Explanations for Itemset Mining by Constraint Programming: A Case Study Using ChEMBL Data

Maksim Koptelov, Albrecht Zimmermann, Patrice Boizumault, Ronan Bureau, Jean-Luc Lamotte

Explanations for Itemset Mining by Constraint Programming: A Case Study using ChEMBL data

Maksim Koptelov[1][0000−0001−9065−2827], Albrecht Zimmermann[1], Patrice Boizumault[1], Ronan Bureau[2], and Jean-Luc Lamotte[1]

[1] UNICAEN, ENSICAEN, CNRS – UMR GREYC, 14000, Caen, France
[2] CERMN/UNICAEN, 14200, Caen, France
name.lastname@unicaen.fr

Abstract. In sensitive applications, such as drug development, offering experts an explanation for why data mining operations arrive at certain results adds a very valuable facet. In this work we benefit from modelling the task as a Constraint Satisfaction Problem (CSP) twice: by adding multiple constraints to the mining process and by deriving pattern failure explanations. We illustrate experimentally how to apply our method on data originally retrieved from the ChEMBL database [14]. We also report some interesting dependencies discovered by our method which are not easy to observe when analysing data manually.

Keywords: Itemset mining · Constraint programming · Explainable AI

1 Introduction

With the recent surge in applications of machine learning, mainly deep learning, techniques to a variety of fields, the need for explanations for those techniques has also increased. Most of the techniques explaining machine learning models exploit the supervised nature of the problem setting, solving problems such as:

- Can we learn a symbolic model giving the same predictions?
- What are the minimal changes that need to be done to a data instance to change its predicted label?
- Can we identify features or image regions that contribute strongly to the prediction result?

In unsupervised data mining, however, especially in constraint-based pattern mining, labeled examples are typically not available, increasing the challenge. As a result, there are arguably more workshops for (interesting) work-in-progress papers on explainable data mining than there are publications that were accepted for conference proceedings or journals on the subject. The ones that do exist ignore explanations of itemset mining, a classical data mining task.

In addition, the questions change: since mined patterns are often starting points for further development, for instance in drug development, or "food for thought" that help formulate research hypothesis, their plausability and persuasiveness need to be supported.

Finally, this is clearly an application dependent subject. There are a number of applications in sensitive areas such as pharmaceutical or medical domains for which explanations are obligatory. As a common example, a chemical compound selected by a black-box classifier from a database of molecules cannot be approved by a pharmacist as a drug candidate, because of the high risks associated with the following production process costs [6].

Indeed, due to such considerations, our partner researchers at CERMN[3] are in need of explanations for their itemset mining and motivated this study. Their task requires finding molecular sub-structures which are able to discriminate between active and inactive molecules. In addition, they are obliged to use multiple constraints regarding the structure and properties of the resulting patterns, and they would like to have explanations on top of that.

More precisely, the answers to the following questions are desired:

- Why is this pattern not frequent/closed/emerging?
- Why could this constraint not be satisfied?
- How did the mining algorithm arrive at this particular solution satisfying the constraint(s)?

Straight-forward answers to these questions, e.g. "the pattern doesn't have enough overall support" or "the pattern has too much support in the class that was not targeted" are tautological and not very satisfying. Instead, a practitioner would be interested in knowing what element of the pattern or which other constraint forced the support below a given threshold or lead to the inclusion of transactions that reduce the growth rate.

These are questions that have already been asked in similar form in the constraint programming (CP) community [18, 12]. We formulate our problem setting as one of constraint-based itemset mining, for which CSP solutions have been proposed [7, 15]. In addition, past work has added explanations to CSP solvers [3, 13]. We therefore base our work in part on proposals made to answer explanatory questions in CP. In this work, we develop an approach for pattern failure explanations, which is our main contribution. We demonstrate the application of it on data derived from the ChEMBL database.

The rest of the paper is organised as follows. Section 2 highlights important works related to the topic. Section 3 outlines the problem setting and used formalisms, including how to derive explanations for itemset mining based on constraint failure. Section 4 shows and discusses the results of our case study on the ChEMBL data. Finally, we review and discuss future improvements in Conclusion.

2 Related Work

Following [9], we define *data mining* as the search for valid, novel, potentially useful, and ultimately understandable patterns in the data. One of the seminal

[3] Centre d'Etude et de Recherche du Médicament de Normandie: https://cermn.unicaen.fr

tasks in data mining is itemset mining, for which a number of more specialized problems, such as frequent itemset mining, frequent closed itemset mining, discriminative or emerging itemset mining and others have been defined [11]. In this paper, we mainly focus on those first three tasks.

Traditional approaches for itemset mining take the form of specialized breadth-first [1, 2, 31] and depth-first algorithms [32, 16]. An alternative approach involves using CP [7, 15], a general declarative methodology for solving constraint satisfaction problems. Constraint programs specify the problem and a general solver tries to find a solution. A clear advantage lies in the universality of this approach. The new task could be modelled by adding new constraints while in traditional approaches the algorithm must be redesigned from scratch each time. Another advantage of CP systems is a possibility of result explanation [3, 13]. In this work we benefit from the latter by modelling the itemset mining task with CP.

Most of the works on explainable AI are focused on explanations for machine learning [27, 25]. While work on directly explainable data mining are rare, interactive data mining has been proposed as a first approximation of interpretable data mining involving both the miner and the domain expert, as well as the data itself [22, 17]. The ultimate goal of such a process is to make pattern mining more practically useful by making the end user understand *during* the mining process how mining results come to pass. Discrimination-aware data mining exists for more than a decade now [26, 19]. It mainly focuses on developing methods for protecting from unfair classification models, especially when they might affect somebody's life. Work on visual data mining [10, 30, 4] attempt to make the data mining process understandable through visualization. Some of them offer explanations for clustering or binary classification tasks [29, 5]. Finally, there are few works which use explanations for improving the data mining results. For instance, [21] tries to mix data mining with domain expert knowledge in order to improve the quality of discovered patterns in the medical domain. Likewise, [20] developed an approach for mining surprising patterns and generating explanations. Based on association rule mining, the approach that they proposed uses expert knowledge to improve the search and provide explanations.

In this paper, we take a step towards explaining itemset mining, one of the core tasks of data mining. This is the first work in this direction to the best of our knowledge.

3 Preliminaries

As described above, the result of a constraint-based pattern mining operation is a set of patterns. A user might want to know now why certain patterns were included and others were not. The straight-forward answer is simple: the patterns satisfied the specified constraints (or not). This might not be sufficient information, however: specifying constraints and deciding on threshold values is not an easy task, and a small change may lead to a large change in results. In addition, especially when a number of complex constraints are combined, their interplay can lead to the inclusion or exclusion of patterns in unexpected man-

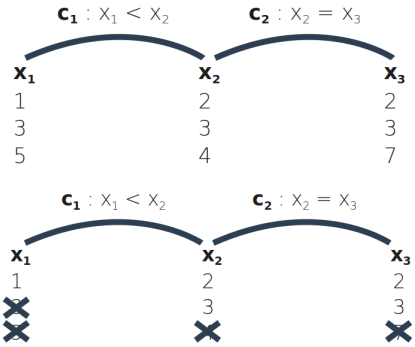

Fig. 1. An example of a CSP (top) and the result of its filtering (bottom)

ners, which are not easy to understand without additional explanation. Gaining such understanding will help in formulating future constraints. Our proposal for fournishing such explanations is to exploit *pattern failure* explanations in CSPs. In this section, we lay out the itemset mining problem, the CSP framework, and how to model itemset mining, as well as how to derive explanations.

3.1 Itemset Mining

The pattern mining task we address in this paper is the classical itemset mining one: *given* a set of *items* $\mathcal{I} = \{i_1, \ldots, i_m\}$, a transaction set $\mathcal{T} = \{t_1, \ldots, t_n \mid t_i \subseteq \mathcal{I}\}$, and a (combination of) constraint(s) $\mathcal{C} : \mathcal{I} \times \mathcal{T} \mapsto \{\text{true,false}\}$, *find* $Th(\mathcal{I}, \mathcal{T}, \mathcal{C}) = \{p \subseteq \mathcal{I} \mid \mathcal{C}(p, \mathcal{T}) = \text{true}\}$.

The *support* of an itemset is the cardinality of the set of transactions in which it is contained: $supp(p, \mathcal{T}) = |\{t \in \mathcal{T} \mid p \subseteq t\}|$. Given a threshold θ_f the minimum support (frequency) constraint is defined as $freq(p, \mathcal{T}) = \text{true} \Leftrightarrow supp(p, \mathcal{T}) \geq \theta_f$. An itemset is *closed* if none of its strict specializations has the same support: $closed(p, \mathcal{T}) = true \Leftrightarrow \forall p' \supset p : supp(p', \mathcal{T}) < supp(p, \mathcal{T})$.

Finally, given a labeling $l : \mathcal{I} \mapsto \{+, -\}$, $\mathcal{T}^+ = \{t \in \mathcal{T} \mid l(t) = +\}$, $\mathcal{T}^- = \mathcal{T} \setminus \mathcal{T}^+$, a quality measure $\sigma : \mathcal{I} \times \mathcal{T}^+ \times \mathcal{T}^- \mapsto \mathbb{R}$, a threshold θ_d, an itemset is *emerging/discriminative*: $disc(p, \mathcal{T}) = \text{true} \Leftrightarrow \sigma(p, \mathcal{T}^+, \mathcal{T}^-) \geq \theta_d$.

3.2 Constraint Programming

General CSP Context A classical CSP is defined by a triplet (V, D, C) in which $V = \{X_1, X_2, ..., X_n\}$ is a set of variables, $D = \{D_1, D_2, ..., D_n\}$ the set of domains of variables, with D_i a finite set containing the possible values for the variable X_i, and $C = \{c_1, c_2, ..., c_k\}$ a set of constraints. A solution of the CSP is a complete instantiation S such that all the constraints C are satisfied by S.

Consider an example with $V = \{X_1, X_2, X_3\}$, $D_1 = \{1, 3, 5\}$, $D_2 = \{2, 3, 4\}$, $D_3 = \{2, 3, 7\}$ and $c_1 : X_1 < X_2$, $c_2 : X_2 = X_3$ (Fig. 1). There are two possible solutions for this problem: $X_1 = 1$, $X_2 = 2$, $X_3 = 2$ or $X_1 = 1$, $X_2 = 3$, $X_3 = 3$.

Explanations for CSPs The CSP framework is not only a powerful tool for modelling different type of constraints, but also for providing explanations (Section 2). In this work, we deal with explanations for value removal as the simplest to implement and interpret.

An *explanation for value removal* is a subset of the set of constraints C such that the conjunction of these constraints leads to the removal of the value a from the domain of the variable X_i. In case of multiple explanations, this expression becomes a disjunction of conjunctions:

$$Expl(X_i \neq a) = \bigvee \Big(\bigwedge_{i \in [1..k]} c_i \implies X_i \neq a \Big).$$

An example of such explanations for the CSP in Fig. 1: $Expl(X_1 \neq 5) = c_1$ (there is no value > 5 in the domain of X_2), $Expl(X_2 \neq 4) = c_2$ (there is no value $= 4$ in the domain of X_3), $Expl(X_1 \neq 3) = c_1 \wedge Expl(X_2 \neq 4) = c_1 \wedge c_2$ (c_2 removes the value 4 from X_2, and c_1 in turn removes 3 from X_1).

Modeling Itemset Mining as a CSP To model the itemset mining problem with CP, we follow [15]: the CSP must be defined by a triplet (V,D,C), in which $V = I \cup T$ a set of variables s.t.: $I = \{I_1, I_2, ..., I_m\}$ a set of items, $T = \{T_1, T_2, ..., T_n\}$ a set of transactions, $D = \{D_{I_1}, ..., D_{I_m}, D_{T_1}, ..., D_{T_n}\}$ a set of domains of variables with $D_i = \{0, 1\}$, $C = \{c_1, c_2, ..., c_k\}$ a set of constraints. As for the latter refined constraints proposed by [15] can be used according to the task.

Consider a toy example. Given a set of transactions $T = \{ACD, ABD, CD\}$ and minimum frequency $\theta_s = 2$, we would like to find all frequent closed patterns. To model the problem as a CSP, we define $DB = \{\{1, 0, 1, 1\}, \{1, 1, 0, 1\}, \{0, 0, 1, 1\}\}$, $V = \{I_1, I_2, I_3, I_4, T_1, T_2, T_3\}$, $D = \{D_{I_1}, D_{I_2}, D_{I_3}, D_{I_4}, D_{T_1}, D_{T_2}, D_{T_3}\}$ with $D_{X_i} = \{0, 1\}$, $C = \{c_1, c_2, ..., c_{11}\}$ with the constraints defined as in Fig. 2.

There are three solutions to this problem: AD, CD, D. Fig. 2 also demonstrates the search process. Branching of the search tree usually stops when a solution is found, then the search backtracks to another branch until all the solutions are retrieved. In our setting, however, we continue the search until all the failures are found (Failure 1-7 in Fig. 2). We use them later to explain a pattern failure which we define as follows. A *pattern failure* is a state of the CSP in which one of the itemset domains is empty:

$$I_1 = [\,] \vee I_2 = [\,] \vee ... \vee I_m = [\,] \implies CSP \rightarrow Fail.$$

3.3 Explanations for Itemset Mining

As explained above, CSPs allow to derive explanations. The default approach does not allow to explain a success (a solution, specific pattern or presence of an item in the solution) in an effective way: it can only say that we have this as a solution because it satisfied all the constraints. However, it *is* possible to explain a failure (no solution at all, a particular pattern does not belong to the solution

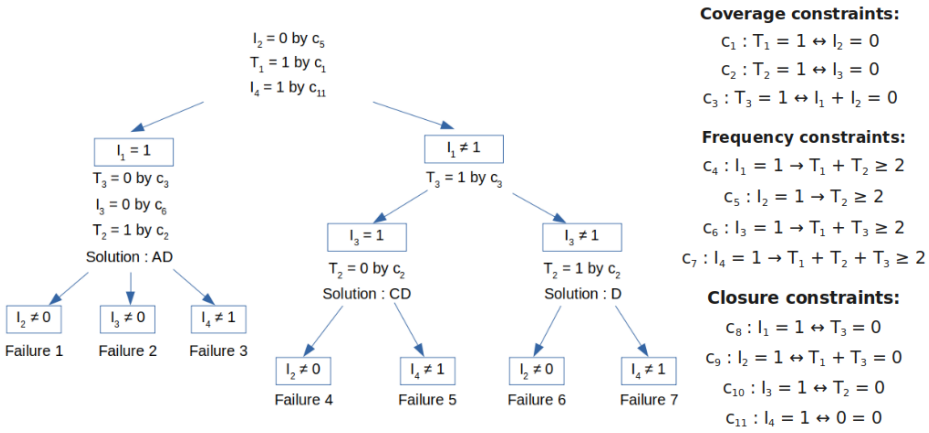

Fig. 2. Constraints and search tree for a toy example of itemset mining by CSP

etc.) more effectively by interpreting constraints which led to that failure. In addition, there could be an exponential number of explanations. We therefore choose to keep only one: the shortest one. Here, we present an approach for that.

Our approach for finding explanations for pattern failure is a 4-step process:

S1 Initialize domains with the elements of a pattern whose failure (i.e. absence from the solution) needs to be explained. The pattern needs to be precisely specified by the user/chemical expert

S2 Obtain different explanations for pattern failure in the form of conjunctions and/or disjunctions of constraints which led to emptying one of the itemset domains

S3 Select the shortest explanation w.r.t. the number of constraints

S4 Interpret the constraints in that explanation using logical inference and/or analysing them manually

Following our example in Figure 2, we can explain, for instance, why pattern AB is not in the solution. The shortest explanation will be:

$$Expl(AB \rightarrow \text{Fail}) = c_5 \wedge c_{11}.$$

We can interpret c_5 (the frequency constraint) as "if B is in the itemset ($I_2 = 1$), the itemset must be frequent ($T \geq 2$)". Since $T \geq 2$ is False, B must be removed from the pattern, which can be rephrased as "the pattern cannot be frequent if B is present". Closed itemset mining aims at avoiding redundant itemsets and the closure constraint checks if all transactions contain the same element as without it the itemset cannot be closed. We can thus interpret c_{11} (the closure constraint) as "there must be D in the itemset ($I_4 = 1$), otherwise it cannot be a closed pattern" ($I_4 = 1$ if and only if *True*, where *True* corresponds to $0 = 0$).

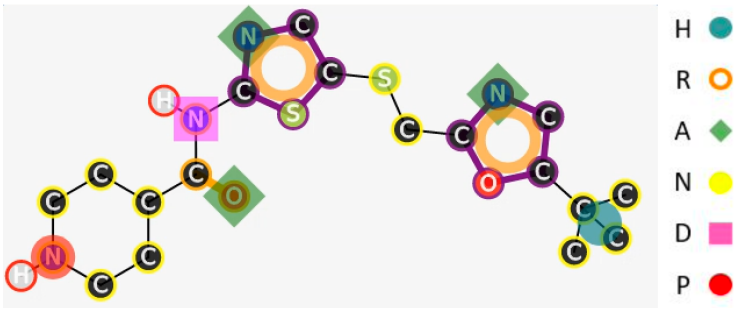

Fig. 3. An example of a molecule (left) and its pharmacophoric features (right): hydrogen-bond acceptors (A) and donors (D), negatively (N) and positively (P), charged ionizable groups, hydrophobic regions (H) and aromatic rings (R)

4 Case study

We illustrate our approach on a set of molecular data, from which we aim to mine combinations of chemically meaningful subgraph patterns.

4.1 Data and representation

Our data originally is a set of BCR-ABL inhibitors (target ID 1862) that have been extracted from the ChEMBL[4] database, a widely used database in computational drug discovery [14]. In this study, we would like to understand the mechanism of action on the BCR-ABL target.

After several steps of preprocessing, our set is composed of 739 molecules, 387 of which are labeled as active and 352 as inactive. A molecule is called *active* if it causes the target to react. If a molecule does not generate a sufficient reaction at the level of the target, it is considered to be *inactive*. Each molecule is represented as the 2D/3D arrangement of molecular features that are necessary for a drug candidate to interact with a biological target in a specific binding site [8]. In total there are 6 features in our data (Figure 3). Graphs in this representation are also referred to as *pharmacophores*, with its *order* O_n equal to its number of vertices (Figure 4). For example, the molecule in Figure 3 includes the following pharmacophores: |P|D||5|, |P|A||5|, |P|A||7|, |P|R||6|, |P|A||12|, |R|R||3|, |R|A||0|, |R|H||1|, |R|H||6|, |A|A||6| etc. (28 in total).

From our data, we mined 258 distinct 2D pharmacophores of O_2 having minimum support 10, using Norns [24]. The objective of the study is to explain why a molecule is active by identifying the pharmacophores which cause activity.

4.2 Mining task

We want to identify combinations of at most 8 such pharmacophores that are shared by a significant number of molecules (at least 12%-15% of the data) and

[4] A manually curated database of bioactive molecules with drug-like properties

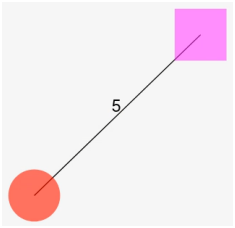

Fig. 4. Example of an O_2 pharmacophore |P|D||5| with a positively charged ionizable group (P) and a hydrogen-bond donor features (D) with the distance 5 between them

appears more often in one of the two classes than in the other, and that are not subsets of each other.

4.3 Experimental setting

Using mined pharmacophores, we can represent each molecule as a transaction encoding whether pharmacophores are present or not, giving us the classical itemset mining setting. We implemented refined constraints from [15] and a CSP solver in Python[5]. We adopted the AC3 algorithm [23] for the latter and implemented the MAC algorithm [28] for backtracking the search.

Following preliminary experiments, we use χ^2 as the discriminatory measure. We implemented χ^2 as in [15]. Constraint thresholds were set to $\theta_{size} = 8$ for size, $\theta_{supp} = 100$ for minimum support, and minimum thresholds for χ^2 ($\theta_{\chi^2} \in \{48, 64, 96, 128\}$). In addition, we experimented with adding a *purity* constraint, i.e. patterns present in one class only, which we defined as follows:

$$I_i = 1 \rightarrow min\Big(\sum_{t^+} DB_{ti} \cdot T_t, \sum_{t^-} DB_{ti} \cdot T_t\Big) = 0 \tag{1}$$

Finally, we try to answer why changing one *item* in a pattern and adding another one changes the class of solution from *pure* (i.e. covering only active or inactive molecules) to *not pure* (covering both active and inactive molecules).

4.4 Experimental results

As can be seen from Table 1, the pure solution constraint reduces the number of results dramatically – on average by three order of magnitude. Moreover, the results corresponding to the pure solution and θ at 48, 64 and 96 remain the same. In addition, there are no inactive solutions in case of pure patterns or with θ at 96 and 128, which is not a problem per se since our aim is to explain active solutions. On one hand, a smaller number of patterns is easier to evaluate manually. The main drawback of this modeling that the pure solutions found cover only 30% of molecules. This is not really desirable for a chemical expert,

[5] https://github.com/koptelovmax/dmbycsp

Table 1. Emerging pattern mining with χ^2 as a discriminative measure, pattern size not exceeding 8 and minimum frequency limited to 100

θ_{χ^2}	Pure	Found solutions			Pattern size			Coverage	Frequency		
		Total	Active	Inactive	min	max	median		min	max	mean
48	×	85037	84530	507	1	8	7	100,00 %	100	682	157,7
	✓	47	47	0	6	8	8	30,85 %	101	175	126,7
64	×	69060	68995	65	1	8	7	100,00 %	100	656	164,6
	✓	47	47	0	6	8	8	30,85 %	101	175	126,7
96	×	44013	44013	0	1	8	7	99,32 %	100	624	179,5
	✓	47	47	0	6	8	8	30,85 %	101	175	126,7
128	×	24645	24645	0	1	8	7	97,56 %	119	547	198,3
	✓	27	27	0	6	8	8	30,04 %	119	175	136,9

and solutions combining to cover most of the molecules are required. We thus go to the next step of our study where we will try to understand the interior mechanics behind our mining process.

Towards explaining pattern failure After discussing with the chemical experts we collaborate with, they asked for an explanations for why changing one item (pharmacophore in our case) leads to changing the class of solution from pure to not pure:

$$ABC \text{ (pure)} \leftrightarrow AEC \text{ (not pure)}.$$

While studying this phenomena in more detail, we realised that the actual change of the class happens when one element is *removed* from the pattern (Fig. 5). In other words:

$$ABC \text{ (pure)} \rightarrow AC \text{ (not pure)} \rightarrow AEC \text{ (not pure)}.$$

We would like to explain the first part: why removing an item makes the pattern not pure. Consider, for instance, the first two lines in the example in Fig. 5, where solution 17863 is pure, and 17902 is not. Our methodology for answering that is the following:

1. Model the problem using the purity constraint (Eq. 1)
2. Explain using our method from Section 3.3 why the combination of molecule features |D|R||1| |D|R||3| |A|H||11| |R|R||1| |R|H||5| is a failure
3. Verify why adding |A|R||0| to the pattern gives a solution. For that:
   (a) Find its purity constraint
   (b) Explain why it became *true*

After an initialization step we move directly to S2 of our approach from Section 3.3, which will generate the following explanations:

$$Expl(|\text{D}|\text{R}||1| \ |\text{D}|\text{R}||3| \ |\text{A}|\text{H}||11| \ |\text{R}|\text{R}||1| \ |\text{R}|\text{H}||5| \rightarrow \text{Fail}) =$$
$$= Expl(|\text{D}|\text{R}||1| \neq 1) \ \vee \ Expl(|\text{D}|\text{R}||3| \neq 1) \ \vee \ Expl(|\text{A}|\text{H}||11| \neq 1) \vee$$

17863	|D|R||1| |D|R||3| **|A|R||0|** |A|H||11| |R|R||1| |R|H||5|	(171: +171 -0)	202.36
17902	|D|R||1| |D|R||3| |A|H||11| |R|R||1| |R|H||5|	(172 : +171 -1)	198.95
17899	|D|R||1| |D|R||3| **|A|R||2|** |A|H||11| |R|R||1| |R|H||5|	(171: +170 -1)	197.42

Fig. 5. Pattern failure example. Columns are: id of solution, pattern, frequency and χ^2 value. Red colour represents removal of an item and blue is adding an item

$\vee\ Expl(|\mathrm{R}|\mathrm{R}||1| \neq 1) \vee\ Expl(|\mathrm{R}|\mathrm{H}||5| \neq 1) = c_{2361} \vee c_{2372} \vee c_{2440} \vee c_{2461} \vee c_{2488}.$

According to S3, the shortest explanation is one of those constraints, for instance:

c_{2361} – purity constraint:
before filtering: $|\mathrm{D}|\mathrm{R}||1| = 1 \rightarrow min(384, 298) = 0$ *False*
after filtering: $|\mathrm{D}|\mathrm{R}||1| = 1 \rightarrow min(171, 1) = 0$ *False*

Finally, we try to interpret this following S4. After filtering, the CSP this constraint remains *false*, but its coverage changes – each of the items in the pattern covers 171 active molecules and 1 inactive one:

c_{2361}: $|\mathrm{D}|\mathrm{R}||1| = 1 \rightarrow min(T_1 + T_2 + ... + T_{356} + T_{379}, T_{429}) = 0$ ***False***

We also know that the inactive molecule is represented by the variable T_{429} (or by ChEMBL ID 1984038).

Next, we would like to explain why adding $|\mathrm{A}|\mathrm{R}||0|$ to the solution makes the pattern pure. For that, one needs to instantiate the CSP with a new pattern including $|\mathrm{A}|\mathrm{R}||0|$ and check its purity constraint after filtering:

$c_{2417} : |\mathrm{A}|\mathrm{R}||0| = 1 \rightarrow min(171, 0) = 0$ *True*

As can be seen from c_{2417}, our pattern is included only in active molecules. To explain for a user who is not a data mining expert why removing $|\mathrm{A}|\mathrm{R}||0|$ from the pattern affects its purity, one can draw the Euler diagram (Fig. 6). In that diagram, the pattern containing all pharmacophores including $|\mathrm{A}|\mathrm{R}||0|$ will be present only in active molecules. This is the type of information which is laborious to observe manually, but can be easily derived using a CSP.

Finally, we would like to explain why the purity constraint associated with $|\mathrm{A}|\mathrm{R}||0|$ becomes *true*, especially given that before filtering it was *false*:

$c_{2417} : |\mathrm{A}|\mathrm{R}||0| = 1 \rightarrow min(T_1 + T_2 + ... + T_{787}, T_{388} + T_{389} + ... + T_{739}) = 0$ *False*

To do that we need to explain why $T_{388} \neq 1$, ..., $T_{739} \neq 1$:

$Expl(T_{388} \neq 1) = c_{388}$, where c_{388} – coverage constraint:
$T_{388} = 1 \leftrightarrow |\mathrm{D}|\mathrm{R}||3| + |\mathrm{A}|\mathrm{H}||11| + |\mathrm{R}|\mathrm{R}||1| = 0$ *False*
...
$Expl(T_{739} \neq 1) = c_{739}$, where c_{739} – coverage constraint:
$T_{739} = 1 \leftrightarrow |\mathrm{D}|\mathrm{R}||1| + |\mathrm{A}|\mathrm{H}||11| + |\mathrm{R}|\mathrm{H}||5| = 0$ *False*

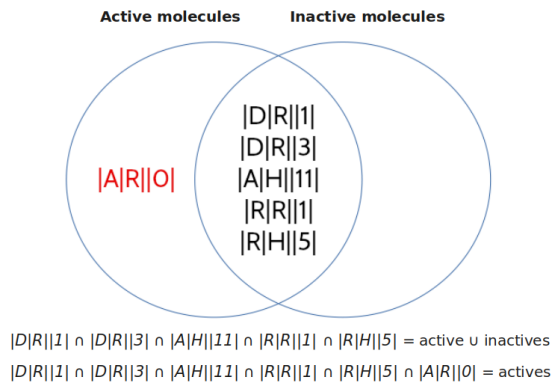

Fig. 6. Active, inactive molecules and their intersection

These constraints can be interpreted as follows: the combination of molecular features |D|R||3| |A|H||11| |R|R||1| must cover molecule T_{388} (ChEMBL ID 1836675), ..., and |D|R||1| |A|H||11| |R|H||5| must cover T_{739} (ChEMBL ID 281470), otherwise the coverage condition fails.

This is the type of information which can be easily retrieved with our method, and which can be useful for chemical experts.

Towards explaining constant constraints outcomes We noticed that certain constraints are always *true* or *false*. For instance, in our example in Figure 2, there are two constraints which always remain constant: c_5 (always *false*) and c_{11} (always *true*). In that toy example they can be interpreted as follows: if there is B in the pattern it is always not frequent (c_5); there must be D in the solution to be closed (c_{11}). Both of these conditions hold for our simple CSP since each solution contains item D and non of them has B.

Now if we verify which constraints remain constant for our ChEMBL set with the constraint thresholds $\theta_{\chi^2} = 128$, $\theta_{size} = 8$, $\theta_{supp} = 100$, allowing pure solutions only, we find that 363 constraints (out of 2510 used to model the CSP) remain constant:

- 159 frequency constraints – always *false*
- 194 discriminative constraints – always *false*
- 2 size constraints – always *true*
- 9 purity constraints – always *true*

If we interpret them, we get the following information:

- frequency constraints – if there is |P|P||3|, |P|D||10|, |P|D||11|, ..., |H|H||9| (159 in total) in the pattern, it is always not frequent
- discriminative constraints – if there is |P|P||3|, |P|D||10|, |P|D||11|, ..., |H|H||9| (194 in total) in the pattern, it is always not discriminating

- size constraints – if the pattern is included in molecule T_{671} (ChEMBL ID 1241863) or T_{696} (ChEMBL ID 1241772) its size is always less than 8
- purity constraints – |P|D||17|, |N|D||8|, |N|D||9|, ..., |R|H||19| (9 in total) are covered only by pure molecules:
  - |P|D||17|, ..., |R|H||19| (7 in total) are included in active molecules only
  - |N|D||8|, |N|D||9| – in inactive molecules only

This information can be read off without rerunning the mining operation. This can be useful for chemical experts to get quick-shot statistics on the data, explain why particular patterns in the solution do not include particular elements, modify the data set, or adjust constraint settings before repeating mining.

5 Conclusion

In this paper, we have explained how one can use constraint failure explanations in CSPs to explain why certain patterns do not appear in a solution set. These explanations can then be used to identify problematic data instances, or to modify constraint parameters. In a chemoinformatics use case, we have shown how such explanations and the identification of particular phenomena can look in practice.

A drawback of our method is that patterns to be explained need to be specified manually, and explanations need to be interpreted to arrive at statements about the data. In future work, we will therefore look at how generate patterns automatically, e.g. by looking at syntactically similar patterns, and how to post-process explanations to highlight interesting data. We would also think about how we could improve the explanations which we already generated. For the last we first need to get a detailed feedback from the experts.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB. vol. 1215, pp. 487–499 (1994)
2. Bodon, F.: A fast apriori implementation. In: FIMI. vol. 3, p. 63 (2003)
3. Bogaerts, B., Gamba, E., Guns, T.: A framework for step-wise explaining how to solve constraint satisfaction problems. Artificial Intelligence **300**, 103550 (2021)
4. Bouali, F., Guettala, A., Venturini, G.: Vizassist: an interactive user assistant for visual data mining. The Visual Computer **32**(11), 1447–1463 (2016)
5. Cortez, P., Embrechts, M.: Using sensitivity analysis and visualization techniques to open black box data mining models. Information Sciences **225**, 1–17 (2013)
6. Couronne, C., Koptelov, M., Zimmermann, A.: Prepep: A light-weight, extensible tool for predicting frequent hitters. In: ECML PKDD. pp. 570–573. Springer (2020)
7. De Raedt, L., Guns, T., Nijssen, S.: Constraint programming for itemset mining. In: KDD. pp. 204–212 (2008)
8. Dror, O., et al.: Novel approach for efficient pharmacophore-based virtual screening: method and applications. J. of chem. inf. and modeling **49**(10), 2333–2343 (2009)

9. Fayyad, U., Piatetsky-Shapiro, G., Smyth, P.: From data mining to knowledge discovery in databases. AI magazine **17**(3), 37–37 (1996)
10. Ferreira, M., Levkowitz, H.: From visual data exploration to visual data mining: A survey. IEEE trans. on visualization and comp. graphics **9**(3), 378–394 (2003)
11. Fournier-Viger, P., Lin, J.C.W., Vo, B., Chi, T., Zhang, J., Le, H.: A survey of itemset mining. Data Mining and Knowledge Discovery **7**(4), e1207 (2017)
12. Freuder, E.: Explaining ourselves: human-aware constraint reasoning. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 31 (2017)
13. Gamba, E., Bogaerts, B., Guns, T.: Efficiently explaining CSPs with unsatisfiable subset optimization. In: (IJCAI). pp. 1381–1388 (2021)
14. Gaulton, A., et al.: The chembl database in 2017. Nucleic acids research **45**(D1), D945–D954 (2017)
15. Guns, T., Nijssen, S., De Raedt, L.: Itemset mining: A constraint programming perspective. Artificial Intelligence **175**(12-13), 1951–1983 (2011)
16. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: A frequent-pattern tree approach. DMKD **8**(1), 53–87 (2004)
17. Holzinger, A., Dehmer, M., Jurisica, I.: Knowledge discovery and interactive data mining in bioinformatics-state-of-the-art, future challenges and research directions. BMC bioinformatics **15**(6), 1–9 (2014)
18. Jussien, N., Ouis, S.: User-friendly explanations for constraint programming. In: Int. Conf. on Principles and Practice of CP (2001)
19. Kashid, A., Kulkarni, V., Patankar, R.: Discrimination-aware data mining: a survey. International Journal of Data Science **2**(1), 70–84 (2017)
20. Kuo, Y.T., Lonie, A., Pearce, A.R., Sonenberg, L.: Mining surprising patterns and their explanations in clinical data. Applied AI **28**(2), 111–138 (2014)
21. Kuo, Y.T., et al.: Domain ontology driven data mining: a medical case study. In: 2007 international workshop on domain driven data mining. pp. 11–17 (2007)
22. Leeuwen, M.: Interactive data exploration using pattern mining. In: Interactive knowledge discovery and data mining in biomedical informatics, pp. 169–182 (2014)
23. Mackworth, A.K.: Consistency in networks of relations. AI **8**(1), 99–118 (1977)
24. Métivier, J.P., et al.: The pharmacophore network: a computational method for exploring structure–activity relationships from a large chemical data set. Journal of Medicinal Chemistry **61**(8), 3551–3564 (2018)
25. Pedreschi, D., et al.: Meaningful explanations of black box ai decision systems. In: AAAI. vol. 33, pp. 9780–9784 (2019)
26. Pedreshi, D., et al.: Discrimination-aware data mining. In: KDD. pp. 560–568 (2008)
27. Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you? explaining the predictions of any classifier. In: KDD. pp. 1135–1144 (2016)
28. Sabin, D., Freuder, E.: Contradicting conventional wisdom in constraint satisfaction. In: Int. Workshop on Principles and Practice of Constraint Programming. pp. 10–20. Springer (1994)
29. Soukup, T., Davidson, I.: Visual data mining: Techniques and tools for data visualization and mining. John Wiley & Sons (2002)
30. Velu, C., Kashwan, K.: Visual data mining techniques for classification of diabetic patients. In: IACC. pp. 1070–1075. IEEE (2013)
31. Wu, H., Lu, Z., Pan, L., Xu, R., Jiang, W.: An improved apriori-based algorithm for association rules mining. In: 6th FSKD. vol. 2, pp. 51–55. IEEE (2009)
32. Zaki, M.J.: Scalable algorithms for association mining. IEEE transactions on knowledge and data engineering **12**(3), 372–390 (2000)