Supporting the Discovery of Relevant Topological Patterns in Attributed Graphs

Julien Salotti* , Marc Plantevit†, Céline Robardet* and Jean-François Boulicaut*
* *Université de Lyon, CNRS, INSA Lyon, LIRIS, UMR5205, F-69621, France*
†*Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France*

***Abstract*—We propose TopGraphVisualizer, a tool to support the discovery of relevant topological patterns in attributed graphs. It relies on a new pattern detection method that crucially needs for sophisticated postprocessing and visualization. A topological pattern is defined as a set of vertex attributes and topological properties (i.e., properties that characterize the role of a vertex within a graph) that strongly co-vary over the vertices of the graph. For instance, such a pattern in a co-authorship attributed graph where vertices represent authors, edges encode coauthorship, and vertex attributes reveal the number of publications in several journals, could be "the higher the number of publications in IEEE ICDM, the higher the closeness centrality of the vertex within the graph". Two different ways of navigation through the topological patterns and the related graph data are provided to the end-user. We exploit graph visualization and exploration techniques from the open platform Gephi. As an illustrative scenario, we consider a co-autorship attributed graph built from DBLP digital library and a video has been produced that describe the main possibilities of the TopGraphVisualizer software.**

***Keywords*-Topological patterns, attributed graphs, structural correlation.**

I. Introduction

Graphs are powerful models of real-world phenomena where vertices represent entities and edges represent their interactions. In real-life graphs, entities are often described by one or more attributes that constitute the attribute vectors associated with the vertices. 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 (e.g., cliques, quasi-cliques [1]), or by computing frequent relationships between vertex attribute values (frequent subgraphs in a collection of graphs or in a single graph [2]), while ignoring the topological status of the vertices within the whole graph, e.g., the vertex connectivity or centrality. The same limitation holds for the methods proposed in [3], [4] and [5], 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. We aim at discovering relevant patterns that integrate information about the connectivity of the vertices and their attribute values.

For us, 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 co-variation. Such co-variation indicates how a set of vertex descriptors tend to monotonically increase or decrease all together. We have recently proposed 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 [6]. We introduced several interestingness measures of topological patterns that differ by the pairs of vertices that are considered while evaluating up and down co-variations between descriptors: (1) Considering all the vertex pairs enables to find patterns that are true all over the graph; (2) 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 have also designed an operator that identifies the top k representative vertices of a topological pattern. As a result, we defined the TopGraphMiner algorithm to discover the topological patterns and their related top k representative vertices [6]. Considering only algorithmic issues, this appears as an extension of the method proposed in [7],

Like for most of the pattern mining techniques, the tedious interpretation phase by the end-user needs for many interactions with both the computed patterns (e.g., collections of topological patterns) and the data (e.g., large attributed graphs). Indeed, we should never forget that pattern detection is just one of the steps towards knowledge discovery: our goal is to disseminate the TopGraphMiner method among practitionners and it crucially needs for sophisticated post-processing techniques. In this ICDM 2012 demo session, we propose TopGraphVisualizer. It is a system that enables to navigate among the patterns computed by means of TopGraphMiner. Two ways of exploration are proposed. First, the end-user can navigate among the patterns. Some operators enable to rank the patterns according to different interestingness measures or enable select some of them given a specified property. A more original feature concerns the exploration based on the top k vertices that are ranked with

978-0-7695-4925-5/12 $26.00 © 2012 IEEE
DOI 10.1109/ICDMW.2012.38

respect to the number of computed patterns they are representative. For any selected vertex, an operator enables to directly retrieve all topological patterns in which the vertex is representative. Finally, selected patterns can be visualized thanks to Gephi which is an interactive visualization and exploration platform [8].

II. Problem Definition

A. 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 $L = \{l_1, \dots, l_p\}$ a set of p attributes associated with each vertex of V. We assume here that they are numerical or ordinal. Important properties of the vertices are also encoded by the edges of the graph that describe inter-relations between vertices. This relation enables to compute some topological properties that summarize the role played by each vertex within the graph. These properties range from a microscopic level (i.e., those that describe a vertex based on its direct neighborhood) to a macroscopic level (i.e., those that characterize a vertex by considering its relationship to all other vertices). Statistical distributions of these properties can be used as vertex descriptors:

Microscopic properties of a vertex v
Degree centrality (denoted DEGREE),
Clustering coefficient (CLUST),
Number of γ-quasi cliques involving v (NBQC),
Size of the largest γ-quasi-cliques involving v (SZQC).

Macroscopic properties of a vertex v
Size of the community involving v (SZCOM),
Closeness centrality (CLOSE),
Betweenness centrality (BETW),
Eigenvector centrality (EGVECT),
Pagerank index (PAGERANK).

Such properties characterize the graph relationship encoded by E. These properties, along with the set of vertex attributes L, constitute the set of vertex descriptors $\mathcal{D}$.

B. Topological patterns over the whole graph

Let us now consider a topological pattern as a set of vertex attributes and topological properties that behave similarly over a large part of the graph vertices. Since targeted topological properties and vertex attributes are of numerical or ordinal type, we propose to capture their similarity by quantifying their co-variation over the graph vertices. A topological pattern P is defined as $\{D_1{}^{s_1}, \cdots, D_\ell{}^{s_\ell}\}$ where D_j is a vertex descriptor from the set of all descriptors $\mathcal{D}$ and $s_j \in \{+, -\}$ is its co-variation sign. For instance, the trend "*the more papers in IEEE ICDM, the higher the Pagerank*" is represented by the topological pattern {ICDM$^+$, PAGERANK$^+$}.

Several signed vertex descriptors co-vary 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 τ measure.

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

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

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

This 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 $-$.

C. Emerging patterns w.r.t. the graph structure

Given a topological pattern P, $Supp_{all}(P)$ considers all possible of vertex pairs and thus does not take into account the graph structure. To measure if the graph structure plays an important role in the support of P, a similar support measure based on pairs that belongs to the set of edges E is $\mathcal{C}_E = \{(u, v) \in V^2 \mid \{u, v\} \in E\}$. The graph support of P can now be defined.

Definition 2 ($Supp_E$): The support of a topological pattern P over the pairs of vertices that are linked in G is:

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

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 $\forall D_j^{s_j} \in P$, $D_j(u) \rhd_{s_j} D_j(v)$ and $D_j(v) \rhd_{s_j} D_j(u)$ at the same time. The support of P over the pairs of vertices that do not belong to $\mathcal{C}_E$ is denoted $Supp_{\overline{E}}(P)$.

These measures allow to evaluate the impact of E on the support of P. We use a 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.

D. 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 {ICDM$^+$, BETW$^-$} would be researchers with a relatively large number of IEEE ICDM papers and a low betweenness centrality measure. Assume $S(P)$ denotes the set of vertex pairs (u, v) that constitutes the support of a topological pattern P:

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

which forms, with V, a directed graph $G_P = (V, S(P))$.

G_P is transitive and acyclic: it admits a topological ordering of its vertices, which is, in the general case, not unique. The top k representative vertices of a pattern P are the k last vertices with respect to this ordering of V. 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.

III. Description of the system

In [6], we have proposed the TopGraphMiner algorithm that enables to discover topological patterns considering and pushing several interestingness measures. We present here TopGraphVisualizer to support the discovery of relevant topological patterns in attributed graphs. TopGraphVisualizer, which is implemented in Java (JSE7), aims at providing to the end-user a set of operators to supply the navigation through the TopGraphMiner's output.

We propose two ways of navigation for identifying the patterns of interest. The end-user can navigate through the topological patterns either by directly applying operators on the patterns or by considering the representative vertices.

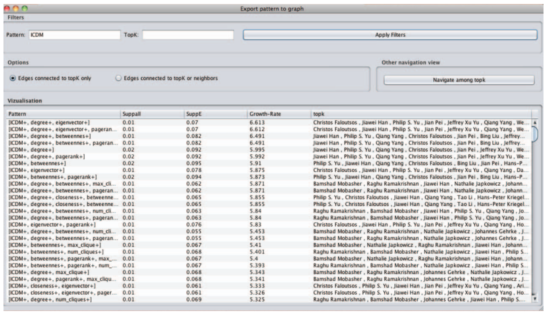
A. Pattern-based navigation

For the end-user, the most natural navigation through TopGraphMiner's output is to apply some operators over the topological patterns themselves. TopGraphVisualizer, illustrated in Figure 1, supports the following functionalities:

Pattern ranking: Topological patterns can be sorted with respect to any interestingness measure (e.g., $Supp_{all}(P), Supp_E(P)$ and $Gr(E, P)$) in ascending order or descending order.

Attribute selection: The end-user can select only patterns containing a set of signed or unsigned attributes.

Vertex selection: The end-user can make a restriction of the topological patterns to those containing a specified vertex among their top k representatives vertices (see Figure 2).



Export pattern to graph

Filters

Pattern: ICDM TopK:

Options

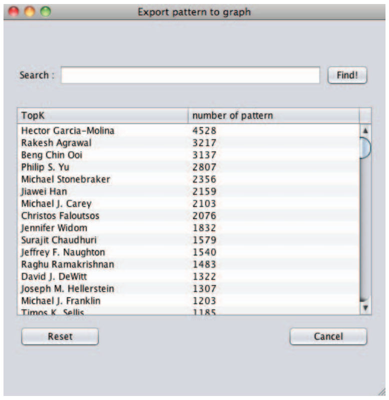
● Edges connected to topK only ○ Edges connected to topK or neighbors

Other navigation view

Vizualisation

Pattern	Suppall	SuppE	Growth-Rate	topk
[ICDM+, degree+, eigenvector+]	0.01	0.07	6.613	Christos Faloutsos , Jiawei Han , Philip S. Yu , Jian Pei , Jeffrey Xu Yu , Qiang Yang , We...
[ICDM+, degree+, eigenvector+, pageran...	0.01	0.07	6.612	Christos Faloutsos , Jiawei Han , Philip S. Yu , Jian Pei , Jeffrey Xu Yu , Qiang Yang , We...
[ICDM+, degree+, betweennes+]	0.01	0.082	6.491	Jiawei Han , Philip S. Yu , Qiang Yang , Christos Faloutsos , Jian Pei , Bing Liu , Jeffrey...
[ICDM+, degree+, betweennes+, pageran...	0.01	0.082	6.491	Jiawei Han , Philip S. Yu , Qiang Yang , Christos Faloutsos , Jian Pei , Bing Liu , Jeffrey...
[ICDM+, degree+]	0.02	0.092	5.995	Jiawei Han , Philip S. Yu , Qiang Yang , Christos Faloutsos , Jian Pei , Jeffrey Xu Yu , We...
[ICDM+, degree+, pagerank+]	0.02	0.092	5.992	Jiawei Han , Philip S. Yu , Qiang Yang , Christos Faloutsos , Jian Pei , Jeffrey Xu Yu , We...
[ICDM+, betweennes+]	0.02	0.095	5.91	Philip S. Yu , Jiawei Han , Qiang Yang , Christos Faloutsos , Bing Liu , Jian Pei , Hans-P...
[ICDM+, eigenvector+]	0.01	0.078	5.875	Christos Faloutsos , Jiawei Han , Philip S. Yu , Jian Pei , Jeffrey Xu Yu , Qiang Yang , Da...
[ICDM+, betweennes+, pagerank+]	0.02	0.094	5.873	Philip S. Yu , Jiawei Han , Qiang Yang , Christos Faloutsos , Jian Pei , Bing Liu , Hans-P...
[ICDM+, degree+, betweennes+, max_cli...	0.01	0.062	5.871	Bamshad Mobasher , Raghu Ramakrishnan , Jiawei Han , Nathalie Japkowicz , Johann...
[ICDM+, degree+, betweennes+, pageran...	0.01	0.062	5.871	Bamshad Mobasher , Raghu Ramakrishnan , Jiawei Han , Nathalie Japkowicz , Johann...
[ICDM+, degree+, closeness+, betweenne...	0.01	0.065	5.855	Philip S. Yu , Christos Faloutsos , Jiawei Han , Qiang Yang , Tao Li , Hans-Peter Kriegel...
[ICDM+, degree+, closeness+, betweenne...	0.01	0.065	5.855	Philip S. Yu , Christos Faloutsos , Jiawei Han , Qiang Yang , Tao Li , Hans-Peter Kriegel...
[ICDM+, degree+, betweennes+, num_cli...	0.01	0.063	5.84	Raghu Ramakrishnan , Bamshad Mobasher , Jiawei Han , Philip S. Yu , Qiang Yang , Jo...
[ICDM+, degree+, betweennes+, pageran...	0.01	0.063	5.84	Raghu Ramakrishnan , Bamshad Mobasher , Jiawei Han , Philip S. Yu , Qiang Yang , Jo...
[ICDM+, eigenvector+, pagerank+]	0.01	0.076	5.83	Christos Faloutsos , Jiawei Han , Philip S. Yu , Jian Pei , Jeffrey Xu Yu , Qiang Yang , Ho...
[ICDM+, degree+, betweennes+, num_cli...	0.01	0.055	5.453	Bamshad Mobasher , Raghu Ramakrishnan , Nathalie Japkowicz , Johannes Gehrke , J...
[ICDM+, degree+, betweennes+, pageran...	0.01	0.055	5.453	Bamshad Mobasher , Raghu Ramakrishnan , Nathalie Japkowicz , Johannes Gehrke , J...
[ICDM+, betweennes+, max_clique+]	0.01	0.067	5.41	Bamshad Mobasher , Nathalie Japkowicz , Raghu Ramakrishnan , Jiawei Han , Johann...
[ICDM+, betweennes+, num_cliques+]	0.01	0.068	5.401	Raghu Ramakrishnan , Bamshad Mobasher , Nathalie Japkowicz , Jiawei Han , Philip S...
[ICDM+, betweennes+, pagerank+, max_...	0.01	0.067	5.4	Bamshad Mobasher , Nathalie Japkowicz , Raghu Ramakrishnan , Jiawei Han , Johann...
[ICDM+, betweennes+, pagerank+, num_...	0.01	0.067	5.393	Raghu Ramakrishnan , Bamshad Mobasher , Nathalie Japkowicz , Jiawei Han , Philip S...
[ICDM+, degree+, max_clique+]	0.01	0.068	5.343	Bamshad Mobasher , Raghu Ramakrishnan , Johannes Gehrke , Nathalie Japkowicz , J...
[ICDM+, degree+, pagerank+, max_cliqu...	0.01	0.068	5.341	Bamshad Mobasher , Raghu Ramakrishnan , Johannes Gehrke , Nathalie Japkowicz , J...
[ICDM+, closeness+, eigenvector+]	0.01	0.061	5.333	Christos Faloutsos , Philip S. Yu , Jiawei Han , Jian Pei , Jeffrey Xu Yu , Qiang Yang , Ari...
[ICDM+, closeness+, eigenvector+, pager...	0.01	0.061	5.326	Christos Faloutsos , Philip S. Yu , Jiawei Han , Jian Pei , Jeffrey Xu Yu , Qiang Yang , Ho...
[ICDM+, degree+, num_cliques+]	0.01	0.069	5.325	Raghu Ramakrishnan , Bamshad Mobasher , Johannes Gehrke , Jiawei Han , Philip S....

Figure 1: Pattern-based navigation interface



Export pattern to graph

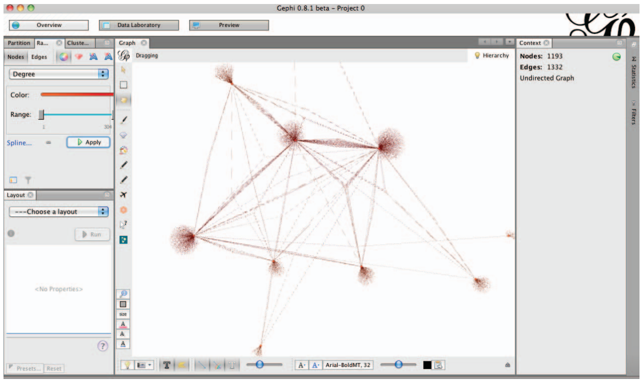
Search :

TopK	number of pattern
Hector Garcia-Molina	4528
Rakesh Agrawal	3217
Beng Chin Ooi	3137
Philip S. Yu	2807
Michael Stonebraker	2356
Jiawei Han	2159
Michael J. Carey	2103
Christos Faloutsos	2076
Jennifer Widom	1832
Surajit Chaudhuri	1579
Jeffrey F. Naughton	1540
Raghu Ramakrishnan	1483
David J. DeWitt	1322
Joseph M. Hellerstein	1307
Michael J. Franklin	1203
Timos K. Sellis	1185

Figure 2: Representative-based navigation interface

B. Representative-based navigation

The navigation through the TopGraphMiner's output can be based on the top k vertices that are ranked with respect to the number of patterns they are representative. Then, for any selected vertex v, an operator enables to the end-user to directly retrieve all topological patterns whose set of top k representative vertices contains v as illustrated in Figure 3.



Nodes: 1193
Edges: 1332
Undirected Graph

Figure 3: Pattern visualization with Gephi platform [8]

C. Pattern visualization

Finally, a selected topological pattern can be projected into the graph. To offer to the end-user a large set of features for interactive visualization and exploration of the topological pattern into the attributed graph, we made the choice to use Gephi an existing open visualization platform for all kinds of graphs [8]. Gephi supports graph visualization, structure, shape, and colors manipulations to reveal hidden properties.

We thus export the projected attributed graph to Gephi format. Two kinds of export are proposed. The first one only consider representative vertices and their neighbours and only edges involving a representative vertex. The second kind of export returns the attributed graph induced by the set of representative vertices and their neighbours. We also apply some state-of-the-art layout algorithms enhancing the visualization of the graph. The end-user can take benefit from the other functionalities of Gephi to analyse the selected topological pattern.

IV. Illustrative Scenario

We give a demonstration of the capabilities of our tool using a co-authorship graph 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 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 selected conferences or journals given in Table I. We also consider 9 topological properties (see Table I). The main characteristics of the attributed graph are given in Table II.

Table I: Vertex descriptors in the DBLP attributed graph

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.
Topological properties:
Degree, Close, Betw, EgVect, PageRank, Clust, SzQC, NbQC, SzCom.

Table II: Main characteristics of the attributed graph used as illustrative scenario.

Attributed graph	DBLP		
#Vertices	$42,252$		
#Edges	$210,320$		
#Vertex attributes	29		
Density	2×10^{-4}		
#Connected Comp.	577		
#Communities	1016		
Topo. prop.	Max	Mean	Std. Dev.
Raw degree	304	9.73	14.22
Degree	7.3×10^{-3}	2.4×10^{-4}	3.4×10^{-4}
Clust	1	0.31	0.29
NbQC	4.6×10^{5}	2.2×10^{2}	7.8×10^{3}
SzQC	35	2.75	4.83
SzCom	$9,342$	40.67	5×10^{2}
Close	1	0.024	0.137
Betw	2.6×10^{6}	1.4×10^{5}	5.7×10^{5}
EgVect	0.003	2.36×10^{-5}	9.91×10^{-5}
PageRank	21.53	0.98	0.98

TopGraphVisualizer supports the discovery of relevant topological patterns in this attributed graph. It enables to find out the patterns among the publications target and some topological properties that are the most correlated to the graph structure. TopGraphVisualizer also enables to navigate among the authors that are representative and visualize the patterns they are involved in and their projection into the graph.

V. Conclusion

We propose TopGraphVisualizer, a tool to support the discovery of relevant topological patterns in attributed graphs. Two different ways of navigation through the topological patterns are provided to the end-user. We take benefit from the graph visualization and exploration techniques available in the open platform Gephi. As illustrative scenario, we consider a co-authorship attributed graph built from DBLP digital library.

VI. Acknowledgements

This work is partly supported by the ANR (French Research National Agency) funded project Foster ANR-2010-COSI-012-02.

References

[1] G. Liu and L. Wong, “Effective pruning techniques for mining quasi-cliques,” in *ECML/PKDD*, 2008, p. 33–49.

[2] B. Bringmann and S. Nijssen, “What is frequent in a single graph?” in *PAKDD*, 2008, p. 858–863.

[3] A. Khan, X. Yan, and K.-L. Wu, “Towards proximity pattern mining in large graphs,” in *SIGMOD*, 2010, p. 867–878.

[4] P.-N. Mougel, C. Rigotti, and O. Gandrillon, “Finding collections of k-clique percolated components in attributed graphs,” in *PAKDD*, 2012.

[5] A. Silva, W. Meira, and M. J. Zaki, “Mining attribute-structure correlated patterns in large attributed graphs,” *PVLDB*, vol. 5, no. 5, p. 466–477, 2012.

[6] A. Prado, M. Plantevit, C. Robardet, and J.-F. Boulicaut, “Mining graph topological patterns: Finding co-variations among vertex descriptors,” *IEEE Trans. Knowl. Data Eng.*, p. 1–14, 2012 (In press).

[7] T. Calders, B. Goethals, and S. Jaroszewicz, “Mining rank-correlated sets of numerical attributes,” in *KDD*, 2006, p. 96–105.

[8] M. Bastian, S. Heymann, and M. Jacomy, “Gephi: An open source software for exploring and manipulating networks,” in *AAAI ICWSM*, 2009.