COMBINED: COmmunity Mining By Inducing Node & Edge Data

Mittul Desai

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Master of Technology

Department of Computer Science & Engineering

June 2016

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

(Signature)

(Mittul Desai)

CS14MTECH11008

(Roll No.)

Approval Sheet

This Thesis entitled COMBINED: COmmunity Mining By Inducing Node & Edge Data by Mittul Desai is approved for the degree of Master of Technology from IIT Hyderabad

C. Krishna Mohan

(——————) Examiner
Dept. of CSE
IIT Hyderabad

(——————) Examiner
Dept. CSE
IIT Hyderabad

V. B. Vineeth

(Dr. Vineeth N Balasubramanian) Adviser
Dept. of CSE
IIT Hyderabad

U. Ram

U. RAMAKRISHNA

(——————) Chairman
Dept. of CSE
IIT Hyderabad

Acknowledgements

I thank my Thesis Guide Dr. Vineeth N B for his invaluable support and guidance and for being accomodative of all my activities and ideas.
I also would like to thank my classmates for being the pinnacle of support in my rough times and helping me cope through my illness and problems.

Abstract

Community Detection is the process of identifying a group of nodes in a graph that are distinguishable in some context. Two sources of information have been studied in detail by the community namely: Edge Structure and Node Attributes. Most work only deals with one of the above methods. However there is some work in the recent years that use them to complement each other. In this paper, we aim to add a new dimension to the problem namely, *edge attributes*. In addition to using the aforementioned methods we add edge attributes to detect communities. Edge attributes uncover micro-communities that might not be easily retrievable using Node attributes and Edge Structure. Especially in social networks, edge attributes might constitute a large part of the information content. This information along with contextual information of the edge helps uncover previously undisclosed communities. It also helps uncover disjoint and overlapping communities. Our approach uncovers the related attributes that form the community.

COMBINED:COmmunity Mining By Integrating Node and Edge Data

Mittul Desai
Indian Institute of Technology
Hyderabad, India
cs14mtech11008@iith.ac.in

Dr. Vineeth N B
Indian Institute of Technology
Hyderabad, India
vineethnb@iith.ac.in

ABSTRACT

Community Detection is the process of identifying a group of nodes in a graph that are distinguishable in some context. Two sources of information have been studied in detail by the community namely: Edge Structure and Node Attributes. Most work only deals with one of the above methods. However there is some work in the recent years that use them to complement each other. In this paper, we aim to add a new dimension to the problem namely, *edge attributes*. In addition to using the aforementioned methods we add edge attributes to detect communities. Edge attributes uncover micro-communities that might not be easily retrievable using Node attributes and Edge Structure. Especially in social networks, edge attributes might constitute a large part of the information content. This information along with contextual information of the edge helps uncover previously undisclosed communities. It also helps uncover disjoint and overlapping communities. Our approach uncovers the related attributes that form the community.

1. INTRODUCTION

Graph systems are irregular structures. They cannot be modelled like a lattice and euclidean measure of distance does not work on graphs. In a random graph, distribution of edges in the graph is more or less homogeneous. Real networks are almost always non-homogeneous. The graph is usually divided into areas of high interaction and areas of low interaction. This is also termed as the small world problem, the fact that there are too many nodes in a graph but the number of nodes a particular node connects to is a small fraction of the total nodes in the graph.

It hence becomes important to be able to identify how these high interaction areas or communities are formed and how can we detect them. Even in our society, communities play a pivotal role in the understanding of the social construct. Society is organized into families, friend cirlces from school and work, villages, towns, religions, etc. Internet has led to a notion of online communities where people with similar interests come together without the geographical barrier. The increasing use of social media and networking is apparent and understanding these 'online' social communities is of interest.

The rest of the paper is divided into 6 sections: Literature survey of the community detection approaches and methods, Related Work to the paper and the proposed method, Motivation for choosing the problem statement, Model description, Experimental Evaluation of our approach and Conclusion.

2. LITERATURE SURVEY

2.1 Types of Communities

The definition of 'Community' is taken in context of the problem that an approach is trying to solve. In this section we go over the types of communities that the community has tried to uncover using various methods. These are broad classifications that can be interpreted in multiple ways as well.

- **Clique**
  A clique is defined as a Maximally Connected Component(MCC) found as a subset of the graph.[8]
- **n-clique**
  An n-clique is either a MCC or a group of nodes with a minimum of n connections amongst each other.[9]
- **k-Plex**
  A k-plex is a Completely Connected Component(CCC) on all nodes of the set except for k nodes.[10,11]
- **LS-set**
  An LS set is a weak form of community where for the group of nodes the internal degree of nodes should be greater than external degree.[13]
- **Lambda set**
  In a Lambda set, edge set of any two vertices in the sub graph is larger than the edge set of a vertex inside the subgraph to a vertex outside the subgraph.[15]
- **Fitness Measure**
  Inclusion of a node in the community depends on the fitness measure of the node in context of a particular community.[13]

- **Modularity based communities**
  The connections between the nodes inside the subgraph should be dense as compared to the boundary nodes.[14]

- **Clusters**
  Communities arrived at using clustering methods.[20]

2.2 Community Detection Approaches

Several methods have been proposed to uncover communities in graphs. In this section we take a rundown some of the common approaches taken for community detection.

- **Graph Partitioning Methods**

  Graph partitioning is the process of converting a graph into groups of vertices of a predefined size such that the edges between such groups are minimal. Number of edges between groups is called *cut size*

  It is important in this method to know the number of partitions and size of partitions beforehand as the trivial solution for not knowing the number of partitions is to have a single large cluster and trivial solution for not having a fixed cluster size is to remove a single node from the graph with minimum degree from the graph[16].

- **Centrality measures**

  In centrality measures, each vertex is considered a data point in space, several distance measures can be used to model the distancce between the points. Several methods are termed under centrality measures, [18,19].

  Minimum k-clustering The diameter of the cluster is the cost function here that is to be minimized. The points are classified so that the diameter minimizes itself.

  k-clustering sum Similar to minimum k-clustering, but the measure used here is the average distance between all points in a cluster.

  k-center Also known as k-means clustering a centroid is iteratively computed to minimize the distance of every point from atleast one cluster center.

  k-median Similar to k-center except for the minimization of median rather than average distance between all points in a cluster.

- **Hierarchical methods for centrality**

  Hierarchichal clustering techniques reveal the multilevel struture of a graph. It is commonly used in social network analysis and marketing. A similarity measure is defined between vertices, after which every pair of vertices are checked for similarity. In agglomerative algorithms, clusters are merged if they have a high enough similarity score while in divisive algorithms, clusters are iteratively split by removing edges to improve overall similarity score.[17]

- **Modularity based methods**

  This is the most popular method of community detection in the literature. Even for a small graph an Exhaustive optimization is impossible due to a large number of ways that can be used to paritition a graph.

  Greedy Techniques. Newman proposed a greedy method by starting with n vertices forming n clusters and adding edges to the graph one-by-one. Edges are chosen to maximize the increase of modularity of the graph. The same is continued till adding any further edge results in a decrease of modularity.[24]

  Simulated Annealing. Guimera and Amaral use simulated annealing in the form of two types of moves, local moves are when a single node is shifted from one cluster to another and a global move that comprises of merging or splitting of communities. All the moves are done through a probabilistic model that uses noise to guard against a local maxima.[25]

  Extremal Optimization. Extremal Optimization is a heuristic search procedure proposed by Boettcher and Percus to achieve substantial gain in computer time at the same accuracy as simulated annealing. The local modularity of a vertex is obtainedthe value of the term in the corresponding sum of local variables expressed as the contribution of the global function at study. A fitness measure is obtained by taking a ratio of the local modularity by its degree.[26]

  Spectral Optimization. Modularity is optimized using the eigen values and eigen vectors of a special matrix.[27]

- **Spectral Partitioning methods**

  Spectral clustering covers all the methods that use the eigenvectors of matrices to partition a graph into clusters. [21]

- **Divisive Algorithms**

  Edge centrality is the measure of the number of shortest paths that an edge belongs to in a given graph. The idea in divisive algorithms is that edges with very high centrality values connect two communities. Thus such edges are removed from the graph to uncover communities. This is done iteratively as edge centrality values tend to change after every such edge is removed.[22,23]

- **Random Walks**

  Random walks are also useful in finding communities in graphs due to the higher percentage of paths leading into a community than out of the community. This means that a random walker spends far more time inside the cluster than outside it. The average number of nodes that a random walker has to cross to reach from a node i to a node j is the distance from i to j, the lower this distance is, the higher the probability that these two nodes are in the same community. [29,30]

- **Statistical inference methods**

  Statistical inference as Mackay puts it aims at deducing properties of data sets, starting from a set of observations and model hypotheses. If the data set is a graph, the model, based on hypotheses on how vertices are connected to each other, has to fit the actual graph topology.[31]

- **Multi-resolution methods**

  In general, multiresolution methods have a freely tunable parameter, that allows to set the characteristic size of the clusters to be detected. The general spin glass framework by Reichardt and Bornholdt is a typical example, where is the resolution parameter. The extension of the method to weighted graphs has been recently discussed.[33,34]

- **Clique percolation methods**

  Clique percolation is based on the concept that cliques are formed inside communities due to the high density of edges inside the community. Also the boundaries between communities having a low density of edges are less likely to produce edges. A k-clique has been mentioned earlier. If two k-cliques have k-1 common edges, they are termed as adjacent k-cliques. A sequence of adjacent k-cliques froms a k-clique chain and the largest connected subgraph formed by k-cliques is the community for that graph.[32]

- **Overlapping Community Detection methods** A community is defined as a part of the subgraph that has a denser set of edges compared to the nodes around it. This can be termed as optimizing a local function with some measure around the edge density.
  Iterative Scan performs a greedy optimization of the function W, starting at a random seed and adding, deleting edges or vertices till the function W cannot be optimized any further.[35] *Rank Removal* removes important vertices from the graph to disconnect the graph into smaller components that are sparsely connected. The importance of the vertex is determined by its centrality score.[36]

- **Heuristic methods** Communities can also be found by looking for non intersecting paths in a graph.

3. RELATED WORK

Community detection can be viewed as a clustering problem where every node belongs to multiple communities. A node shares connections with multiple other nodes and has multiple attributes which leads to participation in multiple communities. Traditionally, two sources of information have been used, namely, Edge Structure and Node Attributes. Node attributes can be user's social network profile and edge structure can be the friends that a user has.

However, community detection algorithms mainly focus on one of the two sources only. Clustering algorithms mostly use node attributes and community formation algorithms resort to edge structure.

An algorithm may not produce optimal communities by only focusing on one of these two sources. They can be combined to give a better view of the communities underlying in a social graph. For example, Attributes help can be useful to find communities for a node with very few edges. In the same way two nodes with identical edge structure may belong to the same community even if one of the nodes has no node attributes.

Recent approaches have used both these sources together. These methods are still naive in adding the two sources as they consider them as independent of each other. These papers use soft membership models that limit the membership of a node in a community by using a common generator for community membership.

A recent paper - the state of the art - tried to solve these issues by using an efficient, scalable, hard modelled approach for community detection. Their approach also encompassed overlapping communities.

We try to model a method that encompasses overlapping, disconnected communities leveraging contextual information. Our model is scalable and is tested in the real world large scale networks.

Our approach is based on a generative model for networks with node & edge attributes. Our approach advances on existing approaches and provides better accuracy as well as scalability. Our approach detects overlapping communities by modelling hard node-community memberships. Soft membership models take the assumption that nodes that share multiple common communities are less likely to be connected. We assume that communities may generate both edge and node attributes and the network. This allows for dependence between network and attributes.

To the best of our knowledge, CDSGEANI(this paper) is the first paper that employs all the three sources of information in a graph while also modelling contextual information. CDSGEANI can detect overlapping, non-overlapping, hierarchical as well as disconnected communities.

4. MOTIVATION

- In email networks, an email between two users can be considered an edge, which has content in terms of the text communicated between the two users. Clearly, users with similar content of communication are much more likely to belong to the same community than those which do not. This observation also applies to other forms of text or chat networks, or even (threaded) community boards which enable interaction between specific pairs of participants.[3]

- In social networking platforms, users may tag an image with keywords. In such cases, it may be possible to construct a network of both people and images in which the edge content corresponds to the keywords which are used for tagging. Such tags provide insightful knowledge about the nature of the underlying community.[2]

- In social media, users may share authorship or browsing behavior for the same content. In such cases,one can create an actor-centric network in which edges are placed between users that share the same content, and the shared content is associated with that edge. Thus, each content-based sharing may induce an edge between two participant nodes.[1]

5. MODEL DESCRIPTION

A lot of study went into understanding how to combine multiple clustering formed using separate data points like edges and nodes. We start this section with a brief overview of Multi-view Clustering and continue to explain Consensus clustering and then go on to propose our model.

5.1 Multi-view clustering

In some applications, the data space can be split into two sets of dimensions that can each be used separately to learn communities in itself. A well known example is that of the web, where web pages can be classified either by the anchor text of inbound links as well as the content of the web pages. Multi-view algorithms train two independent hypotheses which bootstrap to provide labels for unlabeled data. It has been shown that the disagreement of the two independent theses serves as the bound on the error rate. Multi-view clustering approaches use a classifier based on some context of view one and senses the labels of the other view to iteratively bootstrap each other.[5]

The co-EM algorithm[5] is a multi-view version of the EM algorithm for semi-supervised learning. The authors try to use the same algorithm in the unsupervised setting. The algorithm runs E and M step on one view of the data and uses the output as the initial estimations for the E step of the next view. This is repeated iteratively until a convergence is reached. This algorithm seldom converges, so special stopping criteria are used to stop the iterations. [6,7]

Input: Unlabeled Data D $= \{(x_1^{(1)}, x_1^{(2)}), ..., (x_n^{(1)}, x_n^{(1)})\}$.

1. Initialize $\theta_0^{(2)}, T, t = 0.(0)$

   E step view 2: compute expectaion for hidden variables given the model parameters $\theta_0^{(2)}$

   Do until stopping criterion is met:

   1. For $v = 1...2$:
      (a) t = t + 1
      (b) M step view v: Find model parameters $\theta_t^{(v)}$ that maximize the likelihood for the data given the expected values for the hidden variables of view v of iteration t-1.
      (c) E step view v: compute expectation for hidden variables given the model parameters $\theta_t^{(v)}$.
   2. End For v.

   return combined $\theta = \theta_[t-1]^{(1)} U \theta_t^{(2)}$

5.2 Consensus clustering

The problem we faced with multi-view clustering was that the views need to be on the same data space. An example would be to have a partition on the node attributes and use them to bootstrap each other iteratively.

Our problem was a little different, one clustering is to be done in the node space and the other one in the line graph space or edge space. Thus there arose a need to look further. A way to combine already clustered data into an agreeable cluster.

Consensus clustering[6] especially deals with the problem of combining multiple clusters of a dataset without accessing the original attributes. It is termed as a ensemble clustering problem. Three consensus functions are used in the paper[6].

1. **Cluster Similarity Partitioning Algorithm (CSPA).** Pairwise similarity is used to establish a relationship between the ensemble of clusters. Pairwise similarity is calculated based on the fact that two nodes belonging to the same community denotes a stronger connection between them.

2. **HyperGraph Partitioning Algorithm (HGPA).** This algorithm estimates the maximal mutual information with a constraint of the minimum cut. The problem is posed as a partitioning problem of a hypergraph where the hyperedges are made of clusters.

3. **Meta-CLustering Algorithm (MCLA).** A hypergraph is created with nodes as clusters and a partitioning algorithm is applied on the same.

5.3 Proposed Methodology

We develop a probabilistic model that combines edge & node attributes with edge structure and community memberships. Our model is based on these intuitions.

- Nodes in the same community are likely to be connected.
- Communities may overlap.
- Number of communities common between two nodes increase the edge probability.
- Nodes in the same community mostly share common attributes.

We assume that there are N nodes in the network G, each node has K attributes and C communities. We denote the network by G, the node attributes by X (X_{uk} is the k-th attribute of node u) and community memberships by F. We assume for community memberships F, every node u has a positive affiliation weight $F_{uc} \in [0, \infty). If \mathrm{F}_{uc} = 0$, node u does not belong to community c.

Modeling the edges. We consider these three intuitions here:

1. Node-Community affiliation(F) affects the likelihood that a pair of nodes is connected.
2. The effect of every affiliation is different.
3. These effects are independent of one another.

To establish this, we build an affiliation network, where the graph *G(V,E)* is developed from affiliations F_{uc}. We assume that two nodes *u,v* belonging to a community c are connected with the probability:

$$P_{uv}(c) = 1 - exp(-F_{uc}.F_{vc}).$$

Here for any *u,v* if F_{uc} or F_{vc} are 0 then $\mathrm{P}_{uv}(c) = 0$.

For *u,v* to be disconnected, nodes u and v should not be connected in any community c.

$$1 - P_{uv} = \prod_c (1 - P_{uv}(c)) = exp(-\sum_c F_{uc}.F_{vc}).$$

Thus,

$$P_{uv} = 1 - exp(-\sum_c F_{uc}.F_{vc}).$$

$$A_{uv} \sim Bernoulli(P_{uv}).$$

Modeling Node Attributes. Community affiliations can also be used to model node attributes.

Our intuition here is that, based on a node's community affiliations we can predict the value of each of the node's attribute values. Thus we regard F as the input to the logistic model with associated logistic weight factor W_{kc} (for each attribute k and community c).

$$Q_{uk} = \frac{1}{1 + exp(-\sum_c W_{kc}.F_{uc})}$$

$$X_{uk} \sim Bernoulli(Q_{uk}).$$

where W_{kc} is a real valued logistic model parameter for community c and attribute k. The value of W_{kc} is the relevance of each attribute k to the community c.

5.4 Algorithm

Input: A graph G, with Edge Attributes E and Node Attributes N.

1. Create a Line Graph L from the original Graph. Note that Node attributes of Line Graph, LN = E.
2. Form clusters on N and LN using any clustering algorithm.
3. Convert the clustering on LN into clusters of N by using a majority vote node inclusion policy.
4. Run Ensemble Clustering to find a consented clustering on the dataset.

6. EXPERIMENTAL EVALUATION

Twitter data is available in three versions, namely: Spritzer, Sprinkler and Garden Hose which are 1%,10% and 100% of all the tweets. The first version is free for use.

1% of all tweets on twitter accumulates to around 120000000 tweets or 120 million tweets over a single month. We will try to access the Garden Hose version if possible.

This will test its scalability and considering every tweet as an edge will garner tremendous information from the edge attributes namely tweet text and hashtags.

7. CONCLUSION

This paper, when ready, will present a new source of information extraction from social graphs and will be excellent for community detection. Dynamic communities that form momentarily can be captured very well with this approach. This approach will open new avenues for community detection in social graphs.

References

1. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In NIPS 12, 2012.
2. J. Wong, J. McAuley and J. Leskovec. Community Detection in Networks with Node Attributes. In ICDM 13, 2013.
3. Qi, Guo-Jun, Charu C. Aggarwal, and Thomas Huang. Community detection with edge content in social media networks. Data Engineering (ICDE), 2012 IEEE 28th International Conference on. IEEE, 2012.
4. Fortunato, Santo. Community detection in graphs. Physics Reports 486.3 (2010): 75-174.
5. Bickel, Steffen, and Tobias Scheffer. Multi-View Clustering. In ICDM, vol. 4, pp. 19-26. 2004.
6. Topchy, A., Jain, A. K., and Punch, W. (2005). Clustering ensembles: Models of consensus and weak partitions. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 27(12), 1866-1881.
7. Strehl, A., Ghosh, J. (2003). Cluster ensembles—a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research, 3, 583-617.
8. R.D. Luce, Connectivity and generalized cliques in sociometric group structure, Psychometrika 15 (2) (1950) 169190.
9. R.D. Alba, A graph-theoretic definition of a sociometric clique, J. Math. Sociol. 3 (1973) 113126.
10. R.J. Mokken, Cliques, clubs and clans, Qual. Quant. 13 (2) (1979) 161173.
11. S.B. Seidman, B.L. Foster, A graph theoretic generalization of the clique concept, J. Math. Sociol. 6 (1978) 139154.
12. S.B. Seidman, Network structure and minimum degree, Soc. Netw. 5 (1983) 269287.
13. F. Luccio, M. Sami, IEEE Trans. Circuit Th. CT 16 (1969) 184188.
14. U. Feige, D. Peleg, G. Kortsarz, The dense k-subgraph problem, Algorithmica 29 (3) (2001) 410421.
15. S. Borgatti, M. Everett, P. Shirey, LS sets, lambda sets, and other cohesive subsets, Soc. Netw. 12 (1990) 337358.
16. A. Pothen, Graph partitioning algorithms with applications to scientific computing, Tech. rep., Norfolk, VA, USA, 1997.
17. T. Hastie, R. Tibshirani, J.H. Friedman, The Elements of Statistical Learning, Springer, Berlin, Germany, 2001.
18. J.B. MacQueen, Some methods for classification and analysis of multivariate observations, in: L.M.L. Cam, J. Neyman (Eds.), Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, University of California Press, Berkeley, USA, 1967, pp. 281297.
19. S. Lloyd, Least squares quantization in PCM, IEEE Trans. Inform. Theory 28 (2) (1982) 129137.
20. J. Kleinberg, An impossibility theorem for clustering, in: Advances in NIPS 15, MIT Press, Boston, USA, 2002, pp. 446453.
21. W. Donath, A. Hoffman, Lower bounds for the partitioning of graphs, IBM J. Res. Dev. 17 (5) (1973) 420425.
22. M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 78217826.

23. M.E.J. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113

24. M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.

25. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Optimization by simulated annealing, Science 220 (1983) 671680.

26. S. Boettcher, A.G. Percus, Optimization with extremal dynamics, Phys. Rev. Lett. 86 (2001) 52115214.

27. M.E.J. Newman, From the cover: Modularity and community structure in networks, Proc. Natl. Acad. Sci. USA 103 (2006) 85778582.

28. M.E.J. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006) 036104.

29. B.D. Hughes, Random Walks and Random Environments: Random Walks, vol. 1, Clarendon Press, Oxford, UK, 1995.

30. H. Zhou, Distance, dissimilarity index, and network community structure, Phys. Rev. E 67 (6) (2003) 061901.

31. D.J.C. Mackay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, Cambridge, UK, 2003.

32. G. Palla, I. Dernyi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814818.

33. H. Simon, The architecture of complexity, Proc. Am. Phil. Soc. 106 (6) (1962) 467482.

34. T. Heimo, J.M. Kumpula, K. Kaski, J. Saramki, Detecting modules in dense weighted networks with the Potts method, J. Stat. Mech. (2008) P08007.

35. J. Baumes, M.K. Goldberg, M.S. Krishnamoorthy, M.M. Ismail, N. Preston, Finding communities by clustering a graph into overlapping subgraphs, in: N. Guimaraes, P.T. Isaias (Eds.), IADIS AC, IADIS, 2005, pp. 97104.

36. J. Baumes, M. Goldberg, M. Magdon-Ismail, Efficient identification of overlapping communities, in: IEEE International Conference on Intelligence and Security Informatics, ISI, 2005, pp. 2736.

37. Karypis, George, and Vipin Kumar. "METIS–unstructured graph partitioning and sparse matrix ordering system, version 2.0." (1995).