Toplogical Structure Pattern on Graph Mining by using the Top Graphminer

Kusuma Swamy
M.Tech Student, CVSR College of Engineering, Department of Computer Science, A.P. India

S.Kalyani
Associate Professor of Computer Science and Engineering, Anurag group of Institutions, A.P. India

Abstract—In this article, we propose to mine the graph topology of a large attributed graph byfinding 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 tofind patterns that are true all over the graph; (2) taking into account only the vertex pairs that are in a specific neighborhood 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.

I. INTRODUCTION

Real-world phenomena are often depicted by graphs where vertices represent entities and edges represent their relation- ships 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 sup- port 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, or compute frequent relationships between vertex attribute values, 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 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.

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.

II. PROPOSED 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 M, the anti-monotonic frequency constraint is fully-checked on thefly (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 ρ 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.

Algorithm 1 TopGraphMiner
Require: G = (V, E, L), minsup, k
Ensure: 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: D ← T ∪ L
3: M ← ∅
4: for all D ∈ D, in descending order do
5:for all v ∈ V do
6:Compute ρ(D(v)) and ρ(D(v)).
7:end for
8:U B ← COMP UB({D+ }, ρ, ρ)
9:if (U B ≥ minsup) then
10:(supp, topk) ← COMP S UPP({D+ }, k)
11:if (supp ≥ minsup) then
12:M ← M ∪ ({D+ }, topk)
13:E XTEND PATTERN ({D+ })
14:end if
15:end if
16: end for

Algorithm 2 Extend Pattern
Require: P a topological pattern, minsup, k, ρ, ρ
Ensure: Compute all frequent extensions of P and
add them to the global variable M with their top
k representative vertices
1: for all B ∈ D, B greater than the last descriptor
in P do
2:for all s ∈ {+, −} do
3:Q ← P ∪ {B s }
4:if (∀R ⊂ Q, R ∈ M) then
5:UB←min{COMP UB(Q, ρ, ρ),
COMP D EDUC (Q, M)}
6:if (U B ≥ minsup) then
7:(supp, topk) ← COMP S UPP(Q, k)

8:if (supp ≥ minsup) then
9:M ← M ∪ (Q, topk)
10:E XTEND PATTERN (Q)
11:end if
12:end if
13:end if
14:end for
15: end for

Discussion and Optimizations

We discuss other optimizations used in TopGraphMiner algo- rithm and how emerging topological patterns are computed.

Computation of Suppall

The support of P is evaluated by function COMP SUPP that counts the number of pairs of vertices (u,v) such that ∀Asa ∈P, A(u)Bsa A(v). The computation of this measure requires to perform a quadratic operation on the number of vertices. However, 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.

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 GP = (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 $D^-(v)$ be the incoming degree of the vertex v ∈ V ($deg^-(v) = |\{\forall u \in V$ such that $(u,v) \in A\}|$). For any arc (u,v) ∈ A, $deg^-(u) \leq deg^-(v) + 1$.

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

As a result, ordering V with respect to deg^- constitutes a topological sorting of GP . 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(logk).

Computation of SuppCr, SuppE and Gr

Emerging topological patterns can easily be computed by adapting Algorithm 1: the selected descriptor Cr is the last one in the pattern being enumerated (in the ECLAT enumer- ation 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 SuppE(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 SuppE(P) and Suppall(P).

III. Experiment and Result

Real-World attributed graphs

We considered 4 real-world attributed graphs whose charac- teristics 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 con- ferences and journals of the Data Mining and Database communities1 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 common2. 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 19993. 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], that is, each vertex stands for a gene and an edge links two vertices if they are known to interact during the biological transcription process. The vertex attributes associated with each gene are its expression values in each of 348 biological situations. Those situations are as many human tissues from several organs that are healthy or cancerous.

The main characteristics of these graphs are reported in Table 1. 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.

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^{-3}	9.91×10^{-3}	0	8.4×10^{-3}	1.6×10^{-3}	7.5×10^{-4}	0	11.6×10^{-3}	4.11×10^{-3}	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

IV.CONCLUSION

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. Barab´ asi. 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. F¨ urnkranz 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.