Identifying Community Structure in Social Network with Compressive Sensing

Anonymous Author(s)
Affiliation
Address
email

Jinguang Guo, Jake Bloomfeld, Ailing Yu

Abstract

In recent years, sparsity has become a very important concept in computer science and applied mathematics. The main idea is that many classes of natural signals can be described by only a small number of significant degrees of freedom. With compressed sensing, if the true signal is sparse to begin with, accurate, robust, and even perfect signal recovery can be achieved from just a few randomized measurements. After taking Boston University CS 591 – Compressive Sensing with Professor Peter Chin, we wanted to explore this topic and its application to social networks. Specifically, we wanted to dive deeper into the concept of community structure social network and how compressive sensing techniques can be used to identify communities in a social network. This paper explores the concept along with expanding upon some theory and techniques for this approach.

1 Background

In today's world, networks are everywhere. A network is defined as a group or a system of interconnected things or people. In computing, a network is a group of two or more devices that can communicate. A social network is a similar kind of network defined as a network of social interactions and personal relationships. Whether you are part of a social network on Facebook, have a professional network at work, or simply have a group of close friends – it is nearly impossible to avoid being within a network. In the past several years, data network research has increased. Studies between relationships of interconnected data in a wide domain can answer many questions while also generating new ones. Due to the increased volume, variety, velocity and veracity of network data, new modeling tools and techniques are needed.

2 Introduction

In this section, we will introduce the concept of a *clique*. So, formally, what is a clique? A clique, C, in an undirected graph $G = (V, E)$ is a subset of the vertices, $C \subseteq V$ such that every two distinct vertices are adjacent (complete). In simpler terms, a clique is a subset of a network in which the actors (nodes) are more closely and intensely tied to one another than they are to other members of the network. In terms of 'social cliques', people in groups tend to form cliques based on their age, gender, race, religious, interests, etc. When analyzing network data, some issues may arise when attempting to identify cliques within the network. If only given data that represents a sample of a network, can cliques be identified? If so, are they accurate? These are the kind of questions we wish to explore throughout this paper.

3 Question and Problem Statement

Many questions can be formed when dealing with this problem, however, it is important to boil it down to one main question that we are trying to answer: Given sparse network data, can we successfully identify cliques within the network? When analyzing data in networks, the issue of identifying cliques based on limited information frequently arises in variety of applications, in particularly, social networks. Answering this question in regards to social networks has the potential to expand into many other different types of networks. In the case for this paper, we are given a network or graph where the nodes represent people, items, or characters, with their corresponding edge-weights representing their frequency of interaction. The problem to solve is to identify cliques within this given social network by observing the frequencies of these low-order interactions.

4 Application to Compressive Sensing

How does the concept of identifying these low-order social cliques within a network relate to what we've been learning all semester? Backing up a bit, compressive sensing is a method for identifying sparse solutions to linear systems, for example, reconstructing a signal that has a sparse representation in a large dictionary. We have the equation: $y = Ax$, where y is the compressed measurement, A is the sensing matrix, and x is the signal.

5 Simple Example

To start, consider this very simple example of a social network: let's say we are given a network of 4 people: Alex, Bob, Christine and Danielle. Alex, Bob and Christine are all in CS 591 this semester, which meets twice a week. Bob, Christine, and Danielle are all in CS 542 this semester, which meets 3 times a week. From this, we observe that Alex and Bob's interaction level is 2, Bob and Christine's is 5, and Alex and Danielle's is 0, based on how many times they are together throughout the week. Going back to the main question: would we be able to detect these two distinct social cliques?

6 Data

The social network data that we used is from Les Misérables – a French historical novel by Victor Hugo, first published in 1862, that is considered one of the greatest novels of the 19th century. This undirected network contains co-occurrences of characters in the novel. A node represents a character and an edge between two nodes shows that these two characters appeared in the same chapter of the the book. The weight of each link indicates how often such a co-appearance occurred. This data set consists 77 vertices (characters) and 254 edges (co-occurrences). The average degree is 6.6 edges per vertex, and the fill is 0.09 edges per vertex^2. The degree in this context represents the amount of relationships each character has, and the weight represents the frequency level of each relationship. The clustering coefficient is 49.9. The clustering coefficient is a measure of the degree to which nodes in a graph tend to cluster together. Evidence suggests that in most real-world networks, and in particular social networks, nodes tend to create tightly knit groups characterized by a relatively high density of ties; this likelihood tends to be greater than the average probability of a tie randomly established between two nodes. This comes into play when trying to identify cliques.

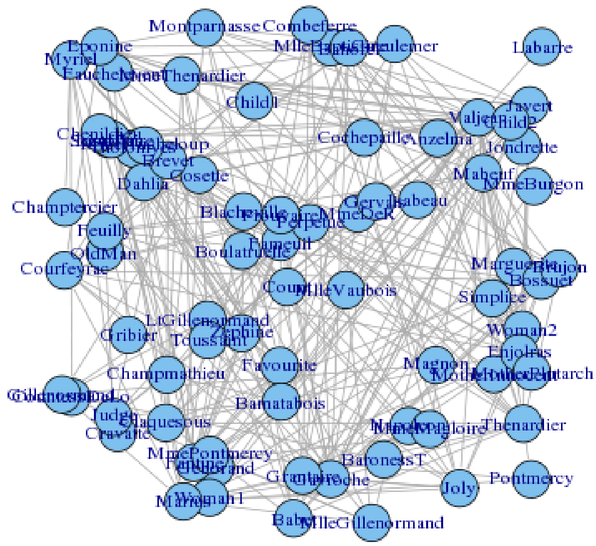

Figure 1: Les Misérables data set visualization (without showing the weight)

7 PART I: Initial Approach Using Basis Pursuit

Let's represent Let's represent a social network as a graph *G* = (*V*, *E*) where *V* = 1, ..., *n* is the set of nodes and *E* ⊂ *V* x *V* is the set of edges. Let *M* ∈ $\mathbb{R}^{nxn}$ be the adjacency matrix of the social network where an element *A*(*i*, *j*) ∈ *R* represents the interaction level between the two nodes *i* and *j*.

Consider the equation: $M(i,j) = \sum_k c_k \phi_k(i,j) + n$.

Assume that M is sparse with respect to a sensing matrix, A = [ϕ_1,...ϕ_N] where in each ϕ_k is a basis function, such that ∃ a subset S [1,...N] with a cardinality |S| « N, such that $c_k = 0$ for $k \notin S$. In this case, n represents noise, and N has no upper bound.

In the dictionary A, each column, or basis, represents a clique, which is a complete subgraph of G with a set of nodes $K \subset V$. Let ϕ_k represent the adjacency matrix of a clique with nodes K. $\phi_k(i,j)$ = 1 if $i, j \in K$ where $i\neq$ j$and 0 in all other cases.$

For simplicity, squeeze matrix M into a vector $b \in \mathbb{R}^M$ where M now equals n(n-1)/2, which is the number of elements in the original M matrix. Now, each $\phi_k \in \mathbb{R}^M$ and A is an M by N matrix.

Given A, we want to recover the sparse representation in $M(i,j)$ =$\sum_k c_k \phi_k(i,j) + n$ by reconstructing x using: $min||x||_1$ where $||b - Ax|| \leq \delta$. Entries in vector x represent unknown frequencies with high-order subsets. Entries in vector b represent observed frequencies with low-order subsets. Dictionary A acts as an operator that can map frequencies from x to b.

The basis pursuit algorithm is a technique for decomposing a signal into an "optimal" superposition of dictionary elements, where the optimization criteria is the L1-norm. Each column of dictionary A represents a clique Adjacency matrix M is a linear combination of several clique basis. Basis Pursuit is said to be "relaxed", as it uses continuous optimization techniques, which means it uses convex optimization. Idea of Basis Pursuit is to replace the difficult sparse problem with an easier optimization problem. We initially tried to implement this algorithm with the inspiration from a previous research paper that explored a similar topic and gave a novel algorithm called Randon Basis Pursuit [3].

8 PART II: Modified Approach

Our initial approach to this problem was attempting to identify cliques within a social network. After attempting to run the Basis Pursuit algorithm on our data, we ran into issues with the complexity of the algorithm: Randon Basis Pursuit mentioned in our reference was not just basis pursuit and it was way more complicated. Lack of time and detailed information of its implementation we decided to slightly modify our approach to try to identify communities in the network, instead of only to identify cliques. In the study of complex networks, a network is said to have community structure if the nodes of the network can be easily grouped into (potentially overlapping) sets of nodes such that each set of nodes is densely connected internally. In the particular case of non-overlapping community finding, this implies that the network divides naturally into groups of nodes with dense connections internally and sparser connections between groups. But overlapping communities are also allowed. The more general definition is based on the principle that pairs of nodes are more likely to be connected if they are both members of the same community, and less likely to be connected if they do not share communities. Communities are a more relaxed than cliques, as they don't have to all be connected to one another, or complete. Being able to identify communities in a network would help the process of identifying cliques, as it is more efficient and simple from a computational standpoint. Cliques are sub-graphs in which every node is connected to every other node in the clique. As nodes can not be more tightly connected than this, it is not surprising that there are many approaches to community detection in networks based on the detection of cliques in a graph and the analysis of how these overlap. As a node can be a member of more than one clique, ne community in these methods an overlapping community structure.

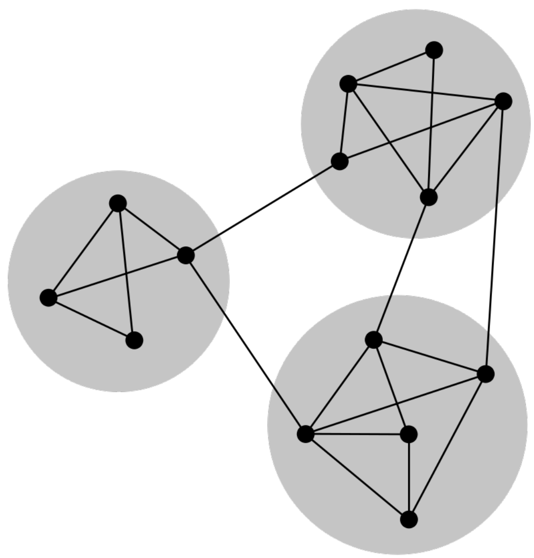

Figure 2: Network Community Structure

9 Algorithms

We used several algorithms, explained below, to help identify community structure of the characters in Les Misérables. We would compare the results to our anticipated results of Basis Pursuit in future works to get further insight.

9.1 Clique-Based Methods

9.1.1 Maximal Cliques

Our first approach was to find the maximal cliques in the social network. This consisted of finding the cliques which are not the sub-graph of any other clique. The classic algorithm to find these is the Bron–Kerbosch algorithm [7]. The overlap of these can be used to define communities in several ways. The simplest is to consider only maximal cliques bigger than a minimum size (number of nodes). The union of these cliques then defines a sub-graph whose components (disconnected parts) then define communities.

9.1.2 Fixed-Size Cliques

Another approach we had is to identify cliques of fixed size, k. The overlap of these can be used to define a type of k-regular graph or a structure which is a generalization of the line graph (the case when k=2) known as a clique graph. The clique graphs have vertices which represent the cliques in the original graph while the edges of the clique graph record the overlap of the clique in the original graph. Applying any community detection method, which assign each node to a community, to the clique graph then assigns each clique to a community. This can then be used to determine community membership of nodes in the cliques. Again, as a node may be in several cliques, it can be a member of several communities. For instance the clique percolation method defines communities as percolation clusters of k-cliques. To do this it finds all k-cliques in a network, that is all the complete sub-graphs of k-nodes. It then defines two k-cliques to be adjacent if they share k-1 nodes, that is this is used to define edges in a clique graph. A community is then defined to be the maximal union of k-cliques in which we can reach any k-clique from any other k-clique through series of k-clique adjacency. That is communities are just the connected components in the clique graph. Since a node can belong to several different k-clique percolation clusters at the same time, the communities can overlap with each other.

9.1.3 Weighted Cliques Percolation

A weighted k-clique is a complete sub-graph with k nodes such that the geometric mean of the k (k - 1) / 2 link weights within the k-clique is greater than a selected threshold value, I. The weighted Clique Percolation Method defines weighted network communities as the percolation clusters of weighted k-cliques. Note that the geometric mean of link weights within a sub-graph is called the intensity of that sub-graph.

9.2 Girvan-Newman Algorithm

Another algorithm we explored was the Girvan–Newman algorithm. This detects communities by progressively removing edges from the original network. The connected components of the remaining network after the removal are the communities. Instead of trying to construct a measure that tells us which edges are the most central to communities, this algorithm focuses on edges that are most likely "between" communities. This concept of "betweenness" is an indicator of highly central nodes in networks. For any node k, vertex betweenness is defined as the number of shortest paths between pairs of nodes that run through it. The Girvan–Newman algorithm extends this definition to the case of edges, defining the "edge betweenness" of an edge as the number of shortest paths between pairs of nodes that run along it. If there is more than one shortest path between a pair of nodes, each path is assigned equal weight such that the total weight of all of the paths is equal to unity. If a network contains communities or groups that are only loosely connected by a few inter-group edges, then all shortest paths between different communities must go along one of these few edges. Thus, the edges connecting communities will have high edge betweenness (at least one of them). By removing these edges, the groups are separated from one another and so the underlying community structure of the network is revealed. This approach gave us better insight on the clique structure of the network.

9.3 Modularity Maximization

Modularity maximization is one of the most widely used methods for community detection in networks. Modularity, is a benefit function that measures the quality of a particular division of a network into communities. Modularity maximization is to detects communities by searching over possible divisions of a network for one or more that have particularly high modularity. Networks with high modularity have dense connections between the nodes within modules but sparse connections between nodes in different modules. Modularity is often used in optimization methods for detecting community structure in networks. A popular modularity maximization approach is the Louvain method, which iteratively optimizes local communities until global modularity can no longer be improved given perturbations to the current community state.

9.4 Other Graph Clustering Methods

We also implemented several other graph clustering methods for comparison, explained below.

9.4.1 DPClus

DPClus is a cluster periphery-tracking algorithm proposed to mine dense sub-graphs in interaction networks. DPClus weights all the nodes in its first step. Then it takes the highest weighted node as the initial cluster and extends this cluster by adding nodes from its neighbors. It uses two parameters din and cpin (din is a value of minimum density and cpin is a minimum value for cluster property) to determine whether a neighbor should be added to the cluster.

9.4.2 IPCA

The algorithm IPCA follows the general approach of cluster expanding based on seeded vertices, as what DPClus did. However, the rules of IPCA for expanding clusters and weighting vertices are somewhat different from that of DPclus especially they target a different topological structure for the resulted clusters. In particular, the algorithm DPClus identifies sub-graphs that satisfy a density condition and certain cluster connectivity property, while the algorithm IPCA looks for sub-graph structures that have a small diameter (or a small average vertex distance) and satisfy a different cluster connectivity-density property. Also, the algorithm IPCA computes the vertex weights only once, based on the original input graph. On the other hand, once a new cluster is identified, the algorithm DPClus removes the cluster and re-computes the vertex weights based on the remaining sub-graph.

9.4.3 CoAch

CoAch (Co re-Attachment based method) is a method originally applied to detect protein complexes in PPI networks by considering their inherent organizations. In particular, protein-complex cores, as the "hearts" of the protein complexes, are first detected from each vertex’s neighborhood graphs. CoAch method does provide insights into the inherent modularity and organization of protein complexes. In addition, in terms of prediction accuracy, the CoAch method also outperforms existing computational methods.

9.4.4 Graph Entropy Clustering

Graph Entropy clustering is a method based on the maximum entropy principle. It explores the space of all possible probability distributions of the data to find one that maximizes the entropy subject to extra conditions based on prior information about the clusters. The prior information is based on the assumption that the elements of a cluster are “similar” to each other in accordance with some statistical measure. As a consequence of such a principle, those distributions of high entropy that satisfy the conditions are favored over others. Searching the space to find the optimal distribution of object in the clusters represents a hard combinatorial problem, which disallows the use of traditional optimization techniques. In general, a supervised classification method will outperform a non-supervised one, since in the first case, the elements of the classes are known priori. Graph Entropy clustering method’s effectiveness is comparable to a supervised one.

10 Experimental Results

In this section, we demonstrated our experimental results of identifying communities in two instances of the Les Misérables characters social networks. We are using two different versions of the Les Misérables data set: one is .gml graph data and the other one is plain text data. We compare approaches in both of the experiment instances.
In the plain text data set experiment we compared clique percolation method while k=3 and k=4, CoAch, graph entropy algorithm, DPClus and IPCA. Among those, clique percolation method seems to give better results given selected value of k. IPCA results contains too many overlapping communities, CoAch only manage to find one large community, while our implementation of graph entropy algorithm and DPClus gave some results which are farther from what we expected and what we observed from other method results.

Index	Community found (k=3)	Community found (k=4)
Community 1	11 25 26 27 59 49 56 28 50 52 60 62 63 65 24	59 49 56 60 62 63 65 66
Community 2	3 2 4	11 56 27 50

Table 1: Comparison of clique percolation method (weighted) while k=3 and k=4

Index	Community found
Community 1	11 25 26 49 28 44 42 34 3 72 71 70 69 4

Table 2: Result of CoAch Algorithm (weighted), which only find one community

Index	Community found
1	39 38 37 30 36 35
2	73 72 71 70 69 28
3	77 59 58 60 61 62 63 64 65 66 67
4	24 13 20 21 22 23 19 18 31 17 32
5	25 26 71 70 69 42 43 41
6	55 25 27
7	10 1 3 2 5 4 7 6 9 8
8	30 36 69 34 25 26 27 46 44 45 28 29 3 2 4 13 12 73 72 71 70 11 39 38 15 14 16 33 56 37 50 35 52
10	40 53
11	70 60 76 62 63 64 65 66 67 69 77 59 58 48 49 47 61 75 74

Table 3: Result of graph entropy algorithm (weighted), which is not showing the correct communities

Index	Community found
1	11 26 27 59 58 49 42 56 28 50 52 60 62 63 64 65 66 70 71 69 72
2	11 26 59 58 49 42 76 56 28 60 61 62 63 64 65 66 67 71 69 70 72
3	24 11 26 27 59 25 42 56 28 50 52 65 71 70 49 69 72
4	24 11 26 27 20 21 22 23 19 18 56 28 17 25
5	28 60 61 62 63 64 65 66 67 69 26 49 42 77 76 72 71 70 11 59 58 56
6	11 25 26 27 59 71 49 56 30 28 36 35 50 65 70 24 69 52 72
7	11 25 39 38 59 71 49 28 30 56 37 36 35 50 27 26 65 70 24 69 52
8	11 25 26 27 59 58 49 55 42 56 28 50 52 63 65
9	11 25 26 27 59 71 49 32 56 28 50 52 65 70 24 69 29 72
...	...
31	11 3 2 5 4
32	11 3 2 4 7
33	11 3 2 4 6
34	11 8 3 2 4
35	11 10 3 2 4
36	11 25 26 27 59 71 49 56 28 50 52 12 65 70 24 69 29 72
37	11 25 26 27 15 71 49 56 28 50 52 65 70 24 59 69 29 72
38	11 25 26 27 59 14 49 56 28 50 52 65 70 24 71 69 29 72
39	11 25 26 27 59 71 16 56 28 50 52 65 70 24 49 69 29 72
40	11 25 26 27 59 71 49 33 56 28 50 52 65 70 24 69 29 72

Table 5: Result of IPCA Algorithm (weighted), which found a lot of highly overlapping communities

Index	Community found
1	49 59 63 60 65 56 66 62 64
2	11 27 56
3	26 25 11
4	24 28 11
5	17 18 19 20 21 22 23 24
6	30 35 36 37 38 39 11
7	42 43 ... 69 70 71 76 72 ... 26
8	2 4 3
9	50 52 56
10	29 45 11
11	58 68
12	74 75 49
13	31 32 11 24
14	67 61 59 63 65
15	40 53
16	47 48 49

Table 4: Result of DPClus Algorithm (weighted), which is showing some wrong communities

In the .gml graph data set experiment we compared clique percolation method , maximal cliques method, graph entropy algorithm, Girvan-Newman algorithm, and modularity maximization. Among those, clique percolation method was giving the same results as in implementation in other experiment. Results of our maximum cliques method implementation contains too many overlapping communities (59 in total). Girvan-Newman algorithm and modularity maximization algorithm both gave similar and well-fitting outcomes (3 communities). (The detailed result of this part of experiment is lengthy, so we are not showing it in this report. However they can be accessed in the Jupyter Notebook files in our project repository.)

11 Conclusion

We studied the network community detection problem in this paper first by trying a new network data representation framework, which allows us to identify social cliques, explore and analyze social network in a compressive sensing perspective. Although we encountered great hardness during its implementation and had to switch to other approaches, we still managed to compare several popular community detection methods in our experiment, including some approaches from Bioinformatics field, and demonstrate and evaluated the effectiveness of different community detecting methods. We hope that we can carry on and finish the implementation of the Basis Pursuit approach of clique detection in the future. We also hope that our work could enhance our understanding of social network analysis and compressive sensing, and more tools from an area could also be ported to another and create exciting results.

12 Acknowledgements

We would like to thank Professor Peter Chin for teaching this enlightening course and all his help on this project. We would also like to thank our grader Ken Zhou for the great work and help during this semester. At last we would like to thank all the scholars whose work and publication inspired us and everyone who gave us support on this project.

References

[1] P. J. Bickel and A. Chen. "A nonparametric view of network models and newman-girvan and other modularities". Proceedings of National Academy of Sciences of the United States of America, 106(50):21068–21073, 2009.

[2] S. Jagabathula and D. Shah. "Inferring rankings under constrained sensing". In Neural Information Processing Systems (NIPS), 2008.

[3] Xiaoye Jiang, Yuan Yao, Han Liu, Leonidas Guibas. "Detecting Network Cliques with Radon Basis Pursuit". Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, PMLR 22:565-573, 2012.

[4] http://networkdata.ics.uci.edu/data/lesmis/

[5] http://www.zess.uni-siegen.de/myoffer/files/file$_{55}2170211052012.pdf$

[6] https://sandipanweb.wordpress.com/2017/12/22/some-optimization-implementing-the-orthogonal-matching-pursuit-omp-and-the-basis-pursuit-bp-algorithms-with-octave-matlab/

[7] https://towardsdatascience.com/graphs-paths-bron-kerbosch-maximal-cliques-e6cab843bc2c

[8] http://www2.unb.ca/ ddu/6634/Lecture$_notes/Lect10_community_R.pdf$

[9] https://www.cs.cmu.edu/ ckingsf/bioinfo-lectures/modularity.pdf

[10] https://jeremykun.com/2014/05/19/community-detection-in-graphs-a-casual-tour/

[11] Convex Optimization https://web.stanford.edu/ boyd/cvxbook/bv$_cvxbook.pdf$

[12] "SPGL1: A solver for large-scale sparse reconstruction". https://www.cs.ubc.ca/ mpf/spgl1/.

[13] Wu, Li, Kwoh and Ng. "A core-attachment based method to detect protein complexes in PPI networks". BMC Bioinformatics, 2009.

[14] Onnela, Saramäki, Kertész, Kaski. "Intensity and coherence of motifs in weighted complex networks". Physical Review E. 71, 2005.

[15] https://en.wikipedia.org/wiki/Community$_structure$

[16] M. Altaf-Ul-Amin, Y. Shinbo, K. Mihara, K. Kurokawa, and S. Kanaya. "Development and implementation of an algorithm for detection of protein complexes in large interaction networks". BMC Bioinformatics, 2006.

[17] Li, Chen, Wang, Hu, and Chen. "Modifying the DPClus Algorithm for Identifying Protein Complexes Based on New Topological Structures". BMC Bioinformatics, 2008.

[18] Min Wu, et al. "A core-attachment based method to detect protein complexes in PPI networks" BMC , 2009.

[19] Aldana-Bobadilla, E.; Kuri-Morales, A. "A Clustering Method Based on the Maximum Entropy Principle". Entropy, 2015.