Mining Graph Topological Patterns: Finding Covariations among Vertex Descriptors

— Source link ☑

A. Prado, Marc Plantevit, Céline Robardet, J-F Boulicaut

Institutions: Lyon College

Related papers:

- Mining attribute-structure correlated patterns in large attributed graphs

- Mining Cohesive Patterns from Graphs with Feature Vectors.

- Finding collections of k -clique percolated components in attributed graphs

- Uncovering the overlapping community structure of complex networks in nature and society

- FINAL: Fast Attributed Network Alignment

Mining Graph Topological Patterns: Finding Co-variations among Vertex Descriptors

Adriana Bechara Prado, Marc Plantevit, Céline Robardet, Jean-François Boulicaut

HAL Id: hal-01351727

https://hal.archives-ouvertes.fr/hal-01351727

Submitted on 7 Mar 2017

Mining graph topological patterns: finding co-variations among vertex descriptors

Adriana Prado , Marc Plantevit , Céline Robardet , Jean-François Boulicaut

Abstract—In this article, 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 correspond to the information conveyed by the vertices themselves and (2) some topological properties, used to describe the connectivity of each vertex in the graph. Such topological properties and attributes are mostly of numerical or ordinal types and their similarity can be captured by quantifying their co-variation, that is, if their largest or smallest values are supported mostly by the same set of vertices. A topological pattern is thus defined as a set of vertex attributes and topological properties that strongly co-vary over the vertices of the graph. Such pattern mining task 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. For instance, a topological pattern in a co-authorship graph, where vertices represent authors, edges encode co-authorship, and vertex attributes reveal the number of publications in several journals, could be *"the higher the number of publications in IEEE TKDE, the higher the closeness centrality of the vertex within the graph"*. Hence, such pattern discloses the fact that the number of times an author publishes at IEEE TKDE is positively correlated to the fact she has co-authored papers with other central authors, inducing a rather short distance to other graph vertices. We propose several interestingness measures of topological patterns that are different w.r.t. the pairs of vertices considered while evaluating up and down co-variations between properties and attributes: (1) considering all the pairs of vertices enables to find patterns that are true all over the graph; (2) taking into account only the vertex pairs that are in a specific order w.r.t. a selected attribute reveals the topological patterns that emerge with respect to 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. An efficient algorithm that combines searching and pruning strategies in the identification of 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 in a feasible time.

Index Terms—Attributed graph mining, topological pattern mining, co-variation.

✦

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], [32]), or compute frequent relationships between vertex attribute values (frequent subgraphs in a collection of graphs [16] or in a single graph [4]), while ignoring the topological status of the vertices within the whole graph, e.g. the vertex connectivity or centrality. The same limitation holds for methods [18], [24], [29], [30] that 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 extract meaningful patterns that integrate information about the connectivity of the vertices and their attribute values.

- A. Prado, C. Robardet, and J-F Boulicaut are with Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France. E-mail: {prenom.nom}@liris.cnrs.fr.
- M. Plantevit is with Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France. E-mail: marc.plantevit@liris.cnrs.fr.

The connectivity of each vertex is described by topological properties that quantify the topological status of the vertex in the graph. Some of these properties are based on the close neighborhood of the vertices, while others describe the connectivity of a vertex by considering its relationship with all other graph vertices. Combining such microscopic and macroscopic properties precisely characterizes the connectivity of the nodes and constitutes an information that may explain why some vertices have similar attribute values. For instance, as topological properties, one may consider the degree of each vertex, which describes the close neighborhood of the vertex, or a centrality measure of the vertices, which depicts the role of the vertex in the whole graph. Depending on the link between vertex attributes and the relationship encoded by the graph, one of these topological properties may co-vary with vertex attributes.

Such topological properties and vertex attributes are mostly of numerical or ordinal types and their similarity can be captured by quantifying their co-variation. Such co-variation indicates how a set of vertex descriptors tend to monotonically increase or decrease all together. Therefore, following the way paved by [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 co-vary over the vertices of the graph. We propose several interestingness measures of topological patterns that are different w.r.t. the pairs of vertices considered while evaluating up and down

co-variations between properties and attributes: (1) considering all the vertex pairs enables to find patterns that are true all over the graph; (2) taking into account only the vertex pairs that are in a specific order with respect to a selected numerical or ordinal attribute reveals the topological patterns that emerge with respect to 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 propose an operator that identifies the top k representative vertices of a topological pattern.

Let us illustrate our proposal on a co-authorship graph depicted in Figure 1, where vertices (from A to P) denote authors, edges encode co-authorship 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 [15]; i denotes the average number of hours per week spent by 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 to each vertex on Figure 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 $P = \{h^+, i^-, \text{BETW}^+\}$, 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 co-authors 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 3 representative vertices are E, I and D (shadowed on Figure 1) appearing respectively 15, 14 and 13 times in the right hand side of the supporting pairs, that is, these vertices have the highest values on h and BETW, and the lowest values on attribute i compare to other vertices.

We propose in this article an algorithm, called TopGraph-Miner, which discovers topological patterns. It has as input an attributed graph, such as the one in Figure 1. Given such graph, it first computes a set of topological properties for every of its vertices. TopGraphMiner then integrates searching and pruning strategies in the identification of the most relevant topological patterns. Finally, it gives to the user the ability to visualize every pattern on the input graph by identifying the top k representative vertices.

Our contribution is therefore threefold. First, we propose a new kind of graph analysis that exploits attributes and topological properties of vertices. Second, to produce such analysis, we provide new insights into co-variation pattern mining by considering up and down co-variations, defining new upper bounds on the support of such co-variations, proposing several interestingness measures of topological pattern, and giving to the user the ability to visualize the patterns
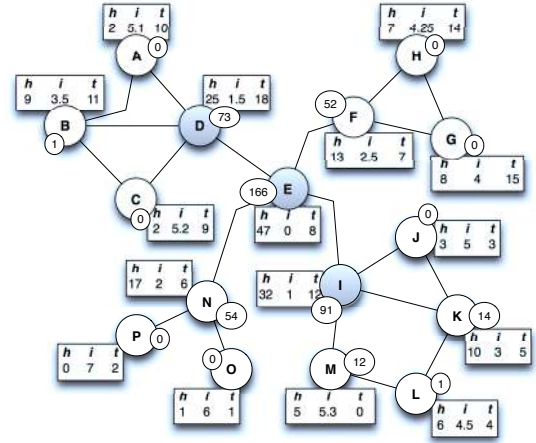

Figure 1. A co-authorship attributed graph toy example.

on the original graph thanks to the identification of the top k representative vertices. Third, to validate our approach, we conducted 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; (4) we provide results on the execution time with and without the pruning strategy; and (5) we study qualitatively some patterns extracted from four real-life networks: a co-authorship network, a movie co-actor network, a patent citation network, and gene interaction network.

This article 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 shown in Sections 6 and 7. Section 8 discusses the related work and Section 9 concludes the article.

2 TOPOLOGICAL VERTEX PROPERTIES

The input of our mining task is a non-directed attributed graph $G = (V, E, L)$, where V is a set of n vertices, E a set of m edges, and $L = \{l_1, \cdots, l_p\}$ a set of p attributes associated to 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 whole graph. Statistical distributions of these properties are generally used to characterize large graphs (see, e.g., [2], [17]). We propose here to use them as vertex descriptors.

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: $\text{DEGREE}(v) = \frac{deg(v)}{n-1}$.

- The clustering coefficient evaluates the connectivity of the neighbors of v (its local density):

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

- To better understand the structure of the neighborhood of v, we also consider the quasi-cliques [21] that involve v. v belongs to a γ-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. We consider two properties based on the quasi-cliques involving v: the size of the largest quasi-clique ($\text{SZQC}(v)$) and the number of quasi-cliques ($\text{NBQC}(v)$).

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, δ_{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).
As topological property, we consider the size of the community of v ($\text{SZCOM}(v)$).

- The relative importance of vertices in a graph can be obtained through centrality measures [11]. Closeness centrality $\text{CLOSE}(v)$ 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: $\text{CLOSE}(v) = \frac{n}{\sum_{u \in V} |shortest\_path(u,v)|}$.

- The betweenness centrality $\text{BETW}(v)$ 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: $\text{BETW}(v) = \sum_{u,w} \mathbb{1}_{shortest\_path(u,w)}(v)$.

- The eigenvector centrality measure (EGVECT) favours 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 to the largest eigenvalue λ of the adjacency matrix A of the graph.

- 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:

$$\text{PAGERANK}(v) = \alpha \sum_j \mathbb{1}_E(\{u,v\}) \frac{\text{PAGERANK}(u)}{deg(u)} + \frac{1-\alpha}{n}$$

where the parameter α is the probability that a random jump to vertex v occurs.

The 9 aforementioned 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 our following mining approach.

3 TOPOLOGICAL PATTERNS OVER NUMERICAL VERTEX DESCRIPTORS

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 co-variation over the vertices of the graph. Topological patterns are defined as $P = D_1{}^{s_1}, \cdots, D_\ell{}^{s_\ell}$, where D_j is a vertex descriptor from $\mathcal{D}$ and $s_j \in \{+,-\}$ is its co-variation sign. Following the example of Figure 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 w.r.t. the pairs of vertices considered while evaluating the support of such patterns.

3.1 Topological patterns over the whole graph

Several signed vertex descriptors co-vary all the more since 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 support of the pattern. This measure can be seen as a generalization of the Kendall's τ measure. When we consider all possible vertex pairs, this interestingness measure is defined as follows:

Definition 1 (Supp$_{all}$): The support of a topological pattern P over all possible pairs of vertices is:

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

where $\rhd_s$ denotes $<$ when s is equal to $+$, and $\rhd_s$ denotes $>$ when s 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 $-$. As mentioned in [5], $Supp_{all}$ is an anti-monotonic measure for positively signed descriptors. This is still true when considering negatively signed ones: adding D^- 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 $D(u) > D(v)$. 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 computation of duplicate topological patterns, we exploit Property 1. Equation (1) displays the number of possible patterns that can be constructed on $\mathcal{D}$ without the symmetrical patterns:

$$2^{|\mathcal{D}|} - 1 + \sum_{k=2}^{|\mathcal{D}|} \binom{|\mathcal{D}|}{k} \times (2^{k-1} - 1) \tag{1}$$

$2^{|\mathcal{D}|} - 1$ is the number of patterns that contain only descriptors with positive sign. The remainder represents those that combine positively and negatively signed descriptors. Considering the patterns of size k, with at least two descriptors, there are $\binom{|\mathcal{D}|}{k}$ such patterns and each gives rise to $(2^{k-1} - 1)$ patterns with at least one negatively signed descriptor. To discard symmetrical patterns, we force the first descriptor to be positively signed.

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 the minimum support threshold.

3.2 Other interestingness measures for topological patterns

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 w.r.t. 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. 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 Figure 1, h^+t^+ and h^+t^- are both frequent with minimum support of 20%. 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 w.r.t. E than h^+t^-, their growth rates being respectively equal to 1.185 and 0.631.

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

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

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:

$$Supp_{C^r}(P) = \frac{|\{(u, v) \in \mathcal{C}_{C^r} \mid \forall D^s \in P : D(u) \rhd_s D(v)\}|}{|\mathcal{C}_{C^r}|}$$

Analogously, the support of P over the pairs of vertices that do not belong to $\mathcal{C}_{C^r}$ is denoted $Supp_{C^{\overline{r}}}(P)$.

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 $\{\mathcal{C}_{C^r}, \mathcal{C}_{C^{\overline{r}}}\}$:

$$Gr(P, C^r) = \frac{Supp_{C^r}(P)}{Supp_{C^{\overline{r}}}(P)} \tag{2}$$

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 $\mathcal{C}_{C^{\overline{r}}}$. If $gr(P, C^r) \gg 1$, P is much more frequent in $\mathcal{C}_{C^r}$ than in $\mathcal{C}_{C^{\overline{r}}}$. 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:

$$Supp_E(P) = \frac{2|\{(u, v) \in \mathcal{C}_E \mid \forall D^s \in P : D(u) \rhd_s D(v)\}|}{|\mathcal{C}_E|}$$

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 $\forall D^s \in P, D(u) \rhd_s D(v)$ and $D(v) \rhd_s D(u)$ at the same time.

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 $\{\mathcal{C}_E, \mathcal{C}_{\overline{E}}\}$:

$$Gr(P, E) = \frac{Supp_E(P)}{Supp_{\overline{E}}(P)} \tag{3}$$

$Gr(P, E)$ 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.

4 TOP k REPESENTATIVE VERTICES OF A TOPOLOGICAL PATTERN

The user may be interested in identifying the vertices that are the most representative of a topological pattern, thus enabling the projection of the patterns back into the graph. For example, the representative vertices of the pattern {IEEE TKDE$^+$, 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:

$$S(P) = \{(u, v) \in V^2 \mid \forall D^s \in P : D(u) \rhd_s D(v)\}$$

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, $\forall D^s \in P : D(u) \rhd_s D(v)$ and $D(v) \rhd_s D(w)$. Thus, $D(u) \rhd_s D(w)$ and $(u, w) \in S(P)$. Therefore, G_P is transitive.

As $\rhd_s \in \{<, >\}$, 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. $\square$

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 largest 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 searching 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 none promising 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 w.r.t. D, such that $1 \le \rho(D(u)) \le |V|$, 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:

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

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 $(A(v_a) \unrhd_{s_a} A(u))$, 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 $(B(v_b) \rhd_{s_b} B(u))$, 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, $\text{Diff}_{A^{s_a} B^{s_b}}$ is the set of vertices that cannot form a supporting pair with u, such that:

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

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:

$$|I^+| = |\{v \in V | A(v) \le A(u) \ and \ v \ne u\}| = \overline{\rho}(A(u)) - 1$$
$$|J^+| = |\{v \in V | B(v) < B(u) \ and \ v \ne u\}| = \underline{\rho}(B(u)) - 1$$
$$|I^-| = |\{v \in V | A(v) \ge A(u) \ and \ v \ne u\}| = |V| - \underline{\rho}(A(u))$$
$$|J^-| = |\{v \in V | B(v) > B(u) \ and \ v \ne u\}| = |V| - \overline{\rho}(B(u))$$

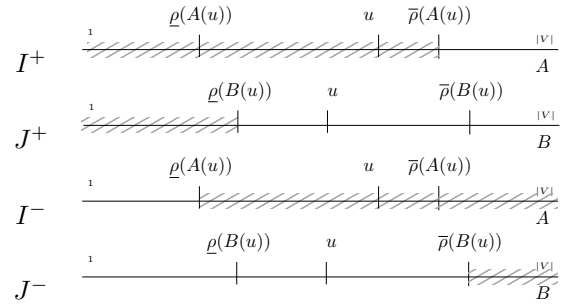Figure 2 illustrates these sets. In every case, the line represents

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

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 $\text{Diff}_{A^{s_a} B^{s_b}}$, 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 $|I^{s_a}| \ge |J^{s_b}|$, then the cardinality of $\text{Diff}_{A^{s_a} B^{s_b}}$ is minimal when $J^{s_b} \subseteq I^{s_a}$. Analogously, if $|I^{s_a}| < |J^{s_b}|$, then $\text{Diff}_{A^{s_a} B^{s_b}}$ can be empty, and thus its cardinality is 0. Thus,

$$|\text{Diff}_{A+B+}| \ge \max\{0, (\overline{\rho}(A(u)) - \underline{\rho}(B(u)))\}$$
$$|\text{Diff}_{A-B-}| \ge \max\{0, (\overline{\rho}(B(u)) - \underline{\rho}(A(u)))\}$$
$$|\text{Diff}_{A+B-}| \ge \max\{0, (\overline{\rho}(A(u)) - 1 - (|V| - \overline{\rho}(B(u))))\}$$
$$|\text{Diff}_{A-B+}| \ge \max\{0, (|V| - \underline{\rho}(A(u))) - (\underline{\rho}(B(u)) - 1))\}$$

To establish an upper bound on the support of a pattern P, we take, for each vertex u, the pair of signed descriptors $A^{s_a} B^{s_b}$ that maximizes $\text{Diff}_{A^{s_a} B^{s_b}}$: $\text{maxDiff}_P(u) = \max_{A^{s_a} B^{s_b} \in P^2} |\text{Diff}_{A^{s_a} B^{s_b}}|$. This leads to the following upper bound:

Theorem 1 (Upper bound on Supp): Let P be a topological pattern,

$$Supp_{all}(P) \leq 1 - \frac{\sum_{u \in V} \text{maxDiff}_P(u)}{n(n-1)} \qquad (4)$$

Proof: For each vertex u, let us consider two descriptors A^{s_a} and B^{s_b} from P such as $\text{maxDiff}_P(u) = |\text{Diff}_{A^{s_a} B^{s_b}}(u)|$. This is a lower bound on the number of vertices v such that $(A(v) \unrhd_{s_a} A(u))$ and $\neg(B(v) \rhd_{s_b} B(u))$. 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 $\binom{n}{2}$. $\square$

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 2 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 itemset 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 [33], [34]. 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 anti-monotonic 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). To avoid the unnecessary expensive computation of the support, we compute the upper bound on the support to prune non-promising topological patterns (function COMP_UB in line 8 of Algorithm 1). This function takes in parameters ρ and $\overline{\rho}$ that are computed in lines 5 to 7. 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 will be discussed in the following subsection.

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 cannot support another pattern Q that contains

exactly the same descriptors but with different signs. Thus, another upper bound on the support of P can be computed by summing the support of all such patterns Q already computed and taking the complement from 1. So, to be stringent, we bound the support by taking the minimum between this value and the upper bound defined in Theorem 4 (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).

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**

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 $\forall A^{s_a} \in$

P, $A(u) \rhd_{s_a} A(v)$. The computation of this measure 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 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 are 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 highest vertices with respect to one of these topological orderings.

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

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 can easily be exploited to retrieve the top k representative vertices of P: when we loop over the vertices of the graph and find in the range trees their incoming degree to compute the support of P, the set of k vertices having the largest incoming degree is maintain 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 the numerator value of Equation (2). When subtracting this value from the support of its direct ancestor, it provides the denominator value. 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

In this section, we report experimental results to illustrate the interest of our approach. We start by describing the 4 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 2.5GHz and 16GB of RAM under Linux operating systems. TopGraphMiner algorithm is implemented in C.

6.1 Real-World attributed graphs

We considered 4 real-world attributed graphs whose characteristics are given in Table 1:

1) DBLP: This co-authorship 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 co-authored at least one paper (no matter the conference or journal). The vertex properties are the number of publications in each of the 29 conferences or journals.

2) MOVIES: Each vertex of this graph represents a movie and an edge exists between two movies if 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 U.S. patents granted between January 1963 and December 1999[3]. We selected only patents of the subcategory "Computer Peripherals". There are 10 vertex attributes as, e.g., the grant year and the corresponding number of claims.

4) GENES: This graph contains gene-gene interactions [31], 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 [19].

The main characteristics of these graphs are reported in Table 1. Many of these properties have a standard-deviation greater than their average, suggesting that they follow power law distributions. 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.

---

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 Int. Sys., SIGKDD Exp., Comm. ACM, IDA J., KAIS, SADM, PVLDB, VLDB J., ACM TKDD.
2. http://www.imdb.com/
3. http://www.nber.org/patents/

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	2×10^{-4}				3.6×10^{-3}				1.7×10^{-4}				0.54×10^{-3}			
#Connected Components	577				33				67				11			
#Communities	1016				56				169				30			
Topological properties	Min	Max	Mean	Std. Dev.	Min	Max	Mean	Std. Dev.	Min	Max	Mean	Std. Dev.	Min	Max	Mean	Std. Dev.
Raw degree	0	304	9.73	14.22	0	118	21.16	19.13	0	313	8.26	9.67	0	68	2.28	6.66
DEGREE	0	7.3×10^{-3}	2.4×10^{-4}	3.4×10^{-4}	0	2.2×10^{-2}	4×10^{-3}	3.5×10^{-3}	0	2.5×10^{-2}	3.6×10^{-4}	5.6×10^{-4}	0	0.04	1.75×10^{-3}	4.5×10^{-3}
CLUST	0	1	0.31	0.29	0	1.57	0.34	0.26	–	–	–	–	0	1.69	0.06	0.18
NBQC	0	4.6×10^5	2.2×10^2	7.8×10^3	0	503	2.96	19.93	–	–	–	–	0	22	0.16	1.26
SZQC	0	35	2.75	4.83	0	52	13.87	11.35	–	–	–	–	0	46	0.84	4.96
SZCOM	0	9,342	40.67	5×10^2	0	1,563	11.5×10^2	5.6×10^2	0	8,178	50.9×10^2	25.9×10^2	0	394	48.73	93.9
CLOSE	0	1	0.024	0.137	0	1	0.010	0.099	0	1	0.005	0.067	0	1	4×10^{-3}	0.06
BETW	0	2.6×10^6	1.4×10^5	5.7×10^5	0	1.6×10^5	1.1×10^4	1.6×10^4	0	20.2×10^6	10.8×10^4	40.4×10^4	0	1.4×10^5	1.4×10^3	5.5×10^3
EGVECT	0	0.003	2.36×10^{-5}	9.91×10^{-5}	0	8.4×10^{-3}	1.6×10^{-4}	7.5×10^{-4}	0	11.6×10^{-3}	4.11×10^{-5}	2.8×10^{-4}	0	0.021	2.00×10^{-4}	2×10^{-3}
PAGERANK	0	21.53	0.98	0.98	0	0.59	0.88	0.59	0	35.98	0.93	0.91	0	7.69	0.31	0.62
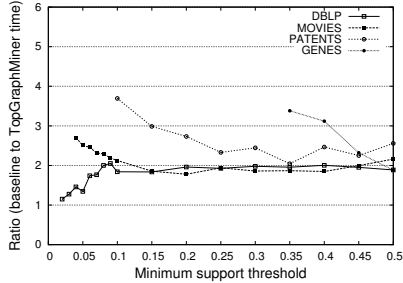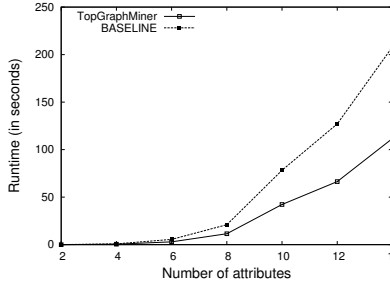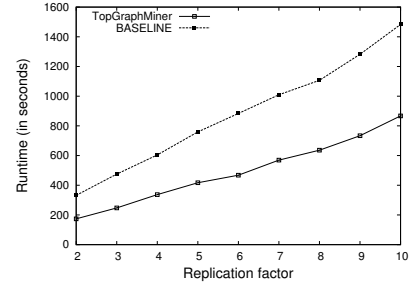
Figure 3. Comparison w.r.t. a baseline technique: execution time ratio (A), execution time w.r.t. number of descriptors (MOVIES, minsup=20%) (B), execution time w.r.t. a replication factor (MOVIES, minsup=20%) (C).

6.2 Performance Study

6.2.1 Comparison with a baseline approach

Since there is no other algorithm that simultaneously computes up and down co-variations using the same support measure as in our approach, we first study the performance of TopGraph-Miner by comparing it with a baseline approach. It consists in using the algorithm of [5], which only computes up co-variations, after having duplicate and reverse each descriptor. For instance, the vertex ranked first w.r.t. the descriptor D^+ is ranked last w.r.t. D^-. Notice that non-sensible patterns, such as $\{D^+, D^-\}$, will be discarded in linear time since their support is 0. Besides, it is necessary to post-process 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 post-processing step.

Figure 3(A) gives the ratio of the execution time of the baseline approach to the execution time of our approach on the 4 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 non-frequent topological patterns with negative signs that are early pruned by the baseline approach. Figure 3(B) shows the execution time spent by both algorithms w.r.t. different numbers of randomly chosen original descriptors from the MOVIES graph, with minimum support of 20%. We can observe that our approach outperforms the baseline one and the gain is more important when the number of descriptors increases. Figure 3(C) gives the execution time spent by both algorithms w.r.t. the number of vertices in the attributed graph MOVIES, with minimum support of 20% (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 Figure 3(A) are more influenced by the number of descriptors than that of vertices.

6.2.2 Empirical complexity of TopGraphMiner

Figures 4(A) and 4(B) 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 than 60%, 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.
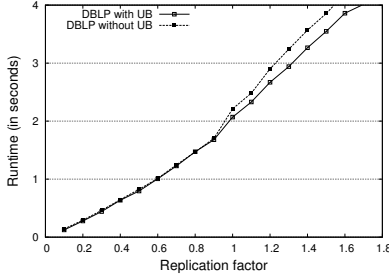

Figure 5. Impact of the upper bound on the execution time w.r.t. a replication factor (DBLP, minsup=40%).

Regarding the efficiency of our pruning technique (i.e., the upper bound on the support), Figure 4(C) gives, for every minimum support, the ratio of the number of pruned patterns, thanks to equation (4), to the number of patterns that, in the end, turned out to be indeed non-frequent. 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 attributed graphs DBLP, GENES, MOVIES, and PATENTS. In fact, when the minimum support is higher than 70%, almost all non-frequent patterns are pruned without computing their exact supports.

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 4 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 w.r.t. the minimum support threshold.

Figure 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 average execution time of 10 runs of TopGraphMiner on DBLP graph w.r.t. a replication factor. We can observe that the use the upper bound reduces the execution time and that this difference increases with the number of vertices. Note that, this difference is lighten by the use of range trees that smooth the impact of the computation of unpromising patterns.

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 Figure 6(A). The vertex attributes that have a correlation higher than 0.7 are VLDB, ICDE and SIGMOD. The more 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, Comm. of ACM, IEEE Int. Sys., CLOSE and CLUST. These correlation measures will help us in the interpretation of the following results.
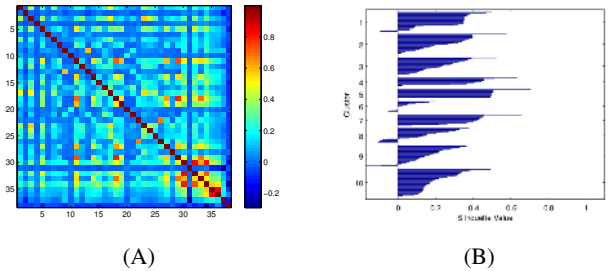

(A)   (B)

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

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% takes 68 seconds. The output contains 263 topological patterns, from which 58 (22%) 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 263×57 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 [27]. It suggests 10 clusters (see Figure 6(B)). The most frequent vertex attributes of each cluster are shown in Table 2. 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.
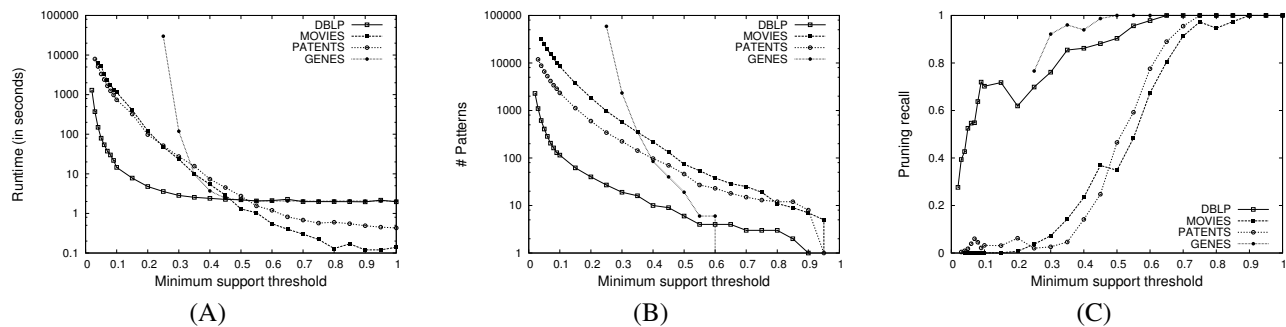
Figure 4. Execution time (A), number of patterns (B), pruning recall (C) w.r.t. minimum support threshold.

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$^-$	$Supp_{all} = 0.19$ $Gr(P, E) = 0.21$	#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$^+$	$Gr(P, \text{PAGERANK}^+) = 253,933$ $Gr(P, E) = 4.8$ $Supp_{all} = 0.12$	#1 H. Garcia-Molina, #2 M. Stone-braker, #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,

Interestingly, 20 of these patterns contain the attribute SAC^- together with positively signed attributes. Examples of such patterns are $\{SAC^-, KDD^+\}$, $\{SAC^-, ECML/PKDD^+\}$, $\{SAC^-, VLDB^+\}$, and $\{SAC^-, SIGMOD^+\}$. This type of pattern can be explained by the fact that SAC scope is larger than that of selected conferences, which are more focused either on Database or Data Mining topics. Since the topics covered by SAC are much 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 area.

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 ($P_{\text{PAGERANK}+}$) w.r.t. 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 Figure 7(A), where the top-10 representative vertices and their direct neighborhoods are displayed. These vertices have a low degree. As mentioned in Subsection 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-5 supporting vertices are mostly researchers from Software engineering and Network areas.

The computation of emerging patterns w.r.t. PAGERANK, with a support threshold of 1% 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 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 CLUST$^-$ suggests that the higher the PAGERANK of the authors (and consequently their DEGREE and BETW), the lower the connectivity of their co-authors. In other words, authors with high PAGERANK have many co-authors that do not publish together. This can be observed on Figure 7(B) where the connectivity between co-authors 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 PVLDB$^+$, DEGREE$^+$ and 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. Figure 7(C) 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 $P_{Topo_1} = \{\text{PAGERANK}^+, \text{DEGREE}^+\}$ and $P_{Topo_2} = \{\text{PAGERANK}^+, \text{BETW}^+\}$, two such emerging patterns whose respective growth-rates are $Gr(P_{Topo_1}, \text{PAGERANK}^+) = 124.69$ and $Gr(P_{Topo_2}, \text{PAGERANK}^+) = 584.46$. These emerging patterns reveal which conferences or journals are more related to the topological properties BETW$^+$ and DEGREE$^+$. To that end, for each publication venue C
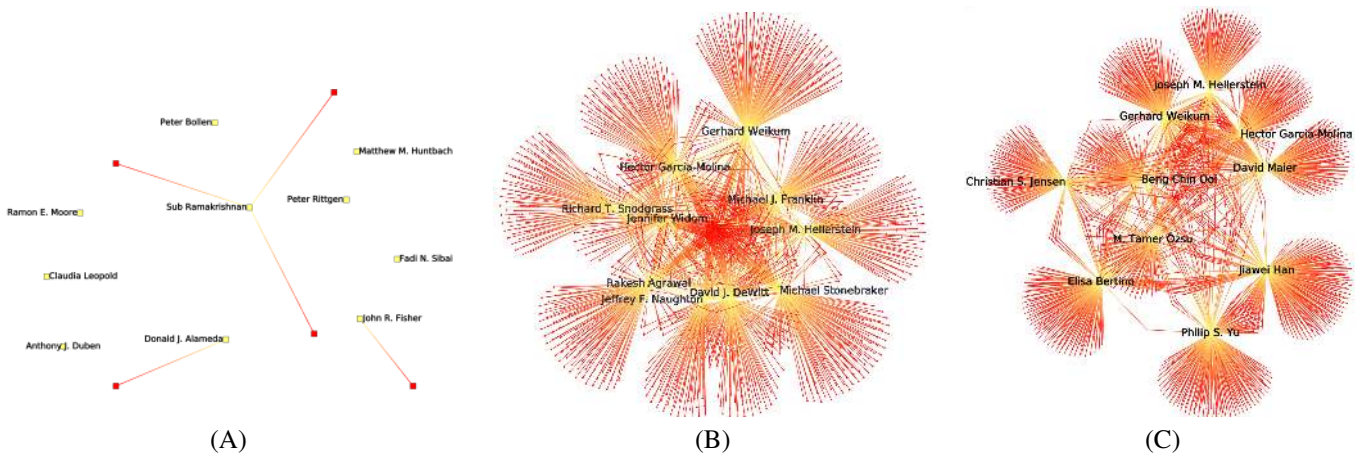
Figure 7. Top 10 vertices supporting P_{all} (A), P_{PAGERANK} (B) and P_E (C) and their connected vertices.

and both emerging patterns P_{Topo}, we compute the ratio $\frac{Gr(P_{Topo}C,\text{PAGERANK}^+)}{Gr(P_{Topo},\text{PAGERANK}^+)}$. Table 4(A) gives the top 5 publications w.r.t. to this ratio. Surprisingly, we observe that Data Mining conferences have a higher impact on the pattern {PAGERANK^+,DEGREE^+}, while Database conferences positively influence the growth-rate of the pattern {PAGERANK^+,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. 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 5 representative authors. They are shown in Table 4(B).

Table 4
Top-5 "impacting" publications in the emergence of DEGREE^+ and BETW^+ w.r.t. PAGERANK^+ (A) along with their top-5 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$^+$ ECML/PKDD$^+$	PAGERANK$^+$ BETW$^+$ 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

7.1.3 What about the IEEE TKDE authors?

We also look for the emerging patterns w.r.t. to the attribute IEEE TKDE, with support threshold of 1% and growth-rate threshold of 3 (their computation takes around 5 hours). We obtain 745 emerging patterns w.r.t the class IEEE TKDE$^+$. The most emerging pattern is $P_{\text{TKDE}} =$

$\text{ICDE}^+, \text{VLDB}^+, \text{BETW}^+, \text{PAGERANK}^+$, with $Gr(P_{\text{TKDE}} = ,\text{TKDE}^+) = 11.75$. This pattern indicates that authors publishing in IEEE TKDE journal tend also to publish papers in the conferences ICDE and VLDB. BETW^+ suggests that these authors are located in the center of the co-authorship graph, while PAGERANK^+ means that they co-authored papers with other researchers that also appear in the center of the graph. It is important to observe that this pattern is also highly structurally correlated ($Gr(P_{\text{TKDE}}, E) = 6.5758$). 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-5 representative authors.

7.2 What movies do we like watching?

Let us now consider the real-world attributed graph MOVIES. Table 5 shows the 4 most frequent topological patterns (with at least 2 descriptors) with their top-5 representative movies. Pattern P_1 suggests that Netflix users tend to rate movies

Table 5
Patterns found in MOVIES and their top-5 movies.

P	Descriptors	Measures	Top-5 movies
P_1	AVG_RATING$^+$ NB_RATINGS$^+$	$Supp_{all} = 0.7$ $Gr(P,E) = 1.05$	#1 The Green Mile, #2 Forrest Gump, #3 The Sixth Sense, #4 Indiana Jones and the Last Crusade, #5 Gladiator
P_2	NB_RATINGS$^+$ CLOSE$^+$	$Supp_{all} = 0.6$ $Gr(P,E) = 0.87$	#1 The Rock, #2 Fahrenheit 9/11, #3 The Godfather, #4 Enemy of the State, #5 Men of Honor
P_3	STD_RATING$^+$ PAGERANK$^-$	$Supp_{all} = 0.58$ $Gr(P,E) = 0.89$	#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$^+$ AVG_RATING$^-$	$Supp_{all} = 0.57$ $Gr(P,E) = 0.94$	#1 Day of the Dead 2: Contagium, #2 raging sharks, #3 My Big Phat Hip Hop Family, #4 The Fallen Ones, #5 Last Days

they like. Its top-10 representative movies are connected (see Figure 8(A)), which indicates they have at least one actor in common. The second pattern P_2 reveals that many users tend to rate movies located in 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 Figure 8(B)). 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 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 non-contemporary movies they like, 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_{PAGERANK} gathers descriptors NB_ACTORS$^+$

Table 6
Top topological patterns in MOVIES.

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

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$^+$ and all the 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 Figure 8(C).

7.3 How do patents cite each other?

We now present some topological patterns found in PATENTS. Table 7 shows the 4 most frequent patterns that involve vertex attributes and topological properties. The companies associated to the top-5 representative vertices of these patterns are also shown. For the pattern P_1, all 5 representatives belong to the same company Canon Kabushiki Kaisha. For the other patterns, at least 2 of the top-5 representative patents belong to the same company.

Table 7
Frequent patterns in PATENTS with the associated companies of their top-5 patents.

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

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 Figure 9(A) and (B)). 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.

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 #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,

The most emerging topological pattern w.r.t. 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 Figure 9(C)).

7.4 Are the known cancer-specific genes the most representative genes of the cancer related patterns?

To validate our approach on the attributed graph GENES, we consider 4 specific patterns made of two vertex attributes: one that corresponds to a healthy tissue (pancreas versus colon) and the other one to the same but cancerous tissue (adenocarcinoma versus carcinoma). The first attribute has a negative sign, whereas the second one has a positive one. Therefore, the most supporting genes of these patterns are those that are over-expressed in cancerous tissue while inhibited in normal one. Table 9 shows these patterns and their associated measures. To validate these patterns, we consider their supporting genes, and more precisely their ranks w.r.t. the patterns' topological ordering. We compute the normalized average ranks of two specific sets of genes known to be over-expressed in pancreas cancer (the genes HLA-DRB4, PPAPDC1B, and THBS1) [6] and colon cancer (the genes ANXA1, GJB2, PSMC5, RPS7) [26]. These values are given in the last two columns of Table 9.

Table 9
Four specific patterns in GENES.

P	Pattern	Measures	PANCREAS AVG RANK	COLON AVG RANK
P_1	PANCREAS NORMAL$^-$ PANCREAS ADENOCARCINOMA$^+$	$Supp_{all} = 0.0125$ $Supp_E = 0.0235$	0.378	0.308
P_2	PANCREAS NORMAL$^-$ PANCREAS CARCINOMA$^+$	$Supp_{all} = 0.0097$ $Supp_E = 0.0205$	0.510	0.183
P_3	COLON NORMAL$^-$ COLON ADENOCARCINOMA$^+$	$Supp_{all} = 0.0162$ $Supp_E = 0.0188$	0.821	0.230
P_4	COLON NORMAL$^-$ COLON CARCINOMA$^+$	$Supp_{all} = 0.0133$ $Supp_E = 0.0202$	0.806	0.306

As can be observed from Table 9, genes related to pancreas cancer highly support the two first patterns: they are in average
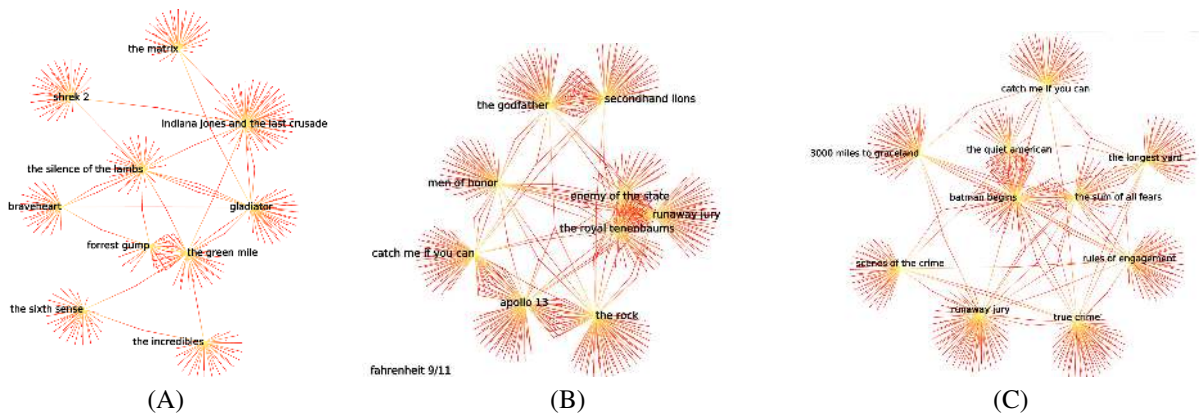
Figure 8. Top 10 vertices supporting P_1 (A), P_2 (B), and P_E (C) and their connected vertices in MOVIES.
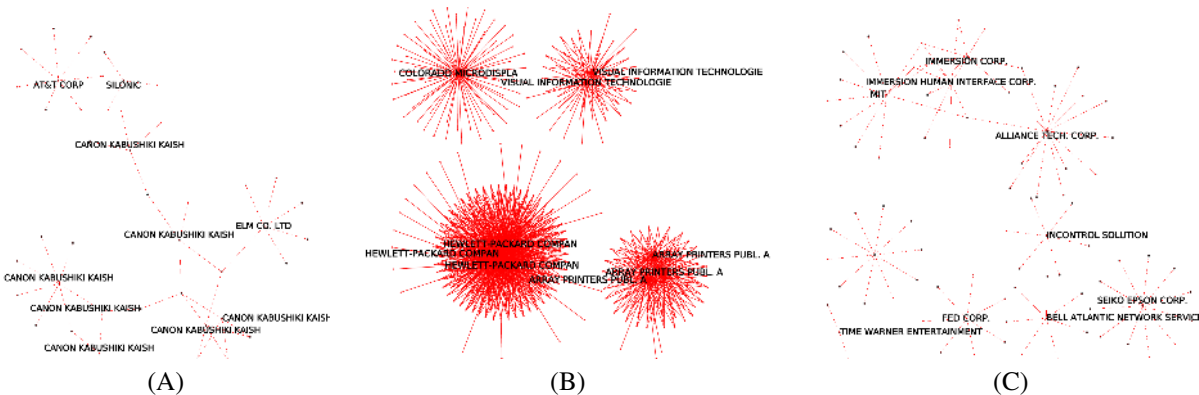

Figure 9. Top 10 vertices supporting P_1 (A), P_2 (B), and P_E (C) and their connected vertices of PATENTS.

in the first half of the ranks, having high values on Pancreas adenocarcinoma and carcinoma and low values on Pancreas normal cells. However, note that it does not happen in the patterns related to colon cancer (P_3 and P_4). On the other hand, the genes identified in [26] are not only involved in colon carcinoma and adenocarcinoma cells, but also in Pancreas cancer cells (see they have low average ranks in the 4 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"*.

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], [14], [20], [35])

whereas pattern mining techniques in such graphs have been less investigated. The pioneering work [23] propose a method to find dense homogeneous subgraphs (i.e., subgraphs whose vertices share a large set of attributes). Silva et al. [30] extract pairs of dense subgraphs and Boolean attribute sets such that the Boolean attributes are strongly associated with the dense subgraph. In [24], the authors 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 [18], where a larger neighborhood is considered. This pattern type relies on a relaxation of the accurate structure 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. [28] extract (not necessarily dense) subgraph with common itemsets.

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.

Co-variation patterns are also known as gradual patterns [9] or rank-correlated itemsets [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 τ statistical measure. However, their approach is not defined to simultaneously discover up and down co-variation patterns as does our approach. Another novelty of our work is the definition of other interestingness measures to capture emerging co-variations. Finally, this work is also the first attempt to use co-variation pattern mining in attributed graph.

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 co-variations 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 enabled 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 co-authorship graph built from the DBLP digital library, a graph derived from movies' characteristics, a citation graph of U.S. 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.

REFERENCES

[1] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. Parameter-free identification of cohesive subgroups in large graphs. In *SDM*, 2012.
[2] R. Albert and A.-L. Barabási. Topology of complex networks: Local events and universality. *Phys. Rev.*, 85:5234–5237, 2000.
[3] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Comp. net. and ISDN systems*, 30(1-7):107–117, 1998.
[4] B. Bringmann and S. Nijssen. What is frequent in a single graph? In *PAKDD*, pages 858–863, 2008.
[5] T. Calders, B. Goethals, and S. Jaroszewicz. Mining rank-correlated sets of numerical attributes. In *KDD*, pages 96–105, 2006.
[6] D. Campagna, L. Cope, et al. Gene expression profiles associated with advanced pancreatic cancer. *Int J Clin Exp Pathol*, 1(1):32–43, 2008.
[7] D. Chakrabarti, Y. Zhan, and C. Faloutsos. R-MAT: A recursive model for graph mining. In *SIAM SDM*, 2004.
[8] H. Cheng, Y. Zhou, and J. X. Yu. Clustering large attributed graphs. *TKDD*, 5(2):12, 2011.
[9] T. Do, A. Laurent, and A. Termier. Efficient parallel mining of closed frequent gradual itemsets. In *IEEE ICDM*, pages 138–147, 2010.
[10] G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *KDD*, pages 43–52, 1999.
[11] L. C. Freeman. A set of measures of centrality based on betweenness. *Sociometry*, 40(1):35–41, 1977.
[12] J. Fürnkranz and A. J. Knobbe. Guest editorial: Global modeling using local patterns. *DMKD*, pages 1–8, 2010.
[13] R. Ge, M. Ester, B. J. Gao, et al. Joint cluster analysis of attribute data and relationship data. *TKDD*, 2(2), 2008.
[14] S. Günnemann et al. A density-based approach for subspace clustering in graphs with feature vectors. In *PKDD*, pages 565–580, 2011.
[15] J. Hirsch. An index to quantify an individual's scientific research output. *PNAS*, 102(46):16569, 2005.
[16] D. Jiang and J. Pei. Mining frequent cross-graph quasi-cliques. *ACM TKDD*, 2(4):1–42, 2009.
[17] U. Kang, C. E. Tsourakakis, A. P. Appel, C. Faloutsos, and J. Leskovec. Hadi: Mining radii of large graphs. *ACM TKDD*, 5(2):8, 2011.
[18] A. Khan, X. Yan, and K.-L. Wu. Towards proximity pattern mining in large graphs. In *SIGMOD*, pages 867–878, 2010.
[19] J. Leyritz, S. Schicklin, et al. Squat: A web tool to mine human, murine and avian sage data. *BMC Bioinformatics*, 9, 2008.
[20] Z.-X. Liao and W.-C. Peng. Clustering spatial data with a geographic constraint. *Knowledge and Information Systems*, pages 1–18, 2012.
[21] G. Liu and L. Wong. Effective pruning techniques for mining quasi-cliques. In *ECML/PKDD*, pages 33–49, 2008.
[22] K. Makino and T. Uno. New algorithms for enumerating all maximal cliques. In *SWAT*, pages 260–272, 2004.
[23] F. Moser, R. Colak, A. Rafiey, and M. Ester. Mining cohesive patterns from graphs with feature vectors. In *SIAM SDM*, pages 593–604, 2009.
[24] P.-N. Mougel, C. Rigotti, and O. Gandrillon. Finding collections of k-clique percolated components in attributed graphs. In *PAKDD*, 2012.
[25] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004.
[26] V. Orian-Rousseau, S. Mink, et al. Genes upregulated in a metastasizing human colon carcinoma cell line. *Int J Cancer*, 5(113):699–705, 2005.
[27] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Comp. & Appl. Math.*, 20(1):53–65, 1987.
[28] J. Sese, M. Seki, and M. Fukuzaki. Mining networks with shared items. In *CIKM*, pages 1681–1684, 2010.
[29] A. Silva, W. Meira, and M. Zaki. Structural correlation pattern mining for large graphs. In *Wks on Mining and Learning with Graphs*, 2010.
[30] A. Silva, W. Meira, and M. J. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB*, 5(5):466–477, 2012.
[31] D. Szklarczyk et al. The string database in 2011. *Nucleic acids research*, 39(suppl 1):D561, 2011.
[32] T. Uno. An efficient algorithm for solving pseudo clique enumeration problem. *Algorithmica*, 56(1):3–16, 2010.
[33] M. J. Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.*, 12(3):372–390, 2000.
[34] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. New algorithms for fast discovery of association rules. In *KDD*, pages 283–286, 1997.
[35] Y. Zhou, H. Cheng, and J. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.