Hardness of Community Detection

Apoorva Bhagwat, advised by Avrim Blum

School of Computer Science, Carnegie Mellon University

Graph clustering is an important problem in many domains. At a high level, given a graph, we want to find sets of related vertices in the graph. This objective can be captured in several ways, so several formulations of the clustering problem exist (such as k-medians). However, until recently, most formulations of clustering did not allow clusters to overlap. Overlapping clusters are of interest in community detection, which motivates a different definition of an (α, β)-cluster [Mishra et al. 2007] in a graph. This document is a report of my research related to (α, β)-clustering. The key result is an NP-hardness result for a variant of this problem.

Additional Key Words and Phrases: Graph clustering, overlapping clusters, (α, β)-clustering, NP-hardness

1. INTRODUCTION

At a high level, graph clustering is the problem of finding sets of 'similar' vertices in a graph. For example, given a graph where vertices represent documents and related documents are connected by edges, we might want to find clusters of documents that address the same topic. Another application is community detection : given a graph where the vertices represent people and edges represent friendships, we might want to search for communities of people.

Several formulations (such as k-medians) of the clustering problem exist, and try to capture the high-level goal stated above. However, a notable shortcoming of most traditional formulations is that they do not allow clusters to overlap. This is fundamentally unnatural in applications such as social network analysis, because in this domain, individuals are expected to be a part of multiple communities at once.

In light of this, [Mishra et al. 2007] introduced the following definition of a cluster in a graph. Given a graph $G = (V, E)$ (where every vertex has a self-loop), a set $S \subseteq V$

is called an (α, β)-cluster if it is internally α-dense and externally β-sparse. In other words, we have the following :

$$\text{For all } u \in S, \; u \text{ has at least } \alpha|S| \text{ neighbors in } S$$

$$\text{For all } v \notin S, \; v \text{ has at most } \beta|S| \text{ neighbors in } S$$

It is not hard to see from this definition that a single vertex can belong to multiple (α, β)-clusters in a graph, so this notion of clustering is well-suited for domains such as community detection in social networks.

2. RELATED WORK

[Mishra et al. 2007] introduced this notion and gave an algorithm that finds all (α, β)-clusters with an additional constraint (the 'ρ-champion constraint') in polynomial time. This algorithm only finds clusters with a ρ-champion, i.e. a vertex is said to 'champion' a cluster S if it has at most $\rho|S|$ outside S.

In a later work, Balcan et al [Balcan et al. 2013] showed that such additional assumptions are necessary for polynomial runtime. The showed instances where the graph has quasi-polynomially many (α, β)-clusters (by picking a random graph from the Erdös-Rényi model). They additionally showed that finding even one approximately large cluster in polynomial time is as hard as the hidden clique problem (for a discussion of the hidden clique problem, see [Alon et al. 1998]), and gave a quasi-polynomial time algorithm to find all such clusters (when α, β are fixed constants).

As far as hardness results are concerned, the existence of this algorithm means that we cannot hope to prove NP-hardness for the (α, β)-clustering problem as formulated here.

3. APPROACHES

In this section, we will discuss several aspects of (α, β)-clustering that I explored over the course of my research.

3.1. Random sparsification of the input graph

Balcan et al's work does not directly address the (α, β)-clustering problem. Their formulation introduces the idea of a (Θ, α, β)-self determined community, which is defined on an 'affinity system' instead of a graph. An affinity system is a set of nodes V where each node has a linear order of preference over all other nodes. For any fixed community size c, we say that u votes for v if v appears in the top Θc of u's votes. Finally, a set S is said to be a (Θ, α, β)-self-determined community if every $u \in S$ is voted for by at least $\alpha|S|$ members in S, and no $v \notin S$ is voted for by more than $\beta|S|$ members of S (with respect to the community size $|S|$).

[Balcan et al. 2013] give an algorithm for this problem for the scenario where Θ, α, β are constants : they find all (Θ, α, β)-self-determined communities in time proportional to $n^{O(1/\alpha)}$. They also show that this exponential dependence on $\frac{1}{\alpha}$ is necessary : there exist instances with $n^{\Omega(1/\alpha)}$ communities.

Given this, a natural approach is to consider conditions under which this problem is tractable even for small (sub-constant) values of α. Certainly it cannot be tractable in all cases due to Balcan et al's construction. One possible way to obtain tractable instances of small α value is to take dense a tractable instance, fix a community size, and randomly delete edges in the corresponding directed graph to obtain a sparse graph.

One approach we considered was the following - for fixed values of Θ, α and β and a fixed community size c, an affinity system can be viewed as a directed graph (which

is a more traditional setting for clustering problems). From such a graph, if we delete edges at random with probability $1 - p$, we hope that the following statements hold :

(a) Every (Θ, α, β)-community (of size c) in the original system is now a $(\Theta, p\alpha, p\beta)$-community with high probability.
(b) If a set S is a $(\Theta, p\alpha, p\beta)$-community in the new system, then it must have been close to being a (Θ, α, β)-community in the original system.

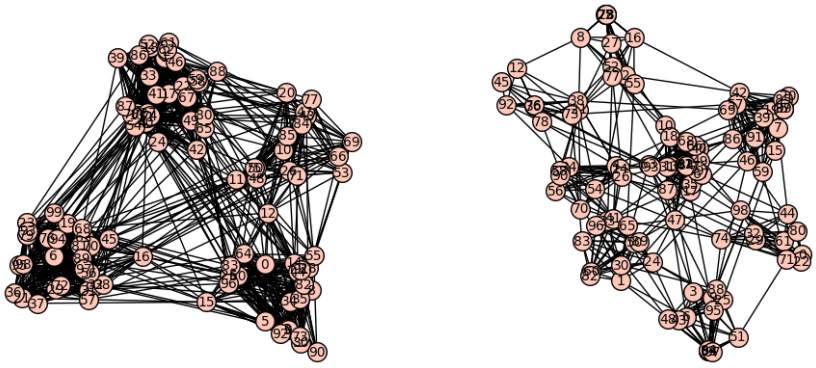
Note that the interesting scenario is when p is a sub-constant parameter (such as $1/\sqrt{n}$, where n is the number of nodes). If we could prove both these statements, we could hope to extend Balcan et al's algorithm to handle small α values. Unfortunately, even though (a) is true, we could not prove (b).

3.2. Evaluating Mishra et al's algorithm on MMSB graphs

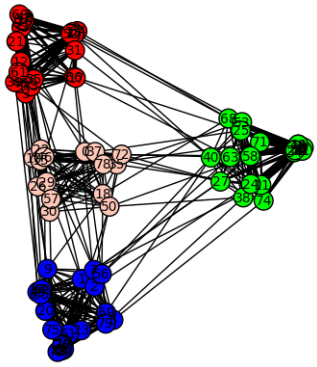
[Airoldi et al. 2008] introduced a model for random graphs known as MMSB (mixed membership stochastic blockmodels). In this model, for each node in the graph, we draw a random membership vector according to a Dirichlet prior. The entries of these vectors are real numbers denoting the affinity of the vertex towards each community. After these draws, the graph can be formed by another random process on these vectors. Specifically, if π_u and π_v are membership vectors of u, v respectively, then there is an edge between u, v with probability $\pi_u^T M \pi_v$, where M is a matrix with diagonal entries being p and off-diagonal entries being q, where $p > q$. In other words, two vertices are more likely to be connected if their randomly drawn membership vectors are affiliated with similar communities. For the sparse regime of the Dirichlet distribution, these graphs naturally contain overlapping communities, so it is interesting to analyze how the (α, β)-clustering algorithm performs on MMSB graphs.

We did not analyze this theoretically, but instead wrote a simulation to cluster MMSB graphs with the (α, β)-clustering algorithm.

MMSB graphs exhibit natural overlapping community structure simply by virtue of the underlying distribution. Here are typical graphs in the model (both with relatively low community overlap) :



However, on running a few simulations, we found that Mishra et al's (α, β)-clustering algorithm did not recover these communities unless the overlap was very small. For instance, here is one graph where the algorithm did quite well (the colors show the clusters marked by the algorithm) :



However, as we increased the overlap parameters, the quality of the clusters found by the algorithm deteriorated quickly. Given this, we did not explore whether we could prove any theorems about the performance of the algorithm.

3.3. Hidden clique hardness for the k-clique densest subgraph problem

For a short period, we explored a problem called the k-clique densest subgraph problem, introduced by [Tsourakakis 2015] as an alternative to other densest subgraph formulations, which are typically NP-hard. Given graph $G = (V, E)$ and a parameter k, this problem asks for a subset $S \subset V$ that induces the largest number of k-cliques (per vertex in S). When k is a constant, this problem can be solved exactly (using flow-based algorithms). [Tsourakakis 2015] proposed an efficient $\frac{1}{k}$-approximation even when k is not a constant. It seems like one should be able to obtain conditional lower bounds for the approximability of this problem. We attempted to show that approximating this problem is as hard as the hidden clique problem, but this direction did not work out.

3.4. Proving NP-hardness of fractional clustering

As noted in the 'related work' section, we cannot hope to easily prove that deciding (for constant α, β) if there is an (α, β)-cluster of a certain size in a graph is NP-hard, because there exists a quasi-polynomial time algorithm that decides this (proving this problem to be NP-hard would falsify the exponential time hypothesis). A natural step is to then relax this problem by dropping the external sparsity condition. Now, we have a problem that is a relaxed version of the clique problem. We show that this problem is still NP-hard.

Definition 3.1. α-fractional clustering.

For any fraction α in $(0, 1]$, let α-fractional clustering be the following problem : given a graph $G = (V, E)$, where each vertex is implicitly considered to be its own neighbor, and a size parameter k, determine if V has a subset S of size k such that every vertex in S has at least αk neighbors in S.

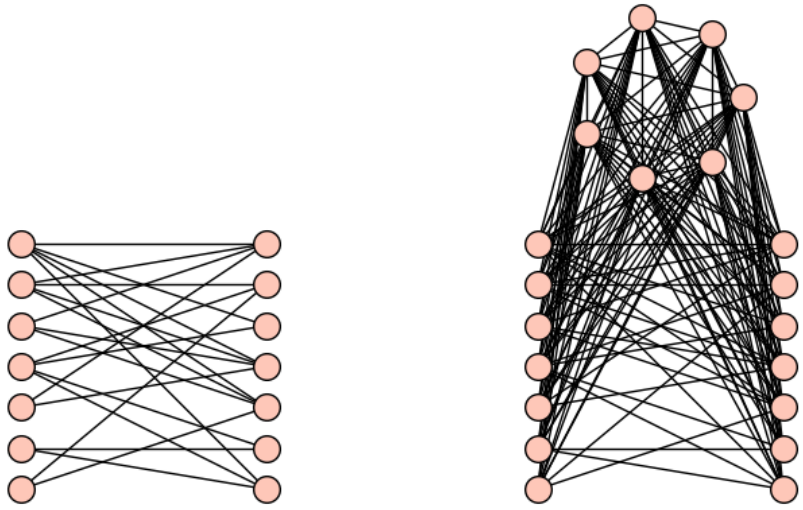
Definition 3.2. Balanced biclique problem.

The balanced biclique problem is as follows : given a bipartite graph $G = (A, B, E)$ and a size parameter j, decide if there is a copy of $K_{j,j}$ (the balanced complete bipartite graph on $2j$ vertices) in G. In other words, decide if there are j vertices in A and j vertices in B that are fully connected to each other.

THEOREM 3.3. *The α-fractional clustering problem is NP-complete if α is a fraction of the form $\frac{p+1}{p+2}$ for some natural number p (including 0).*

PROOF. This problem is easily seen to be in NP. We now reduce the balanced biclique problem (which is known to be NP-complete) to α-fractional clustering (where $\alpha = \frac{p+1}{p+2}$) as follows. Suppose we are given an instance of the balanced biclique problem, i.e. a bipartite graph $G = (A, B, E)$ and a size parameter j. From this, we create an instance of α-fractional clustering. Construct the graph H as follows : we add a clique of size $pj - 1$ (call this C) to G, and connect all vertices in this clique to all vertices in G. Moreover, we set the size parameter k to be $(p + 2)j - 1$. We claim that G has a copy of $K_{j,j}$ iff H has an α-fractional cluster of size k.

Here is an illustration of the construction with $j = 4$, $p = 2$ (i.e. $k = pj - 1 = 7$). The left hand side is a random bipartite graph, and the right hand side is the same graph with a clique fully connected to it.



ONLY IF : Suppose G has a copy of $K_{j,j}$. Then, consider the union of this $K_{j,j}$ and the newly added clique C in H - call this set S. Note that any vertex in C is connected to all vertices in S. Moreover, any vertex in the $K_{j,j}$ subgraph is connected to itself, the other side of the $K_{j,j}$ and all of C, which is a total of $1 + j + (pj - 1) = (p + 1)j$ vertices. The size of S is $(p + 2)j - 1$, so overall, every vertex in S is connected to at least a $\frac{p+1}{p+2}$ fraction of vertices in S.

IF : Now, suppose H contains an α-fractional cluster S of size k. We will prove that $|S \cap A|$ and $|S \cap B|$ must both be at least j. Suppose for a contradiction that (without loss of generality) $|S \cap A| < j$. Then, any vertex in $|S \cap B|$ has at most $j - 1$ neighbors in $|S \cap A|$ and at most $pj - 1$ neighbors in C. Moreover, it's connected to itself, so in total it has as most $(p + 1)j - 1$ neighbors. The size of S is $k = (p + 2)j - 1$, so it's connected to at most a $\frac{(p+1)j-1}{(p+2)j-1}$ fraction of S, which is strictly smaller that $\frac{p+1}{p+2} = \alpha$. Thus, S cannot be an α-fractional cluster.

This proves that $|S \cap A|, |S \cap B| \geq j$. Now, any vertex in $u \in A$ is connected to at least $\frac{p+1}{p+2} \cdot k = \frac{p+1}{p+2} \cdot [(p+2)j - 1] = (p+1)j - \frac{p+1}{p+2}$ neighbors in S. Since $\frac{p+1}{p+2}$ is always a fraction smaller than 1, it must have at least $(p + 1)j$ neighbors in S. Of these, 1 is u itself, and

at most $pj - 1$ neighbors can be in C. Thus, u must have at least j neighbors in B. This argument shows that vertices in $S \cap A$ and $S \cap B$ must be fully connected to each other.

This shows that the balanced biclique problem Karp-reduces to α-fractional clustering, which shows that the latter is NP-hard. □

4. FUTURE DIRECTIONS

Here are the some directions that we either pursued and couldn't finish or that came out of questions explored over the course of our research :

4.1. Improving the NP-hardness reduction

One immediate hope is to generalize the reduction so that it proves NP-hardness for arbitrary values of α. Intuitively, there should be no reason why, say, $\frac{4}{7}$-fractional clustering wouldn't be NP-hard, but it is not obvious how our reduction could be generalized to handle these fractions as well. An immediate next step might be to create a reduction that handles all constant fractions uniformly.

4.2. Random sparsification

This is the direction that we originally started exploring, as noted in the 'approcahes' section. Balcan et al's algorithm (when used on graphs instead of affinity systems) requires clusters to be dense (i.e. the degrees are required to be proportional to cluster the size). However, this is often not true in practice, because we might still be interested in finding large clusters in social networks that are relatively sparse (i.e. communities with low α-values). However, Balcan et al give a lower bound for this case : there exist affinity systems with $n^{\Omega(1/\alpha)}$ many self-determined communities. On the other hand, there construction for this lower bound is adversarial and unlikely to arise in practice. Thus, one might explore the following question : if we assume that

the edges in a sparse social network are (random) realizations of a denser underlying graph, can we hope to recover sparser clusters too?

4.3. Hardness of approximation for min-degree density

We showed that the α-fractional clustering problem is NP-hard when α is a fraction of a certain form. That natural question after this is whether we can efficiently find approximate solutions to this problem (with respect to the cluster density). To put this another way, we can consider the following gap-version of the α-fractional clustering problem (this is a promise problem) :

Fix some constant α. We are given a graph G and a number k. We are promised that one of the following is true :

(1) G contains an α-fractional cluster of size k
(2) Every set of vertices of size k in G has minimum degree at most $(\alpha - \epsilon)k$

Can we distinguish between these two cases? In other words, can we approximate the optimal fractional min-degree of k-node subgraphs up to some constant error ϵ?

4.4. Relationship to the densest k-subgraph problem

The α-fractional clustering problem (as defined in 3.1) appears to be similar to the densest k-subgraph problem [Feige et al. 2001]. The key difference in these two problems is that the densest k-subgraph problem asks for a graph of a specific size that has high *average* degree, whereas α-fractional clustering asks for a graph of a certain size that has high *minimum* degree. If one could find an explicit way to relate these two notions (perhaps a good mapping reduction between these two problems), then one could transfer algorithms and hardness results for densest k-subgraph to α-fractional clustering.

ACKNOWLEDGMENTS

I would like to thank my advisor Prof Avrim Blum for his continuous guidance, and Prof Venkat Guruswami for the core idea of the NP-hardness reduction.

REFERENCES

Edoardo M Airoldi, David M Blei, Stephen E Fienberg, and Eric P Xing. 2008. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research* 9, Sep (2008), 1981–2014.

Noga Alon, Sanjeev Arora, Rajsekar Manokaran, Dana Moshkovitz, and Omri Weinstein. 2011. Inapproximability of densest κ-subgraph from average case hardness. *Unpublished manuscript* 1 (2011).

Noga Alon, Michael Krivelevich, and Benny Sudakov. 1998. Finding a large hidden clique in a random graph. *Random Structures and Algorithms* 13, 3-4 (1998), 457–466.

Maria-Florina Balcan, Christian Borgs, Mark Braverman, Jennifer Chayes, and Shang-Hua Teng. 2013. Finding endogenously formed communities. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 767–783.

Uriel Feige, David Peleg, and Guy Kortsarz. 2001. The dense k-subgraph problem. *Algorithmica* 29, 3 (2001), 410–421.

Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E Tarjan. 2007. Clustering social networks. In *International Workshop on Algorithms and Models for the Web-Graph*. Springer, 56–67.

Charalampos Tsourakakis. 2015. The k-clique densest subgraph problem. In *Proceedings of the 24th international conference on world wide web*. ACM, 1122–1132.

Van Vu. 2014. A simple SVD algorithm for finding hidden partitions. *arXiv preprint arXiv:1404.3918* (2014).