Abstract and Local Concepts in Attributed Networks

Henry Soldano[1,2], Guillaume Santini[1], and Dominique Bouthinon[1]

[1] Université Paris 13, Sorbonne Paris Cité, L.I.P.N UMR-CNRS 7030
F-93430, Villetaneuse, France
[2] Atelier de BioInformatique, ISYEB - UMR 7205 CNRS MNHN UPMC EPHE, Museum National d'Histoire Naturelle, F-75005, Paris, France

Abstract. We consider attribute pattern mining in attributed graphs through recent developments of Formal Concept Analysis. The corresponding methods restrain the extensional space 2^O, i.e. the space of possible pattern extensions in the object set O, to a subset satisfying structural properties. When considering an attributed graph, we consider its vertices as the objects under study, each described in a pattern language, as 2^I where I is an attribute set. The restriction of the extensional space depends then on the graph topology. We consider two levels. At the global level, the core idea is to reduce the extension of each pattern in such a way that the corresponding *abstract* extension induces a subgraph made of dense parts whose nodes satisfy some connectivity property. At the local level a pattern has various extensions each associated to one dense part. We obtain that way abstract closed patterns and local closed patterns, together with abstract and local implication rules. Overall, we propose here a way to extract information associated to the attributes labelling the graph vertices, according to its topology. We consider in particular the detection of communities in subgraphs of an attributed network associated to local closed patterns and local implications.

1 Introduction

We consider an attributed graph $G(O, E)$ where E is the edge set and whose vertices in O are labelled by a description in an attribute pattern language with a lattice structure, typically 2^I where I is a set of binary attributes. A way recently investigated to search for frequent patterns is to define a restricted extensional space which is the range of an *interior operator* on 2^O(see Section 3). The idea of such an operator in the case of an attributed graph is to minimally reduce a vertex subset until all the vertices of the reduced vertex subset, also called an *abstract vertex subset*, satisfies some connectivity property within the corresponding induced subgraph[1]. We call a *graph abstraction* the corresponding extensional space, i.e. the range of the interior operator mentioned above. A typical example of a connectivity property is the $\mathrm{degree} \geq k$ property which is such that all vertices of an abstract vertex subset e have a degree greater than or equal to k in the subgraph G_e induced by e. Another example is the k-clique abstraction in which vertices in e have to belong to some k-clique in G_e. This approach, based on a previous work on abstraction in Formal Concept Analysis [2] produces *abstract closed patterns* i.e. maximal attribute patterns, obtained by applying a closure operator on the *abstract support sets* obtained when applying the interior operator to the support sets[3].

[3] In data mining the extension of a pattern in a set of objects is also called its *support set*.

In this case, the set of (abstract support set, abstract closed pattern) pairs forms a lattice called an *abstract concept lattice*, and we obtain a set of related *abstract implications* denoted by $\Box q \rightarrow \Box w$ that hold whenever the abstract support set of q is included in the abstract support set of w.

Recent works in attributed graph mining are interested in searching for local patterns made of a constraint on a subset of attributes together with a density constraint on a vertex subset, and this using various notions of maximality [3,4]. In a companion article [5], we have defined *local closed patterns* corresponding to maximal attribute patterns each associated to one dense subgraph, allowing to extract *local implications*, particular to specific dense groups of objects. For that purpose Formal Concept Analysis (FCA) had to be extended in order to take into account this notion of locality. In that case, several closure operators may be applied to the same pattern: a closed pattern will then be local as the closure will depend on which region of the extensional space is concerned. The simplest example is obtained by considering that the support set of a pattern induces a subgraph made of various connected components, and associating to each connected component a local closed pattern, i.e. the most specific pattern shared by the vertices of this connected component. In what follows, the subgraph induced by the pattern q support set is simply called the *pattern q subgraph*. Formally, the dense vertex subsets we consider as elements of the extensional space form a partial order called a *confluence* in a recent investigation in Formal Concept Analysis [6] and close to, but different from, *confluent familiies* recently investigated in [7]. The confluence structure generalizes the abstraction structure and may have several minimal elements (see Section 4). In the case mentioned above, a simple *graph confluence*, the minimal elements are the singletons $\{v\}$, where v is a vertex, and the elements of the confluence are *connected vertex subsets* i.e. vertex subsets each inducing a connected subgraph. The structure of the set of (local support set, local closed pattern) pairs, we call *local concepts*, has been shown to be a more general structure generalizing the lattice structure, and called a *pre-confluence* [5]. We call *local concept pre-confluence* the ordered set of local concepts. Again we may associate to this structure a set of implications, called *local implications* written $\Box_m q \rightarrow \Box_m w$ where m is any minimal element of the confluence. Such a local implication means that the (unique) local support set including m of pattern q is included in the local support set of w including m. In the simple example mentioned above, $\Box_{\{v\}} q \rightarrow \Box_{\{v\}} w$ holds whenever i) v belongs to the support set of q and ii) the connected component containing v of the pattern q subgraph is included in the connected component containing v of the pattern w subgraph.

Both approaches can be mixed by considering the simple graph confluence F mentioned above together with a graph abstraction A. What happens then is that $F_A = F \cap A$ also is a confluence, we call a *cc*-confluence. In practice, this means that we choose some abstraction A, for instance considering the $\mathrm{degree} >= k$ graph abstraction and then consider among its elements only connected vertex subsets. When investigating some attributed graph we have then to chose A, or consider a set of graph abstractions ordered by set theoretic inclusion $A_1, \ldots, A_n$ obtained for instance by increasing k of the $\mathrm{degree} >= k$. As we will see in our experiments, we may extract this way local patterns and implications at different levels.

However connected components of abstract subgraphs as represented in cc-confluences does not always completely capture the idea of communities as considered in social network analysis. As discussed in [5], we may however enlarge the local closed patterns approach by deriving a new graph G_T from G whose vertex set T is a set of vertex subsets of G. Figure 1 displays a graph whose vertices represents pupils on a school in the West of Scotland, whose edges represent friendship relations and whose vertex attributes concern substance use and sporting activity[4]. As a running example we consider the empty pattern subgraph i.e. the whole graph. By applying a 3-clique graph abstraction we select the vertices related by colored or bold edges. We will then obtain 4 local closed patterns l_i each representing the most specific attribute pattern occurring in a connected component of the empty pattern abstract subgraph. However, in this example the largest connected component is clearly made of distinct dense parts, i.e. *communities*, we would like to consider when defining local closed patterns. Fortunately, when considering k-communities [8] we can solve this problem by applying the cc-confluence approach to a new graph derived from the original graph. More precisely, a k-community is a vertex subset in a graph G that corresponds to a connected component in a derived graph G_T. The vertices of G_T are k-cliques in G and an edge relates two vertices whenever the corresponding k-cliques share $k-1$ vertices in G. Each colored subgraph in Figure 1 defines such a 3-community. When labelling each vertex in

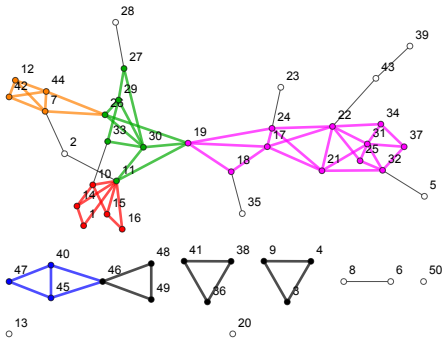

Fig. 1. The original friendship graph of a group of West Scotland pupils. The pupils and edges forming 3-communities of size at least 4 are colored. Empty vertices do not belong to the 3-clique abstraction of the graph.

G_T by the intersection of the corresponding k-clique vertex labels, the local closed pattern associated to the connected component of some pattern subgraph in G_T represents the most specific attribute pattern occurring in the corresponding k-community.

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

Section 2 describes the attributed graphs used in our experiments. Section 3 presents abstract concept lattices, abstract implications and graph abstractions. Section 4 defines local concept pre-confluences, related local implications and cc-confluences. In Section 5 we show how using derived c-confluences we extract the set of 3-communities associated to pattern subgraphs, and we display the local concept pre-confluence of the teenage friendship attributed network displayed Figure 1. In Section 6 we briefly discuss the implementation used in our experiments.

2 Datasets

We will further consider experiments in two datasets. In both cases the data is described as a graph $G = (O, E)$ whose vertices have as labels elements of 2^I where I is a set of items, i.e. binary attributes. As objects are not always described using binary attributes, the binarization preprocessing is described when necessary.

2.1 Teenage Friends and Lifestyle Study

The dataset is denoted as *s50-1* and is a standard attributed graph dataset[5]. It represents 148 friendship relations between 50 pupils of a school in the West of Scotland, and labels concern the substance use (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 attributes 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

2.2 A DBLP dataset

This is the DBLP dataset as described in [9]. 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[6]during the 1/1990 to 2/2011 period. An edge links two authors whenever they

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

[6] 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

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[7].

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.

3 Abstract closed patterns in attributed networks

3.1 Abstract closed patterns

A standard pattern mining consists in considering the set of occurrences of a pattern q, belonging to some pattern language L with a lattice structure [8], as a subset of an object set O. This language is partially ordered following a general-to-specific ordering and each object o is described as a particular pattern $d(o)$. A pattern q occurs in object o whenever $d(o)$ is less specific than q. The set of occurrences $\mathrm{ext}(q)$ of a pattern q is called its *support set* or its *extension* in O. A pattern q is said *support-closed* whenever it is a maximal pattern (i.e. maximally specific) among those that are equivalent in what they share the same support set as q. Now, whenever there is a unique support closed pattern corresponding to a given support set e, as it is the case in the standard FCA or itemset mining framework, an intension function $\mathrm{int}(e)$ returns the support-closed pattern associated to the support set e. This means that we relate a pattern q to the corresponding support closed pattern by applying the *closure operator* $\mathrm{int} \circ \mathrm{ext}$ to q. The pattern language L typically is 2^I where I is a set of binary attributes (aka items). With no loss of generality we will further use such a pattern language. The closure operator then simply intersects the object descriptions of the support set of the entry pattern.

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 implication rules[10] that represents all the implications $t \to t'$ that hold on O, i.e. such that $\mathrm{ext}(t) \subseteq \mathrm{ext}(t')$, is defined as follows:

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

We define hereunder closure operators and also *interior operators* that will be further used to restrict the support sets to be *abstract support sets*. In what follows all ordered sets are finite, and in particular any topped meet-semilattice (resp. pointed join-semilattice) is a lattice.

---

[7] 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};

[8] We recall that 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*) $x \vee y$

Definition 1 *Let U be an ordered set and $f : U \to U$ a self map such that for any $x, y \in U$, 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, an interior operator, or also a projection.

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

Ranges of interior operators on lattices are called *abstractions* and are characterized by the following Proposition:

Proposition 1 *(see [2]) A subset A of $X = 2^O$ is the range $p[X]$ of some interior operator p on X, if and only if for any elements x, y in A, their join $x \cup y$ also belongs to A and A contains the empty set. The interior operator is related to its range as follows:*

$p(x) = \sup_{\{a \in A | a \leq x\}} a$.

Let then p be the interior operator associated to some abstraction A, $p(x)$ is the greatest element of A included in x. Closed pattern analysis has been recently extended to *abstract* closed pattern analysis by noticing that applying an interior operator on the extensional space 2^O we obtain again closure operators on the pattern language 2^I [11,2]:

Proposition 2 *Let $X = 2^O$ and $L = 2^I$, p be an interior operator on 2^O, and $A = p[X]$ be the associated abstraction, we have that $(int, p \circ ext)$ is a Galois connection on (A, L), i.e.:*

$f = \text{int} \circ p \circ \text{ext}$ *is a closure operator on L,*

The *abstract support set* of pattern q is defined as $p \circ \text{ext}(q)$. There is then a unique *abstract support closed* pattern, i.e. a most specific pattern among all patterns sharing the same abstract support set, which is obtained as $f(q) = \text{int} \circ p \circ \text{ext}(q)$. $f(q)$ is then called an *abstract closed pattern*. This leads to the definition of abstract concepts organized in a concept lattice:

Corollary 1 *[2]. The set of (abstract support set, abstract closed pattern) pairs $(e = ext(c), c = int(e))$, ordered following A, is a lattice called an abstract concept lattice.*

Note that, as p is monotone, whenever $\text{ext}(q) \subseteq \text{ext}(w)$, i.e. $q \to w$ is valid we also have $\text{ext}_A(p) = p \circ \text{ext}(q) \subseteq \text{ext}_A(w) = p \circ \text{ext}(w)$, i.e. the abstract implication $\Box^A q \to \Box^A w$ is also valid.

This way we obtain *abstract min-max basis* with the same definition as in section 3.1 except that ext_A replaces ext and therefore abstract implications relate minimal elements (i.e. A-generators) to maximal element (the abstract closed pattern, or A-closed pattern) of the same abstract equivalence class. We have then the following definition:

Definition 2 *The* abstract min-max basis m_A *of valid abstract implications is defined as*

$m_A = \{\Box_A g \to \Box_A f \backslash g \mid f$ *is an A-closed pattern*, g *is a A-generator* , $f \neq g$, $\text{ext}_A(g) = \text{ext}_A(f)\}$

3.2 Graph abstractions

These ideas has been applied to attributed graphs by defining graph abstractions [1]. The set of objects O is then the set of vertices of a graph $G = (O, E)$ and each vertex o is labelled by an attribute pattern $d(o) \in 2^I$.

A graph abstraction is an abstraction of 2^O defined through a characteristic property $P(x, e)$ which expresses some minimal connectivity requirement of the vertex x within the subgraph G_e induced by some vertex subset e:

Lemma 1 *Let P be such that*

- *$P(x, e)$ implies $x \in e$ and*
- *$e \subseteq e'$ and $P(x, e)$ implies $P(x, e')$,*

and let q be a mapping defined by $q(e) = \{x \in e | P(x, e)\}$, then the mapping p defined by $p(e) = \text{fixedpoint}(q, e)$ is an interior operator on 2^O

$p(e)$ represents the greatest vertex subset of e inducing a subgraph whose vertices all satisfy the associated characteristic property. We give hereunder examples of graph abstractions, defined through their characteristic property and exemplified in Figure 2.

1. $\text{degree} \geq k$.
2. $k\text{-club} \geq s$: x has to belong to at least one k-club of size at least s in G_e. This is a relaxation of the notion of clique[12]: a k-club 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 3-clique, is a 1-club of size 3 (Figure 2-a). Figure 2-b represents a 2-club of size 6 and therefore a 2-club≥ 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 [13](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-vertex cliques that can be reached from each other through a series of adjacent k-vertex cliques (where adjacency means sharing k - 1 nodes). Maximal k-clique groups are denoted as k-cliques communities and formalize the idea of community in complex networks [14].

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.

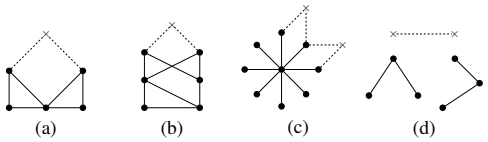

Fig. 2. Graph abstractions corresponding to various vertex characteristic properties. In each graph plain circles and plain lines form the abstract subgraph, crosses and dotted lines represent the vertices and edges out of the abstract subgraph. (a) x has to belong to a triangle, (b) x has to belong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

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.

3.3 Experiments

Some experiments on the two datasets described in Section 2 have been performed and presented in [1]. We discuss here some new details and experiments on the DBLP dataset. The experiment consisted in applying a $\mathrm{degree} \geq k$ abstraction with increasing k-values and we focussed in abstract patterns obtained with $k = 16$ which corresponds to a very strong abstraction: in an abstract support set each author is required to have 16 co-authors within the abstract support set. We obtained few abstract closed patterns and in particular the abstract closed pattern $\mathrm{VLDBJ}, \mathrm{ICDE}, \mathrm{SIGMOD}, \mathrm{VLDB}$ and the related abstract implication $\Box\ \mathrm{VLDBJ} \rightarrow \Box\ \mathrm{ICDE}, \mathrm{SIGMOD}, \mathrm{VLDB}$. Both the abstract closed pattern and its abstract generator VLDBJ have an abstract support set of 38 among the 1276 VLDBJ authors in the dataset. The implication states that a dense group of co-authors that have published in the Very Large Database Journal also have published in several database conferences. We present Figure 3 the corresponding subgraph. Such a very dense co-authoring subgraph within the VLDBJ subgraph is somewhat unexpected. We made then some investigations in the DBLP repository, focussing of these authors, and found an article whose abstract begins as follows:

> A group of senior database researchers gathers every few years to assess the state of database research ...

with the following reference:

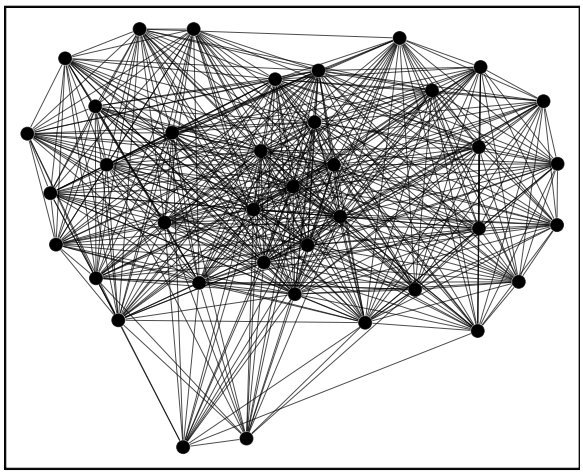

Fig. 3. The subgraph obtained when applying the $\mathrm{degree} \geq 16$ abstraction to the VLDBJ subgraph in the DBLP co-authoring experiment.

[j56] Serge Abiteboul, Rakesh Agrawal, Philip A. Bernstein, Michael J. Carey, Stefano Ceri, W. Bruce Croft, David J. DeWitt, Michael J. Franklin, Hector Garcia-Molina, Dieter Gawlick, Jim Gray, Laura M. Haas, Alon Y. Halevy, Joseph M. Hellerstein, Yannis E. Ioannidis, Martin L. Kersten, Michael J. Pazzani, Michael Lesk, David Maier, Jeffrey F. Naughton, Hans-Jörg Schek, Timos K. Sellis, Avi Silberschatz, Michael Stonebraker, Richard T. Snodgrass, Jeffrey D. Ullman, Gerhard Weikum, Jennifer Widom, Stanley B. Zdonik: **The Lowell database research self-assessment.** Commun. ACM 48(5): 111-118 (2005)

In some sense the explanation of the pattern we discovered is straightforward. However, the whole purpose of pattern mining is to find unexpected patterns, hidden within large datasets, and interpret them in order to acquire some new knowledge. It is exactly what happens here: we were not aware of these regular meetings of senior database researchers, and we learned something new, though, of course, this knowledge is clearly widely known within the database community.

When considering a weaker abstraction, namely here a $\mathrm{degree} \geq 4$ abstraction, we obtain more abstract closed patterns sometimes made of several connected components. Figure 4 represents the $\mathrm{DMKD}, \mathrm{IDArev}$ pattern subgraph together with the subgraph induced by the abstract support set of the pattern. This abstract subgraph is made of two connected components, the one in the right part of the Figure is made of 10 vertices and we are then interested in knowing whether there is some more specific pattern than the abstract closed pattern $\mathrm{DMKD}, \mathrm{IDArev}$ which would be shared by this connected component. Answering such questions means mining at a local level the attributed graph, and this is the subject of the next section.

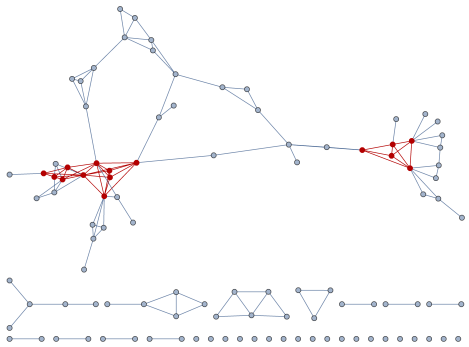

Fig. 4. The DMKD,IDArev pattern subgraph in the DBLP co-authoring experiment. The red vertices and edges represent the subgraph induced by the degree ≥ 4 abstract support set.

4 Local closed patterns in attributed networks

In [5] we introduced locality in the closure framework with as main motivation investigating local patterns in attributed graphs. We first summarize here main definitions and results. For that purpose we have to consider pre-confluences and confluences which are structures weaker than lattices investigated in [1,5]. Confluences, in particular, are close to but different from confluent families as defined in [7]. We further denote by E^x the up sets $\{y \in E | y \geq x\}$ of an ordered set E, by E_x its down sets $\{y \in E | y \leq x\}$, and by $\min(E)$ the set of its minimal elements.

First note that partial orders considered here are all finite. We first define a pre-confluence as an ordered structures that generalize the lattice structure:

Definition 3 *F is a pre-confluence if and only if for any $m \in \min(F)$, $F^m = \{x \in F \mid x \geq m\}$ is a lattice.*

A consequence of this definition is that a (finite) lattice is a pre-confluence with a minimum. The structure has a partial join operator:

Lemma 3 *For any $x, y \in F^m$ their least upper bound does not depend on m:*

1. *$x \vee_F y$ is the least element of $F^x \cap F^y$*

This means that a pre-confluence is a union of lattices in which joins coincide. A particular case is which of a pre-confluence included in a host lattice and which is join preserving:

Definition 4 *Let T be a lattice and $F \subseteq T$ be a pre-confluence with as join $\vee_F = \vee_T$, F is called a confluence of T.*

An abstraction of T, as defined above is a confluence of T with $\bot_T$ as minimum. We have then the following property when considering 2^O as the host lattice:

Proposition 3 *Let $X = 2^O$ be a lattice, $F \subseteq X$ is a confluence of X if and only if for any $x, y \in F^m$ with $m \in \min(F)$, we have that $x \cup y$ belongs to F.*

A confluence, is then associated to a set of interior operators:

Proposition 4 *Let F be a confluence of a lattice X, and $m \in \min(F)$,*

- *$p_m : X^m \to X^m$, such that $p_m(x) = \vee_{q \in F^m \cap X_x} q$, is an interior operator and $p_m[X^m] = F^m$.*

We are concerned here with extensional confluences, i.e. confluences of $X = 2^O$[5] that generalize extensional abstractions as graph abstractions. In this case, let x be an element of X greater than or equal to some minimal element m of F, then $p_m(x)$ returns the greatest subset of x in F that includes m. In the example that follows we define a graph confluence by only considering vertex subsets inducing connected subgraphs of some graph $G = (O, E)$.

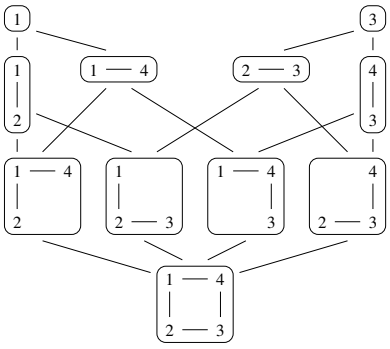

Fig. 5. A square graph (in the bottom of the figure) and the Hass diagram of the confluence F^{1+3} of connected vertex subsets that contain vertices 1 or 3.

Example 1. Let $G = (O, E)$ be a graph (displayed at the bottom of Figure 5) whose vertex set is $O = \{1, 2, 3, 4\}$. Let $F \subseteq 2^O$ be the set of vertex subsets inducing connected subgraphs of G. We call them connected vertex subsets. F is a confluence whose set of minimal elements is $\min(F) = \{\{1\}, \{2\}, \{3\}, \{4\}\}$, i.e. the set of singletons of 2^O. The union of two connected vertex subsets that each contains a given vertex s obviously also is a connected vertex subsets and therefore F is a confluence of 2^O. In what follows, by abuse of notation we write p_s and F^s rather than $p_{\{s\}}$ and $F^{\{s\}}$.The interior operator p_s projects then any vertex subset e containing vertex s on the connected component of the subgraph G_e induced by e that contains s. The up set F^s is then the set of connected vertex subsets containing s and the union of all these F^s represents the whole set of connected subgraphs of G. The subset $F^{1+3} = F^1 \cup F^3$ representing connected vertex subsets containing vertices 1 or 3 also is a confluence. Figure 5 displays the diagram of F^{1+3}. □

The support set e of a pattern q may then be projected, through interior operators, on various smaller *local support sets* $\{e_i\}$ corresponding, in the graph confluence case, to the connected components of the pattern subgraph. These interior operators are associated to *local closure operators*[5]:

Proposition 5 *Let F be a confluence of $X = 2^O$, m a minimal element of F and $L_{int(m)}$ be the down set of the pattern lattice L whose elements q are such that $q \geq \text{int}(m)$, then*

$f_m = \text{int} \circ p_m \circ \text{ext}$ *is a closure operator on* $L_{int(m)}$

In the graph confluence case, let $e = \text{ext}(q)$, $p_s(e)$ is the connected component of the pattern subgraph G_e to which belongs the vertex s. Obviously $p_s(e) = p_v(e)$ for any vertex v in the same connected component. Therefore $f_s(q)$ is a *local closed pattern* w.r.t. any vertex in this connected component, i.e. the most specific pattern shared by the vertices in the connected component.

Now an important result is that that the set of local support sets is a pre-confluence:

Theorem 1. *The mapping $h : F \to F : h(e) = p_m \circ \text{ext} \circ \text{int}(e)$ for $m \leq e$ is a closure operator on F and $E = h[F]$ is a pre-confluence.*

$h(e)$ is therefore the local support set of $\text{int}(e)$ that contains $m \leq e$. $h[F]$ is a pre-confluence isomorphic to the set P of *local concept pairs* defined as follows:

Definition 5 *The set of local concept pairs $P = \{(e, l) \mid e = p_m \circ \text{ext}(l), l = \text{int}(e), m \leq e\}$ is called a* local concept pre-confluence.

To summarize we have defined local concepts as (local support set, local closed patterns) pairs and shown they were organized in a structure with possibly several minimal elements, therefore generalizing the concept lattice definition. In the simple graph confluence exemplified above the local support sets simply are the connected components of the pattern subgraphs. We will now extend graph confluences by intersecting this simple graph confluence with abstractions.

4.1 Cc-confluences

We remark now that we can freely intersect confluences:

Proposition 6 *Let F_1 and F_2 be confluences of X, then $F_1 \cap F_2$ is a confluence*

And as abstractions of X are confluences of X with the bottom element of X as their unique minimal element, the above proposition means we can freely intersect abstractions with confluences to build smaller confluences. Many confluences can then be derived from a graph confluence by intersecting it with some abstractions. We call this family of confluences the *cc-confluences*. For instance, considering A as the k-clique abstraction, we obtain the *cc*-confluence of connected subgraphs of G made of k-cliques. Note that *cc*-confluences have an important property: rather than considering the minimal elements of F when defining local closure operators we can consider

vertices. This is because given a vertex subset v in some pattern subgraph, all minimal elements containing v belong to the same connected component as v, and therefore the local support sets are the same. This is computationally important as this means that when considering local support sets we only need to consider each vertex in the support set and associate it to the connected component to which it belongs.

4.2 Local implications

Inclusion of local support sets define local implication rules $\Box_m q \rightarrow \Box_m w$ where m is a minimal element of F_A in the support set of q. Note that, as the local support set of pattern q is obtained by applying an interior operator, which is monotone, to the support set of q, we have that whenever $\Box q \rightarrow \Box w$ is valid and $m \subseteq \mathrm{ext}_A(w)$, we also have that $\Box_m q \rightarrow \Box_m w$ is valid, i.e. we may infer the latter local rule from the former abstract rule.

We search now for a basis B of valid local implication rules from which we may infer any local implication rules. We will consider a basis B_m for a given minimal element m of F and obtain the whole basis $B = \cup B_m$ by joining these bases. Consider a given abstract closed pattern c whose abstract support set has a connected component that contains m, and let $l = f_m(c)$ be the corresponding local closed pattern, with respect to the cc-confluence F_A. We have then that the implication rule $\Box_m c \rightarrow \Box_m l$ holds. We select then a basis B_m of *informative* ($l \neq c$) and *irredundant* (there is no other rule $\Box_m c' \rightarrow \Box_m l$ with c' less specific than c in the rule set) ones. From B_m we may infer all local implication rules associated to m by applying standard axioms in the same way as in the case of the *min-max basis* in the standard closed or abstract framework. The basis $B = \cup B_m$ represents the local knowledge deriving from the reduction of the extensional space from abstraction A to cc-confluence F_A.

4.3 Experiments on cc-confluences

To shortly examplifiy local implications and local concepts we come back to our experiments on the DBLP dataset in Section 3.3 and specifically the abstract closed pattern $\mathrm{DMKD, IDArev}$, with respect to the $\mathrm{degree} \geq 4$ abstraction, whose abstract support set is represented in red on Figure 4. When considering the connected component on the left of FIgure 4, we obtain the local implication DMKD,IDArev $\rightarrow_{268924}$ DMgroup stating that in the connected component of the abstract subgraph to which belongs the author 268924, each author has also published in some data mining conference belonging to $\mathrm{DMgroup}$.

Now, when considering the Teenage Friends attributed graph displayed Figure 1, clearly the friendship relations are organized in 3-cliques, therefore any stronger abstraction will be poorly informative. However, as mentioned in Section 1, when considering the 3-clique abstract graph associated to the empty pattern the unique connected component could be separated in several (overlapping) communities (displayed in various colors). We discuss and exemplify in the next section how to apply the local closure strategy to discover such subcommunities in an attributed graph.

5 Derived cc-graph confluences

In what follows, we will consider a family $T \subseteq 2^O$ of vertex subsets, and consider T as the vertex set of a graph $G_T = (T, E_T)$ *derived* from G. The simple graph confluence F of 2^T is then the new extensional space and we will search for the corresponding local closed patterns. The local support sets are afterwards transformed into support sets in 2^O. Let $u : 2^T \rightarrow 2^O$ be such that $u(e_T) = \cup_{t \in e_T} t$. $u(e_T)$ is called the *flattening* of e_T. We consider then the two maps ext_T and $\mathrm{int_T}$ defined as follows:

- $\mathrm{ext}_T : L \rightarrow 2^T$ with $\mathrm{ext}_T(q) = \{t | t \subseteq \mathrm{ext}(q)\}$
- $\mathrm{int}_T : 2^T \rightarrow L$ with $\mathrm{int}_T(e_T) = \mathrm{int} \circ u(e_T)$

$\mathrm{ext}_T(q)$ represents the support set of q in T when considering that q occurs in t whenever q occurs in all elements of t (seen as a subset of O). Conversely $\mathrm{int}_T(e_T)$ represents the greatest pattern in L whose support set in T includes e_T, i.e. whose support set in O contains, as subsets, the elements of e_T. Now, consider as T the family of k-cliques of G and that $(t_1, t_2) \in E_T$ whenever t_1 and t_2 share $k-1$ vertices in G. A k-community in G [8] is a vertex subset that results from the flattening (in the sens defined above) of some connected component of G_T. The local closed patterns w.r.t. F are then most specific patterns occurring in k-communities of pattern subgraphs of G. This way we obtain a local concept pre-confluence, and associated local implications, whose local support sets are these k-communities.

5.1 Experiments on derived *cc*-confluences

Coming back to our Teenage Friendship attributed graph, we have applied this strategy and built the derived graph G_T where T is the set of 3-cliques of the original attributed graph. We display Figures 6 and 7 the local concept pre-confluence of 3-communities which size is greater than or equal to 4 members[9]. The minimal 3-communities are the lowest ones on both Figures. Each element of the pre-confluence represents a (3-community, local closed pattern) pair but may be associated to several non redundant local implications. This happens for one 3-community displayed on the right at the bottom of Figure 6 and associated to two local implications each represented in a square. Each square displays in red the 3-community, and in red+green+blue the abstract support set of the abstract closed pattern forming the left part of the implication. In Figure 7 we have a unique maximal 3-community on the top, and a hierarchy of sub-communities.

6 Implementation

In our experiments we use the CORON software[15] to compute frequent closed patterns, according to some frequency threshold, then apply a set of PYTHON functions

[9] Formally, this means that we also apply to the derived graph an abstraction to avoid connected components corresponding to 3-communities smaller than 4 members.

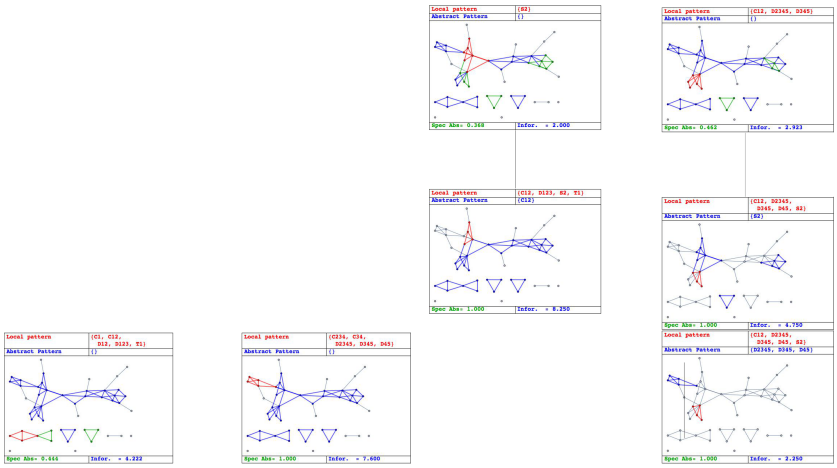

Fig. 6. The pre-confluence of size ≥ 4 3-communities of the Teenage Friendship graph(part-I)

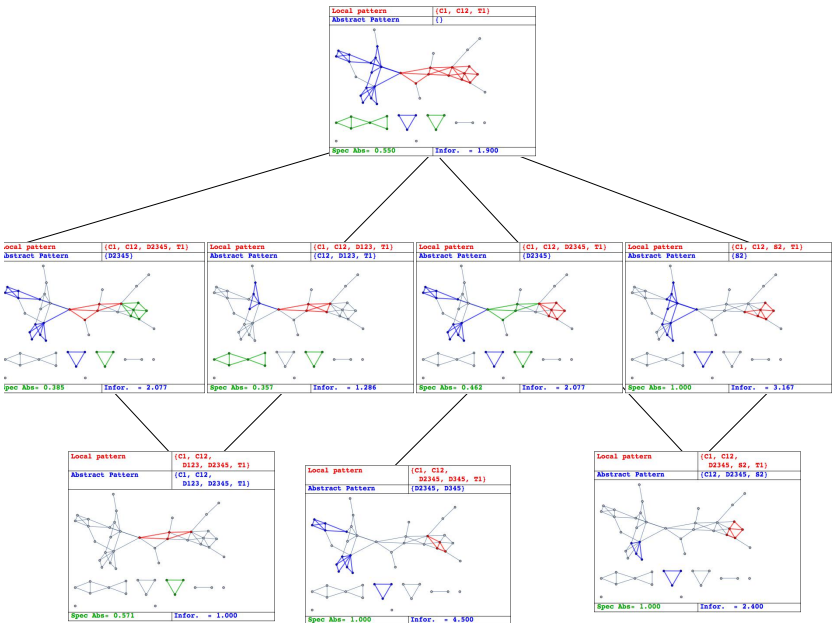

Fig. 7. The pre-confluence of size ≥ 4 3-communities of the Teenage Friendship graph(part-II)

as a post processing[10]. Starting from the set of frequent (possibly abstract) closed patterns C we then compute for each such pattern $c \in C$ the subgraph induced by its (abstract) support set, extract the various connected components $\{e_1, ..., e_k\}$ that are large enough, compute the corresponding local closed patterns $\{c_1, ..., c_k\}$ and output the corresponding local implications. When computing k-communities, we start from the k-clique graph abstract closed and build the k-clique graph G_T, where T is the set of k-cliques in G, compute the local closures corresponding to the subgraphs of G_T induced by the connected components of our abstract closed patterns, and output the corresponding triples, where the local support sets are flattened to be expressed as subsets of O.

In a work in progress, we consider a more efficient computation of abstract and local closed patterns and related implications. The corresponding algorithm uses a divide and conquer strategy similar to that proposed in [7], and therefore allows to directly apply the frequency constraints at the abstract and local level.

7 Conclusion

In the present article we are interested in addressing problems in which the extensional space, made of the vertex subsets of an attributed network, is constrained according to connectivity properties. We have first considered abstract vertex subsets in which a constraint have to be satisfied by each vertex in the subgraph they induce, as for instance a minimum degree constraint. The extensional space is in this case a particular lattice called an abstraction. We have then shown, benefiting from previous work in FCA, how abstract support closed patterns, i.e. maximal patterns among those sharing the same abstract support set, could be obtained using a closure operator. This led to define a wide class of abstract concept lattices, whose elements are (abstract support set, abstract closed pattern) pairs, each corresponding to a particular abstraction. We obtain this way a global information on how the graph topology is related to the patterns support sets. We have then considered a way to extract local knowledge from an attributed network. For that purpose, using a recent extension of FCA to local extensional spaces, called confluences, we have related each pattern to various local support sets, corresponding to connected components in subgraphs induced by abstract vertex subsets. We obtain this way a set of local concepts, organized in a generalization of the lattice structure called a pre-confluence. Furthermore we have defined both abstract implications and local implications rules representing knowledge which is valid at the abstract and local levels, i.e., regarding the latter, in the vicinity of particular vertices. Finally we have applied these ideas to define the pre-confluence of 3-communities in a social network. These 3-communities are in fact sub-communities as each is a 3-community in some subnetwork induced by an attribute pattern. Overall, what we propose here is a new way, brought by recent developments in Formal Concept Analysis, to explore social networks as attributed graphs. Future works concerns, on the extensional side, applying these ideas to attributed directed graphs or multiplex networks. We also consider to use abstract and local extensional constraints while extending the pattern language to a

[10] The corresponding software is to be found in https://lipn.univ-paris13.fr/~santini/ .

wider class of pattern languages. First, as in [16,11,17] by building a meet-semilattice adatpted to the mining problem and using interior operators to reduce it to a tractable language. This has been in particular successfully applied to graph mining [18].Then, as in [6,7] by considering confluent languages allowing to consider connectivity within the pattern language.

References

1. Soldano, H., Santini, G.: Graph abstraction for closed pattern mining in attributed network. In Schaub, T., Friedrich, G., O'Sullivan, B., eds.: European Conference in Artificial Intelligence (ECAI). Volume 263 of Frontiers in Artificial Intelligence and Applications., IOS Press (2014) 849–854
2. Soldano, H., Ventos, V.: Abstract Concept Lattices. In Valtchev, P., Jäschke, R., eds.: International Conference on Formal Concept Analysis (ICFCA). Volume 6628 of LNAI., Springer, Heidelberg (2011) 235–250
3. Mougel, P.N., Rigotti, C., Gandrillon, O.: 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., Springer (2012) 181–192
4. Silva, A., Meira, Jr., W., Zaki, M.J.: Mining attribute-structure correlated patterns in large attributed graphs. Proc. VLDB Endow. **5**(5) (January 2012) 466–477
5. Soldano, H.: Extensional confluences and local closure operators. In Baixeries, J., Sacarea, C., Ojeda-Aciego, M., eds.: Formal Concept Analysis - 13th International Conference, ICFCA 2015, Nerja, Spain, June 23-26, 2015, Proceedings. Volume 9113 of Lecture Notes in Computer Science., Springer (2015) 128–144
6. Soldano, H.: Closed patterns and abstraction beyond lattices. In Glodeanu, C.V., Kaytoue, M., Sacarea, C., eds.: Formal Concept Analysis 12th International Conference, ICFCA 2014, Cluj-Napoca, Romania, June 10-13. Volume 8478 of Lecture Notes in Computer Science., Springer (2014) 203–218
7. Boley, M., Horváth, T., Poigné, A., Wrobel, S.: Listing closed sets of strongly accessible set systems with applications to data mining. Theor. Comput. Sci. **411**(3) (2010) 691–700
8. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043) (Jun 2005) 814–818
9. Bechara Prado, A., Plantevit, M., Robardet, C., Boulicaut, J.F.: Mining Graph Topological Patterns: Finding Co-variations among Vertex Descriptors. IEEE Transactions on Knowledge and Data Engineering **25**(9) (September 2013) 2090–2104
10. Pasquier, N., Taouil, R., Bastide, Y., Stumme, G., Lakhal, L.: Generating a condensed representation for association rules. Journal Intelligent Information Systems (JIIS) **24**(1) (2005) 29–60
11. Pernelle, N., Rousset, M.C., Soldano, H., Ventos, V.: Zoom: a nested Galois lattices-based system for conceptual clustering. J. of Experimental and Theoretical Artificial Intelligence **2/3**(14) (2002) 157–187
12. Balasundaram, B., Butenko, S., Trukhanov, S.: Novel approaches for analyzing biological networks. Journal of Combinatorial Optimization **10** (2005) 23–39
13. Barabàsi, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**(5439) (1999) 509–512
14. Palla, G., Derenyi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043) (Jun 2005) 814–818

15. Szathmary, L., Napoli, A.: Coron: A framework for levelwise itemset mining algorithms. In Ganter, B., Godin, R., Nguifo, E.M., eds.: Third International Conference on Formal Concept Analysis (ICFCA'05), Lens, France, Supplementary Proceedings. (2005) 110–113 Supplementary Proceedings.
16. Ganter, B., Kuznetsov, S.O.: Pattern structures and their projections. ICCS-01, LNCS **2120** (2001) 129–142
17. Ferré, S., Ridoux, O.: An introduction to logical information systems. Information Processing and Management **40**(3) (2004) 383–419
18. Kuznetsov, S.O., Samokhin, M.V.: Learning closed sets of labeled graphs for chemical applications. In Kramer, S., Pfahringer, B., eds.: ILP. Volume 3625 of Lecture Notes in Computer Science., Springer (2005) 190–208