# Mining Graph Topological Patterns: Finding Covariations among Vertex Descriptors

Adriana Prado, Marc Plantevit, *Member*, *IEEE*, Céline Robardet, and Jean-François Boulicaut

**Abstract**—We propose to mine the graph topology of a large attributed graph by finding regularities among vertex descriptors. Such descriptors are of two types: 1) the vertex attributes that convey the information of the vertices themselves and 2) some topological properties used to describe the connectivity of the vertices. These descriptors are mostly of numerical or ordinal types and their similarity can be captured by quantifying their covariation. Mining topological patterns relies on frequent pattern mining and graph topology analysis to reveal the links that exist between the relation encoded by the graph and the vertex attributes. We propose three interestingness measures of topological patterns that differ by the pairs of vertices considered while evaluating up and down co-variations between vertex descriptors. An efficient algorithm that combines search and pruning strategies to look for the most relevant topological patterns is presented. Besides a classical empirical study, we report case studies on four real-life networks showing that our approach provides valuable knowledge.

**Index Terms**—Data mining, mining methods and analysis, attributed graph mining, topological patterns

## 1 Introduction

Real-world phenomena are often depicted by graphs where vertices represent entities and edges represent their relationships or interactions. Entities are also described by one or more attributes that constitute the attribute vectors associated with the vertices of the attributed graph. Existing methods that support the discovery of local patterns in graphs mainly focus on the topological structure of the patterns, by extracting specific subgraphs while ignoring the vertex properties (cliques [22], quasi-cliques [21], [31]), or compute frequent relationships between vertex attribute values (frequent subgraphs in a collection of graphs [17] or in a single graph [4]), while ignoring the topological status of the vertices within the whole graph, for example, the vertex connectivity or centrality. The same limitation holds for the methods proposed in [19], [24], [28], and [29], which identify sets of vertices that share local attributes and that are close neighbors. Such approaches only focus on a local neighborhood of the vertices and do not consider the connectivity of the vertex in the whole graph. In this paper, we propose to compute relevant patterns that integrate information about the connectivity of the vertices and their attribute values.

The connectivity of each vertex is described by topological properties that quantify its topological status in the graph. Some of these properties are based on the close neighborhood of the vertices (e.g., the vertex degree), while others describe the connectivity of a vertex by considering its relationship with all other vertices (e.g., the centrality measures). Combining such microscopic and macroscopic properties characterizes the connectivity of the vertices and it may be a sound basis to explain why some vertices have similar attribute values.

Such topological properties and vertex attributes are mostly of numerical or ordinal types and their similarity can be captured by quantifying their covariation. Such covariation indicates how a set of vertex descriptors tend to monotonically increase or decrease all together. Therefore, following the way paved by Calders et al. [5], we propose to mine rank-correlated sets over graph descriptors by extracting topological patterns defined as a set of vertex properties and attributes that strongly covary over the vertices of the graph. We introduce several interestingness measures of topological patterns that differ by the pairs of vertices that are considered while evaluating up and down covariations between descriptors: 1) Considering all the vertex pairs enables to find patterns that are true all over the graph; 2) Including only the vertex pairs that are in a specific order regarding a selected numerical or ordinal attribute reveals the topological patterns that emerge with this attribute; 3) Examining the vertex pairs that are connected in the graph makes it possible to identify patterns that are *structurally correlated* to the relationship encoded by the graph. We also design an operator that identifies the top $k$ representative vertices of a topological pattern.

Let us illustrate our proposal on a coauthorship graph depicted in Fig. 1, where vertices (from $A$ to $P$) denote authors, edges encode coauthorship relations, and three attributes describe author: $h$ corresponds to the author $h$-index, which attempts to measure both the productivity and the impact of the published work of each author [16]; $i$ denotes the average number of hours per week spent by

- A. Prado, C. Robardet, and J-F. Boulicaut are with the Université de Lyon, LyonTech Campus de la doua, INSA-Lyon, LIRIS CNRS UMR 5205, Bâtiment Blaise Pascal, F-69621 Villeurbanne Cedex, France. E-mail: adriana_bechara@yahoo.com.br, {Celine.Robardet, jean-francois.boulicaut}@insa-lyon.fr.
- M. Plantevit is with the Université de Lyon, LyonTech Campus de la doua, Université Lyon 1, LIRIS CNRS UMR 5205, Bâtiment Nautibus, 43, Bd du 11 Novembre 1918, F-69622 Villeurbanne Cedex, France. E-mail: marc.plantevit@liris.cnrs.fr.

*Manuscript received 9 Apr. 2012; revised 16 July 2012; accepted 25 July 2012; published online 1 Aug. 2012.*
*Recommended for acceptance by M. Ester.*
*For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDE-2012-04-0240.*
*Digital Object Identifier no. 10.1109/TKDE.2012.154.*

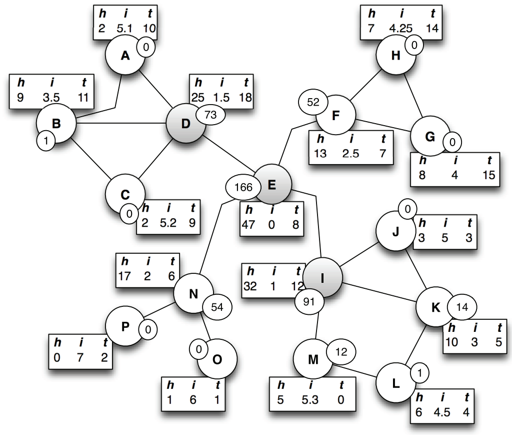

Fig. 1. A coauthorship attributed graph toy example.

each author on instructional duties; and $t$ designates the number of publications the author had in the IEEE TKDE journal. As topological property, we consider the betweenness centrality measure that is the number of times a vertex appears on a shortest path of the graph (see Section 2). This value is in a circle associated with each vertex on Fig. 1. For instance, vertex $D$ has attribute values $h = 25$, $i = 1.5$, and $t = 18$ and a betweenness centrality value equal to 73. One of the topological patterns extracted from this attributed graph is $P = \{h^+, i^-, \mathrm{BETW}^+\}$, whose meaning is *the higher the value of attribute h, the lower the value of attribute i and the higher the betweenness centrality of a vertex*. In other words, authors that tend to have a high $h$-index, tend to have a low instructional duty and publish articles with coauthors that are also central in the graph, inducing a rather small distance to other vertices. This topological pattern combines a topological property (BETW) with two vertex attributes ($h$ and $i$) and is supported by 89 pairs of vertices among the $\binom{16}{2}$ possible pairs over the graph. Its top three representative vertices are $E, I$, and $D$ (shadowed on Fig. 1). These vertices have the highest values on $h$ and BETW, and the lowest values on attribute $i$ compared to other vertices. Therefore, these dominant vertices have a significant impact on the support of this pattern.

In this paper, we design an algorithm, called TopGraphMiner, which discovers topological patterns. Given an attributed graph, such as the one in Fig. 1, it first computes a set of topological properties for every of its vertices. TopGraphMiner then integrates search and pruning strategies to look for the most relevant topological patterns. Finally, it gives to the user the ability to visualize every pattern on the input graph by identifying its top $k$ representative vertices.

Our main contributions are as follows: First, we propose a new kind of graph analysis that exploits attributes and topological properties of vertices. Second, to produce such an analysis, we provide new insights into simpler covariation pattern mining [5] by considering up and down covariations. We define new upper bounds on the support of such covariations. We also introduce several interestingness measures for topological patterns, and a valuable support to pattern visualization on the original graph thanks to the identification of its top $k$ representative vertices. Third, we present an empirical study that includes:

1. A comparison of TopGraphMiner with a baseline approach;
2. A study of its empirical complexity;
3. An analysis of the pruning capability of the proposed upper bound; and
4. A qualitative feedback on some patterns extracted from four real-life networks: a coauthorship network, a movie coactor network, a patent citation network, and a gene interaction network.

This paper is structured as follows: Section 2 presents topological vertex properties. Sections 3 and 4 introduce our new model for mining topological patterns. Our algorithm is defined in Section 5. Its efficiency and its effectiveness are discussed in Sections 6 and 7. Section 8 addresses the related work and Section 9 concludes.

## 2 Topological Vertex Properties

The input of our mining task is a nondirected attributed graph $G = (V, E, L)$, where $V$ is a set of $n$ vertices, $E$ a set of $m$ edges, and $L = \{l_1, \ldots, l_p\}$ a set of $p$ attributes associated with each vertex of $V$, which may be numerical or ordinal.

Important properties of the vertices are also encoded by the edges of the graph, which describe inter-relations between vertices. From this relation, we can compute some topological properties that synthesize the role played by each vertex in the graph. The topological properties we are interested in range from a microscopic level—those that described a vertex based on its direct neighborhood—to a macroscopic level—those that characterize a vertex by considering its relationship to all other vertices in the graph. Statistical distributions of these properties are generally used to depict large graphs (see, e.g., [2], [18]). We propose here to use them as vertex descriptors. Some examples of these properties are given from Fig. 1.

### 2.1 Microscopic Properties

We propose to use four topological properties to describe the direct neighborhood of a vertex $v$:

- The degree of $v$ is the number of edges incident to $v$ ($deg(v) = |\{u \in V,\ \{u, v\} \in E\}|$). When normalized by the maximum number of edges a vertex can have, it is called the degree centrality coefficient: $\mathrm{DEGREE}(v) = \frac{deg(v)}{n-1}$ (e.g., $\mathrm{DEGREE}(B) = \frac{3}{15}$).
- The clustering coefficient evaluates the connectivity of the neighbors of $v$ and thus its local density:

$$\mathrm{CLUST}(v) = \frac{2 \mid \{\{u, w\} \in E, \{u, v\} \in E \wedge \{v, w\} \in E\}|}{deg(v)(deg(v) - 1)},$$

(e.g., $\mathrm{CLUST}(B) = \frac{2|\{\{A,D\}\{C,D\}\}|}{3\times 2} = \frac{2}{3}$).
- To better understand the structure of the neighborhood of $v$, we also consider the quasi-cliques [21] that involve $v$. $v$ belongs to a $\gamma$-quasi clique $Q$ iff the graph $G_Q$ induced by the set of vertices $Q$ is connected and satisfies

$$\forall u \in Q,\ deg_{G_Q}(u) \geq \lceil \gamma(|Q|-1) \rceil,$$

where $deg_{G_Q}(u)$ is the degree of $u$ in $G_Q$ (e.g.,$\{A, B, C, D\}$ is a 2/3-quasi clique since

$$deg_{G_Q}(A) = deg_{G_Q}(c) = 2 \geq \left\lceil \frac{2}{3}(4-1) \right\rceil \quad \text{and}$$
$$deg_{G_Q}(B) = deg_{G_Q}(D) = 3 \geq 2).$$

We consider two properties based on the quasi-cliques involving $v$: the size of the largest quasi-clique (SZQC($v$)) and the number of quasi-cliques (NBQC($v$)).

## 2.2 Macroscopic Properties

We consider five macroscopic topological properties to characterize a vertex while taking into account its connectivity to all other vertices of the graph.

- Vertex communities can be computed by looking for a partition of $V$ that maximizes the Newman's modularity measure [25]. This criterion is based on the proportion of edges that fall within the community minus the expected such proportion if edges were distributed at random:
$$Q = \frac{1}{4m}\sum_{u,v}\left(\mathbb{1}_E(\{u,v\}) - \frac{deg(u)deg(v)}{2m}\right)\delta_{c_u,c_v},$$
where $c_v$ is the community assigned to $v$, $\delta_{c_u,c_v}$ is the Kronecker delta ($\delta_{c_u,c_v} = 1$ if $c_u = c_v$ and $\delta_{c_u,c_v} = 0$ otherwise), $\mathbb{1}_E(\{u,v\})$ is the indicator function of the set $E$ ($\mathbb{1}_E(\{u,v\}) = 1$ if $\{u,v\} \in E$, 0 otherwise). For example, according to such a definition, the four communities on Fig. 1 are $\{A,B,C,D\}, \{E,N,O,P\}, \{F,G,H\}, \{I,J,K,L,M\}$. As topological property, we consider the size of the community of $v$(SZCOM($v$)).
- The relative importance of vertices in a graph can be obtained through centrality measures [11]. Closeness centrality CLOSE($v$) is defined as the inverse of the average distance between $v$ and all other vertices that are reachable from it. The distance between two vertices is defined as the number of edges of the shortest path between them:
$$\mathrm{CLOSE}(v) = \frac{n}{\sum_{u \in V} |shortest\_path(u,v)|}$$
(e.g., CLOSE($B$) = 0.021 and CLOSE($E$) = 0.037).
- The betweenness centrality BETW($v$) of $v$ is equal to the number of times a vertex appears on a shortest path in the graph. It is evaluated by first computing all the shortest paths between every pair of vertices, and then counting the number of times a vertex appears on these paths: $\mathrm{BETW}(v) = \sum_{u,w} \mathbb{1}_{shortest\_path(u,w)}(v)$ (e.g., BETW($B$) = 1 and BETW($E$) = 166).
- The eigenvector centrality measure (EGVECT) favors vertices that are connected to vertices with high eigenvector centrality. This recursive definition can be expressed by the following eigenvector equation $Ax = \lambda x$ which is solved by the eigenvector $x$ associated with the largest eigenvalue $\lambda$ of the adjacency matrix $A$ of the graph (e.g., EGVECT($B$) = 0.093 and EGVECT($E$) = 0.114).
- The PAGERANK index [3] is based on a random walk on the vertices of the graph, where the probability to go from one vertex to another is modelled as a Markov chain in which the states are vertices and the transition probabilities are computed based on the edges of the graph. This index reflects the probability that the random walk ends at the vertex itself:
$$\mathrm{PAGERANK}(v) = \alpha \sum_j \mathbb{1}_E(\{u,v\}) \frac{\mathrm{PAGERANK}(u)}{deg(u)} + \frac{1-\alpha}{n},$$
where the parameter $\alpha$ is the probability that a random jump to vertex $v$ occurs (e.g.,
$$\mathrm{PAGERANK}(B) = 1.11 \quad \text{and}$$
$$\mathrm{PAGERANK}(E) = 1.50).$$

These nine topological properties characterizes the graph relationship encoded by $E$. These properties, along with the set of vertex attributes $L$, constitutes the set of vertex descriptors $\mathcal{D}$ used in this paper.

# 3 Topological Patterns

Let us now consider topological patterns as a set of vertex attributes and topological properties that behave similarly over a large part of the vertices of the graph. We assume that all topological properties and vertex attributes are of numerical or ordinal type, and we propose to capture their similarity by quantifying their covariation over the vertices of the graph. Topological patterns are defined as $P = \{D_1{}^{s_1}, \ldots, D_\ell{}^{s_\ell}\}$, where $D_j$ is a vertex descriptor from $\mathcal{D}$ and $s_j \in \{+,-\}$ is its covariation sign. Following the example of Fig. 1, the trend "*the more papers in IEEE TKDE (t) the lower the average number of hours per week spent on instructional duties (i)*" is represented by the pattern $\{t^+, i^-\}$. In the following, we propose three interestingness measures that are different with respect to the pairs of vertices considered while evaluating the support of such patterns.

## 3.1 Topological Patterns over the Whole Graph

Several signed vertex descriptors covary if the orders induced by each of them on the set of vertices are consistent. This consistency is evaluated by the number of vertex pairs ordered the same way by all descriptors. The number of such pairs constitutes the so-called support of the pattern. This measure can be seen as a generalization of the Kendall's $\tau$ measure. When we consider all possible vertex pairs, this interestingness measure is defined as follows.

**Definition 1 (*Supp<sub>all</sub>*).** *The support of a topological pattern $P$ over all possible pairs of vertices is*

$$Supp_{all}(P) = \frac{|\{(u,v) \in V^2 \mid \forall D_j^{s_j} \in P : D_j(u) \triangleright_{s_j} D_j(v)\}|}{\binom{n}{2}},$$

*where $\triangleright_{s_j}$ denotes $<$ when $s_j$ is equal to $+$, and $\triangleright_{s_j}$ denotes $>$ when $s_j$ is equal to $-$.*

This measure gives the number of vertex pairs $(u, v)$ such that $u$ is strictly lower than $v$ on all descriptors with sign $+$, and $u$ is strictly higher than $v$ on descriptors with sign $-$. For instance, the pattern $P = \{t^+, i^-\}$ is supported by 85 pairs among the 120 possibles ones, hence $Supp_{all}(P) = 0.71$.

As mentioned in [5], $Supp_{all}$ is an antimonotonic measure for positively signed descriptors. This is still true when considering negatively signed ones: adding $D^-_{l+1}$ to a pattern $P$ leads to a support lower than or equal to that of $P$ since the pairs $(u, v)$ that support $P$ must also satisfy $D_{l+1}(u) > D_{l+1}(v)$. Besides, when adding descriptors with negative sign, the support of some patterns can be deduced from others, the latter referred to as symmetrical patterns.

**Property 1 (Support of Symmetrical Patterns).** *Let $P$ be a topological pattern and $\overline{P}$ be its symmetrical, that is, $\forall D_j^{s_j} \in P$, $D_j^{\overline{s_j}} \in \overline{P}$, with $\overline{s_j} = \{+, -\} \setminus \{s_j\}$. If a pair $(u, v)$ of $V^2$ contributes to the support of $P$, then the pair $(v, u)$ contributes to the support of $\overline{P}$. Thus, we have $Supp_{all}(P) = Supp_{all}(\overline{P})$.*

Topological patterns and their symmetrical patterns are semantically equivalent. To avoid the irrelevant computation of duplicate topological patterns, we exploit Property 1.

Thus, mining frequent topological patterns consists in computing all sets of signed descriptors $P$, but not their symmetrical ones, such that $Supp_{all}(P) \geq minsup$, where $minsup$ is a user-defined minimum support threshold.

## 3.2 Other Interestingness Measures

To identify most interesting topological patterns, we propose to give to the end-user the possibility of guiding its data mining process by querying the patterns with respect to their correlation with the relationship encoded by the graph or with a selected descriptor. Therefore, we revisit the notion of emerging patterns [10] by identifying the patterns whose support is significantly greater (i.e., according to a growth-rate threshold) in a specific subset of vertex pairs than in the remaining ones. This subset can be defined in different ways according to the end-user's motivations: either it is defined by the vertex pairs that are ordered with respect to a selected descriptor called the class descriptor, or it is equal to $E$, the set of edges. Whereas the former highlights the correlation of a pattern with the class descriptor, the latter enables to characterize the importance of the graph structure within the support of the topological pattern. For instance, considering the toy example of Fig. 1, $h^+t^+$ and $h^+t^-$ are both frequent with minimum support of 20 percent. Note that although these patterns are contradicting, they are both output by our approach when only the frequency constraint is considered. The extraction of emerging patterns with respect to $t$ outputs the pattern $h^+t^+$ as the frequency of $h^+$ is significantly greater in $t^+$ than in $t^-$ (with a factor of 2.13). $h^+t^+$ is more emerging with respect to $E$ than $h^+t^-$, their growth rates being, respectively, equal to 1.23 and 0.59.

### 3.2.1 Emerging Patterns w.r.t. a Selected Descriptor

Let us consider a selected descriptor $C \in \mathcal{D}$ and a sign $r \in \{+, -\}$. The set of pairs of vertices that are ordered by $C^r$ is

$$\mathcal{C}_{C^r} = \{(u, v) \in V^2 \mid C(u) \triangleright_r C(v)\}.$$

The support measure based on the vertex pairs of $\mathcal{C}_{C^r}$ is defined below.

**Definition 2 ($Supp_{C^r}$).** *The support of a topological pattern $P$ over $C^r$ is*

$$Supp_{C^r}(P) = \frac{|\{(u, v) \in \mathcal{C}_{C^r} \mid \forall D_j^{s_j} \in P : D_j(u) \triangleright_{s_j} D_j(v)\}|}{|\mathcal{C}_{C^r}|}.$$

Analogously, the support of $P$ over the pairs of vertices that do not belong to $\mathcal{C}_{C^r}$ is denoted $Supp_{\overline{C^r}}(P)$. To evaluate the impact of $C^r$ on the support of $P$, we consider the growth rate of the support of $P$ over the partition of vertex pairs $\{\mathcal{C}_{C^r}, \mathcal{C}_{\overline{C^r}}\}$: $Gr(P, C^r) = \frac{Supp_{C^r}(P)}{Supp_{\overline{C^r}}(P)}$.

If $Gr(P, C^r)$ is greater than a minimum growth-rate threshold, then $P$ is referred to as emerging with respect to $C^r$. If $Gr(P, C^r) \approx 1$, $P$ is as frequent in $\mathcal{C}_{C^r}$ as in $\mathcal{C}_{\overline{C^r}}$. If $gr(P, C^r) \gg 1$, $P$ is much more frequent in $\mathcal{C}_{C^r}$ than in $\mathcal{C}_{\overline{C^r}}$. For example, $Gr(\{h^+, i^-, \mathrm{BETW}^+\}, t^+) = 2.31$. The intuition behind this definition is to identify the topological patterns that are mostly supported by pairs of vertices that are also ordered by the selected descriptor.

### 3.2.2 Emerging Patterns w.r.t. the Graph Structure

It is interesting to measure if the graph structure plays an important role in the support of a topological pattern $P$. To this end, we define a similar support measure based on pairs that belongs to $E$, the set of edges of the graph:

$$\mathcal{C}_E = \{(u, v) \in V^2 \mid \{u, v\} \in E\}.$$

Based on this set of pairs, we define the support of $P$ as.

**Definition 3 ($Supp_E$).** *The support of a topological pattern $P$ over the pairs of vertices that are linked in $G$ is*

$$Supp_E(P) = \frac{2|\{(u, v) \in \mathcal{C}_E \mid \forall D_j^{s_j} \in P : D_j(u) \triangleright_{s_j} D_j(v)\}|}{|\mathcal{C}_E|}.$$

The maximum value of the numerator is $\frac{|\mathcal{C}_E|}{2}$ since: 1) if $(u, v) \in \mathcal{C}_E$ then $(v, u) \in \mathcal{C}_E$, and 2) it is not possible that $\forall D_j^{s_j} \in P$, $D_j(u) \triangleright_{s_j} D_j(v)$, and $D_j(v) \triangleright_{s_j} D_j(u)$ at the same time. For instance, the pattern $\{h^+, i^-\}$ is supported by all the 20 possible pairs that are edges, its support is, thus, equal to 1. The support of $P$ over the pairs of vertices that do not belong to $\mathcal{C}_E$ is denoted $Supp_{\overline{E}}(P)$.

As before, to evaluate the impact of $E$ on the support of $P$, we consider the growth rate of the support of $P$ over the partition of vertex pairs $\{\mathcal{C}_E, \mathcal{C}_{\overline{E}}\}$: $Gr(P, E) = \frac{Supp_E(P)}{Supp_{\overline{E}}(P)}$.

$Gr(P, E)$ enables to assess the impact of the graph structure on the pattern. Therefore, if $Gr(P, E) \gg 1$, $P$ is said to be *structurally* correlated. If $Gr(P, E) \ll 1$, the graph structure tends to inhibit the support of $P$. For example, on Fig. 1, the most structurally correlated pattern is $P = \{h^+, t^+, \mathrm{BETW}^+\}$ with $Gr(P, E) = 1.628$.

# 4 Top k Representative Vertices

The user may be interested in identifying the vertices that are the most representative of a given topological pattern, thus enabling the projection of the patterns back into the graph. For example, the representative vertices of the pattern $\{t^+, \mathrm{BETW}^-\}$ would be researchers with a relatively large number of IEEE TKDE papers and a low betweenness centrality measure.

We denote by $S(P)$ the set of vertex pairs $(u, v)$ that constitutes the support of a topological pattern $P$:

$$S(P) = \{(u, v) \in V^2 \mid \forall D_j^{s_j} \in P : D_j(u) \triangleright_{s_j} D_j(v)\},$$

which forms, with $V$, a directed graph $G_P = (V, S(P))$. This graph satisfies the following property.

**Property 2.** *The graph $G_P = (V, S(P))$ is transitive and acyclic.*

**Proof.** Let us consider $(u, v) \in V^2$ and $(v, w) \in V^2$ such that, $\forall D_j^{s_j} \in P : D_j(u) \triangleright_{s_j} D_j(v)$ and $D_j(v) \triangleright_{s_j} D_j(w)$. Thus, $D_j(u) \triangleright_{s_j} D_j(w)$ and $(u, w) \in S(P)$. Therefore, $G_P$ is transitive.

As $\triangleright_s \in \{<, >\}$, it stands for a strict inequality. Thus, if $(u, v) \in S(P)$, $(v, u) \notin S(P)$. Furthermore, as $G_P$ is transitive, if there exists a path between $u$ and $v$, there is also an arc $(u, v) \in S(P)$. Therefore, $(v, u) \notin S(P)$ and we can conclude that $G_P$ is acyclic. □

As $G_P$ is acyclic, it admits a topological ordering of its vertices, which is, in the general case, not unique. The top $k$ representative vertices of a topological pattern $P$ are identified on the basis of such a topological ordering of $V$ and are the $k$ last vertices with respect to this ordering. Considering that an arc $(u, v) \in S(P)$ is such that $v$ dominates $u$ on $P$, this vertex set contains the most dominant vertices on $P$. The top $k$ representative vertices of $P$ can be easily identified by ordering the vertices by their incoming degree as shown in Section 5.3.2.

# 5 Algorithm TopGraphMiner

Having described the topological pattern domain, this section aims at presenting TopGraphMiner, an efficient algorithm that combines search and pruning strategies to identify the most relevant topological patterns. Indeed, as the support counting is quadratic in the number of vertices, it is important to avoid, in linear time, some useless support computation. To this end, we derive an upper bound on the support used to safely prune nonpromising topological patterns.

## 5.1 Upper Bound on the Support Measure

To define an upper bound on the support of a given topological pattern which benefits from the presence of ties in the descriptors, a rank value $\rho(D(u))$ is associated with each numerical descriptor value $D(u)$ [5]. $\rho(D(u))$ is the index of $u$ in $V$ when $V$ is sorted in ascending order with respect to $D$, such that $1 \leq \rho(D(u)) \leq |V|$, ties being handled arbitrarily. Actually, due to the presence of ties, there are many possible rankings, but in all of them, the ranks of a given value range in an interval defined by $[\underline{\rho}(D(u)), \overline{\rho}(D(u))]$ with:

$$\underline{\rho}(D(u)) = \min\{\rho(D(v)) \mid v \in V \textit{ and } D(v) = D(u)\},$$
$$\overline{\rho}(D(u)) = \max\{\rho(D(v)) \mid v \in V \textit{ and } D(v) = D(u)\}.$$

For instance, on graph of Fig. 1, $\underline{\rho}(\mathrm{BETW}(B)) = 8$ and $\overline{\rho}(\mathrm{BETW}(B)) = 9$. Given two descriptors $A$ and $B$ and their respective signs $s_a$ and $s_b$, the ranking intervals over these descriptors can be used to establish a lower bound on the number of vertices that cannot form a supporting pair with $u$. If $v_a$ is a vertex such that $(A(v_a) \trianglerighteq_{s_a} A(u))$, then the pair $(u, v_a)$ cannot support $A^{s_a}B^{s_b}$. On the other hand, if a vertex $v_b$ does not satisfy $(B(v_b) \triangleright_{s_b} B(u))$, then the pair $(v_b, u)$ cannot support $A^{s_a}B^{s_b}$ either. We denote $I^{s_a}$ and $J^{s_b}$ the sets of vertices $v_a$ and $v_b$, respectively. Then, $\mathrm{Diff}_{A^{s_a}B^{s_b}}$ is the set of vertices that cannot form a supporting pair with $u$:

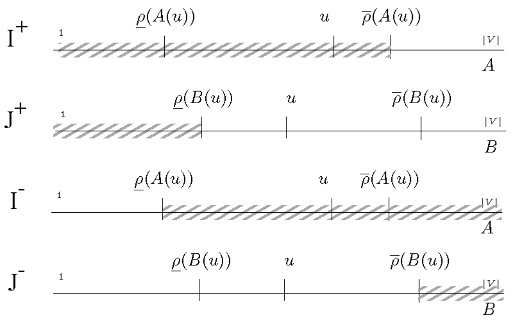

Fig. 2. Illustration of the computation of $\mathrm{Diff}_{A^{s_a}B^{s_b}}$.

$$\mathrm{Diff}_{A^{s_a}B^{s_b}} = \{v \in V \mid v \in I^{s_a} \wedge v \notin J^{s_b}\}.$$

Depending on the values of $s_a$ and $s_b$, the cardinality of $I^{s_a}$ and $J^{s_b}$ can easily be computed from the end points of the ranking intervals:

$$|I^+| = |\{v \in V \mid A(v) \leq A(u) \textit{ and } v \neq u\}| = \overline{\rho}(A(u)) - 1,$$
$$|J^+| = |\{v \in V \mid B(v) < B(u) \textit{ and } v \neq u\}| = \underline{\rho}(B(u)) - 1,$$
$$|I^-| = |\{v \in V \mid A(v) \geq A(u) \textit{ and } v \neq u\}| = |V| - \underline{\rho}(A(u)),$$
$$|J^-| = |\{v \in V \mid B(v) > B(u) \textit{ and } v \neq u\}| = |V| - \overline{\rho}(B(u)).$$

Fig. 2 illustrates these sets. In every case, the line represents the vertices sorted by the descriptor depicted on the right, in ascending order. In each line, we distinguish a given vertex $u$ and the end points of the interval containing the vertices with the same value as $u$ ($\underline{\rho}(D(u))$ and $\overline{\rho}(D(u))$). Besides, the hatched gray rectangle gives the set $I^{s_a}$ or $J^{s_b}$.

Since we cannot derive the exact cardinality of $\mathrm{Diff}_{A^{s_a}B^{s_b}}$, given that we do not know how the sets $I^{s_a}$ and $J^{s_b}$ intersect, we compute a lower bound on it. If $|I^{s_a}| \geq |J^{s_b}|$, then the cardinality of $\mathrm{Diff}_{A^{s_a}B^{s_b}}$ is minimal when $J^{s_b} \subseteq I^{s_a}$. Analogously, if $|I^{s_a}| < |J^{s_b}|$, then $\mathrm{Diff}_{A^{s_a}B^{s_b}}$ can be empty, and thus its cardinality is 0. Thus,

$$|\mathrm{Diff}_{A^+B^+}| \geq \max\{0, (\overline{\rho}(A(u)) - \underline{\rho}(B(u)))\},$$
$$|\mathrm{Diff}_{A^-B^-}| \geq \max\{0, (\overline{\rho}(B(u)) - \underline{\rho}(A(u)))\},$$
$$|\mathrm{Diff}_{A^+B^-}| \geq \max\{0, (\overline{\rho}(A(u)) - 1 - (|V| - \overline{\rho}(B(u))))\},$$
$$|\mathrm{Diff}_{A^-B^+}| \geq \max\{0, (|V| - \underline{\rho}(A(u))) - (\underline{\rho}(B(u)) - 1))\}.$$

To establish an upper bound on the support of a pattern $P$, we take, for each vertex $u$, the pair of signed descriptors $A^{s_a}B^{s_b}$ of $P$ that maximizes $\mathrm{Diff}_{A^{s_a}B^{s_b}}$: $\mathrm{maxDiff}_P(u) = \max_{A^{s_a}B^{s_b} \in P^2} |\mathrm{Diff}_{A^{s_a}B^{s_b}}|$. For instance, $\mathrm{maxDiff}_{\{i^- \mathrm{BETW}^+\}}(B) = \max\{|\mathrm{Diff}_{i^- \mathrm{BETW}^+}|, |\mathrm{Diff}_{\mathrm{BETW}^+, i^-}|\} = \max\{16 - 7 - 8 + 1, 9 - 1 - 16 + 7\} = 2$. This leads to the following upper bound:

**Theorem 1 (Upper Bound on *Supp*).** *Let $P$ be a topological pattern,*

$$Supp_{all}(P) \leq 1 - \frac{\sum_{u \in V} \mathrm{maxDiff}_P(u)}{n(n-1)}. \qquad (1)$$

**Proof.** For each vertex $u$, let us consider two descriptors $A^{s_a}$ and $B^{s_b}$ from $P$ such as $\mathrm{maxDiff}_P(u) = |\mathrm{Diff}_{A^{s_a}B^{s_b}}(u)|$. This is a lower bound on the number of vertices $v$ such that $(A(v) \trianglerighteq_{s_a} A(u))$ and $\neg(B(v) \triangleright_{s_b} B(u))$. For each such vertex $v$, neither $(u,v)$ nor $(v,u)$ contributes to $Supp_{all}(P)$. If we sum these numbers over all vertices from $V$, we get a lower bound on the number of ordered pairs that cannot support $P$. Since every ordered pair of vertices $(u,v)$ is taken into account twice, we need to divide it by 2 to get a lower bound on the pairs of vertices that do not contribute to the support of $P$. Finally, we divide the upper bound by $\binom{n}{2}$. □

Observe that this upper bound on $Supp_{all}$ is very convenient since its computation is in $O(|V|)$, whereas the computation of $Supp_{all}$ is in $O(|V|^2)$. On the one hand, it requires storing two additional values for every descriptor and every vertex (the end points of the ranking intervals). On the other hand, since we are enumerating descriptors and not descriptor values (as in item set mining) this is not costly in terms of memory usage.

## 5.2 Algorithm

TopGraphMiner computes frequent topological patterns and their top $k$ representative vertices from an attributed graph (see Algorithms 1 and 2). It takes in input the graph $G = (V, E, L)$ and two parameters: $minsup$ and $k$. In Line 1 of Algorithm 1, it performs the computation of topological vertex properties. The computation of topological patterns is done in an ECLAT-based way [32]. More precisely, all the subsets of a pattern $P$ are always evaluated before $P$ itself. In this way, by storing all frequent patterns in the hash-tree $\mathcal{M}$, the antimonotonic frequency constraint is fully checked on the fly (Line 4, in Algorithm 2). We start by enumerating the singleton positive descriptors to avoid the generation of duplicate patterns. Larger patterns are recursively generated by the function EXTEND_PATTERN (see Line 13, in Algorithm 1). We compute the upper bound on the support to prune nonpromising topological patterns (function COMP_UB in Line 8 of Algorithm 1). This function is the strict application of Theorem 1 (see supplementary material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2012.154, for the pseudocode). When this upper bound is greater than the minimum threshold, the exact support is computed (function COMP_SUPP in Algorithms 1 and 2). This step and its optimization are discussed in the following section.

**Algorithm 1.** TopGraphMiner

```
    Require: $G = (V, E, L)$, $minsup$, $k$
    Ensure: $\mathcal{M}$: the frequent topological patterns and
            their top $k$ representative vertices.
 1: Compute $T$, the set of topological properties of $G$
    that associate a numerical value to vertices of $V$
    based on the relation $E$.
 2: $\mathcal{D} \leftarrow T \cup L$
 3: $\mathcal{M} \leftarrow \emptyset$
 4: for all $D \in \mathcal{D}$, in descending order do
 5:     for all $v \in V$ do
 6:         Compute $\overline{\rho}(D(v))$ and $\underline{\rho}(D(v))$.
 7:     end for
 8:     $UB \leftarrow$ COMP_UB($\{D^+\}, \overline{\rho}, \underline{\rho}$)
 9:     if ($UB \geq minsup$) then
10:         $(supp, topk) \leftarrow$ COMP_SUPP($\{D^+\}, k$)
11:         if ($supp \geq minsup$) then
12:             $\mathcal{M} \leftarrow \mathcal{M} \cup (\{D^+\}, topk)$
13:             EXTEND_PATTERN($\{D^+\}$)
14:         end if
15:     end if
16: end for
```

**Algorithm 2.** Extend_Pattern

```
    Require: $P$ a topological pattern, $minsup$, $k$, $\overline{\rho}$, $\underline{\rho}$
    Ensure: Compute all frequent extensions of $P$ and
        add them to the global variable $\mathcal{M}$ with their top
        $k$ representative vertices
 1: for all $B \in \mathcal{D}$, $B$ greater than the last descriptor in
    $P$ do
 2:   for all $s \in \{+, -\}$ do
 3:       $Q \leftarrow P \cup \{B^s\}$
 4:       if ($\forall R \subset Q$, $R \in \mathcal{M}$) then
 5:           $UB \leftarrow \min\{$COMP_UB($Q, \overline{\rho}, \underline{\rho}$),
              COMP_DEDUC($Q, \mathcal{M}$)$\}$
 6:           if ($UB \geq minsup$) then
 7:               $(supp, topk) \leftarrow$ COMP_SUPP($Q, k$)
 8:               if ($supp \geq minsup$) then
 9:                   $\mathcal{M} \leftarrow \mathcal{M} \cup (Q, topk)$
10:                   Extend_Pattern($Q$)
11:               end if
12:           end if
13:       end if
14:   end for
15: end for
```

Another optimization is based on the deduction of the support from already evaluated patterns (function COMP_DEDUC in Line 5 of Algorithm 2). A pair of vertices that supports a pattern $P$ can support pattern $PA^+$ or pattern $PA^-$, or none of them. Thus, another upper bound on $Supp_{all}(PA^-)$ is $Supp_{all}(P) - Supp_{all}(PA^+)$. Note that these patterns have already been considered before the evaluation of $PA^-$. So, to be stringent, we bound the support by taking the minimum between this value and the upper bound defined in Theorem 1 (see Line 5 in Algorithm 2). When computing the support of the pattern, the top $k$ representative vertices are also identified (see Section 5.3.2).

## 5.3 Discussion and Optimizations

We discuss other optimizations used in TopGraphMiner algorithm and how emerging topological patterns are computed.

### 5.3.1 Computation of $Supp_{all}$

The support of $P$ is evaluated by function COMP_SUPP that counts the number of pairs of vertices $(u,v)$ such that

TABLE 1
Main Characteristics of the Graphs DBLP, MOVIES, PATENTS, and GENES

| Attributed graph | DBLP | | | MOVIES | | | PATENTS | | | GENES | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| #Vertices | $42,252$ | | | $5,972$ | | | $24,282$ | | | $4,711$ | | |
| #Edges | $210,320$ | | | $64,338$ | | | $100,246$ | | | $6,036$ | | |
| #Vertex attributes | 29 | | | 5 | | | 10 | | | 348 | | |
| Density | $2 \times 10^{-4}$ | | | $3.6 \times 10^{-3}$ | | | $1.7 \times 10^{-4}$ | | | $0.54 \times 10^{-3}$ | | |
| #Connected Comp. | 577 | | | 33 | | | 67 | | | 11 | | |
| #Communities | 1016 | | | 56 | | | 169 | | | 30 | | |
| Topo. prop. | Max | Mean | Std. Dev. | Max | Mean | Std. Dev. | Max | Mean | Std. Dev. | Max | Mean | Std. Dev. |
| Raw degree | 304 | 9.73 | 14.22 | 118 | 21.16 | 19.13 | 186 | 4.98 | 5.47 | 68 | 2.28 | 6.66 |
| DEGREE | $7.3 \times 10^{-3}$ | $2.4 \times 10^{-4}$ | $3.4 \times 10^{-4}$ | $2.2 \times 10^{-2}$ | $4 \times 10^{-3}$ | $3.5 \times 10^{-3}$ | $2.5 \times 10^{-2}$ | $3.6 \times 10^{-4}$ | $5.6 \times 10^{-4}$ | 0.04 | $1.75 \times 10^{-3}$ | $4.5 \times 10^{-3}$ |
| CLUST | 1 | 0.31 | 0.29 | 1.57 | 0.34 | 0.26 | – | – | – | 1.69 | 0.06 | 0.18 |
| NBQC | $4.6 \times 10^{5}$ | $2.2 \times 10^{2}$ | $7.8 \times 10^{3}$ | 503 | 2.96 | 19.93 | – | – | – | 22 | 0.16 | 1.26 |
| SZQC | 35 | 2.75 | 4.83 | 52 | 13.87 | 11.35 | – | – | – | 46 | 0.84 | 4.96 |
| SZCOM | $9,342$ | 40.67 | $5 \times 10^{2}$ | $1,563$ | $11.5 \times 10^{2}$ | $5.6 \times 10^{2}$ | $8,178$ | $50.9 \times 10^{2}$ | $25.9 \times 10^{2}$ | 394 | 48.73 | 93.9 |
| CLOSE | 1 | 0.024 | 0.137 | 1 | 0.010 | 0.099 | 1 | 0.005 | 0.067 | 1 | $4 \times 10^{-3}$ | 0.06 |
| BETW | $2.6 \times 10^{6}$ | $1.4 \times 10^{5}$ | $5.7 \times 10^{5}$ | $1.6 \times 10^{5}$ | $1.1 \times 10^{4}$ | $1.6 \times 10^{4}$ | $20.2 \times 10^{6}$ | $10.8 \times 10^{4}$ | $40.4 \times 10^{4}$ | $1.4 \times 10^{5}$ | $1.4 \times 10^{3}$ | $5.5 \times 10^{3}$ |
| EGVECT | 0.003 | $2.36 \times 10^{-5}$ | $9.91 \times 10^{-5}$ | $8.4 \times 10^{-3}$ | $1.6 \times 10^{-4}$ | $7.5 \times 10^{-4}$ | $11.6 \times 10^{-3}$ | $4.11 \times 10^{-5}$ | $2.8 \times 10^{-4}$ | 0.021 | $2.00 \times 10^{-4}$ | $2 \times 10^{-3}$ |
| PAGERANK | 21.53 | 0.98 | 0.98 | 0.59 | 0.88 | 0.59 | 35.98 | 0.93 | 0.91 | 7.69 | 0.31 | 0.62 |

$\forall A^{s_a} \in P,\ A(u) \rhd_{s_a} A(v)$. This computation requires to perform a quadratic operation on the number of vertices. However, as proposed in [5], a more directed search for all vertices that have smaller or greater values on all descriptors in $P$ is implemented by using range trees and it enable good performances when $|P|$ is not too large. For a singleton pattern $\{D^+\}$, the range tree is simply a binary search tree where each node contains a value $x$ of $D$ along with two values: $y^+$, that is, the number of vertices that is lower than or equal to $x$, and $y^-$, that is, the number of vertices having a value greater or equal to $x$. Then, to compute the support of $\{D^+\}$, we simply loop over the vertices of the graph, find their corresponding nodes in the range tree and sum the $y^+$ values of their left subtrees. When extending a pattern $P$, every node in the range tree is expanded to contain a nested range tree that corresponds to the added descriptor. To compute the support, we loop over the graph vertices, find their corresponding nodes in the inner range trees and sum up the $y^+$ (resp. $y^-$) values for positive (resp. negative) descriptors of their left (resp. right) subtrees.

#### 5.3.2 *Computation of the Top $k$ Representatives*

As explained in Section 4, the vertex pairs $S(P)$ that support a topological pattern $P$ define a transitive acyclic directed graph $G_P = (V, S(P))$ (see Property 2) that admits at least one topological ordering of its vertices. The top $k$ representative vertices are the $k$ last vertices with respect to one of these orderings.

**Property 3.** *Let $G = (V, A)$ be a transitive directed graph and let $Deg^-(v)$ be the incoming degree of the vertex $v \in V$ ($deg^-(v) = |\{\forall u \in V$ such that $(u, v) \in A\}|$). For any arc $(u, v) \in A$, $deg^-(u) \leq deg^-(v) + 1$.*

**Proof.** Given an arc $(u, v) \in A$, $\forall t \in V$ such that $(t, u) \in A$, by transitivity of $G$ there exists an arc $(t, v) \in A$. Therefore, $deg^-(u) \leq deg^-(v) + 1$. □

As a result, ordering $V$ with respect to $deg^-$ constitutes a topological sorting of $G_P$. The range trees used for computing the support of $P$ is exploited to retrieve the top $k$ representative vertices of $P$: during the loop over the vertices of the graph, their incoming degree is considered and the set of $k$ vertices having the largest incoming degree is maintained in a heap, using operations in $O(\log k)$.

#### 5.3.3 *Computation of $Supp_{C^r}$, $Supp_E$, and $Gr$*

Emerging topological patterns can easily be computed by adapting Algorithm 1: the selected descriptor $C^r$ is the last one in the pattern being enumerated (in the ECLAT enumeration fashion, the last descriptor in the pattern is the first to be enumerated), and when enumerated, its support provides $Supp_{C^r}(P)$. When subtracting this value from the support of its direct ancestor, it provides $Supp_{\overline{C^r}}(P)$. We, therefore, retrieve only those patterns with a growth-rate higher than a threshold. The computation of $Supp_E(P)$ can be done in a time complexity proportional to the number of edges in the graph. Finally, $Gr(P, E)$ can be deduced from $Supp_E(P)$ and $Supp_{all}(P)$.

## 6 PERFORMANCE STUDY

We report experimental results to illustrate the interest of our approach. We start by describing the four attributed graphs we use in our experiments. Then, we provide a performance study. Qualitative results are given in the next section. All experiments were performed on a cluster. Nodes are equipped with 16 processors at 2.5 GHz and 16 GB of RAM under Linux operating systems. TopGraphMiner algorithm is implemented in C.

### 6.1 Real-World Attributed Graphs

We considered four real-world attributed graphs whose characteristics are given in Table 1:

1. DBLP: This coauthorship graph is built from the DBLP digital library. Each vertex represents an author who published at least one paper in one of the major conferences and journals of the data mining and database communities[1] between January 1990 and February 2011. Each edge links two authors who coauthored at least one paper (no matter the conference or journal). The vertex properties are the number of publications in each of the 29 selected conferences or journals.
2. MOVIES: Each vertex of this graph represents a movie and an edge exists between two movies if

[1] 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 International Systems, SIGKDD Exploration, Communication ACM, IDA Journal, KAIS, SADM, PVLDB, VLDB Journal, ACM TKDD.

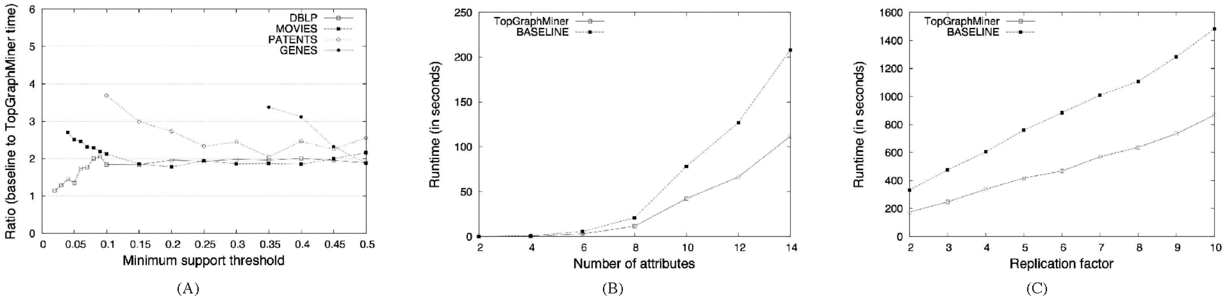

Fig. 3. Comparison w.r.t. a baseline technique: (A) execution time ratio, (B) execution time w.r.t. the number of descriptors (MOVIES, $\text{minsup} = 20\%$), and (C) execution time w.r.t. a replication factor (MOVIES, $\text{minsup} = 20\%$).

they have an actor in common.[2] The vertex attributes are based on movie ratings from Netflix customers: the number of ratings, their average and standard deviation values, the release year of the movie and its number of actors.

3. PATENTS: It is a graph derived from a subset of the citation graph of US patents granted between January 1963 and December 1999.[3] We selected only patents of the subcategory "Computer Peripherals." There are 10 vertex attributes as, for example, the grant year and the corresponding number of claims.
4. GENES: This graph contains gene-gene interactions [30], that is, each vertex stands for a gene and an edge links two vertices if they are known to interact during the biological transcription process. The vertex attributes associated with each gene are its expression values in each of 348 biological situations. Those situations are as many human tissues from several organs that are healthy or cancerous.

The main characteristics of these graphs are reported in Table 1. All these properties have a minimum value of 0. Many of these properties have a standard-deviation greater than their average, suggesting that they follow power law distributions. The computation of the topological descriptor values in these networks take few hours. For instance, the computation of centrality measures in DBLP, which is the most expensive, takes around 4 hours. Note that we do not compute NBQC, SZQC, and CLUST for the attributed graph PATENTS, since it is a directed graph and, as such, there are very few dense quasi-cliques and triangles.

## 6.2 Performance Study

### 6.2.1 Comparison with a Baseline Approach

Since there is no other algorithm that simultaneously computes up and down covariations using the same support measure as in our approach, we first study the performance of TopGraphMiner by comparing it with a baseline approach. It consists in using the algorithm of [5], which only computes up covariations, after having duplicate and reverse each descriptor. For instance, the vertex ranked first with respect to the descriptor $D^+$ is ranked last with respect to $D^-$. Notice that nonsensible patterns, such as $\{D^+, D^-\}$, will be discarded in linear time since their support is 0. Besides, it is necessary to postprocess the output patterns to remove the symmetrical patterns. This additional step is quadratic in the size of the output and can be computationnaly expensive. However, for these experiments, we do not take into account the execution time of this postprocessing step.

Fig. 3A gives the ratio of the execution time of the baseline approach to the execution time of our approach on the four attributed graphs. We can see that for the graphs MOVIES, PATENTS, and GENES, our approach is at least twice as faster as the baseline. Besides, the lower the support, the higher this ratio is. Notice that we were not able to compute topological patterns for low support values on the graph GENES, since there are many vertex attributes. This behavior shows that our approach is more efficient than the baseline one and that this efficiency does not only rely on the fact that the number of descriptors of the graphs is twice as smaller than the one used by the baseline approach, but also on the pruning capability. With the DBLP graph, however, the ratio decreases for lower supports. This can be explained by the fact that there are many nonfrequent topological patterns with negative signs that are early pruned by the baseline approach. Fig. 3B shows the execution time spent by both algorithms with respect to different numbers of randomly chosen original descriptors from the MOVIES graph, with minimum support of 20 percent. We can observe that our approach outperforms the baseline one and the gain is more important when the number of descriptors increases. Fig. 3C gives the execution time spent by both algorithms with respect to the number of vertices in the attributed graph MOVIES, with minimum support of 20 percent (the $x$-axis gives the replication factor). We can notice that TopGraphMiner is faster than the baseline approach and this especially as the number of vertices increases. Although the computation of the support of the patterns is quadratic in the number of vertices, the execution times do not increase accordingly due to the use of the range trees. We can, therefore, conclude that the results shown in Fig. 3A are more influenced by the number of descriptors than that of vertices.

### 6.2.2 Empirical Complexity of TopGraphMiner

Figs. 4A and 4B present, respectively, the execution time of TopGraphMiner and the number of obtained frequent patterns according to the minimum support threshold. The execution time is strongly related to the number of frequent topological patterns, even if the computation of the support may impact the execution time when the number of vertices is high. For example, for minimum supports greater

[2] http://www.imdb.com/.
[3] http://www.nber.org/patents/.

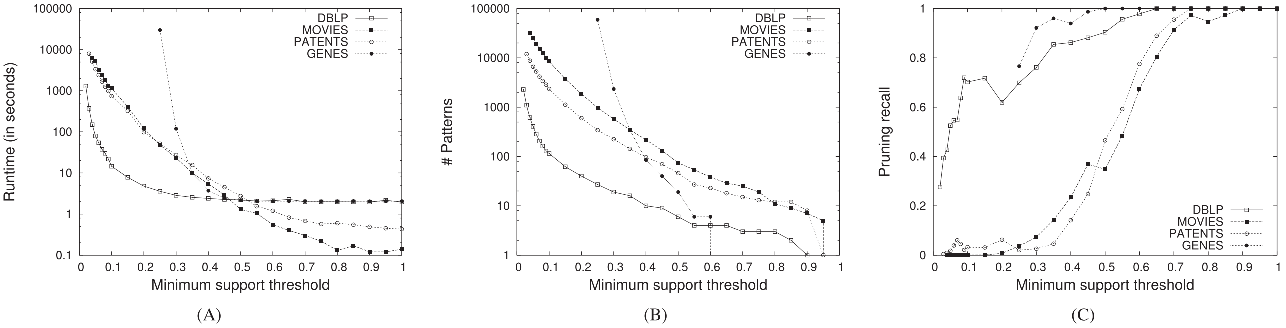

Fig. 4. (A) Execution time, (B) number of patterns, and (C) pruning recall w.r.t. minimum support threshold.

than 60 percent, the number of frequent patterns in the graphs MOVIES and PATENTS is greater than that in DBLP. Nevertheless, the extraction of the patterns is faster in the former two since they have fewer vertices than the latter.

Regarding the efficiency of our pruning technique (i.e., the upper bound on the support), Fig. 4C gives, for every minimum support, the ratio of the number of pruned patterns, thanks to (1), to the number of patterns that, in the end, turned out to be indeed nonfrequent. In other words, it gives the recall of our pruning technique. It can be seen from this figure that the technique is very efficient for high minimum support values on the four attributed graphs. In fact, when the minimum support is higher than 70 percent, almost all nonfrequent patterns are pruned without computing their exact support.

However, for lower support values, we observe that the upper bound is less stringent on MOVIES and PATENTS, while keeping its performance for the graphs GENES and DBLP. This behavior can be explained by the fact that the upper bound exploits tie values, whose ratio is higher in the latter two. Let us consider the ratio of tie values defined as

$$Ties(G) = \frac{\sum_{D\in\mathcal{D}} \sum_{u,v\in V^2 u<v} D(u) = D(v)}{|\mathcal{D}|.\binom{n}{2}}.$$

In the four real-world attributed graphs, this ratio is

| Graphs | DBLP | MOVIES | PATENTS | GENES |
|---|---|---|---|---|
| $Ties(G)$ | 0.8 | 0.16 | 0.32 | 0.69 |

The presence of many tie values in DBLP and GENES descriptors may explain the robustness of the upper bound with respect to the minimum support threshold.

Fig. 5 shows the execution time of TopGraphMiner with and without using the upper bound. To show that the use of the upper bound is more and more advantageous as the number of vertices of the graph grows, we plot the execution time of TopGraphMiner on GENES graph with respect to a replication factor. We can observe that the use of the upper bound reduces the execution time and that this difference increases with the number of vertices. Note that this difference is lightened by the use of range trees, which reduce the impact of the computation of unpromising patterns.

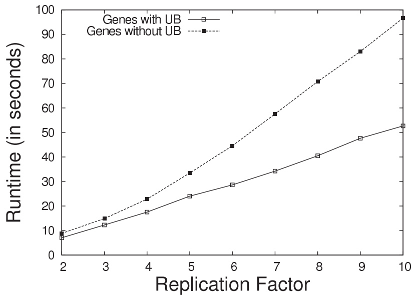

Fig. 5. Impact of the upper bound on the execution time with respect to a replication factor (GENES, $minsup = 40$ percent).

## 7 Case Studies

We now analyze the effectiveness of our approach on the real-world attributed graphs.

### 7.1 Tell Us Where You Publish, We Tell You How Important You Are

We examine the results obtained by TopGraphMiner on the DBLP attributed graph regarding the following questions:

- Are there any interesting patterns among publications?
- Are there interesting trends between some authors' publications and topological properties?
- What about IEEE TKDE authors?

Before extracting topological patterns with TopGraphMiner, we compute correlations between descriptors. The resulting correlation matrix is reported in Fig. 6A. The vertex attributes that have a correlation higher than 0.7 are VLDB, ICDE, and SIGMOD. The most correlated topological properties are, on the one hand, BETW, DEGREE, and PAGERANK and, on the other hand, SZQC and NBQC. The vertex attributes and the topological properties that are not correlated with any other (with a correlation always lower than 0.2) are: SAC, Communication of

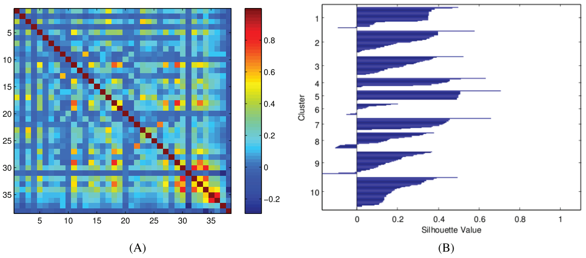

Fig. 6. (A) Correlation matrix between vertex attributes (1 to 29) and topological properties (30 to 38) in DBLP. (B) Silhouette plot of the K-means clustering on some topological patterns.

TABLE 2
Most Frequent Vertex Attributes in Clusters of Patterns Found in the DBLP Attributed Graph

| Cluster | # patterns | Most frequent vertex attributes | Cluster | # patterns | Most frequent vertex attributes |
|---|---|---|---|---|---|
| 1 | 32 | SAC$^-$, IJCAI$^+$ | 6 | 17 | KAIS$^+$, SDM$^+$, PAKDD$^+$, KDD$^+$ |
| 2 | 34 | CIKM$^+$, PAKDD$^+$ | 7 | 18 | IEEE TKDE$^+$ |
| 3 | 28 | SIGMOD$^+$ | 8 | 24 | VLDB$^+$, VLDBJ$^+$, PVLDB$^+$ |
| 4 | 15 | AAAI$^+$ | 9 | 34 | ICDM$^+$, PKDD$^+$, KDD$^+$ |
| 5 | 15 | CommACM$^+$ | 10 | 46 | ICDE$^+$, SIGMOD$^+$, TKDE$^+$, VLDB$^+$ |

TABLE 3
Top Topological Patterns in the DBLP Attributed Graph

| $P$ | Descriptors | Measures | Top-5 Representative vertices |
|---|---|---|---|
| $P_{all}$ | SAC$^+$, SZCOM$^-$ | $Supp_{all} = 0.19$<br>$Gr(P, E) = 0.21$ | #1 F. N. Sibai, #2 M. M. Huntbach, #3 C. Leopold, #4 A. J. Duben, #5 P. Rittgen, |
| $P_{\text{PAGERANK}^+}$ | ICDE$^+$, DEGREE$^+$, BETW$^+$, CLUST$^-$, NBQC$^+$, SZQC$^+$ | $Gr(P, \text{PAGERANK}^+) = 253,933$<br>$Gr(P, E) = 4.8$<br>$Supp_{all} = 0.12$ | #1 H. Garcia-Molina, #2 M. Stonebraker, #3 G. Weikum #4 R. Agrawal, #5 M. J. Franklin, |
| $P_E$ | PVLDB$^+$, DEGREE$^+$, BETW$^+$ | $Gr(P, E) = 6.9682$ | #1 G. Weikum, #2 J. Han, #3 D. Maier #4 P. S. Yu, #5 H. Garcia-Molina, |

ACM, IEEE International Systems, CLOSE and CLUST. These correlation measures will help us in the interpretation of the following results.

### 7.1.1 Topological Patterns on Conferences and Journals

Let us first consider topological patterns among publications venues. Mining all frequent topological patterns with a support threshold of 1 percent takes 68 seconds. The output contains 263 topological patterns, from which 58 (22 percent) involve negatively signed attributes. To better understand the type of information retrieved by these 263 patterns, we performed a clustering analysis of the topological patterns. We use K-means algorithm on the $263 \times 57$ Boolean matrix where the rows correspond to the patterns and the columns to the signed vertex attributes $(2 \times 29 - 1)$. We use the cosine distance and employ the silhouette plot to determine the number of clusters. It suggests 10 clusters (see Fig. 6B). The most frequent vertex attributes of each cluster are shown in Table 2, that is the vertex attributes that appear in at least in 2/3 of the cluster patterns. We can observe that the majority of the clusters are homogeneous, referring either to data mining or to database publications. For instance, Clusters 1, 2, 6, and 9 refer to data mining publications, while Clusters 3, 8, and 10 clearly refer to database publications. Other clusters are related to a specific conference/journal.

Interestingly, 20 of these patterns contain the attribute $SAC^-$ together with positively signed attributes. Examples of such patterns are $\{\text{SAC}^-, \text{KDD}^+\}$, $\{\text{SAC}^-, \text{ECML/PKDD}^+\}$, $\{\text{SAC}^-, \text{VLDB}^+\}$, and $\{\text{SAC}^-, \text{SIGMOD}^+\}$. This type of pattern can be explained by the fact that SAC scope is larger than that of the other selected conferences, which are more focused either on database or data mining topics. Since the topics covered by SAC are much more general (e.g., programming languages, geometric constraints and reasoning, and applied biometrics), it is not surprising that many authors that have several publications in SAC conference series have none or few publications in the data mining or database areas.

### 7.1.2 Are There Interesting Trends between Author Publications and Topological Properties?

Table 3 reports the most frequent topological pattern ($P_{all}$), the most emerging pattern ($P_{\text{PAGERANK}^+}$) with respect to PAGERANK$^+$ and the most structurally correlated topological pattern ($P_E$). $P_{all}$ is formed by descriptors SAC$^+$ and SZCOM$^-$. Its meaning is that SAC authors tend to belong to small communities, that is, these authors are rather isolated in the graph as illustrated in Fig. 7A, where the top-10 representative vertices and their direct neighborhoods are displayed. These vertices have a low degree. As mentioned in Section 7.1.1, the scope of the SAC conference is much wider than database and data mining topics. This makes this pattern sensible and justifies that 1) this pattern is not much correlated to the graph structure ($Gr(P, E) = 0.21$), and 2) its top-five supporting vertices are mostly researchers from software engineering and network areas.

The computation of emerging patterns with respect to PAGERANK, with a support threshold of 1 percent and a growth-rate threshold of 3, takes around 6 hours and produces 4,313 patterns. The most emerging pattern $P_{\text{PAGERANK}^+}$ (see Table 3) contains many topological properties with a positive sign, except CLUST, which has a negative sign. As we have seen before, PAGERANK is highly

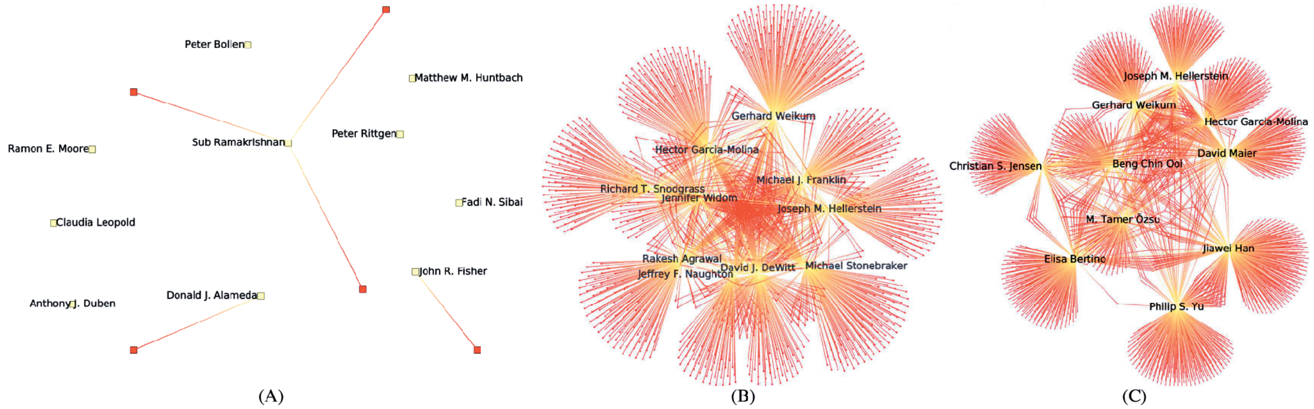

Fig. 7. (A) Top 10 vertices supporting $P_{all}$, (B) $P_{\text{PAGERANK}}$, and (C) $P_E$ and their connected vertices in DBLP.

TABLE 4
Top-Five "Impacting" Publications in the Emergence of $\mathrm{DEGREE}^+$ and $\mathrm{BETW}^+$ w.r.t. $\mathrm{PAGERANK}^+$ (A) along with Their Top-Five Authors (B)

(A)

| Rank | $P_{Topo_1}$ | | $P_{Topo_2}$ | |
|---|---|---|---|---|
| | Publication | Factor | Publication | Factor |
| 1 | ECML/PKDD$^+$ | 2.5 | PVLDB$^+$ | 5.67 |
| 2 | IEEE TKDE$^+$ | 2.28 | EDBT$^+$ | 5.11 |
| 3 | PAKDD$^+$ | 2.21 | VLDB J.$^+$ | 4.35 |
| 4 | DASFAA$^+$ | 2.09 | SIGMOD$^+$ | 4.25 |
| 5 | ICDM$^+$ | 1.95 | ICDE$^+$ | 3.42 |

(B)

| PageRank$^+$ Degree$^+$<br>ECML/PKDD$^+$ | PageRank$^+$ Betw$^+$<br>PVLDB$^+$ |
|---|---|
| Christos Faloutsos | Gerhard Weikum |
| Jiawei Han | Jiawei Han |
| Philip S. Yu | David Maier |
| Bing Liu | Philip S. Yu |
| C. Lee Giles | Hector Garcia-Molina |

correlated with DEGREE and BETW. Therefore, it is not surprising that both appear in the pattern. On the other hand, the presence of the property $\mathrm{CLUST}^-$ suggests that the higher the PAGERANK of the authors (and consequently their DEGREE and BETW), the lower the connectivity of their coauthors. In other words, authors with high PAGERANK have many coauthors that do not publish together. This can be observed on Fig. 7B where the connectivity between coauthors of the top-10 representative vertices is low. Those that advise many PhD students can be seen as typical examples of these authors.

The most structurally correlated topological pattern $P_E$ gathers the descriptors $\mathrm{PVLDB}^+$, $\mathrm{DEGREE}^+$, and $\mathrm{BETW}^+$. PVLDB is at the same time a well-established conference and journal in the data mining and database communities. This pattern is strongly structurally correlated ($Gr(P, E) > 5$), i.e., it tends to be more supported by pairs that are edges than arbitrary pairs of vertices. Fig. 7C displays its top-10 representative vertices.

We can also use emerging topological patterns, made only of topological properties, to compare the relative importance of conferences and journals. Let us consider

$$P_{Topo_1} = \{\mathrm{PAGERANK}^+, \mathrm{DEGREE}^+\} \quad \text{and}$$
$$P_{Topo_2} = \{\mathrm{PAGERANK}^+, \mathrm{BETW}^+\},$$

two such emerging patterns whose respective growth-rates are $Gr(P_{Topo_1}, \mathrm{PAGERANK}^+) = 124.69$ and $Gr(P_{Topo_2}, \mathrm{PAGERANK}^+) = 584.46$. These emerging patterns reveal which conferences or journals are more related to the topological properties $\mathrm{BETW}^+$ and $\mathrm{DEGREE}^+$. To that end, for each publication venue $C$ and both emerging patterns $P_{Topo_1}$ and $P_{Topo_2}$, we compute the ratio $\frac{Gr(P_{Topo_i} C, \mathrm{PAGERANK}^+)}{Gr(P_{Topo_i}, \mathrm{PAGERANK}^+)}$. Table 4A gives the top-five publications with respect to this ratio. Surprisingly, we observe that data mining conferences have a higher impact on the pattern $\{\mathrm{PAGERANK}^+, \mathrm{DEGREE}^+\}$, while database conferences positively influence the growth-rate of the pattern $\{\mathrm{PAGERANK}^+, \mathrm{BETW}^+\}$. Since data mining intersects many other research areas, these results may be explained by the fact that data mining authors may also publish with many others from different areas, such as database and machine learning ones. On the other hand, as database is an older well-established research field, database authors tend to appear at the center of the graph. For the most impacting publications, we identify the top-five representative authors. They are shown in Table 4B.

TABLE 5
Patterns Found in Movies and Their Top-Five Movies

| $P$ | Descriptors | Measures | Top-5 movies |
|---|---|---|---|
| $P_1$ | AVG_RATING$^+$<br>NB_RATINGS$^+$ | $Supp_{all} = 0.7$<br>$Gr(P, E) = 1.05$ | #1 The Green Mile, #2 Forrest Gump, #3 The Sixth Sense, #4 Indiana Jones and the Last Crusade, #5 Gladiator |
| $P_2$ | NB_RATINGS$^+$<br>Close$^+$ | $Supp_{all} = 0.6$<br>$Gr(P, E) = 0.87$ | #1 The Rock, #2 Fahrenheit 9/11, #3 The Godfather, #4 Enemy of the State, #5 Men of Honor |
| $P_3$ | STD_RATING$^+$<br>PageRank$^-$ | $Supp_{all} = 0.58$<br>$Gr(P, E) = 0.89$ | #1 There's no Business Like Show Business, #2 Michael Moore Hates America, #3 Digimon: The Movie, #4 Blown Away, #5 Benjamin Smoke |
| $P_4$ | YEAR$^+$<br>AVG_RATING$^-$ | $Supp_{all} = 0.57$<br>$Gr(P, E) = 0.94$ | #1 Day of the Dead 2: Contagium, #2 raging sharks, #3 My Big Phat Hip Hop Family, #4 The Fallen Ones, #5 Last Days |

### 7.1.3 What about the IEEE TKDE Authors?

We also look for the emerging patterns with respect to the attribute IEEE TKDE, with support threshold of 1 percent and growth-rate threshold of 3 (their computation takes around 5 hours). We obtain 745 emerging patterns with respect to the class IEEE $\mathrm{TKDE}^+$. The most emerging pattern is $P_{\mathrm{TKDE}} = \mathrm{ICDE}^+, \mathrm{VLDB}^+, \mathrm{BETW}^+, \mathrm{PAGERANK}^+$, with $Gr(P_{\mathrm{TKDE}}, \mathrm{TKDE}^+) = 11.75$. This pattern indicates that authors publishing in IEEE TKDE journal tend also to publish papers in the conferences ICDE and VLDB. $\mathrm{BETW}^+$ suggests that these authors are located at the center of the coauthorship graph, while $\mathrm{PAGERANK}^+$ means that they coauthored papers with other researchers that also appear at the center of the graph. It is important to observe that this pattern is also highly structurally correlated ($Gr(P_{\mathrm{TKDE}}, E) = 6.5758$). Furthermore, this pattern is sensible since it is supported by well-established researchers in the Database community: Christos Faloutsos, Jiawei Han, Philip S. Yu, Beng Chin Ooi, and Hector Garcia-Molina are its top-five representative authors.

## 7.2 Do We Only Appreciate Blockbusters?

Let us now consider the real-world attributed graph MOVIES. Table 5 shows the four most frequent topological patterns (with at least two descriptors) with their top-five representative movies. Pattern $P_1$ suggests that Netflix users tend to rate movies they like. Its top-10 representative movies are connected (see Fig. 8A), which indicates they have at least one actor in common. The second pattern $P_2$ reveals that many users tend to rate movies located at the center of the graph, that is, movies with "major" actors (e.g., R. de Niro, S. Connery, T. Hanks, B. Willis, H. Ford, etc.). Therefore, the supporting vertices of this pattern is made of major blockbusters (see Fig. 8B). Pattern $P_3$ indicates that controversial movies (those with a high rating standard deviation) tend to be isolated within the graph (lower PAGERANK): they are more independent films without well-known actors. Note that all the supporting movies of this pattern have a degree of 0. Finally, pattern $P_4$ suggests

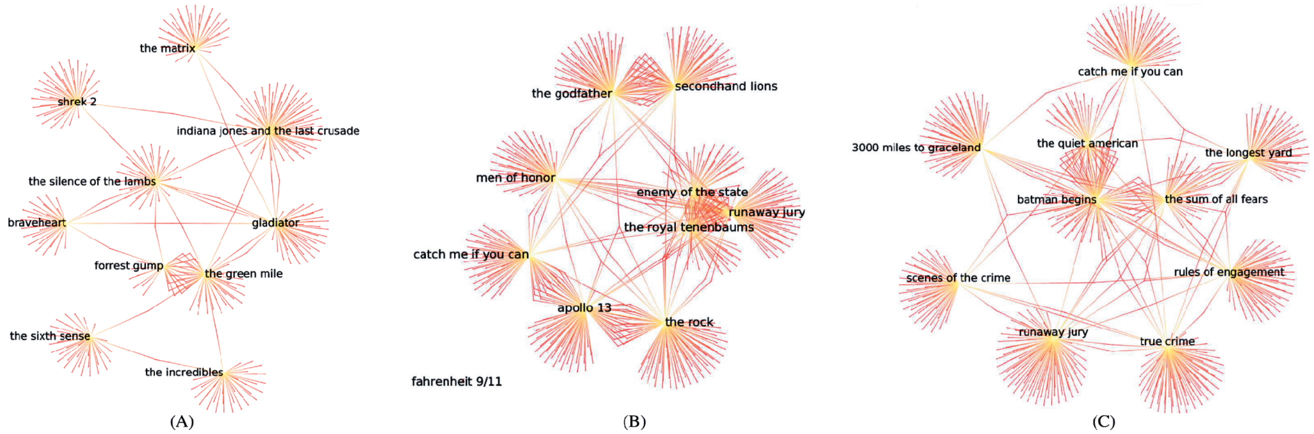

Fig. 8. (A) Top 10 vertices supporting $P_1$, (B) $P_2$, and (C) $P_E$ and their connected vertices in MOVIES.

that older movies are better rated. This can be due to the fact that the ratings were given between 1998 and 2005. Therefore, Netflix users tend to rate only noncontemporary movies they like and to forget those they did not like over time.

Table 6 shows the most emerging topological pattern with respect to the PAGERANK and the most structurally correlated pattern. Pattern $P_{\text{PAGERANK}}$ gathers descriptors NB_ACTORS$^+$ and all the centrality measures, plus STD_RATING$^-$ and NB_RATINGS$^+$. As the edges of MOVIES encode the fact that two movies share at least one actor, it is not surprising that this pattern associates NB_ACTORS$^+$ with all centrality measures. Furthermore, the attribute STD_RATING$^-$ indicates that the representative movies of this pattern are consensual.

The most structurally correlated topological pattern $P_E$ reveals that recent movies (YEAR$^+$) tend to play a central role within the graph (BETW$^+$, EGVECT$^+$, PAGERANK$^+$) and their neighbors tend to be not connected (CLUST$^-$), since it is not common that several movies share the same casting. The projection of its top-10 representative vertices on the graph is given in Fig. 8C.

### 7.3 How Do Patents Cite Each Other?

We now present some topological patterns found in PATENTS. Table 7 shows the four most frequent patterns that involve vertex attributes and topological properties. The companies associated with the top-five representative vertices of these patterns are also shown. For the pattern $P_1$, all five representatives belong to the same company Canon Kabushiki Kaisha. For the other patterns, at least two of the top-five representative patents belong to the same company.

Patterns $P_1$ and $P_2$ are sensible since to have a high PAGERANK value, a vertex must have high inner or outer degree (see Figs. 9A and 9B). $P_3$ means that "the younger the patents, the lower the PAGERANK." This knowledge nugget is meaningful as older patents are more widely cited than younger ones. All its top-10 representative patents have a degree of 0. $P_4$ reveals that the higher the number of claims, the higher the PAGERANK of the patent. This can be explained by the fact that the claims of the patents may refer to many previously granted patents.

The most emerging topological pattern with respect to PAGERANK indicates that the more generic a patent is, the more its location tends to be central in the graph (see Table 8). The vertex attribute named GENERAL is related to the number of times the patent is cited by subsequent patents that belong to a wide range of fields. $P_E$ discloses the fact that the more recent a patent is, the higher the number of citations to previously granted patents, whereas it tends to be not cited and consequently its PAGERANK tends to be low (see Fig. 9C).

### 7.4 Are Topological Patterns in GENES Relevant with Respect to Prior Knowledge?

To validate our approach using the attributed graph GENES, we analyze four specific patterns given in Table 9, along with their corresponding measures. Each pattern is composed of two vertex attributes. The first attribute is a biological situation that corresponds to a healthy tissue: normal pancreas ($P_1$ and $P_2$) or normal colon ($P_3$ and $P_4$). The second attribute is a biological situation that corresponds to the same but cancerous tissue (adenocarcinoma

TABLE 6
Top Topological Patterns in MOVIES

| $P$ | Descriptors | Measures | Top-5 movies |
|---|---|---|---|
| $P_{\text{PAGERANK}+}$ | NB_ACTORS$^+$<br>STD_RATING$^-$<br>NB_RATINGS$^+$<br>DEGREE$^+$ CLOSE$^+$<br>BETW$^+$ EGVECT$^+$<br>NBQC$^+$ SZQC$^+$<br>SZCOM$^+$ | $Supp_{all} = 0.052$<br>$Gr(P, \text{PAGERANK}^+) = 11,789$<br>$Gr(P, E) = 0.32$ | #1 The Godfather,<br>#2 Crimson Tide,<br>#3 The Untouchables,<br>#4 The Hunt for Red October,<br>#5 Apollo 13, |
| $P_E$ | YEAR$^+$, BETW$^+$, EGVECT$^+$, PAGERANK$^+$, CLUST$^-$ | $Supp_{all} = 0.05$<br>$Gr(P, E) = 2.78$ | #1 Catch me if you can,<br>#2 True Crime,<br>#3 Batman Begins,<br>#4 The Quiet American,<br>#5 Scenes of the Crime, |

TABLE 7
Frequent Patterns in PATENTS with the Associated Companies of Their Top-Five Patents

| $P$ | Descriptors | Measures | Companies associated with the Top-5 patents |
|---|---|---|---|
| $P_1$ | INNER_DEGREE$^+$, PAGERANK$^+$ | $Supp_{all} = 0.62$<br>$Gr(P, E) = 1.82$ | #1 Canon Kabushiki Kaisha |
| $P_2$ | OUTER_DEGREE$^+$, PAGERANK$^+$ | $Supp_{all} = 0.59$<br>$Gr(P, E) = 0.82$ | #1 Hewlett-Packard Company, #2 Colorado Microdisplay |
| $P_3$ | GRAND_YEAR$^+$, PAGERANK$^-$ | $Supp_{all} = 0.52$<br>$Gr(P, E) = 1.28$ | #1 Nippondenso Co., #2 Canon Kabushiki Kaisha #3 Sony Co., #4 Intel Co., #4 Ricoh Company |
| $P_4$ | NB_CLAIMS$^+$, PAGERANK$^+$ | $Supp_{all} = 0.51$<br>$Gr(P, E) = 1.01$ | #1 Canon Kabushiki Kaisha, #2 National Instruments #3 VPL Research Inc., #4 Xerox Co. |

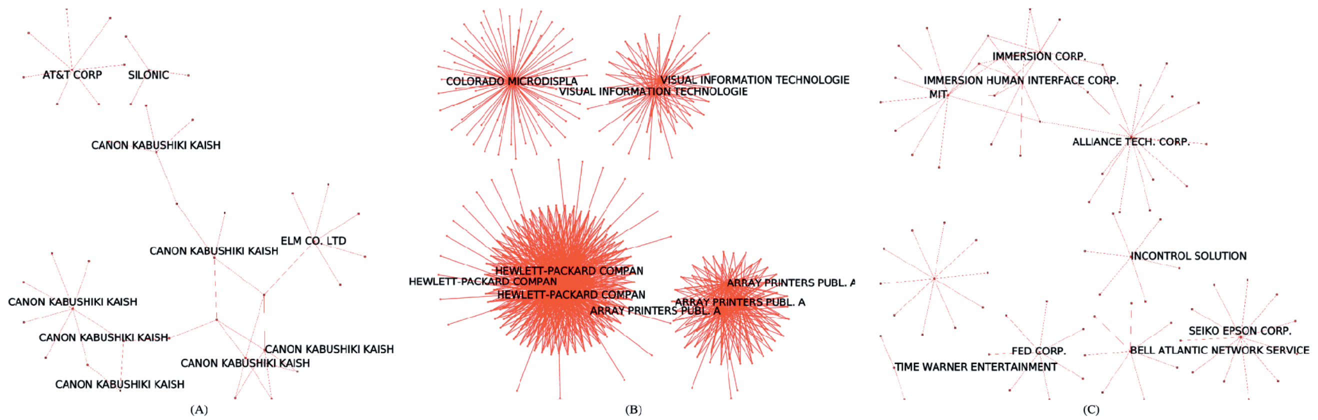

Fig. 9. (A) Top 10 vertices supporting $P_1$, (B) $P_2$, and (C) $P_E$ and their connected vertices of PATENTS.

for $P_1$ and $P_3$, and carcinoma for $P_2$ and $P_4$). Moreover, the first attribute is negatively signed, whereas the second one is positively signed. So, the most representative genes of these patterns are those that are inhibited in a normal tissue but are overexpressed in a cancerous one.

To verify whether the given patterns are sensible, we study their supporting genes. More precisely, for each pattern, we consider the ranks achieved by some of their supporting genes known to be involved in pancreatic and colon cancer. We compute the normalized average ranks of two specific sets of genes known to be overexpressed in pancreatic cancer (HLA-DRB4, PPAPDC1B, THBS1) [6] and also in colon cancer (ANXA1, GJB2, PSMC5, RPS7) [26]. The lower the ranks achieved by such genes, the more representative they are.

The computed average ranks are given in the fourth and fifth columns of Table 9. As can be observed from the fourth column, the genes defined as associated with pancreatic cancer in [6] indeed highly support the patterns related to pancreatic tissue, $P_1$ and $P_2$. On average, they are ranked above the first half of the ranks. We can also observe that the same genes are not the most representative of the patterns related to colon cancer ($P_3$ and $P_4$), since they achieved high average ranks for these patterns (above 0.8). From the fifth column of the same table, we can observe that the genes identified in [26] as associated with colon cancer are not only related to this type of cancer, but also to pancreatic cancer, as they have low average ranks for all four patterns. This is exactly what is claimed in [26]: *"the genes ANXA1, GJB2, RPS7 were ALSO identified as metastasis-specific of pancreatic metastatic tumor cells versus their nonmetastatic counterparts."* Finally, all patterns are positively correlated to the graph structure, which means that they are more supported by pairs of genes that are known to interact than arbitrary gene pairs.

## 8 Related Work

Graph mining is an active topic in data mining. In the literature, there exist two main trends to analyze graphs. On the one hand, graphs are studied at a macroscopic level by considering statistical graph properties (e.g., diameter, degree distribution) [2], [7]. On the other hand, sophisticated graph properties are discovered by using a local pattern mining approach. Recent approaches mine attributed graphs which convey more information. In such graphs, information is locally available on vertices by means of attribute values. As argued by Moser et al. [23], *"often features and edges contain complementary information, i.e., neither the relationships can be derived from the feature vectors nor vice versa."*

Attributed graphs are extensively studied by means of clustering techniques (see, e.g., [1], [8], [13], [15], [20], [33]) whereas pattern mining techniques in such graphs have been less investigated. The pioneering work [23] proposes a method to find dense homogeneous subgraphs (i.e., subgraphs whose vertices share a large set of attributes). Similar to this work, Günnemann et al. [14] propose a method based on subspace clustering and dense subgraph mining to extract nonredundant subgraphs that are homogenous with respect to vertex attributes. Silva et al. [29] extract pairs of dense subgraphs and Boolean attribute sets such that the Boolean attributes are strongly associated with the dense subgraphs. In [24], Mougel et al. propose the task of finding the collections of homogeneous $k$-clique percolated components (i.e., components made of overlapping cliques sharing a common set of true valued attributes) in Boolean attributed graphs. Another approach is presented in [19], where a larger neighborhood is considered. This pattern type relies on a relaxation of the accurate structure

TABLE 8
Top Topological Patterns in PATENTS

| $P$ | Descriptors | Measures | Top-5 companies |
|---|---|---|---|
| $P_{\text{PAGERANK}+}$ | INNER_DEGREE$^+$, GENERAL$^+$, CLOSE$^+$, BETW$^+$, PAGERANK$^+$ | $Supp_{all} = 0.02$ $Gr(P,E) = 1094.08$ | #1 VISUA #2 University of Pittsburgh #3 Sony #4 CADWARE #5 TALIGENT |
| $P_E$ | GRAND_YEAR$^+$, OUTER_DEGREE$^+$, INNER_DEGREE$^-$, PAGERANK$^-$ | $Supp_{all} = 0.03$ $Gr(P,E) = 7.47$ | #1 Immersion Corp., #2 MIT, #3 Immersion Human Interface Corp., #4 Fed Corp., #5 Time Warner Entertainement, |

TABLE 9
Four Specific Patterns in GENES

| $P$ | Pattern | Measures | PANCREAS AVG RANK | COLON AVG RANK |
|---|---|---|---|---|
| $P_1$ | PANCREAS NORMAL$^-$<br>PANCREAS ADENOCARCINOMA$^+$ | $Supp_{all} = 0.0125$<br>$Gr(P,E) = 1.877$ | 0.378 | 0.308 |
| $P_2$ | PANCREAS NORMAL$^-$<br>PANCREAS CARCINOMA$^+$ | $Supp_{all} = 0.0097$<br>$Gr(P,E) = 1.941$ | 0.510 | 0.183 |
| $P_3$ | COLON NORMAL$^-$<br>COLON ADENOCARCINOMA$^+$ | $Supp_{all} = 0.0162$<br>$Gr(P,E) = 1.586$ | 0.821 | 0.230 |
| $P_4$ | COLON NORMAL$^-$<br>COLON CARCINOMA$^+$ | $Supp_{all} = 0.0133$<br>$Gr(P,E) = 1.050$ | 0.806 | 0.306 |

constraint on subgraphs. Roughly speaking, they propose a probabilistic approach to both construct the neighborhood of a vertex and propagate information into this neighborhood. Following the same motivation, Sese et al. [27] extract (not necessarily dense) subgraph with common item sets.

Note that these approaches use a single type of topological information based on the neighborhood of the vertices. Furthermore, they do not handle numerical attributes as in our proposal. However, global statistical analysis [11] of a single graph considers several measures to describe the graph topology, but does not benefit from vertex attributes. Besides, current local pattern mining techniques on attributed graphs do not consider numerical attributes nor macroscopic topological properties. To the best of our knowledge, our paper represents a first attempt to combine both microscopic and macroscopic analysis on graphs by means of (emerging) topological pattern mining. Indeed, several approaches aim at building global models from local patterns [12], but none of them tries to combine information from different graph granularity levels.

Covariation patterns are also known as gradual patterns [9] or rank-correlated item sets [5]. Do et al. [9] use a support measure based on the length of the longest path between ordered objects. This measure has some drawbacks w.r.t. computational and semantics aspects. Calders et al. [5] introduce a support measure based on the Kendall's $\tau$ statistical measure. However, their approach is not defined to simultaneously discover up and down covariation patterns as does our approach. Another novelty of our work is the definition of other interestingness measures to capture emerging covariations. Finally, this work is also the first attempt to use covariation pattern mining in attributed graphs.

## 9 Conclusion and Future Directions

We propose TopGraphMiner, an algorithm that supports network analysis by finding regularities among vertex topological properties and attributes. It mines frequent topological patterns as up and down covariations involving both attributes and topological properties of graph vertices. In addition, we define two interestingness measures to capture the significance of a pattern with respect to either a given descriptor, or the relationship encoded by the graph edges. Furthermore, by identifying the top $k$ representative vertices of a topological pattern, we support a better interaction with end-users. Experimental results illustrate the added value of our approach. In particular, we report on four real-world case studies: a coauthorship graph built from the DBLP digital library, a graph derived from movies' characteristics, a citation graph of US patents, and a protein-protein interaction graph. These case studies show the capability of TopGraphMiner to discover sensible patterns. Our work opens several perspectives. A short-term perspective would be to extend our framework to take into account the information conveyed by categorical vertex descriptors. Another interesting perspective would be to adapt the topological pattern mining approach to dynamic graphs by, for instance, identifying unexpected topological patterns over time.

## Acknowledgments

The authors would like to thank Pr. Toon Calders and the anonymous reviewers for their comments and Julien Salotti for his help. They also gratefully acknowledge support from the CNRS/IN2P3 Computing Center. This work was partially supported by the ANR FOSTER ANR-2010-COSI-012-02.

## References

[1] L. Akoglu et al., "PICS; Parameter-Free Identification of Cohesive Subgroups in Large Graphs," *Proc. 12th SIAM Int'l Conf. Data Mining*, pp. 439-450, 2012.

[2] R. Albert and A.-L. Barabási, "Topology of Complex Networks: Local Events and Universality," *Physics Rev.*, vol. 85, pp. 5234-5237, 2000.

[3] S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, vol. 30, nos. 1-7, pp. 107-117, 1998.

[4] B. Bringmann and S. Nijssen, "What Is Frequent in a Single Graph?" *Proc. 12th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD)*, pp. 858-863, 2008.

[5] T. Calders, B. Goethals, and S. Jaroszewicz, "Mining Rank-Correlated Sets of Numerical Attributes," *Proc. 12th ACM SIGKDD Int'l Conf. Knowledge (KDD)*, pp. 96-105, 2006.

[6] D. Campagna et al., "Gene Expression Profiles Associated with Advanced Pancreatic Cancer," *Int'l J. Clinical and Experimental Pathology*, vol. 1, no. 1, pp. 32-43, 2008.

[7] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-MAT: A Recursive Model for Graph Mining," *Proc. Fourth SIAM Int'l Conf. Data Mining (SDM)*, 2004.

[8] H. Cheng, Y. Zhou, and J.X. Yu, "Clustering Large Attributed Graphs," *ACM Trans. Knowledge Discovery from Data* , vol. 5, no. 2, p. 12, 2011.

[9] T. Do, A. Laurent, and A. Termier, "Efficient Parallel Mining of Closed Frequent Gradual Itemsets," *Proc. IEEE Int'l Conf. Data Mining (ICDM)*, pp. 138-147, 2010.

[10] G. Dong and J. Li, "Efficient Mining of Emerging Patterns: Discovering Trends and Differences," *Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, pp. 43-52, 1999.

[11] L.C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35-41, 1977.

[12] J. Fürnkranz and A.J. Knobbe, "Guest Editorial: Global Modeling Using Local Patterns," *Data Mining and Knowledge Discovery*, vol. 21, no. 1, pp. 1-8, 2010.

[13] R. Ge et al., "Joint Cluster Analysis of Attribute Data and Relationship Data," *ACM Trans. Knowledge Discovery from Data*, vol. 2, no. 2, pp. 1-35, 2008.

[14] S. Günnemann et al., "Subspace Clustering Meets Dense Subgraph Mining: A Synthesis of Two Paradigms," *Proc. IEEE 10th Int'l Conf. Data Mining (ICDM)*, pp. 845-850, 2010.

[15] S. Günnemann et al., "A Density-Based Approach for Subspace Clustering in Graphs with Feature Vectors," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (PKDD)*, pp. 565-580, 2011.

[16] J. Hirsch, "An Index to Quantify an Individual's Scientific Research Output," *Proc. Nat'l Academy of Sciences USA*, vol. 102, no. 46, pp. 16569-16572, 2005.

[17] D. Jiang and J. Pei, "Mining Frequent Cross-Graph Quasi-Cliques," *ACM Trans. Knowledge Discovery from Data*, vol. 2, no. 4, pp. 1-42, 2009.

[18] U. Kang, C.E. Tsourakakis, A.P. Appel, C. Faloutsos, and J. Leskovec, "HADI: Mining Radii of Large Graphs," *ACM Trans. Knowledge Discovery from Data*, vol. 5, no. 2, p. 8, 2011.

[19] A. Khan, X. Yan, and K.-L. Wu, "Towards Proximity Pattern Mining in Large Graphs," *Proc. ACM SIGMOD Int'l Conf. Management of Data*, pp. 867-878, 2010.

[20] Z.-X. Liao and W.-C. Peng, "Clustering Spatial Data with a Geographic Constraint," *Knowledge and Information Systems*, vol. 31, no. 1, pp. 1-18, 2012.

[21] G. Liu and L. Wong, "Effective Pruning Techniques for Mining Quasi-Cliques," *Proc. European Conf. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pp. 33-49, 2008.

[22] K. Makino and T. Uno, "New Algorithms for Enumerating All Maximal Cliques," *Proc. First Scandinavian Workshop Algorithm Theory (SWAT),* pp. 260-272, 2004.

[23] F. Moser, R. Colak, A. Rafiey, and M. Ester, "Mining Cohesive Patterns from Graphs with Feature Vectors," *Proc. SIAM Int'l Conf. Data Mining (SDM),* pp. 593-604, 2009.

[24] P.-N. Mougel, C. Rigotti, and O. Gandrillon, "Finding Collections of K-Clique Percolated Components in Attributed Graphs," *Proc. 16th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD),* 2012.

[25] M.E.J. Newman, "Fast Algorithm for Detecting Community Structure in Networks," *Physics Rev. E,* vol. 69, p. 066133, 2004.

[26] V. Orian-Rousseau et al., "Genes Upregulated in a Metastasizing Human Colon Carcinoma Cell Line," *Int'l J. Cancer,* vol. 5, no. 113, pp. 699-705, 2005.

[27] J. Sese, M. Seki, and M. Fukuzaki, "Mining Networks with Shared Items," *Proc. 19th ACM Int'l Conf. Information and Knowledge Management (CIKM),* pp. 1681-1684, 2010.

[28] A. Silva, W. Meira, and M. Zaki, "Structural Correlation Pattern Mining for Large Graphs," *Proc. Eight Workshop Mining and Learning with Graphs,* 2010.

[29] A. Silva, W. Meira, and M.J. Zaki, "Mining Attribute-Structure Correlated Patterns in Large Attributed Graphs," *Proc. VLDB Endowment,* vol. 5, no. 5, pp. 466-477, 2012.

[30] D. Szklarczyk et al., "The String Database in 2011," *Nucleic Acids Research,* vol. 39, no. suppl 1, p. D561, 2011.

[31] T. Uno, "An Efficient Algorithm for Solving Pseudo Clique Enumeration Problem," *Algorithmica,* vol. 56, no. 1, pp. 3-16, 2010.

[32] M.J. Zaki, "Scalable Algorithms for Association Mining," *IEEE Trans. Knowledge Data Eng.,* vol. 12, no. 3, pp. 372-390, May/June 2000.

[33] Y. Zhou, H. Cheng, and J. Yu, "Graph Clustering based on Structural/Attribute Similarities," *Proc. VLDB Endowment,* vol. 2, no. 1, pp. 718-729, 2009.

**Adriana Prado** received the PhD degree in computer science from the University of Antwerp, Belgium, in 2009. In 2010, she worked as a postdoctoral researcher at Hubert-Curien Lab (Saint Etienne) in the Machine Learning Team. Since 2011, she is a postdoctoral researcher at INSA-Lyon, LIRIS lab. Her research interests include the fields of data mining, query languages, and databases in general.

**Marc Plantevit** received the PhD degree in computer science from the University of Montpellier 2, France, in 2008. He is currently an associate professor at the University Claude Bernard Lyon 1, France, and member the DM2L group in the LIRIS lab. His research is mainly concerned with pattern mining, especially in sequence and graph data. He is a member of the IEEE.

**Céline Robardet** is an associate professor at INSA-Lyon and a member of the LIRIS lab. Her main research interests include several aspects of data mining and combinatorial optimization. She is particularly interested in: clustering analysis, pattern extraction under constraints, and complex dynamic network analysis.

**Jean-François Boulicaut** is professor of computer science at INSA-Lyon. He is the leader of the Data Mining and Machine Learning research group in LIRIS lab. His own expertise concerns the design of unsupervized approaches within a constraint-based data mining setting. He is a member of the *Data Mining and Knowledge Discovery journal* editorial board and has served on the program committees of every major data mining conference.