2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining

Attributed Graph Clustering: an Attribute-aware Graph Embedding Approach

Esra Akbas Peixiang Zhao
Department of Computer Science
Florida State University, Tallahassee, Florida 32306-4530
akbas@cs.fsu.edu zhao@cs.fsu.edu

***Abstract*—Graph clustering is a fundamental problem in social network analysis, the goal of which is to group vertices of a graph into a series of densely knitted clusters with each cluster well separated from all the others. Classical graph clustering methods take advantage of the graph topology to model and quantify vertex proximity. With the proliferation of rich graph contents, such as user profiles in social networks, and gene annotations in protein interaction networks, it is essential to consider both the structure and content information of graphs for high-quality graph clustering. In this paper, we propose a *graph embedding* approach to clustering content-enriched graphs. The key idea is to embed each vertex of a graph into a continuous vector space where the localized structural and attributive information of vertices can be encoded in a unified, latent representation. Specifically, we quantify vertex-wise attribute proximity into edge weights, and employ truncated, *attribute-aware* random walks to learn the latent representations for vertices. We evaluate our attribute-aware graph embedding method in real-world attributed graphs, and the results demonstrate its effectiveness in comparison with state-of-the-art algorithms.**

I. Introduction

Graph clustering is a fundamental problem for networked data. the objective of which is to partition a graph into a series of densely connected subgraphs. In real-world applications, besides interlinked graph topologies, a proliferation of graph contents have been witnessed that exhibit crucial attributes and graph properties. Consequently, the classical graph clustering methods need to be revamped to support the so-called *attributed graph clustering* [1], [8], which has found numerous applications, and has the potential to yield more informative and better-quality graph clusters.

However, attributed graph clustering is challenging because topological structures and attributive graph contents are two completely different types of information pertaining to graphs. Clustering solely based on either type will lead to inaccurate, or even contradicting, clusters [8]. In this paper, we introduce a novel approach to incorporate both graph structure cohesiveness and attribute homogeneity for attributed graph clustering. The main idea is to design a unified, latent representation for each vertex u of a graph G such that both graph connectivity and vertex attribute proximity in the localized region of u can be jointly embedded into a continuous vector space. In particular, pairwise vertex-attribute similarity between u and its incident vertices is first quantified and embedded as edge weights of G. A series of truncated, weight-biased random walks originating from u are further generated to capture localized, attribute-aware structure information surrounding u. Inspired by the recent work of graph embedding [2], [5], [4], these random walks are employed to learn a latent representation, $r(u) \in \mathbb{R}^d$, of u, which lies in a continuous, d-dimensional vector space. As a consequence, attributed graph clustering is cast to the traditional, d-dimensional data clustering problem, which can be addressed by numerous algorithms, *e.g.*, k-Medoids. The main contributions of our work is summarized as follows,

1) We propose a novel, attribute-aware graph embedding framework for attributed graph clustering. It provides a natural and principled approach to encoding the localized structure and attribute information of vertices to a unified, latent representation in a low-dimensional space, within which graph structure cohesiveness and vertex attribute homogeneity are well preserved (Section IV);
2) We design an efficient and cost-effective graph embedding algorithm that transforms an attributed graph into its vertex-based, latent representation, which can be fed as input to any clustering method toward solving the attributed graph clustering problem (Section V);
3) We perform experimental studies in a series of real-world graphs in comparison with state-of-the-art attributed graph clustering techniques. The results demonstrate the effectiveness of our method in terms of both structure cohesiveness and attribute homogeneity in the resultant clusters (Section VI).

II. Related Work

Attributed graph clustering. There has been a rich literature for attributed graph clustering [1]. The straightforward idea is to define some vertex-wise distance/similarity metric that takes into account both structure and attribute information of vertices in a graph. SA-Cluster [8] transforms a graph into another augmented graph with new, artificial *attribute vertices* representing distinct vertex attribute values. A new

ASONAM'17, July 31 - August 03, 2017, Sydney, Australia

ACM ISBN 978-1-4503-4993-2/17/07?/$15.00
http://dx.doi.org/10.1145/3110025.3110092

attribute edge (u, u') is further created to bridge an original vertex u with an attribute vertex u', if one of u's attributes takes on the value of u'. This way, vertices sharing the same attribute values are connected via common attribute vertices. A random walk-based distance measure is defined upon the augmented graph to estimate vertex closeness in terms of both structure cohesiveness and attribute proximity. SA-Cluster is computationally expensive as the augmented graph can be excessively large, and it involves costly matrix multiplication in random-walk based computation and weight tuning.

Another main stream of related work build upon generative probabilistic models, where graph structures and attributes hinge on hidden variables for cluster membership. GBAGC [6] is a Bayesian probabilistic model in which cluster labels of vertices are explicitly represented as a hidden variable. A joint probability distribution is further estimated over the space of all possible clusterings, and a variational inference algorithm is developed to find the posterior distribution with highest probability. CESNA [7] assumes clusters will generate both graph structures and vertex attributes based on an affiliation network model and a separate logistic model. These probabilistic clustering methods, however, have to undertake a time-demanding optimization process for parameter estimation. In addition, choosing appropriate a priori distributions turns out to be non-trivial.

Graph Embedding. There have been numerous approaches to learning low-dimensional representations for graphs. DeepWalk [4] takes advantage of local structure information of vertices based on the Skip-Gram model and treats random walks as the equivalent of sentences. When applied in multi-label graph classification, DeepWalk can successfully encode the global graph structure even in the presence of missing information in graphs. Line [5] is a scalable graph embedding method that uses edge sampling for model inference. It naturally breaks the limitation of the classical stochastic gradient decent method adopted for graph embedding without compromising embedding efficiency. GraRep [2] refines DeepWalk by introducing an explicit loss function of the Skip-Gram model defined on the graph, and extends Line by capturing k-step ($k > 2$) high-order information for learning latent representations.

Our work differs from graph embedding solutions in that (1) graph embedding is typically optimized for graph classification, while our work target primarily on attributed graph clustering; (2) graph embedding encodes the mere graph structure into low-dimensional spaces, while our work takes account of graph attribute information and extends the existing frameworks toward *attribute-aware* graph embedding.

III. Problem Formulation

We consider clustering graphs where vertices are affiliated with multidimensional attributes. We refer to these complex graphs as *attributed graphs*, formally defined as follows,

Definition *1 (**Attributed Graph**):* An attributed graph is a three tuple $G = (V, E, \mathcal{A})$, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, and $\mathcal{A} = \{A_1, \ldots, A_n\}$ is a set of n attributes associated with vertices of V. That is, for each $u \in V$, there is an attribute vector $\mathcal{A}(u) = (A_1(u), \ldots, A_n(u))$ associated with u, where $A_l(u)$ is the attribute value of u on the lth attribute A_l ($1 \leq l \leq n$).

In this paper, we consider attributed graphs as undirected, connected, simple graphs, and all the vertex attributes conform to a unique multidimensional schema, $\mathcal{A}$. We further assume that each vertex attribute A_i has a finite set of discrete values and the number of possible values (or cardinality) of A_i is $|A_i|$. For attributes with continuous or infinitely countable values, we can transform them into discrete values by binning or histogram techniques. Given a vertex u in an attributed graph G, we denote all its neighboring vertices as $N_1(u) = \{v | v \in V, (u, v) \in E\}$. Analogously, we denote all the vertices that are l ($l \geq 1$) hops away from u as $N_l(u) = \{v | v \in V, d(u, v) = l\}$, where $d(\cdot)$ is the shortest unit distance function defined upon G. If l is small, $N_l(u)$ consists of all vertices that are in the local vicinity of u. In principle, if vertices in $N_l(u)$ are densely connected and share similar vertex attributes with u, they are likely to be in the same cluster as u belongs to.

Definition *2 (**Attributed Graph Clustering**):* Given an attributed graph G, we partition G into k mutually exclusive, collectively exhaustive subgraphs $G_i = (V_i, E_i, \mathcal{A})$ with an objective to obtain the following graph clustering properties:

1) **structure closeness:** Vertices within the same clusters are closely connected while vertices in different clusters are far apart;
2) **attribute homogeneity:** Vertices in the same clusters have similar attribute values, while vertices in different clusters differ significantly in attribute values.

We note that in the classical graph clustering problem, only the first objective is examined, while for attributed graph clustering, we consider both objectives for graph clustering.

IV. Attribute-aware Graph Embedding

In this section, we discuss our attribute-aware graph embedding framework for attributed graph clustering. The goal is to transform each vertex u into a latent, low-dimensional feature vector $f(u) \in \mathbb{R}^d$, where d is a small number for latent dimensions. This way, both vertex attributes and local graph structure information of u are encoded in $f(u)$ such that vertices of the same cluster will have similar feature vectors.

A. Vertex Attribute Embedding

Given an attributed graph G, our first step is to embed the information of vertex attribute similarity into a transformed, weighted graph $G' = (V, E; W)$, where $W : E \to \mathbb{R}_{\geq 0}$. Specifically, for each edge $e = (u, v) \in E$, we assign an edge weight $w(e)$ to quantify the vertex attribute similarity for u and v. As a result, the vertex attribute information of G is encoded into the weighted graph G' as edge weights.

The straightforward way to quantify the multidimensional attribute similarity of two adjacent vertices u and v is based on a dimension-wise evaluation of attribute values for u and

v, respectively. We define an indicator function $\mathbb{1}_{A_i}(u,v)$ for the attribute A_i of u and v as follows,

$$\mathbb{1}_{A_i}(u,v) = \begin{cases} 1 & \text{if } A_i(u) = A_i(v) \\ 0 & \text{otherwise} \end{cases}$$

Then the vertex attribute similarity, $s_0(u,v)$, of vertices u and v is computed as

$$s_0(u,v) = \frac{\sum_{i=1}^{n} \mathbb{1}_{A_i}(u,v)}{n} \tag{1}$$

In an attributed graph G, some adjacent vertices u and v within the same cluster may share few, or even no, identical vertex attribute values. In this case, their structure information plays a primary role leading them to the same cluster. However, if we solely rely on structure information, it is likely u and v will be assigned to different clusters by mistake. We thus extend the computation of vertex attribute similarity by taking into consideration neighboring vertices of u and v, respectively. That is, if u and v share few or no common vertex attribute values, the vertices in their vicinity may still hold identical or similar vertex attribute values, if u and v belong to the same cluster. Formally, we consider vertices in $N_l(u)$ which are l hops away from u. For each vertex attribute $A_i (1 \leq i \leq n)$, we maintain a histogram vector, $H_{A_i}(u)$, with a total number of $|A_i|$ entries, each of which corresponds to a feasible value $a_t \in \mathsf{Domain}(A_i)$, and maintains the value as

$$H_{A_i}(u)[t] = \frac{|\{v | v \in N_l(u), A_i(v) = a_t\}|}{|N_l(u)|}, 1 \leq t \leq |A_i| \tag{2}$$

That is, the tth element of the vector $H_{A_i}(u)$ maintains the percentage of vertices whose attribute value upon the attribute A_i is equal to a_t $(1 \leq t \leq |A_i|)$ *w.r.t.* all vertices that are l hops away from u. As a result, $H_{A_i}(u)$ maintains the *distribution* of vertex attribute values of A_i for the vertices close to u. So the vertex attribute similarity of u and v in terms of their l-hop neighbors can be formally defined as

$$s_l(u,v) = \frac{\sum_{i=1}^{n} \mathsf{sim}(H_{A_i}(u), H_{A_i}(v))}{n} \tag{3}$$

where $\mathsf{sim}(\cdot,\cdot)$ is a similarity function for t-dimension vectors. For adjacent vertices u and v, where $(u,v) \in E$, we synthesize the overall vertex attribute similarity, $s(u,v)$, by considering both the vertex attribute similarity of u and v per se (Equation 1), and those of the vertices within the localized vicinity up to L hops away from u and v, dampened by the neighborhood distance (Equation 3):

$$s(u,v) = \sum_{l=0}^{L} \frac{s_l(u,v)}{2^l}. \tag{4}$$

Finally, we assign $s(u,v)$ as the edge weight of the edge (u,v) in the transformed graph G', where the information of vertex attribute similarities is effectively embedded.

B. Structure Embedding

We consider a series of short-length random walks to capture structure closeness in the local vicinity of vertices. Specifically, for each vertex $u \in V$, we generate a group γ of truncated random walks rooted at u, denoted as $\mathcal{W}_l^t(u) = (u, v_1, \ldots, v_t)$, where $1 \leq l \leq \gamma$, and t is the length (*i.e.*, the number of edges) of random walks. Each random walk is generated as follows: we start from the vertex $v_0 = u$, and at each step i $(0 \leq i \leq t-1)$, we choose the next vertex $v_{i+1} \in N_1(v_i)$ with the probability:

$$\Pr(v_i, v_{i+1}) = \frac{s(v_i, v_{i+1})}{\sum_{v_j \in N_1(v_i)} s(v_i, v_j)}$$

where $s(v_i, v_{i+1})$ is the weight of the edge (v_i, v_{i+1}) in G', as defined in Equation 4. That is, the truncated random walks $\mathcal{W}_l^t(u)$ are generated in a *biased* fashion that edges with larger weights will be chosen with higher probabilities. Note that edge weights in G' indicate the vertex-wise attribute similarity, as discussed in Section IV-A. As a result, the truncated random walks rooted from u are *attribute-aware*, and encode both structure closeness and attribute homogeneity in the local vicinity of u, if t is set small.

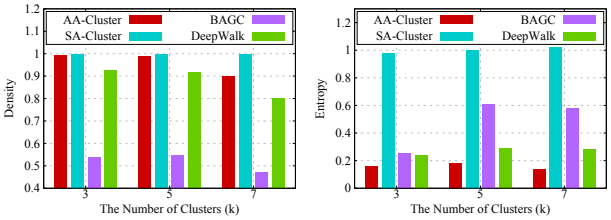
Inspired by recent advances in language modeling and deep learning [3], we treat each attribute-aware random walk as a short sentence or phrase, and each vertex of the graph as a word in a special language. Our goal is to learn a latent representation $\Phi : v \in V \rightarrow \mathbb{R}^d$ that maps each vertex into a low-dimensional vector, $\Phi(u)$. Following the intuition of DeepWalk [4], we relax the formulation of random walks as follows: (1) a random walk passing through a vertex $v_i \in V$ as the center of the walk is treated as a bi-directional random walk rooted at v_i; that is, we consider the transformed random walk originated from v_i and encompassing preceding and subsequent vertices in a window of size $2w$; (2) we ignore the ordering of vertices in random walks. Such relaxations are useful in particular for the latent representation learning as the order independence assumption captures a sense of "closeness" provided by random walks. Furthermore, they simplify the learning process and save the training time. To this end, deriving the latent representation of vertices is formulated as an optimization problem:

$$\min_{\Phi}(-\log \Pr(\{v_{i-w}, \ldots, v_{i-1}, v_i, v_{i+1}, \ldots, v_{i+w}\})) \tag{5}$$

To tackle this problem, we take advantage of SkipGram [3] to maximize the concurrence probability among words (vertices) within a window w in a sentence (a truncated random walk). we use Hierarchical Softmax and stochastic gradient descent (SGD) to optimize the approximation of probability distributions and parameter estimation.

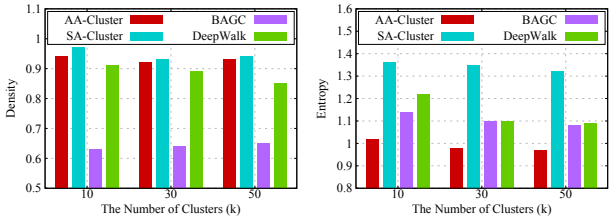
V. Attributed Graph Clustering Algorithm

Based on the attributed-aware graph embedding framework discussed in Section IV, it becomes straightforward to support clustering for attributed graphs. Given an attributed graph G, we first embed vertex attribute similarity information into a weighted graph G', where the parameter L regulates the scope of vertex neighborhood for the quantification of vertex attribute similarity. We then embed the structure information of G' by mapping vertices into d-dimensional latent representations, which encode both structure closeness and attribute homogeneity within the local vicinity of vertices, and thus are important indicators of the cluster membership of vertices. Once the



(a) Density (b) Entropy

Fig. 1: **Clustering Quality in Political Blog Datasets**



(a) Density (b) Entropy

Fig. 2: **Clustering Quality in DBLP Datasets**

original graph is transformed into its latent, d-dimensional representations, we can use any traditional data clustering method, such as kMedoids, to partition d-dimensional vectors into the final k graph clusters.

VI. Experiments

In this section, we present the experimental results for the proposed method, AA-Cluster (**a**ttribute-**a**ware graph **c**lustering). We compare AA-Cluster with three state-of-the art methods: (1) SA-Cluster [8] combines vertex attributes and graph structures in a unified distance measure for attributed graph clustering; (2)BAGC [6] is a Bayesian probabilistic approach for attributed graph clustering; (3) DeepWalk [4] learns the graph structure information as latent features in a low-dimensional space without consideration of vertex attributes. All the experiments were carried out in a Linux workstation running RedHat Enterprise Server 6.5 with 16 Intel Xeon 2.3GHz CPUs and 128GB of memory.

We consider two real-world attributed graphs in our experimental studies: **Political Blogs**[1] is a network of hyperlinks between web blogs on US politics recorded in 2005 with $1,490$ vertices and $19,090$ edges. Each blog has an attribute pertaining to its political leaning as either *liberal* or *conservative*. **DBLP**[2] is a co-authorship graph consisting of authors in different research areas with $27,199$ authors as vertices and $66,832$ collaborations as edges. To compare the effectiveness of different attributed graph cluttering methods, we adopt *clustering density* and *clustering entropy* as evaluation metrics [1],

We first apply attributed graph clustering methods in the Political Blog graph, and the clustering quality results are illustrated in Figure 1. By varying the number k of graph clusters, we recognize that the density of clusters generated by AA-Cluster is very close to that by SA-Cluster, both of which are consistently higher than the density results of BAGC and DeepWalk (Figure 1(a)). Meanwhile, graph clusters generated by AA-Cluster have significantly smaller entropy than those generated by the other methods, meaning that AA-Cluster leads to more homogeneous graph clusters *w.r.t.* vertex attributes (Figure 1(b)). Therefore, AA-Cluster results in both structurally dense, and attribute-wise homogeneous graph clusters.

We then perform experimental studied in the DBLP graph, and the clustering quality results are presented in Figure 2. By tuning the number k of resultant graph clusters, we notice that in terms of both density (Figure 2(a)) and entropy (Figure 2(b)), AA-Cluster outperforms SA-Cluster BAGC, and DeepWalk in generating high-quality graph clusters. In addition, the clustering quality of AA-Cluster is stable and insensitive to the number k of graph clusters generated.

[1] http://www-personal.umich.edu/~mejn/netdata

[2] http://dblp.uni-trier.de/xml

VII. Conclusion

Graph clustering has played a fundamental role in modeling, structuring, and understanding large-scale networks. In many real-world settings, we are concerned with not only interconnected graph structures, but rich graph contents characterized by vertex attributes during graph clustering. In this paper, we devised a new attributed graph clustering method that combines both vertex attributes and graph structure information within a general, unified attributed-aware graph embedding framework. We designed the graph embedding algorithm to encode an attributed graph into a low-dimensional latent representation. As a result, the attribute-aware cluster information is well preserved during graph embedding. We evaluated our method, AA-Cluster, in real-world graphs, and the results validated the effectiveness of AA-Cluster, compared with existing attributed graph clustering techniques.

Acknowledgment

This work was supported in part by the National Science Foundation under Grant No.1743142. Any opinions, findings, and conclusions in this paper are those of the author(s) and do not necessarily reflect the funding agencies.

References

[1] C. Bothorel, J. D. Cruz, M. Magnani, and B. Micenkova. Clustering attributed graphs: Models, measures and methods. *Network Science*, 3:408–444, 2015.

[2] S. Cao, W. Lu, and Q. Xu. GraRep: Learning graph representations with global structural information. In *Proceedings of CIKM'15*, pages 891–900, 2015.

[3] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13*, pages 3111–3119, 2013.

[4] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *Proceedings of KDD'14*, pages 701–710, 2014.

[5] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of WWW'15*, pages 1067–1077, 2015.

[6] Z. Xu, Y. Ke, Y. Wang, H. Cheng, and J. Cheng. GBAGC: A general bayesian framework for attributed graph clustering. *ACM Trans. Knowl. Discov. Data*, 9(1):5:1–5:43, 2014.

[7] J. Yang, J. J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *Proceedings of ICDM'13*, pages 1151–1156, 2013.

[8] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *Proc. VLDB Endow.*, 2(1):718–729, 2009.