ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-849

Graph abstraction for closed pattern mining in attributed networks

Henry Soldano and **Guillaume Santini** [1]

Abstract. We address the problem of finding patterns in an attributed graph. Our approach consists in extending the standard methodology of frequent closed pattern mining to the case in which the set of objects, in which are found the pattern supports, is the set of vertices of a graph, typically representing a social network. The core idea is then to define graph abstractions as subsets of the vertices satisfying some connectivity property within the corresponding induced subgraphs. Preliminary experiments illustrate the reduction in closed patterns we obtain as well as what kind of abstract knowledge is found via abstract implications rules.

1 Introduction

We address here the problem of discovering patterns in an attributed graph. Most previous work focus on the topological structure of the patterns, thus ignoring the vertex properties, or consider only local or semi-local patterns [9]. In [3] patterns on co-variations between vertex attributes are investigated in which topological attributes are added to the original vertex attributes and in [13] the authors investigate the correlation between the support of an attribute set and the occurrence of dense subgraphs. These works, either starts from the graph and consider vertex attributes as some additional information to consider when searching for interesting patterns or consider patterns as structure/attributes pairs.

What we propose in this paper is to consider attribute patterns and to submit their occurrences to connectivity constraints. We consider attribute patterns in the standard closed itemset mining approach developed in Formal concept Analysis (FCA)[6], Galois Analysis [4], and Data Mining (see for instance [11]). These methods search for frequent support-closed attribute patterns, easily computed using a closure operator, together with the corresponding rule bases. We use then the graph $G = (O, E)$ in the following way: each pattern support $e \subseteq O$, as a set of vertices, induces a subgraph G_e of G, and this subgraph is then simplified by removing vertices in various ways, denoted as graph abstractions. The general idea is that the vertices of this *abstract subgraph* all satisfy some topological constraint, as for instance a degree exceeding some threshold, and form the *abstract support* of the pattern. We define graph abstractions in such a way that the standard machinery is preserved: we can still have a closure operator, and easily compute *abstract closed term* and *abstract rules*. As a result we find less closed patterns, each original implication rule is preserved, but new rules appear, revealing some new knowledge which holds on some abstract level.

Technically, we benefit from the notion of *extensional abstraction* that has been recently introduced [12, 14] and that consists in only considering a subset A of the support space 2^O. Accordingly, the support of any pattern, i.e. a subset of O, is reduced into a smaller *abstract support* belonging to A. It has been shown that the main properties mentioned above are preserved through such abstractions: the corresponding *abstract support-closed patterns* are closed patterns according to an abstract closure operator, they form a lattice smaller than the original one, the abstract equivalence relation is coarser than the original one, and implication bases are defined in the same way as in the non abstract case. Such an abstraction always represents some external a priori information that results in simplified representations. Abstract closed patterns have mainly been investigated when the a priori information was some categorization, as a taxonomy or a partition.This has led to alpha lattices and alpha closed patterns [17].

The main purpose of this paper is to exhibit a new kind of abstraction relying, as an a priori information, on a graph connecting the objects of O. This means that when searching for closed frequent patterns and for rules that hold in some dataset of objects, we can take advantage of the graph relating these objects.

In the following example, we consider the graph $G = (O, E) = (\{a, b, c, d, e, f\}, \{a\text{-}b, a\text{-}c, a\text{-}d, a\text{-}e, b\text{-}c, d\text{-}e, d\text{-}f, e\text{-}f\})$. The vertices in O are objects whose labels are itemsets, i.e. subsets of the attribute set {x,y,z,k,w} according to the boolean Table 1. Consider the pattern $t = xy$ whose support is $\{a, b, c, d, f\}$. The corresponding induced subgraph is then $(\{a, b, c, d, f\}, \{a\text{-}b, a\text{-}c, a\text{-}d, b\text{-}c, d\text{-}f\})$. We consider now that an abstract support is such that in the corresponding induced subgraph all vertices have a degree greater than or equal to 2. As a result we will remove from the support $\text{ext}(xy)$ the vertex f whose degree is strictly smaller than 2, then consider the subgraph induced with the remaining vertices $\{a, b, c, d\}$, remove d whose degree is now only 1, and, observing that we have now an induced subgraph satisfying the degree requirements, state that we have reached a fix-point that represents the abstract support of t. This is illustrated in Figure 1.

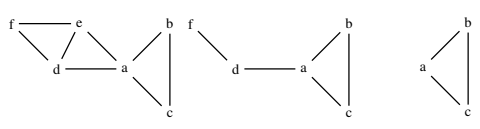

Figure 1. Given the graph G drawn in part (a), the subset of vertices $\{a, b, c, d, f\}$ induces the subgraph drawn in part (b). The graph abstraction oft the latter associated to the property $degree \geq 2$ is drawn in part (c).

As the abstract support of xy is $\{a, b, c\}$, the closure is now obtained by intersecting the corresponding object descriptions and results in the closed pattern xyk. What happened here is that if we consider as equivalent two patterns with equal support, the equiv-

[1] Université Paris 13, Sorbonne Paris Cite, L.I.P.N UMR-CNRS 7030 F-93430, Villetaneuse, France

Objects/Items	x	y	z	k	w
a	1	1	1	1	0
b	1	1	0	1	1
c	1	1	0	1	0
d	1	1	1	0	0
e	0	1	0	1	1
f	1	1	0	1	1

Table 1. The boolean table relating the objects to the items in $xyzkw$. The pattern xy has support $\{a, b, c, d, f\}$. The corresponding closed pattern, obtained by intersecting the corresponding lines is also xy. The abstract support (see Figure 1) is $\{a, b, c\}$ and the abstract closed pattern is then xyk.

alence classes corresponding respectively to supports $\{a, b, c, d, f\}$ (to which belongs xy) and $\{a, b, c, f\}$ (to which belongs xyk) are merged into a class of the new equivalence relation associated to abstract supports: both xy and xyk have now the same abstract support. Now recall that each closed pattern is the maximal element of its equivalence class: this is straightforward as intersecting elements results in a greatest lower bound, and this is also true regarding the new equivalence relation as the abstract closed pattern is also obtained by intersecting a subset of object descriptions. Now, the *min-max* basis of implication rules, representing the set of $t \rightarrow q$ implications that hold on O, is obtained by considering implications $t_1 \rightarrow t_2 \backslash t_1$ where t_1 is a generator, i.e. a minimal pattern of some equivalence class, and $t_2 \neq t_1$ the corresponding closed pattern. Such a rule in our dataset is for instance $x \rightarrow y$. When considering the abstract supports, the rule still holds but a new min-max rule is now $x \rightarrow yk$. The intuitive meaning of the latter rule is then: in our dataset any object o, which belongs to of group of objects in which x occurs and that all have degree at least 2 in the induced subgraph they define, is also an occurrence of yk. Such a group is called an *abstract group*. To summarize we have obtained a new abstract knowledge, revealing a relation between patterns that depends on the connectivity of the network under study.

2 Closed patterns and abstract closed pattern

2.1 Preliminaries

Definition 1 *Let E be an ordered set and $f : E \rightarrow E$ a self map such that for any $x, y \in E$, f is monotone, i.e. $x \leq y \implies f(x) \leq f(y)$ and idempotent, i.e. $f(f(x)) = f(x)$, then:*

- If f is extensive, i.e. $f(x) \geq x$, f is called a closure operator

- If f is intensive, i.e. $f(x) \leq x$, f is called a dual closure operator or a projection.

In the first case, an element such that $x = f(x)$ is called a closed element.

We define hereunder a closure subset of an ordered set E as the range $f[E]$ of a closure operator on E, and recall a well known result on closure subsets of complete $\wedge$-semilattices.[2]

Proposition 1 *Let T be a lattice. A subset C of T is a closure subset if and only if C is closed under meet. The closure $f : T \rightarrow T$ is then defined as $f(x) = \wedge_{\{c \in C | c \geq x\}} c$ and C is a lattice.*

When the language is the power set of some set X, the meet operator simply is the intersection operator $\cap$. As a consequence, closed patterns can be searched for by performing intersection operations. We will also further need the dual proposition which states that a subset A of T is a *dual closure subset*, also denoted as an *abstraction*, whenever A is closed under joins. The projection $p : T \rightarrow T$ is then defined as $p(x) = \vee_{\{a \in A | a \leq x\}} a$, A is a lattice and $\perp$ belongs to A. In particular when T is a powerset 2^K, $p(x) = \cup_{\{a \in A | a \subseteq x\}} a$.

The standard case in which closed patterns are searched for is when the language is a lattice and that closure of a pattern relies on the occurrences of the pattern in a set of objects. In data mining the set of occurrences is known as the *support* of the pattern.

Definition 2 *Let L be a partial order and O a set of objects, a relation of occurrence on $L \times O$ is such that if $t_1 \geq t_2$ and t_1 occurs in o then t_2 occurs in o.*

The support *of t in O is defined as* $\mathrm{ext}(t) = \{o \in O \mid t \text{ occurs in } o\}$.

The cover $S(o)$ *of o is defined as the part of L whose elements occur in the object o.*

Whenever a pattern occurs in some object o then a more general pattern also occurs in o, i.e. $t_1 \geq t_2 \Rightarrow \mathrm{ext}(t_1) \subseteq \mathrm{ext}(t_2)$.

When L is a lattice, the interesting case is the one in which objects can be described as elements of L:

Proposition 2 *Let the pattern language L be a lattice and O be a set of objects. If, for any object o, the cover of o has a greatest element $d(o)$, denoted as the* description *of o in T, then for any subset e of O*

$$int(e) = \bigwedge_{o \in e} d(o)$$

is the greatest element that covers all objects of e, and is called the intension *of e, and $(\mathrm{int}, \mathrm{ext})$ is a Galois connection on $(2^O, T)$.*

Corollary 1 $\mathrm{int} \circ \mathrm{ext}$ *and* $\mathrm{ext} \circ \mathrm{int}$ *are closure operators respectively on T and 2^O and the corresponding sets of closed elements are anti-isomorphic*[3] *lattices whose related pairs (t, e) form a lattice called a* Galois lattice.

Let us consider the equivalence relation on L such that $t \equiv t'$ if and only if $\mathrm{ext}(t) = \mathrm{ext}(t')$. The maximal elements of an equivalence class associated to some support are then defined as support-closed. On the conditions of the Proposition 2, such a class has a greatest element that can be obtained from any of its elements t by applying the closure operator: $f(t) = \mathrm{int} \circ \mathrm{ext}(t)$. The support-closed elements form exactly the closure subset $f[T]$ and each of them represents the class associated to its support. In this case, f is then denoted as a *support closure* operator. In the standard case, the lattice is a powerset 2^X of attributes, the description of an object i is the subset of attributes in relation with i and the Galois lattice formed by pairs of corresponding closed elements in 2^X and 2^O ordered following 2^O is called in the FCA community *a concept lattice*[6]. In data mining, the elements of X are denoted as items and patterns are therefore itemsets. Proposition 2 follows from, for instance, Theorem 2 in [5].

The set of frequent support closed patterns, i.e. the support-closed elements with support greater than or equal to some threshold $\mathrm{minsupp}$ represents then all the equivalence classes corresponding to frequent supports. Such a class has also minimal elements, called generators. When the patterns belong to 2^X, the min-max basis of

[2] In a lattice any pair of elements (x, y) has a greatest lower bound $x \wedge y$ (or *meet*) and a least upper bound (or *join*) $x \vee y$. All ordered sets considered here are finite, and as all lattices are finite lattices they are also complete lattices: any subset of a lattice T is then closed under arbitrary meet and arbitrary join.

[3] i.e.isomorphic to the dual of $f[T]$

implication rules[11] that represents all the implications $t \rightarrow t'$ that hold on O, i.e. such that $\text{ext}(t) \subseteq \text{ext}(t')$, is defined as follows:

$m = \{g \rightarrow f \backslash g \mid f \text{ is a closed pattern }, g \text{ is a generator } f \neq g, \text{ext}(t) = \text{ext}(f)\}$

2.2 Abstract closed patterns

Projected or abstract Galois lattices have been recently defined by noticing that applying a projection operator on T [7, 12] or 2^O (or both) [12, 17] when there exists a Galois connection between them, we obtain again closure operators and lattices of closure subsets. Because of the equivalence between projections (dual closures) and abstractions mentioned above, the corresponding Galois lattices are also denoted as *abstract Galois lattices*[14].

Proposition 3 *Let* (int, ext) *be a Galois connection on* $(2^O, T)$.

- Let p *be a projection on* T*, then* $(p \circ \text{int}, \text{ext})$ *defines a Galois connection on* $((2^O, p(T))$

- Let p *be a projection on* 2^O*, then* $(\text{int}, p \circ \text{ext})$ *defines a Galois connection on* $(p(2^O), T)$

In both cases the closure subsets form a Galois lattice, respectively called intensional and extensional abstract Galois lattices.

In the remaining of this article we consider abstract closed patterns as those obtained in *extensional abstract Galois lattices*, (abstract Galois lattices for short) by constraining the space 2^O. The general idea, as proposed in [14] is that an abstract Galois lattice is obtained by selecting as an extensional space a subset A of 2^O closed under union i.e. an abstraction (or dual closure subset) and therefore such that $A = p_A(2^O)$ where p_A is a projection on 2^O. The intuitive meaning is that the abstract support $\text{ext}_A(t)$ of some pattern t will then be the greatest element of A contained in its (standard) extension, i.e. $\text{ext}_A = p_A \circ \text{ext}$ and the corresponding *abstract support closure operator with respect to* A is therefore $f_A = \text{int} \circ p_A \circ \text{ext}$.

Such an abstraction on 2^O always represent an external *a priori* information representing the user's view on the data. When the objects are categorized, for instance in a taxonomy, the categorization itself, when closed under union, forms an abstraction. In this case an object o is in the abstract support $p \circ \text{ext}(t)$ of a pattern t whenever the objects of some category containing o all belong to $\text{ext}(t)$. The main extensional abstraction that has been investigated is the alpha abstraction, which also starts from an external categorization[17]. Whenever the abstract support replaces the standard support, the inclusion order on abstract support also defines an *abstract min-max basis* with the same definition as in section 2.1 except that ext_A replaces ext.

3 Graph abstractions to investigate closed patterns when the objects form a (social) network

We consider that the set of objects O is the set of vertices of a graph $G = (O, E)$ whose edges represents some relation between objects. A vertex is labelled with an element from a language of patterns L. From now on, without loss of generality, we will consider a set of attributes (or items) X, and 2^X as the pattern language. As mentioned above we know that there exists a closure operator on 2^X such that a closed pattern is the maximal element (in the inclusion order) of the equivalent class of patterns sharing the same support. To obtain abstract closed patterns we will rely on the graph structure and will use *induced subgraphs* whose definition we recall now: the subgraph $G_{O'}$ *induced by a subset* O' of O is such that $G_{O'} = (O', E')$ where E' contains all the edges of E relating two vertices of O'.

3.1 Graph abstractions

Following the dual of proposition 1 an abstraction $A \subseteq 2^O$ is defined as a part of 2^O closed under union, i.e. $\emptyset$ belongs to A and whenever a, b are elements of A, $a \cup b$ also belongs to A. An abstraction can equivalently be obtained by considering a projection operator on 2^O and defining the abstraction as the image $p[2^O]$. This operator projects any element e of 2^O on the maximal element of A included in e. p is then defined as: $p(e) = \bigcup_{a \in A, a \subseteq e} a$ and rewrites as:

$$p(e) = \{x \in e \mid \exists a \in A \text{ s.t. } x \in a \text{ and } a \subseteq e\},$$

and e belongs to the abstraction $A = p[2^O]$ iff $e = p(e)$.

The following Lemma defines a way to build abstractions.

Lemma 1 *Let* $P : O \times 2^O \rightarrow \{\text{true}, \text{false}\}$ *be such that*

- $x \notin e$ *implies* $P(x, e)$ *is* false
- $e \subseteq e'$ *and* $P(x, e)$ *implies* $P(x, e')$

The iteration of the function q *defined as* $q(e) = \{x \in e | P(x, e)\}$ *reaches a fixed-point and the operator* p *defined as* $p(e) = \text{fixed} - \text{point}(q, e)$ *is a projection operator.* P *is then called the characteristic property of the corresponding abstraction.*

A graph abstraction will be defined through a characteristic property $P(x, e)$ which expresses some minimal connectivity requirement of the vertex x within the induced subgraph G_e. Following Lemma 1, P has to be monotone in e, i.e. if the connectivity property is satisfied in the induced subgraph G_e, it has to be still satisfied in any larger induced subgraph $G_{e'} \supseteq G_e$. This leads to a large class of graph abstractions, as for instance the $\text{degree} \geq k$-graph abstraction $A_{\text{degree} \geq k}$ that states that a subset of vertices e belongs to $A_{\text{degree} \geq k}$ whenever $d(x) \geq k$ for all x in G_e.

An *abstract group* is any subset of vertices e such that e belongs to the graph abstraction A. For instance an element of $A_{\text{degree} \geq k}$ is called a $\text{degree} \geq k$-abstract group and contains only vertices whose degree in the subgraph induced by the group is larger than or equal to k. This means that the abstract support of some pattern is the largest abstract group included in the pattern support.

We give hereunder examples of graph abstractions, defined through their characteristic property and exemplified in Figure 2.

1. $\text{degree} \geq k$ (see above and Figure 1).
2. $k\text{-clan} \geq s$: x has to belong to at least one k-clan of size at least s in G_e. This is a relaxation of the notion of clique[1]: a k-clan is a subset c of vertices such that there is a path of length $\leq k$ between any pair of vertices in G_c. A triangle, a clique of size 3, is a 1-clan of size 3 (Figure 2-a). Figure 2-b represents a 2-clan of size 6 and therefore a 2-clan$\geq$ 6 abstract group.
3. $\text{nearStar}(k, d)$: x has to have degree at least k or there must be a path of length at most d between x and some y with degree at least k. For instance, the simplest $\text{nearStar}(8, 1)$ abstract group is a central node connected with 8 nodes. Such an abstraction is useful when we want the abstraction to preserve hubs [2](i.e high degree vertices) together with their (low degree) neighbors (see Figure 2-c).
4. $cc \geq s$: x has to belong to a connected component of size at least s in G_e (see Figure 2-d).
5. $k\text{-cliqueGroup} \geq s$: x has to belong to a k-clique group of size at least s. A k-clique group is a union of k-cliques (cliques of size k) that can be reached from each other through a series of adjacent k-cliques (where adjacency means sharing k - 1 nodes).

Maximal k-cliques groups are denoted as k-cliques communities and formalize the idea of community in complex networks [10].

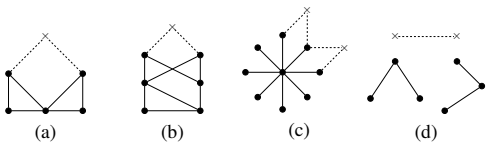

Figure 2. Graph abstractions corresponding to various vertex characteristic properties. In each graph plain circles and plain lines form the abstract subgraph, crosses and dotted lines represent the vertices and edges out of the abstract subgraph. (a) x has to belong to a triangle, (b) x has to belong to a 2-clan of size at least 6, (c) x has degree at least 8 or has to be connected to a vertex y of degree at least 8, (d) x has to belong to a connected component whose size is at least 3.

Finally, it is interesting to note that we can combine two (or more) abstractions A_1 and A_2 in two ways, defining a new composite abstraction either stronger or weaker than both A_1 and A_2. For instance, we may want to consider an abstract subgraph where vertices both have a degree larger than some k and belong to a connected component exceeding a minimal size s. On the contrary, we may want an abstract subgraph such that at least one of the two characteristic properties is satisfied by all the vertices. This would be the case for instance, if we want to keep both vertices that have a degree larger than, say 10, and vertices in a star, i.e connected to a hub which degree is at least 50. The following lemma states that we can freely combine abstractions in both directions.

Lemma 2 *Let P_1 and P_2 two characteristic properties of abstractions defined on the same object set O, and let $P_1 \wedge P_2$ and $P_1 \vee P_2$ be defined as follows:*

- $P_1 \wedge P_2(x, e) = P_1(x, e) \wedge P_2(x, e)$
- $P_1 \vee P_2(x, e) = P_1(x, e) \vee P_2(x, e)$

Both $P_1 \wedge P_2$ and $P_1 \vee P_2$ are characteristic properties of abstractions.

Finally note that requiring a frequency property also corresponds to an abstraction whose characteristic property is $P_m(x, e) = |e| \geq \text{minsupp}$, and that can be therefore combined to any abstraction, therefore defining frequent abstract closed patterns.

3.2 Graph-based closed patterns computation and analysis

When we have defined abstractions and corresponding projections, graph-based abstract closed patterns are also de facto defined. Using the projection operator p, we can compute abstract supports $p \circ \text{ext}(t)$ and abstract closures $\text{int} \circ p \circ \text{ext}(t)$. All top-down generate and close algorithms, like LCM [16] can then be adapted to direct computation of abstract closed patterns[4]. In the experiments in the next section we have used an indirect approach: we first compute frequent closed patterns and corresponding generators using the CORON software[15]. Starting from the closed patterns t and their supports, we then compute the abstract closed patterns $\text{int} \circ p \circ \text{ext}(t)$. Finally we consider for each abstract closed pattern t_A the generators of all the closed patterns that have produced t_A and select the minimal elements among them in order to obtain the corresponding abstract generators[5]. From abstract generators and abstract closed terms, computing the min-max implication rule basis is straightforward. On one hand, the indirect approach needs prior computation of the (non abstract) closed patterns, and this can be much more costly than the direct computation of abstract closed patterns. On the other hand, once this first computation is performed, we can apply as many abstract computations we need, varying graph abstractions and their parameters, and this can be cost-saving when investigating some new large attributed graph (see Section 4.3).

We describe hereunder a generic algorithm, relying on the abstraction characteristic property, to compute the projection of some subset of the set of objects O:

// Given $e \subseteq O$ and a characteristic property P
$u \leftarrow$ false
$e' \leftarrow e$
While $u =$ false
  $u \leftarrow$ true
  For all vertex x in e'
    If $P(x, e')$ is false
      $u \leftarrow$ false
      $e' \leftarrow e' - \{x\}$
    endIf
  endFor
endWhile
// As $u =$ false, $P(x, e')$ is true for all x in e'
// $e' = p(e')$ is the abstraction of e with respect to P

This generic algorithm is in $O(n^2 * d)$ where d is the cost of computing $P(x, e')$. In the graph abstraction case, computing $P(x, e')$ requires to update the induced subgraph $G_{e'}$ when some vertex is removed from e'. Furthermore, the cost d depends on the characteristic property and will be small as far as the property needs to consider only close neighbors of x. For instance, considering the degree $\geq k$ abstraction, first, there is no need to access neighbors of x, and furthermore, rather than explicitly updating $G_{e'}$ when some x is removed from e' it is more efficient to decrease the degree of the vertices connected to x in e'. Another example is the $cc \geq s$ graph abstraction, in which computing the abstraction of some e comes down to compute the connected components of G_e and to remove the small ones with no need to iterate the process.

4 Experiments

We consider here some preliminary experiments in three datasets. In all three experiments, the data is described as a graph $G = (O, E)$ whose vertices have as labels elements of 2^X where X is a set of items, i.e. binary attributes. As objects are not always described using binary attributes, the binarisation preprocessing is described when necessary. In all experiments we used *degree* $\geq k$ as the graph abstraction. We also experimented with the conjunction of degree size and connected component size in the third dataset, but we did not observe interesting results to report here.

In the three cases, we first generate the frequent closed patterns, each associated with the generators of its equivalence class, and deduce the corresponding min-max basis. We then project the frequent

[4] Work in progress

[5] Recall that each closed pattern that produces an abstract closed pattern t_A represents an equivalence class of patterns that will be included in the class of t_A in the new equivalence relation relying on abstract supports.

closed patterns to obtain the abstract closed patterns and compute the corresponding abstract generators and abstract min-max basis. We are interested in the reduction in the number of closed patterns, and in what, new and abstract, knowledge appears, when abstracting.

4.1 A simple case study

The dataset is extracted from the PhD thesis of P.N. Mougel [8] and was used to illustrate the problem of mining an attributed graph with patterns collections of dense subgraphs. The dataset represents a graph of 18 vertices (persons), connected by edges representing friendship relations. Each vertex is labelled with a subset of musical tastes among {rock, folk, pop, blues, jazz}. The graph is reproduced in Figure 3.

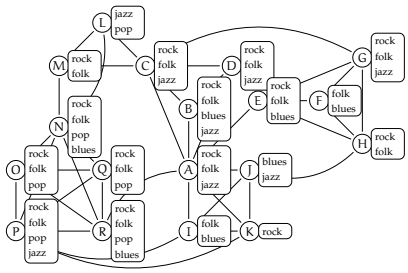

Figure 3. The labeled graph of musical tastes

Using $\text{minsupp} = 5/18$, we obtain 11 closed patterns (including the empty support pattern). When computing the min-max basis, we obtain four rules involving two closed patterns. Rules {rock, jazz} → {folk} and {folk, jazz} → {rock} are obtained from the closed pattern {rock, folk, jazz} while rules {rock, pop} → {folk} and {folk, pop} → {rock} are obtained from {rock, folk, pop}. After applying a $(\text{degree} \geq 3)$ graph abstraction, we obtain only 6 closed patterns and the following abstract rules: {rock} → {folk}, {jazz} → {rock, folk}, and {pop} → {rock, folk}. This results in a simpler view of musical tastes relying on the friendship relation. The last rule, for instance, means that any person who likes pop music and belongs to a group of friends who also like pop music, also likes rock and folk music, or more simply: a group of friends who loves pop music also love rock and folk music. The abstraction process defines what is required to be a group: with $\text{degree} \geq 3$, in a group each person has at least three friends in the group. Note that the abstraction process reduces the supports and that several equivalence classes of patterns collapse on the same abstract equivalence class. These classes are represented by the corresponding closed patterns. We report on Table 2 this collapsing process. Each line contains the abstract closed pattern (A. Patt.), its abstract support size (A. s.), the corresponding number of connected components (Cc), the closed patterns whose classes have been merged (M. Patts.), and the size of the union of corresponding supports (T. s.).

4.2 Teenage Friends and Lifestyle Study

The dataset is denoted as *s50-1* and is a standard attributed graph dataset[6]. It represents 148 friendship relations between 50 pupils of a school in the West of Scotland, and labels concern the substance use

[6] http://www.stats.ox.ac.uk/~snijders/siena/s50_data.htm

A. Patt.	A. s.	Cc	M. Patts.	T. s.
∅	18	1	∅	18
{folk}	13	3	{{ folk}}	15
{rock folk}	9	2	{{ rock, folk}, {rock}}	14
{rock, folk, jazz }	4	1	{{ rock, folk, jazz }, {jazz}}	8
{rock, folk, pop }	5	1	{{ rock, folk, pop }, {pop}}	6
All	0	0	{{ folk, blues }, {pop}}	7

Table 2. Abstract closed patterns vs standard closed patterns in the musical tastes dataset

(tobacco, cannabis and alcohol) and sporting activity. Values of the corresponding variables are ordered. The binarization process consists in defining variables representing the value intervals. T stands for Tobacco consumption and has values 1 (no smoking), 2 (occasional) and 3 (regular). C stands for cannabis consumption and has values 1 (never tries) to 4, D stands for alcohol consumption and has values 1 (does not drink) to 5, and S stands for sporting activity and has two values 1 (occasional) and (2) regular. A binary variable represents an interval, as for instance C23 that has value 1 whenever the value of C is in $[2, 3]$. For sake of simplicity we have merged the two highest values in variables T,C and D. For instance values 4 and 5 in alcohol consumption are merged into a 4m (4 and more) value. We report hereunder the binary variables whose conjunctions allow to represent any interval: for instance D=2 is obtained as {D12,D23m}.

Tobacco	Cannabis	Alcohol
T1,T2m	C1,C12,C23m,C3m	D1,D12,D123,D23m,D34m,D4m

We have computed the frequent closed patterns with minimal frequency $\text{minsupp} = 0.25$ and obtained 65 nodes and 66 (generator, closed) pairs only 15 of which led to informative min-max rules as in the other pairs the difference between generators and closed terms only relied on the binarization process. For instance the pair ({D4m, S2} , {D234m, D34m, D4m, S2}) leads to the rule {D4m, S2}) → ∅, as whenever $D = 4m$ we also have D234m and D34m. We applied then the $\text{degree} \geq 2$ graph abstraction filter resulting in 36 closed patterns resulting in also 15 informative rules. However, these abstract rules bring a considerable amount of new abstract knowledge. For instance, at the abstract level, we have the rule S1 → {C3m, D4m} which means that a group of pupils that have only occasional sporting activity also is also a group of regular cannabis and alcohol consumers. However, note that the abstract support is 3 which means that we have found a unique triangle of friends that have in common the occasional sporting activity. In fact, this is a case in which the loss in support is drastic as there are overall 13 pupils having occasional sporting activity. In other cases, the loss is much smaller, revealing groups of pupils sharing the same behaviors. For instance, the pattern {C1, D123} is observed in 28 pupils and have an abstract support of 16 and results in the abstract rule {C1, D123} → T1, i.e. a group of pupils that have never tried Cannabis and are at worst moderate alcohol consumers, also is a group of non smoker. However when adding S2, the regular sporting activity, to this behavior, there are still 21 pupils having this behavior but the abstract support is empty. As we see in Figure 4 this is because requiring S2 destroys the groups of pupils sharing {C1, D123}.

4.3 A DBLP dataset

This is the DBLP dataset as described in [3]. There is 45131 vertices, 228188 edges and 555 connected components. Vertices are authors that have published at least one paper in one among 29 journal or conference of the Database and Datamining communities[7] during the

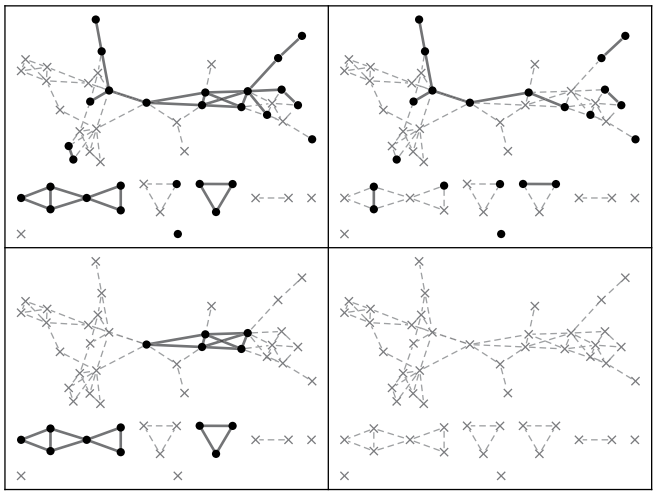

Figure 4. Subgraphs of pupils sharing a pattern.Vertices and edges of each subgraph are in plain circles and bold lines. On the left the subgraphs induced by the support (top) and degree $\geq$ 2-abstract support (bottom) of {C1,D123,T1}. On the right, the corresponding subgraphs reveal that adding S2 to the pattern removes from the standard support few vertices (top figure) but completely destroys the abstract support (bottom figure).

1/1990 to 2/2011 period. An edge links two authors whenever they are coauthors of at least one article. The conferences are clustered in three clusters: DB (databases), DM (data mining) and AI (artificial intelligence) according to a conference ranking site categorization[8].

The binary attributes are the journal and conference names together with the three clusters. An attribute has value 1 if the author has published in the corresponding journal or conference or cluster.

Using minsupp $= 1\%$ we have obtained 205 closed patterns and applied a strong abstraction filter, requiring that an author belongs to a subset of the pattern support whose induced subgraph contain only authors with at least 16 coauthors in the subgraph, i.e a very dense subgraph. As a result we found 36 closed patterns with non empty supports as 169 equivalence classes were merged in the empty support class, 21 classes were unchanged, 11 abstract classes regrouped two classes, 2 abstract classes regrouped 4 classes and 2 abstract classes regrouped 8 classes. The unique abstract rule corresponding to one of the latter abstract classes states that authors in a group of authors that have published in VLDBJ, have also published in ICDE, SIGMOD, VLDB (and therefore in a DB conference). A group here is a subset of authors all of degree at least 16 in the graph induced by the group. As a result, from the 1276 authors forming the support of the closed pattern {VLDBJ}, only 38 remains in the resulting abstract support. Among the eight classes being merged the only implication rule stated that an author that has published in VLDBJ has also published in at least one conference of the DB cluster. Again, the abstraction process has revealed some hidden knowledge at the price of drastically reducing the number of individuals on which this knowledge relies.

[7] Conferences: KDD, ICDM, ECML/PKDD, PAKDD, SIAM DM, AAAI, ICML, IJCAI, IDA, DASFAA, VLDB, CIKM, SIGMOD, PODS, ICDE, EDBT, ICDT, SAC ? Journals: IEEE TKDE, DAMI, IEEE Int. Sys., SIGKDD Exp., Comm. ACM, IDA J., KAIS, SADM, PVLDB, VLDB J., ACM TKDD

[8] http://webdocs.cs.ualberta.ca/~zaiane/htmldocs/ConfRanking.html. DB = {VLDB, SIGMOD, PODS, ICDE, ICDT, EDBT, DASFAA, CIKM}; DM= {SIGKDD Explorations, ICDM, PAKDD, ECML/PKDD, SDM}; AI= {IJCAI, AAAI, ICML, ECML/PKDD};

5 Conclusion

We have introduced the notion of graph abstraction that relies on a connectivity property and investigated the abstract closed patterns obtained by considering the corresponding notion of abstract support. Preliminary but promising experiments show the resulting reduction in the number of closed patterns as well as the kind of abstract knowledge that can be extracted. Further work includes a direct computation of abstract closed patterns, which is necessary for scalability purpose, and some investigation about the role of graph abstraction in detecting attribute based communities.

REFERENCES

[1] Balabhaskar Balasundaram, Sergiy Butenko, and Svyatoslav Trukhanov, 'Novel approaches for analyzing biological networks', *Journal of Combinatorial Optimization*, **10**, 23–39, (2005).

[2] Albert-Làszlò Barabàsi and Réka Albert, 'Emergence of scaling in random networks', *Science*, **286**(5439), 509–512, (1999).

[3] Adriana Bechara Prado, Marc Plantevit, Céline Robardet, and Jean-Francois Boulicaut, 'Mining Graph Topological Patterns: Finding Covariations among Vertex Descriptors', *IEEE Transactions on Knowledge and Data Engineering*, **25**(9), 2090–2104, (September 2013).

[4] Nathalie Caspard and Bernard Monjardet, 'The lattices of closure systems, closure operators, and implicational systems on a finite set: a survey', *Discrete Appl. Math.*, **127**(2), 241–269, (2003).

[5] Edwin Diday and Richard Emilion, 'Maximal and stochastic galois lattices', *Discrete Appl. Math.*, **127**(2), 271–284, (2003).

[6] B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical Foundations*, Springer Verlag, 1999.

[7] Bernhard Ganter and Sergei O. Kuznetsov, 'Pattern structures and their projections', *ICCS-01, LNCS*, **2120**, 129–142, (2001).

[8] Pierre Nicolas Mougel, *Finding homogeneous collections of dense subgraphs using constraint-based data mining approaches*, Ph.D. dissertation, Lyon, INSA, 2012.

[9] Pierre-Nicolas Mougel, Christophe Rigotti, and Olivier Gandrillon, 'Finding collections of k-clique percolated components in attributed graphs', in *PAKDD(2), Advances in Knowledge Discovery and Data Mining - 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29 - June 1, 2012*, volume 7302 of *Lecture Notes in Computer Science*, pp. 181–192. Springer, (2012).

[10] Gergely Palla, Imre Derenyi, Illes Farkas, and Tamas Vicsek, 'Uncovering the overlapping community structure of complex networks in nature and society.', *Nature*, **435**(7043), 814–818, (Jun 2005).

[11] Nicolas Pasquier, Rafik Taouil, Yves Bastide, Gerd Stumme, and Lotfi Lakhal, 'Generating a condensed representation for association rules', *Journal Intelligent Information Systems (JIIS)*, **24**(1), 29–60, (2005).

[12] Nathalie Pernelle, Marie-Christine Rousset, Henry Soldano, and Véronique Ventos, 'Zoom: a nested Galois lattices-based system for conceptual clustering', *J. of Experimental and Theoretical Artificial Intelligence*, **2/3**(14), 157–187, (2002).

[13] Arlei Silva, Wagner Meira, Jr., and Mohammed J. Zaki, 'Mining attribute-structure correlated patterns in large attributed graphs', *Proc. VLDB Endow.*, **5**(5), 466–477, (January 2012).

[14] Henry Soldano and Véronique Ventos, 'Abstract Concept Lattices', in *International Conference on Formal Concept Analysis (ICFCA)*, eds., P. Valtchev and R. Jäschke, volume 6628 of *LNAI*, pp. 235–250. Springer, Heidelberg, (2011).

[15] Laszlo Szathmary and Amedeo Napoli, 'Coron: A framework for levelwise itemset mining algorithms', in *Third International Conference on Formal Concept Analysis (ICFCA'05), Lens, France, Supplementary Proceedings*, eds., Bernhard Ganter, Robert Godin, and Engelbert Mephu Nguifo, pp. 110–113, (2005). Supplementary Proceedings.

[16] Takeaki Uno, Tatsuya Asai, Yuzo Uchida, and Hiroki Arimura, 'An efficient algorithm for enumerating closed patterns in transaction databases', in *Discovery Science*, pp. 16–31, (2004).

[17] Véronique Ventos and Henry Soldano, 'Alpha Galois lattices: An overview', in *International Conference on Formal Concept Analysis (ICFCA)*, eds., B. Ganter and R. Godin (Eds), volume 3403 of *Lecture Notes on Computer Science*, 298–313, Springer Verlag, (2005).