Applications of Cohesive Subgraph Detection Algorithms to Analyzing Socio-Technical Networks

Dan Suthers
Dept. of ICS, University of Hawaii
suthers@hawaii.edu

Abstract

Socio-technical networks can be productively modeled at several granularities, including the interaction of actors, how this interaction is mediated by digital artifacts, and sociograms that model direct ties between the actors themselves. Cohesive subgraph detection algorithms (CSDA, a.k.a. "community detection algorithms") are often applied to sociograms, but also have utility in analyzing graphs corresponding to other levels of modeling. This paper illustrates applications of CSDA to graphs modeling interaction and mediated association. It reviews some leading candidate algorithms (particularly InfoMap, link communities, the Louvain method, and weakly connected components, all of which are available in R), and evaluates them with respect to how useful they have been in analyzing a large dataset derived from a network of educators known as Tapped In. This practitioner-oriented evaluation is a complement to more formal benchmark based studies common in the literature.

1. Introduction

Social network analysis has made substantial contributions to the study of networks of actors mediated by technology, which we will term "socio-technical networks". Much of this work applies network analytic methods to networks of actor-actor ties (hence "social network" analysis), but socio-technical systems may be modeled productively at other levels of analysis, using different kinds of networks. A "tie" abstracts away from the actual acts of the actors involved and the relationships between these acts that constitute interaction. It also abstracts away from the means by which the actors interact: the media involved. Our (see acknowledgements) work has explored the utility of alternative levels of representation for understanding a complex socio-technical system, representations that retain some of the information that is lost when abstracting to actor-actor ties. We represent interaction using networks in which nodes represent actions and directed edges represent uptake (ways in which the actions are contingent upon other actions): we call these "uptake graphs" [24]. We represent mediated associations with networks in which nodes include both actors and the media objects through which they interact and directed edges represent read-write relationships weighted by frequency: we call these "associograms" [26].

Although we have applied various network analytic methods to these alternative graph representations, this paper focuses on the application of cohesive subgraph detection algorithms commonly known as "community detection algorithms". We prefer to avoid the connotations of the term "community", except when claiming that we are detecting communities as the layman might understand the term. In addition to finding communities or sub-communities of actors in actor-tie networks, these algorithms can also be used on other types of graphs, such as the two just introduced, to find structures that represent other phenomena of interest. We show that cohesive subgraph detection on associograms finds "communities" of actors *and* artifacts that are more strongly affiliated with each other that with others. This is of interest from an actor-network theory perspective [13] as we can now find clusters of not only human but also media actors that participate in 'assembling' social phenomena. We also evaluate cohesive subgraph detection on uptake (interaction) graphs to find clusters of activity that are more strongly related to each other, and hence (depending on the kind of activity represented and the granularity of analysis) can be interpreted as "sessions" of interaction.

This paper will illustrate applications of cohesive subgraph detection algorithms (henceforth CSDA) to these alternative representations, drawing on a substantial corpus of participants in a socio-technical network known as "Tapped-In". Thus the paper will exemplify the broader applicability of CSDAs as a class to different kinds of questions we might ask about socio-technical systems. Yet there are many different instances of CSDA, which vary in their

URI: http://hdl.handle.net/10125/41412
ISBN: 978-0-9981331-0-2

requirements, strengths and indeed quality. This paper will introduce readers not familiar with the diversity of these algorithms to some of the leading candidates, chosen because they perform well in published benchmark studies, are computationally efficient, and are available in data analytics packages. We emphasize algorithms available in igraph (http://igraph.org/), because this package has a comprehensive collection of cohesive subgraph analysis algorithms and is implemented in C, Python and R. For each of the applications the paper will identify algorithms that meet basic requirements, and then compare them empirically on how well they helped analyze relevant aspects of Tapped-In. The empirical comparisons offered here will not be systematic and exhaustive as one might find in papers based on automated benchmarks (e.g., [10, 17]), but have the advantage that they illustrate issues and practical value relevant to those who want to apply analytics to real world problems.

The remainder of this paper offers a brief survey of cohesive subgraph algorithms, elaborates on the levels of analysis of socio-technical networks discussed above, and introduces the Tapped-In data corpus used for comparing different algorithms. Then there is one section for each level/representation, in which the merits of the candidate algorithms are compared and illustrated using examples on Tapped-In data. Finally the paper summarizes recommendations and directions for further work mandated by the limitations of this study.

2. Cohesive Subgraph Detection

During the past decade there has been considerable development of cohesive subgraph or "community" detection algorithms, resulting in a substantial literature and diversity of algorithms. The field is complicated by the fact that there is no single accepted definition of what constitutes a cohesive subgraph, although there are several widely used metrics. This section identifies several types of CSDA and identifies top contenders in terms of functionality, performance and availability. See [2, 7, 16] for more comprehensive reviews.

Connected components are maximal subgraphs in which every vertex is reachable from every other vertex. In a directed graph, a ***strongly connected component*** is what one obtains by applying this definition respecting arc (directed edge) direction, while a ***weakly connected component*** is what one obtains by ignoring edge direction (treating the graph as undirected). Connected components are straightforward to compute, and algorithms for doing so can be useful, as will be exemplified by one portion of the case study of this paper. However, this definition is too strict for most applications. Consider for example two large groups of people in which everyone within a group interacts with everyone else but there is no interaction between groups, so they form two connected components. If just a single person in one group now interacts with someone in the other group by this definition the two groups collapse into one "community", violating our intuition that members of a community should be more uniformly connected with each other. We want a definition that allows for some ties between the two large clusters but still recognizes that most of the connectivity is within-cluster.

Newman's ***modularity*** metric [15] captures this intuition well. Given a proposed partitioning of a graph, it provides a measure of the extent to which edges in the graph connect vertices in the same partition greater than would be expected at random. (This is equivalent to measuring the extent to which edges connect vertices within the partition more than between partitions.) The assumption that a good cohesive subgraph detection algorithm should partition vertices in a manner achieving high modularity scores has led to algorithms that seek the optimal partitioning under the modularity metric [17].

Brandes et al. [4] have shown that finding the partitioning that gives the maximum modularity score is NP-Hard, which in layman's terms means that any possible algorithm for doing so is expected to take a very long time on large graphs (e.g., days, weeks or even years depending on graph size). Various faster approximation methods have been developed (more than we can review here). An approximation algorithm by Blondel et al. [3] and known as the ***Louvain method*** for the university at which it was developed is widely used. This greedy hierarchical algorithm is fast and produces good quality results in benchmark studies [10], although it is known to suffer from resolution problems [11], meaning that it will lump fine-grained cohesive subgraphs together. This will be another of the algorithms used this paper.

Modularity is based on a static view of the structure of a graph. An alternative view is to see the graph as a structure in which dynamic processes take place. Examples include communication, flow of substances, or travel. Information or substances move throughout the network according to its connectivity. Cohesive subgraphs can then be defined according to how the information or substance of interest would tend to stay within a region of the graph (due to its high internal connectivity) rather than move between regions.

Based on these intuitions (and other mathematical background that cannot be reviewed here), Rosvall and colleagues [20] have developed a class of cohesive

subgraph detection algorithms called ***InfoMap***. InfoMap is described metaphorically (this is not the actual computation done) in terms of a "random walker" that moves to adjacent vertices chosen randomly from among the neighbors of its current location. We can encode the walk by giving each vertex a binary ID. The description of the walk will take a certain amount of space to store. We can compress this description by using a "map": partition the graph and give each partition a code. When a walker enters a partition we give the code, and thereafter only need to give the shorter local label of the vertices within the partition. This reduces the overall length of the coding, but the amount by which the length of the coding is reduced will be determined by how well our partition corresponds to regions in which the random walker spends more 'time' (steps) before exiting. The best partition is one with the shortest description of the walk. We don't actually code walks: this metaphor motivates an information theoretic approach. A *Map Equation L* is defined that predicts the hypothetical encoding length just described, and the algorithm tries to minimize the value of *L* using a hierarchal strategy similar to the Louvain method. InfoMap has also performed very well on benchmarks [10], and does not suffer from the resolution limit of modularity based methods. However, InfoMap is known to have a field of view limit that results in over-partitioning of "long-range" or linear communities into smaller segments [21].

The above algorithms all provide *partitions* of the vertices of a graph. That is, each vertex can be in only one cohesive subgraph or "community". It is easy to imagine situations in which this is not a viable assumption. For example, a person may participate in professional, recreational, and family communities; communities that overlap on this person. The "overlapping community detection" problem is at least as ill-defined as the community detection problem, as there are not universally agreed upon criteria for what constitutes a good solution, but again many algorithms are proposed and several are promising. Here we discuss two that are prominent in the literature.

One of the earliest algorithms, by Palla et al. [18], uses a method known as ***clique percolation***. A *k-clique* is a graph of *k* vertices, all connected to each other. Clique percolation is based on the intuition that members of a community need not be connected to everyone else in the community, but they should be fully connected to some subset (a clique), *and* this subset should be well connected to other cliques. For a given *k*, the algorithm finds all *k*-cliques and then "percolates" them to find adjacent cliques that overlap on *k-1* members. A given vertex can belong to more than one clique, and hence multiple communities. The algorithm can be directed to find clique communities on cliques of sizes 3 on up, giving a hierarchical analysis of community structure. Clique percolation has been applied successfully in various domains [18]. The original version does not work on bipartite graphs, as bipartite graphs have *no* cliques above *k=2*. This problem has been remedied, but the algorithm remains computationally slow [2].

Evans & Lambiotte [5] and Ahn et al. [1] take an alternate approach: compute ***link communities***, partitions of edges or links rather than vertices. It is reasonable to assume that edges can be partitioned into non-overlapping communities, as the placement of each edge is more highly constrained by the relationship between vertices rather than the role of a single vertex. Evans & Lambiotte [5] find link communities by first constructing a *line graph*, a graph in which vertices represent edges in the original graph and edges are placed between vertices in the line graph representing edges in the original graph that are co-incident upon some vertex. Conventional partitioning is then applied to the line graph. Ahn et al. [1] compute link communities directly in the original graph as follows. Two edges are considered for potential *similarity* if they share a "keystone" vertex on one end. The similarity of the two edges is then computed based on the connectivity of the vertices on the other end, specifically the extent to which the two vertices themselves connect to overlapping sets of vertices, using the Jaccard index or the more general Tanimoto coefficient in the case of directed weighted graphs (see supplement to [1]). This similarity metric then drives traditional agglomerative (bottom-up) hierarchical clustering of the edges, and a *partition density metric* (playing a role similar to modularity) is applied to choose the granularity of partition. Regardless of the method used, the resulting link communities then induce overlapping sets on the vertices, as a vertex can participate in more than one community if multiple edges that have been partitioned differently from each other are incident on the vertex. This approach is theoretically appealing because it treats relationships as primary and agents as multidimensional based on the relationships they participate in.

Various computer-generated graphs have been used as "benchmarks" to test CSDAs. For example, Newman & Girvan [17] used randomly generated graphs in which all of the target subgraphs are of the same size and all vertices have the same degree. Since real world networks exhibit power-law distributions of community size and degree, the *LFR benchmark* of Lancichinetti, Fortunato & Radicchi [12] construct benchmark graphs that follow these distributions. Benchmark-based tests provide valuable information about the various proposed algorithms and have been

used by many other authors. The present paper adds a different perspective to this literature, illustrating various specific applications of CSDA and comparing and assessing their practical value in answering relevant questions using data we have already analyzed, enabling comparison of the CSD results to the "ground truth" of our existing understanding.

The algorithms chosen for the following case study were chosen in part based on their prominence in the literature and quality of results in prior benchmark studies. Another criterion was availability of efficient implementations enabling the experiments described below and use by readers of this paper. For these reasons we restrict this study to clustering and cohesive subgraph detection algorithms available in the igraph package (http://igraph.org/), which is implemented in R, Python and C: connected components, Louvain, InfoMap, and link communities.

3. Levels of Analysis in STNs

This work was motivated in part by a view of socio-technical networks as complex systems in which relevant phenomena involve processes at multiple levels of agency (e.g., individual, small group, network). Ties arise from many individual events, and network level phenomena such communities arise from the aggregation of many ties. We developed a framework for multi-level analysis known as ***Traces*** [26, 23]. In this framework, logs of events in the various media are abstracted and merged into a single abstract transcript of events, and this is used to derive a series of representations that support levels of analysis of interaction, affiliations and ties. Three kinds of graphs model interaction. ***Contingency graphs*** record how events such as chatting or posting a message are observably related to prior events by temporal and spatial proximity and by content. There may be multiple contingencies between any pair of events. ***Uptake graphs*** aggregate the multiple contingencies between each pair of events into a single directed edge weighted by a vector of contingency weights, to model how each act may be "taking up" prior acts [24]. ***Session graphs*** are abstractions of uptake graphs: they cluster events into spatiotemporal sessions with uptake relationships between sessions. Relationships between actors and media artifacts are abstracted from interaction graphs to obtain directed (read/write relations) weighted (by number of events) bipartite (actor vs artifact) graphs that we call ***associograms*** [26], Associograms can be folded (factoring out media artifacts) into traditional ***sociograms***. The entire process is automated from the log files to the graphs [23]. Links are retained between these levels of representation, so that (for example) information concerning *how* ties came about is not lost once we reach the summative sociograms.

We explored the utility of cohesive subgraph detection algorithms on some of the graphs just described. CSD was applied to uptake graphs for two analytic tasks: to find subgraphs that correspond to spatiotemporal sessions, and to characterize the internal interactive structure of single sessions. The suitability of several algorithms (Louvain, InfoMap and variations on connected components) for the first analytic task will be discussed in this paper. CSD was also applied to associograms for community detection, with the novelty that the communities are formed by both human and non-human "actants" [13], and we can characterize the mediated nature of the communities by examining the nature of the media artifacts involved [25]. We also discuss four algorithms (Louvain, InfoMap, clique percolation, and link communities) for this application in the present paper, and test two of them. Finally, we applied CSDA to sociograms, but the analytic value of CSD at this level of representation is already well known and will not be discussed further in this paper.

4. Tapped In

We describe the specific setting studied briefly so that the examples and claims about utility of the algorithms will make sense to the reader. We have been studying a data corpus from SRI International's Tapped In® (tappedin.org), an international online network of educators engaged in diverse forms of informal and formal professional development and peer support designed by Mark Schlager and colleagues [6, 22]. Tapped In was motivated by the desire to understand how to initiate and manage large heterogeneous communities of educators, how such communities evolve, and the benefits that participants derive from their involvement (Mark Schlager, personal communication). It included activities that were sponsored by formal organizations mixed with volunteer driven and other unsponsored activities, in both synchronous and asynchronous media, with participants from across all career stages and occupations related to education. Online activity included tenant-sponsored courses, workshops, seminars, mentoring programs, and other collaborative activities; approximately 40-60 public activities per month designed by Tapped In members; and considerable volunteer activity. Data collection capabilities captured all activity, including chat data, discussion board interactions, and file sharing. The trend in current social media is for such data to be treated as a proprietary asset unavailable to external researchers, so the Tapped In data corpus offers a

valuable opportunity for study of how large heterogeneous socio-technical networks evolve. The case studies in the following sections illustrate applications of CSDA that we have found to be useful in understanding Tapped In as a socio-technical system at several levels of analysis in the Traces framework.

5. Finding Sessions

The analytic task discussed in this section is segmenting records of interaction over spans of time in multiple synchronous chat rooms into "sessions". In technical terms, the problem is to parse interaction graphs (such as our uptake graphs) into subgraphs that correspond to spatiotemporal regions in which continuous sessions of interaction separable from other such sessions took place. The internal structure of each such subgraph can then be analyzed to understand the nature of the sessions. A secondary task is to construct an inter-session graph of relationships between sessions. Interaction graphs consist of vertices for events (e.g., chat contributions) and edges for relationships between these events (in our case, contingencies that have been aggregated into uptake). CSDA that partition this graph such that the density of edges within a partition is greater than between partitions (e.g., as measured by modularity or the map equation) offer plausible candidates for sessions. The remaining edges that cross partitions can then be used to construct the inter-session graph.

Some of the synchronous chat activity in Tapped In was scheduled in a public calendar of events. However, we did not want to rely on the calendar, as the actual interactive session might start and end at times different than the advertised times, and we also wanted to also detect opportunistic unscheduled sessions. Participants would typically begin to show up in the designated room shortly before the event began and have informal chat until the start time arrived. There might be some informal discussion at the end, after which participants left and the room was unoccupied. The interaction graphs for scheduled sessions usually have clear gaps between sessions, but in a few rooms (particularly the site-wide "Reception" room), activity is semi-continuous, with indistinct boundaries between what might be called "sessions". We find that each of these kinds of graphs is best handled with a different CSDA, as discussed below. All of the tests were on time spans of data that we had studied extensively previously, so the author was quite familiar with the sessions that should be detected.

Initially the author explored how to construct an uptake graph in a manner such that the Louvain method (as implemented in Gephi, gephi.org) would partition the graph in a manner that corresponded well to the scheduled sessions. Uptake graphs were derived from combinations of contingencies (proximal event, lexical overlap, address and reply by name, same actor: see [23]) for one month of chat data. The tests examined different weightings on these contingencies to see how the weighting affected the outcome. One of these contingencies, *proximal event* (PE 120), is placed between two events occurring within a 120 second time window in the same room, with a decaying weight. The initial finding was that the higher PE was weighted relative to the others, the better the results. Therefore a test was run using *only* the PE contingency, leading to greatly improved results. That is, sessions are best detected when looking for clusters of events close in time and space ignoring other relationships between events. The only loss is that the very few sessions that traveled between rooms were fragmented. There were also a few cases where manually identified sessions were split in half due to a long silence, e.g., when teachers were asked to look at an external web page. Extending the time window of PE to 240 combined some of these sessions. (These numbers are clearly specific to the dataset, as different populations carrying out different tasks would vary in rate of interaction. For example, [19] found a 30 second window to be sufficient for analyzing student IRC chats.) But longer windows incur greater computational cost since there are many more contingencies in the graph.

The author then realized that the computation was nearly equivalent to finding weakly connected components (WCC) on the interaction graph. If only PE contingencies with a 120 second window are used and there is typically a gap between sessions during which a room is empty, there will be no contingencies between sessions, so a simpler and faster algorithm for identifying weakly connected components would be sufficient to identify the majority of sessions correctly. This motivated inclusion of a WCC algorithm in subsequent tests. The tests also included InfoMap, a new algorithm at the time that was receiving a lot of attention in the literature and that performed well on benchmarks.

These subsequent tests compared three algorithms on one day of data, the day that we have studied most extensively, using uptake graphs based on the PE 120 second contingency alone. The Python implementation of igraph (http://igraph.org/python/doc/igraph.Graph-class.html) versions of the algorithms were used: igraph's *community_multilevel* for the Louvain method, *community_infomap* for InfoMap, and *components* with *mode=WEAK* for WCC. InfoMap broke sessions up into roughly equal sized chunks (2628 clusters with modularity 0.918). Recall that InfoMap is known to have a field of view limit that

results in over-partitioning of "long-range" or linear communities: these are precisely the kinds communities we expect in a directed acyclic graph of chronologically oriented phenomena such as chats. Louvain did fairly well, with some small problems that have known explanations based on issues with the data. There were 2502 clusters and 7 inter-session edges, with modularity 0.934. All the intersession edges were between portions of sessions in the same room where there was a bit of a lull. WCC did the best in terms of clean partitioning. There are 2496 clusters with modularity 0.928, and by definition no inter-session links. All the sessions inspected had sensible start, end and contents. Based on these analyses, WCC was chosen as the best method to quickly identify sessions for further analysis.

It is not necessary to construct an interaction graph with PE to identify sessions demarcated by periods of complete inactivity in chat rooms. To avoid the computational complexity of constructing a graph and calling igraph's *components* on it, a queue-based method of assigning sessions according to time gaps was implemented. A global session ID is set to 0 and an event queue is created for each room, limited to a time window (presently 120 seconds). Every time a new event is seen in a room the algorithm first updates the queue to flush events older than 120 seconds. If no events remain in the queue then the global session ID is incremented, and the room and the new event are given this new session ID. If events remain in the queue the present session ID for the room is assigned to the neevent. Results were verified to be equivalent to the graph based WCC method.

This simple scheme works well with our data to single out the majority of sessions, but fails to capture three kinds of situations of interest. (1) Sometimes two sessions may be scheduled back to back in the same room: this would only be detectable by consulting the calendar or (if none exists) attempting to discern changes in participants or interaction structure (perhaps using the Louvain method). (2) There are also a few cases of sessions crossing rooms. In particular, Tapped In sometimes ran orientation sessions for newbies, who might meet in the Reception room and then follow the facilitator to different rooms. The Louvain method run on an interaction graph that includes the "same actor" contingency (which was allowed to cross rooms) was able to detect these moving sessions and cluster the events in different rooms as a single session. (3) Some interaction in Tapped In was opportunistic and ongoing in a manner not clearly demarcated by gaps. The most common example of this was the main Reception room. When users logged in, if they had not changed their preferences otherwise, they were directed to this room. Volunteers were almost always present during waking hours for North America, and spontaneous conversations ensued (e.g., to greet a known person or to direct a newbie to the appropriate room for a scheduled event). Tenant organizations also had their own receptions and public rooms. Uptake graphs of interaction in these kinds of settings have indistinct boundaries between sessions. There might be bursts of activity when participants log in for a scheduled event, but there can also be continuous interactions over periods of time as participants chat while waiting for new arrivals. For these regions of activity, the Louvain method was found to be capable of discerning general periods of discussion (InfoMap being hampered by the field of view limit on these linear graphs).

In summary, we found that CSD can be used to identify spatiotemporal regions of activity or "sessions" in the socio-technical network that form the settings of events by which phenomena seen at more abstracted levels of analysis are constructed (e.g., actor-actor ties, or online communities). Different algorithms may be more suited to this task depending on the desired definition of a setting and how this is reflected in the structural and attribute characteristics of the interaction graph. The Louvain method works well, but for networks that have structured events, simpler algorithms that detect WCC may apply.

6. Segmenting Sessions

The second application is less developed and will be discussed only briefly. Once sessions have been identified in a corpus of data on a socio-technical network, the analyst may wish to characterize the internal structure of sessions, perhaps to automate selecting sessions with certain interactional attributes, or to study a particular session of interest. CSDA may have some utility in segmenting sessions into internal episodes or phases of activity, to the extent that the graph reflects such phases structurally.

We experimented with chat sessions chosen based on prior knowledge of their structure. One session took place during a Tapped In online festival and consisted of two invited talks, so it is dominated first by one and then another speaker. If the Louvain method is run with a slightly coarser resolution (1.5 in Gephi), it identifies three phases that match the actual conversation well: the two speakers' talks and some concluding discussion. Another session is more challenging: an interactive discussion between teachers in a session on mentoring in the schools. The facilitator plays a strong role in helping the discussion progress through a series of essential questions. Here the partitioning appears to have some logic, but we have not yet done formal evaluation.

A limitation of CSDAs for this application is that they are based purely on the structure of the graph. Vital information may be found in the qualitative nature of the uptake relations (in Traces, these are encoded in vectors on the edges but are not available to the CSD algorithms), and other information that is best found by natural language processing. Successful application of CSD would require capturing relevant information more directly in the graph structure.

Another caveat is that if the same CSDA algorithm used to find sessions is applied within a session it may have already found all the structure it has to offer. Therefore, something different may need to be done to find structure within the sessions, whether (for example) changing a resolution parameter or applying a different method altogether.

7. Communities of Actors and Artifacts

At the level of analysis considered in this section, we are concerned with how actors form communities in conjunction with the digital artifacts through which the actors interact. The approach is inspired partly by Latour's [13] presentation of actor-network theory, which emphasizes how human and non-human actors or "actants" together assemble to form what we call social phenomena. It is also inspired by Licoppe & Smoreda's [14] studies of how the nature of the medium of interaction chosen by an actor reflects and reaffirms the nature of that actor's relationships with other actors, and our own observation that the same might be said of how communities (rather than individuals) choose to interact. To understand socio-technical networks at this level of analysis, we use bipartite multimodal directed weighted graphs. They are bipartite because all edges go strictly between actors and artifacts, and multimodal because the artifact nodes can be categorized into different kinds of mediators that they represent, in our case including chat rooms, discussion forums and files. Directed edges (arcs) indicate read/write relations: an arc goes from an actor to an artifact if the actor has read that artifact, and from an artifact to an actor if the actor modified the artifact (the direction indicates a form of dependency). Weights on the arcs indicate the number of events on which the arc is based. We refer to these graphs as ***associograms*** to distinguish them within the more general category of affiliation networks by highlighting how they capture mediated associations [26]. Cohesive subgraph detection of associograms is proposed as a way to identify not only networks of actors that form communities but also the mediated nature of those communities.

An associogram was constructed for our Tapped In data, including vertices for human actors, chat rooms, discussion forums, and file sharing. Write relations were derived from chatting in a room, posting to a forum, or uploading a file (weighted in the former two cases). Weighted read relations were derived from being present in a chat room when someone chats, opening a discussion forum, or downloading a file. Detection of cohesive subgraphs of this graph identifies not only communities of participants, but also whether synchronous chat or asynchronous discussion forums and file sharing mediate these communities. Attributes of both the human actors and the mediating artifacts can be used to identify the real-world attributes of the communities behind the partitions. The graphs we are working with are too large and complex for visualization of the full graph to be helpful: generally interpretation is done by applying a filter to inspect the members of one partition at a time in a tabular display such as Gephi's Data Laboratory.

In an analysis previously published [25], we applied the Louvain method to this graph to identify mediated communities. A contribution of that publication was to show that cohesive subgraphs obtained purely through graph-theoretic characteristics produced partitions that can be interpreted as corresponding to real-world social activities. (The paper also characterized the distribution of communities in Tapped In and their diverse character.) Former Tapped In staff members manually examined the attributes of leading human and digital participants in each partition and identified the known underlying activities. They were able to form meaningful interpretations of all partitions inspected.

For example, the largest partition was immediately identified as consisting of activity around the Tapped In reception room along with popular After School Online (ASO) events. Participants would often enter the reception, interact with volunteer facilitators, and go to the appropriate room for ASO events. The reception, various ASO rooms, and facilitators were in this partition, along with various individuals for whom this was presumably their primary form of participation as they were not classified elsewhere. Another large partition corresponded to an communities of practice program run by a major Midwestern school district, a paid tenant of Tapped In. The artifacts were balanced between chats and discussions. Interestingly, offline organizations do not necessarily map one-to-one to online communities. Another major tenant, a West Coast university with a teacher education program, manifested in distinct partitions corresponding to various online classes and professors working with in-service teachers. Two offline organizations in a Southern state showed up as a single online community because they were both

facilitated by the same person and there was known to be crossover between them.

In addition to reflecting activities that our Tapped In colleagues were aware of, the analysis also identified communities that they were unaware of, but that made sense once they inspected the attributes of the participating actors and media (e.g., room descriptions). The Tapped In colleagues felt that this would have been a valuable tool when they were stewarding the site. For example, if a hithero unknown community was found they could use centrality metrics to identify key actors and provide these persons with support.

The results also showed some limitations of modularity partitioning. The large Reception/ASO partition actually includes distinct sub communities holding ASO events in various rooms for specific topics (e.g., Art, Math, Blogging, Language Arts, Teacher Training, Web tools, Writing), but they were apparently bound together by the Reception common entry point and presence of the same volunteer facilitators in both the reception and ASO events. The facilitators themselves were placed in this partition and so were not present in many other partitions corresponding to events that they were known to participate in under various roles. This obscures important information about who potentially bridges between sub-communities, providing the Tapped In network with overall coherence or a "transcendent community" [8]. We particularly noted this problem for seven facilitators. Although a comparison of the quality of partitioning by the Louvain method to InfoMap would be interesting to see whether InfoMap more appropriately partitions the first mega-community, the need for overlapping community detection is much greater, so we turn instead to this alternative.

The assumption that each human actor or digital actant participates in only one community was tenable when segmenting interaction data into sessions because the whole point was to identify sets of events that can be analyzed for their internal structure. In contrast, with the associogram we are interested in how actors and actants assemble into communities without this assumption. Thus, algorithms for finding potentially overlapping communities in graphs are relevant.

Link communities [1, 5] work efficiently on bipartite graphs and are appropriate for this application, not only because (like other overlapping community algorithms) they allow multiple community membership, but also because they leverage the structure of the associogram in the right way. First, consider the fact that an edge (link) in an associogram connects an actor to an artifact mediating interaction with other actors. Thus, basing communities on edges brings the assemblage of both the actor and artifact into one community, reflecting the fact that the community is constructed by the interaction represented by this assemblage. This addresses the problem we had with strict partitioning where the mediating artifact (e.g., a topic room) might end up in one partition and the relevant actors (e.g., facilitators) in another. Furthermore, Ahn's algorithm [1] brings the right entities together through the computation of the Jaccard similarity metric. Consider two cases: (1) Two edges share an artifact as the keystone vertex: The edges are placed in the same community only if the actor nodes at the other end are similar, i.e., they have a high overlap in the nodes to which *they* connect. This would mean that the two actors sharing the keystone artifact also share other artifacts. Their pattern of mediation is similar, so they are likely to have some kind of mediated association via the keystone artifact and all the other ones. Thus, it is appropriate for these two edges to induce placement of the two actors in the same community as the mediating artifact. (2) Two edges share an actor as the keystone vertex: There are artifacts at the other end, and the two edges are placed in the same community only if the actors that these artifacts connect to have high overlap. In other words, many actors are sharing these artifacts as a means of interaction. So, putting the edges in the same community brings together the actor and two of the artifacts via which the actor interacts with similar sets of other actors into one community. Thus we expect link communities to give better results both by telling us which entities overlap between communities and by producing more accurate communities overall.

A package called ***linkcomm*** is available in R [9], so link communities meet our criteria of CSDA available in common analysis packages. However, this was not available at the time we tested link communities. Instead, a member of our lab, Anthony Christe, obtained the C++ and Python versions of the Ahn algorithm from Yong-Yeol Ahn, and then ported the C++ version to Java in order to implement some improvements we needed for data input and output. Christe's implementation uses the resulting clusters to produce overlapping community graphs suitable for inspection in tools such as Gephi as follows: a vertex for each link community is created, and directed arcs are created from this vertex to all of the vertices of the original graph that are incident on the edges in the given link community. Each vertex is also given an integer count of the number of communities it participates in. Then we can use Gephi's Ego Network filter to inspect each community by choosing the community node as the ego with a distance of 1. Entering Gephi's Data Laboratory with Visible Graph Only checked in Configuration, we have a table of all

members of the community. Vertex attributes from the original graph are transferred to this new graph as well to aid in the interpretation. A known issue with link communities is that it produces large numbers of small communities, some of which consist of single edges. This long tail can be ignored and we focus on the larger vertex communities induced by link communities.

The link community results allow actors and artifacts to be assigned to multiple cohesive subgraphs, enabling us to identify those that bridge communities. As expected, the two most active volunteer facilitators are each found in hundreds of link communities, and the other known facilitators also show up in about a hundred each. The associogram representation also lets us identify media actants that participate in multiple communities. As expected, the Tapped In Reception room is involved in hundreds of communities, reflecting its role for a way station onto other activities, and the ASO room, the location of multiple scheduled events, appears in dozens of communities. Other less expected community bridging actants were also identified, including facilitator offices, two public discussion forums, and a third floor reception room.

Turning now to the clusters themselves, the large cluster that absorbed the Tapped In Reception, facilitators, and ASO has been broken up. The largest cluster includes only the Tapped In Reception, the facilitators, and large numbers of other actors (10234 of the 17619 actors in the data). This appears to be the "chatting in reception community". Now the After School Online activity and other activity associated with facilitator offices has been separated out. The well-defined partitions from the Louvain analysis have corresponding clusters based on link communities. For example, we see clusters for the previously mentioned Midwestern community of practice program, another for the two organizations in a southern state, and again multiple clusters for educators in the Western university running classes. We have not yet undertaken a systematic comparison of the Louvain and link community clusters because interpretation is time consuming, but all indications are that the link community clusters are equally interpretable and find similar important communities.

This work shows that CSD on graphs that represent directed weighted mediated associations between actors and the media via which they interact can find not only communities of actors but also identify the nature of their interaction (e.g., synchronous or asynchronous, and whether affiliated with an organization). Although the Louvain method of CSD by modularity partitioning used in [25] enabled us to identify the presence of communities in a network, accurate identification of individual participation and especially of individuals who bridge communities (quite important for those managing the network) requires overlapping community detection. This section made a theoretical argument that the link community algorithm of Ahn et al. [1] is particularly suited to the associogram structure, and summarized results showing that such an algorithm identifies individuals bridging communities and shows promise for more refined clustering of a giant partition from the Louvain analysis.

8. Conclusions

Socio-technical networks are complex systems, with relevant processes and phenomena occurring at multiple granularities from individual through small group to network levels of agency, and mediated by multiple digital artifacts. Understanding such systems requires more than summary representations such as actor-tie sociograms. We developed a hierarchical analytic framework called Traces to help address this problem. Each level of analysis uses different graph representations to capture phenomena of interest. This paper adds to the literature on understanding socio-technical networks by showing how cohesive subgraph detection on these graphs can expose useful structural information.

The paper also adds to the literature on "community detection" or cohesive subgraph detection algorithms (CSDA). Although it is already well known that CSDA can be applied to graphs representing a variety of phenomena, the present paper describes and illustrates further applications of CSDA, and also provides a practitioner-oriented guide to which CSDA algorithms may be useful for which applications, complementing the more abstract benchmark literature.

The paper outlined several investigations, each of which can be taken further and some of which are in progress for future publication. Other algorithms can be investigated for session detection in graphs that lack discrete boundaries required by the WCC algorithm. The area needing the most work within the Traces framework is the construction of uptake graphs that make the structure of a session structurally apparent. We have yet to evaluate InfoMap in comparison to the Louvain method with ground-truthed community data using the Tapped-In associograms, and a more systematic ground-truthed comparison to overlapping communities induced by link communities is required as well. Rather than evaluating new algorithms only on abstract benchmarks, we advocate considering how they fare on specialized graphs constructed to answer questions of interest to those who study socio-technical networks, and then evaluating the algorithms using real

world data for which some of the painstaking work of constructing "ground truth" has been done.

9. Acknowledgements

Many thanks to Anthony Christe, Kar-Hai Chu, Nathan Dwyer and Devan Rosen for their collaboration on this project; Mark Schlager, Patti Schank and Judi Fusco of SRI for sharing their data and expertise; and the reviewers for excellent suggestions for a longer version of this paper. This work was partially supported by NSF Award 0943147.

10. References

[1] Y.-Y. Ahn, J. P. Bagrow and S. Lehmann, *Link communities reveal multiscale complexity in networks*, Nature, 466 (2010), pp. 761-765.

[2] A.-L. Barabási, *Network Science*, Cambridge University Press, 2016.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, *Fast unfolding of communities in large networks*, Journal of Statistical Mechanics: Theory and Experiment, doi:/10.1088/1742-5468/2008/10/P10008 (2008).

[4] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski and D. Wagner, *On modularity clustering*, IEEE Transactions on Knowledge and Data Engineering, 20 (2008), pp. 172-188.

[5] T. S. Evans and R. Lambiotte, *Line graphs, link partitions, and overlapping communities*, Physical Review E, 80 (2009), pp. 016105-1-8.

[6] U. Farooq, P. Schank, A. Harris, J. Fusco and M. Schlager, *Sustaining a community computing infrastructure for online teacher professional development: A Case Study of Designing Tapped In*, Computer Supported Cooperative Work, 16 (2007), pp. 397-429.

[7] S. Fortunato, *Community detection in graphs*, Physics Reports, 486 (2010), pp. 75-174.

[8] S. Joseph, V. Lid and D. D. Suthers, *Transcendent Communities*, in C. Chinn, G. Erkens and S. Puntambekar, eds., *The Computer Supported Collaborative Learning (CSCL) Conference 2007*, International Society of the Learning Sciences, New Brunswick, 2007, pp. 317-319.

[9] A. T. Kalinka, *The generation, visualization, and analysis of link communities in arbitrary networks with the R package linkcomm*, Cran R Project, 2014.

[10] A. Lancichinetti and S. Fortunato, *Community detection algorithms: A comparative analysis*, Physical Review E, 80 (2009), pp. 056117-1-11.

[11] A. Lancichinetti and S. Fortunato, *Limits of modularity maximization in community detection*, Phys. Rev. E, 84 (2011), pp. 066122.

[12] A. Lancichinetti, S. Fortunato and F. Radicchi, *Benchmark graphs for testing community detection algorithms*, Physical Review E, 78 (2008), pp. 046110-1-5.

[13] B. Latour, *Reassembling the Social: An Introduction to Actor-Network-Theory*, Oxford University Press, New York, 2005.

[14] C. Licoppe and Z. Smoreda, *Are social networks technologically embedded? How networks are changing today with changes in communication technology*, Social Networks, 27 (2005), pp. 317-335.

[15] M. E. J. Newman, *Mixing patterns in networks*, Physical Review E, 67 (2003), pp. 026126.

[16] M. E. J. Newman, *Networks: An Introduction*, Oxford University Press, 2010.

[17] M. E. J. Newman and M. Girvan, *Finding and evaluating community structure in networks*, Physical Review E, 69 (2004), pp. 026113-1-15.

[18] G. Palla, I. Derényi, I. Farkas and T. Vicsek, *Uncovering the overlapping community structure of complex networks in nature and society*, Nature, 435 (2005), pp. 814-818.

[19] D. Rosen and M. Corbit, *Social network analysis in virtual environments*, *Proc. 20th ACM conference on Hypertext and hypermedia (HT '09)*, ACM, New York, NY, 2009, pp. 317-322.

[20] M. Rosvall, D. Axelsson and C. T. Bergstrom, *The map equation*, arXiv:0906.1405v2 [physics.soc-ph] (2009), pp. 1-9.

[21] M. T. Schaub, J.-C. Delvenne, S. N. Yaliraki and M. Barahona, *Markov dynamics as a zooming lens for multiscale community detection: non clique-like communities and the field-of-view limit*, PLOS ONE, 7 (2012), pp. e32210.

[22] M. Schlager, J. Fusco and P. Schank, *Evolution of an Online Education Community of Practice*, in K. Renninger and W. Shumar, eds., *Building Virtual Communities*, Cambridge University Press, Cambridge, 2002, pp. 129-158.

[23] D. D. Suthers, *From micro-contingencies to network-level phenomena: Multilevel analysis of activity and actors in heterogeneous networked learning environments*, *Proc. Fifth International Conference on Learning Analytics and Knowledge. ACM, New York, NY, USA.* , ACM, New York, NY, 2015, pp. 368-377.

[24] D. D. Suthers, N. Dwyer, R. Medina and R. Vatrapu, *A framework for conceptualizing, representing, and analyzing distributed interaction*, International Journal of Computer Supported Collaborative Learning, 5 (2010), pp. 5-42.

[25] D. D. Suthers, J. Fusco, P. Schank, K.-H. Chu and M. Schlager, *Discovery of community structures in a heterogeneous professional online network*, *Proc. Hawaii International Conference on the System Sciences (HICSS-46), January 7-10, 2013, Grand Wailea, Maui, Hawai'i (CD-ROM)*, Institute of Electrical and Electronics Engineers, Inc. (IEEE), New Brunswick, 2013.

[26] D. D. Suthers and D. Rosen, *A unified framework for multi-level analysis of distributed learning* in G. Conole, D. Gašević, P. Long and G. Siemens, eds., *Proceedings of the First International Conference on Learning Analytics & Knowledge, Banff, Alberta, February 27-March 1, 2011*, ACM, New York, NY, 2011, pp. 64-74.