Exploring Multiple Clustering in Attributed Graphs

Conference Paper · April 2015

DOI: 10.13140/RG.2.1.1813.8727

CITATION
1

READS
205

4 authors:

Gustavo Paiva Guedes
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
36 PUBLICATIONS **21** CITATIONS
SEE PROFILE

Eduardo Ogasawara
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
143 PUBLICATIONS **1,029** CITATIONS
SEE PROFILE

Eduardo Bezerra
Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET/RJ)
41 PUBLICATIONS **51** CITATIONS
SEE PROFILE

Geraldo Xexéo
Federal University of Rio de Janeiro
109 PUBLICATIONS **245** CITATIONS
SEE PROFILE

Exploring Multiple Clusterings in Attributed Graphs

Gustavo Paiva Guedes
COPPE/UFRJ - CEFET/RJ
gguedes@cefet-rj.br

Eduardo Bezerra
CEFET/RJ
ebezerra@cefet-rj.br

Eduardo Ogasawara
CEFET/RJ
eogasawara@cefet-rj.br

Geraldo Xexéo
IM/DCC & PESC/COPPE - UFRJ
xexeo@cos.ufrj.br

ABSTRACT

Many graph clustering algorithms aim at generating a single partitioning (clustering) of the data. However, there can be many ways a dataset can be clustered. From a exploratory analisys perspective, given a dataset, the availability of many different and non-redundant clusterings is important for data understanding. Each one of these clusterings could provide a different insight about the data. In this paper, we propose M-CRAG, a novel algorithm that generates multiple non-redundant clusterings from an attributed graph. We focus on attributed graphs, in which each vertex is associated to a n-tuple of attributes (e.g., in a social network, users have interests, gender, age, etc.). M-CRAG adds artificial edges between similar vertices of the attributed graph, which results in an augmented attributed graph. This new graph is then given as input to our clustering algorithm (CRAG). Experimental results indicate that M-CRAG is effective in providing multiple clusterings from an attributed graph.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: [Database applications - Data mining]

Keywords

Attributed graph clustering, multiple clustering, spectral clustering

1. INTRODUCTION

Graph clustering is a well-known problem in data mining. The goal of graph clustering is to cluster vertices of a graph, in such a way that vertices inside a cluster are densely connected, and vertices of different clusters are sparsely connected. Typical graph clustering applications include community detection and link prediction [11].

SAC'15 April 13-17, 2015, Salamanca, Spain.

http://dx.doi.org/10.1145/2695664.2696008

Many graph clustering algorithms only consider structural properties of graphs in the clustering process, hence ignoring vertices properties. In this work, we consider the existence of vertex attributes when clustering a graph. This type of graph is called attributed graphs [15]. A prototypical example of an attributed graph is a social network [10], in which the topological structure represents relationships between users, and the vertex properties describe characteristics and roles of each person [15].

Most clustering algorithms identify only one partition of the data [5]. However, it is possible that a single partition of the data does not provide sufficient insight. In this case, it would be interesting to explore other clustering solutions. In this work, we aim at discovering multiple clustering solutions, by combining information taken from the graph structure and from attributes associated to each vertex of this graph. Although there has been much work on multiple clusterings [8, 4, 12] and attributed graph clustering [15], to the best of our knowledge, we present a novel approach to generate multiple non-redundant clusterings from attributed graphs.

This paper is organized as follows. In Section 2 we present related work. M-CRAG, our multiple clustering algorithm, is described in Section 3. In Section 4 we present an experimental evaluation of M-CRAG. Section 5 concludes.

2. RELATED WORK

In the arena of attributed graph clustering, SA-cluster algorithm [15] generates clusters with a cohesive intra-cluster structure and homogeneous vertex properties, balancing the structural and attribute similarities. Before clustering, this algorithm adds a set of attribute vertices and attribute edges to the original graph. After this, a neighborhood random walk model is applied to unify the structural and attribute similarities, measuring the vertex closeness on the augmented graph. Finally, the K-Medoids framework is used to perform the clustering. The output of SA-cluster is an unique clustering solution. Similar, to the approach proposed by SA-Cluster, we aim at combining information on the graph structure and on vertices properties. However, our purpose is to generate *multiple* clustering solutions.

In the field of multiple clustering, Müller et al. [8] present an example of customer segmentation problem, in which each customer manifests multiple possible behaviors. Niu et al. [12] describe an approach that uses subspaces to generate multiple clustering solutions in each generated view and penalize redundancy between the views. In Bae et al.

[3], instance-level constraints are used together with agglomerative clustering to find an alternative clustering. In this approach, two clustering solutions are achieved. However, in general, there could be more than two alternative clusterings. Cui et al [4] developed an approach that can find more than two alternative clustering solutions, by exploring the subspaces ortoghonal to the clustering solutions found in previous iterations. Although all these approaches focuses on multiple clusterings, none of them are appropriate for multiple clustering in attributes graph.

3. PROBLEM STATEMENT

As described above, we aim at providing multiple non-redundant clustering solutions in a social network. We focus on undirected attributed graphs. Formally, an attributed graph G is defined as a 4-tuple $G = (V, E, \Lambda, f)$, where $V = \{v_1, v_2, ...v_n\}$ is a set of n vertices, E is a set of edges, which are unordered pairs of elements of V, $\Lambda = \{a_1, a_2, ..., a_m\}$ is a set of m attributes, and f is a function that maps edges to a pair of vertices. In an attributed graph G, each vertex v_i is associated with an attribute vector of length m.

It turns out the above definition provides two possible views for the underling data. The first one, the topological view, corresponds to the structure of the graph. The second view is the relational one, were each vertex $v_i \in V$ is represented by a m-tuple of attribute values.

Given an attributed graph $G = (V, E, \Lambda, f)$, we tackle the problem of generating a set of alternative non-redundant clusterings of vertices by combining both topological and relational information.

4. M-CRAG

In this Section, we present M-CRAG, an algorithm which aims at generating several non-redundant clusterings from an attributed graph G, by combining both topological and relational information. The main idea of M-CRAG is to add artificial edges between similar vertices in G, so that each generated augmented graph contains both original and artificial edges. After that, each augmented graph is processed to generate a clustering solution. This process is actually divided into two algorithms, M-CRAG (Algorithm 1) and CRAG (Algorithm 2).

Algorithm 1 M-CRAG(G, d, t, k)

Input:
- Attributed Graph $G = (V, E, \Lambda, f)$
- d = neighborhood distance
- t = Maximum threshold for NMI
- k = number of clusters

Output: $\mathcal{C}$, a set of non-redundant clustering solutions of vertices in G.

1: $\mathcal{C} \leftarrow \emptyset$
2: $\mathcal{F}$ = chooseAttributes($G.\Lambda$)
3: **for all** $attrSet \in \mathcal{F}$ **do**
4: $c \leftarrow$ CRAG($G, attrSet, d, k$)
5: **if** $\text{MAX}_{\text{NMI}}(\mathcal{C}, c) \leq t$ **then**
6: $\mathcal{C} \leftarrow \mathcal{C} \cup \{c\}$
7: **end if**
8: **end for**
9: **return** $\mathcal{C}$

M-CRAG initiates by receiving three parameters: an attributed graph G, a value d representing the neighborhood distance to search neighbors for a given vertex, and a threshold t, used to determine if two clusterings are sufficiently dissimilar. Following Strehl et al. [14], we adopt NMI to compare two clusterings, c and c', according to Eq. 1, where $H(c)$ is the entropy of clustering c, and $MI(c, c')$ is the mutual information of clusterings c and c'. $\text{NMI}(c, c')$ equals 1 if c and c' are identical, and 0 if c and c' are independent.

$$\text{NMI}(c, c') = \frac{MI(c, c')}{\sqrt{(H(c) \times H(c'))}} \quad (1)$$

In step 2, the algorithm invokes chooseAttributes, which selects a family $\mathcal{F}$ of attribute sets. Then, for each element in $\mathcal{F}$, M-CRAG invokes CRAG, which in turn produces an unique clustering c for each $attrSet$. Thus, the number of multiple clusterings generated by M-CRAG is at most $|\mathcal{F}|$. In the next step, function MAX_{NMI} is invoked to compare clustering c and all clusterings in $\mathcal{C}$. If all NMI values from comparisons between c and each clustering currently in $\mathcal{C}$ are bellow t, then c is added to $\mathcal{C}$, indicating that c is a sufficiently different clustering. The function chooseAttributes optimizes the number of attributes to be explored and its description is outside of the scope of this paper.

CRAG is an auxiliar algorithm which receives G, a set of attributes $attrSet$, and a positive integer d. In step 2, function getSimilarityThreshold() computes the similarity between all pair of vertices at distance δ, $1 < \delta \leq d$. The function analyses the distribution of these similarities and returns a value that corresponds to the threshold for the 20% most similar vertices at a distance δ using the Pareto principle [6], also known as "the 80-20 rule".

In step 4, for each v_i, function getNeighborhood() returns $\mathcal{N}_{v_i}$ the set of all vertices at distance δ from v_i. The similarity between v_i and each $v_j \in \mathcal{N}_{v_i}$ is computed and, if it is greater than s, an artificial edge between v_i and v_j is created. In step 12, the augmented graph is provided to a spectral clustering algorithm [10]. The resulting clustering is finally returned to M-CRAG.

Algorithm 2 CRAG($G, attrSet, d, k$)

Input:
- Attributed Graph $G = (V, E, \Lambda, f)$
- $attrSet$ = set of attributes
- d = neighborhood distance
- k = number of clusters

Output: one clustering solution of vertices in G.

1: $E^{'} \leftarrow \emptyset$
2: $s \leftarrow$ getSimilarityThreshold($G, attrSet, d$)
3: **for all** $v_i \in G.V$ **do**
4: $\mathcal{N}_{v_i} \leftarrow$ getNeighborhood(G, v_i, d)
5: **for all** $v_j \in \mathcal{N}_{v_i}$ **do**
6: **if** similarity($v_i, v_j, attrSet$) $\geq s$ **then**
7: $E^{'} \leftarrow E^{'} \cup \{edge(v_i, v_j)\}$
8: **end if**
9: **end for**
10: **end for**
11: $G.E \leftarrow G.E \cup E^{'}$
12: **return** spectralClustering(G, k)

5. EXPERIMENTAL STUDY

DataSet. `MQD500` [2] is an attributed graph dataset extracted from a Brazilian online social network named MQD [1]. This dataset is anonymized. Each vertex of this graph represents a user. An edge between two users in MQD represents a friendship relation. This dataset was generated from a random walk on the giant connected component of the entire MQD in the year 2014. For each vertex in the graph, this dataset presents two attributes, *age* and *extroversion*. In particular, the latter attribute comes from a personality test based on the Five Factor personality inventory.

Experimental Settings. We compared three kinds of clusterings for each dataset. These clusterings are labelled `Struc`, `Attr` and `CRAG`. `Struc` clusterings were generated using a spectral clustering algorithm and take into account only topological information from the graph. $\texttt{Attr}_l$ clusterings were generated taking into account only relational information from the graph in order to create a similarity matrix (by using Euclidean distance) that is used as input for spectral clustering algorithm. In $\texttt{Attr}_l$ clusterings, l is a mnemonic which indicates the set of attributes considered in the computation of similarities between vertices. Finally, $\texttt{CRAG}_l$ clusterings were produced using M-GRAG; in this case, l has the same meaning as aforementioned. Note that the set of attributes denoted by l is defined by the function chooseAttributes (see Section 4).

An important element we explored during our experiments was the number k of clusters in each clustering c. We used the knee point K [9] of the objective function curve to determine the adequate range for exploring k. The knee point computed for MQD500 dataset is 4. We varied k between 2 and 16.

We use NMI (Eq. 1) to compare the similarity of two given clusterings. If c and c' are two clusterings generated from the same dataset, and $\text{NMI}(c, c')$ is low, then c and c' are sufficiently different, i.e., they provide alternative perspectives of the data.

Another important aspect is to measure the quality of each clustering produced. Thus, following Zhou et al. [15], we use density (Eq. 2) and entropy (Eq. 3) to evaluate the quality of each clustering produced by M-CRAG. In the following, we briefly explain these two measures.

Density represents the sum of the number of intra-group edges divided by the total number of edges. This results in a value between 0 and 1, where 0 is the worst value, representing that all edges are extra-groups and 1 is the best value, representing that all edges are intra-groups.

$$D(\{V_i\}_{i=1}^{k}) = \sum_{i=1}^{k} \frac{|\{(v_p, v_q)|v_p, v_q \in V_i, (v_p, v_q) \in E\}|}{|E|} \quad (2)$$

Entropy is an uncertainty measure for a probability distribution. Eq. 3 defines the normalized entropy of attribute a_i with relation to a set of clusters $\{V_i\}$, where $H(a_i, V_j)$ represents the entropy of a_i on cluster V_j. The computation of $H(a_i, V_j)$ varies according to the type of attribute. For discrete attributes a_i and continuous attribute a_i, please refer to Shannon [13] and Lazio et. al. [7] respectively. We constraint the value of entropy to be in the interval [0,1] by using a normalizing constant $\mathcal{Z}$ in Eq. 3.

$$\text{S}(a_i, \{V_i\}_{i=1}^{k}) = \frac{1}{\mathcal{Z}} \sum_{j=1}^{k} \frac{|V_j|}{|V|} H(a_i, V_j) \quad (3)$$

It is worth pointing out that density and entropy are adequate to topological and relational perspectives, respectively. Thus, it is expected that `Struc` and `Attr` clusterings evaluate well according to density and entropy, respectively. Hence, `Struc` and `Attr` clusterings can be considered as baseline references with relation to clustering quality, for density and entropy, respectively.

Results. Following the experimental settings specified, the plots presented in this Section uses labels g, a, e, ae to identify clusterings that were generated considering attributes *gender*, *age*, *extroversion* and a combination of *age* with *extroversion*, respectively.

We start analyzing NMI considering `Struc` and all clusterings produced by M-CRAG algorithm. Figure 1 shows that most comparisons between clusterings are below 0.1, indicating that generated clusterings are very dissimilar. M-CRAG was capable of generating non-redundant clusterings.

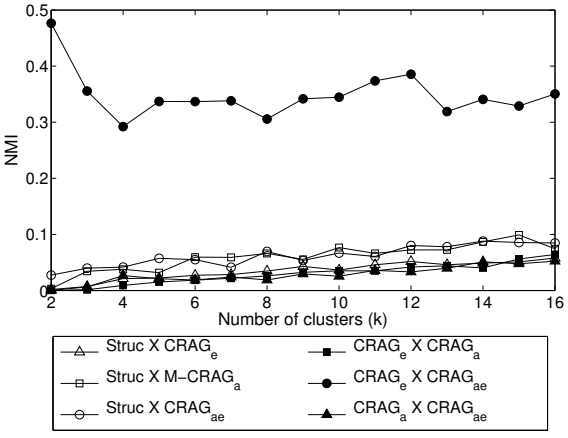

Figure 1: NMI values for MQD dataset.

In fact, only the comparison of $\texttt{CRAG}_e$ and $\texttt{CRAG}_{ae}$ results in a value above 0.1. We observe that 80.70% of the edges in the attributed graph from which $\texttt{CRAG}_e$ was generated are the same as $\texttt{CRAG}_{ae}$. There is a large number common artificial edges in both M-CRAG_e and M-CRAG_{ae} clusterings. This can be explained by the data distribution of *age* and *extroversion*. Since *age* data distribution has a lower variance (0.006) than *extroversion* (0.025), the latter has performed more relevant while computing the similarity between vertices. This behavior does not occur between $\texttt{CRAG}_a$ and $\texttt{CRAG}_{ae}$, in which only 53.64% of edges are common.

Following NMI results, we present the quality of each clustering produced, according to density and entropy measures. Figure 2 shows density values of different clusterings generated for MQD500 dataset. All `CRAG` clusterings had a performance in between $\texttt{Attr}_{ae}$ and `Struc`, which is a good result.

In the case of entropy, we analyze both attributes: *extroversion* and *age*. Figure 3 shows entropy values for *extroversion* in MQD500 dataset. $\texttt{CRAG}_e$ provides the best performance from topological based clusterings. $\texttt{CRAG}_a$ and `Struc` show the worst performance. Their bad performance are due to their distance function that does consider extrover-

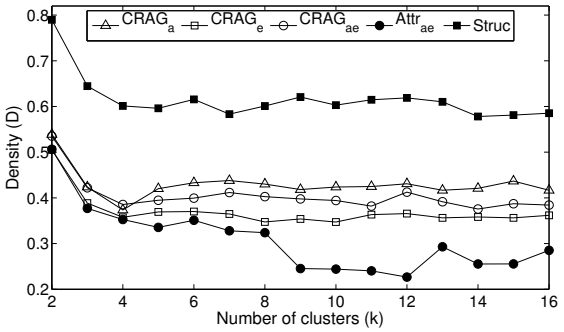

Figure 2: Density of MQD dataset.

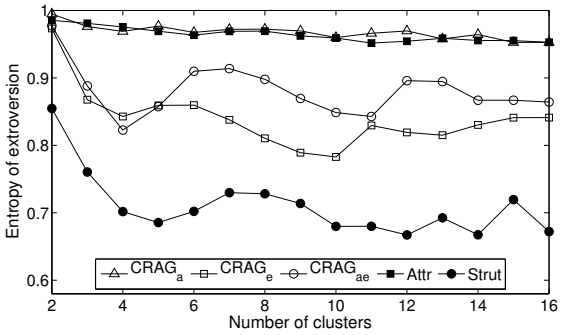

Figure 3: Entropy for extroversion attribute of MQD dataset.

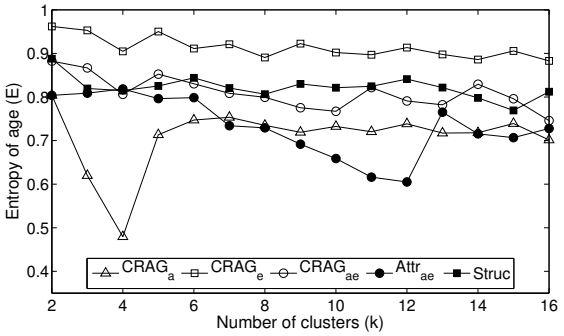

Figure 4: Entropy for age attribute of MQD dataset.

sion attribute. Meanwhile, $\texttt{CRAG}_{ae}$ shows a middle term performance between $\texttt{CRAG}_a$ and $\texttt{CRAG}_e$. Given that the goal of M-CRAG is to produce alternative clusterings, it is important to mention that none of the generated clusterings had an entropy significantly lower than the entropy of `Struc`.

Figure 4 shows entropy values for *age* attribute in MQD500 data set. Following the aforementioned reasons, the best entropy from topological based clusterings was obtained by $\texttt{CRAG}_a$. Sometimes it has outperformed $\texttt{Attr}_{ae}$ clustering (for $k = 3$, 4, 5, 6 and 13). $\texttt{CRAG}_e$ showed the worst performance, whereas $\texttt{CRAG}_{ae}$ shows a middle term performance between $\texttt{CRAG}_a$ and $\texttt{CRAG}_e$.

6. CONCLUSIONS

In this paper, we explored the problem of generating multiple non-redundant clusterings in attributed graphs. Our approach combines relational and topological information present in a graph. We introduced a new multiple clustering algorithm, M-CRAG, which generates non-redundant alternative clustering solutions. The main objective of M-CRAG is not to produce better quality clusterings, when compared to clusterings generated taking either topological or relational information into account. Instead, it aims at producing alternative clusterings with similar quality. Indeed, our experiments in a real dataset (MQD500) indicate that M-CRAG is able to find alternative non-redundant clusterings with comparable quality.

7. REFERENCES

[1] Meu querido diário. `http://www.meuqueridodiario.com.br`.

[2] Mqd500 dataset. `http://sourceforge.net/p/gpca/wiki/MQD500/`.

[3] E. Bae and J. Bailey. Coala: A novel approach for the extraction of an alternate clustering of high quality and high dissimilarity. In *ICDM*, pages 53–62. IEEE Computer Society, 2006.

[4] Y. Cui, X. Z. Fern, and J. G. Dy. Learning multiple nonredundant clusterings. *ACM Trans. Knowl. Discov. Data*, 4(3):15:1–15:32, Oct. 2010.

[5] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review, 1999.

[6] R. Koch. *The 80/20 Principle: The Secret of Achieving More with Less*. A Currency book. Doubleday, 1999.

[7] A. C. G. V. Lazo and P. N. Rathie. On the entropy of continuous probability distributions (corresp.). *IEEE Transactions on Information Theory*, 24(1):120–122, 1978.

[8] E. Müller, S. Günnemann, I. Färber, and T. Seidl. Discovering multiple clustering solutions: Grouping objects in different views of the data. In G. I. Webb, B. L. 0001, C. Zhang, D. Gunopulos, and X. Wu, editors, *ICDM*, page 1220. IEEE Computer Society, 2010.

[9] H. Munaga, M. D. R. M. Sree, and J. V. R. Murthy. Article: Dentrac: A density based trajectory clustering tool. *International Journal of Computer Applications*, 41(10):17–21, March 2012. Full text available.

[10] D. F. Nettleton. Data mining of social networks represented as graphs. *Computer Science Review*, 7:1–34, 2013.

[11] M. Newman. *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA, 2010.

[12] D. Niu, J. G. Dy, and M. I. Jordan. Multiple non-redundant spectral clustering views. In J. Fürnkranz and T. Joachims, editors, *ICML*, pages 831–838. Omnipress, 2010.

[13] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379 – 423, Jul 1948.

[14] A. Strehl and J. Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.*, 3:583–617, Mar. 2003.

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