Efficient Pattern Mining in Attributed Graphs

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

1. *Université Paris 13, Sorbonne Paris Cité, L.I.P.N. UMR-CNRS 7030*
   F-93430, Villetaneuse, France
   henry.soldano,guillaume.santini,db@lipn.univ-paris13.fr
2. *Museum National d'Histoire Naturelle, ABI, ISYEB - UMR 7205 CNRS MNHN*
   UPMC EPHE, F-75005, Paris, France
   henry.soldano@mnhn.fr

RÉSUMÉ. *Nous nous intéressons à la découverte de motifs dans un graphe dont les sommets sont étiquetés dans un langage de motifs partiellement ordonné. L'idée générale de ce travail est de réduire le sous-graphe dont les sommets ont un certain motif de manière à ce qu'une certaine propriété soit satisfaite par les sommets ou les arêtes du sous-graphe réduit. Ce sous-graphe réduit est appelé le* coeur *(core) du sous-graphe. On appellera ici* communautés *les composantes connexes de ces coeurs. Chaque coeur ou communauté est associé au motif le plus spécifique (le motif fermé) apparaissant dans ses sommets, une communauté étant aussi associé à des règles locales exprimant ce qui singularise celle-ci dans le sous-graphe dont elle provient. Nous proposons deux algorithmes, le premier pour énumérer les coeurs fréquents et les motifs fermés associés, le second pour énumérer les communautés fréquentes et les règles locales associées.*

ABSTRACT. *We consider mining undirected graphs whose vertices are labelled by partially ordered patterns, as labelled social networks. The general idea is to reduce the subgraph in which some pattern occur to its core, according to some core property, then consider as communities the connected components of this core. To each core or community is associated the most specific pattern that occurs in its vertex subset. This leads to closed pattern mining problems for which we propose efficient algorithms. The first algorithm enumerates all frequent (core closed pattern, core subgraph) pairs while the second enumerates (core closed pattern, community, community closed pattern) triples representing local implications that hold within the related community. We experiment the resulting program* minerLC *on various kind of core properties and datasets of various sizes.*

MOTS-CLÉS : *graphes attribués, fouille de données, motifs fermés, analyse formelle de concepts*

KEYWORDS: *attributed networks, core subgraphs, closed pattern mining, formal concept analysis*

DOI:10.3166/RIA.0.1-**??** © 2016 Lavoisier

1. Introduction

For some years there are a lot of interest in investigating complex networks, as regulatory networks, or social networks as co-authoring networks, with the purpose of discovering and modeling their structure (Palla *et al.*, 2005). In particular, many methods have been developed to extract from such networks cohesive and dense subnetworks, called communities. There are two directions in works about extracting communities. The first is reminiscent of clustering methods and tends to divide the network in an optimal way, according to some measure, in communities in which vertices are highly connected and which have few links to other communities (Flake *et al.*, 2000). The second is based on core decomposition (Palla *et al.*, 2005 ; Seidman, 1983) and consists in first reducing the graph to a subgraph satisfying some connectivity property and then extracting communities as connected components of this subgraph (Seidman, 1983 ; Saito *et al.*, 2009) resulting respectively in k-core communities and k-dense communities. In the latter approach, the communities can also be extracted from a graph derived from the original graph resulting in more cohesive k-clique communities (Palla *et al.*, 2005) that demonstrate to be an adequate representation for various complex networks.

More recently, complex networks analysis has been extended to take into account information provided as labels about vertices or edges. The network is then called a *labelled* or *attributed* network and may be subjected to constraints about the attribute information. Various work have then been developed to mine and rank patterns as pairs of constraints on topology and labels (Mougel *et al.*, 2012 ; Silva *et al.*, 2012 ; Galbrun *et al.*, 2014). Recently a new approach has been presented extending the *closed pattern mining* methodology to attributed graphs by considering the vertices as transactions and reducing the support set of the attributed patterns using *abstraction operators* (Soldano, Santini, 2014). Such an operator reduces the vertex set of some subgraph G' to a subset inducing the core subgraph of G' according to some vertex core property. Equivalent vertex core functions, together with efficient algorithms, were previously defined to extend the k-core definition proposed in (Seidman, 1983) to a class of generalized cores (Batagelj, Zaversnik, 2011) but were not used to investigate labelled graphs. Graph abstractions were then further extended to *graph confluences* in order to investigate the set of k-cliques communities in subgraphs induced by attribute patterns (Soldano *et al.*, 2015).

In the present article we consider vertex cores, following (Batagelj, Zaversnik, 2011), expressed through *vertex core properties* together with edge cores, that generalize the k-dense edge core. Edge cores are subnetworks whose edges satisfy some property. In both cases, to any attribute pattern q is associated a q core subgraph together with the most specific pattern shared by its vertices and that is called a *core closed pattern*. We first consider mining frequent closed patterns and related cores, given some core property. Then we consider connected components of such q cores as communities to each of which we associate a *local closed pattern*, i.e. the most specific pattern shared by the vertices of the community. We then mine frequent local closed patterns each associated to a core closed pattern and a community, and extract from them a set of *local implications rules*. We propose efficient algorithms for these two tasks and introduce the k-nearstar, a vertex core property weaker than the k-core property and whose purpose is to focus on *hubs* which have important applications in complex networks (see for instance (Lareau *et al.*, 2015)).

2. Closed pattern mining

The standard closed pattern mining framework considers the occurrences of a pattern q, belonging to some pattern language L, within an object set V. The language is partially ordered following a general-to-specific ordering, and each object v is described as a pattern $d(v)$. In the itemset mining framework, $L = 2^I$ where I is a set of binary attributes (aka items) and pattern q occurs in object v whenever $q \subseteq d(v)$ i.e. $d(v)$ is more specific than q. The set of occurrences $\mathrm{ext}(q)$ of a pattern q is called its *support set* in V. A pattern q is said *support-closed* whenever it is a maximal pattern (i.e. it is maximally specific) among those that are equivalent in what they share the same support set. Considering only support closed patterns leads to a condensed representation of all patterns, which is useful when mining patterns according to a frequency criteria. When the pattern language is a lattice, as in the itemset mining framework, there is a unique support closed pattern associated to a given support set. This means that we can relate a pattern q to $f(q)$, the most specific pattern with same support set $\mathrm{ext}(q) = V'$. The operator f turns to be a closure operator(see above) and is obtained as $f(q) = \mathrm{int} \circ \mathrm{ext}(q)$ where the int operator is such that $\mathrm{int}(V')$ returns the most specific pattern occurring in V'. In the itemset mining case, the closure operator simply intersect the object descriptions of the support set of the entry pattern, i.e. $\mathrm{int}(V') = \cap_{v \in V'} d(v)$. We define hereunder closure operators and also *interior operators* that will be further used to restrict the support sets and define vertex subset of graph cores.

DÉFINITION 1. — *Let S be an ordered set and $f : S \to S$ a self map such that for any $x, y \in S$, 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 $f(x) \geq x$, then f is called a* closure operator *while if $f(x) \leq x$, then f is called an* interior operator.

Whenever f is a closure operator on a pattern language and $q = f(q)$, q is called a closed pattern. Closed pattern analysis has been recently extended to *abstract* closed pattern analysis by noticing that applying an interior operator on the object space 2^V we obtain again closure operators (Pernelle *et al.*, 2002):

PROPOSITION 2. — *Let p be an interior operator on 2^V, then $f = \mathrm{int} \circ p \circ \mathrm{ext}$ is a closure operator on L.*

$p \circ \mathrm{ext}(q)$ is then the *abstract support set* of q and $f(q)$ the associated *abstract closed pattern*.

There is obviously less abstract closed patterns than closed patterns, as two patterns q and q' with same support set may share the same abstract support set, leading to a more condensed representation of patterns occurring in V. In what follows we consider as the object set the vertex set of some graph (V, E) and core properties will lead to interior operators.

3. Cores and Structural Communities in an Attributed Graph

3.1. Graph cores

Let $G = (E, V)$ be a graph. we define the edge function es by $\mathrm{es}(V') = \{(x, y) \in E \mid x, y \in V' \subseteq V\}$, i.e. $\mathrm{es}(V'))$ contains all the edges that have both their vertices in V'. The subgraph induced by the vertex subset $V' \subseteq V$ is then $(V', \mathrm{es}(V')$. In the same way, we define the vertex function vs by $\mathrm{vs}(E') = \{x \in V \mid (x, y) \in E'\}$, i.e. $\mathrm{vs}(E')$ contains all the vertices belonging to some edge in E'. The subgraph induced by the edge subset $E' \subseteq E$ is then $(\mathrm{vs}(E'), E')$.

We will define vertex properties as $P : V \times 2^V \to \{\mathrm{true}, \mathrm{false}\}$ mappings where $P(v, V')$ is true whenever vertex x satisfies some condition within the $(V', \mathrm{es}(V'))$ subgraph, and similarly define edge properties as $P^E : E \times 2^E \to \{\mathrm{true}, \mathrm{false}\}$ mappings where $P^E(e, E')$ is true whenever edge e satisfies some condition within the subgraph $(\mathrm{vs}(E'), E')$. We then say that these subgraphs have respectively property P and P^E.

The core subgraph of a graph (V, E) will be defined either as its *vertex P-core* i.e. the largest subgraph $(V', \mathrm{es}(V'))$ which has vertex property P or as its edge P^E-core, i.e. the largest subgraph $(\mathrm{vs}(E'), E')$ which has edge property P^E.

EXEMPLE 3. — The k-core of a graph, is the vertex core $G_c = (V_c, \mathrm{es}(V_c))$ whose vertex property states that $P(v, X)$ is true whenever the degree of v in $(X, \mathrm{es}(X))$ is greater than or equal to k (Seidman, 1983). □

In (Batagelj, Zaversnik, 2011) vertex cores were associated to vertex properties functions on $\mathbf{N}$ while in (Soldano, Santini, 2014) vertex properties are boolean as here. Both definitions results in the same class of vertex cores. Edge cores are *fully generalized cores* as defined in (Francisco, Oliveira, 2010) though in the latter work cores were defined through a family of subgraphs.

EXEMPLE 4. — The k-dense is an edge core defined in (Saito *et al.*, 2009) as the maximal subgraph in which $P^E(\{x, y\}, E')$ is true, i.e. in which each edge $\{v, v'\}$ is such that v and v' have at least $k - 2$ common neighbors.

□

3.2. Subgraph cores

To define a vertex (resp. edge) core, we need the vertex (resp. edge) property to be such that for any subgraph there does exist a unique maximal subgraph of it which has the property. In what follows we write X standing for either V or E and define accordingly *core properties*:

DÉFINITION 5. — *A property P such that for any $X' \subseteq X$ there is a unique maximal subset $X'' = p(X')$ of X' such that $P(x, X'')$ is true for all elements x in X'', is called a* core property *of G.*

The p operator associated to P is called is called a *core operator*. When $X = V$, p is a *vertex core operator*, while when $X = E$, we rather write p^E the corresponding *edge core operator*. Obviously not all properties are core properties. A core operator exists whenever the core property is *monotone* i.e. for any $x \in X_1 \subseteq X$ we have that $P(x, X_1)$ and $X_2 \supseteq X_1$ implies $P(x, X_2)$.

PROPOSITION 6. — *If a property P is monotone, then P is a core property.*

The vertex k-core property of Example 3 is obviously monotone: the degree of node v in a graph cannot decrease when adding vertices and induced edges to the graph. In the same way, the common neighborhood of the two vertices forming some edge e in a graph cannot decrease when adding edges to the graphe, and as a consequence the edge k-dense core property of Example 4 is also monotone. A generic algorithm for computing the core vertex subset $p(V)$ of a graph (V, E) in the monotone case has been given in (Batagelj, Zaversnik, 2011). The following algorithm returns the core subgraph according to either a vertex or an edge property P:

core((V,E), P)
(V, E) is a graph and P a vertex (resp. edge)
monotone property
  $C \leftarrow V$ // (resp. $C \leftarrow E$)
  While $\exists x \in C$ such that $P(x, C)$ is false **do**
    $C \leftarrow C \setminus \{x\}$
  endWhile
  Return $(C, \mathrm{es}(C))$ // (resp. $(\mathrm{vs}(C), C)$)

We have then the following property that allows to associate closed patterns to vertex cores:

PROPOSITION 7. — *A core operator is an interior operator on 2^X.*

As a result, according to Proposition 2, f defined as $f(q) = \mathrm{int} \circ p \circ \mathrm{ext}(q)$ is the closure operator returning the most specific pattern shared by all vertices of the vertex core of the pattern q subgraph of G. This pattern is called a *vertex core closed pattern*. To obtain a closure operator associated to edge core subgraphs we need to go back and forth from vertex subsets to edge subsets:

PROPOSITION 8. — *Let p^E an edge core operator, then the vertex operator defined by $\forall V' \subseteq V, p(V') = \mathrm{vs} \circ p^E \circ \mathrm{es}(V')$ is an interior operator on 2^V.*

Therefore f defined as $f(q) = \mathrm{int} \circ p \circ \mathrm{ext}(q)$ is also a closure operator that returns the most specific pattern shared by all vertices found in the edge core of the pattern q subgraph of G. This pattern is called an *edge core closed pattern*. In what follows the core of a pattern q subgraph is simply called a q core, and $f(q)$ is the most specific pattern shared by the vertices of a q core.

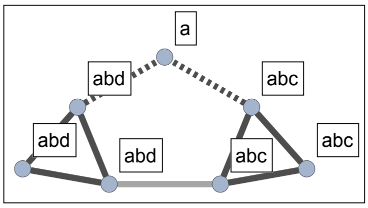

FIGURE *1. A 2-core (whole graph). The 3-dense edge core is made of the two triangles (bold edges) while the associated 3-dense vertex core is obtained by joining the two triangles by the grey edge*

3.3. Structural communities

Cores are not necessarily connected, while the notion of a community requires that their members are connected in some way. Following (Saito *et al.*, 2009) we define hereunder a *structural community* on a graph, or a *community* for short, as a connected component of its core. To each such community we may associate the most specific pattern shared by its vertices. Such patterns are technically *local closed patterns* as defined in (Soldano *et al.*, 2015), and to any pattern q, and any vertex v in its q core, we may associate $f_v(q)$ which is the most specific pattern shared by the vertices of the community that contains v. Technically, f_v is a closure operator defined as $f_v(q) = \mathrm{int} \circ p_v \circ \mathrm{ext}(q)$ where p_v is an interior operator (Soldano *et al.*, 2015).

When the core is a vertex core, p_v reduces a set of vertices V' to the vertices of the connected component containing v of the vertex core $(p(V'), \mathrm{es}(p(V')))$. When the core is an edge core, p_v reduces a set of vertices V' to the vertices of the connected component containing the vertex v of the edge core $(\mathrm{vs} \circ p^E \circ \mathrm{es}(V'), p^E \circ \mathrm{es}(V'))$. In both cases, the communities of a pattern q subgraph, called pattern q communities, are related to the most specific pattern $f_v(q)$ shared by the members of the community.

EXEMPLE 9. — Following the example of Figure 1 we consider now the two communities associated to the 3-dense edge core.Their associated closed pattern are respectively abd (left triangle) and abc (right triangle). The associated 3-dense vertex core has only one community with closed pattern ab.

□

Given a graph G and a core property P, inclusion of communities define local implication rules denoted $\Box_v q \rightarrow \Box_v w$. Such a local rule holds whenever v is a vertex in a q community and w is shared by all members of this community. and may be also rewritten as $\Box_C q \rightarrow \Box_C w$ where C is the q community of v. Among these rules we consider those where q is a core closed pattern, i.e. $q = f(q)$. Such a rule asserts the following new *local knowledge*: pattern $f_v(c) \setminus c$ were not shared by all vertices in the pattern c core while it is common to all vertices in the c community containing v. Such rules are said to be *type-1 rules*.

EXEMPLE 10. — Following Example 9 we consider now the two communities associated to the 3-dense edge core. We obtain two type-1 rules $r_1 = \Box_{\text{left}} a \rightarrow \Box_{\text{left}} abd$ where "left" is a node of the left triangle, and $r_2 = \Box_{\text{right}} a \rightarrow \Box_{\text{right}} abc$ where "right" is a node of the right triangle.

□

3.4. Closed Pattern Mining in Attributed Graphs

We consider two pattern mining problems. Problem I consists in enumerating cores of patterns subgraphs and related core closed patterns:

– Find the set of all pairs (V', c) where V' is the vertex set, of size at least s, of the core of some pattern subgraph of G and c is the corresponding core closed pattern. Cores are either vertex or edge cores.

Problem II enumerates local rules and related communies whose left part are core closed patterns, i.e. type-1 local rules.

– Find the set of all triples (c, V'', l) where c is a core closed pattern according to core property P_1, V'' a c community of size at least ls, and l the associated local closed pattern.

In (Soldano, Santini, 2014) a solution to Problem I was proposed, regarding vertex cores, by first using a frequent closed pattern mining algorithm as a possibly costly preprocessing step, then applying interior operators, computing core closed patterns and finally removing duplicates. In what follows we propose a one-step algorithm adapted form the Divide and Conquer algorithm designed to efficiently compute closed itemsets described in (Boley *et al.*, 2010). and implemented in PARAMINER (Negrevergne *et al.*, 2013). The Boley's et al algorithm was shown to be correct and complete whatever is the closure operator, and may then be applied to compute core closed patterns. It is a standard top-down depth first search that divides the set of descendants of some closed pattern q into those containing some item i and those not containing i, therefore ensuring that each closed pattern is enumerated once. The division is obtained by maintaining an exclusion list of items EL. The Boley's et al algorithm was also shown as *polynomial delay* in the closed itemsets mining case, i.e. the delay between outputting two closed patterns was polynomial in the dataset size. This basically relies on the fact that each closure computation is polynomial and that whenever a closed pattern is avoided (as previously output) the whole branch in the search is pruned. As far as core propertieonly requires to consider the neighborhood of vertices or edges, the polynomial delay property is preserved. The algorithm that follows requires a graph $G = (V, E)$, a set of items I, a dataset D describing vertices as itemsets, a core property P, a frequency threshold s. It outputs the frequent pairs (c, Vc) where c is a core closed pattern and V_c its core support set We denote by $G(q)$ the pattern q subgraph of G.

- Core enumeration algorithm (G, I, D, P, s)
$G_1 \leftarrow core(G, P)$
$\text{enum}(\text{int}(\text{vs}(G_1)), G_1, \emptyset)$
- Function $\text{enum}(c, G_1, EL)$
Require: a core closed pattern c, an exclusion list EL
Ensure: outputs the frequent $(c', V_{c'})$ pairs
where $c' \supseteq c$ and contains no items of EL
$Output\ (c, \text{vs}(G_1))$
for all $x \in I \setminus c$ **do**
/ Generate all augmentations of c */*
$G_x \leftarrow \text{core}(G(c \cup \{x\}), P)$
if $\mid \text{vs}(G_x) \mid < s$ **then Exit**
$q \leftarrow \text{int}(\text{vs}(G_x))$
if $q \cap \text{EL} = \emptyset$ **then**
$\text{enum}(q, G_x, EL)$
$\text{EL} \leftarrow \text{EL} \cup \{x\}$
end if
end for
- Function $\text{int}(V_1)$
return $\cap_{v \in V_1)} D(v)$

Regarding Problem II, we use a similar algorithm, except that the core subgraph has to be divided into connected components each leading to a local closed pattern to form a (c, V'', l) triple. The two algorithms are implemented in one program named *minerLC* [1] that uses dataset reduction techniques (Uno *et al.*, 2004) to reduce the subgraphs during the depth-first traversal of the pattern space.

4. Experiments

We have experimented both algorithms on a variety of datasets. We present the experiments regarding Problem I and only mention experiments on Problem II at the end of this section. We performed our experiments in a variety of attributed graphs ranging from medium to large graphs with medium to large sets of items. We give Table 1 the main characteristics of these datasets. For each dataset, we indicate the number of edges ($|E|$), vertices ($|V|$) and labels ($|L|$), the average vertex degree ($\overline{deg(v)}$) and number of labels per vertex ($\overline{|l(v)|}$). DBLP.S was used in Silva (Silva *et al.*, 2012) while DBLP.C.ICDM (or DBLP.C for short), DBLP.XL and LASTFM were used in (Galbrun *et al.*, 2014).

We consider k-degree, k-dense and k-nearstar core properties, with values of k ranging from 0 to 40. k-degree and k-dense definitions are recalled in Section 3.1. k-nearstar is a vertex core property defined as follows: $P(v, V')$ is true if vertex v has degree at least k (v is a *star* node) or is linked to a vertex of degree at least k (v is a *satellite* node) in the subgraph $(V', \text{es}(V'))$. The nearstar-k core property is

Tableau 1. Datasets characteristics

Nom	$\|V\|$	$\|E\|$	$\|L\|$	$\overline{deg(v)}$	$\overline{\|l(v)\|}$
DBLP.C	2012	5079	4588	5.05	23.43
DBLP.XL	929937	3461697	92237	7.44	16.54
LastFM	1892	12717	17630	13.44	40.07
DBLP.S	108032	276658	23254	5.12	13.93

intended to detect "hubs" in scale-free networks, as the DBLP co-authoring networks experimented here, i.e. high degree nodes together with the nodes to which they are connected.

We report results for increasing k values, i.e. for weaker to stronger core properties, for a constant frequency level s. We also rank the patterns according to their *specificity* which is define as follows: given a pair (V_c, c) the specificity of c is the ratio $S(c) = |V_c|/|\text{ext}(c)|$ i.e. the ratio of vertices in the pattern c core subgraph to those in the whole pattern c subgraph.

In the upper part of Figure 2 we report the CPU time (left) and number of cores (right) versus k for the three core properties when requiring frequency at least 5. As expected reducing pattern subgraphs to core subgraphs strongly reduces the number of pattern subgraphs and therefore strongly reduces the number of closed patterns, leading to a much more condensed representation of frequent patterns. The number of k-denses and k-cores is close, while there is much more k-nearstar cores which is expected since the k-nearstar constraint is weaker: k-nearstar allows two star vertices ($d(v) \geq k$) to be connected together *via* a satellite vertex ($d(v) \geq 1$). The total CPU time strongly decreases with the strength k of the constraint and closely follows the number of core closed patterns. The lower part of Figure 2 displays the number of (core subgraph, core closed pattern) pairs in 4 datasets applying respectively the k-dense and k-nearstar core properties. The k-core curves (not shown) are above but closely follow the k-dense curves. The minimal support values s are respectively 5 (DBLP.C), 10 (LASTFM), 40 (DBLP.S), 50 (DBLP.XL). Regarding the largest datasets DBLP.S and DBLP.XL the range of k starts respectively at $k = 4$ for DBLP.S and $k = 6$ for DBLP.XL in the k-dense case and $k = 4$ for DBLP.S and $k = 15$ for DBLP.XL in the k-dense case. Below these value the number of pairs (more than 10^6 pairs) and CPU times become very large . Applying strong constraints clearly allows to focus on a much smaller subset of patterns. Figure 3 displays examples of 6-nearstar cores as found in the DBLP.C graph.

Table 2 we report the top 10 core closed patterns found in the DBLP.C dataset according to their specificity [2]when applying the 6-nearstar core operator.

Finally regarding the Problem II experiments we obtained similar CPU times and the number of communities was close to the number of cores, i.e. most of cores had only one connected component. This was unexpected and clearly depends on the da-

2. The words forming the patterns are reported as found in the dataset

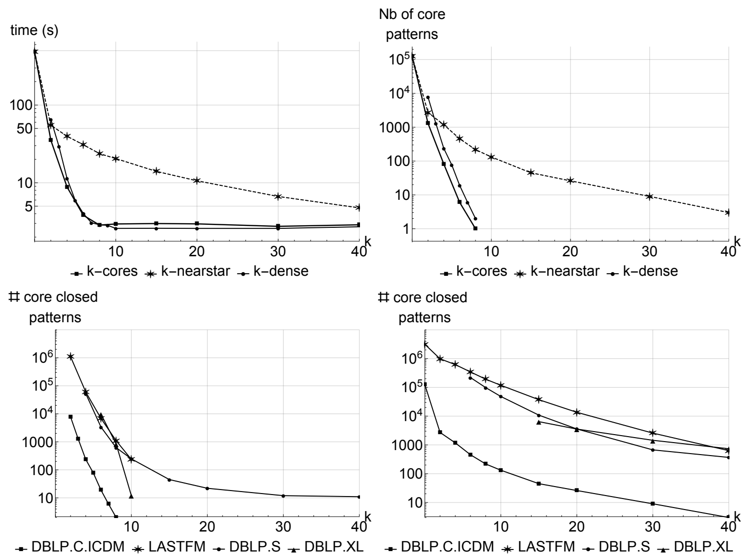

FIGURE 2. *Upper subfigures: Total execution time (left) and number of (V', c) pairs in the DBLP.C problem for k-core, k-dense and k-nearstar core properties with minimum frequency threshold $s = 5$. Lower subfigures: the number of (V', c) pairs in 4 datasets for k-dense (left) and k-nearstar (right) core properties.*

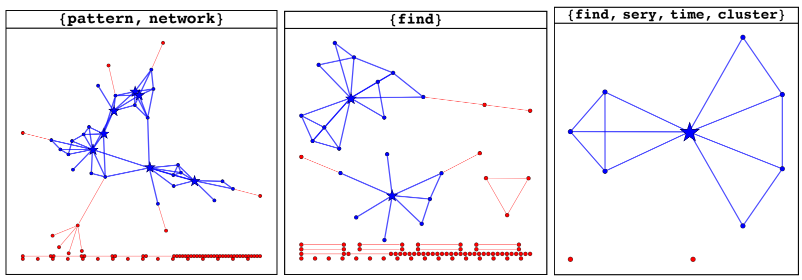

FIGURE 3. *6-nearstar cores from DBLP.C (blue vertices and bold edges). On the left the core displays several stars linked by edges or connected through their satellites. The central picture represents the {find} core made of two communities. On the right, the core of a pattern more specific than* find.

tasets. We are currently investigating how to strengthen the connectivity requirements of structural communities as defined here.

Tableau 2. Top-10 6-nearstar core closed patterns from DBLP.C (specificity).

6-nearstar core closed patterns	$\lvert ext(c) \rvert$	$\lvert Vc \rvert$	$Specif$
high similar search efficy cluster	7	7	1.
object similar search efficy cluster	7	7	1.
sequenty de extraction pattern mine	7	7	1.
warp sery time cluster	7	7	1.
databas graph pattern efficy mine	8	7	0.875
find sery time cluster mine	8	7	0.875
find sery mine	11	9	0.82
find sery time,mine	10	8	0.8
find sery time,cluster	10	8	0.8
find sery	15	12	0.8

5. Conclusion

We have proposed a general presentation of cores and structural communities, considering the latter as connected components of edge or vertex cores. Our purpose was to import techniques from closed pattern mining to mine attributed graphs according to various form and level of granularity. This results in a generic algorithm, implemented in a modular program in which each new core operator can be incorporated by introducing the corresponding core property. The algorithm solves efficiently two kinds of mining problems, either enumerating cores of attribute pattern subgraphs and associated maximal patterns, or extracting structural communities and associated closed patterns through local rules expressing to what extent some community has items that are not common to the whole associated core. The program handles rather large datasets, as the DBLP.XL. As cores we have investigated the k-core, the k-dense edge core, which has been recently used to investigate the internet network dynamics (Orsini *et al.*, 2014) and also introduced the k-nearstar vertex core whose purpose is to investigate hubs-related subgraphs. The framework is very flexible and we intend to extend the methodology to directed networks, introducing appropriate cores as D-cores, to graphs whose edges are labelled, which means considering edges as labelled objects in which patterns occur, and finally to multiplex networks, for which appropriate core definitions are yet to be investigated.

Bibliographie

Batagelj V., Zaversnik M. (2011). Fast algorithms for determining (generalized) core groups in social networks. *Adv. Data Analysis and Classification*, vol. 5, nº 2, p. 129–145.

Boley M., Horváth T., Poigné A., Wrobel S. (2010). Listing closed sets of strongly accessible set systems with applications to data mining. *Theor. Comput. Sci.*, vol. 411, nº 3, p. 691-700.

Flake G. W., Lawrence S., Giles C. L. (2000). Efficient identification of web communities. In *6th ACM SIGKDD international conference on knowledge discovery and data mining, boston, ma, usa, 2000*, p. 150–160.

Francisco A. P., Oliveira A. L. (2010). Fully generalized graph cores. In L. da F. Costa, A. Evsukoff, G. Mangioni, R. Menezes (Eds.), *Complex networks - second international workshop, complenet 2010, rio de janeiro, brazil, october 13-15, 2010, revised selected papers*, vol. 116, p. 22–34. Springer.

Galbrun E., Gionis A., Tatti N. (2014). Overlapping community detection in labeled graphs. *Data Min. Knowl. Discov.*, vol. 28, n° 5-6, p. 1586–1610.

Lareau C. A., White B. C., Oberg A. L., McKinney B. A. (2015). Differential co-expression network centrality and machine learning feature selection for identifying susceptibility hubs in networks with scale-free structure. *BioData Mining*, vol. 8, p. 5.

Mougel P.-N., Rigotti C., Gandrillon O. (2012). 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*, vol. 7302, p. 181-192. Springer.

Negrevergne B., Termier A., Rousset M.-C., Méhaut J.-F. (2013). Paraminer: a generic pattern mining algorithm for multi-core architectures. *Data Mining and Knowledge Discovery*, vol. 28, n° 3, p. 593-633.

Orsini C., Gregori E., Lenzini L., Krioukov D. (2014, décembre). Evolution of the internet k-dense structure. *IEEE/ACM Trans. Netw.*, vol. 22, n° 6, p. 1769–1780.

Palla G., Derenyi I., Farkas I., Vicsek T. (2005, Jun). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, vol. 435, n° 7043, p. 814–818.

Pernelle N., Rousset M.-C., Soldano H., Ventos V. (2002). Zoom: a nested Galois lattices-based system for conceptual clustering. *J. of Experimental and Theoretical Artificial Intelligence*, vol. 2/3, n° 14, p. 157–187.

Saito K., Yamada T., Kazama K. (2009). The k-dense method to extract communities from complex networks. In D. A. Zighed, S. Tsumoto, Z. W. Ras, H. Hacid (Eds.), *Mining complex data*, vol. 165, p. 243–257. Springer.

Seidman S. B. (1983). Network structure and minimum degree. *Social Networks*, vol. 5, p. 269–287.

Silva A., Meira W., Jr., Zaki M. J. (2012, janvier). Mining attribute-structure correlated patterns in large attributed graphs. *Proc. VLDB Endow.*, vol. 5, n° 5, p. 466–477.

Soldano H., Santini G. (2014). Graph abstraction for closed pattern mining in attributed network. In T. Schaub, G. Friedrich, B. O'Sullivan (Eds.), *European conference in artificial intelligence (ECAI)*, vol. 263, p. 849–854. IOS Press.

Soldano H., Santini G., Bouthinon D. (2015, November 9-11). Local knowledge discovery in attributed graphs. In A. Esposito (Ed.), *27th IEEE international conference on tools with artificial intelligence, ICTAI, vietri sul mare, italy*, p. 250–257. IEEE Computer Society.

Uno T., Kiyomi M., Arimura H. (2004). LCM ver. 2: Efficient mining algorithms for frequent/closed/maximal itemsets. In R. J. Bayardo Jr., B. Goethals, M. J. Zaki (Eds.), *FIMI '04, proceedings of the IEEE ICDM workshop on frequent itemset mining implementations, brighton, uk, november 1, 2004*, vol. 126. CEUR-WS.org.