our plugins in term of size of the lattice and of running time.

Keywords: Formal concept analysis · Lattice · Pattern structures · Interval-based sequences · Maximal common subsequences · Shortest common supersequences

1 Introduction

Sequences appear in many areas: sequences of words in a text, trajectories, surfing on the internet, or buying products in a supermarket. A sequence is a succession $\langle x_i \rangle$ of symbols, sets or events. Sequence mining is a topic of data mining which aims at finding frequent patterns in a dataset of sequences. Many algorithms have been proposed for mining sequential patterns, such as GSP [25], PrefixSpan [24], CloSpan [29], etc. These algorithms take as input a dataset of sequences and a minimum support threshold, and generate all frequent subsequences. Some algorithms mine time-point sequences $\langle (t_i, x_i) \rangle$, where an item x_i occurred at a timestamp t_i, for example for discovering episodes in a long time-point sequence [23, 26]. In real world applications, events may persist in time, or in an interval of time $(\underline{t}_i, \overline{t}_i)$, we call these sequences, *interval-based sequences* $\langle (\underline{t}_i, \overline{t}_i, X_i) \rangle$, where X_i is an itemset. They are mostly analysed using Allen's interval relations [1]. To quote from Kam and Fu's work on discovering temporal interval sequences [18], the patterns discovered are of type "event A's occurrence time overlaps with that of event B and both of these events occur before event C appears". Other works also used Allen's relations to discover interval based patterns [17, 28].

Formal Concept Analysis (FCA) appears in 1982 [27], then in the Ganter and Wille's 1999 work [14], it is issued from a branch of applied lattice theory that first appeared in the book of Barbut and Monjardet in 1970 [2]. The lattice property guarantees both a hierarchy of clusters, and a complete and consistent navigation structure for interactive approaches [11]. The formalism of pattern structures [13, 20] and abstract conceptual navigation [10, 9] extend FCA to deal with non-binary data, where data is described by patterns such that the pattern space must be organised as a semi-lattice in order to maintain a Galois connection between objects and their descriptions. By FCA framework, pattern lattice and bases of rules are defined, where a concept is composed of a subset of objects together with their common patterns, and a rule possesses patterns in premises and conclusions. However, pattern lattices are huge, often untractable [19], and the need for approaches to drive the search towards the most relevant patterns is a current challenge. Logical Concept Analysis [12] is a generalization of FCA in which sets of attributes are replaced by logical expressions. The power set of attributes mentioned by the Galois connection is replaced by an arbitrary set of formulas to which are associated a deduction relation (i.e., subsumption), and conjunctive and disjunctive operations, and therefore forms a lattice. Inspired by pattern structures, the NextPriorityConcept algorithm, introduced in a recent article [8] proposes a user-driven pattern mining approach for heterogeneous and complex data as input. This algorithm allows a generic pattern computation through specific *descriptions* of objects by predicates. It also proposes to reduce predecessors of a concept by the refinement of a set of objects into a fewer one through specific user exploration *strategies*, resulting in a reduction of the number of generated patterns. Some algorithms appear within FCA framework for analysing sequence data; we can mention works for mining medical care trajectories using pattern structures [5, 6], sequence mining to discover rare patterns [7], and other studies on demographic sequences [15, 16]. But for discovering interval-based sequence using FCA methods, we found fewer works. We can cite Kaytoue et al.'s work on gene expression data [21].

In this article, we propose a new sequence mining approaches using the NextPriorityConcept algorithm, with descriptions and strategies dedicated to interval-based sequences. We propose two different descriptions that describe a subset of objects by subsequences or supersequences. We also propose five strategies of pattern exploration in order to generate a reduction of a cluster of interval-based sequences, i.e., its predecessors in the pattern lattice.

Section 2 introduces basic definitions related to interval sequence mining and a short description of the NextPriorityConcept algorithm. Section 3 will be dedicated to our new interval-based sequence descriptions and strategies. Experimental results are presented in Section 4.

2 Preliminaries

2.1 Interval-based Sequences

A sequence s is a succession of itemsets from a dictionary Σ, often in the form of $s =< X_i >_{i \leq n}$, where $X_i \subseteq \Sigma$ is a subset of items i.e., itemset. A temporal

sequence is a sequence where each itemset X_i must have an associated timestamp t_i. An *Event* (or *Time frame*) E, is a triple $E = (\underline{t}, \overline{t}, X)$ where $X \subseteq \Sigma$ is an *itemset*, $\underline{t}$ is the starting time and $\overline{t}$ is the ending time, $\underline{t} \leq \overline{t}$. For better readability we refer to $(\underline{t}, \overline{t})$ by T.

Interval-based sequence. An *interval-based sequence* (or *Time frame sequence*) $s = \langle (T_i, X_i) \rangle_{i \leq n}$ is a list of events (or time frames), verifying $\overline{t_i} < \underline{t_{i+1}}$, thus an interval-based sequence is a list of separate intervals containing itemsets. The size of the interval-based sequence is the number of its time frames. We refer to the interval-based sequence by *sequence*.

Consider the example in Figure 1 for an alphabet $\Sigma = \{C, M, P, H\}$ (where C stands for Castle, M for Museum, P for Public Garden and H for Historical Place), the sequences represent trajectories of visits of three tourists $s1, s2$ and $s3$. In this example, visitors may be in two or more different locations at the same interval as the intervals are large enough and we may don't have the exact interval of each location.

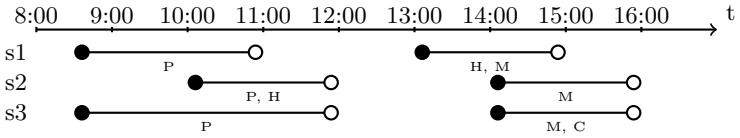


Fig. 1. Example of interval-based sequences

Subinterval. For two intervals, $T = (\underline{t}, \overline{t})$ and $T' = (\underline{t'}, \overline{t'})$, we say that T is subinterval T', if : $\underline{t} \geq \underline{t'}$ and $\overline{t} \leq \overline{t'}$ and we write $T \preceq T'$, that corresponds to the containing relation from Allen's relations [1].

Projections. We introduce the *projection* operator Φ of a sequence s, over a given interval T, that selects all the itemsets of the sequence included in this interval : $\Phi_T(s) = \{X' \ : \ T' \preceq T \text{ and } (T', X') \in s\}$. Dually, the *projection* operator Φ, over an itemset $X \subseteq \Sigma$ selects all the intervals where the items of X may occur: $\Phi_X(s) = \{T' \ : \ X' \subseteq X \text{ and } (T', X') \in s\}$. $\Phi_\Sigma(s)$ represents a set of all the intervals in s.

Subsequence. A sequence s, is *subsequence* of another sequence s', $s \Subset s'$ if for all $(T, X) \in s$, there exists $(T', X') \in s'$ such that $T \preceq T'$ and $X \subseteq X'$. We also say that s' is **supersequence** of s.

Affix. A prefix/suffix of a sequence $s = \langle (T_i, X_i) \rangle_{i \leq n}$ according to a window w, is the subsequence of s composed by the first/last w time frames of s, $\text{prefix}(s, w) = \langle (T_i, X_i) \rangle_{1 \leq i \leq w}$, $\text{suffix}(s, w) = \langle (T_i, X_i) \rangle_{(n-w) < i \leq n}$.

Cardinality. For a set of sequences A, an item $x \in \Sigma$ and an interval T, the function *card* gives the number of sequences $a \in A$ possessing the item x in the projection of a over T, $x \in \Phi_T(a)$.

$$card(A, T, x) = |\{a \ : \ x \in \Phi_T(a), a \in A\}| \tag{1}$$

When $card(A, T, x)$ is maximal, we denote $card(A, T, x)$ by $card_{max}(A, T)$. We define $card_{min}(A, T)$ in the same maner when $card(A, T, x)$ is minimal.

From example in Figure 1 we have, $\Phi_{(10:00,11:00)}(s2) = \{P, H\}$, the prefix of $s1$ is $\langle(08{:}30, 11{:}00), P)\rangle$, and all three tourists were in the museum from 14:00 to 15:00, so $\langle(14{:}00, 15{:}00), M)\rangle$ is subsequence of $s1, s2$ and $s3$. For $A = \{s1, s2, s3\}$ and $T = (11{:}00, 12{:}00)$ $card(A, T, P) = card_{\max}(A, T) = 2$.

2.2 Description of the NextPriorityConcept algorithm

The NextPriorityConcept algorithm [8] computes concepts for heterogeneous and complex data for a set of objects G, its main characteristics are:

Heterogeneous data as input, described by specific predicates. The algorithm introduces the notion of *description* δ as an application to provide predicates describing a set of objects $A \subseteq G$. Each concept $(A, \delta(A))$ is composed of a subset of objects A and a set of predicates $\delta(A)$ describing them. Such generic use of predicates makes it possible to consider heterogeneous data as input, i.e., numerical, discrete or more complex data. However, unlike classical pattern structures, predicates are not globally computed in a preprocessing step, but locally for each concept.

Concept lattice generation. The NextPriorityConcept algorithm is inspired by Bordat's algorithm[3], also found in Linding's work [22], that recursively computes the immediate successors of a concept, starting with the bottom concept. It is a dual version that computes the immediate predecessors of a concept, starting with the top concept $(G, \delta(G))$ containing the whole set of objects, until no more concepts can be generated. The use of a priority queue ensures that each concept is generated before its predecessors, and a mechanism of propagation of constraints ensures that meets will be computed. NextPriorityConcept computes a concept lattice and therefore is positioned in FCA framework, with the possibility of extraction of rules, closure computations or navigation in the lattice, that can be useful in many fields of pattern mining and discovery.

Predecessors selection by specific strategies. The algorithm also introduces the notion of *strategy* σ to provide predicates (called *selectors*) describing candidates for an object reduction of a concept $(A, \delta(A))$ i.e., predecessors of $(A, \delta(A))$ in the pattern lattice. A selector proposes a way to refine the description $\delta(A)$ to a reduced set $A' \subset A$ of objects. Several strategies are possible to generate predecessors of a concept, going from the naive strategy classically used in FCA that considers all the possible predecessors, to strategies reducing the number of predecessors in order to obtain smaller lattices. Selectors are only used for the predecessors' generation, they are not kept either in the description or in the final set of predicates. Therefore, choosing or testing several strategies at each iteration in a user-driven pattern discovery approach would be interesting.

The main result in [8] states that the NextPriorityConcept algorithm computes the formal context $\langle G, P, I_P\rangle$ and its concept lattice (where P is the set of predicates describing the objects in G, and $I_P = \{(a, p),\ a \in G, p \in P\ :\ p(a)\}$ is the relation between objects and predicates) if description δ verifies $\delta(A) \sqsubseteq \delta(A')$

for $A' \subseteq A$. The run-time of the NextPriorityConcept algorithm has a complexity $O(|\mathcal{B}|\,|G|\,|P|^2\,(c_\sigma + c_\delta))$ (where $\mathcal{B}$ is the number of concepts, c_σ is the cost of the strategy and c_δ is the cost of the description), and a space memory in $O(w\,|P|^2)$ (where w is the width of the concept lattice).

3 NextPriorityConcept for sequences

In order to mine interval-based sequences with NextPriorityConcept algorithm, we have to define descriptions and strategies for sequences. Consider a set G of sequences whose size is smaller than n, defined on an alphabet Σ as input:

A description δ is a mapping $\delta : 2^G \to 2^P$ which defines a set of predicates $\delta(A)$ describing any subset $A \subseteq G$ of sequences. Predicates are of form, *"is subsequence/supersequence of"*.

A strategy σ is a mapping $\sigma : 2^G \to 2^P$ which defines a set of selectors $\sigma(A)$ to select strict subset A' of A as predecessor candidates of any concept $(A, \delta(A))$ in the pattern lattice.

Predicates are computed using the subsequence relation in the form *"is subsequence of "* or *"is supersequence of"*. For better readability, the sets $\delta(A)$ and $\sigma(A)$ will be treated either as sets of predicates/selectors, or as sets of sequences, they can reciprocally be deduced from each other.

3.1 Description for interval sequences

We define two descriptions for a subset $A \subseteq G$ of sequences. The maximal common time frame description `MCTF` refers to the classical maximal common subsequence description [4] and corresponds to the set of maximal subsequences of all sequences in A. The shortest supersequence time frame description `SSTF` contains all minimal supersequences of sequences in A.

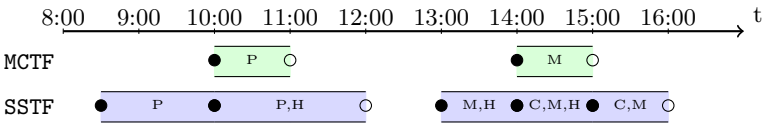


Fig. 2. $\delta_{\text{MCTF}}(\{s1, s2, s3\})$ and $\delta_{\text{SSTF}}(\{s1, s2, s3\})$ for $s1, s2$ and $s3$ in Figure 1.

Figure 2 represents $\delta_{\text{MCTF}}(A)$ and $\delta_{\text{SSTF}}(A)$ for $A = \{s1, s2, s3\}$ from Figure 1. We can observe that `MCTF` could be interpreted as "conjunction" where $(14{:}00, 15{:}00, \{M\})$ in δ_{MCTF} means that all $s1, s2$ and $s3$ contain $(14{:}00, 15{:}00, \{M\})$. Dually, `SSTF` could be interpreted as "disjunction". More formally, `MCTF` and `SSTF` are defined for a subset $A \subseteq G$ of sequences by:

Maximal Common Time Frame (MCTF) description.

$$\delta_{\text{MCTF}}(A) = \{\langle (T, X) \rangle \ : \forall a \in A, X \subseteq \Phi_T(a)\} \tag{2}$$

Shortest Shared Time Frame (SSTF) description.

$$\delta_{\text{SSTF}}(A) = \{\langle (T, X) \rangle \ : \forall a \in A, \Phi_T(a) \subseteq X\} \tag{3}$$

To compute the two descriptions of a set A of sequences, we iterate on the sequences of A, and update the resulting sequences of $\delta(A)$ with the common parts. Therefore the complexity of the description is $c_\delta^{MCTF} = c_\delta^{SSTF} = O(s\,|A|\,log(|A|)) \leq O(s\,|G|\,log(|G|))$ where s is the maximal size of the computed sequences. We have to ensure that the NextPriorityConcept algorithm generates a concept lattice. These descriptions must verify $\delta(A) \sqsubseteq \delta(A')$ for $A' \subseteq A$:

Proposition 1 *For $A' \subseteq A \subseteq G$, we have the following two properties:*

1. $\delta_{MCTF}(A) \sqsubseteq \delta_{MCTF}(A')$
2. $\delta_{SSTF}(A) \sqsubseteq \delta_{SSTF}(A')$

Proof: Let A and A' be two subsets of G such that $A' \subseteq A$.

1. Let $c \in \delta_{\text{MCTF}}(A)$, i.e., c is a maximal subsequence of A. From $A' \subseteq A$ we can deduce that c is also a subsequence of sequences in A', but c is not necessarily a maximal subsequence for A'. If c is a maximal subsequence in A' then $c \in \delta_{\text{MCTF}}(A')$. Otherwise, there exists $c' \in \delta_{\text{MCTF}}(A')$ such that c is a subsequence of c'. In these two cases, we can deduce that, $\delta_{\text{MCTF}}(A) \sqsubseteq \delta_{\text{MCTF}}(A')$.
2. Let $s \in \delta_{\text{SSTF}}(A)$, i.e., s is the supersequence of all sequences in A. From $A' \subseteq A$ we can deduce that s is also a supersequence of sequences in A' but not necessarily the shortest one. If s is a shortest supersequence in A' then $s \in \delta_{\text{SSTF}}(A')$. Otherwise, there exists $s' \in \delta_{\text{SSTF}}(A')$ such that s is supersequence of s'. We can deduce that, $\delta_{\text{SSTF}}(A) \sqsubseteq \delta_{\text{SSTF}}(A')$.

□

3.2 Strategies and selectors for time frame sequences

Strategies are used by the NextPriorityConcept algorithm to refine each concept $(A, \delta(A))$ into concepts with fewer objects (sequences) and more specific descriptions. It is important to clarify that a strategy must be used with a description composed of predicates of the same kind. Recall that `MCTF` description defines "is subsequence of" predicates whereas `SSTF` description defines "is supersequence of" predicates. We define one subseqeunce strategy for the `MCTF` description and four supersequence strategies for the `SSTF` description.

Strategy with subsequence selectors for `MCTF` description: The *Augmented Minimum Cardinality* strategy computes all the possible refinements of a concept $(A, \delta_{\text{MCTF}}(A))$ by adding in the events of sequences of δ_{MCTF} any item with a minimal cardinality $card(A, T, x)$ for each time frame T. More formally, σ_{AMC} is defined for $A \subseteq G$ by:

Augmented Minimum Cardinality.

$$\begin{aligned}\sigma_{\text{AMC}}(A) = \{\langle (T, X)\rangle :\ &\forall a \in A, \Phi_T(a) \subseteq X \text{ and } \forall x \in X \\ &card(A, T, x) = |A| \ \vee\ card(A, T, x) = card_{min}(A, T)\}\end{aligned} \tag{4}$$

The cost for this strategy is clearly equal to the cost of `MCTF` description, $c_\sigma^{AMC} = c_\delta^{MCTF}$.

Figure 3, represents the generated Hasse diagram of sequences in Figure 1 using the `MCTF` description and the `AMC` strategy, where in each concept the

symbol $ represents the identifier of the concept, and the symbol # represents the number of sequences inside the concept, i.e., its support. The concept $0 contains the description of the 3 visits. The concept $1 describes the two visits, $s1$ and $s3$ as they were in the Public Garden from 08:30 to 11:00 then in the Museum from 14:00 to 15:00.

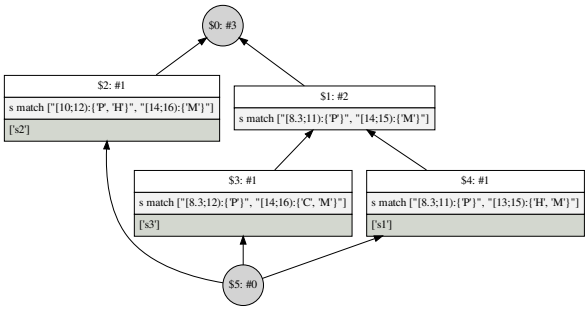


Fig. 3. Hasse diagram of the reduced concept lattice for the AMC strategy and the MCTF description

Strategies with supersequence selectors for `SSTF` description: First, the *Simple Time Frame* strategy consists in simply generating selectors by deleting one item from each itemset on the `SSTF` description. More formally, σ_{STF} is defined for a subset of sequences $A \subseteq G$, and the `SSTF` description, by:

Simple Time Frame strategy (STF).

$$\sigma_{\text{STF}}(A, \delta_{\text{SSTF}}) = \{\langle (T, X \backslash \{x\}) \rangle \ : \forall x \in X,\ (T, X) \in s, s \in \delta_{\text{SSTF}}(A)\} \quad (5)$$

To implement this strategy, we have to consider any item of any sequence of the description, thus a complexity $c_\sigma^{STF} = O(s\, m\, c_\delta^{SSTF}) \leq O(s\, |\Sigma|\, c_\delta^{SSTF})$ where m is the maximal number of items in any time frame in the sequences of the description.

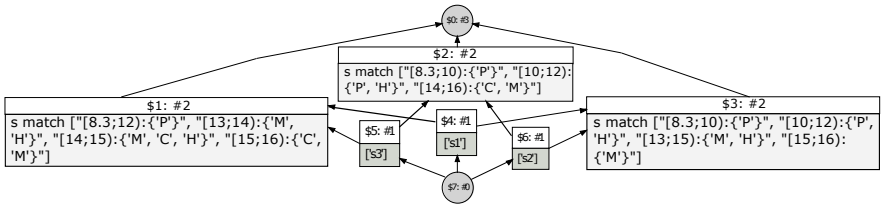


Fig. 4. Hasse diagram of the reduced concept lattice for the STF strategy and the SSTF description

Figure 4 represents the generated Hasse diagram using the `SSTF` description and the `STF` strategy. Concepts $1, $2, and $3 contain descriptions of $\{s1, s3\}$, $\{s2, s3\}$, and $\{s1, s2\}$. Concept $2 shows that at least one of $s2$ or $s3$ visited P from 08:30 to 10:00, then H or P from 10:00 to 12:00, and finally C or M from 14:00 to 16:00. The strategy constructs a lattice with all concepts, hence it is time consuming. So, we thought about strategies that may reduce the size of the

lattice, and the time complexity. The *Bounds Time Frame* strategy consists in deleting only the items of cardinalities minimal or maximal. The *Window Affix Time Frame* strategy uses a *window* parameter w and generates the prefix and suffix. The *Alphabet Time Frame* strategy, deletes one item of the alphabet from all the time frames. More Formally, these strategies are defined for a subset A of sequences and the description δ_{SSTF} by:

Bounds Time Frame (BTF), for an integer *card*:

$$\sigma_{\text{BTF}}(A, c) = \{\langle (T, X \backslash \{x\}) \rangle \ : \ (T, X) \in s, \\ card(A, T, x) = c, s \in \delta_{\text{SSTF}}(A)\} \tag{6}$$

In particular, we can consider $\sigma_{\text{BTF}}(A, card_{max})$ and $\sigma_{\text{BTF}}(A, card_{min})$

Window Affix Time Frame strategy (WATF), for a window size w:

$$\sigma_{\text{WATF}}(A, w) = \{\langle (T, X) \rangle \ : \ (T, X) \in s - \text{prefix}(s, w), \\ (T, X) \in s - \text{suffix}(s, w), s \in \delta_{\text{SSTF}}(A)\} \tag{7}$$

Here the $s - \text{prefix}(s, w)$ means s without time frames in $\text{prefix}(s, w)$, same for $s - \text{prefix}(s, w)$.

Alphabet Time Frame strategy (ATF):

$$\sigma_{\text{ATF}}(A) = \{\langle (T, X \backslash \{x\}) \rangle \ : \forall (T, X) \in s, \ \forall x \in \Sigma, \ s \in \delta_{\text{SSTF}}(A)\} \tag{8}$$

The cost of the `BTF` strategy is $c_{\sigma}^{BTF} = O(s\, c_{\delta}^{SSTF})$, as it must calculate the predicates of `SSTF`, then iterate on the resulting sequences. For the `WATF` strategy, $c_{\sigma}^{WATF} = c_{\delta}^{SSTF}$, the w parameter is constant, so the cost is equal only to the cost of δ_{SSTF}. For the `ATF` strategy $c_{\sigma}^{ATF} = O(s\, c_{\delta}^{SSTF})$.

The NextPriorityConcept allows a user-driven approach for the data analyst to choose strategies that respond the best to the specifications of the data. With the `SSTF` description, the data analyst have a choice of 4 strategies. The `BTF` strategy allows a generation of predecessors where frequent or non-frequent events may not appear (maximum and minimum cardinalities). The `WATF` strategy focuses of events that appear first or last at the same interval. The `ATF` strategy focuses on clusters where some events may not appear. The use of these strategies reduces the time complexity of the lattice generation process and generate a smaller lattice than the `STF` strategy.

4 Experiments

In this section, we experimentally evaluate our descriptions and strategies for mining interval sequences. Our approach is different from previous works. We are not mining all frequent interval-based sequences, but we use our strategies and descriptions to mine only the relevant ones. To experimentally assess the effectiveness of our descriptions and strategies, we use `GALACTIC`[1] (**GA**lois **LA**ttices, **C**oncept **T**heory, **I**mplicational systems and **C**losures), a development platform

[1] https://galactic.univ-lr.fr

of the NextPriorityConcept algorithm, which mixed with a system of plugins, makes it possible easy integration of new kinds of data (descriptions and strategies). We have implemented new plugins for sequences. Experiments were performed on an Intel Core i7 2.20GHz machine with 32GB main memory. We run our experiments on two real datasets:

GeoLuciole **dataset** is issued from classical GPS trajectories of people's deplacements in the city of La Rochelle in France. By matching the GPS coordinates to districts of the city, raw data are transformed into semantic sequences. The data have been collected by a specific application named GeoLuciole that we have developed for the DA3T[2] project. The data contains only 15 trajectories with an average size of sequences equals to 2[3].

Wine-City **dataset** is issued from the museum "La cité du vin" in Bordeaux, France[4], gathered from the visits over a period of one year (May 2016 to May 2017). The museum is a large "open-space", where visitors are free to explore the museum the way they want. The trajectories in this dataset are of size 9 on average.

Comparison of descriptions

Here we compare our two descriptions in terms of running time and lattice size. We use the `MCTF` description with the `AMC` strategy, and the `SSTF` description with the `STF` strategy. These two strategies generate all possibles subseqeunces/supersequences. Results for the Wine-City dataset are given in Table 1. We can observe that `MCTF` is far faster than `SSTF`. It generates a lattice of 149 concepts in about 2 minutes from 500 sequences, while the `SSTF` description, stops with 1024 concepts generated in about 30 minutes from only 10 sequences. These descriptions are two different ways of representing the data. The `SSTF` description is clearly richer than the `MCTF` description that is the classical maximal common subsequences description extended to intervals, but `SSTF` is not adapted to huge datasets.

	data size	4	6	10	100	500	1000
$\sigma_{\text{STF}}(\delta_{\text{SSTF}})$	# concepts	16	33	**1024**			
	time(ms)	2878	11760	**1927039**			
	time/concept(ms)	179,87	356,36	**1881,87**			
$\sigma_{\text{AMC}}(\delta_{\text{MCTF}})$	# concepts	6	12	13	67	**149**	**132**
	time(ms)	267	771	838	17198	**141281**	**246437**
	time/concept(ms)	44.5	64.25	64.46	256.68	948.19	1866.94

Table 1. # of concepts and execution time for δ_{MCTF} and δ_{SSTF} descriptions for the Wine city dataset

[2] System for the Analysis of Numerical Traces for the development of Tourist Territories (Dispositif d'Analyse des Traces numériques pour la valorisation des Territoires Touristiques)

[3] It was planned to collect more data during the holidays on Mars and April, but unfortunately, this was impossible due to the world pandemic Covid-19

[4] https://www.laciteduvin.com/en

Comparison of strategies

We focus now on the SSTF description to compare the four strategies; STF, BTF, WATF and ATF. Recall that the STF strategy generates all possible supersequences whereas BTF, WATF and ATF focus on special supersequences (prefix, suffix, according to a window). Figure 5 shows the running time and the number of concepts generated using the WINE-CITY dataset. Compared to the STF strategy, we can clearly see that the other strategies are faster, and generate fewer concepts. The WATF strategy is the best in this example, especially with $w = 1$, with $w = 2$: the result approximates that of the BTF strategy. We run the BTF strategy with $card_{min}$ and $card_{max}$: we observe that the running time is better, and we obtain fewer concepts with $card_{min}$. The complexity of NEXTPRIORITYCONCEPT depends on the size of the lattice, therefore a reduction to more relevant concepts also reduce the running time. Table 2 presents a comparaison between the

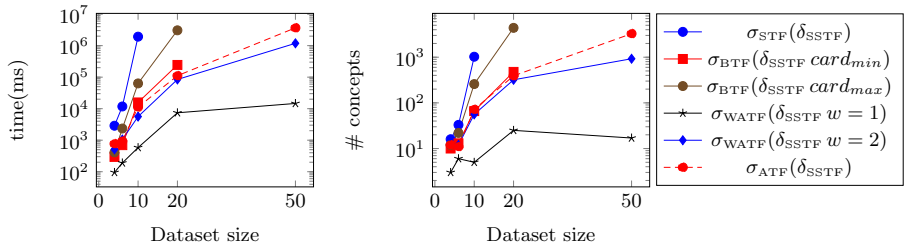


Fig. 5. Running time and size of the lattice using the δ_{SSTF} description and the four strategies for the WINE-CITY dataset

four strategies of the SSTF description with the GEOLUCIOLE dataset. The BTF, ATF and WATF strategies are compared to the STF in terms of compression ratio i.e., the ratio between the number of concepts obtained with the STF strategy by the number of concepts obtained with each of the other strategies. Table 2 shows the effectiveness of our strategies in reducing the number of concepts. We can also observe that the compression ratio is improved as we increase the size of the data for all strategies. ATF strategy performs better and generates fewer concepts compared to BTF. The compression ratio is low with the WATF strategy, because the size of sequences is close to the window, and thus as we raise the windows the number of concepts get closer to the STF strategy. The behaviour of WATF strategy is linked to the average size of the sequences. The data analyst can variate parameters such as the cardinality for BTF, or the window for WATF, to generate only relevant concepts.

5 Conclusion

In this paper, we presented a sequence mining approach using the NEXTPRIORITYCONCEPT algorithm. This algorithm allows a generic pattern computation through specific *descriptions* and *strategies*.

We presented two descriptions and five strategies for analysing interval-based sequences. The two descriptions represent two different approaches for representing a set of sequences. The first one MCTF is the classical maximal common

dataset size	4	5	6	7	10	15
	# Concepts					
σ_{STF}	16	32	64	**128**	**672**	**16640**
	Compression ratio					
$\sigma_{\text{BTF}}(card_{min})$	2.28	**4.57**	4.92	4.74	18.66	96.18
$\sigma_{\text{BTF}}(card_{max})$	**5.33**	2.66	**7.11**	3.45	28	92.96
σ_{ATF}	4	2.13	4	8.53	24.88	130
$\sigma_{\text{WATF}}(w = 1)$	1.77	3.2	3.76	**10.66**	**32**	**386.97**
$\sigma_{\text{WATF}}(w = 2)$	1.45	2.13	1.42	4.26	7.38	49.52

Table 2. # of concepts and compression ratio using δ_{SSTF} description and the four strategies `STF`, `BTF`, `ATF` and `WATF` for the GeoLuciole dataset

subsequences, whereas the second one `SSTF` provides a richer description of interval sequences. We presented one strategy for the `MCTF` description, and four strategies for the `SSTF` description that can be tested in a user-driven approach in order to generate fewer concepts and more relevant data. Therefore, we will focus on reducing the time complexity of our plugins, and create more configurable ones that respond the best to the particularity of the data we want to treat.

References

1. Allen, J.F.: An interval-based representation of temporal knowledge. In: IJCAI. vol. 81, pp. 221–226 (1981)
2. Barbut, M., Monjardet, B.: Ordres et classifications : Algèbre et combinatoire. Hachette, Paris (1970), 2 tomes
3. Bordat, J.P.: Calcul pratique du treillis de Galois d'une correspondance. Mathématiques et Sciences humaines **96**, 31–47 (1986)
4. Boukhetta, D.C., Bertet K., R.J.: Sequence mining using FCA and NextPriorityConcept algorithm. In: The 15th International Conference on Concept Lattices and Their Applications (CLA 2020) (2020)
5. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: Fca and pattern structures for mining care trajectories (2013)
6. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: On projections of sequential pattern structures (with an application on care trajectories) (2013)
7. Codocedo, V., Bosc, G., Kaytoue, M., Boulicaut, J.F., Napoli, A.: A proposition for sequence mining using pattern structures. In: International Conference on Formal Concept Analysis. pp. 106–121. Springer (2017)
8. Demko, Ch., Bertet, K., Faucher, C., Viaud, J.F., Kuznetsov, S.O.: NextPriorityConcept: A new and generic algorithm computing concepts from complex and heterogeneous data. arXiv preprint arXiv:1912.11038 (2019)
9. Ferré, S.: Systèmes d'information logiques : un paradigme logico-contextuel pour interroger, naviguer et apprendre. Doctorat, University of Rennes 1, France (Oct 2002)
10. Ferré, S.: Reconciling Expressivity and Usability in Information Access - From Filesystems to the Semantic Web. Habilitation, University of Rennes 1, France (november 2014)

11. Ferré, S.: Reconciling expressivity and usability in information access from file systems to the semantic web (2014)
12. Ferré, S., Ridoux, O.: A logical generalization of formal concept analysis. vol. 1867, pp. 371–384 (03 2000)
13. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. In: LNCS of International Conference on Conceptual Structures (ICCS'01). pp. 129–142 (2001)
14. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematical foundations. Springer Verlag, Berlin (1999)
15. Gizdatullin, D., Baixeries, J., Ignatov, D.I., Mitrofanova, E., Muratova, A., Espy, T.H.: Learning interpretable prefix-based patterns from demographic sequences. In: International Conference on Intelligent Data Processing: Theory and Applications. pp. 74–91. Springer (2016)
16. Gizdatullin, D., Ignatov, D., Mitrofanova, E., Muratova, A.: Classification of demographic sequences based on pattern structures and emerging patterns. In: Supplementary Proceedings of 14th International Conference on Formal Concept Analysis, ICFCA. pp. 49–66 (2017)
17. Guyet, T., Quiniou, R.: Extracting temporal patterns from interval-based sequences. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
18. Kam, P.s., Fu, A.W.C.: Discovering temporal patterns for interval-based events. In: International Conference on Data Warehousing and Knowledge discovery. pp. 317–326. Springer (2000)
19. Kaytoue, M.: Contributions to Pattern Discovery. Habilitation, University of Lyon, France (february 2020)
20. Kaytoue, M., Codocedo, V., Buzmakov, A., Baixeries, J., Kuznetsov, S.O., Napoli, A.: Pattern structures and concept lattices for data mining and knowledge processing. In: In Proceedings of ECML-PKDDl (2015)
21. Kaytoue, M., Duplessis, S., Kuznetsov, S.O., Napoli, A.: Two fca-based methods for mining gene expression data. In: International Conference on Formal Concept Analysis. pp. 251–266. Springer (2009)
22. Linding, C.: Fast concept analysis. In: Working with Conceptual Structures-Contributions to ICC. pp. 235–248 (2002)
23. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. Data mining and knowledge discovery **1**(3), 259–289 (1997)
24. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H., Chen, Q., Dayal, U., Hsu, M.C.: Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: icccn. p. 0215. IEEE (2001)
25. Srikant, R., Agrawal, R.: Mining sequential patterns: Generalizations and performance improvements, edbt (1996)
26. Sun, X., Orlowska, M.E., Zhou, X.: Finding event-oriented patterns in long temporal sequences. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 15–26. Springer (2003)
27. Wille, R.: Restructuring lattice theory : an approach based on hierarchies of concepts. Ordered sets pp. 445–470 (1982), i. Rival (ed.), Dordrecht-Boston, Reidel.
28. Winarko, E., Roddick, J.F.: Armada–an algorithm for discovering richer relative temporal association rules from interval-based data. Data & Knowledge Engineering **63**(1), 76–90 (2007)
29. Yan, X., Han, J., Afshar, R.: Clospan: Mining: Closed sequential patterns in large datasets. In: Proceedings of the 2003 SIAM international conference on data mining. pp. 166–177. SIAM (2003)

Continuous Attributes for FCA-based Machine Learning*

Dmitry V. Vinogradov[1,2][0000−0001−5761−4706]

[1] FRC Computer Science and Control RAS, Moscow 119333, Russia
vinogradov.d.w@gmail.com
[2] Russian State University for Humanities, Moscow 125993, Russia
http://isdwiki.rsuh.ru

Abstract. In this paper we extend previously developed approach to FCA-based machine learning with discrete attributes to the case with objects described by continuous attributes. We combine the logistic regression with an entropy-based separation of attribute values, which is similar to Quinlan's approach to dealing with continuous attributes. We apply Cox-Snell and McFadden significance criteria to logistic regression. Finally, we present the results of applying the new version of FCA-based learning system to the analysis of Wine Quality dataset from UCI Machine Learning Repository.

Keywords: FCA · Machine Learning · continuous attributes · entropy.

Introduction

In [8] the author extended some FCA ideas [1] by developing a probabilistic approach to Machine Learning (ML) based on similarity operation. FCA provides a very efficient representation for training objects by means of bitsets (fixed length strings of bits) with bit-wise multiplication as a similarity between them. The previous version of the ML program (called 'VKF system' in honor to Prof. V.K. Finn) was applicable to objects described by discrete attributes only. However, a variety of interesting data needs both discrete and continuous attributes for their representation. The first step to include continuous cases to VKF system is an analogue of J.R. Quinlan's approach to similar problem for C4.5 decision tree algorithm [5]. He splits the whole domain of a continuous attribute into several intervals to reach minimal mean entropy.

To obtain bitset representation from such division we introduce indicator variables and combine their values. The main result asserts that bit-wise multiplication corresponds to a convex hull of intervals of values under similarity, which was studied in [2] and [4] in terms of interval pattern structures within, an FCA-based approach to analysis of data with continuous attributes. A more important problem is to discover complex combinations of continuous attributes.

* partially supported by RFBR grant 18-29-03063mk.

Since our main goal is to discover a classifier, we apply Bayes Machine Learning ideas to generate such complex attributes through well-known logistic regression.

The key question is to detect significance of essential relationships (interactions) between pairs of attributes. Hence, we apply well-known Cox-Snell and McFadden criteria to discover such interactions.

The structure of the paper is as follows. In Section 1 we recall general definitions and some facts from FCA. Section 2 covers main algorithms of VKF-method. Section 3 describes new results. Subsection 3.1 reproduces Quinlan's technique to separate continuous feature domain into several intervals. It also introduces a representation of the occurrence of an attribute value in some interval by bitset. Subsection 3.2 introduces a logistic regression approach to discovering relationships between continuous features.

1 Formal Concept Analysis (FCA)

A **(finite) context** is a triple (G, M, I) where G and M are finite sets and $I \subseteq G \times M$. The elements of G and M are called **objects** and **attributes**, respectively. As usual, we write gIm instead of $\langle g, m\rangle \in I$ to denote that object g has attribute m.

For $A \subseteq G$ and $B \subseteq M$, define

$$A' = \{m \in M | \forall g \in A(gIm)\}, \tag{1}$$

$$B' = \{g \in G | \forall m \in B(gIm)\}; \tag{2}$$

so A' is the set of attributes common to all the objects in A and B' is the set of objects possessing all the attributes in B. The maps $(\cdot)' : A \mapsto A'$ and $(\cdot)' : B \mapsto B'$ are called **derivation operators** (**polars**) of the context (G, M, I).

If we fix attribute subsets $\{g_1\}' \subset M$ and $\{g_2\}' \subset M$ for objects $g_1 \in G$ and $g_2 \in G$, respectively, with corresponding bitsets, then the derivation operator on a pair of objects corresponds to bit-wise multiplication, since $\{g_1, g_2\}' = \{g_1\}' \cap \{g_2\}'$. More generally, the polars correspond to the iteration of bit-wise multiplication (in arbitrary order) of corresponding bitset-represented objects and attributes, respectively. The last remark is important, since bit-wise multiplication is a basic operation of modern CPU and GPGPU. The aim of the article is to invent a bitset representation of continuous features in such a way that bit-wise multiplication of resulting bitsets has clear meaning with respect to original values!

A **concept** of the context (G, M, I) is defined to be a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The first component A of the concept (A, B) is called the **extent** of the concept, and the second component B is called its **intent**. The set of all concepts of the context (G, M, I) is denoted by $L(G, M, I)$.

Definition 1. *For $(A, B) \in L(G, M, I)$, $g \in G$, and $m \in M$ define*

$$CbO((A, B), g) = ((A \cup \{g\})'', B \cap \{g\}'), \tag{3}$$

$$CbO((A, B), m) = (A \cap \{m\}', (B \cup \{m\})''). \tag{4}$$

We call these operations CbO because the first one is used in well-known Close-by-One (CbO) Algorithm [3] for generating all concepts from $L(G, M, I)$.

Lemma 1. *Let (G, M, I) be a context, $(A, B) \in L(G, M, I)$, $g \in G$, and $m \in M$. Then*

$$CbO((A, B), g) = (A, B) \vee (\{g\}'', \{g\}'), \tag{5}$$

$$CbO((A, B), m) = (A, B) \wedge (\{m\}', \{m\}''). \tag{6}$$

This lemma proves the correctness of definition 1 of operations CbO. Most important property of these operations is represented in the following

Lemma 2. *Let (G, M, I) be a context, $(A_1, B_1), (A_2, B_2) \in L(G, M, I)$, $g \in G$, and $m \in M$. Then*

$$(A_1, B_1) \leq (A_2, B_2) \Rightarrow CbO((A_1, B_1), g) \leq CbO((A_2, B_2), g), \tag{7}$$

$$(A_1, B_1) \leq (A_2, B_2) \Rightarrow CbO((A_1, B_1), m) \leq CbO((A_2, B_2), m). \tag{8}$$

2 FCA-based Machine Learning

We deal with supervised Machine Learning. Hence we have training examples together with the target values on them. All examples are described by binary attributes from M, i.e. they can be given by bitsets of fixed length. Usually, a subset of examples is used as a **test sample** G^τ for checking the quality of training. The training examples are divided into positive G^+ and negative G^- subsets according to the value of the target attribute. The elements of G^+ and G_- make the **training sample**, elements of G^- are called **counter-examples** (**obstacles**). Formal context (G^+, M, I) is the main data set.

The well-known example $(S, S, \neq)$ of context for Boolean algebra demonstrates difficulties of brute force approach. For Boolean algebra of all subsets of n elements the context uses n^2 bits, and all the concepts need $n \cdot 2^n$ bits. For $n = 32$ the first number is 1 Kb (or 128 bytes) and the second one is 16 Gigabytes! The time complexity is exponential too.

Hence we need to replace computation of the whole lattice of all concepts by randomized algorithms to generate a random subset of the lattice. The author introduced and investigated mathematical properties of several algorithms of this kind, the best of which are variants of coupling Markov chains.

Now we represent the classical version of coupling Markov chain that is a core of probabilistic approach to machine learning based on FCA (VKF-method).

Data: context (G^+, M, I), external function $CbO(\ ,\ )$
Result: random concept $(A, B) \in L(G^+, M, I)$
$X := G \sqcup M$; $(A, B) := (M', M)$; $(C, D) = (G, G')$;
while $((A \neq C) \vee (B \neq D))$ **do**
  select random element $x \in X$;
  $(A, B) := CbO((A, B), x)$;
  $(C, D) := CbO((C, D), x)$;
end

Algorithm 1: Coupling Markov chain

The ordering of two concepts $(A, B) \leq (C, D)$ at any intermediate step of the while loop of Algorithm 1 is defined by Lemma 2.

For Boolean lattice (contranomial) context the author [8] computed the mean length of trajectory of Algorithm 1 as $n \sum_{j=1}^{n} \frac{1}{n}$ and proved strong concentration of length of arbitrary trajectory about its mean. For $n = 32$ the mean is ≤ 130, hence every trajectory generates about 260 (since two concepts is a state of the coupling Markov chain) subsets. Hence, only a small fraction of concepts occurs during computation of a moderate size subset of the Boolean algebra. We have in mind that there are 4,294,967,296 elements of Boolean algebra on 32 attributes.

Machine Learning procedure has two steps: induction and prediction. At the first step the system generate hypotheses about causes of the target property from training sample. At the prediction step the system applies the hypotheses to predict the target value for test examples.

The induction step of FCA-based learning applies the Coupling Markov chain Algorithm 1 to generate a random formal concept $(A, B) \in L(G^+, M, I)$. The program saves the concept (A, B) if there is no obstacle (counter-example) $o \in G^-$ such that $B \subseteq o'$.

Data: number N of concepts to generate
Result: random sample S of formal concepts without obstacles
$G^+ :=$ (+)-examples, $M :=$ attributes; $I \subseteq G^+ \times M$ is a formal context for (+)-examples;
$G^- :=$ (-)-examples; $S := \emptyset$; $i := 0$;
while $(i < N)$ **do**
  Generate concept (A, B) by Algorithm 1;
  $hasObstacle :=$ **false**;
  for $(o \in G^-)$ **do**
    if $(B \subseteq o')$ **then**
      $hasObstacle :=$ **true**;
    end
  end
  if $(hasObstacle =$ ***false***$)$ **then**
    $S := S \cup \{(A, B)\}$;
    $i := i + 1$;
  end
end

Algorithm 2: Inductive generalization

Condition $(B \subseteq o')$ of Algorithm 2 means the inclusion of intent B of concept (A, B) into the intent of counter-example o.

If a concept "avoids" all such obstacles it is added to the result set of all the concepts without obstacles.

We replace a time-consuming deterministic algorithm (for instance, "Close-by-One" [3]) for generation of all concepts by the probabilistic one to randomly generate the prescribed number of concepts.

The goal of Markov chain approach is to select a random sample of formal concepts without computation of the (possibly exponential size) set $L(G, M, I)$ of all the concepts.

Finally, machine learning program predicts the target class of test examples and compares the results of prediction with the original target value.

Data: random sample S of concepts, list G^τ of test objects
Result: prediction of target class of G^τ elements
for $(o \in G^\tau)$ **do**
  $PredictPositively(o) :=$ **false**;
  for $((A, B) \in S^+)$ **do**
    if $(B \subseteq o')$ **then**
      $PredictPositively(o) :=$ **true**;
    end
  end
end

Algorithm 3: Prediction of target class by analogy

The author proved [8] the following theorem to estimate parameter N from Algorithm 2.

Test object o is an ε-**important** if probability of all concepts (A, B) with $B \subseteq \{o\}'$ exceeds ε.

Theorem 1. *For $n = |M|$ and for any $\varepsilon > 0$ and $1 > \delta > 0$ random sample S of concepts of cardinality*

$$N \geq \frac{2 \cdot (n+1) - 2 \cdot \log_2 \delta}{\varepsilon} \tag{9}$$

with probability $> 1 - \delta$ has property that every ε-important object o contains some concept $(A, B) \in S$ such that $B \subseteq \{o\}'$.

This theorem is an analogue of the famous results of V. Vapnik and A. Chervonenkis [7] from Computational Learning Theory (here $n + 1$ corresponds to $\log_2 d$, where d is a VC-dimension).

From the practical point of view this theorem asserts the sufficiency of polynomial number of random concepts as causes of the target property to minimize 1-type error (wrong prediction of positive test examples) with respect to prediction by analogy (Algorithm 3).

3 Continuous attributes

3.1 Entropy approach

Let $G = G^+ \cup G^-$ be a disjoint union of training examples G^+ and counterexamples G^-. Interval $[a, b) \subseteq \mathbb{R}$ of values of continuous attribute $V : G \to \mathbb{R}$ generates three subsets

$$G^+[a, b) = \{g \in G^+ : a \leq V(g) < b\},$$

$$G^-[a,b) = \{g \in G^- : a \le V(g) < b\},$$
$$G[a,b) = \{g \in G : a \le V(g) < b\}$$

.

Definition 2. ***Entropy*** *of interval* $[a,b) \subseteq \mathbb{R}$ *of values of continuous attribute* $V : G \to \mathbb{R}$ *is*

$$\mathrm{ent}[a,b) = -\frac{|G^+[a,b)|}{|G[a,b)|} \cdot \log_2 \left(\frac{|G^+[a,b)|}{|G[a,b)|} \right) - \frac{|G^-[a,b)|}{|G[a,b)|} \cdot \log_2 \left(\frac{|G^-[a,b)|}{|G[a,b)|} \right) \quad (10)$$

Mean information *for partition* $a < r < b$ *of interval* $[a,b) \subseteq \mathbb{R}$ *of values of continuous attribute* $V : G \to \mathbb{R}$ *is*

$$\inf[a,r,b) = \frac{|E[a,r)|}{|E[a,b)|} \cdot \mathrm{ent}[a,r) + \frac{|E[r,b)|}{|E[a,b)|} \cdot \mathrm{ent}[r,b). \quad (11)$$

Threshold *is a value* $V = r$ *with minimal mean information.*

For continuous attribute $V : G \to \mathbb{R}$ denote $a = \min V$ by v_0 and let v_{l+1} be an arbitrary number greater then $b = \max V$. Thresholds $\{v_1 < \ldots < v_l\}$ are computed sequentially by splitting the largest entropy subinterval.

These constructions were introduced by J.R. Quinlan for C4.5, the well-known system for learning Decision Trees [5].

Definition 3. *For each* $1 \le i \le l$ *indicator (Boolean) variables corresponds to*

$$\delta_i^V(g) = 1 \Leftrightarrow V(g) \ge v_i \quad (12)$$
$$\sigma_i^V(g) = 1 \Leftrightarrow V(g) < v_i \quad (13)$$

Then string $\delta_1^V(g) \ldots \delta_l^V(g) \sigma_1^V(g) \ldots \sigma_l^V(g)$ *is a* ***bitset-representation*** *of continuous attribute* V *on element* $g \in G$.

Lemma 3. *Let* $\delta_1^{(1)} \ldots \delta_l^{(1)} \sigma_1^{(1)} \ldots \sigma_l^{(1)}$ *represent* $v_i \le V(A_1) < v_j$ *and* $\delta_1^{(2)} \ldots \delta_l^{(2)} \sigma_1^{(2)} \ldots \sigma_l^{(2)}$ *represent* $v_n \le V(A_2) < v_m$. *Then*

$$(\delta_1^{(1)} \& \delta_1^{(2)}) \ldots (\delta_l^{(1)} \& \delta_l^{(2)}) (\sigma_1^{(1)} \& \sigma_1^{(2)}) \ldots (\sigma_l^{(1)} \& \sigma_l^{(2)})$$

corresponds to $\min\{v_i, v_n\} \le V((A_1 \cup A_2)'') < \max\{v_j, v_m\}$.

In other words, Lemma 3 asserts that the result of bit-wise multiplication of bitset representations is a convex hull of its arguments' intervals.

The proof follows immediately from definition 3.

Similar bitset presentation for continuous features was mentioned earlier in [4] for interval pattern structures. However this work uses a priori given subdivision of a feature domain into disjoint subintervals.

3.2 Logistic regression between attributes

A **classifier** is a map $c : \mathbb{R}^d \to \{0, 1\}$, where $\mathbb{R}^d$ is a domain of objects to classify (described by d attributes) and $\{0, 1\}$ are *class marks*.

Probability distribution of $\langle \boldsymbol{X}, K\rangle \in \mathbb{R}^d \times \{0, 1\}$ can be decomposed as

$$p_{\boldsymbol{X},K}(\boldsymbol{x}, k) = p_{\boldsymbol{X}}(\boldsymbol{x}) \cdot p_{K|\boldsymbol{X}}(k \mid \boldsymbol{x}),$$

where $p_{\boldsymbol{X}}(\boldsymbol{x})$ is a marginal distribution of objects and $p_{K|\boldsymbol{X}}(k \mid \boldsymbol{x})$ is a conditional distribution of marks on given object, i.e. for every $\boldsymbol{x} \in \mathbb{R}^d$ the following $p_{K|\boldsymbol{X}}(k \mid \boldsymbol{x}) = \mathbb{P}\{K = k \mid \boldsymbol{X} = \boldsymbol{x}\}$ holds.

Error probability of classifier $c : \mathbb{R}^d \to \{0, 1\}$ is

$$R(c) = \mathbb{P}\left\{c(\boldsymbol{X}) \neq K\right\}. \tag{14}$$

Bayes classifier $b : \mathbb{R}^d \to \{0, 1\}$ with respect to $p_{K|\boldsymbol{X}}(k \mid \boldsymbol{x})$ corresponds to

$$b(\boldsymbol{x}) = 1 \Leftrightarrow p_{K|\boldsymbol{X}}(1 \mid \boldsymbol{x}) > \frac{1}{2} > p_{K|\boldsymbol{X}}(0 \mid \boldsymbol{x}) \tag{15}$$

We remind well-known

Theorem 2. *The Bayes classifier b has the minimal error probability:*

$$\forall c : \mathbb{R}^d \to \{0, 1\}\,[R(b) = \mathbb{P}\{b(\boldsymbol{X}) \neq K\} \leq R(c)]$$

Bayes Theorem implies

$$p_{K|\boldsymbol{X}}(1 \mid \boldsymbol{x}) = \frac{p_{\boldsymbol{X}|K}(\boldsymbol{x} \mid 1) \cdot \mathbb{P}\{K = 1\}}{p_{\boldsymbol{X}|K}(\boldsymbol{x} \mid 1) \cdot \mathbb{P}\{K = 1\} + p_{\boldsymbol{X}|K}(\boldsymbol{x} \mid 0) \cdot \mathbb{P}\{K = 0\}} =$$
$$= \frac{1}{1 + \frac{p_{\boldsymbol{X}|K}(\boldsymbol{x}|0) \cdot \mathbb{P}\{K=0\}}{p_{\boldsymbol{X}|K}(\boldsymbol{x}|1) \cdot \mathbb{P}\{K=1\}}} = \frac{1}{1 + \exp\{-a(\boldsymbol{x})\}} = \sigma(a(\boldsymbol{x}))$$

where $a(\boldsymbol{x}) = \log \frac{p_{\boldsymbol{X}|K}(\boldsymbol{x}|1) \cdot \mathbb{P}\{K=1\}}{p_{\boldsymbol{X}|K}(\boldsymbol{x}|0) \cdot \mathbb{P}\{K=0\}}$ and $\sigma(y) = \frac{1}{1+\exp\{-y\}}$ is the well-known **logistic function**.

Equation (15) transforms to

$$b(\boldsymbol{x}) = 1 \Leftrightarrow a(\boldsymbol{x}) > 0 \tag{16}$$

Let approximate unknown $a(\boldsymbol{x}) = \log \frac{p_{\boldsymbol{X}|K}(\boldsymbol{x}|1) \cdot \mathbb{P}\{K=1\}}{p_{\boldsymbol{X}|K}(\boldsymbol{x}|0) \cdot \mathbb{P}\{K=0\}}$ by linear combination $\boldsymbol{w}^T \cdot \varphi(\boldsymbol{x})$ of basis functions $\varphi_i : \mathbb{R}^d \to \mathbb{R}$ $(i = 1, \ldots, m)$ with respect to unknown weights $\boldsymbol{w} \in \mathbb{R}^m$.

For training sample $\langle \boldsymbol{x}_1, k_1\rangle, \ldots, \langle \boldsymbol{x}_n, k_n\rangle$ introduce $t_j = 2k_j - 1$. Then

$$\log\{p(t_1, \ldots, t_n \mid \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n, \boldsymbol{w})\} = -\sum_{j=1}^{n} \log\left[1 + \exp\{-t_j \sum_{i=1}^{m} w_i \varphi_i(\boldsymbol{x}_j)\}\right].$$

Lemma 4. $\log\left[1 + \exp\{-t \cdot \sum_{i=1}^{m} w_i \varphi_i\}\right]$ *is a convex function of* $\boldsymbol{w}$.

Hence, the logarithm of likelihood

$$L(w_1,\dots,w_m) = -\sum_{j=1}^{n} \log\left[1+\exp\{-t_j \sum_{i=1}^{m} w_i\varphi_i(\boldsymbol{x}_j)\}\right] \to \max \tag{17}$$

is concave.

Newton-Raphson method leads to iterative procedure

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - (\nabla_{\boldsymbol{w}^T}\nabla_{\boldsymbol{w}} L(\boldsymbol{w}_t))^{-1}\cdot\nabla_{\boldsymbol{w}} L(\boldsymbol{w}_t). \tag{18}$$

Use $s_j = \frac{1}{1+\exp\{t_j\cdot(w^T\cdot\Phi(x_j))\}}$ we obtain

$$\nabla L(\boldsymbol{w}) = -\Phi^T \mathrm{diag}(t_1,\dots,t_n)\boldsymbol{s}, \nabla\nabla L(\boldsymbol{w}) = \Phi^T R\Phi,$$

where $R = \mathrm{diag}(s_1(1-s_1), s_2(1-s_2),\dots,s_n(1-s_n))$ is diagonal matrix with elements $s_1(1-s_1), s_2(1-s_2),\dots,s_n(1-s_n)$ and $\mathrm{diag}(t_1,\dots,t_n)\boldsymbol{s}$ is vector with coordinates $t_1s_1, t_2s_2,\dots,t_ns_n$.

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \left(\Phi^T R\Phi\right)^{-1}\Phi^T\mathrm{diag}(t)\boldsymbol{s} = (\Phi^T R\Phi)^{-1}\Phi^T R\boldsymbol{z}, \tag{19}$$

where $\boldsymbol{z} = \Phi\boldsymbol{w}_t + R^{-1}\mathrm{diag}(t_1,\dots,t_n)\boldsymbol{s}$ are iterative calculated weights.

As usual, the ridge regression helps to avoid ill-conditioned situation

$$\boldsymbol{w}_{t+1} = (\Phi^T R\Phi + \lambda\cdot I)^{-1}\cdot(\Phi^T R\boldsymbol{z}).$$

In the computer program 'VKF system' we use standard basis: constant 1 and attributes themselves.

At last, we need a criterion for significance of regression. For logistic regression two types of criteria were applied:

Criterion of Cox-Snell declares attribute V_k significant, if

$$R^2 = 1-\exp\{2(L(w_0,\dots,w_{k-1}) - L(w_0,\dots,w_{k-1},w_k))/n\} \geq \sigma. \tag{20}$$

McFadden criterion declares attribute V_k significant, if

$$1 - \frac{L(w_0,\dots,w_{k-1},w_k)}{L(w_0,\dots,w_{k-1})} \geq \sigma. \tag{21}$$

Conclusion

We have extended the 'VKF system' approach to FCA-based machine learning on examples with both discrete and continuous attributes.

Experiments with Wine Quality Dataset [6] demonstrate a very good behavior of the proposed approach. For red wines with high scores (more than 7) all examples were classified correctly.

The pair-wise logistic regression is combined with single threshold computation. Lemma 3 gives a condition of non-triviality of similarity on values of continuous attribute: if the corresponding part of the resulting bitset is non-void, then the values $V(B')$ belong to a common interval.

When analyzing relationship between 'alcohol' and 'sulphates' for red wines we observe a phenomenon directly corresponding to the well-known

Lemma 5. *Disjunction* $x_{i_1} \vee \ldots \vee x_{i_k}$ *of Boolean variables holds, if and only if* $x_{i_1} + \ldots + x_{i_k} \geq \sigma$ *holds for any* $0 < \sigma < 1$.

Positive (but slightly different) weights correspond to different scaling of various attributes. So we have not only conjuction of attributes by also a disjunction. Similar case is a relationship between 'citric acid' and 'alcohol'.The situation with the pair ('pH', 'alcohol') is radically different. The alcohol's weight is positive, whereas pH's weight is negative. With the help of aforementioned lemma and standard logic we obtain the implication ('pH'⇒'alcohol').

Acknowledgements

The author would like to thank Prof. S.O. Kuznetsov for motivation and support. He is also grateful to his colleagues from Dorodnicyn Computing Center of Federal Research Center "Computer Science and Control" of Russian Academy of Science and Russian State University for Humanities for long-term support and useful discussions. Finally, the author is grateful to anonymous reviewers, whose comments helped to significantly improve the presentation.

References

1. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
2. Kaytoue, M., Kuznetsov, S.O., Napoli, A., Duplessis, S.: Mining gene expression data with pattern structures in formal concept analysis. Inf. Sci. 181(10): 1989-2001 (2011)
3. Kuznetsov, S.O.: A Fast Algorithm for Computing all Intersections of Objects in a Finite Semi-Lattice. Autom. Doc. Math. Linguist. 27 (5), 11-21 (1993)
4. Makhalova, M., Kuznetsov, S.O., Napoli, A.: Numerical Pattern Mining through Compression. 2019 Data Compress. Conf. Proc. IEEE. 112-121 (2019)
5. Quinlan, J.R.: C4.5 Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
6. UCI Machine Learning Repository: Wine Quality Data Set, https://archive.ics.uci.edu/ml/datasets/Wine+Quality. Last accessed 20 June 2020
7. Vapnik, V., Chervonenkis, A.: On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. Theory Probab. Appl. 16 (2), 264–280 (2004)
8. Vinogradov, D.V.: Machine Learning Based on Similarity Operation. In: Kuznetsov S., Osipov G., Stefanuk V. (eds) Artificial Intelligence. RCAI 2018. Communications in Computer and Information Science. **934**, 46–59 (2018)
9. Vinogradov, D.V.: On Object Representation by Bit Strings for the VKF-Method. Autom. Doc. Math. Linguist. 52 (4), 113–116 (2018)

Granular Computing and Incremental Classification

Xenia Naidenova[1] and Vladimir Parkhomenko[2]

[1] Military Medical Academy, Saint-Petersburg, Russia
ksennaidd@gmail.com,
ORCID 0000-0003-2377-7093
[2] Peter the Great St. Petersburg Polytechnic University, Saint-Petersburg, Russia
parhomenko.v@gmail.com,
ORCID 0000-0001-7757-377X

Abstract. A problem of incremental granular computing is considered in the tasks of classification reasoning. Objects, values of attributes, partitions of objects (classifications) and Good Maximally Redundant Tests (GMRTs) (special kind of formal concepts) are considered as granules. The paper deals with inferring GMRTs. They are good tests because they cover the largest possible number of objects w. r. t. inclusion relation on the set of all object subsets. Two kinds of classification subcontexts are defined: attributive and object ones. The context decomposition leads to a mode of incremental learning GMRTs. Four cases of incremental learning are proposed: adding a new object (attribute value) and deleting an object (attribute value). Some illustrative examples of four cases of incremental learning are given too.

Keywords: Granular computing · incremental classification · good test · formal concept

1 Introduction

Information granules are becoming important entities in data processing at the different levels of data abstraction. Information granules have also contributed to increasing the precision in data processing [9]. Application of information granules is one of the problem-solving methods based on decomposing a big problem into subtasks.

Several studies devoted to evolving information granules to adapt to changes in the streams of data are described in [12]. The process of forming information granules is often associated with the removal of some element of data or dealing with incomplete data [1]. Generally, we consider object, property, class of objects, and classification as the main granules of human classification reasoning.

The paper deals with inferring good classification (diagnostic) tests. Tests are good because they cover the largest possible number of objects w. r. t. the inclusion relation on the set of all object subsets.

Two kinds of classification subcontexts are defined: attributive and object ones. The context decomposition leads to a mode of incremental learning good

classification tests. Four cases of incremental learning are proposed: adding a new object (attribute value) and deleting an object (attribute value). Some heuristic rules allowing decreasing the computational complexity of inferring good tests are considered too.

In [7], it is considered the link between Good Test Analysis (GTA) and Formal Concept Analysis (FCA) [5]. To give a target classification of objects, we use an additional attribute $KL \notin U$. In Tab. 1, we have classification KL containing two classes: the objects in whose descriptions the target value $k(+)$ appears and all the other objects.

Table 1. Example of classification

Index	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	KL
1	2	3	1	2	1	2	2	2	k(+)
2	2	2	2	2	1	2	0	2	k(+)
3	2	3	1	3	1	2	2	2	k(+)
4	2	2	3	1	1	0	2	2	k(+)
5	2	3	2	2	1	2	2	0	k(+)
6	2	3	1	3	0	0	2	2	k(+)
7	2	3	1	2	1	2	2	0	k(-)
8	1	4	2	1	0	2	0	2	k(-)
9	2	3	1	3	1	0	2	2	k(-)
10	1	4	2	2	1	2	2	2	k(-)
11	2	4	1	2	0	2	2	2	k(-)
12	2	4	2	2	1	2	0	0	k(-)

2 Three interrelated sets of classes, objects, and properties

The "atom" of plausible human reasoning is a concept. The concepts are represented by theirs names. We shall consider the following roles of names in reasonings: a name can be the name of object, the name of class of objects and the name of classification or collection of classes. With respect to the role of name in knowledge representation schemes, it can be the name of attribute or attribute's value. A class of objects may contain only one object; hence the name of the object is a case of the name of a class. For example, fir-tree can be regarded as the name of a tree or the name of a class of trees. Each attribute genarates a classification of a given set of objects; hence the names of attributes can be the names of classifications and the attribute values can be the names of classes. In the knowledge bases, the sets of names for objects, classes and classifications must not intersect.

Let k be the name of an objects' class, c be the name of a property of objects (value of an attribute), and g be the name of an object. Each class or property has only one maximal set of objects as its interpretation that is the set of objects belonging to this class or possessing this property: $k \rightarrow I(k) = \{g : g \leq k\}$, $c \rightarrow I(c) = \{g : \ g \leq c\}$, where the relation '$\leq$' denotes 'is a' relation and has causal nature (the dress is red, an apple is a fruit). Each object has only one corresponding set of all its properties:$C(g) = \{c : g \leq c\}$. We shall say that $C(g)$ is the description of object g. The link $g \rightarrow C(g)$ is also of causal nature. We shall say that $C(k) = \{\cap C(g) : g \leq k\}$ is the description of class k, where $C(k)$ is a collection of properties associated with each object of class k. The link $k \rightarrow C(k)$ is also of causal nature. Figure 1 illustrates the causal links between classes of objects, properties of objects, and objects.

Clearly, each description (a set of properties) has one and only one interpretation (the set of objects possessing this set of properties). But the same set of objects can be the interpretation of different descriptions (equivalent with respect to their interpretations). The equivalent descriptions of the same class are said to be the different names of this class. The task of inferring the equivalence relations between names of classes and properties underlies the processes of plausible reasoning.

The identity has the following logical content: class K is equivalent to property k ($(K \leftrightarrow k)$) if and only if the interpretations $I(K), I(k)$ on the set of conceivable objects are equal $I(K) = I(k)$. It is possible to define also the relationship of approximate identity between concepts: k approximates B ($k \leq B$) if and only if the relation $I(k) \subseteq I(K)$ is satisfied. We can consider instead of one property (concept) any subset of properties joined by the union $\cup$ operation: $(c_1 \cup c_2 \cup \ldots \cup c_i \cup \ldots \cup c_n) \leq B$.

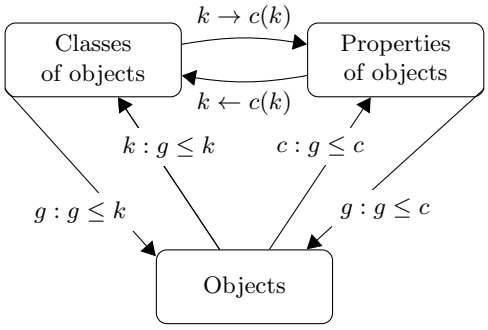


Fig. 1. Links between Objects, Classes, and Properties of Objects

Connection directed from 'Properties of objects' to 'Classes of objects' is constructed by learning from examples of objects and their classes. This connection

is also of causal nature, it is expressed via the "if – then" rule: to say that "if tiger, then mammals" means to say that I(tiger) $\subseteq$ I(mammals).

3 Background definitions

.

Let G = {1, 2,. . ., N} be the set of objects' indices (objects, for short) and M = {m_1, m_2, . . ., m_j, . . . m_q} be the set of attributes' values (values, for short). Each object is described by a set of values from M. The object descriptions are represented by rows of a table R the columns of which are associated with the attributes taking their values in M. Denote a description of $g \in G$ by $\delta(g)$. Let D_+ and G_+ ($D_- = D/D_+$) and $G_- = G/G_+$) be the sets of positive or negative object descriptions and the set of indices of these objects, respectively. The definition of good tests is based on two mapping $2^G \to 2^M$, $2^M \to 2^G$. Let A $\subseteq$ G, B $\subseteq$ M. Denote by B_i, $B_i \subseteq M$, i = 1,. . ., N the description of object with index i. The relations $2^G \to 2^M$, $2^M \to 2^G$ are: A' = val(A) = {intersection of all B_i: $B_i \subseteq M$, i $\in$ A} and B' = obj(B) = {i: i $\in$ G, B $\subseteq B_i$}.

These mapping are the Galois's correspondences. Operations val(A), obj(B) are reasoning operations (derivation operations). We introduce two generalization operations: generalization_of(B) = B''= val(obj(B); generalization_ of(A) = A'' = obj(val(A)). These operations are the closure operations [8]. A set A is closed if A = obj(val(A)). A set B is closed if B = val(obj(B)). For g $\in$ G and m $\in$ M, g' is called object intent and m' is called value extent. We illustrate the derivation and generalization operations (Tab. 1):

A = {4, 8}, val(A) = {$x_4 = 1$, $x_8 = 2$}; A'' = obj({x_4 =1, $x_8 = 2$}) = {4, 8} = A;

m = {$x_8 = 0$}, obj ({$x_8 = 0$}) = {5, 7, 12}; m'' = val({5, 7,12}) = {$x_1 = 2$, $x_4 = 2$, $x_5 = 1$, $x_6 = 2$, $x_8 = 0$ }; B = {$x_4 = 3$, $x_5 = 1$}, obj({B}) = {3,9}; B'' = val({3, 9}) = {$x_1 = 2$, $x_2 = 3$, $x_3 = 1$, $x_4 = 3$, $x_5 = 1$, $x_7 = 2$, $x_8 = 2$}.

Definition 1. A Diagnostic Test (DT) for G_+ is a pair (A, B) such that $B \subseteq M, A = obj(B) \neq \emptyset$, $A \subseteq G_+$, and $obj(B) \cap \delta(g) = \emptyset, (\forall g) g \in G_-$.

Definition 2. A Diagnostic Test (DT) for G_+ is maximally redundant if $obj(B \cup m) \subset A$ for all $m \in M \setminus B$.

Definition 3. A Diagnostic Test (DT) for G_+ is good iff any extension $A^* = A \cup i$, $i \in G_+ \setminus A$, implies that $(A^*, val(A^*))$ is not a test for G_+.

Note that the definition of tests for G_+ does not differ from the definition of positive hypotheses given in [6] and [4] in the language of predicates. Definitions 2, 3, 4 remain true if G_+ is replaced by G_-. In what follows, we are interested in inferring GMRTs for positive class of objects. As far as Formal Concept Analysis development and application, the following surveys [11,10,3,2] can be seen

Some examples of formal concepts are in Tab. 1. Let us check if a pair (A, B) = ((1,5, 6,7,9), (x_1=2, $x_2 = 3$, $x_7 = 2$)) is a concept or not. It is a concept, because obj((x_1=2, $x_2 = 3$, $x_7 = 2$) = (1,5,6,7,9) = A. However, this concept does not distinguish the classes of objects. Pair ((11,12), ($x_1 = 2$, $x_2 = 4$, x_4 =

2)) is a concept and a test for k(−), but not a good one, because there is pair ((8, 10, 11, 12), ($x_2 = 4$)) such that (11, 12) ⊂ (8, 10, 11 12).

4 Incremental learning GMRTs

Define two kinds of subtasks:

1. to find all GMRTs intents of which are included in the description of an object;
2. to find all GMRTs into intents of which a given set of values is included.

To solve these subtasks, we need to form subcontexts (projections) of a given classification context.

Definition 4. Let $B \subseteq M$. The object projection $proj(B, G_+)$ on G_+ is $proj(B, G_+) = \{\delta(g) \cap B | g \in G_+, \delta(g) \cap B \neq \emptyset$, and $(obj(\delta(g) \cap B), \delta(g) \cap B)$ is a test for $G_+\}$.

An example of object projection $proj(d_2)$ on G_+ is in Tab. 2.

Table 2. Example of object 2 projection

Index	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	KL	Test?
1	2			2	1	2		2	k(+)	no
2	2	2	2	2	1	2	0	2	k(+)	yes
3	2				1	2		2	k(+)	yes
4	2	2			1			2	k(+)	yes
5	2		2	2	1	2			k(+)	no
6	2							2	k(+)	no

Definition 5. Let $X \subseteq M$. The value projection $proj$ (X, G_+) on G_+ is $proj(X, G_+) = \{\delta(g) | g \in G_+, X \subseteq \delta(g)\}$.

An example of value projection $proj(x_5 = 1, x_6 = 2)$ on G_+ can be presented as descriptions of 1, 2, 3 and 5 objects.

5 Four cases of incremental learning of classification context

We propose four cases of modifying classification contexts: adding/removing objects and adding/removing values of attributes. Modification of GMRTs is based on a decomposition of classification contexts into value and object subcontexts and inferring GMRTs in them. Let $STGOOD_+$ and $STGOOD_-$ be the current sets of extents of GMRTs for positive and negative class of objects, respectively.

We mean that the process in which a change of the classification context implies only updating the sets $STGOOD_+$ and $STGOOD_-$. The classification context can be changed as follows: a new object is added with the indication of its class membership; an object is deleted from G+ or G-; a value is introduced into the classification context; a value is deleted from the classification context. Updating $STGOOD_+$ and $STGOOD_-$ is performed with the use of only subcontexts associated with added (deleted) object or value.

Case 1 The following actions are necessary:

1. Checking whether it is possible to extend the extents of some existing GMRTs for the class to which a new object belongs (a class of positive objects, for certainty).
2. Inferring all GMRTs, intents of which are included into the new object description; for this goal, the first kind subtask is used.
3. Deleting GMRTs for positive class extents of which are included in the extent of any new GMRTs.
4. Checking the validity of GMRTs for negative objects, and, if it is necessary, modifying invalid GMRTs (test for negative objects is invalid if its intent is included in a new (positive) object description); for this goal, the second kind subtask is used.

Let $s \in STGOOD_-$ and $Y = val(s)$. If $Y \subseteq t_{new}(+)$, then s should be deleted from $STGOOD_-$ because (s, Y) is invalid test for G_-.

Proposition 1. $obj(Y)$ forms the subcontext for finding corrected tests for G_-.

Proof. $Y \subseteq X \leftrightarrow obj(X) \subseteq obj(Y)$. Assume that there exists a GMRT (with intent Z) for G_- such that $obj(Z) \not\subset$ and $\neq obj(Y)$. Then $obj(Z)$ contains some objects not belonging to $obj(Y)$ and Z will be included in some descriptions of objects not belonging to $obj(Y)$ and, consequently, Z has been obtained at the previous steps of the incremental algorithm for finding all GMRTs for G_-.

Consider an example of adding object in the process of inferring GMRTs for the data in Tab. 1. Let us fix the classification context with 3 first objects from G_+ and all the objects of G_-. For this current situation we have one GMRT for G_+, namely, ((1,2,3) (x_1=2, $x_5 = 1$, $x_6 = 2$, $x_8 = 2$)) and one GMRT for G_-, namely, ((8,10, 11, 12), ($x_2 = 4$, $x_6 = 2$)). As a result of adding object 4 into positive class of objects, we obtain a new GMRT for positive class of objects ((2,4), ($x_1 = 2$, $x_2 = 2$, $x_5 = 1$, $x_8 = 2$)). GMRTs for negative class of objects do not changed.

Case 2 Suppose that an object g is deleted from G_+ (G_-) The following actions are necessary:

1. $\forall s$, $s \in STGOOD_+$ ($STGOOD_-$), $g \in s$, delete g from s; in this connection, we observe that $(s \backslash g, val(s \backslash g))$ remains to be the test for G_+ (G_-).
2. We denote modified test $(s \backslash g, val((s \backslash g))$ by MT. We have the following possibilities:

- the intent of MT has not changed; then MT is a GMRT for $G_+(G_-)$ in the modified context;
- the intent of MT has changed and the extent of MT is included in the extent of an existing GMRT for $G_+(G_-)$, then MT must be deleted; otherwise MT is a GMRT for $G_+(G_-)$.

An example of deleting object. Let us fix the whole classification context (Tab. 1). We have one GMRT for G- obtained in this context: ((8,10,11,12), (x_2 = 4, x_6 = 2). Delete object 8. We have one modified test: ((10, 11, 12), (x_2 =4, x_4 = 2, x_6 = 2)) and it is the GMRT.

Case 3 Suppose that a new value m^* is added to the classification context: m^* appears in the descriptions of some positive and negative objects and $M ::= m^* \cup M$. The task of finding all GMRTs for $G_+(G_-)$ whose intents contain m^* is reduced to the task of the second kind. The subcontext for this task is determined by the set of all objects whose descriptions contain m^*. As result, we obtain all the GMRTs $(obj(Y), Y)$ for $G_+(G_-)$ such that $m^* \in Y$. We can add a set of values if we want that all these values will be included simultaneously in the intents of GMRTs.

Case 4 Suppose that some value m is deleted from the classification context. Let a GMRT $(obj(X), X)$ for $G_+(G_-)$ be transformed into $(obj(X \setminus m), X \setminus m)$. Then we have $((X \setminus m) \subset X) \leftrightarrow (obj(X) \subseteq obj(X \setminus m))$.

Consider two possibilities: $obj(X \setminus m) = obj(X)$ and $obj(X) \subset obj(X \setminus m)$. In the first case, $(obj(X \setminus m), X \setminus m)$ is a GMRT for $G_+(G_-)$. In the second case, $(obj(X \setminus m), X \setminus m)$ is not a test. However, $obj(X \setminus m)$ can contain extents of new GMRTs for $G_+(G_-)$ and these tests can be obtained by using the subtask of the second kind.

An example of deleting value $x_6 = 2$. The GMRT for the negative class ((8, 10, 11, 12), ($x_8 = 4$)) remains GMRT. The GMRTs ((2, 4), ($x_1 = 2$, $x_2 = 2$, $x_5 = 1$, $x_8 = 2$)) remains GMRT, but the pair ((1, 2, 3), ($x_1 = 2$, $x_5 = 1$, $x_8 = 2$)) is not a test for positive objects after deleting $x_6 = 2$. It is impossible to find any GMRTs for positive objects 1, 2, 3, 4.

Recognizing the class membership for a new object not belonging to training set is performed as follows:

- If (and only if) description of object contains an intent of GMRT of only one class, then the object can be assigned to this class;
- If description of an object does not contain any intent of GMRTs, then we have the case of uncertainty.

In two last cases, it is necessary to continue learning by adding new objects or to change the classification context.

6 Conclusions

The paper examines the relationship between an incremental model of good classification test inferring with granular computing. In the process of finding

good tests, the following granules are highlighted: objects, attribute values, and object classes. The incremental test inferring is carried out as a process in which granules can be added and removed, changing the classification context. The decomposition of classification contexts into subcontexts is considered based on the selection of objects and values of attributes (granules). Thus, granules become active elements of test inferring and allow this process to be made data driving based on data selection.

Acknowledgements

The research was partially supported by Russian Foundation for Basic Research, research project No. 18-07-00098A. This research work was supported by the Academic Excellence Project 5-100 proposed by Peter the Great St. Petersburg Polytechnic University.

References

1. Al-Hmouz, R., Pedrycz, W., Balamash, A.S., Morfeq, A.: Granular description of data in a non-stationary environment. Soft Computing 22(2), 523–540 (Jan 2018), https://doi.org/10.1007/s00500-016-2352-2
2. Buzmakov, A., Kuznetsov, S.O., Napoli, A.: A new approach to classification by means of jumping emerging patterns. In: CEUR Workshop Proceedings. vol. 939, pp. 15–22 (2012)
3. Codocedo, V., Napoli, A.: Formal concept analysis and information retrieval – a survey. In: Baixeries, J., Sacarea, C., Ojeda-Aciego, M. (eds.) Proceedings of the 13th ICFCA. pp. 61–77 (2015)
4. Finn, V.: On machine-oriented formalization of plausible reasoning in the style of F.Bacon–J.S. Mill. Semiotika i Informatika 20, 35–101 (1983), (in Russian)
5. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer, Berlin (1999)
6. Ganter, B., Kuznetsov, S.O.: Formalizing hypotheses with concepts. In: Conceptual Structures: Logical, Linguistic, and Computational Issues: Proceedings of the 8th International Conference on Conceptual Structures. pp. 342–356 (2000)
7. Naidenova, X.: Good classification tests as formal concepts. In: Domenach, F., Ignatov, D., Poelmans, J. (eds.) Proceedings of the 10th International Conference on Formal Concept Analysis. vol. 7278, pp. 211–226 (2012)
8. Ore, O.: Galois connections. Trans. Amer. Math. Soc 55, 494–513 (1944)
9. Pedrycz, W.: Allocation of information granularity in optimization and decision-making models: Towards building the foundations of granular computing. European Journal of Operational Research 232(1), 137 – 145 (2014)
10. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Systems with Applications 40(16), 6538 – 6560 (2013)
11. Poelmans, J., Kuznetsov, S.O., Ignatov, D.I., Dedene, G.: Formal Concept Analysis in knowledge processing: A survey on models and techniques. Expert systems with applications 40(16), 6601–6623 (NOV 15 2013)
12. Xu, J., Wang, G., Li, T., Pedrycz, W.: Local-density-based optimal granulation and manifold information granule description. IEEE Transactions on Cybernetics 48(10), 2795–2808 (Oct 2018)

Concept Lattice and Soft Sets. Application to the Medical Image Analysis.

Anca Ch. Pascu[1 [0000−1111−2222−3333]], Laurent Nana[2 [1111−2222−3333−4444]], and Mayssa Tayachi[3 [2222−−3333−4444−5555]]

Université de Brest, Lab-STICC, France
anca.pascu@univ-brest.fr
laurent.nana@univ-brest.fr

Abstract. In this paper we recall the basic mathematical fundamentals of Formal Concept Analysis (FCA) and analyse the possibilities to define soft sets from a concept lattice. We propose two ways of finding soft sets in a concept lattice of FCA model. We give also an example of soft set in the framework of digital medical image. We prove the usefulness of working with soft sets for the ROI - RONI categorization of a medical image.

Keywords: Formal Concept Analysis (FCA) · Objects · Attributes · Formal Concept · Concept Lattice (Galois Lattice) · Soft Set · Digital Medical Image · Region Of Interest (ROI) · Region Of Non Interest (RONI).

1 Introduction

This paper is an investigation of the links between two great mathematical theories: Formal Concept Analysis (FCA) and Soft Sets (SS). FCA is founded last 90' by a team of German and French researchers in Darmstadt and Paris. Theoretically speaking, it is based by a main theorem using the notion of Galois connection. The FCA has been developed a lot, especially from an application point of view, finding a good number of applications in different fields. The SS is newer (end of 90' by Molodtsov ([3]) and less known, perhaps. It is a pure categorization theory considered by some as a generalization of fuzzy sets theory. At the glance, the common point of both theories is the fact that they both are categorization theories.

In the theory of categorization, there are two classes of models:

1. Models whose categorization objects belong to a space X and for which the categorization criteria are also defined within the space X;
2. Models whose categorization objects are in a space X and for which the categorization criteria are defined outside of the space X;

The soft set theory is a theory of categorization giving a categorization of a space X, following a set of parameters outside X.

Both the soft set theory and FCA model are models of the second type. It is for this reason that we decided to study the two models from the point of view of their basic mathematical concepts. We found that the FCA model is a special case of a soft set. It is claimed that in particular classes of applications soft set modeling may be more profitable.

2 FCA model

The FCA model is presented following [1].

Definition 1. *(Formal context) A formal context,* **K,** is a triple **K** = (O, A, I) where O is a set of *objects*, A is a set of *attributes* and I is a binary relation from O to A defined by:
$\forall o \in O, a \in A$,
$I(o, a) = 1$ when the object o has the attribute a

$I(o, a) = 0$ otherwise.

Starting from binary relation I, one defines two *derivation operators* $\mathrm{I}^{\uparrow}$ and $\mathrm{I}^{\downarrow}$.

Definition 2. *(Derivation operators) Let $\mathcal{P}(O)$ ans $\mathcal{P}(A)$ be the set of al subsets of O respectively, A.*
The operator $I^{\uparrow}$ is defined as follows:
$I^{\uparrow} : \mathcal{P}(O) \to \mathcal{P}(A)$. For $X \subset O$,

$$I^{\uparrow}(X) = \{a \in A / I(o, a) = 1,\ \forall o \in X\} \tag{1}$$

The operator $I^{\downarrow}$ is defined as follows: $I^{\downarrow} : \mathcal{P}(A) \to \mathcal{P}(O)$. For $Y \subset A$,

$$I^{\downarrow}(X) = \{o \in O / I(o, a) = 1,\ \forall a \in Y\} \tag{2}$$

Three properties are established in [1]:

1. $\mathrm{X}_1 \subseteq \mathrm{X}_2 \Rightarrow \mathrm{I}^{\uparrow}(\mathrm{X}_1) \supseteq \mathrm{I}^{\uparrow}(\mathrm{X}_2)$;
2. $\mathrm{X} \subseteq \mathrm{I}^{\downarrow}(\mathrm{I}^{\uparrow}(\mathrm{X}))$ and $\mathrm{Y} \subseteq \mathrm{I}^{\uparrow}(\mathrm{I}^{\downarrow}(\mathrm{Y}))$;
3. $\mathrm{I}^{\uparrow}(\mathrm{I}^{\downarrow}(\mathrm{I}^{\uparrow}(\mathrm{X}))) = \mathrm{I}^{\uparrow}(\mathrm{X})$ and $\mathrm{I}^{\downarrow}(\mathrm{I}^{\uparrow}(\mathrm{I}^{\downarrow}(\mathrm{X}))) = \mathrm{I}^{\downarrow}(\mathrm{Y})$.

Definition 3. *(Formal concept)[1] A formal concept, C of a formal context* **K** *is a pair C = (X,Y) with X⊆ O, Y ⊆ A, $X = I^{\downarrow}(Y)$ and $Y = I^{\uparrow}(X)$. X is called the* extent*, denoted by Ext, and Y is called the* intent*, denoted by Int of the formal concept C.*

Definition 4. *(A formal concepts order)[1] Let be two concepts, C_1 and C_2. An order relation $\preceq$ is introduced by:*
$C_1 \preceq C_2 \Longleftrightarrow Ext(C_1) \subseteq Ext(C_2) \Longleftrightarrow Int(C_2) \subseteq Int(C_1)$

Taking into account properties 2 and 3, it is proved [1] that the set of all formal concepts of a context **K** is a *complete lattice* denoted by $\mathcal{G}(\mathbf{K})$. This lattice verifies the property of *Galois connection* [2] and it is is called *Galois lattice of concepts.*

3 Basic concepts of soft sets theory

Mathematical categorization theory contains several categorization models each based on a mathematical theory, be it algebraic, geometric, probabilistic or other. In classical models of categorization, the idea is to split a space X in categories taking into account one or several criteria defined also based on the space X. So these criteria are in some way internal to the space X.

A *soft set* is a model giving a categorization on a space X taking into account a *cognitive element* external to X. This feature of externality represents another point of view of categorization. In the following section we will explore this cognitive feature of categorization of soft sets in the framework of FCA.

We give some basic elements from soft set theory [3][4] A *soft set* is a parameterized family of sets - intuitively, this is "soft" because the boundary of the set depends on the parameters. Formally, a soft set is defined by:

Definition 5. *(Soft set) Let X be an initial universe set and E a set of parameters with respect to X. Let $\mathcal{P}(X)$ denote the power set of X and* $\mathbf{A} \subset$ E. A pair (F, $\mathbf{A}$) is called a soft set over X, where F is a mapping given by F : $\mathbf{A} \rightarrow \mathcal{P}$(X). In other words, a soft set (F, $\mathbf{A}$) over X is a parameterized family of subsets of X. For e $\in \mathbf{A}$, F (e) may be considered as the set of e-elements or e-approximate elements of the soft sets (F, $\mathbf{A}$). Thus (F, $\mathbf{A}$) is defined as:

$$(F, \mathbf{A}) = \{\text{F(e)} \in \mathcal{P}(\text{X}) \text{ if e} \in \mathbf{A}\} \text{ and } (\text{F}, \mathbf{A}) = \emptyset \text{ if e} \notin \text{A} \tag{3}$$

Remark 1. A soft set is a categorization of a space X guided by a set of parameters $\mathbf{A}$ and a function F establishing a correspondence between a parameter ans a subset of X.

4 Two soft set models of concept lattice

We explore in this section two possibilities to interpret the FCA model as a soft set. The "conceptual metaphor " generating this parallel is that relations between objects and attributes in the FCA model can be viewed as parameters. We have two options : either classify objects or classify formal concepts from the FCA model.

In the first case, one classifies FCA objects and the set $\mathbf{A}$ of parameters is the set A of FCA attributes. That is the categorization is constructed following attributes.
In the second case, one classifies FCA formal concepts and the set $\mathbf{A}$ of parameters is a subset of cartesian product O x A of FCA model.

Remark 2. The second SS model (Soft set 2) is possible because of the duality between extension and intension in the FCA model and the property of Galois connection.

4.1 First soft sets models of FCA

Definition 6. *(Soft set 1) Let be the formal context* $\mathbf{K}$ = (O, A, I) and its associated concept lattice $\mathcal{G}(\mathbf{K})$.
We define a soft set SS1 as follows:

SS1 = $\{F_1, \mathbf{A}\}$ where $\mathbf{A} \subset$ A, $F_1 : A \rightarrow \mathcal{P}(O)$ defined by:

$F_1(a) = \{o \in O \ / \ I(o,a) = 1 \text{ and } a \in A\}$
$F_1(a) = \emptyset$, otherwise.

4.2 Second soft sets model of FCA

Definition 7. *(Soft set 2) Let be the formal context* $\mathbf{K}$ = (O, A, I) and its associated concept lattice $\mathcal{G}(\mathbf{K})$.
. We define a soft set SS2 as follows:
SS2 = $\{F_2, \mathbf{A}\}$ where $\mathbf{A} \subset$ O x A, $F_2 : \mathbf{A} \rightarrow \mathcal{P}(\mathcal{G}(\mathbf{K}))$.

Remark 3. In this case, the categorization space is all the concept lattice.

1. One categorizes formal concepts, (not objects as in Soft set 1) and the clusters are related to a set of parameters chosen in the set O x A of FCA model.
2. F_2 can be defined in several way depending on the purpose of the categorization. The advantage is that one can choose as parameters, object-attribute pairs, therefore, take as the set of parameters $\mathbf{A}$ a subset of the space (O x A) of the FCA model.

5 Image analysis application

From mathematical point of view, a grey-level digital image is a function I(x,y) defined on Z^2 with values in [0, 255]. The space X is a discrete space. In medical image case, we need to distinguish the header (HD) containing patient informations, from the anatomical object (AO) from the background (BG) and, inside the anatomical object, the disease area (DA). In the following example, we process a medical image with FCA model but taking into account the point of view of soft sets. The image is a grayscale image, of size 512*512. The image is divided into 8*8 non overlapping blocks B1, B1,.....B8. The size of each block is 8*8 pixels. Their position is given in Table 1. The FCA attributes are the following: Entropy, Header(H), Background (BG) and Anatomical Object(AO). The FCA objects are the blocks. One applies Soft set 2 model.

In the digital medical image applications, image characteristics must be splited in two categories: the category of characteristics expressing the position versus the standard partition (background, header, anatomical object, disease area) with values 0,1 and the category of characteristics expressing the values of features related to the intensity function I(x, y).

We split entropy's values into 4 intervals: Entropy 1 = [0, 0.25[; Entropy 2= [0.25, 0.50[; Entropy 3 = [0.50, 0.75[; Entropy 4 = [0.75, 1].
The set of parameters is defined on O x A of FCA model by with O = {B2,B6, B9, B12} and A = {Entropy 4, AO}, so A = {B2, B6, B9, B12} x {Entropy 4, AO}
Function F_2, $F_2 : \mathbf{A} \rightarrow \mathcal{P}(\mathcal{G}(\mathbf{K}))$ is defined by: for a pair p in $\mathbf{A}$, $F_2(p)$ = the subset of formal concepts in the sub-lattice corresponding to a characteristic considered as the most important in the analysis. In our case, it is chosen Entropy 4.

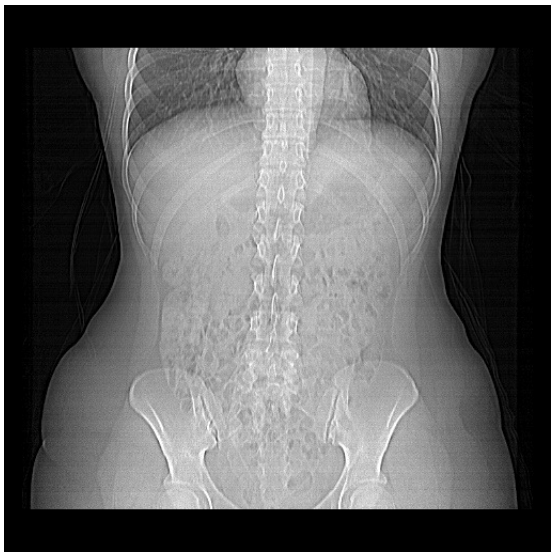


Fig. 1. A medical image

Table 1. Blocks location

B1	B2	B3	B4
B5	B6	B7	B8
B9	B10	B11	B12
B13	B14	B15	B16

Locations are dependent on pixels.

Table 2. Blocks position and blocks entropy

Block number	Block position in columns	Block position in rows	Entropy
B1	0 - 63	0 - 63	0.5239
B2	64 -127	0 - 63	0.8223
B3	128 - 255	0 - 63	0.6223
B4	256 - 511	0 - 63	0.4123
B5	0 - 63	64 -127	0.5326
B6	64 -127	64 -127	0.8774
B7	128 - 255	64 -127	0.5820
B8	256 - 511	64 -127	0.6699
B9	0 - 63	128 - 255	0.7789
B10	64 -127	128 - 255	0.4236
B11	128 - 255	128 - 255	0.6982
B12	255 - 511	128 - 255	0.7899
B13	0 - 63	256 - 511	0.2314
B14	64 -127	256 - 511	0.2369
B15	128 - 255	256 - 511	0.4789
B16	256 - 511	256 - 511	0.5693

	Entropy 1	Entropy 2	Entropy 3	Entropy 4	BG	AO	DA
B1			X		X	X	
B2				X	X	X	
B3			X		X	X	X
B4		X			X	X	
B5			X		X	X	
B6				X		X	
B7			X			X	
B8			X		X	X	
B9					X	X	
B10		X				X	
B11			X			X	
B12				X	X	X	
B13	X				X	X	
B14	X				X	X	
B15		X			X	X	
B16			X		X	X	

Fig. 2. The FCA context of a medical image

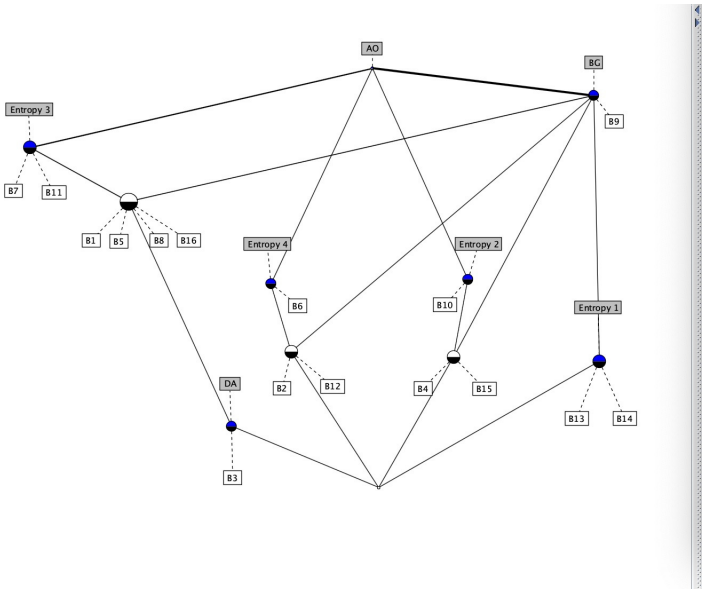


Fig. 3. The FCA lattice of a medical image

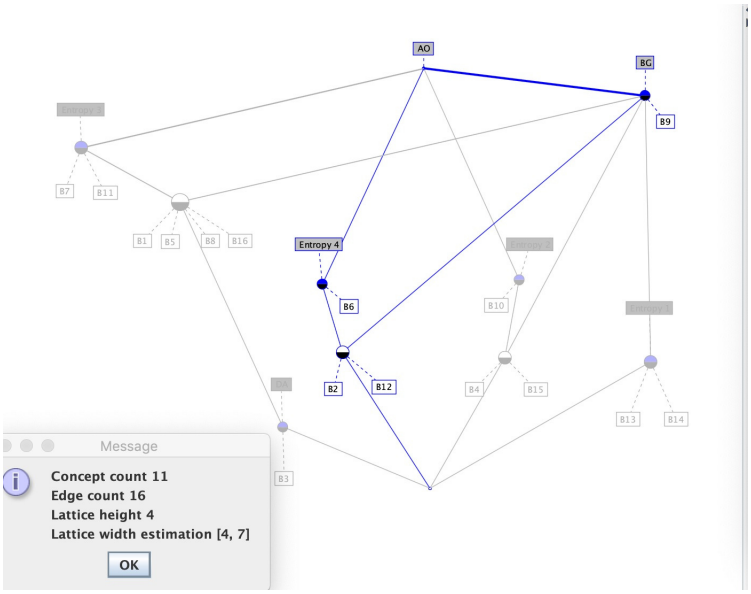


Fig. 4. The sub-lattice of a medical image

6 Conclusions

1. In this paper we did a bridge between FCA model and Soft Set Theory. We have analyzed the the FCA model from the point of view of Soft Sets.
2. Because of the fact that the attributes in FCA model can be considered as parameters in the categorization of objects, we propose two types of models of FCA as soft set: the first one that categorizes objects of the FCA model, the second one that categorizes formal concepts of FCA model.
3. We can find out that the main theorem in FCA proved more than 20 years ago, notably that the concepts lattice is a Galois lattice make possible to view some sub lattice or path of concepts lattice as soft sets.
4. All the examples of FCA are processed with Conexp 1.5.
5. An application of FCA – Soft Set in the domain of medical image was presented. The point of view Soft Set may be useful in a definition of the texture for the medical image.
6. Developing a tool dedicated in particular to process some soft set-type functions can be useful in the library of digital image analysis.

Developing a tool dedicated in particular to process some soft sets F-type functions can be useful in the library of digital image analysis.

References

1. Ganter, B., Stumme, G., Wille, R.: Formal Concept Analysis. Fondation and Application. Springer, Heidelberg (2005).
2. Barbut, M, Monjardet B.: Ordre et Classification (2 tomes), Paris, Hachette Université (1970).
3. Molodtsov,D., A.,: Soft set theory – first results. **Computers and mathematics with applications**, (37),19–31 (1999).
4. Onyeozili, I. A., Gwary T. M.: A Study Of The Fundamentals Of Soft Set Theory.International Journal of Scientific and Technology research **3**(4), 2277-8616 (2014)
5. http://conexp.sourceforge.net

Estimation of Errors Rates for FCA-based Knowledge Discovery*

Dmitry V. Vinogradov[1,2][0000−0001−5761−4706]

[1] FRC Computer Science and Control RAS, Moscow 119333, Russia
vinogradov.d.w@gmail.com
[2] Russian State University for Humanities, Moscow 125993, Russia
http://isdwiki.rsuh.ru

Abstract. In this paper we present an approach for estimating the error rate of bitset representation of object descriptions in FCA-based knowledge discovery. Errors of this kind lead to overfitting phenomenon. The key technique used in our approach is based on the Möbius function on finite partial ordered sets, which was introduced by G.-C. Rota.

Keywords: FCA · JSM-method · Möbius function · error rates.

Introduction

'JSM-method of automatic hypotheses generation' is an approach to Knowledge Discovery that was proposed by V.K. Finn (see, [2], [3]). Initial goal was to formalize 'Inductive Logic' proposed by sir John Stuart Mill in 1848 (at the same time as Boolean Logic was proposed by John Boole) using Boolean Algebra and Many-Valued Logic. This approach has been extended to predict the target property of test examples with the help of generated causes of the property (also called 'hypotheses'). This approach is actually a Machine Learning techniques but high computational complexity (see [7]) is an obstacle to its scalability.

Later JSM-method has been extended to arbitrary (lower semi-)lattices of object descrptions, where similarity operation on object descriptions (wedge operation) satisfies usual idempotent, commutative and associative laws. In [1], [7] FCA-based techniques [4] were applied to JSM-method. FCA provides a very efficient representation for training objects by means of bitsets (fixed length strings of bits) with bit-wise multiplication as similarity operation on them.

The author [12] investigated the 'overfitting' phenomenon for JSM-method by means of so-called 'phantom' or 'accidental' hypotheses. In practice they occur when generated hypotheses are contained in descriptions of examples of the opposite sign (so-called 'counter-examples') that do not possess the target property. JSM-method rejects such hypotheses by means of the 'forbidding counter-example test' (FCET), however the remaining hypotheses can be contained in test examples and erroneously classify them as having the target property, hence causing the 'overfitting'. This phenomenon was experimentally detected

* partially supported by RFBR grant 18-29-03063mk.

by RSUH student L.A. Yakimova within her master project under supervision of the author. Another approach to the overfitting of FCA-based Machine Learning was invented and studied in [8].

The analysis of such situations reveals that FCET can reject some phantom hypotheses if the data contains errors in values of some attributes. This paper considers a possibility to estimate error rates in attribute values taking into account the lattice structures on them. Without loss of generality, we restrict ourselves to the case of single target attribute. The general case is reduced to this one by assuming the independence of errors rates of values of different attributes.

1 Basic definitions and results

1.1 Bitset Encoder Algorithm

A **(finite) context** is a triple (G, M, I) where G and M are finite sets and $I \subseteq G \times M$. The elements of G and M are called **objects** and **attributes**, respectively. As usual, we write gIm instead of $\langle g, m\rangle \in I$ to denote that object g has attribute m.

For $A \subseteq G$ and $B \subseteq M$, define

$$A' = \{m \in M | \forall g \in A(gIm)\}, \tag{1}$$

$$B' = \{g \in G | \forall m \in B(gIm)\}; \tag{2}$$

so A' is the set of attributes common to all the objects in A and B' is the set of objects possesing all the attributes in B. The maps $(\cdot)' : A \mapsto A'$ and $(\cdot)' : B \mapsto B'$ are called **derivation operators** (**polars**) of the context (G, M, I).

A **concept** of the context (G, M, I) is defined to be a pair (A, B), where $A \subseteq G$, $B \subseteq M$, $A' = B$, and $B' = A$. The first component A of the concept (A, B) is called the **extent** of the concept, and the second component B is called its **intent**. The set of all concepts of the context (G, M, I) is denoted by $L(G, M, I)$.

Let (G, M, I) be a context. For concepts (A_1, B_1) and (A_2, B_2) in $L(G, M, I)$ we write $(A_1, B_1) \leq (A_2, B_2)$, if $B_1 \subseteq B_2$. The relation $\leq$ is a **partial order** on $L(G, M, I)$.

Element $x \in L$ of finite lattice $\langle L, \wedge, \vee\rangle$ is called $\vee$-**irreducible**, if $x \neq \emptyset$ and for all $y, z \in L$ $y < x$ and $z < x$ imply $y \vee z < x$. Element $x \in L$ of finite lattice $\langle L, \wedge, \vee\rangle$ is called $\wedge$-**irreducible**, if $x \neq \emptyset$ and for all $y, z \in L$ $x < y$ and $x < z$ imply $x < y \wedge z$.

If subsets of attributes $\{g_i\}' \subset M$ and $\{g_j\}' \subset M$ are intents of objects $g_i \in G$ and $g_j \in G$, respectively, with corresponding bitsets, then the derivation operator on a pair of objects corresponds to bit-wise multiplication, since $\{g_i, g_j\}' = \{g_i\}' \cap \{g_j\}'$. Moreover, polars correspond to iteration of bit-wise multiplication (in arbitrary order) of corresponding bitset-represented objects and attributes, respectively. The last remark is important, since bit-wise multiplication is a basic operation of modern CPU and GPGPU. In terms of JSM-method the binary

operation $\cap : 2^M \times 2^M \to 2^M$ is called 'similarity operation'. This operation defines a low-semilattice on subsets of attributes.

Descriptions of objects can combine discrete and continuous attrtibutes. Here we restrict ourselves to discrete case only, and we will consider the continuous case in another paper. The set of objects' descriptions must be a part of extended set F of 'fragments' supplied with binary 'similarity' operation $\wedge : F \times F \to F$, which is idempotent $x \wedge x = x$, commutative $x \wedge y = y \wedge x$, and associative $x \wedge (y \wedge z) = (x \wedge y) \wedge z$. Moreover, there is the minimal element $\emptyset$ satisfying $x \wedge \emptyset = \emptyset$ (so-called 'trivial fragment'). In FCA this construction is realized by means of *pattern structures* [5]. We convert a fragment into a subset of attributes in such a way that similarity operation becomes set-theoretic intersection (and bit-wise multiplication on corresponding bitsets). We reformulate Basic Theorem 1 of FCA in the following form to construct such an algorithm for encoding values of each attribute, then we form their concatenation for encoding whole object descriptions.

Theorem 1. *[4] For every finite lattice $\langle L, \wedge, \vee \rangle$ let G be a (super)set of all $\wedge$-irreducible elements and M be a (super)set of all $\vee$-irreducible elements. For $gIm \Leftrightarrow g \geq m$ the formal context (G, M, I) generates $L(G, M, I)$, which is isomorphic to the original lattice $\langle L, \wedge, \vee \rangle$.*

In [13] we used this theorem to prove correctness of the following algorithm with respect to the property that similarity operation between values corresponds to the bit-wise multiplication between their codes:

Data: set V of values of current attribute
Result: matrix B with rows as bitset codes of values
$V := topological\_sort(V)$; // topological sorting
$T := order\_matrix$; // transitive closure of cover relation
$\forall i[Del[i] = false]$; // deleted columns
for $(index = 2; index < n; ++index)$ **do**
  for $(indx = 1; indx < index; ++indx)$ **do**
    for $(ndx = 0; ndx < indx; ++ndx)$ **do**
      if $(T[\,][V[index]] == T[\,][V[indx]] \& T[\,][V[ndx]])$ **then**
        $Del[V[index]] := true$;
      end
    end
  end
end
for $(index = 2; index < n; ++index)$ **do**
  for $(indx = 1; indx < index; ++indx)$ **do**
    if $\neg Del[indx]$ **then**
      $\Rightarrow B[indx][index] := T[index][indx]$;
    end
  end
end

Algorithm 1: Encoder Algorithm

For instance, when we consider famous Mushroom Data Set [11] from Machine Learning Repository at University of California in/ Irvine, the following values of 'spore print color' are available: black (k), brown (n), buff (b), chocolate (c), green (r), orange (o), purple (u), white (w), and yellow (y). Let us concentrate on *black, brown, buff, chocolate*, and *yellow* values. The corresponding semi-lattice is shown in Fig. 1.

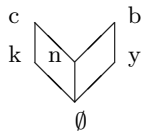


Fig. 1. A fragment of spore print color values semi-lattice

We added $\emptyset$ value to denote the absence of similarity of spore print colors between several training examples that generate some hypotheses.

The order corresponds to "to be more specific/general" relation between values. For example, *buff* is brown-yellow and *chocolate* is black-brown. Hence the similarity between mushrooms with chocolate spore print (c) and ones with buff spore print (b) has brown color (n) of spore print as common.

The algorithm has the following steps:

1. Topological sorting of elements of the semilattice.
2. In the context of order $\geq$ look for columns that coincide with bit-wise multiplication of previous ones (every such column corresponds to $\vee$-reducible element).
3. All found ($\vee$-reducible) columns are removed.
4. Rows of reduced context form bitset representations of the corresponding values.

We sort the values as $k < n < c < y < b$ in correspondence with the partial order on them. Then Theorem 1 gives a big context corresponding to $\geq$.

values	k	n	c	y	b
k	1	0	0	0	0
n	0	1	0	0	0
c	1	1	1	0	0
y	0	0	0	1	0
b	0	1	0	1	1

The column 'c' is equal to the product of columns 'k' and 'n' since $c = k \vee n$. Hence it is reducible. The other $\vee$-reducible element of the lattice is 'b' (again column 'b' is a product of columns 'n' and 'y').

The rows of reduced context

values	k	n	y
k	1	0	0
n	0	1	0
c	1	1	0
y	0	0	1
b	0	1	1

are the bitset representations of the corresponding colors of spore print.

The encoding of the whole description is a concatenation of bit-set presentation of values of its features in some fixed order. Since the bitset representations of different values of same feature have same length, the bit-wise multiplication of the whole representations reduces to the bit-wise multiplications of the corresponding parts, and the similarity between objects is given by the component-wise similarity of their intents.

1.2 Overfitting Phenomenon for JSM-method

We begin with a demonstration of the phenomenon by JSM-method's application to a school problem in geometry. The task is to learn sufficient conditions on a convex polygon to be circled and to predict this property for test examples. Hence there are two target classes: the positive one (with a possible circle around the figure) and the negative one.

The training sample contains regular triangle, rectangular triangle, square, isosceles trapezoid, and diamond (the last figure is negative, the rest contains positive training examples). The test sample contains isosceles triangle, rectangle, and deltoid. We consider the most general case of the corresponding polygon. Hence, for instance, isosceles trapezoid has bases of different sizes and differs from rectangle.

We represent each polygon by a subset of attributes from the following list:

(a) the figure is a triangle;
(b) the figure is a quadrangle;
(c) the figure has a right angle;
(d) the figure has a pair of equal length sides;
(e) all sides of the figure have same length;
(f) the figure has a pair of parallel sides;
(g) the figure has a pair of equal angles;
(h) all angles of the figure are equal.
(i) the sum of the opposite angles of the quadrangle is equal to π.

Hence, the training context $(G^+, M = \{a, b, c, d, e, f, g, h, i\}, I)$ is

training objects	a	b	c	d	e	f	g	h	i
regular triangle	1	0	0	1	1	0	1	1	0
rectangular triangle	1	0	1	0	0	0	0	0	0
square	0	1	1	1	1	1	1	1	1
isosceles trapezoid	0	1	0	1	0	1	1	0	1

This context generates the following concepts with extents of cardinality > 1

$$\langle\{regular\ triangle, rectangular\ triangle\}, \{a\}\rangle,$$
$$\langle\{regular\ triangle, square\}, \{d, e, g, h\}\rangle,$$
$$\langle\{regular\ triangle, square, isosceles\ trapezoid\}, \{d, g\}\rangle,$$
$$\langle\{square, isosceles\ trapezoid\}, \{b, d, f, g, i\}\rangle.$$

The first concept $\langle\{a\}', \{a\}\rangle$ corresponds to the well-known geometric theorem "**Each triangle can be circumscribed by a circle**". The second one expresses the famous fact about regular polygons "**Vertices of regular polygon lie on a circle**". This concept has the form $\langle\{e, h\}', \{e, h\}''\rangle$. The fourth concept represented as $\langle\{i\}', \{i\}''\rangle$ corresponds to the well-known geometric theorem "**Every quadrangle with the sum of opposite angles equal to π can be circumscribed by a circle**". The third concept is a 'phantom' because its extent contains two types of objects: a regular_triangle with the first 'real' cause, and quadrangles with the sum of opposite angles equal to π (square and isosceles trapezoid). Its intent consists of 'accidental common' attributes. Luckily, the forbidden counter-example test (FCET) procedure rejects this concept because of the counter-example

$counter-example$	a	b	c	d	e	f	g	h	i
$diamond$	0	1	0	1	1	1	1	0	0

However, the situation is very subtle, since the 4th concept has two types of objects in its extent. However, it corresponds to the true geometric fact! We think that this concept is a 'real cause' as opposed to 'phantom concept' 3.

Test sample G^τ contains

$test\ objects$	a	b	c	d	e	f	g	h	i
$isosceles\ triangle$	1	0	0	1	0	0	1	0	0
$rectangle$	0	1	1	1	0	1	1	1	1
$deltoid$	0	1	0	1	0	0	1	0	0

JSM-method predicts the first test (isosceles triangle) positively through the 1st concept. The second test object (rectangle) is classified positively by applying the 4th concept. JSM-method might incorrectly predict the target property of the last test case if the 3rd hypothesis is not rejected. This is exactly the phenomenon of overfitting: the hypothesis is consistent with the training sample, but it leads to the incorrect classification of test examples.

A similar situation occurs in real data experiments with the use of JSM-method. For example, consdier an application of JSM-method to the study of toxicity of substituted nitrobenzenes [6]. The data was collected by pharmacologists from the Liverpool University. RSUH student Anastasia S. Oparysheva detected suspicious phenomenon when she analyzed the results of this experiment with respect to overfitting within her undergraduate project [9] under the supervision of the author.

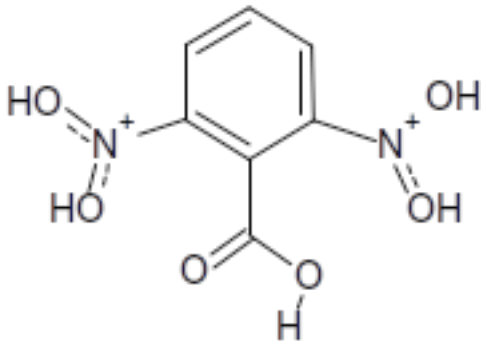


Fig. 2. Concept suspicious to be a phantom

Concept (hypothetical cause of toxicity) 28 with extent consisting 2 elements (training examples 37 and 39) has the form shown in figure 2.

However, example 37 and other 6 training examples generate alternative hypothesis 3. Similarly, example 39 contains alternative concept 41 with 16 elements extent. There is a plausible assertion that hypothesis 28 is a 'phantom' because of an accidental coincidence fragment between two training examples each of which contains some different 'real cause' (example 37 has 'real cause' 3, and example 39 contains 'real cause' 41).

This case was not unique. More than 10 percent of concepts without counter-examples exhibit the same behavior. The aim of her study was to study empirically the overfitting phenomenon, which was previously investigated theoretically by the author in [12]. We present some results from this article below.

An attribute is called **essential**, if it appears in some 'real cause'. Here 'cause' is a set of attributes. Assume for the sake of simplicity that two 'real causes' have no common attributes. Other attributes are called **accompanying**. Hence we partition set M of all attributes into three subsets: the first 'real cause', the second one, and accompanying attributes.

Term 'real cause' corresponds to a generator of intent $\{a\}$ of 1st concept and $\{i\}$ of 4th concept in our initial illustrative example. However in mathematical study below it means just a special subset of attributes other than the accompanying ones. Last attributes form building blocks for 'phantom' concepts. So 'real' in 'real causes' means nothing! It's simply initially introduced term to distinct the group of essential attributes from accompanying ones.

Assume for the sake of simplicity that counter-examples do not contain any essential attribute. Now we introduce probabilistic model to simultaneously generate accompanying attributes for a pair of training examples and m counter-examples.

Denote the number of counter-examples by m, and the number of accompanying attributes by n. It is clear that the accompanying attributes of training objects and counterexamples form a $(2 + m) \times n$ binary matrix. It contains

$N = (2+m) \cdot n$ bits. Accompanying attributes are generated by Bernoulli series of N tests.

Bernoulli series of N tests is the probability distribution on $\{0,1\}^N$ with

$$\mathbf{P}(x_1 = \delta_1, \ldots, x_N = \delta_N) = \prod_{j=1}^{N} p^{\delta_j} \cdot (1-p)^{1-\delta_j},$$

where $0 < p < 1$. The number $p > 0$ is called *success probability* $x_j = 1$ in test j.

Then we set all the attributes of first "real" cause to 1s, all attributes of the second "real" cause to 0s, and add accompanying attributes of first training example to obtain first training example itself. We generate the second training example by setting attributes of the first "real" causes to 0, and of the second one to 1. All counter-examples have 0 in positions corresponding to both "real" causes.

As result we obtain context $2 \times |M|$ and list of m counter-examples. What is the probability to generate concept with 2 element extent without any counter-example from the list?

Theorem 2. *If number n of random attributes tends to infinity, the probability of success equals to $\sqrt{\frac{a}{n}}$, and there are $m = b \cdot \sqrt{n}$ counterexamples, then the probability of an accidental formal concept with 2 elements extent and without any counterexample is $1 - e^{-a} - a \cdot e^{-a} \cdot [1 - e^{-b \cdot \sqrt{a}}]$ at limit.*

Note, that even smaller number $1 - e^{-a} - a \cdot e^{-a}$ is positive, since it coincides with the probability that the Poisson variable Y_a with mean a has value $Y_a > 1$.

Recently Lyudmila A. Yakimova, a former master student of the Russian State University for Humanities made experimental studies [15] on behavior of Machine Learning procedures based on FCA. She also detected the essential overfitting phenomenon. For example, on Mushroom Data Set [11] the JSM method generates several 'phantom' concepts. And as consequence, their use resulted in the wrong classification of toadstools as eatable mushrooms.

Another result of Yakimova's study is a higher rate of 'phantom' concepts than its estimate by the theorem. The reason is in the difference of frequency of appearance of different attributes. Moreover, Yakimova's experiments do not detect overfitting phenomenon for VKF method of Machine Learning based on FCA [14].

2 Error Rates for Values

2.1 Problem Explanation

While checking the condition of forbidding counter-examples, the similarity of some training examples can be contained in description of a counter-example. JSM-method rejects such similarities, however some suspicious hypotheses may be missed if some values of attributes were entered erroneously. Can we estimate the rate of such errors?

Consider again Figure 1. Assume that an expert mistakenly replaces buff spore print color (b) by yellow one (y) for some counter-example. Then the similarity with a mushroom with chocolate spore print has common brown color (n) and can not be included into counter-example, so the procedure saves the hypothesis. If such similarity is phantom it leads to overfitting.

It is clear that value $w \in V$ frequently replaces value $v \in V$ when $w \leq v$. The case of totally fatal mistake is ignored in this study. Denote the rate of such errors $r(v|w)$. The problem is that there does not exist a way to discover v, we see only w as a value entered by an expert. To resolve this difficulty the Möbius function from the incidence algebra is used.

2.2 Möbius functions on finite partial ordered sets

In fundamental work [10] Gian-Carlo Rota introduced the definition of Möbius function on (locally) finite partial ordered sets. It is a working tool for our approach. Below we will recall some key concepts and results of this theory.

Consider the set of real-valued functions of two variables on V with the property $f(x, y) = 0$, if $x \not\leq y$. It has the structure of an associative algebra over the real field if we define the product of such functions $h = f \cdot g$ as

$$h(x, y) = \sum_{z: x \leq z \leq y} f(x, z) \cdot g(z, y). \tag{3}$$

Addition and multiplication by constants are defined in a natural way. This structure is called **incidence algebra** of the given poset. This algebra has the **identity** element $\delta(x, y) = 1$ if $x = y$ and $\delta(x, y) = 0$ otherwise, the **Kronecker delta**.

The **zeta** function $\zeta(x, y)$ is an element of incidence algebra such that $\zeta(x, y) = 1$ if $x \leq y$ and $\zeta(x, y) = 0$ otherwise. It has the inverse element $\mu(x, y)$, **Möbius function**. The proof of the next statement is trivial check.

Proposition 1. *Function defined by induction as* $\mu(x, x) = 1$ *and*

$$\mu(x, y) = -\sum_{z: x \leq z < y} \mu(x, z) \tag{4}$$

is the inverse element to zeta function.

Proposition 2. *Let* $f(x)$ *be a real-valued function, defined on (locally) finite p.o.set* V*. Let an element* $v \in V$ *exists with property that* $f(x) = 0$ *unless* $x \leq v$*. Suppose that*

$$g(x) = \sum_{y: x \leq y} f(y). \tag{5}$$

Then

$$f(u) = \sum_{z: u \leq z} \mu(u, z) g(z). \tag{6}$$

Proof. The function $g(x)$ is well-defined since it equals to $\sum_{y:x\leq y\leq v} f(y)$, which is finite for a locally finite poset.

Substituting the right side of 5 into the right side of 6 and simplifying, we get

$$\sum_{z:u\leq z} \mu(u,z)g(z) = \sum_{z:u\leq z} \sum_{y:z\leq y} \mu(u,z)f(y) = \sum_{z:u\leq z} \sum_{y} \mu(u,z)\zeta(z,y)f(y).$$

Interchanging the order of summation, this becomes

$$\sum_{y} f(y) \sum_{z:u\leq z} \mu(u,z)\zeta(z,y) = \sum_{y} f(y)\delta(u,y) = f(u).$$

2.3 Algorithm

We can collect statistics for mistakenly missed phantom hypotheses h (with help of negative examples from tests sample). Let hypothesis h be included into some negative example o (either from the training or test sample) when we omit the values of the attribute under study. Such inclusions are called **pruned**.

Let us fix value x of the attribute under study. We compute the fraction $q_w(x)$ of pruned inclusions of hypotheses with value x into counter-examples with value w with respect to total of all pruned inclusions of hypotheses with value x into negative examples (either from training or tests samples).

Then we have

$$g_w(x) = \sum_{v:x\leq v} (\zeta(w,v) - \delta(w,v))r(v \mid w). \tag{7}$$

Here $\zeta(w,v)-\delta(w,v)$ determines the condition $w < v$ since there exists erroneous replacement invisible v by observable $w < v$. Summation holds because rates of different replaces are additive.

At first, we use Proposition 1 to compute Möbius functions for every attribute values lattice.

Then we compute statistics $g_w(x)$ by application of pruning inclusions.

Finally, we apply Proposition 2 to estimate errors rates as

$$r(v|w) = \sum_{z:v\leq z} \mu(v,z)q_w(z). \tag{8}$$

The omitted factor $\zeta(w,v) - \delta(w,v)$ means $w < v$.

Conclusion

We applied Möbius functions on finite posets to estimate rates of mistakenly replaces of attribute value by a smaller one that leads to overfitting in JSM-method. Experiments with Mushroom Dataset [11] demonstrate a very small (less than 0.01) rate of erroneous replacements 'buff' color of spore print by 'yellow' one.

Acknowledgments

Author would like to thank his colleagues from Dorodnicyn Computing Center of Federal Research Center "Computer Science and Control" of Russian Academy of Science and Russian State University for Humanities for long-term support and useful discussions. Author is grateful to anonymous reviewers, whose comments helped significantly improve the presentation.

References

1. Blinova, V.G. et al.: Toxicology Analysis by Means of the JSM-method. *Bioinform.* **19**(10): 1201–1207 (2003)
2. Finn, V.K.: J.S. Mill's inductive methods in artificial intelligence systems I. *Sci. Tech. Inform. Proc.* **38**, 385–402 (2011)
3. Finn, V.K.: J.S. Mill's inductive methods in artificial intelligence systems II. *Sci. Tech. Inform. Proc.* **39**, 241–261 (2012)
4. Ganter, B., Wille, R.: Formal Concept Analysis. Springer, Berlin (1999)
5. Ganter, B., Kuznetsov, S.O.: In: Stumme G. and Delugach H. (eds.) *Proc. 9th International Conference on Conceptual Structures (ICCS 2001)*, Lecture Notes in Artificial Intelligence, **2120**, 129–142 (2001)
6. Kharchevnikova, N.V. et al.: A JSM intelligent system for toxicity. Analysis of functional cumulation of chemical compounds. *Autom. Doc. Math. Linguist.* **51**, 20–26 (2017)
7. Kuznetsov, S.O.: Complexity of Learning in Concept Lattices from Positive and Negative Examples. *Discrete Applied Mathematics*, no. 142(1-3). – 111–125 (2004)
8. Makhalova, T., Kuznetsov, S.O.: On Overfitting of Classifiers Making a Lattice. In: Bertet K., Borchmann D., Cellier P., Ferré S. (eds) *Formal Concept Analysis. ICFCA 2017.* Lecture Notes in Computer Science, **10308**, 184–197 (2017)
9. Oparysheva, A.S.: Search for Accidental Hypotheses in Real Data. *Baccalaureate work (adv.: Vinogradov, D.V.)* Russian State University for Humanities, Moscow (2018) (in Russian)
10. Rota, G.C.: On the Foundations of Combinatorial Theory I. Theory of Möbius Functions. *Z. Wahrscheinlichkeitstheorie verw Gebiete* **2**, 340–368 (1964)
11. UCI Machine Learning Repository: Mushroom Data Set, https://archive.ics.uci.edu/ml/datasets/Mushroom. Last accessed 10 May 2020
12. Vinogradov, D.V.: Accidental formal concepts in the presence of counterexamples. In: *Proceedings of International Workshop on Formal Concept Analysis for Knowledge Discovery (FCA4KD 2017)*: CEUR Workshop Proceedings. **1921**, 104–112 (2017)
13. Vinogradov, D.V.: On Object Representation by Bit Strings for the VKF-Method. *Autom. Doc. Math. Linguist.* **52**, 113–116 (2018)
14. Vinogradov, D.V.: Continuous Attributes for FCA-based Machine Learning. *8th Workshop "What can FCA do for Artificial Intelligence?"* (2020)
15. Yakimova, L.A.: Experimental studies on behavior of solvers based in binary similarity operation. *Master's Degree Thesis (adv.: Vinogradov, D.V.)* Russian State University for Humanities, Moscow (2020) (in Russian)

Building a Representation Context Based on Attribute Exploration Algorithms

Jaume Baixeries[1], Victor Codocedo[2], Mehdi Kaytoue[3], and Amedeo Napoli[4]

[1] Computer Science Department. Universitat Politècnica de Catalunya. 08032, Barcelona. Catalonia.
[2] Departamento de Informática. Universidad Técnica Federico Santa María. Campus San Joaquín, Santiago de Chile.
[3] Infologic R&D, 99 avenue de Lyon, 26500 Bourg-Lès-Valence, France.
[4] Université de Lorraine, CNRS, Inria, LORIA, 54000 Nancy, France.

Corresponding author: victor.codocedo@inf.utfsm.cl

Abstract. In this paper we discuss the problem of constructing the representation context of a pattern structure. First, we present a naive algorithm that computes the representation context of a pattern structure using an algorithm which is a variation of attribute exploration. Then, we study a different sampling technique for reducing the size of the representation context. Finally we show how to build such a reduced context and we discuss the possible ways of doing it.

1 Introduction

Formal Concept Analysis (FCA) has dealt with non binary data mainly in two different ways: one is by *scaling* [9] the original dataset into a formal context. This method has some drawbacks as it generates a formal context with larger dimensions than the original dataset and, depending on the method used to scale, some information may also be lost. Another way to deal with non binary data is by generalizing FCA into "pattern structures" [10,15], which allows handling complex data directly. Thus, instead of a analysing a binary relation between some objects and their attributes, it analyses a relation between objects and their representations, the values of which form a meet-semilattice. Pattern structures have proved useful for different unrelated tasks, such as, for instance to analyze gene expressions [14,13], compute database dependencies [1,2] or compute biclusters [6], or other structures data, like trees, intervals, graphs, sequences, fuzzy and heterogeneous data, among others [12].

Any pattern structure can be represented by an equivalent formal context, which is consequently called a "representation context" [10,4]. By *equivalent* we mean that there is a bijection between the intents in the representation context and the pattern-intents in the pattern structure. In fact, this means that both contain the same set of extents.

Pattern structures are difficult to calculate precisely because of the complex nature of the object representations they model. Often, pattern concept lattices

are very large and their associated sets of pattern concepts are difficult to handle. In this article we tackle this problem by mining a reduced "representation context" of pattern structures by means of *irreducible patterns* (IPs) [7,11]. IPs act as a basis –in the linear-algebraic sense– of the space of patterns, i.e. each pattern can be represented as a linear combination of the *join* of a subset of IPs. In the standard FCA notation, IPs correspond to the intents of $\bigwedge$-irreducible concepts. More specifically, we propose an algorithm that calculates the extents of a pattern structure while simultaneously calculating a small representation context. The later allows for a cheap calculation of most pattern extents and it is incrementally completed with *samples* obtained from the actual pattern structure. The algorithm is based on attribute exploration [8] (*object exploration* in this case) that instead of a domain-expert, uses an oracle to validate the question "is set X an extent in the pattern structure?". In the negative case, the oracle should be able to sample a new attribute to add in the representation context which is heuristically selected to be an IP (or close to it).

We first present a naive algorithm based on NextClosure [8] and attribute exploration techniques. Then, we present an alternative way to "call the oracle" with a sampling strategy which strongly depends on the nature of the object descriptions. We show that the sampling strategy is able to obtain very small representation contexts, and as well, it sometimes generates with the highest probability the representation contexts with irreducible attributes only.

This work mainly relies on studies about the formalization of representation contexts such as [10] and [4], and on the book [8]. Moreover, a short version of this paper was published in the proceedings of the ICFCA 2019 Conference [5].

2 Notation and Background

In the following we introduce some definitions needed for the development of this article. The notations used are based on [9]. A formal context $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ is a triple where $\mathtt{G}$ is a set of objects, $\mathtt{M}$ is a set of attributes and $\mathtt{I} \subseteq \mathtt{G} \times \mathtt{M}$ is an incidence relation where $(g, m) \in \mathtt{I}$ denotes that "*object g has the attribute m*". The derivation operators are denoted as $' : \wp(\mathtt{G}) \to \wp(\mathtt{M})$ and $' : \wp(\mathtt{M}) \to \wp(\mathtt{G})$.

A many-valued context $\mathcal{M} = (\mathtt{G}, \mathtt{N}, \mathtt{W}, \mathtt{J})$ is a data table where, in addition to $\mathtt{G}$ and $\mathtt{M}$, we define a set of values $\mathtt{W}^m$ for each attribute $m \in \mathtt{M}$ (where $\mathtt{W} = \bigcup \mathtt{W}^m$ for all $m \in \mathtt{M}$) such that $m(g) = w$ denotes that *"the value of the attribute m for the object g is w"* (with $w \in \mathtt{W}^m$). Additionally, in this document we will consider each $\mathtt{W}^m$ as an ordered set where $\mathtt{W}^m_i$ denotes the i-th element in the set. An example of a many-valued context can be found in Table 1 where $a(r_1) = 1$, $\mathtt{W}^b = \{1, 2\}$ and $\mathtt{W}^b_2 = 2$.

The pattern structure framework is a generalization of FCA [10] where objects may have *complex* descriptions. A pattern structure is a triple $(\mathtt{G}, (\mathtt{D}, \sqcap), \delta)$ where $\mathtt{G}$ is a set of objects, $(\mathtt{D}, \sqcap)$ is a semi-lattice of complex object descriptions, and δ is a function that assigns to each object in $\mathtt{G}$ a description in $\mathtt{D}$. The derivation operators for a pattern structure are denoted as $(\cdot)^{\square} : \mathtt{G} \to \mathtt{D}$ and

$(\cdot)^{\diamond} : \mathtt{D} \to \mathtt{G}$.

$$A \subseteq \mathtt{G}\ ;\ A^{\square} = \bigsqcap_{g \in A} \delta(g)$$
$$d \in \mathtt{D}\ ;\ d^{\diamond} = \{g \in \mathtt{G} \mid d \sqsubseteq \delta(g)\}$$

Pattern structures come in different flavors depending on the nature of the representation for which they are intended. Regardless of the nature of data, any pattern structure can be represented with a formal context, as the next definition shows:

Definition 1 ([10]). *Let* $(\mathtt{G}, (\mathtt{D}, \sqcap), \delta)$ *be a pattern structure, and let* $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ *be a context such that* $\mathtt{M} \subseteq \mathtt{D}$ *and* $(g, m) \in \mathtt{I} \iff m \sqsubseteq \delta(g)$.

If every intent of $(\mathtt{G}, (\mathtt{D}, \sqcap), \delta)$ *is of the form*

$$\bigsqcup X = \bigsqcap \{d \in \mathtt{D} \mid (\forall x \in X) x \sqsubseteq d\}$$

for some $X \subseteq \mathtt{M}$ *(i.e.* $\mathtt{M}$ *is* $\bigsqcup$*-dense in* $(\mathtt{D}, \sqcap)$*), then* $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ *is a "representation context" of* $(\mathtt{G}, (\mathtt{D}, \sqcap), \delta)$

The condition that $\mathtt{M}$ is $\bigsqcup$-dense in $(\mathtt{D}, \sqcap)$ means that any element in $\mathtt{D}$ can be represented (uniquely) as the *meet* of the filters of a set of descriptions $X \subseteq \mathtt{M}$ in $(\mathtt{D}, \sqcap)$.

Representation contexts yield concept lattices that are isomorphic to the pattern concept lattices of their corresponding pattern structures. Moreover, there is a bijection between the intents in the representation context and the pattern-intents in the pattern structure. For any pattern-intent $d \in \mathtt{D}$ in the pattern structure we have an intent $X \subseteq \mathtt{M}$ in the representation context such that $d = \bigsqcup X$ and $X = {\downarrow} d \cap \mathtt{M}$ where $\downarrow d$ is the ideal of d in $(\mathtt{D}, \sqcap)$.

3 Computing a Representation Context: a First and Naive Approach

3.1 The NaiveRepContext Algorithm

We first present a simple and naive method for building a representation context from a given pattern structure. Algorithm 1 shows this method in the form of a procedure called NaiveRepContext. This procedure is based on the standard NextClosure algorithm [8], to which some changes –marked with an asterisk– have been performed. These changes represent the interaction of the algorithm with the oracle.

The algorithm receives as inputs a copy of a many-valued context $\mathcal{M} = (\mathtt{G}, \mathtt{N}, \mathtt{W}, \mathtt{J})$ and implementations for the derivation operators $(\cdot)^{\square}$ and $(\cdot)^{\diamond}$ corresponding to the pattern structure $(\mathtt{G}, (\mathtt{D}, \sqcap), \delta)$ defined over $\mathcal{M}$. To distinguish the attributes in the many-valued context from those in the representation context

created by Algorithm 1, we will refer to $\mathtt{N}$ as a set of *columns* in the many-valued context.

The algorithm starts by building the representation context $\mathcal{K}$. The set of objects is the same set of objects in the pattern structure, while the set of attributes and the incidence relation are initially empty. Line 12 checks whether a set of objects (B'') in the representation context is an extent in the pattern structure ($B^{\square\diamond}$). If this is the case, the algorithm continues executing NextClosure. Otherwise, the algorithm adds to the representation context a new attribute corresponding to the pattern associated to the mismatching closure. It also adds to the incidence relation the pairs *object-attribute*, i.e. $(h, B^{\square})$ as defined in line 14. Line 24 outputs the calculated representation context.

Algorithm 1 A Naive Calculation of the Representation Context.

```
1: procedure NaiveRepContext(M, (·)□, (·)◇)                    ▷ M = (G, N, W, J)
2:     M ← ∅
3:     I ← ∅
4:     K ← (G, M, I)
5:     A ← ∅
6:     while A ≠ G do
7:         for g ∈ G in reverse order do
8:             if g ∈ A then
9:                 A ← A \ {g}
10:            else
11:                B ← A ∪ {g}
12:                if B'' ≠ B^□◇ then                           ▷ (*)
13:                    M ← M ∪ {B^□}                            ▷ (*)
14:                    I ← I ∪ {(h, B^□) | h ∈ B^□◇}            ▷ (*)
15:                end if
16:                if B''\A contains no element < g then
17:                    A ← B''
18:                    Exit For
19:                end if
20:            end if
21:        end for
22:        Output A
23:    end while
24:    return K
25: end procedure
```

Proposition 1.
Algorithm 1 computes a representation context $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ *of* $(\mathtt{G}, (\mathtt{D}, \sqcap), \delta)$.

Proof. We show that $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ meets the conditions in Definition 1. Similarly to NextClosure, Algorithm 1 enumerates all closures –given an arbitrary closure operator– in lectic order. However, Algorithm 1 uses two different closure operators, namely the standard closure operator of FCA defined over the representation context under construction $(\cdot)''$, and the one defined by the two derivation operators in the pattern structure, i.e. $(\cdot)^{\square}$ and $(\cdot)^{\diamond}$. Both closure operators are made to coincide by the new instructions in the algorithm. When this is not the case (i.e. $B'' \neq B^{\square\diamond}$ for a given $B \subseteq \mathtt{G}$), a new attribute is added to the representation context in the shape of $B^{\square}$. Additionally, the pair $(h, B^{\square})$ is added to

the incidence relation of the representation context for all objects $h \in B$. This in turn ensures that $B'' = B^{\Box\diamond}$ holds in the modified representation context.

A consequence of the equality $B'' = B^{\Box\diamond}$ is that the set of extents in the representation context is the same as the set of extents in the pattern structure. This also means that there is a one-to-one correspondence between the intents in the representation context and the patterns in the pattern structure, i.e. for any extent $B'' = B^{\Box\diamond}$, the intent $B''' = B'$ corresponds to the pattern $B^{\Box\diamond\Box} = B^{\Box}$ ($B''' = B'$ and $B^{\Box\diamond\Box} = B^{\Box}$ are properties of the derivation operators [9]). Thus, we have that any element in D, which can be represented as $B^{\Box}$ for an arbitrary $B \subseteq \mathtt{G}$, is of the form $\bigsqcup B'$ or $\bigsqcap\{d \in \mathtt{D} \mid (\forall m \in B')m \sqsubseteq d\}$. Consequently, $\mathtt{M}$ is $\bigsqcup$-dense in $(\mathtt{D}, \sqcap)$ and $(\mathtt{G}, \mathtt{M}, \mathtt{I})$ is an RC of the pattern structure $(\mathtt{G}, (\mathtt{D}, \sqcap), \delta)$. □

3.2 An Example of Execution

Table 3 shows the execution trace of NaiveRepContext over an interval pattern structure defined over the many-valued formal context on Table 1. Columns in Table 3 are: row number, a candidate set B in Algorithm 1, its closure in the representation context, its closure in the pattern structure, the result of the test in line 12 of Algorithm 1, and the pattern-intent $B^{\Box}$. Notice that there are 26 entries in the last column of the table that correspond to the 26 different attributes in the resulting representation context in Table 4. The latter are enumerated in the order they appear in Table 3.

Examining the first row in Table 3, we observe that the algorithm initially calculates the closure of set $\{r_7\}$. At this point the representation context is empty so the closure of $\{r_7\}$ corresponds exactly to $\mathtt{G}$. However, using the pattern structure we find out that $\{r_7\}^{\Box\diamond} = \{r_7\}$ (test $B'' = B^{\Box\diamond}$ fails) and we need to add something to the representation context. Algorithms 1 adds to $\mathtt{M}$ the attribute corresponding to $\{r_7\}^{\Box}$ which is labelled as attribute d_1 in the representation context of Table 4 (thus ensuring that $\{r_7\}'' = \{r_7\}$)). The procedure is the same for the following sets in the lectic enumeration until we calculate the closure of $\{r_3\}$.

Two important things occur at this point: Firstly, $\{r_3\}^{\Box\diamond} = \{r_3\}$. Secondly, there is enough information in the representation context so that $\{r_3\}'' = \{r_3\}$ as well. Consequently, no new attribute is added to the representation context.

Let us discuss this example in more depth. The representation context at this point is conformed by the same elements in Table 4 truncated at column d_{10}. At this point, we should observe that $\{r_3\}^{\Box} = \bigsqcup\{r_3\}' = \bigsqcup\{d_6, d_8, d_9, d_{10}\}$.

Another important observation is that we do not really need to verify in the pattern structure that $\{r_3\}^{\Box\diamond} = \{r_3\}''$. Because closure is an extensive operation ($B \subseteq B''$ and $B \subseteq B^{\Box\diamond}$) at any point in the execution of Algorithm 1 we have that $B \subseteq B^{\Box\diamond} \subseteq B''$. Thus, $B = B'' \implies B = B^{\Box\diamond}$. This is indeed quite important since, as shown in Table 3, usually $B'' = B^{\Box\diamond}$ is true more often than not (in this example, 38 times out of 64). By avoiding the calculation of $B^{\Box\diamond}$ within the pattern structure we can avoid the costly calculation of pattern similarities and subsumptions by means of the representation context.

Table 1: Example Dataset 1

	a	b	c	d	e
r_1	1	1	1	1	1
r_2	2	1	1	1	1
r_3	3	1	2	2	2
r_4	3	2	3	2	2
r_5	3	1	2	3	2
r_6	1	1	2	2	2
r_7	1	1	2	4	2

Table 2: Example Dataset 2

	a	b	c	d
r_1	1	4	3	2
r_2	2	1	4	3
r_3	3	2	1	4
r_4	4	3	2	1

4 Computing Smaller Representation Contexts

4.1 Extending the NaiveRepContext Algorithm with Sampling

Algorithm 1 is able to calculate the representation context of the interval pattern structure created for the many-valued context in Table 1 rather inefficiently since it creates a different attribute for almost every interval pattern in the pattern structure. As previously described, Algorithm 1 is able to avoid creating attributes for some interval patterns when there is enough information in the partial representation context. More formally, it does not include a new attribute d in the representation context if and only if there exists a set $Y \subseteq \mathtt{M}$ such that $d = \bigsqcup Y$ or $d = \bigcap_{m \in Y} m^{\diamond}$. Knowing this, we are interested in examining whether there is a way to calculate a smaller representation context given an interval pattern structure definition.

Table 6 shows the execution trace of Algorithm 1 over the interval pattern structure defined over Table 2. We can observe that the algorithm generates 14 different attributes in the representation context, one for each interval pattern concept except for the top and bottom concepts. Notice that this pattern structure contains $2^4 = 16$ interval pattern concepts and since it has 4 objects, we can conclude that its associated concept lattice is Boolean.

This example is interesting because it is a worst-case scenario for Algorithm 1. In fact, it generated the largest possible representation context for the interval pattern structure derived from Table 2. Moreover, it verified each extent closure in the pattern structure. This example is even more interesting considering that for such an interval pattern structure there is a known *smallest* representation context which corresponds to a *contranominal scale* [8] that for this example would contain only 4 attributes as shown in Table 5.

To make matters worse, we can show that Algorithm 1 would behave the same for an interval pattern structure with an associated Boolean concept lattice of any size. Actually, this is a consequence of the lectic enumeration of object sets performed by Algorithm 1 which implies that whenever $B_0 \subseteq B_1$ (with B_0, B_1 closed sets in $\mathtt{G}$) then, B_0 is enumerated before B_1. Since a pattern-intent $B_1^{\square}$ is not included in the representation context only when there exists a set $Y \subseteq \mathtt{M}$ such that $\bigcap_{m \in Y} m^{\diamond} = B_1$ we have that $(\forall m \in Y) B_1 \subseteq m^{\diamond}$. Consequently, $B_1^{\square}$ would not be included in the representation context only when all $m \in X$ had been already enumerated which cannot be the case because of lectic numeration.

Table 3: Execution Trace of NaiveRepContext over Table 1

	B	B''	$B^{\Box\diamond}$	$B'' = B^{\Box\diamond}$	$B^{\Box}$
1	r_7	G	r_7	$False$	$\langle[1,1],[1,1],[2,2],[4,4],[2,2]\rangle$
2	r_6	G	r_6	$False$	$\langle[1,1],[1,1],[2,2],[2,2],[2,2]\rangle$
3	r_6, r_7	G	r_6, r_7	$False$	$\langle[1,1],[1,1],[2,2],[2,4],[2,2]\rangle$
4	r_5	G	r_5	$False$	$\langle[3,3],[1,1],[2,2],[3,3],[2,2]\rangle$
5	r_5, r_7	G	r_5, r_7	$False$	$\langle[1,3],[1,1],[2,2],[3,4],[2,2]\rangle$
6	r_5, r_6	G	r_3, r_5, r_6	$False$	$\langle[1,3],[1,1],[2,2],[2,3],[2,2]\rangle$
7	r_4	G	r_4	$False$	$\langle[3,3],[2,2],[3,3],[2,2],[2,2]\rangle$
8	r_4, r_7	G	r_3, r_4, r_5, r_6, r_7	$False$	$\langle[1,3],[1,2],[2,3],[2,4],[2,2]\rangle$
9	r_4, r_6	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_6	$False$	$\langle[1,3],[1,2],[2,3],[2,2],[2,2]\rangle$
10	r_4, r_5	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_5	$False$	$\langle[3,3],[1,2],[2,3],[2,3],[2,2]\rangle$
11	r_3	r_3	r_3	$True$	-
12	r_3, r_7	r_3, r_4, r_5, r_6, r_7	r_3, r_5, r_6, r_7	$False$	$\langle[1,3],[1,1],[2,2],[2,4],[2,2]\rangle$
13	r_3, r_6	r_3, r_6	r_3, r_6	$True$	-
14	r_3, r_6, r_7	r_3, r_5, r_6, r_7	r_3, r_5, r_6, r_7	$True$	-
15	r_3, r_5	r_3, r_5	r_3, r_5	$True$	-
16	r_3, r_5, r_7	r_3, r_5, r_6, r_7	r_3, r_5, r_6, r_7	$True$	-
17	r_3, r_5, r_6	r_3, r_5, r_6	r_3, r_5, r_6	$True$	-
18	r_3, r_5, r_6, r_7	r_3, r_5, r_6, r_7	r_3, r_5, r_6, r_7	$True$	-
19	r_3, r_4	r_3, r_4	r_3, r_4	$True$	-
20	r_3, r_4, r_7	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_5, r_6, r_7	$True$	-
21	r_3, r_4, r_6	r_3, r_4, r_6	r_3, r_4, r_6	$True$	-
22	r_3, r_4, r_6, r_7	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_5, r_6, r_7	$True$	-
23	r_3, r_4, r_5	r_3, r_4, r_5	r_3, r_4, r_5	$True$	-
24	r_3, r_4, r_5, r_7	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_5, r_6, r_7	$True$	-
25	r_3, r_4, r_5, r_6	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_5, r_6	$False$	$\langle[1,3],[1,2],[2,3],[2,3],[2,2]\rangle$
26	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_5, r_6, r_7	r_3, r_4, r_5, r_6, r_7	$True$	-
27	r_2	G	r_2	$False$	$\langle[2,2],[1,1],[1,1],[1,1],[1,1]\rangle$
28	r_2, r_7	G	r_1, r_2, r_6, r_7	$False$	$\langle[1,2],[1,1],[1,2],[1,4],[1,2]\rangle$
29	r_2, r_6	r_1, r_2, r_6, r_7	r_1, r_2, r_6	$False$	$\langle[1,2],[1,1],[1,2],[1,2],[1,2]\rangle$
30	r_2, r_5	G	r_2, r_3, r_5	$False$	$\langle[2,3],[1,1],[1,2],[1,3],[1,2]\rangle$
31	r_2, r_4	G	r_2, r_3, r_4	$False$	$\langle[2,3],[1,2],[1,3],[1,2],[1,2]\rangle$
32	r_2, r_3	r_2, r_3	r_2, r_3	$True$	-
33	r_2, r_3, r_7	G	$r_1, r_2, r_3, r_5, r_6, r_7$	$False$	$\langle[1,3],[1,1],[1,2],[1,4],[1,2]\rangle$
34	r_2, r_3, r_6	$r_1, r_2, r_3, r_5, r_6, r_7$	r_1, r_2, r_3, r_6	$False$	$\langle[1,3],[1,1],[1,2],[1,2],[1,2]\rangle$
35	r_2, r_3, r_5	r_2, r_3, r_5	r_2, r_3, r_5	$True$	-
36	r_2, r_3, r_5, r_7	$r_1, r_2, r_3, r_5, r_6, r_7$	$r_1, r_2, r_3, r_5, r_6, r_7$	$True$	-
37	r_2, r_3, r_5, r_6	$r_1, r_2, r_3, r_5, r_6, r_7$	r_1, r_2, r_3, r_5, r_6	$False$	$\langle[1,3],[1,1],[1,2],[1,3],[1,2]\rangle$
38	r_2, r_3, r_4	r_2, r_3, r_4	r_2, r_3, r_4	$True$	-
39	r_2, r_3, r_4, r_7	G	G	$True$	-
40	r_2, r_3, r_4, r_6	G	r_1, r_2, r_3, r_4, r_6	$False$	$\langle[1,3],[1,2],[1,3],[1,2],[1,2]\rangle$
41	r_2, r_3, r_4, r_5	G	r_2, r_3, r_4, r_5	$False$	$\langle[2,3],[1,2],[1,3],[1,3],[1,2]\rangle$
42	r_2, r_3, r_4, r_5, r_7	G	G	$True$	-
43	r_2, r_3, r_4, r_5, r_6	G	$r_1, r_2, r_3, r_4, r_5, r_6$	$False$	$\langle[1,3],[1,2],[1,3],[1,3],[1,2]\rangle$
44	r_1	r_1, r_2, r_6	r_1	$False$	$\langle[1,1],[1,1],[1,1],[1,1],[1,1]\rangle$
45	r_1, r_7	r_1, r_2, r_6, r_7	r_1, r_6, r_7	$False$	$\langle[1,1],[1,1],[1,2],[1,4],[1,2]\rangle$
46	r_1, r_6	r_1, r_6	r_1, r_6	$True$	-
47	r_1, r_6, r_7	r_1, r_6, r_7	r_1, r_6, r_7	$True$	-
48	r_1, r_5	r_1, r_2, r_3, r_5, r_6	r_1, r_2, r_3, r_5, r_6	$True$	-
49	r_1, r_4	r_1, r_2, r_3, r_4, r_6	r_1, r_2, r_3, r_4, r_6	$True$	-
50	r_1, r_3	r_1, r_2, r_3, r_6	r_1, r_2, r_3, r_6	$True$	-
51	r_1, r_2	r_1, r_2, r_6	r_1, r_2	$False$	$\langle[1,2],[1,1],[1,1],[1,1],[1,1]\rangle$
52	r_1, r_2, r_7	r_1, r_2, r_6, r_7	r_1, r_2, r_6, r_7	$True$	-
53	r_1, r_2, r_6	r_1, r_2, r_6	r_1, r_2, r_6	$True$	-
54	r_1, r_2, r_6, r_7	r_1, r_2, r_6, r_7	r_1, r_2, r_6, r_7	$True$	-
55	r_1, r_2, r_5	r_1, r_2, r_3, r_5, r_6	r_1, r_2, r_3, r_5, r_6	$True$	-
56	r_1, r_2, r_4	r_1, r_2, r_3, r_4, r_6	r_1, r_2, r_3, r_4, r_6	$True$	-
57	r_1, r_2, r_3	r_1, r_2, r_3, r_6	r_1, r_2, r_3, r_6	$True$	-
58	r_1, r_2, r_3, r_6, r_7	$r_1, r_2, r_3, r_5, r_6, r_7$	$r_1, r_2, r_3, r_5, r_6, r_7$	$True$	-
59	r_1, r_2, r_3, r_5	r_1, r_2, r_3, r_5, r_6	r_1, r_2, r_3, r_5, r_6	$True$	-
60	$r_1, r_2, r_3, r_5, r_6, r_7$	$r_1, r_2, r_3, r_5, r_6, r_7$	$r_1, r_2, r_3, r_5, r_6, r_7$	$True$	-
61	r_1, r_2, r_3, r_4	r_1, r_2, r_3, r_4, r_6	r_1, r_2, r_3, r_4, r_6	$True$	-
62	$r_1, r_2, r_3, r_4, r_6, r_7$	G	G	$True$	-
63	r_1, r_2, r_3, r_4, r_5	$r_1, r_2, r_3, r_4, r_5, r_6$	$r_1, r_2, r_3, r_4, r_5, r_6$	$True$	-
64	G	G	G	$True$	-

Table 4: Representation Context of the Interval Pattern Structure of Table 1

	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}	d_{17}	d_{18}	d_{19}	d_{20}	d_{21}	d_{22}	d_{23}	d_{24}	d_{25}	d_{26}
r_1														×	×			×	×	×	×		×	×	×	×
r_2													×	×	×	×	×	×	×	×	×	×	×			×
r_3						×		×	×	×	×	×				×	×	×	×	×	×	×	×			
r_4							×	×	×	×		×					×				×	×	×			
r_5				×	×	×		×		×	×	×				×		×		×		×	×			
r_6		×	×			×		×	×		×	×		×	×			×	×	×	×		×		×	
r_7	×		×		×			×			×			×				×							×	

Table 5: Contranominal scale generated by Algorithm 3

	X_1	X_2	X_3	X_4
r_1		×	×	×
r_2	×		×	×
r_3	×	×		×
r_4	×	×	×	

Given a pattern structure with an associated Boolean concept lattice we would like that ideally, Algorithm 1 only adds irreducible attributes to the representation context. In this way we would maintain the $\bigsqcup$-dense in $(\mathtt{D}, \sqcap)$ property with a minimum number of attributes. Table 6 shows in gray those rows that represent the relevant attributes to add to the representation context. Adding only these four attributes would be enough to represent the set of interval pattern concepts. Nevertheless, algorithms that compute the irreducible attributes of a closure system have exponential complexity in the size of its formal context[3]. Instead, we resort to a sampling-based strategy to retrieve attributes with a large image in $\mathtt{G}$.

Table 6: Execution Trace of NaiveRepContext over Table 2

	B	B''	$B^{\square\diamond}$	$B'' = B^{\square\diamond}$	$B^{\square}$
1	r_4	r_1, r_2, r_3, r_4	r_4	$False$	$\langle[4,4],[3,3],[2,2],[1,1]\rangle$
2	r_3	r_1, r_2, r_3, r_4	r_3	$False$	$\langle[3,3],[2,2],[1,1],[4,4]\rangle$
3	r_3, r_4	r_1, r_2, r_3, r_4	r_3, r_4	$False$	$\langle[3,4],[2,3],[1,2],[1,4]\rangle$
4	r_2	r_1, r_2, r_3, r_4	r_2	$False$	$\langle[2,2],[1,1],[4,4],[3,3]\rangle$
5	r_2, r_4	r_1, r_2, r_3, r_4	r_2, r_4	$False$	$\langle[2,4],[1,3],[2,4],[1,3]\rangle$
6	r_2, r_3	r_1, r_2, r_3, r_4	r_2, r_3	$False$	$\langle[2,3],[1,2],[1,4],[3,4]\rangle$
7	r_2, r_3, r_4	r_1, r_2, r_3, r_4	r_2, r_3, r_4	$False$	$\langle[2,4],[1,3],[1,4],[1,4]\rangle$
8	r_1	r_1, r_2, r_3, r_4	r_1	$False$	$\langle[1,1],[4,4],[3,3],[2,2]\rangle$
9	r_1, r_4	r_1, r_2, r_3, r_4	r_1, r_4	$False$	$\langle[1,4],[3,4],[2,3],[1,2]\rangle$
10	r_1, r_3	r_1, r_2, r_3, r_4	r_1, r_3	$False$	$\langle[1,3],[2,4],[1,3],[2,4]\rangle$
11	r_1, r_3, r_4	r_1, r_2, r_3, r_4	r_1, r_3, r_4	$False$	$\langle[1,4],[2,4],[1,3],[1,4]\rangle$
12	r_1, r_2	r_1, r_2, r_3, r_4	r_1, r_2	$False$	$\langle[1,2],[1,4],[3,4],[2,3]\rangle$
13	r_1, r_2, r_4	r_1, r_2, r_3, r_4	r_1, r_2, r_4	$False$	$\langle[1,4],[1,4],[2,4],[1,3]\rangle$
14	r_1, r_2, r_3	r_1, r_2, r_3, r_4	r_1, r_2, r_3	$False$	$\langle[1,3],[1,4],[1,4],[2,4]\rangle$
15	r_1, r_2, r_3, r_4	r_1, r_2, r_3, r_4	r_1, r_2, r_3, r_4	$True$	-

4.2 Sampling Attributes for Interval Pattern Structures

Our sampling method is based on picking large convex regions in the space. To achieve this, we use a many-valued context denoted as $(\mathtt{G}, \mathtt{N}, \mathtt{W}, \mathtt{J})$ to differentiate it from the sets in the representation context $(\mathtt{G}, \mathtt{M}, \mathtt{I})$. Given a set B and its closure in the representation context B'' –which we know to be different from $B^{\square\diamond}$– the goal of the sampling procedure is to find a set X such that $B \subseteq (X \cap B'')$. The sampling procedure starts by picking a random column in the many-valued context $n \in \mathtt{N}$. Then, we randomly pick a side to trim the ordered set $\mathtt{W}^n$ (*left* or *right*) to generate a new set $\hat{\mathtt{W}}^n$. A candidate set $X = \{g \in \mathtt{G} \mid n(g) \in \hat{\mathtt{W}}^n\}$ is created and we check whether $B \subseteq X$. If so, we return X if and only if $B'' \not\subseteq X$. In a different case, we pick a random column from $\mathtt{N}$ and proceed with the same instructions. This is a basic description of the sampling algorithm described in Algorithm 3. Some details on the creation of $\hat{\mathtt{W}}^n$ are left described only in the pseudocode.

Algorithm 3 is able to generate large convex regions by considering single dimensions of the space. It basically tries to answer the question: *Which is the largest region in any dimension which contains set B?* For example, let us try to sample an extent for the first row of Table 6 where $B = \{r_4\}$ and $B'' = \{r_1, r_2, r_3, r_4\}$. The many-valued context is showed in Table 2. Let us pick column b such that $W^b = \{1, 2, 3, 4\}$. Next, we pick *right* to create $\hat{\mathtt{W}}^b = \{1, 2, 3\}$. The candidate set is then $X_1 = \{r_2, r_3, r_4\}$ for which we have that $B \subseteq X_1$ and $B'' \not\subseteq X_1$ so we return $X_1 = \{r_2, r_3, r_4\}$.

We can observe that in the previous example $\{r_2, r_3, r_4\}$ corresponds to the image of an irreducible attribute in the representation context. Of course, this is a happy accident because we have made not-so-random decisions. True random decisions may lead to add reducible attributes into the representation context. However, since the random decisions are likely to pick large regions in the space, and thus attributes with a large image in $\mathtt{G}$, they are likely to be irreducible attributes.

Algorithm 2 adapts a sampling method into the generation of the representation context. Line 14 calls the sampling procedure such as the one described in Algorithm 3. Line 13 has been changed for a **while** instruction instead of an **if** instruction. This is because in this case the closure B'' should converge to $B^{\square\diamond}$ by the addition of one or more attributes into the representation context. This convergence is ensured since in the worst case scenario Algorithm 3 returns $B^{\square\diamond}$ as an attribute for the representation context.

Let us finish the previous example by using Algorithm 2. We notice that with $X_1 = \{r_2, r_3, r_4\}$ as the first object of the representation context we have that $\{r_4\}'' = \{r_2, r_3, r_4\}$ which is again different from $\{r_4\}^{\square\diamond}$. Consequently, Algorithm 2 calls for a new sample which by the same procedure described could be $X_2 = \{r_1, r_3, r_4\}$ (another happy accident). Notice that it cannot be again $\{r_2, r_3, r_4\}$ because of line 18 of Algorithm 3. Next, we have that $\{r_4\}'' = \{r_3, r_4\}$ calls for a new sample. This new sample could be $X_3 = \{r_1, r_2, r_4\}$ which renders $\{r_4\}'' = \{r_4\}^{\square\diamond}$. Algorithm 2 proceeds to calculate $\{r_3\}''$ which in the current state of the representation context yields $\{r_3, r_4\}$. If the SAMPLE procedure re-

Algorithm 2 A General Algorithm for Computing a Representation Context.

```
1: procedure RepresentationContext(𝓜, (·)^□, (·)^◇)                    ▷ 𝓜 = (G, N, W, J)
2:     M ← ∅
3:     I ← ∅
4:     𝒦 ← (G, M, I)
5:     A ← ∅
6:     while A ≠ G do
7:         for g ∈ G in reverse order do
8:             if g ∈ A then
9:                 A ← A \ {g}
10:            else
11:                B ← A ∪ {g}
12:                if B ≠ B'' then
13:                    while B'' ≠ B^□◇ do                              ▷ (*)
14:                        X ← Sample(B, B'', 𝓜)                        ▷ (*)
15:                        M ← M ∪ {m_X}                    ▷ (m_X is a new attribute)
16:                        I ← I ∪ {(h, m_X) | h ∈ X}                   ▷ (*)
17:                    end while
18:                end if
19:                if B''\A contains no element < g then
20:                    A ← B''
21:                    Exit For
22:                end if
23:            end if
24:        end for
25:        Output A
26:    end while
27:    return 𝒦
28: end procedure
```

turns $X_4 = \{r_1, r_2, r_3\}$ we have then that $\{r_3\}'' = \{r_3\}^{\square\diamond}$. It should be noticed that at this point the representation context is complete w.r.t. the interval pattern structure. This is, any subsequent closure will be calculated in the representation context alone. Table 5 shows the contranominal scale corresponding to the representation context generated.

5 Conclusions

In this paper we have presented a sampling strategy in order to compute a smaller representation context for an interval pattern structure. We propose two algorithms to achieve such a computation, a first naive version based on object exploration, and a second improved version that uses a sampling oracle to quickly find irreducible patterns. These irreducible pattern can be considered the *basis* of a pattern structure. This paper is a first step towards the computation of minimal representation of a pattern structure by means of sampling techniques.

Acknowledgments. This research work has been supported by the recognition of 2017SGR-856 (MACDA) from AGAUR (Generalitat de Catalunya), and the grant TIN2017-89244-R from MINECO (Ministerio de Economía y Competitividad).

References

1. Jaume Baixeries, Mehdi Kaytoue, and Amedeo Napoli. Computing Similarity Dependencies with Pattern Structures. In *Concept Lattices and their Applications*,

Algorithm 3 An interval pattern extent sampler algorithm.

```
1: procedure SAMPLE(B, B'', M)                                   ▷ M = (G, N, W, J)
2:     found ← False
3:     states ← list of size |N|
4:     for n ∈ N do
5:         side ← pick randomly from {0, 1}
6:         states[n] ← (side, 0, |W_n|)
7:     end for
8:     while not found do
9:         n ← pick randomly from N
10:        side, i, j ← states[n]
11:        a = i + side
12:        b = j + (side − 1)
13:        if a < b then
14:            X ← {g ∈ G | W_a^n ≤ n(g) ≤ W_b^n}
15:            side ← not side
16:            if B ⊆ X then
17:                i, j ← a, b
18:                if B'' ⊈ X then
19:                    found ← True
20:                end if
21:            end if
22:            states[n] = (side, i, j)
23:        end if
24:    end while
25:    return X
26: end procedure
```


volume 1062, pages 33–44. CEUR Workshop Proceedings (ceur-ws.org), 2013.

2. Jaume Baixeries, Mehdi Kaytoue, and Amedeo Napoli. Characterizing Functional Dependencies in Formal Concept Analysis with Pattern Structures. *Annals of Mathematics and Artificial Intelligence*, 72(1–2):129–149, 2014.
3. Alexandre Bazin. Comparing algorithms for computing lower covers of implication-closed sets. In *Proceedings of the Thirteenth International Conference on Concept Lattices and Their Applications, Moscow, Russia, July 18-22, 2016.*, pages 21–31, 2016.
4. Aleksey Buzmakov. *Formal Concept Analysis and Pattern Structures for mining Structured Data*. PhD thesis, University of Lorraine, Nancy, France, 2015.
5. Víctor Codocedo, Jaume Baixeries, Mehdi Kaytoue, and Amedeo Napoli. Sampling Representation Contexts with Attribute Exploration. In Diana Cristea, Florence Le Ber, and Baris Sertkaya, editors, *Proceedings of the 15th International Conference on Formal Concept Analysis (ICFCA)*, Lecture Notes in Computer Science 11511, pages 307–314. Springer, 2019.
6. Víctor Codocedo and Amedeo Napoli. Lattice-based biclustering using Partition Pattern Structures. In *Proceedings of ECAI 2014*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 213–218. IOS Press, 2014.
7. Brian A. Davey and Hilary A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, UK, 1990.
8. Benhard Ganter and Sergei Obiedkov. *Conceptual Exploration*. Springer, Berlin, 2016.
9. Benhard Ganter and Rudolph Wille. *Formal Concept Analysis*. Springer, Berlin, 1999.
10. Bernhard Ganter and Sergei O. Kuznetsov. Pattern Structures and Their Projections. In *Proceedings of ICCS 2001*, Lecture Notes in Computer Science 2120, pages 129–142. Springer, 2001.

11. Michel Habib and Lhouari Nourine. Representation of lattices via set-colored posets. *Discrete Applied Mathematics*, 249:64–73, 2018.
12. Mehdi Kaytoue, Víctor Codocedo, Aleksey Buzmakov, Jaume Baixeries, Sergei O Kuznetsov, and Amedeo Napoli. Pattern Structures and Concept Lattices for Data Mining and Knowledge Processing. In *Proceedings of ECML-PKDD 2015 (Part III)*, volume 9286 of *Lecture Notes in Computer Science*, pages 227–231. Springer, 2015.
13. Mehdi Kaytoue, Sergei O. Kuznetsov, and Amedeo Napoli. Revisiting Numerical Pattern Mining with Formal Concept Analysis. In Toby Walsh, editor, *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 1342–1347. IJCAI/AAAI, 2011.
14. Mehdi Kaytoue, Sergei O. Kuznetsov, Amedeo Napoli, and Sébastien Duplessis. Mining gene expression data with pattern structures in Formal Concept Analysis. *Information Sciences*, 181(10):1989–2001, 2011.
15. Sergei O. Kuznetsov. Pattern Structures for Analyzing Complex Data. In Hiroshi Sakai, Mihir K. Chakraborty, Aboul Ella Hassanien, Dominik Slezak, and William Zhu, editors, *RSFDGrC*, volume 5908 of *Lecture Notes in Computer Science*, pages 33–44. Springer, 2009.