Deep Graph Clustering in Social Network

Pengwei Hu
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
csphu@comp.polyu.edu.hk

Keith C. C. Chan
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
keith.chan@polyu.edu.hk

Tiantian He
Department of Computing
The Hong Kong Polytechnic University
Hong Kong
csthe@comp.polyu.edu.hk

ABSTRACT
In this paper, we present deep attributes residue graph algorithm (DARG), a novel model for learning deep representations of graph. The algorithm can discover clusters by taking into consideration node relevance. DARG does so by first learns attributes relevance and cluster deep representations of vertices appearing in a graph, unlike existing work, integrates content interactions of the nodes into the graph learning process. First, the relevance of contents between each node pair within the network is abstracted. Then we turn the problem of computing the first k eigenvectors in spectral clustering into a computing deep representations task. This model just need learns content information to represent vertices appearing in a graph and without the need for considering topological information. Such content information is much easier to obtain than topological links in the real world. We conduct an experiment on SNAP Facebook dataset, empirical results demonstrate that proposed approach significantly outperforms other state-of-the-art methods in such task.

Keywords
Graph clustering, Community detection, Attributes association

1. INTRODUCTION
A number of real-world problems about discovering organizational principles can be represented as complex networks with vertices representing different objects and edges representing relationships between objects [1]. In particular, social networks is a typical complex network can be viewed as relevant communities where a community would contain users that are connected tighter. Identifying such communities of users has proven to be a challenging task due to the lack of whole topological information, a plethora of attributes mixed in contents and intractability of methods for detecting them.

There are several methods to clustering social network by use link or content information, such as Spectral Clustering cluster nodes with single modal similarity [2] and iTopicModel consider both link and content information [3]. In [4], this problem solved from a deep neural network view. However, we should know various attributes can be considered as associating with different objects, other than private topological link structure. Social networks contain rich content information like tags and profiles, and researchers are very interested in finding significant communities of social networks. It is desirable to have a way to exploit the deep relevance of users and content internal relation. Hence, we introduce DARG, a novel deep attributes residue graph method, which extracts relevant attribute pairs first and feeds them to a deep neural network for reconstructing new representations to clustering step.

WWW 2017 Companion, April 3-7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3051158

2. METHODOLOGY

2.1 Attributes Relevance
Different attributes value may have different contributions in nodes clustering. Interesting attribute value pair also has influence in the clustering job. For example, education degree could be a good attribute for grouping users with similar institutions. Hence, we like to make use of user attribute values relevance to eliminate the irrelevant content information while generating new nodes similarity within relevant internal attributes for the clustering task. To discover all such kinds of associated pairwise attribute values, we bring in a residual analysis approach from [5-6] to make a reasonable statistics to prove whether there is a relevance of $attribute\ value_p$ and $attribute\ value_j$.

Let us consider cor_{pj} as the number of users that have $attribute\ value_p$ and also have $attribute\ value_j$. We define $exp_{pj} = \frac{cor_{p+}cor_{+j}}{T}$ as the desired value of cor_{pj}, where $cor_{p+} = \sum_{k=1}^{n} cor_{pk}$, $cor_{+j} = \sum_{k=1}^{m} cor_{kj}$ and $T = \sum_{m,n} cor_{mn}$. We bring in an approach from [6] to make a reasonable statistics to prove whether there is a relevance of two attribute values.

$$R_{pj} = \frac{z_{pj}}{\sqrt{(1-\frac{cor_{p+}}{T})(1-\frac{cor_{+j}}{T})}} \quad (1)$$

where $(1-\frac{cor_{p+}}{T})(1-\frac{cor_{+j}}{T})$ is defined as the maximal likelihood of z_{pj}.

$$z_{pj} = \frac{cor_{pj}-exp_{pj}}{\sqrt{exp_{pj}}} \quad (2)$$

R_{pj} has an approximately normal distribution with a mean of approximately zero and a variance of approximately one. If the value of R_{pj} exceeds1.96 [6], it would be considered there is a correlation between $attribute\ value_p$ and $attribute\ value_j$. We use these to evaluate whether the strong relevance exists in two attribute values. After determining the association between attribute values, we then move to the similarity problem base on such associations. To achieve a similarity matrix of users, we assess nodes similarity by use of Jaccard similarity to calculate new associations of each user.

2.2 Deep Graph Representation

Let $G = (V, E)$ be a graph with vertex set $V = \{v_1, \ldots, v_n\}$ representing all the nodes. Two vertices have a similarity s_{ij} between the two nodes, and the edge is weighted by s_{ij}. To obtain new representation of normalized similarity matrix, we use the same protocol as in [2]. We like give a graph G with its similarity matrix S, then we send normalized S as n nodes to Autoencoder. Generally, the optimization target of Autoencoder is to minimize the reconstruction error for the output $h_{W,b}(x) \approx x^{(i)}$ can approximate the input training set $D^{-1}S$. We define ρ as a sparsity parameter and s_2 as the number of the hidden neurons. We use β controls the weight of the sparsity penalty term. The overall cost function of Sparse Auto-Encoder can be defining as:

$$J_{sparse}(W,b) = J(W,b) + \beta \sum_{j=1}^{s_2} KL(\rho||\hat{\rho}_j) \quad (3)$$

Where

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})] \quad (4)$$

$$KL(\rho||\hat{\rho}_j) = \rho \log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \quad (5)$$

Eq. (4) is average activation of hidden unit j and we need to enforce the constraint $\hat{\rho}_j = \rho$ and Eq. (5) is the Kullback-Leibler divergence [7]. We consider stack Sparse Auto-Encoders layer by layer to form a whole deep neural network. And then we apply K-means to compute clusters of users.

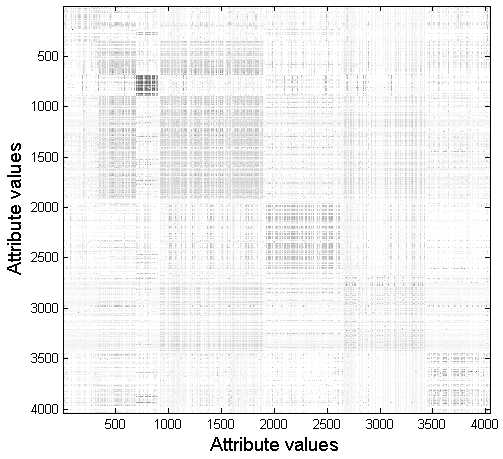

Figure 1. Calculated nodes similarity degree matrix with attribute relevance.

3. RESULTS AND DISCUSSION

The data which are used to clustering social network communities come from Stanford Large Network Dataset Collection [8]. It collected from Facebook and there are 4039 nodes, 88234 edges consist in this dataset. According to their description, the user profiles associated with users can amount to 175 which include birthday, education, languages work, etc. That is to say, each user can be encoded to 175 attributes as representation. Moreover, they also identified 193 known communities which mean our standard target clusters is 193. After running content relevance calculation, we have obtained the similarity matrix based on content relevance. In Fig.1 the calculated new nodes similarity is shown in a gray picture. NMI is an information-theoretic measure that can measure the degree of matching between the clusters discovered by proposed algorithms. We adopted NMI as the evaluation metric in our experiment. We also obtained performance of some clustering methods include FCAN [5], iTopicModel [3] and original Spectral Clustering [2] with the Facebook dataset. The experimental results on all methods are shown in Table 1. In addition, iTopicModel and FCAN also take consideration into both link information and content information. We can see that proposed approach performed better than iTopicModel and SGC and very close to FCAN in terms of NMI. In this regard, the proposed method just use content information and achieve a satisfactory performance.

Method	**DRAG**	FCAN	iTopicModel	SGC
NMI	**0.483**	0.49	0.34	0.39

Table 1. Comparison of NMI scores for Facebook

4. CONCLUSIONS

A graph based approach for social network community clustering is proposed. In summary, we introduce attributes relevance for content information to learn better nodes similarity and take Stacked Autoencoder to transform the calculated graph similarity matrix to the output graph embedding. Based on attributes measure, we can uncover some clues that mean vertices belong to the same cluster. Deep learning also subtly replace the step of find k largest eigenvalues of the normalized graph similarity matrix in spectral clustering. Experimental results on ego Facebook dataset demonstrate that DARG achieves a good performance under readily accessible content information.

5. REFERENCES

[1] M. Girvan and M. E. Newman. 2002. Community structure in social and biological networks. *Proc. Nat. Acad. Sci*., vol. 99, no. 12, pp. 7821–7826.

[2] U. Luxburg. 2007. A tutorial on spectral clustering. *Statist. Comput*, vol. 17, no. 4, pp. 395–426.

[3] Y. Sun, J. Han, J. Gao, and Y. Yu. 2009. iTopicModel: Information network integrated topic modeling. In *Proc. 9th IEEE Int. Conf. Data Mining*, pp. 493–502.

[4] Tian, F., Gao, B., Cui, Q., Chen, E. and Liu, T.Y. 2014. Learning Deep Representations for Graph Clustering. In *AAAI,* pp.1293-1299.

[5] Hu, L. and Chan, K.C., 2016. Fuzzy Clustering in a Complex Network Based on Content Relevance and Link Structures. *IEEE Transactions on Fuzzy Systems*, 24(2), pp.456-470.

[6] K. C. C. Chan, A. K. C. Wong, and D. K. Y. Chiu. 1994. Learning sequential patterns for probabilistic inductive prediction. *IEEE Trans. Syst., Man, Cybern*, vol. 24, pp.1532 -1547.

[7] Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science,* 313(5786):504–507.

[8] J. McAuley and J. Leskovec. 2012. Learning to Discover Social Circles in Ego Networks. *In Advances in neural information processing systems* (pp. 539-547).