Distance-Based Community Search (Invited Talk Extended Abstract)

Francesco Bonchi[1,2(✉)]

[1] ISI Foundation, Turin, Italy
francesco.bonchi@isi.it
[2] Eurecat, Barcelona, Spain
http://francescobonchi.com/

1 Community Search

Suppose we have identified a set of subjects in a terrorist network suspected of organizing an attack. Which other subjects, likely to be involved, should we keep under control? Similarly, given a set of patients infected with a viral disease, which other people should we monitor? Given a set of companies trading anomalously on the stock market: is there any connection among them that could explain the anomaly? Given a set of proteins of interest, which other proteins participate in pathways with them? Given a set of users in a social network that clicked an ad, to which other users (by the principle of "homophily") should the same ad be shown?

Each of these questions can be modeled as a graph-query problem: given a graph $G = (V, E)$ where (V is a set of vertices representing entities and E is a set of edges modeling the relations among the entities) and given a set of query vertices $Q \subseteq V$, find a subgraph H of G which *"explains"* the connections existing among the vertices in Q, that is to say that H must be connected and contain all query vertices in Q.

Several problems of this type have been studied under different names, e.g., *community search* [3,6,17], *seed set expansion* [2,10], *connectivity subgraphs* [1,7,15,18], just to mention a few. While optimizing for different objective functions, the bulk of this literature aims at finding a "community" around the set of query vertices Q: the (more or less) implicit assumption is that *the vertices in Q belong to the same community*, and a good solution will contain other vertices belonging to the same community of Q. As we showed in our work in [15], when such an assumption is satisfied, these methods return reasonable subgraphs, but when the query vertices belong to different modules of the input graph, these methods tend to return too large a subgraph, often so large as to be meaningless and unusable in real applications. Moreover, the assumption is not so realistic in practice. In fact, we have a set of vertices that we *believe* are of interest for the application at hand and we want to further investigate them: why should we assume they belong to the same community? Moreover, if we have already knowledge of the communities, then why do we need to *"reconstruct"* the community around Q?

B. Catania et al. (Eds.): SOFSEM 2019, LNCS 11376, pp. 21–27, 2019.
https://doi.org/10.1007/978-3-030-10801-4_2

2 The Minimum Wiener Connector

In our work in [15] we take a different approach: instead of trying to "reconstruct" the community around Q we seek a *small* connector, i.e., a connected subgraph of the input graph which contains Q and a small set of *important additional vertices*. These additional vertices could explain the relation among the vertices in Q, or could participate in some function by acting as important links among the vertices in Q. We achieve this by defining a new, *parameter-free* problem where, although the size of the solution connector is left unconstrained, the objective function itself takes care of keeping it small.

Specifically, given a graph $G = (V, E)$ and a set of query vertices $Q \subseteq V$, our problem asks for the connector H^* minimizing the sum of shortest-path distances among all pairs of vertices (i.e., the *Wiener index* [19]) in the solution H^*:

$$H^* = \operatorname*{arg\,min}_{G[S]:Q\subseteq S\subseteq V} \sum_{\{u,v\}\in S} d_{G[S]}(u,v)$$

where $G[S]$ denotes the subgraph induced by a set of nodes S, and $d_{G[S]}(u,v)$ denotes the shortest-path distance between nodes u and v in $G[S]$. We call H^* the *minimum Wiener connector* for query Q.

This is a very natural problem to study: shortest paths define fundamental structural properties of graphs, playing a role in all the basic mechanisms of networks such as their evolution [11] and the formation of communities [8]. The fraction of shortest paths that a vertex takes part in is called its *betweenness centrality* [4], and is a well established measure of the importance of a vertex, i.e., the extent to which an actor has control over information flow. A consequence of our definition of minimum Wiener connector is that our solutions tend to include vertices which hold an important position in the network, i.e., vertices with high betweenness centrality.

Consider social or biological networks with their modular structure [8] (i.e., the existence of communities of vertices densely connected inside, and sparsely connected with the outside). When the query vertices Q belong to the same community, the additional nodes added to Q to form the minimum Wiener connector will tend to belong to the same community. In particular, these will typically be vertices with higher "centrality" than those in Q: these are likely to be influential vertices playing leadership roles in the community. These might be good users for spreading information, or to target for a *viral marketing* campaign [9].

Instead, when the query vertices in Q belong to different communities, the additional vertices added to Q to form the minimum Wiener connector will contain vertices adjacent to edges that "bridge" the different communities. These also have strategic importance: information has to go over these bridges to propagate from a community to others, thus the vertices incident to bridges enjoy a strategically favorable position because they can block information, or access it before other individuals in their community. These vertices are said to span a *"structural hole"* [5]: they are the best candidates to target for blocking the spread of rumors or viral diseases in a social network, or the spread of malware

in a network of computers. In a protein-protein interaction network these vertices can represent proteins that play a key role in linking modules and whose removal can have different phenotypic effects.

In [15] we show that, when the number of query vertices is small, the *minimum Wiener connector* can be found in polynomial time. However, in the general case our problem is **NP**-hard and it has no PTAS unless $\mathbf{P} = \mathbf{NP}$: note that, while the inapproximability result says that the problem cannot be approximated within *every* constant, it leaves open the possibility of approximating it within *some* constant. In fact, our central result is an efficient constant-factor approximation algorithm, which runs in $\widetilde{O}(|Q||E|)$ time. We also devise integer-programming formulations of our problem. We use them to compare our solutions for small graphs with those found using state-of-the art solvers, and show empirically that our solutions are indeed close to optimal. Our experiments confirm that our method produces solutions which are smaller in size, denser, and which include more central nodes than the methods in the literature, regardless of whether the query vertices belong to the same community or not.

3 The Minimum Inefficiency Subgraph

A common aspect of almost all the literature on community search is to require the solution to be a *connected* subgraph. The *requirement of connectedness* is a strongly restrictive one. Consider, for example, a biologist inspecting a set of proteins that she *suspects* could be cooperating in some biomedical setting. It may very well be the case that one of the proteins is not related to the others: in this case, forcing the sought subgraph to connect them all might produce poor quality solutions, while at the same time hiding an otherwise good solution. By relaxing the connectedness condition, the outlier protein can be kept disconnected, thus returning a much better solution to the biologist. Another consequence of the connectedness requirement is that by trying to connect possibly unrelated vertices, the resulting solutions end up being very large.

In our work in [16], we study the *selective connector problem*: given a graph $G = (V, E)$ and a set of query vertices $Q \subseteq V$, find a superset $S \supseteq Q$ of vertices such that its induced subgraph, denoted $G[S]$, has some good "cohesiveness" properties, but is not necessarily connected. Abstractly, we would like our selective connector $G[S]$ to have the following desirable properties:

- **Parsimonious vertex addition.** Vertices should be added to Q to form the solution S, if and only if they help form more "cohesive" subgraphs by better connecting the vertices in Q. Roughly speaking, this ensures that the only vertices added are those which serve to better explain the connection between the elements of Q (or a subset thereof).
- **Outlier tolerance.** If Q contains vertices which are "far" from the rest of Q, those should remain disconnected in the solution S and be considered as outliers. The necessity for this stems from the fact that real-world query-sets are likely to contain some vertices that are erroneously suspected of being related.

- **Multi-community awareness.** If the query vertices Q belong to two or more communities, then the connector should be able to recognize this situation, detect the communities, and refrain from imposing connectedness between them.

A natural way to define the cohesiveness of a subgraph $G[S]$ is to consider the shortest-path distance $d_{G[S]}(u, v)$ between every pair of vertices $u, v \in S$, as done in the previous section. One issue with shortest-path distance is that, when the connectedness requirement is dropped, pairs of vertices can be disconnected, thus yielding an infinite distance. A simple yet elegant workaround to this issue is to use the reciprocal of the shortest-path distance [13]; this has the useful property of handling ∞ neatly (assuming by convention that $\infty^{-1} = 0$). This is the idea at the heart of *network efficiency*, a graph-theoretic notion that was introduced by Latora and Marchiori [12] as a measure of how efficiently a network $G = (V, E)$ can exchange information:

$$\mathcal{E}(G) = \frac{1}{|V|(|V|-1)} \sum_{\substack{u,v \in V \\ u \neq v}} \frac{1}{d_G(u, v)}.$$

Unfortunately, defining the selective connector problem as finding the subgraph $G[S]$ with $S \supseteq Q$ that *maximizes network efficiency* would be meaningless. In fact, the normalization factor $|V|(|V|-1)$ allows vertices totally unrelated to Q to be added to improve the efficiency; clearly violating our driving principle of parsimonious vertex addition. Based on the above arguments, we introduce the measure of the *inefficiency* of a graph $G = (V, E)$, defined as follows:

$$\mathcal{I}(G) = \sum_{\substack{u,v \in V \\ u \neq v}} 1 - \frac{1}{d_G(u, v)}.$$

Hence, we define the selective connector problem as the *parameter-free* problem which requires extracting the subgraph $G[S]$, with $S \supseteq Q$, that *minimizes network inefficiency*. With this definition, each pair of vertices in the subgraph $G[S]$ produces a cost between 0 and 1, which is minimum when the two vertices are neighbors, grows with their distance, and is maximum when the two vertices are not reachable from one another. Parsimony in adding vertices is handled by the sum of costs over all pairs of vertices in the connector; adding one vertex v to a partial solution S incurs $|S|$ more terms in the summation. The inclusion of v is worth the additional cost only if these costs are small and if v helps reduce the distances between vertices in S. Moreover, note that by allowing disconnections in the solution, the second and third design principles above (i.e., outliers and multiple communities) naturally follow from the parsimonious vertex addition.

The *Minimum Inefficiency Subgraph* (**mis**) problem is **NP**-hard, and we prove that it remains hard even if we constrain the input graph G to have a diameter of at most 3. Therefore, we devise an algorithm that is based on first building a complete connector for the query vertices and then *relaxing* the connectedness

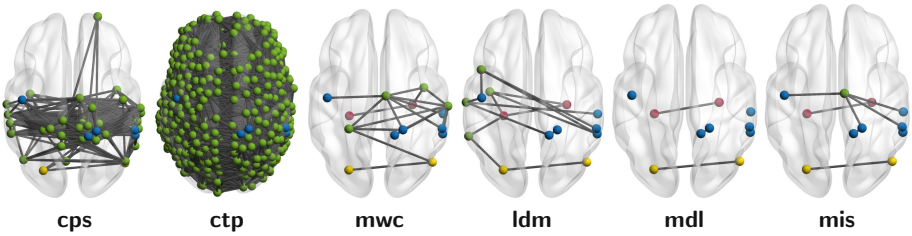

Fig. 1. Comparison between Minimum Inefficiency Subgraph (**mis**) and other notions in the literature on a cortical connectivity network. Query vertices are colored w.r.t. their known functionalities: memory and motor function (blue vertices), emotions (yellow vertices), visual processing (red vertices). The green vertices are the ones added to produce the solution. More details on the case study can be found in [16]. (Color figure online)

requirement by *greedily* removing non-query vertices. Our experiments show that in 99% of problem settings, our greedy relaxing algorithm produces solutions no worse than those produced by an exhaustive search, while at the same time being orders of magnitude more efficient. We empirically confirm that the **mis** is a selective connector: i.e., tolerant to outliers and able to detect multiple communities. Besides, the selective connectors produced by our method are smaller, denser, and include vertices that have higher centrality than the ones produced by the state-of-the-art methods. We also show interesting case studies in a variety of application domains (such as human brain, cancer, food networks, and social networks), confirming the quality of our proposal (Fig. 1).

4 Adaptive Community Search in Dynamic Networks

Although community search has received a great deal of attention in the last few years, most of the literature so far has focused on static networks. However, many of the networks of interest carry time information which can be very important for understanding the dynamics of interactions between the vertices. For instance, interactome, which is the set of molecular interactions in a cell, can be modeled as a network, in which the vertices are proteins and through their connections can perform biological functions. However, these connections are not constantly active, and therefore a dynamic analysis is more appropriate for understanding properly this complex network [14]. In communication networks, for example, the edges represent correspondence between two actors of the network. If a user A communicates with a user B at some time t_0 and later in time, the user B communicates with a user C the flow of information can pass from user A to user C, but not in the opposite direction.

In our ongoing work we are studying the problem of community search in dynamic networks with adaptive query updates. Our objective is to find a temporal connector that includes all the vertices of interest, connecting them with

"temporal paths" that should be seen as paths both in space (i.e., network structure) and in time (i.e., network evolution). Since the network changes constantly in time, we expect that the connectors evolve as well. Therefore, it is natural that the query set is enriched during the evolution, with new vertices, that formed part of the solution of the previous time instances. As long as the added vertices remain related to the initial query set, they are maintained to it. Otherwise, they are removed from the query set. In this way, the connector keeps evolving in time and keeps monitoring the evolution of the interactions among the vertices of interest. We call this problem *temporal adaptive community search*.

Acknowledgements. I wish to thank all the co-authors of the various papers on which this invited talk is built: Natali Ruchansky, Ioanna Tsalouchidou, David García-Soriano, Francesco Gullo, Nicolas Kourtellis, Ricardo Baeza-Yates.

References

1. Akoglu, L., et al.: Mining connection pathways for marked nodes in large graphs. In: SDM (2013)
2. Andersen, R., Lang, K.J.: Communities from seed sets. In: WWW (2006)
3. Barbieri, N., Bonchi, F., Galimberti, E., Gullo, F.: Efficient and effective community search. DAMI **29**(5), 1406–1433 (2015)
4. Bavelas, A.: A mathematical model of group structure. Hum. Organ. **7**, 16–30 (1948)
5. Burt, R.: Structural Holes: The Social Structure of Competition. Harvard University Press (1992)
6. Cui, W., Xiao, Y., Wang, H., Wang, W.: Local search of communities in large graphs. In: SIGMOD (2014)
7. Faloutsos, C., McCurley, K.S., Tomkins, A.: Fast discovery of connection subgraphs. In: KDD (2004)
8. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. PNAS **99**(12), 7821–7826 (2002)
9. Kempe, D., Kleinberg, J.M., Tardos, É.: Maximizing the spread of influence through a social network. In: KDD (2003)
10. Kloumann, I.M., Kleinberg, J.M.: Community membership identification from small seed sets. In: KDD (2014)
11. Kossinets, G., Watts, D.J.: Empirical analysis of an evolving social network. Science **311**(5757), 88–90 (2006)
12. Latora, V., Marchiori, M.: Efficient behavior of small-world networks. Phys. Rev. Lett. **87**(19), 198701 (2001)
13. Marchiori, M., Latora, V.: Harmony in the small-world. Phys. A: Stat. Mech. Appl. **285**(3–4), 539–546 (2000)
14. Przytycka, T., Singh, M., Slonim, D.: Toward the dynamic interactome: it's about time. Brief. Bioinform. **11**(1), 15–29 (2010). https://doi.org/10.1093/bib/bbp057
15. Ruchansky, N., Bonchi, F., García-Soriano, D., Gullo, F., Kourtellis, N.: The minimum wiener connector problem. In: SIGMOD (2015)
16. Ruchansky, N., Bonchi, F., García-Soriano, D., Gullo, F., Kourtellis, N.: To be connected, or not to be connected: that is the minimum inefficiency subgraph problem. In: CIKM (2017)

17. Sozio, M., Gionis, A.: The community-search problem and how to plan a successful cocktail party. In: KDD (2010)
18. Tong, H., Faloutsos, C.: Center-piece subgraphs: problem definition and fast solutions. In: KDD, pp. 404–413 (2006)
19. Wiener, H.: Structural determination of paraffin boiling points. J. Am. Chem. Soc. **69**(1), 17–20 (1947)