Discovering Communities and Anomalies in Attributed Graphs: Interactive Visual Exploration and Summarization

BRYAN PEROZZI, Stony Brook University, Google Research
LEMAN AKOGLU, Stony Brook University, Carnegie Mellon University

Given a network with node attributes, how can we identify communities and spot anomalies? How can we characterize, describe, or summarize the network in a succinct way? Community extraction requires a measure of quality for connected subgraphs (e.g., social circles). Existing subgraph measures, however, either consider only the connectedness of nodes inside the community and ignore the cross-edges at the boundary (e.g., density) or only quantify the structure of the community and ignore the node attributes (e.g., conductance). In this work, we focus on node-attributed networks and introduce: (1) a *new measure of subgraph quality* for attributed communities called `normality`, (2) a *community extraction* algorithm that uses `normality` to extract communities and a few characterizing attributes per community, and (3) a *summarization and interactive visualization* approach for attributed graph exploration. More specifically, (1) we first introduce a new measure to quantify the normality of an attributed subgraph. Our `normality` measure carefully utilizes structure and attributes together to quantify both the internal consistency and external separability. We then formulate an objective function to automatically infer a few attributes (called the "focus") and respective attribute weights, so as to maximize the `normality` score of a given subgraph. Most notably, unlike many other approaches, our measure allows for many cross-edges as long as they can be "exonerated;" i.e., either (i) are expected under a null graph model, and/or (ii) their boundary nodes do not exhibit the focus attributes. Next, (2) we propose AMEN (for Attributed Mining of Entity Networks), an algorithm that simultaneously discovers the communities and their respective focus in a given graph, with a goal to maximize the total `normality`. Communities for which a focus that yields high `normality` cannot be found are considered low quality or anomalous. Last, (3) we formulate a summarization task with a multi-criteria objective, which selects a subset of the communities that (i) cover the entire graph well, are (ii) high quality and (iii) diverse in their focus attributes. We further design an interactive visualization interface that presents the communities to a user in an interpretable, user-friendly fashion. The user can explore all the communities, analyze various algorithm-generated summaries, as well as devise their own summaries interactively to characterize the network in a succinct way. As the experiments on real-world attributed graphs show, our proposed approaches effectively find anomalous communities and outperform several existing measures and methods, such as conductance, density, OddBall, and SODA. We also conduct extensive user studies to measure the capability and efficiency that our approach provides to the users toward network summarization, exploration, and sensemaking.

CCS Concepts: • **Information systems** → **Data mining**; • **Computing methodologies** → **Anomaly detection**; **Cluster analysis**; • **Human-centered computing** → Interactive systems and tools;

This work is supported by NSF CAREER 1452425, NSF IIS-1017181, an ARO Young Investigator Program grant under Contract no. W911NF-14-1-0029, DARPA Transparent Computing Program under Contract no. FA8650-15-C-7561, an R&D gift from Northrop Grumman Aerospace Systems, and a faculty gift from Facebook.

Authors' addresses: B. Perozzi, Google Research, 111 8th Avenue, New York, NY 10011; email: bryan.perozzi@gmail.com; L. Akoglu, H. John Heinz III College, Carnegie Mellon University, 4800 Forbes Ave, Pittsburgh, PA 15213; email: lakoglu@cs.cmu.edu.

https://doi.org/10.1145/3139241

Additional Key Words and Phrases: Attributed graphs, network measures, community extraction, anomaly mining, ego networks, social circles, visual analytics, summarization, human-in-the-loop, interaction design

ACM Reference format:
Bryan Perozzi and Leman Akoglu. 2018. Discovering Communities and Anomalies in Attributed Graphs: Interactive Visual Exploration and Summarization. *ACM Trans. Knowl. Discov. Data.* 12, 2, Article 24 (January 2018), 40 pages.
https://doi.org/10.1145/3139241

1 INTRODUCTION

Quantifying the quality of connected subgraphs,[1] that for instance corresponds to communities in social networks, has long been a research area of interest, as it finds applications in network community extraction, graph partitioning, and anomaly mining. Most of the existing approaches focus on unattributed or plain graphs and hence only utilize structure. For example, methods that use Newman's modularity [54] as a quality measure aim to partition a graph into communities in which the edges mostly reside among nodes that belong to the same communities. Andersen et al. [5] use the conductance measure to find local subgraphs with small relative cut size. Similarly, METIS [33] and spectral clustering [56] are graph partitioning algorithms that aim to minimize the total cut across the partitions.

While quality measures adopted by these methods concern the cut sizes, several other measures only quantify the connectedness among the nodes inside a subgraph. For example, Charikar [12] aims to find dense subgraphs with large average degree. Tsourakakis et al. [73] use edge density to find tightly connected subgraphs. Similarly, Akoglu et al. [2] adopts edge density to find anomalous ego networks. For a list of other subgraph quality measures in unattributed graphs, we refer to [79].

In contrast to numerous measures that focus only on structural quality, there exist only a few attempts that also utilize attributes. For example, several methods aim to find communities in attributed graphs that not only are dense but also exhibit attribute coherence [4, 22, 27]. Most recently, others also quantify the connectivity at the boundary to find outlier subgraphs or nodes in attributed graphs [29, 61].

In summary, most existing measures either only quantify the structural quality and do not utilize node attributes, or measure only the internal quality and do not consider the boundary. Those that quantify the connectivity at the boundary penalize large number of cross-edges, where methods that optimize such measures seek subgraphs with small cut sizes. As we argue in this article, however, there exist scenarios in which cross-edges are expected and should not be penalized. Moreover, several measures are defined as a global objective for community detection or graph partitioning and are not applicable to individual subgraphs.

In this work, we define a community to be high quality when its nodes are (1) internally well connected and similar to each other on a specific attribute subspace (we call these shared attributes the community *focus*, following the terminology in our recent work [61]), as well as (2) externally well separated from and/or dissimilar to the nodes at their boundary. Based on this definition, we introduce a new measure, called `normality`, to quantify the quality (or normality) of attributed communities, which carefully utilizes both structure and attributes together to quantify their internal consistency within, as well as external separability at the boundary.

Different from many others, our measure allows for many cross-edges *as long as* they can be "exonerated;" i.e., either (i) are expected under a null model, and/or (ii) their boundary nodes do not

[1]Throughout text, we use the words {subgraph, community, cluster, social circle (or circle for short), and neighborhood} interchangeably.

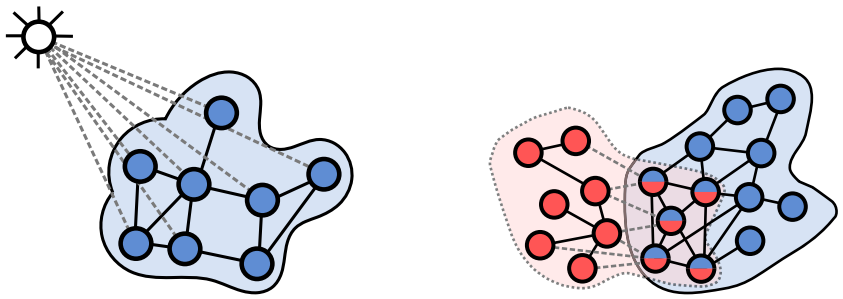


Fig. 1. (Left) A community and a hub node, and (right) two overlapping communities.

exhibit the same (focus) attributes for which the nodes inside the community share similar values. This raises the question of whether communities with many edges at their boundary could be considered as high quality. To make our argument more concrete, we discuss two specific scenarios that could drive the emergence of many cross-edges at the boundary of good communities in real-world attributed graphs.

The first scenario is where the cross-edges are due to hub nodes in real graphs. Consider Figure 1 (left) as an example, which shows a well-structured community and a hub node in the host graph. Notice that the hub node connects to a considerable fraction of the nodes in the community, creating many cross-edges at its boundary. These edges, however, are not surprising. In fact, they are expected since hub nodes by definition connect to a large body of nodes in a graph. While the quality of such a community is diminished based on conductance, our measure simply exonerates those edges as unsurprising and does not penalize the community normality.

Another scenario where many cross-edges at the boundary of good communities arise is when the communities overlap. An example is illustrated in Figure 1 (right) in which two overlapping communities are shown. Good communities are expected to have many internal edges among their nodes, which implies that overlap nodes have many edges to non-overlap nodes in both communities. This in turn creates many cross-edges for both communities at their boundary. These edges, however, are driven by the internal consistency and should not affect their (external) quality. Provided that overlapping communities have sufficiently different focus attributes that make them stand-alone and separable (e.g., football vs. chess group), our normality measure exonerates such cross-edges based on the dissimilarity of boundary nodes to internal nodes. On the other hand, measures that ignore attributes such as modularity and conductance penalize these cross-edges. In fact, such measures are expected to perform poorly for graphs with many overlapping communities such as social networks.

We note that the focus attributes of the communities may be latent and unknown *a priori*, especially in high dimensions. Therefore, we formulate an objective function that utilizes our `normality` measure to automatically *infer* the focus attributes and their respective weights, so as to maximize the normality score of a community. Finally, we propose an algorithm called AMEN that simultaneously finds the communities and their respective focus in a given attributed graph, with a goal to maximize the total `normality`. AMEN employs an alternating optimization scheme in which we alternate between assignment of nodes to communities and inference of respective focus attributes. Communities with low normality scores, i.e., for which a focus that yields high normality cannot be found, are considered low quality or anomalous.

We then shift focus to the sense-making problem of attributed graphs. Visualizing a graph with only a few hundred nodes remains a challenge for today's advanced graph layout algorithms [75], which often produce a "hairball." The problem is even more exacerbated when each node is further

associated with a potentially long list of various attributes. To this end, we utilize our community extraction algorithm to decompose a given attributed graph into its constituent parts. We then design an interactive interface that enables users to explore, characterize, and build alternative summaries of the network.

Overall, the main contributions of our work can be listed as follows:

- *Community normality score for attributed graphs:* We propose a new measure to quantify the quality (`normality`) of the structure (related to topology) as well as the focus (related to attributes) of communities in attributed graphs. Our `normality` score can be seen as a generalization of Newman's modularity [54] to graphs with (multiple) node attributes. Intuitively, it quantifies the extent which a community is "coherent;" i.e., (i) internally consistent and (ii) externally separated from its boundary. We then use this measure for anomalous community mining.
- *Community focus extraction:* Our normality formulation lends itself to automatically identifying the *focus* of a community when it is unknown/latent. The focus extraction corresponds to identifying a few attributes and their associated importance (i.e., weight) that maximize the `normality` score of the community. As such, it identifies the attributes that make a community as coherent, and hence as normal as possible. Those communities for which a proper focus cannot be identified receive low normality scores and are deemed as lesser quality.
- *Community and focus extraction:* We propose the AMEN algorithm that simultaneously finds coherent communities and their focus, in case both are unknown in a given graph, with an objective to maximize the total normality. Our formulation allows for overlapping communities with hard assignments, where a node can belong to multiple communities. Further, both of our community and focus extraction approaches, based on normality maximization, lend themselves to efficient optimization procedures.
- *Community summarization:* We next formulate a summarization task with a multi-criteria objective, which selects a subset of the communities that (i) cover the entire graph well, are (ii) high quality and (iii) diverse in their characteristics. The criteria can be adjusted to obtain various alternative summaries of the communities.
- *Community exploration:* We design an interactive visualization interface that presents the communities to a user in an interpretable, user-friendly fashion. The user can explore all the communities, analyze various algorithm-generated summaries, as well as devise their own summaries interactively to characterize the network in a succinct way.

Experiments on real-world attributed graphs show the effectiveness of our approach. Specifically, we show the utility of our measure in spotting anomalies in attributed graphs, where AMEN outperforms existing approaches including conductance, density, OddBall [2], and SODA [29]. We also conduct extensive user studies to measure the capability and efficiency that our approach provides to the users toward network summarization, exploration, and sensemaking.

A preliminary version of this article appeared at the 2016 SIAM International Conference on Data Mining [60].

The article is organized as follows: We give background and preliminaries that are stepping stones to our `normality` measure in Section 2. Sections 3–6 respectively present (1) the `normality` measure, (2) community and focus extraction, (3) community summarization, and (4) interactive visualization and exploration. We evaluate our proposed methods through extensive experiments and user studies in Section 7, discuss related work in Section 8 and conclude in Section 9.

We publicly share the source code for our community extraction and summarization algorithms (in Matlab), and our interactive visualization toolkit (in Tableau) for reproducibility as well as

academic and non-commercial use at http://bit.ly/2wcttyP (iSCAN for interactive Summarization and Characterization of Attributed Networks).

2 BACKGROUND AND PRELIMINARIES

The community normality measure we propose in this work is inspired by two other graph measures, namely modularity and assortativity. In this section, we first briefly describe those measures, and then highlight the differences of our measure from them.

Modularity. Newman's modularity [54] is a measure of network structure that quantifies the extent which the network divides into modules (a.k.a. clusters, communities, groups, circles, etc.). Networks with high modularity have dense connectivity among the nodes within communities, but sparse connectivity between nodes from different communities. Specifically, modularity is written as

$$M = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j), \tag{1}$$

where c_i denotes the community assignment of node i and $\delta(\cdot)$ is the Kronecker delta function, taking the value 1 if two nodes belong to the same community and 0 otherwise. Node degrees are denoted by k, and $m = |\mathcal{E}|$ is the number of edges in the graph with adjacency matrix A.

The first term of modularity is equal to the fraction of edges that fall between nodes in the same community. That is, $\frac{1}{2m} \sum_{ij} A_{ij} \delta(c_i, c_j) = \frac{1}{m} \sum_{(i,j)\in\mathcal{E}} \delta(c_i, c_j)$. The second term, $\frac{1}{2m} \sum_{ij} \frac{k_i k_j}{2m} \delta(c_i, c_j)$, is the expected fraction of edges between nodes in the same community under the null-model: the random graph with the same degree distribution for which $\frac{k_i k_j}{2m}$ is the probability that two nodes of degrees k_i and k_j are connected to each other. Modularity in Equation (1) is then the difference between the actual and the expected fraction of edges between nodes in the same community. The larger the difference, the more modular the network.

Assortativity. Modularity is defined for non-attributed graphs and solely quantifies structure. Newman also studied attributed networks, in particular their mixing properties, and has proposed a similar formula to measure their what is called *assortativity* (a.k.a. homophily) [52]. Homophily is known as the extent which the same type of nodes connect to one another, especially in social networks [48]. Specifically, for a graph in which every node is associated with a single, *nominal/categorical* attribute (e.g., gender and nationality), its assortativity is written similar to Equation (1) as

$$H^{(\text{nom})} = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(a_i, a_j), \tag{2}$$

where this time a_i depicts the attribute value of node i (e.g., what nationality i belongs to). As such, assortativity is the difference between the actual and the expected fraction of edges between nodes of the same type.

For a perfectly mixed network, in which all edges fall between nodes of the same type, i.e., $\delta(c_i, c_j) = 1$ whenever $A_{ij} = 1$, assortativity is maximum and written as

$$H^{(\text{nom})}_{\max} = 1 - \frac{1}{2m} \sum_{ij} \frac{k_i k_j}{2m} \delta(a_i, a_j) \quad < 1.$$

The normalized value of assortativity is $\frac{H}{H_{\max}}$, which takes the value 1 for a perfectly mixed network, becomes negative for disassortative networks, and 0 for networks for which attributes and structure are uncorrelated.

Scalar assortativity. Equation (2) is for networks with a nominal/categorical attribute a, such as gender, nationality, and the like. For a *numerical/scalar* attribute x, like income, age, and the like, one can derive a corresponding formula using the co-variance of the attribute values among connected nodes. Specifically, $cov(x_i, x_j) = \frac{1}{2m} \sum_{ij} A_{ij}(x_i - \mu)(x_j - \mu)$, where $\mu = \frac{1}{2m} \sum_i k_i x_i$ is the mean value of attribute x over the edge-ends, and k_i denotes the degree of node i (note that the average here is over the edges rather than the nodes). From $cov(x_i, x_j)$ one can derive the assortativity for numerically attributed networks (see [55, p. 228]) as

$$cov(x_i, x_j) = H^{(\text{num})} = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) x_i x_j, \tag{3}$$

where assortativity is positive when x_i, x_j are both small or both large (w.r.t. the mean), and is negative if they vary in opposite directions. Zero assortativity means the attributes of connected nodes are uncorrelated.

Modularity vs. Assortativity. We remark that assortativity is a global measure of a given graph, and does not (at least not directly) utilize any community/clustering structure of the graph. As such, it is more of an attribute based measure than a structure based one. In contrast, modularity solely quantifies the network structure and does not utilize any attribute information.

The specific applications that leverage these two measures have also been different. Modularity is often used as an objective function in community detection and graph partitioning [10, 16, 54, 67]. Assortativity, on the other hand, has often been used in measuring homophily in social science studies, e.g., in analyzing how school children of different races and genders interact [49], and how people from various nationalities are segregated in residential areas [45].

Nevertheless, despite the differences between modularity and assortativity, the two quantities are related. It was observed that assortative networks are likely more modular and tend to break into communities in which "like is connected to like" [51]. In other words, one can think of assortativity as a driving force of modular structure in networks, one that influences the emergence of communities.

In the following section, we introduce a new measure to quantify the quality of a community. Our measure is inspired by assortativity and modularity but exhibits key differences. First, our normality measure utilizes *both structure and attributes* together. Second, our formulation generalizes to graphs with *multiple* node attributes (as opposed to assortativity which is defined only for a single node attribute). In addition, we identify a *subset of attributes* and their relevance weights that maximize the normality score of a community. This helps us find out the *focus* of the community, i.e., specific attributes or common traits around which the community members "click." Finally, our measure allows for *overlapping* communities unlike the disjoint partitions enforced in modularity optimization.

3 COMMUNITY NORMALITY MEASURE FOR ATTRIBUTED GRAPHS

In this section, we describe two intuitive criteria that define a high quality community in an attributed graph, namely, (i) internal consistency (Section 3.1.1), and (ii) external separability (Section 3.1.2). As we go along, we formulate our community normality measure that obeys these criteria.

First, we introduce the notation. Let $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ denote an attributed graph with $|\mathcal{V}| = n$ nodes, $|\mathcal{E}| = m$ edges, and $|\mathcal{A}| = d$ attributes or features, where $\mathcal{A}$ can be large and not restricted to a single attribute unlike in the definition of assortativity. A community of G is defined as a set of nodes C and the edges among them, $(i, j) \in \mathcal{E},\ \{i, j\} \in C$. We denote by B the set of boundary

nodes, which are outside the community but have at least one edge to some node in the community, i.e., $(c, b) \in \mathcal{E}, c \in C, b \in B, C \cap B = \emptyset$.

3.1 Attributed Community Normality

We consider a community to be of high quality based on two criteria: (i) internal consistency and (ii) external separability. Intuitively, *a "good" community has many internal edges among its members where they share a set of attributes with similar values.* In other words, a common set of attributes makes the community members highly similar, around which they "click." In addition, *a good community has either only a few edges at its boundary or many of the cross-edges can be "exonerated."* Simply put, our measure will exonerate the cross-edges at the boundary (a) if they are expected under a random graph model, or (b) if the boundary nodes are dissimilar to the community members with respect to the same attributes that make the community members similar to each other. In the following, we describe these two criteria more formally.

3.1.1 Internal Consistency. To quantify the internal consistency of a community, we propose to generalize the scalar assortativity in Equation (3), which is defined over a graph with a single attribute to be defined for a set of nodes with multiple attributes. We denote by $\mathbf{x_i}$ a *vector* of attributes that node i is associated with. The internal consistency I of a community C is then written as

$$I = cov(\mathbf{x_i}, \mathbf{x_j}) = \sum_{f=1}^{|\mathcal{A}|} \left(\sum_{\substack{i \in C, j \in C, \\ i \neq j}} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \mathbf{x_i}(f) \mathbf{x_j}(f) \right)$$

$$= \sum_{\substack{i \in C, j \in C, \\ i \neq j}} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \sum_{f=1}^{d} \mathbf{x_i}(f) \mathbf{x_j}(f) = \sum_{\substack{i \in C, j \in C, \\ i \neq j}} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \mathbf{x_i}^T \mathbf{x_j}. \quad (4)$$

We remark that $\mathbf{x_i}^T \mathbf{x_j}$ translates to the dot-product similarity between the attribute vectors of community members. However, it treats all of the attributes as equally important in quantifying the similarity among nodes. In general, it is more reasonable to assume that the community members come together around *a few* common attributes (e.g., same school and same hobby). This is expected especially in very high dimensions. We refer to those attributes upon which community members agree, i.e., have similar values, as the *focus* (attributes) of a community.

Therefore, we modify the internal consistency score by introducing a non-negative weight vector $\mathbf{w}$ to compute the weighted node similarities as

$$I = \sum_{\substack{i \in C, j \in C, \\ i \neq j}} \left(A_{ij} - \frac{k_i k_j}{2m} \right) sim_{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_j} | \mathbf{w}), \quad (5)$$

where similarity $sim_{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_j}|\mathbf{w}) = \mathbf{w}^T (\mathbf{x_i} \odot \mathbf{x_j})$, $\odot$ denotes the Hadamard or element-wise product, and $\mathbf{w}$ is a vector with the attribute weights. This corresponds to weighted dot-product similarity, $(\mathbf{w}^{1/2} \odot \mathbf{x_i})^T (\mathbf{w}^{1/2} \odot \mathbf{x_j})$. We expect $\mathbf{w}$ to be sparse, in which only a few attributes corresponding to the community *focus* have large values and zero elsewhere.

Note that each community C is associated with its own weight vector $\mathbf{w_C}$, i.e., focus, which is potentially different across communities. Moreover, the attribute weights are often latent. For defining our normality criteria, we can assume $\mathbf{w}$ is known. Later, in this section, we will show that thanks to our normality formulation, we can infer this weight vector so as to make a given community as internally consistent and externally well-separated, i.e., as normal as possible. Next, we discuss the properties captured by Equation (5).

First, notice that the internal consistency is decreased by missing edges inside a community, as $A_{ij} = 0$ for $(i, j) \notin \mathcal{E}$. Second, the existence of an edge is rewarded as much as the "surprise" of the edge. Recall from Section 2 that $\frac{k_i k_j}{2m}$ denotes the probability that two nodes of degrees k_i and k_j are connected to each other by chance in a random network with the same degree distribution as the original graph. As such, we define the surprise of an edge $(i, j) \in \mathcal{E}$ as $(1 - \frac{k_i k_j}{2m})$. The smaller $\frac{k_i k_j}{2m}$ is for an existing edge inside a community, the more surprising it is and the more it contributes to the quality of the community.

These two properties quantify the *structure* of the community. On the other hand, the similarity function quantifies the *attribute* coherence. Specifically the more similar the community nodes can be made by some choice of $\mathbf{w}$, the higher I becomes. If no such weights can be found, internal consistency reduces even if the community is a complete graph with no missing edges.

Overall, a community with (1) many existing and (2) "surprising" internal edges among its members where (3) (a set of) attributes make them highly similar receives a high internal consistency score.

3.1.2 External Separability. As much as we desire a community to be internally consistent, we consider it to be of high quality if it is also well separated from its boundary. In particular, a well-separated community either has (1) a few cross-edges at its boundary, or (2) many cross-edges that can be "exonerated." A cross-edge $(i, b) \in \mathcal{E}$ $(i \in C, b \in B)$ is exonerated either when it is unsurprising (i.e., expected under the null model) or when internal node i is dissimilar to boundary node b based on the focus attribute weights. The latter criterion ensures that what makes the community members similar to one another does not also make them similar to the boundary nodes, but rather differentiates them. The external separability E of a community C is then written as

$$E = - \sum_{\substack{i \in C, b \in B, \\ (i,b) \in \mathcal{E}}} \left(1 - \min\left(1, \frac{k_i k_b}{2m}\right)\right) \; sim_{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_b} | \mathbf{w}) \; \leq 0. \tag{6}$$

Unlike internal consistency in Equation (5), external separability considers only the boundary edges and quantifies the degree that these cross-edges can be exonerated. As discussed above, cross-edges are exonerated in two possible ways. First, a cross-edge may be unsurprising; in which case the term $(1 - \min(1, \frac{k_i k_b}{2m}))$ becomes small or ideally zero. Second, the boundary node of a cross-edge may not share the same focus attributes with the internal node; in which case the term $sim_{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_b} | \mathbf{w})$ becomes small or ideally zero. The higher the number of cross-edges that can be exonerated, the larger the external separability (note the negative sign) and hence the quality of a community becomes.

Notice that good communities by our definition differ from quasi-cliques for which only internal quality measures, such as edge-density [59, 73] or average degree [12, 25], are defined. Different from those and besides internal consistency, we also quantify the quality of the boundary of a community. Our normality measure is also different from other commonly used measures that quantify the boundary, such as modularity [54] or conductance [5], for which good communities are expected to have only a few cross-edges. In contrast, our formulation allows for *many* cross-edges *as long as* they are unsurprising or if surprising, and can be exonerated by the community *focus*. These advantages arise from the fact that we utilize both structure and attributes in a systematic and intuitive way to define our measure.

3.1.3 Community Normality Measure. Having defined the two criteria for the quality of a community, our normality measure N is written as the sum of the two quantities I and E, where high quality communities are expected to have both high internal consistency and high external

separability.

$$N = I + E = \sum_{\substack{i\in C, j\in C,\\ i\neq j}} \left(A_{ij} - \frac{k_i k_j}{2m}\right) sim_{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_j}) - \sum_{\substack{i\in C, b\in B\\ (i,b)\in \mathcal{E}}} \left(1 - \min\left(1, \frac{k_i k_b}{2m}\right)\right) sim_{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_b}). \quad (7)$$

For a community with the highest normality, all possible internal edges exist and are surprising for which pairwise similarities are high, such that the first term is maximized. Moreover, either it has no cross-edges or the similarity or surprise of existing cross-edges between community and boundary nodes is near zero, such that the second term vanishes. Communities of a graph for which the normality score is negative are anomalous or of lesser quality.

Choice of similarity function. To this end, we considered the node attributes to be scalar variables where $sim_{\mathbf{w}}(\mathbf{x_i}, \mathbf{x_j}|\mathbf{w})$ is the weighted dot-product similarity. If the attributes are categorical (e.g., hair color, and occupation), one can instead use the Kronecker delta function $\delta(\cdot)$ that takes the value 1 if two nodes exhibit the same value for a categorical attribute and 0 otherwise.

The choice of the similarity function is especially important for binary attributes (e.g., likes-cooking and has-car). While those can be thought of as categorical variables taking the values $\{0, 1\}$, using Kronecker δ becomes undesirable for nodes inside a community. The reason is that internal consistency as measured by the δ function is the same both when all the community nodes exhibit a particular binary attribute as it is when all nodes are missing an attribute. However, one may not want to characterize a community based on attributes that its members do not exhibit even if the agreement is large. Therefore, we propose to use dot-product for computing internal consistency and Kronecker δ in computing external separability for binary attributed graphs.

3.2 Community Focus Extraction

Given a community and its set of focus attributes, we can directly use Equation (7) to compute its normality score. However, the focus of a community is often latent and hard to guess without any prior knowledge, especially in high dimensions where the nodes are associated with a long list of attributes. Moreover, even if the community focus is known *a priori*, it is hard to assign weights to those attributes in which case one may reside to the binary relevance weights for the attributes that may be too simplistic.

Therefore, we propose an approach to *infer* the attribute weight vector for a given community, so as to maximize its normality score. In particular, we want to find a $\mathbf{w_C}$ for a community C that makes it have as high normality as possible, such that connected nodes in the community are very similar and connected nodes at the boundary are very dissimilar. As such, we leverage our normality score as an objective function to optimize to infer the best $\mathbf{w_C}$ for community C. As discussed previously, this objective also has the nice property of quantifying structure, by penalizing non-existing in-edges and existing cross-edges. Then, our objective for focus extraction is

$$\max_{\mathbf{w_C}} \; N(C). \quad (8)$$

By reorganizing the terms that do not depend on $\mathbf{w_C}$, we can rewrite (8) (based on Equation (7)) as

$$\max_{\mathbf{w_C}} \quad \mathbf{w_C}^T \cdot \left[\sum_{\substack{i\in C, j\in C,\\ i\neq j}} \left(A_{ij} - \frac{k_i k_j}{2m}\right) sim(\mathbf{x_i}, \mathbf{x_j}) - \sum_{\substack{i\in C, b\in B\\ (i,b)\in \mathcal{E}}} \left(1 - \min\left(1, \frac{k_i k_b}{2m}\right)\right) sim(\mathbf{x_i}, \mathbf{x_b})\right]$$

$$\max_{\mathbf{w_C}} \quad \mathbf{w_C}^T \cdot (\mathbf{x}_I + \mathbf{x}_E), \quad (9)$$

where $\mathbf{x}_I$ and $\mathbf{x}_E$ are vectors that can be directly computed from data. Moreover, the similarity function $sim(\mathbf{x_i}, \mathbf{x_j})$ can be replaced by either $(\mathbf{x_i} \odot \mathbf{x_j})$ or $\delta(\mathbf{x_i}, \mathbf{x_j})$ depending on the type of the node attributes.

3.2.1 Size Invariant Scoring. The normality score in Equation (9) grows in magnitude with the size of the community C being considered. Normalization is desirable then, in order to compare across differences in both community and boundary size.

We note that I is the maximum in the case of a fully connected community the members of which all agree upon the focus attributes. Therefore, $I_{\max} = |C|^2$, where $sim_{\max}(\mathbf{x_i}(f), \mathbf{x_j}(f)) = 1,\ \forall f \in \mathcal{A}$, provided that the attributes $\mathbf{x_i}(f)$ are normalized to $[0, 1]$ for each node i. On the other hand, the minimum is negative, when there exists no internal edges and pairwise similarities are maximum. That is, $I_{\min} = \sum_{\substack{i \in C, j \in C, \\ i \neq j}} -\frac{k_i k_j}{2m}$. To normalize the internal consistency I, we subtract $I_{\min}$ and divide by $I_{\max} - I_{\min}$, which is equivalent to a weighted version of edge density.

To normalize external separability, we derive a measure similar to conductance [5], i.e., ratio of boundary or cut edges to the total volume (sum of the degrees of the community nodes). The difference is that each edge is weighted based on its surprise and the similarity of its end nodes. In particular, we define $\mathbf{x}_{\tilde{I}} = \sum_{\substack{i \in C, j \in C \\ (i,j) \in \mathcal{E}}} (1 - \min(1, \frac{k_i k_j}{2m}))(\mathbf{x_i} \odot \mathbf{x_j})$. Note that similar to E, $\tilde{I}$ considers only the existing edges in the graph. Therefore, $\tilde{I} - E$ can be seen as the total weighted volume of the community.

Overall, we scale our measure as given in Equation (10), where the division of the vectors in the second term is element-wise. This ensures that $\hat{\mathbf{x}}_I(f) \in [0, 1]$ and $\hat{\mathbf{x}}_E(f) \in [-1, 0]$.

$$\hat{N} = {\mathbf{w_C}}^T \cdot (\hat{\mathbf{x}}_I + \hat{\mathbf{x}}_E) = {\mathbf{w_C}}^T \left(\frac{\mathbf{x}_I - I_{\min}}{I_{\max} - I_{\min}} + \frac{\mathbf{x}_E}{\mathbf{x}_{\tilde{I}} - \mathbf{x}_E} \right). \tag{10}$$

3.2.2 Solving the Objective. The normalized objective function can be written as

$$\max_{\mathbf{w_C}} \quad {\mathbf{w_C}}^T \cdot (\hat{\mathbf{x}}_I + \hat{\mathbf{x}}_E) \tag{11}$$

$$\textbf{s.t.} \quad \|\mathbf{w_C}\|_p = 1, \ \ \mathbf{w_C}(f) \geq 0, \ \forall f = 1 \ldots d. \tag{12}$$

Note that we introduce a set of constraints on $\mathbf{w_C}$ to fully formulate the objective. In particular, we require the attribute weights to be non-negative and that $\mathbf{w_C}$ is normalized to its p-norm. These constraints also facilitate the interpretation of the weights. In the following, we let $\mathbf{x} = (\hat{\mathbf{x}}_I + \hat{\mathbf{x}}_E)$, where $\mathbf{x}(f) \in [-1, 1]$.

There are various ways to choose p, yielding different interpretations. If one uses $\|\mathbf{w_C}\|_{p=1}$, a.k.a. the L_1 norm, the solution picks the *single* attribute with the largest $\mathbf{x}$ entry as the focus. That is, $\mathbf{w_C}(f) = 1$, where $\max(\mathbf{x}) = \mathbf{x}(f)$ and 0 otherwise. One can then interpret this as the most important attribute that characterizes the community. Note that $\mathbf{x}$ may contain only negative entries, in which case the largest negative entry is selected. This implies that there exists no attribute that can make the normality positive, and hence such a community is considered anomalous. Note that when $p = 1$, $\hat{N} \in [-1, 1]$.

If one instead uses $\|\mathbf{w_C}\|_{p=2}$, a.k.a. the L_2 norm, then the solution becomes $\mathbf{w_C}(f) = \mathbf{x}(f)$, where $\mathbf{x}(f) > 0$ and 0 otherwise, where $\mathbf{w_C}$ is then unit-normalized. That is, it picks the attributes that correspond to all but the positive entries in $\mathbf{x}$ as the community focus. It is easy to show that if there are *multiple* attributes that can make the normality positive, the L_2 formulation produces an objective value (i.e., normality score ${\mathbf{w_C}}^T\mathbf{x}$) that is higher than that of the L_1 formulation. This agrees with intuition; the more the attributes with positive $\mathbf{x}$ entries, the larger the normality score should grow. On the other hand, if there exists no positive entries in $\mathbf{x}$, we select the single

attribute with the largest negative entry and consider the community as anomalous. As such, $\hat{N} \in [-1, \sum_f \mathbf{w}_C(f)]$ when $p = 2$.

While these are the two most commonly used norms, one can also enforce $\mathbf{w}_C(f) \leq \frac{1}{k}$, for each f, to obtain the largest k entries of $\mathbf{x}$ that can be interpreted as the top-k most relevant attributes for the community (note that those may involve both positive and negative entries). As such, $\mathbf{x}$ provides a systematic and intuitive way to rank the attributes by their relevance to a community where p can be chosen depending on how one aims to interpret the focus.

Finally, notice that the solution to the optimization is quite straightforward where the complexity mainly revolves around computing the $\mathbf{x}$ vector. Specifically, the complexity is $O(|C|^2 d + |\mathcal{E}_B|d)$ for computing $\mathbf{x}$ and $O(d)$ for finding the maximum (L_1) or positive entries (L_2), where $\mathcal{E}_B$ is the number of cross-edges that is upper-bounded by $|C||B|$.

4 DISCOVERING COMMUNITIES AND ANOMALIES

In this work, we focus on assessing the normality of ego networks (or shortly egonets) of attributed graphs, even though we emphasize that entity communities are defined more generally and can consist of any set of connected nodes in such graphs. An egonet is defined for each node (the ego) in the graph and consists of the node, its neighbors, and all the connections among them. That is, an egonet is the induced 1-step community around each node. In the past, we studied egonets to spot structural anomalies in unattributed graphs [2]. The prevalence and availability of meta information enable us to extend our work to attributed graphs and also facilitate the characterization of the anomalies.

In their recent work [24], Gleich et al. showed that egonets themselves are good communities with respect to conductance, where they have low conductance cuts, especially in small sizes. On the other hand, it is unrealistic to assume that all egonets are formed around a single community. Especially in online social networks where people often have a large number of connections, the egonets consist of multiple communities (also referred as communities or circles [46]), e.g., high-school friends, relatives, and so on. Assuming that each egonet is a unit community that could make it difficult to infer a common focus $\mathbf{w}$ that would make all the nodes similar and unfairly reduce the normality score. Therefore, in this section, we aim to address the problem of inferring both the latent communities C (or circles, clusters, etc.) inside an egonet and their corresponding focus $\mathbf{w}_C$ for each $C \in \mathcal{C}$. The normality of an egonet is then defined as a function of the normality scores of its communities.

In our formulation, we denote by G_i the egonet of node i. Our goal is then to find the local circles (i.e., communities) $\mathcal{C} = \{C_1, \ldots, C_R\}$ of G_i as well as their focus vectors $\mathcal{W} = \{\mathbf{w_1}, \ldots, \mathbf{w_R}\}$. Each C_r is assumed to contain the ego i, i.e., $i \in C_r$, $\forall r$. It is often true that a node belongs to multiple circles in (esp. social) networks, where the nodes are "hard-assigned" to the circles. In other words, the circles are allowed to overlap. We remark that although we focus on ego networks G_i in this work, our proceeding formulation and algorithm for circle and focus extraction directly apply to any attributed graph G as a whole or any subgraph of it.

4.1 Problem Formulation

Given an egonet G_i, we aim to extract/infer both its circles $\mathcal{C}$ and their focus vectors $\mathcal{W}$, which are both latent. To solve this problem, we leverage our normality score to build an objective function to be optimized. In particular, our goal is to find circles of the egonet with a maximum total normality score. This construction enforces each circle to be internally consistent and externally well separated from its boundary circles. As such, overlapping circles are expected to have different focus so that the separation can be achieved. This implies that the nodes that belong to multiple circles are expected to have different reasons to take place in each, which is intuitive, e.g., in the social network settings where a person may belong to a family circle, a school-friends circle, a

hobby circle, and so on. The objective formulation is as follows, where we normalize the normality scores as in Equation (10) during the optimization:

$$\max_{C,\mathcal{W}} \quad \frac{1}{R}\sum_{r=1}^{R}\left[\sum_{\substack{i\in C_r, j\in C_r,\\ i\neq j}}\left(A_{ij}-\frac{k_i k_j}{2m}\right) sim(\mathbf{x_i},\mathbf{x_j}|\mathbf{w_r}) - \sum_{\substack{i\in C_r, b\in B_r\\ (i,b)\in\mathcal{E}}}\left(1-\min\left(1,\frac{k_i k_b}{2m}\right)\right) sim(\mathbf{x_i},\mathbf{x_b}|\mathbf{w_r})\right] \tag{13}$$

$$\text{s.t.} \quad \|\mathbf{w_r}\|_p = 1, \mathbf{w_r}(f) \geq 0,\ \forall r = 1\ldots R,\ \forall f = 1\ldots d. \tag{14}$$

Note that for a circle C_r of G_i, the boundary nodes in B_r can include nodes from both outside the egonet G_i and inside G_i but outside C_r. To increase the normality of C_r, both types of edges should be exonerated through $\mathbf{w_r}$.

4.2 AMEN Algorithm

The objective formulation in Equation (13) includes two types of latent variables to be inferred. The first is the assignment of nodes to (potentially multiple) circles and the second is the weight vector, i.e., focus for each circle. Given a circle, we already know how to infer a focus vector to maximize its normality as discussed in Section 3.2. Given the focus vectors, we can also estimate the best assignments of nodes to the circles. This kind of iterative approach suggests an alternating optimization scheme to solve our objective function. The pseudo-code of our algorithm, called AMEN (for Attributed Mining of Entity Networks, in this case Ego Networks), is given in Algorithm 1.

In this alternating scheme, we start with an initial set of circles (line 1). We first assume these to be fixed and infer their attribute weight vectors $\mathbf{w_r}$'s (line 4). Next, we assume that $\mathbf{w_r}$'s are fixed,

ALGORITHM 1: AMEN: Attributed Mining of Entity Networks (Ego Networks)

Input: $G_i = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, node attribute vectors $\mathbf{x}_u\ \forall u \in \mathcal{V}$, number of communities R
Output: Set C of communities C_r's, set of focus vectors $\mathcal{W}$ of $\mathbf{w_r}$'s $\forall r = 1\ldots R$, and normality score $\hat{N}$ of G_i

1: Initialize $C^{(1)}$ carefully with R communities
2: Initialize $C^{(0)}$ randomly, $t \leftarrow 1$
3: **while** $C^{(t)}$ not close to $C^{(t-1)}$ **do**
4: Given $C^{(t)}$, infer focus attribute vector $\mathbf{w_r}^{(t+1)}$ for each $C_r \in C^{(t)}$ based on formulation in Equations (11)–(12)
5: **for each** circle $1\ldots R$ **do**
6: Randomly set $C_r^{(t-1)}$, $t' \leftarrow t$
7: **while** $C_r^{(t')}$ not close to $C_r^{(t'-1)}$ **do**
8: $t' \leftarrow t' + 1$
9: **for each** node $v \in B_r^{(t')}$ **do**
10: **if** inclusion of v increases $\hat{N}(C_r^{(t')}|\mathbf{w_r}^{(t+1)})$ **then** $C_r^{(t')} \leftarrow C_r^{(t')} \cup v$
11: **end for**
12: **for each** node $u \in C_r^{(t')}$ **do**
13: **if** exclusion of u increases $\hat{N}(C_r^{(t')}|\mathbf{w_r}^{(t+1)})$ **then** $C_r^{(t')} \leftarrow C_r^{(t')} \backslash u$
14: **end for**
15: **end while**
16: $C_r^{(t+1)} \leftarrow C_r^{(t')}$
17: **end for**
18: $t \leftarrow t + 1$
19: **end while**
20: **return** $C_r^{(t)}$, $\mathbf{w_r}^{(t)}$; $\forall r$, $\hat{N}(G_i) = \frac{1}{R}\sum_r \hat{N}(C_r^{(t)}|\mathbf{w_r}^{(t)})$

and switch to the reassignment of nodes to the circles. In particular, we try to include the nodes at the boundary of each circle (lines 9–11) and exclude the nodes inside each circle (lines 12–14), the addition or deletion of which improves the normality score, respectively. We continue alternating between these two steps, i.e., inferring focus vectors and reassignments, as long as there exist circles for which the normality can be increased. At convergence, we output the refined circles, their respective focus vectors, and the normality score of the egonet (line 20).

Note that we can refine the node assignments for each circle independent of the others (line 5), since the objective function is modular, i.e., additive over the circles (Equation (13)). As a result, some nodes may end up being included in multiple circles, which drives the emergence of overlapping circles. Moreover, it is possible that some nodes end up unassigned due to exclusions (line 13). Those are the nodes that do not belong to any particular circle in a meaningful way.

4.2.1 Initialization. As AMEN is a heuristic optimization algorithm, it is likely to find locally optimal solutions. To obtain good results, trying multiple and good initializations for the circles is essential. Therefore, instead of starting with a completely random and uninformed initialization, one can use an overlapping community detection algorithm for unattributed graphs (e.g., [6, 36, 76, 80]) to initialize the circles.

A randomized (i.e., non-deterministic) choice of such an algorithm can further help in creating various different initializations. We propose such a randomized procedure in Algorithm 2 to construct an initial subgraphs from a seed node. It computes the delta-improvement in `normality` of adding each node at the boundary of the current subgraph (lines 3–6). The additive nature of `normality` enables incremental and efficient updating when a new node is added to a community without having to recompute it from scratch. If none of them increases the score, the current subgraph is returned (line 7). Otherwise, it picks at random one of the boundary nodes with positive normality improvement that is among the top $(1 - \alpha)$ fraction and adds it to the subgraph (lines 8–12).

The user-defined parameter α that controls the "greediness" vs. "randomness" of the algorithm; $\alpha = 1$ corresponds to the deterministic best-first greedy strategy. We set $\alpha = 0.85$ in our experiments, which allows for finding a different initial subgraph with potentially different *focus attributes* around the seed node every time we run the (non-deterministic) construction scheme. We also pick the R seed nodes at random. The authors in [24] have studied various alternative ways to selecting seeds nodes for local community extraction methods.

4.2.2 Selecting the Number of Circles R. The proposed algorithm takes the number of circles R to be extracted as input parameter. Choosing this parameter is mainly a model selection problem, which we address with well-established principles such as regularization. In particular, we use the Bayesian Information Criterion (BIC) to penalize our objective function against complex models, so that we find sufficiently few circles around each ego. Our regularized objective is written as

$$\max_{\mathcal{C},\mathcal{W}} \; N(G_i) - |\mathcal{W}| \log k_i, \tag{15}$$

where $N(G_i)$ is the normality score of egonet G_i as given in Equation (13) that is the original objective function, k_i is the number of neighbors (i.e., degree) of ego i, and $|\mathcal{W}|$ denotes the total number of non-zero parameters inferred for a particular number of circles R, i.e., $\sum_{r=1}^{R} \|\mathbf{w_r}\|_0 = \sum_r \#(i|\mathbf{w_r}(i) \neq 0)$.

4.2.3 Complexity Analysis. As discussed in Section 3.2.2, inferring a new attribute weight vector $\mathbf{w}$ for a given circle requires $O(|C|^2 d + |C||B|d)$.

Given updated $\mathbf{w}$, recomputing normality takes the same computational steps except one can only focus on the non-zero attribute weights. Let d' denote $\#(i|\mathbf{w}(i) \neq 0)$, where we expect $d' \ll d$

ALGORITHM 2: Construction {*Build Initial Subgraph*}

Input: seed node s, $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, node attribute vectors $\mathbf{x}_u \; \forall u \in \mathcal{V}$, α
Output: initial subgraph g

```
g = s
while true do
    B := boundary nodes of g
    for each b ∈ B do
        ΔN_b := N(g ∪ b) − N(g)
    end for
    if ΔN_b ≤ 0, ∀b ∈ B then return g
    maxΔ := maximum ΔN_b
    minΔ := minimum positive ΔN_b
    B_cand := boundary nodes for which:
            ΔN_b ≥ minΔ + α ∗ (maxΔ − minΔ)
    pick v ∈ B_cand at random
    g := g ∪ v
end while
```

for large d. As such, recomputing normality is $O(|C|^2 d' + |C||B|d')$. Updating the normality score when a new node v is to be added to a circle (assuming $\mathbf{w}$ is fixed) requires $2|C|d' + k_v d'$, where k_v denotes v's degree in the *original* graph (recall that boundary nodes of a circle can be outside the egonet). Specifically, the edges from v to circle nodes are no longer cross-edges and the edges from v to non-circle nodes become cross-edges, which require k_v operations to update external separability E. The complexity to delete a node u is the same and can be written as $O(|C|d' + k_u d')$. If we were to only add nodes to our circles, the number of additions would be bounded by the number of nodes $n = |\mathcal{V}|$ in the egonet. However, as we allow for deletion as well, the number of add/delete operations can be larger. Assuming it is a constant multiple of n, we can write the complexity of refining a circle as $O(nd'(|C| + k_{\max}))$.

Overall, the complexity of AMEN per community per iteration becomes $O(\max[(|C|^2 + |C||B|)d,\ (n|C| + nk_{\max})d'])$. This is multiplied by the number of circles R and the total number of iterations t, both of which we expect to be relatively small constants.

5 SUMMARIZING COMMUNITIES

From exploratory perspective, it is critical to obtain a few, representative communities to characterize an input network such that a user can visualize them for sensemaking. Representative, however, is hard to define and may depend on context. It has been argued in various works [8, 57, 65] that clustering as an exploratory task should be able to provide *alternative views* of the input data and let the user interact with and explore these different views.

To this end, we first extract a large number of various communities from an input graph and then formulate a summarization scheme that identifies only a small number of communities that well represent the input network. In this section, we represent each community to "focus" on one attribute only for simplicity. In other words, we assume each circle forms around a *single* subject (e.g., high school friends). Recall from Section 3.2.2 that the solution is then the attribute that corresponds to the index with the largest value in vector $\mathbf{x}$.

For our running examples in this and the next section, we will use an anonymized Facebook college network, one of 100 such networks first studied in [72]. These networks include only

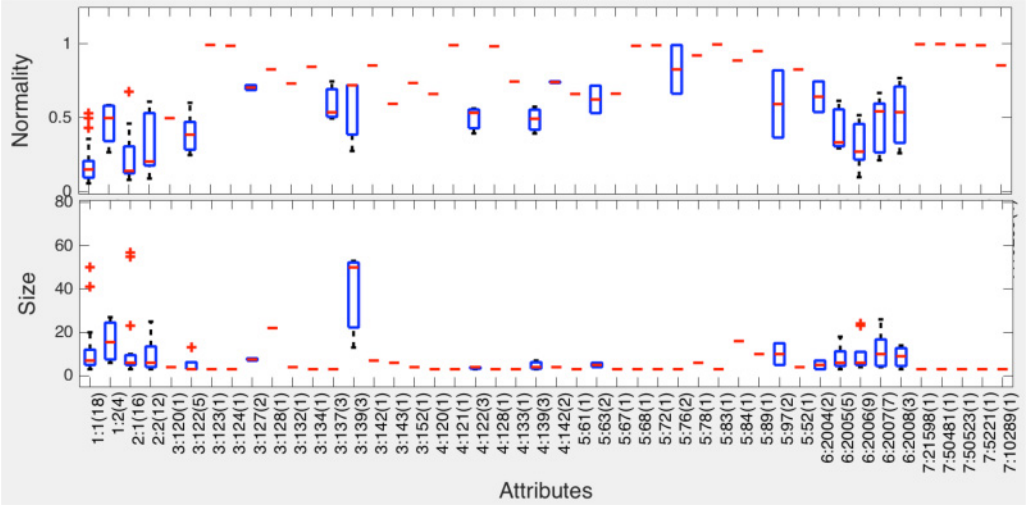


Fig. 2. Normality and size distribution (in boxplots) of communities per specific attribute:value focus.

the intra-school friendship links. The school networks are attributed, in which nodes exhibit seven categorical attributes: `student/faculty flag`, `gender`, `major`, `2nd major/minor (if applicable)`, `dorm`, `graduation year`, `high school`.

As an example, we take a student with 220 friends and consider its egonetwork (all friends and their 6,215 friendship links). We extract R =125 communities from this egonetwork, with average size 9.42 and average normality 0.48. One way to summarize those communities is to look at their size, normality, and focus attribute distribution. To this end, we show in Figure 2 the normality and size for communities of each distinct focus. Labels on the x-axis are of the form `attribute index:attribute value (number of communities with this <attribute:value> focus)`. For example, the algorithm finds 3 communities with attribute 3 (`major`) and value 139 (anonymized). While one can spot various large communities around certain focus attributes, it is hard to quickly describe the make-up of this egonetwork. When we filter out those with size and normality below the average, we obtain 25 communities. However, three issues remain: (1) filtering based on average is arbitrary, (2) the summary is not succinct as there are still many communities above the average, and (3) the distributional summary does not reflect the extent of overlap between the communities.

5.1 Formulation

To address these issues, we formulate a summarization task with the objective of selecting a user-specified number of communities that (1) are high quality (in our case, high normality), (2) cover the input graph well (such that most nodes are represented in the summary), and (3) are diverse in their focus (cover the attribute space well).

More formally, let $\mathcal{C} = \{C_1, \ldots, C_R\}$ denote the set of communities returned by Algorithm 1, and K be the number of communities to be selected for the summary S. Let $N(C_i)$ and $\mathcal{A}(C_i)$, $i = \{1, \ldots, R\}$, respectively, denote the `normality` and the *focus* attribute index for each community in $\mathcal{C}$.

For a subset of communities $S = \{C_{i_1}, \ldots, C_{i_K}\}$, the coverage is defined as the number of unique nodes in the union of communities in S divided by the total number of nodes in input graph G, i.e., $coverage(C_{i_1}, \ldots, C_{i_K}) = \frac{|\cup_{k=1,\ldots,K} C_{i_k}|}{n}$. Similarly, diversity of a subset of communities is the

number of unique attributes they focus on divided by the total number of attributes in G, i.e., $diversity(C_{i_1}, \ldots, C_{i_K}) = \frac{|\bigcup_{k=1,\ldots,K} \mathcal{A}(C_{i_k})|}{d}$.

Our goal is then to identify K out of R communities such that a weighted combination of (1) average normality, (2) coverage, and (3) diversity is maximized:

$$\max_{\substack{S \subseteq C \\ |S|=K}} f(S) = \alpha\ avgNormality(S) + \beta\ coverage(S) + (1-\alpha-\beta)\ diversity(S)$$
$$= \alpha\ \frac{\sum_{C \in S} N(C)}{K} + \beta\ \frac{|\bigcup_{C \in S} C|}{n} + (1-\alpha-\beta)\ \frac{|\bigcup_{C \in S} \mathcal{A}(C)|}{d}$$

Note that all three quantities—average normality, coverage, and diversity—are in the same scale and take values in $[0, 1]$. The respective weights are user-specified and sum to 1, where $0 \leq \alpha, \beta \leq 1$. The user can adjust those weights to put more or less importance on quality, coverage or attribute diversity in the summary, for which we build an interactive interface (to be described in the next section).

5.2 Subset Selection Algorithm

The summarization problem as described above is essentially a subset selection problem for function maximization. In general, it is a hard combinatorial problem as there is a large number of possible subsets even with the cardinality constraint, i.e. when subset size is fixed (in our case, to K).

Fortunately, our objective function $f(\cdot)$ exhibits three key properties that enable us to use a greedy selection algorithm with an approximation guarantee. In particular, provided K, n, d (denominators) are fixed, it is easy to show that our set function $f : 2^C \rightarrow \mathbb{R}_+$ is

(i) *non-negative*; since all three quantities of interest take values in $[0, 1]$ and the respective weights are non-negative and sum to 1.

(ii) *monotonic*; since for every $A \subseteq B \subseteq C$, $f(A) \leq f(B)$. That is, adding more communities to a set cannot decrease the numerators, i.e., total normality, number of covered nodes, and number of covered attributes.

(iii) and *submodular*; since for every $A \subseteq B \subseteq C$ and $C \in C \backslash B$, $f(A \cup \{C\}) - f(A) \geq f(B \cup \{C\}) - f(B)$. That is, adding a community C to a smaller set can increase the function value at least as much as adding it to its superset. Specifically, $C \in C \backslash B$ would increase the total normality equally for A and B due to the additive definition, but can increase the node and attribute coverage more for A than B since B already covers at least the same nodes and attributes as A.

The greedy algorithm by Nemhauser et al. [50] starts with the empty set S_0, and in iteration k, adds the element (in our case, the community) that maximizes the incremental improvement in function value $\Delta_f(C|S_{k-1}) = f(S_{k-1} \cup \{C\}) - f(S_{k-1})$:

$$S_k = S_{k-1} \cup \left\{ \arg \max_{C \in C \backslash S_{k-1}} \Delta_f(C|S_{k-1}) \right\}.$$

Based on their proof, one can show that when $k = K$,

$$f(S_K) \geq \left(1 - \frac{1}{e}\right) \max_{|S| \leq K} f(S).$$

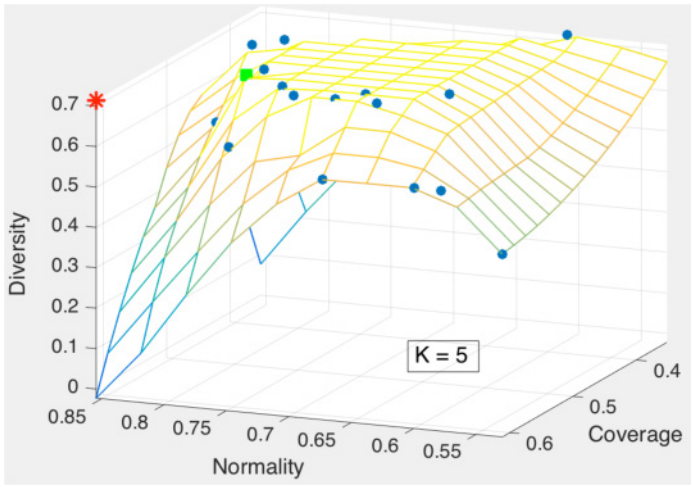


Fig. 3. Diversity, normality, and coverage obtained by our summarization for various $(\alpha, \beta, 1-\alpha-\beta)$ weights for $K = 5$ communities. (Red) star: highest values across all combinations, (green) square: result that is closest to red star.

In other words, the simple greedy search heuristic achieves at least 63% of *optimum* set's objective value.

Figure 3 illustrates the results achieved by our subset selection algorithm on one of the egonets for $K = 5$. The (blue) circles show avg. normality, coverage, and diversity values for various $(\alpha, \beta, 1-\alpha-\beta)$ triples, and the (red) star at the corner depicts the highest values across all combinations. Results around the "knee" of the surface formed by various parameter combinations provide a good tradeoff between the quantities of interest, e.g., the (green) square.

We also run our summarization/subset selection algorithm on the 125 communities extracted from the egonet of the student with 220 friends. For ease of 2-D illustration, we set the weight on diversity to 0, in other words $\alpha + \beta = 1$. Figure 4 shows the node coverage vs. total normality[2] for various $K = \{5, 10, \ldots, 30\}$, where symbols on each curve depict the values for varying (α, β) pairs.

Notice the "knee" in all the curves where coverage can only be slightly increased for increasing β but normality decays sharply as α decreases. As such, points right around the "knee" provide a good tradeoff between coverage and normality. For $K = 5$, the summary at the "knee" corresponds to $\alpha = \beta = 0.5$ and includes the following communities as shown in Table 1.

This summary is easy to comprehend by humans and represents more than 66% of the nodes in the egonetwork. Increasing K does not improve the coverage of the summary without compromising the average normality considerably.

6 INTERACTIVE VISUALIZATION AND HUMAN-IN-THE-LOOP SUMMARIZATION

As a last component, we design a new graphical interface (GI)[3] for users to visualize the communities (Section 4), analyze algorithmic summaries for various K, α, and β (Section 5), as well as build their own summaries by interactively exploring and selecting communities.

[2]In subset selection we use the average normality, however, we plot the (unnormalized) total normality to avoid overplotting curves for different K.

[3]Designed and developed in Tableau, www.tableau.com.

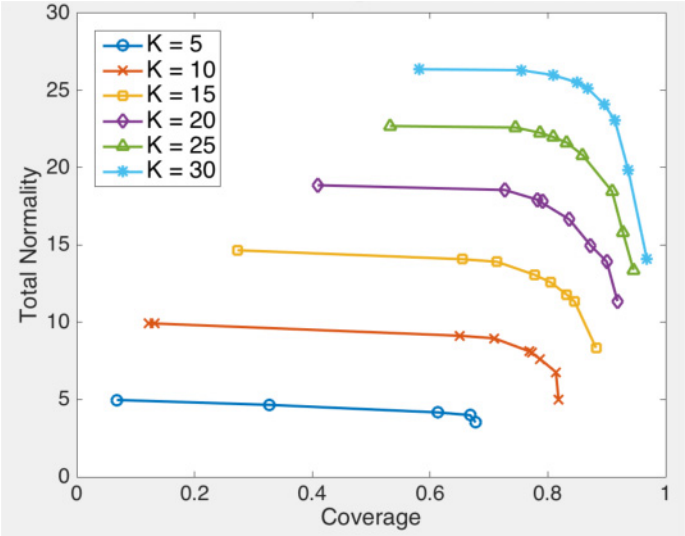


Fig. 4. Total normality vs. coverage for various K (different curves) and varying (α, β) weights for normality and coverage (corresponding to symbols on each curve) as obtained by our subset selection algorithm.

Table 1. Summary of Communities at "Knee" ($\alpha = \beta = 0.5$) in Curve $K = 5$

	Size $\lvert C \rvert$	Normality $N(C)$	Focus attribute $\mathcal{A}(C)$
1	10	0.9510	`dorm`
2	15	0.8207	`dorm`
3	22	0.8275	`major`
4	53	0.7200	`major`
5	57	0.6754	`gender`

Before we dive into the details of the inner-workings, we illustrate the end-product in Figure 5, which shows the interactive exploration and visualization interface presented to a user. It contains three main panels, the filtering panel (left), the community exploration panel (middle), and algorithmic-summary panel (right).

The middle panel is the primary *community exploration view*, which shows all the social circles of an input egonetwork identified by our community extraction algorithm. In this view, social circles are shown with circles, with size proportional to the number of nodes they contain. Color indicates the primary attribute that characterizes a circle (e.g., major), i.e., the *focus attribute*. Circles are positioned in 2-d such that the more they overlap, the closer they are placed to each other. This helps the user to quickly identify overlapping/redundant ones, as well as observe the distribution of social circles by size and characterizing attributes. Hovering over each circle displays (on-demand) its size, list of members, and its `normality`.

The left panel is for *filtering*, where the user can selectively view in the middle panel only the circles of certain focus attribute(s), size(s), and those within certain `normality`.

The right and the final panel is the *summary view*, which displays K circles as selected by our summarization algorithm such that a weighted combination of circle quality, network coverage, and attribute diversity is maximized. K and the respective weights for those quantities are input by

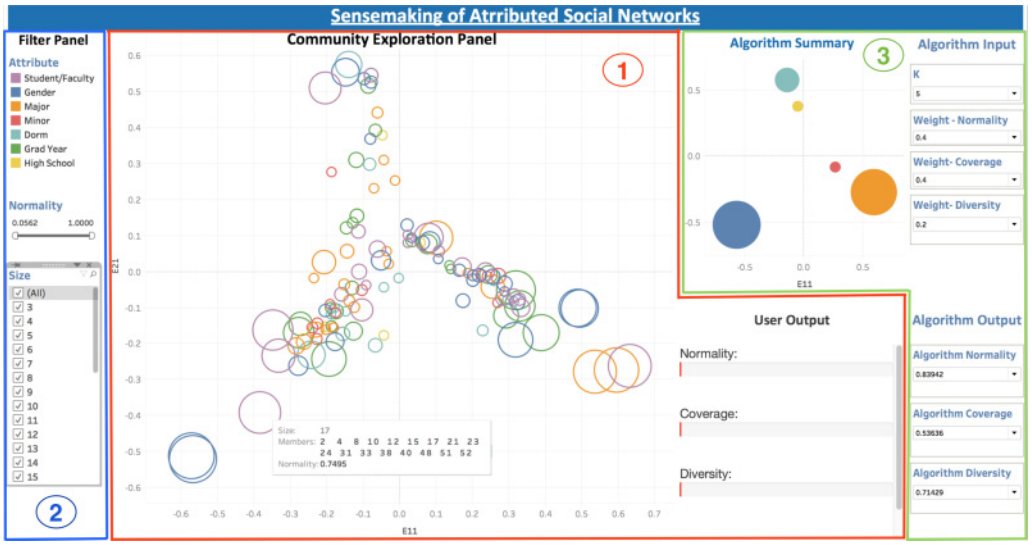


Fig. 5. Main interactive visualization interface with three panels: (middle) community exploration, (left) filtering, and (right) algorithmic summary.

the user (top right). Average normality, egonet coverage, and attribute diversity of the summary is displayed (bottom right). The user can also devise their own summary by selecting (through a click) the circles they would like to include in the summary, and can use the algorithmic summary for guidance. This enables users to explore and build alternative summaries.

We describe our design ideas for these three main components of our GI in the following subsections.

6.1 Exploratory and Interactive Summarization

Our first challenge is to visualize all the extracted communities in a fashion that can help the user to explore them and build their own summary by selecting a representative subset effectively and efficiently. For summarization purposes, the visualization should help the user quickly grasp the size, normality, and the focus attribute of each community. In addition, the amount of overlap between the communities should be presented in an effective way, since the user would want to avoid selecting largely overlapping communities in order to efficiently increase the coverage with a few communities.

To this end, our idea is to visualize each community as a circle in 2-d as shown in Figure 6. Each circle is colored by its focus attribute and circle size is proportional to community size. Hovering over each circle displays the exact size, normality, and community members. Importantly, the higher the overlap between two communities, the closer the center points of corresponding circles are placed. We define the distance between two communities C_k and C_l as

$$dist(C_k, C_l) = 1 - \frac{|C_k \cap C_l|}{\min(|C_k|, |C_l|)},$$

and compute the $R \times R$ distances between all pairs of extracted communities. Multi-dimensional scaling (MDS) is used to find a 2-d embedding of the communities such that the pairwise distances as defined above are preserved in the Euclidean space as much as possible. As such, largely

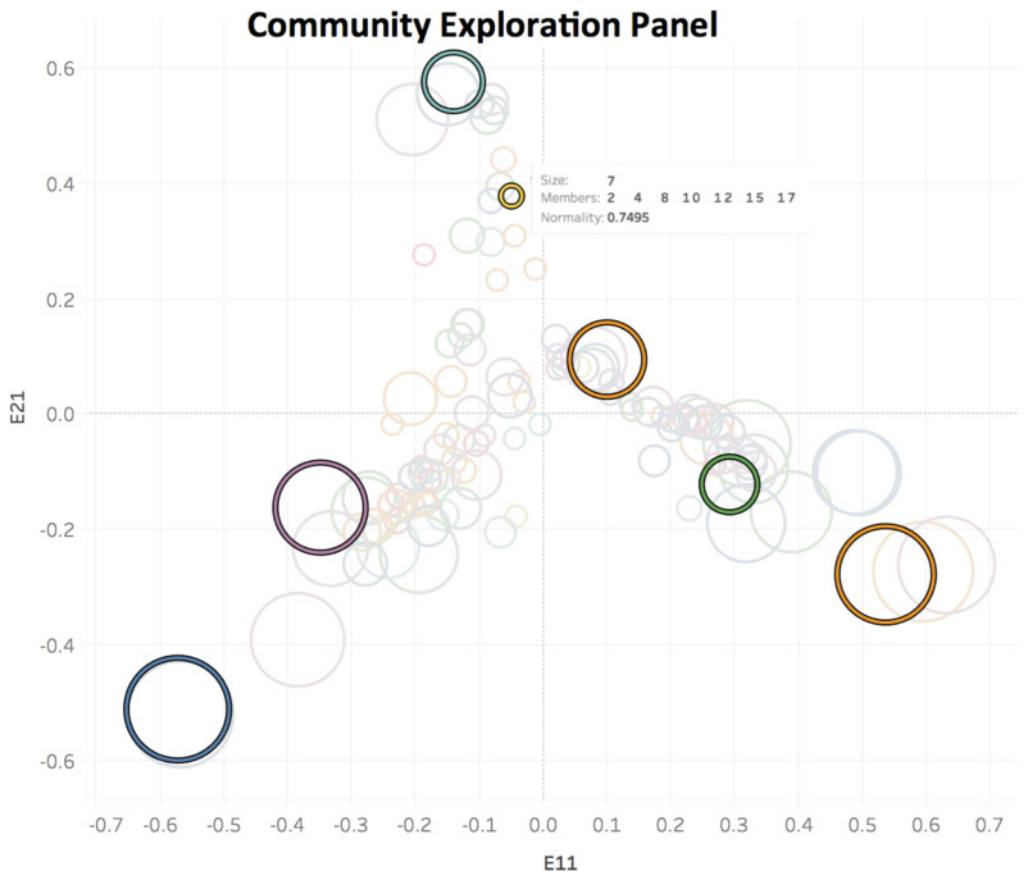


Fig. 6. Community exploration panel. Users can view, select, and de-select circles for summarization, and be displayed the avg. normality, coverage, and diversity of their current summary.

overlapping as well as nested communities are clustered in the display. The user can then aim to select large circles that are spread out in the 2-d embedding in order to most effectively increase coverage.

Figure 6 shows seven (highlighted) circles selected by the user. Each time a user selects (or de-selects) a circle to be included in (or excluded from) the summary, quantities of interest—normality, coverage, diversity—are displayed in (blue) bar plots as in Figure 8 (bottom left). Red vertical lines show the values *before* the last selection, which, in the figure, increased normality, decreased coverage, and did not change diversity.

6.2 Filtering

While searching for circles to select for their summary, the user may want to focus on communities with certain properties. As such, we introduce a panel for filtering communities by focus attribute, normality, and size, as shown in Figure 7. The user can click on the attribute names of interest, use a horizontal slider to specify a range for normality, as well as check/uncheck size values to display only the communities that meet all specified criteria in the mid-panel (Figure 6).

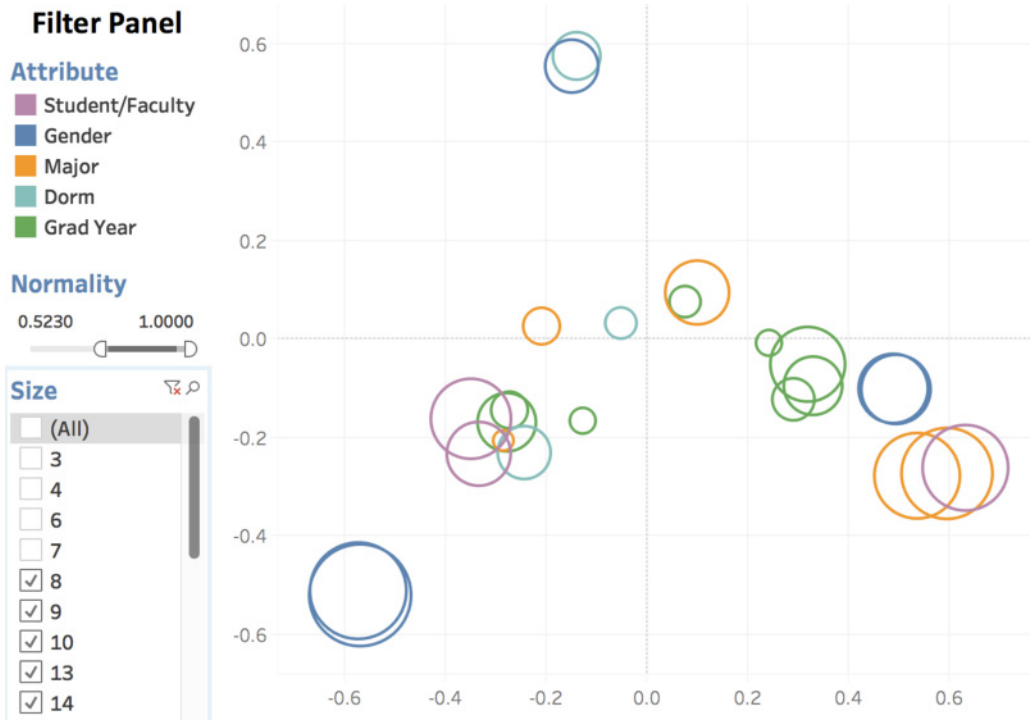


Fig. 7. Filtering (by attribute, normality, and size) panel.

6.3 Algorithm-Guided Human-in-the-Loop Summarization

Finally, we integrate a panel that enables the user to display the output from our summarization algorithm. To do so, the user enters their choice for K, and any two of the weights for normality, coverage, and diversity from drop-down lists (upon which the third remaining weight is automatically set so that their sum is 1) as shown in Figure 8 (top right). Upon user input, K algorithm-selected circles are displayed to the user in a separate plot (top left), along with the quantities of interest (bottom right). This output is likely to guide the user in revising their own summary (Figure 6), toward an alternative and/or better summary than the algorithm's (w.r.t. objective value, recall that the greedy algorithm is not exact).

7 EXPERIMENTS

We evaluate our normality measure and algorithm using several real-world graphs. We first analyze the ranking behavior of our measure in detail, and study the high-quality communities and the type of anomalies we find. Next, we evaluate the performance of AMEN in detecting anomalous communities where we inject anomalies in our graphs by perturbing the high-quality communities. We further compare to the following existing measures and methods: average degree density [12], cut ratio [79], conductance [5], Flake-ODF [21], OddBall [2], Attribute-Weighted Normalized Cut (AW-NCut) [28], and SODA [29]. All but AW-NCut and SODA are structural approaches and ignore the attributes. Finally, we conduct user studies to evaluate the usability of our summarization and visualization approaches in terms of effectiveness (i.e., quality of summaries) and efficiency (i.e., time to summarize).

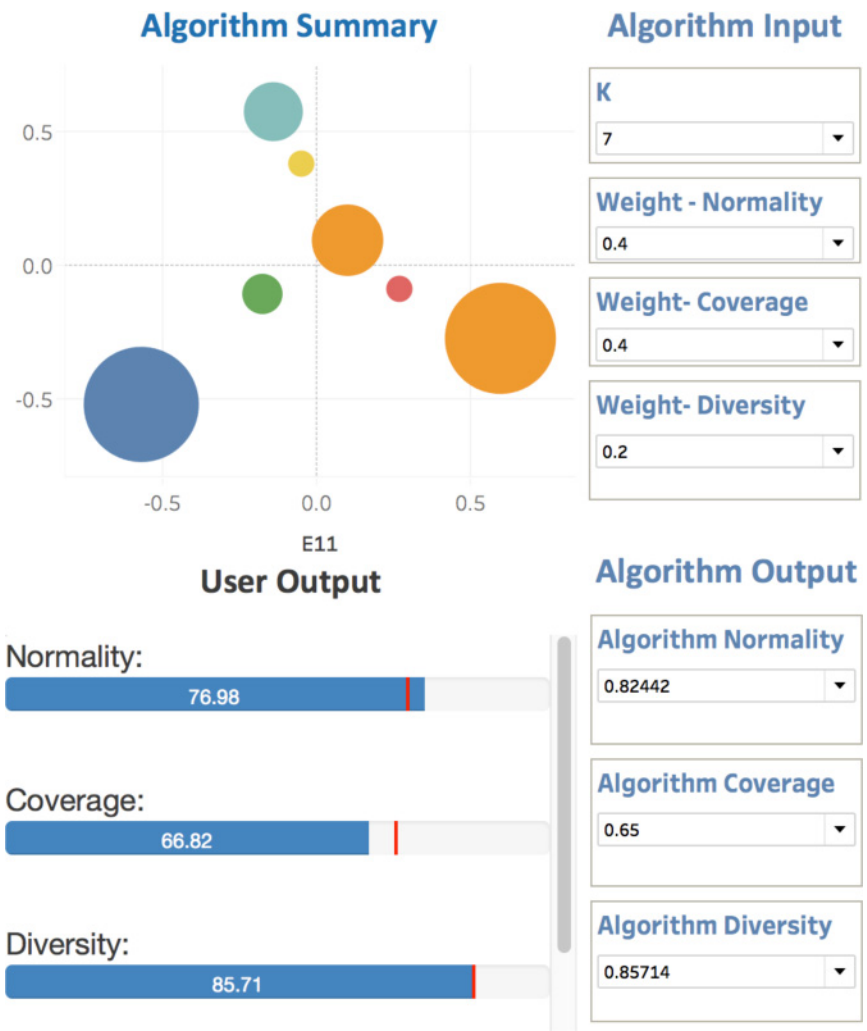


Fig. 8. Algorithmic summary panel. Users (top right) input desired K and weights for quantities of interests, and be displayed (top left) algorithm-selected communities, and (bottom right) algorithmic summary results.

7.1 Dataset Description

A summary of real-world graphs used in this work is given in Table 2.[4] Facebook, Twitter, and Google+ each consists of a collection of ego networks containing ground-truth social circles. The edges in Facebook represent undirected friendship, while the edges in Twitter and Google+ represent a directed "follower" relationship. In our experiments below, we treat all edges as undirected. The attributes for Facebook and Google+ contain profile information about the users, such as their employer, location, or university. The attributes in Twitter are derived from the hashtags and usernames tweeted by each user.

The ground-truth labelings for each network were created differently. The circles from Facebook were manually labeled by 10 survey participants. In Twitter, the circles annotate 1,000 ego

[4]Datasets used in this work can be obtained from http://snap.stanford.edu/data/ and https://code.google.com/p/scpm/.

Table 2. Real-World Networks Used in This Work

Name	$n = \|\mathcal{V}\|$	$m = \|\mathcal{E}\|$	$d = \|\mathcal{A}\|$	$\|C\|$	$\|S\|$
Facebook*	4,039	88,234	42-576	193	21.93
Twitter*	81,306	1,768,149	1–2,271	4,869	12.51
Google+*	107,614	13,673,453	1–4,122	479	134.75
DBLP	108,030	276,658	23,285	N/A	N/A
Citeseer	294,104	782,147	206,430	N/A	N/A
LastFM	272,412	350,239	3,929,101	N/A	N/A

Name	Nodes	Edges	Attributes
Facebook*	Users	Friendships	User profile information
Twitter*	Users	Follow relations	Hashtags and user mentions
Google+*	Users	Follow relations	User profile information
DBLP	Authors	Co-authorships	Title terms used in articles
Citeseer	Articles	Citations	Abstract terms used in articles
LastFM	Users	Friendships	Music pieces listened to

n: number of nodes, m: number of edges, d: number of attributes, $|C|$: number of circles, $|S|$: average circle size. Asterisk (*) depicts datasets with ground truth circles.

networks that have publicly shared Twitter lists. Similarly, in Google+, the circles are for 133 ego networks that have publicly shared some of their circles. We note that these circles are noisy, and frequently do not assign a label to all users in the ego network (this is especially true for Twitter and Google+, where labeling the entire ego network was not an explicit goal).

DBLP, Citeseer, and LastFM are attributed graphs that do not have ground-truth circle membership available. The edges in DBLP represent co-authorship relations between academics, while the edges in Citeseer are citations between articles. In LastFM, the edges capture friendships relations between music listeners. The attributes in DBLP are words used in article titles, and the attributes in Citeseer are terms used in article abstracts. The attributes in LastFM are the names of artists that a user has listened to.

When a graph has ground-truth circles, we consider these circles as the entity communities. For the other graphs, we treat their egonets themselves as the entity communities as they do not contain any circle information. For the purposes of our experiments, we only consider communities that have at least three members and at least one non-zero attribute.

7.2 Analysis of Normality Results

We begin our study by investigating statistics of the communities we consider. Figure 9(a) shows the distributions of community size. We see that the entity communities in LastFM are the smallest, DBLP and Citeseer have very similar community sizes, and Google+ sizes are the most varying, up to 500. We contrast this with the average fraction of edges cut by each community (binned by size), as shown in Figure 9(b). Notice that the communities in Google+ and Citeseer have many cross-edges regardless of size, while a few large communities in Facebook, Twitter, and LastFM have more internal than external edges.

Next, to examine the influence of attributes upon each community, we turn to our normality statistic $\mathbf{x} = (\hat{\mathbf{x}}_I + \hat{\mathbf{x}}_E)$. For each community, we count the number of positive entries of $\mathbf{x}$, and present their distribution $\#(\mathbf{x}(i) > 0)$ in Figure 10(a). As expected, only a small number of attributes are relevant for most communities, so a sparse $\mathbf{w}$ is obtained by L_2. We see that both LastFM and

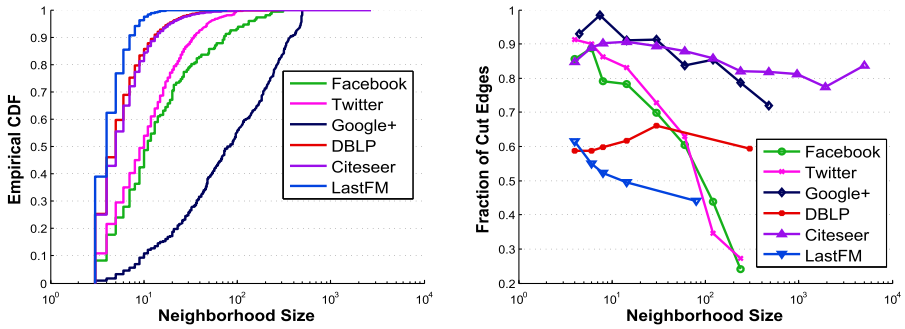


Fig. 9. (a) Community size distribution (cdf) and (b) average fraction of cross-edges to total internal and external edges $\frac{c_s}{m_s+c_s}$ vs. community size.

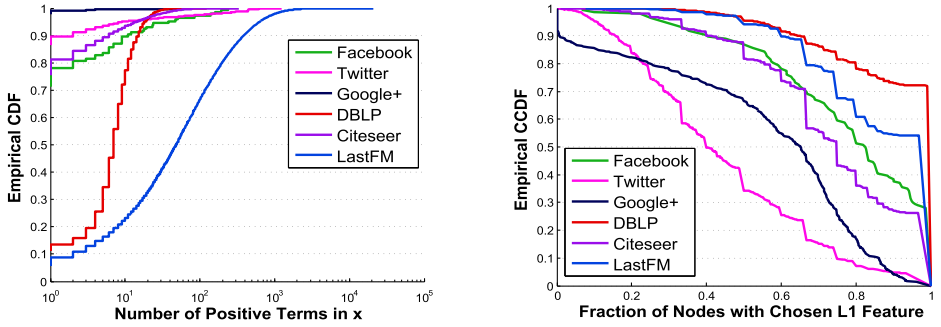


Fig. 10. Distribution of communities w.r.t. (a) count of positive terms in $\mathbf{x}$ (cdf) and (b) fraction of nodes exhibiting the L_1-selected attribute (ccdf).

DBLP have many positive attributes for each community, followed by Facebook, Citeseer, and Twitter. The worst communities are found in Google+, where 99% of the circles do not have *any* attribute that successfully exonerates them. Figure 10(b) shows the fraction of nodes in each community that exhibit the attribute that would be chosen by L_1 maximization. Again, DBLP and LastFM have the most agreement inside each community, followed closely by Facebook and Citeseer, while Twitter and Google+ communities are the most variable in exhibiting the highest ranked attribute.

Figure 11 presents the distribution of normality scores across communities for both L_1 (a) and L_2 (b) maximization. We see that 89% of the DBLP and 95% of the LastFM communities have positive normality scores, while with the exception of a few very good communities, the rest are mostly negative. Using L_2 optimization does not dramatically change where the bulk of the distribution is. The majority of DBLP communities gain additional score, and 10% of LastFM communities gain a lot of score. In other graphs, we see a small improvement, where Google+ again presents the worst performance (its best circle score by L_2 is $\approx$0.6).

The contribution from each positive attribute to the community score for several of our datasets is shown in Figure 12.[5] Across datasets, we see that relevance drops fast, and essentially zeroes out after around 20 attributes. An interesting difference occurs between DBLP and LastFM, which both have many good communities, but achieve them in different ways. In DBLP, the first few

[5]We omit Facebook and Google+, as they each had a limited number of communities with multiple positive attributes.

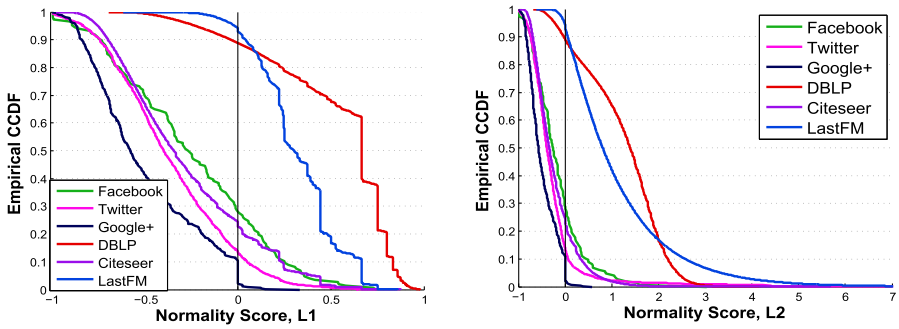


Fig. 11. Distribution of communities w.r.t. normality score (ccdf); $\mathbf{w}$ constraint by (a) L_1 and (b) L_2.

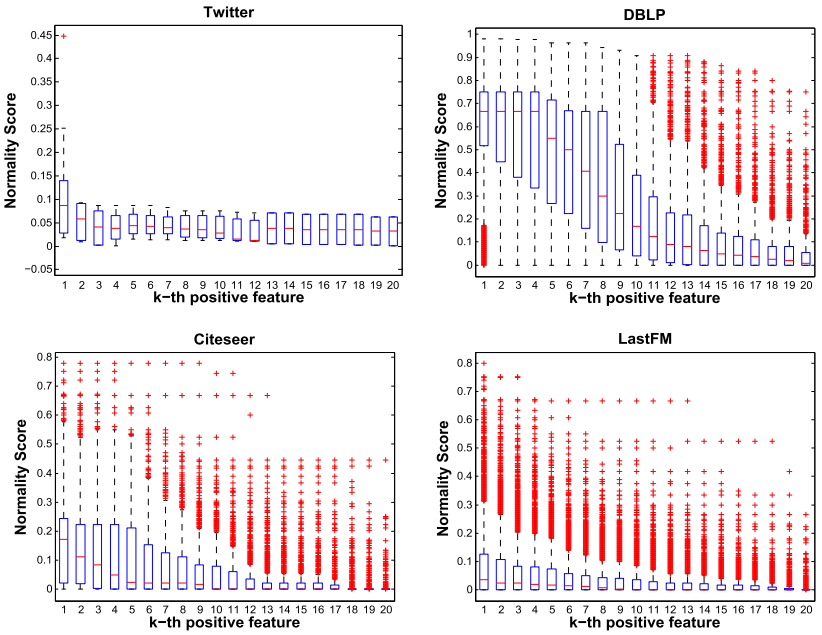


Fig. 12. Normality score of communities based on kth most positive attribute with highest $\mathbf{x}$ entry.

positive attributes are all about equally good, but then they degrade quickly. On the other hand, LastFM attributes are usually not as good, but many of them can add up to achieve a community with a high score (and hence the long tail in Figure 11(b)).

We next examine the difference between the internal consistency I, and the negated external separability $-E$ in Figure 13. In this plot, the communities near the diagonal ($y = x$) have a score of 0, while those further away have an imbalance between I and E becoming increasingly good/bad. The comparison between L_1 and L_2 shows that L_2 constraints on $\mathbf{w}$ increases both the internal and external score for good circles, while bad circles stay the same. This illustrates how the L_2 optimization allows the "rich to get richer," facilitating the improvement of good circles' scores by allowing them to get farther away from the diagonal.

We conclude our analysis by examining normality scores as a function of community size for both L_1 and L_2 constraints on $\mathbf{w}$, as shown in Figure 14. As size increases, we expect a decreasing trend in normality score, as larger neighboorhoods are roughly speaking "less compact." This

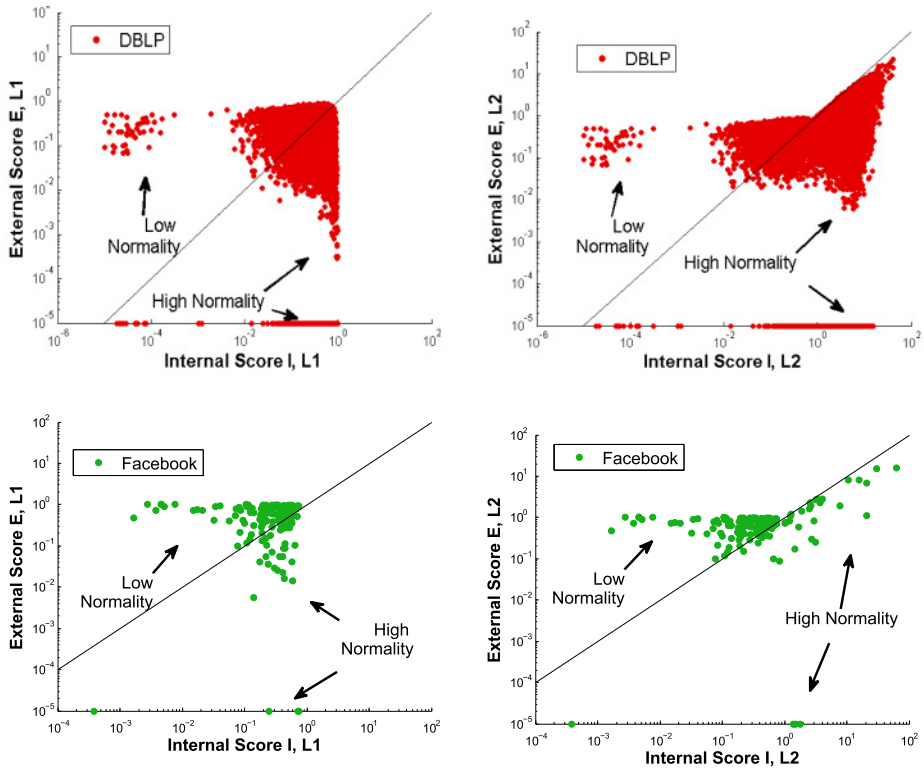


Fig. 13. Internal consistency I vs. negated external separability $-E$ for communities (dots) in DBLP (top) and Facebook (bottom), using $\mathbf{w}$ constraint by (left) L_1 and (right) L_2.

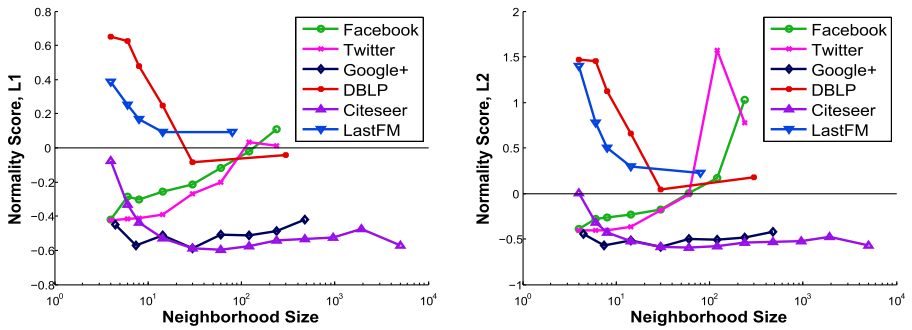


Fig. 14. Average normality score vs. community size; $\mathbf{w}$ constraint by (a) L_1 and (b) L_2.

behavior is clearly apparent in DBLP, Citeseer, and LastFM. An interesting deviation from our expectation occurs in Facebook and Twitter, where larger circles have an increasingly positive normality score. Google+ again behaves differently than the rest of the datasets, exhibiting fairly constant poor normality across changing community sizes.

7.3 Case Studies

Figure 15 presents the L_2-normality scores of communities vs. fraction of their cross-edges. As expected, normality gets lower for communities with increasingly larger cut ratios. However,

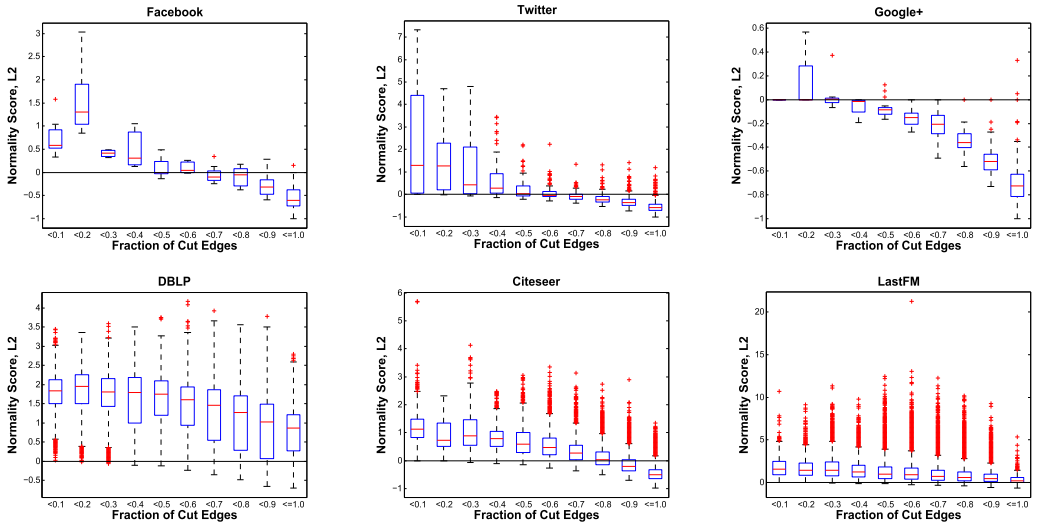


Fig. 15. L_2-normality score vs. fraction of cross-edges to total internal and external edges $\frac{c_s}{m_s+c_s}$.

notice the many communities in DBLP and LastFM with high normality scores even at very high cut ratios. We note this phenomenon to some extent in all the graphs we considered (even occurring a few times in Facebook and Google+). These are exactly the type of communities that are considered low quality by solely structural measures, but achieve high score when attributes and surprise are accounted for.

We illustrate several examples to high and low normality communities from various graphs in Figure 16. For each diagram, the inner circle represents nodes inside an entity community, and the outer circle is the community's corresponding boundary. Colors depict presence (red) or absence (gray) of the attribute chosen by L_1 constraint on $\mathbf{w}$. The dashed edges depict edges "exonerated" by the illustrated attribute. Example communities are arranged by score from high (top-left) to low (bottom-right).

In order of score, we first see a very high scoring community ($L_1 = 0.979$) of authors from DBLP (Figure 16(a)). This ideal community has tight connections between its members and an attribute that exonerates its entire boundary. Its L_2 score of 2.17 indicates that it has more than one attribute that defines it. Next, we see a more realistic example of a good community, this time from Twitter (Figure 16(b)). It still has a high normality score ($L_1 = 0.724$) and good exoneration, but is lacking some internal edges. The next community, from Google+ (Figure 16(c)), illustrates the effects of noisy attributes. We see that the entity community contains a single feature which is in the majority of its members, and not present at all in its boundary. However, several community nodes do not exhibit this feature—which leaves some edges un-exonerated and lowers the community's score. The final three examples (Figure 16(d–f)) are communities of increasingly lower quality. These communities have a few internal edges, and boundaries that cannot be exonerated by any particular attribute.

7.4 Re-Circling the Ground Truth Circles

Next, we study the performance of AMEN in finding good communities in a given graph. Here, we focus on the egonets of Twitter, Facebook, and Google+ that contain ground truth circles. We first find the normality scores of all ground truth circles and consider the normality of an egonet

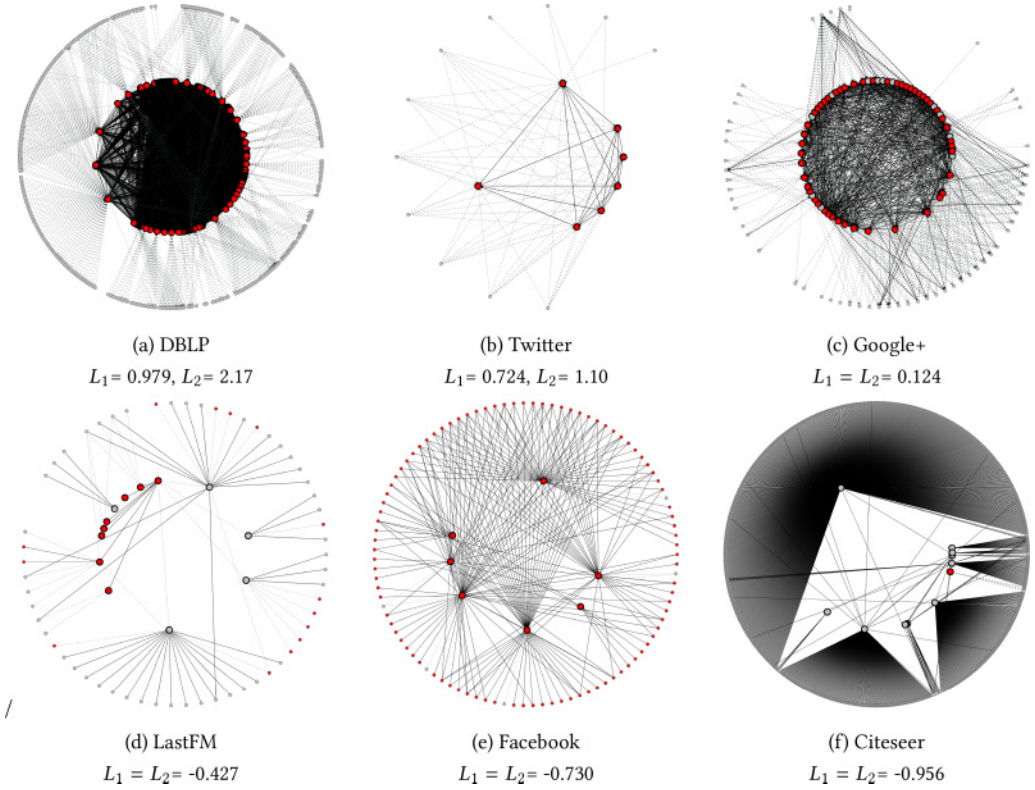


Fig. 16. Example communities (inner circles) and their boundaries (outer circles) in our real-world graphs from high (top-left) to low (bottom-right) normality scores. Colors depict presence (red) or absence (gray) of the attribute chosen by L_1 constraint on $\mathbf{w}$. Dashed edges are "exonerated."

to be the average scores of the circles it contains. Next, we run AMEN on each egonet as the input graph, where we initialize the communities with its ground-truth circles in Line 1 of Algorithm 1. AMEN *refines* these circles as long as the total normality improves.

Figure 17 shows examples of ego networks that have been "re-circled." The original ego networks (with ego removed, for clarity) are shown on the left, and the output of running AMEN on each egonet is shown on the right. Each color represents membership in a distinct circle, and nodes that have multiple colors are members in each corresponding circle (the circles overlap). We see that AMEN finds structurally cohesive, attributed clusters despite the presence of noisy initializations. These clusters can overlap, which illustrates how different initializations expose different focus attributes. We also see that AMEN does not necessarily label all nodes. This can occur as a result of a particular initialization, or because they lack the attributes which define their adjacent circles. We note that nodes of the second variety can be viewed as a form of *focused outliers*, as defined by [61].

Figure 18 presents the normality scores of the egonets before (i.e., with original circles) and after (i.e., after AMEN re-circling). We show both the change in the aggregate egonet score (Figure 18(a)) and for each individual circle (Figure 18(b)). We see that AMEN improves the normality of a large fraction of egonets (and also of most circles), especially for those with initially low normality scores. This suggests that considerable number of ground-truth circles are not well defined (in line with Figure 11), which can be improved automatically by AMEN.

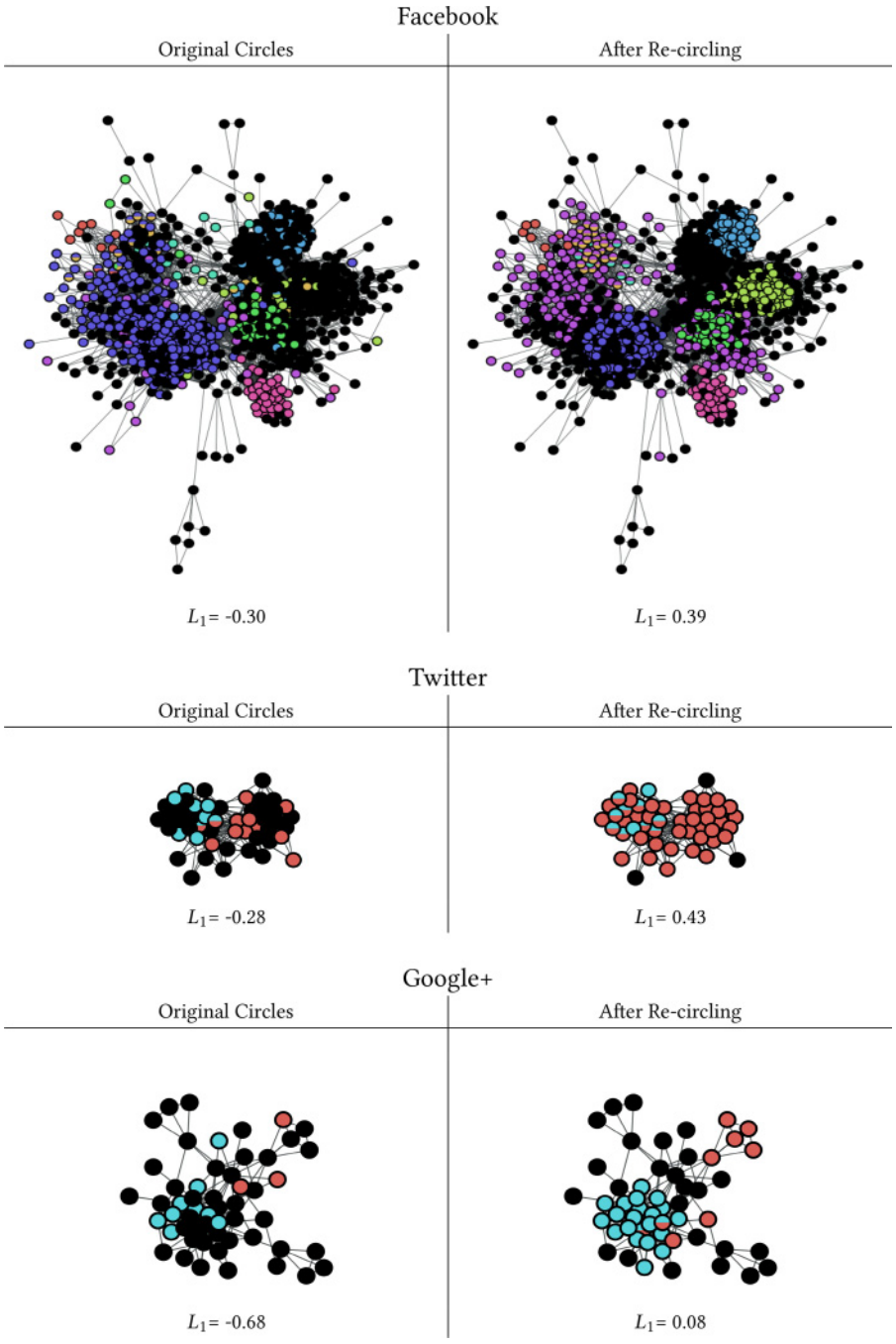
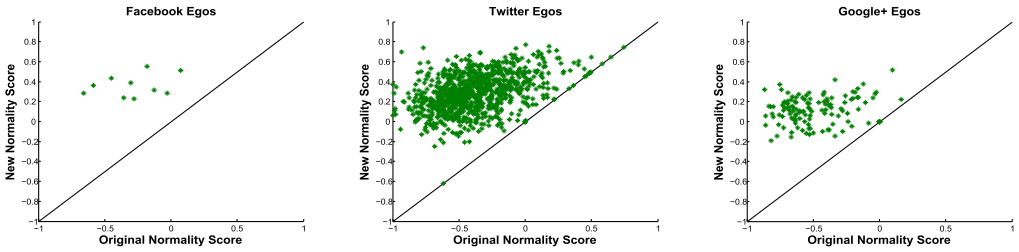
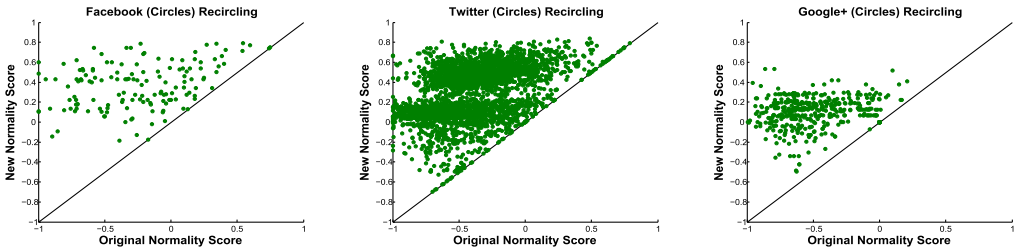


Fig. 17. Example ego networks (with ego removed) *before* (left) and *after* "re-circling" (right) by AMEN. Colors indicate circle membership, and black indicates the absence of any assigned membership.



(a) Ego Normality Improvement



(b) Circle Normality Improvement

Fig. 18. Effects of re-circling on ego networks with ground truth communities. (a) L_1-normality score improvement by AMEN vs. the original L_1-normality score for each ego network in Facebook, Twitter, and Google+. (b) Individual score improvement for each ground-truth circle.

7.5 Anomaly Detection

In the next set of experiments, we study the anomaly detection performance. We create ground truth anomalies using the egonets from DBLP, Citeseer, and LastFM, which we perturb to obtain anomalous communities. Perturbations involve disruptions in (1) structure, (2) attribute space, and (3) both. We start by choosing "good" communities, specifically, small egonets that we expect to have low conductance cuts [24]. From these egonets, we choose 5% of them as anomalous, i.e., to be perturbed. To perturb structure, we rewire inside edges to random outside nodes with rewiring probability p. To perturb attributes, we replace k attributes of inside nodes with corresponding k attributes of randomly picked outside nodes with probability q (note that this inheritance keeps the attributes of outside nodes unchanged). For structure and attribute perturbation, we respectively vary p or q from 0.05 to 0.50. To perturb both, we vary them simultaneously. The larger the perturbation intensities p and q, the more disrupted the egonet. We expect this process to create anomalous (ground truth) egonets, which are structurally poor and/or for which it is also hard to find common focus attributes that would yield high normality.

We evaluate our `normality` measure in its ability to rank the ground truth anomalies high. Specifically, we rank the communities by their normality and report the AUC (i.e., average precision) of the precision-recall plots for each p/q perturbation intensity.

We compare our performance in anomaly detection against the following existing measures and methods. Notation used in baselines: $\mathcal{E}(C) = \{(i,j) \in \mathcal{E} : i \in C, j \in C\}$ (edges induced by C); $cut(C) = \sum_{i \in C, b \in B, (i,b) \in \mathcal{E}} 1$ (cut size induced by C); $vol(C) = \sum_{i \in C} k_i$ (sum of degrees in C).

— *Average degree* [12], $\frac{2|\mathcal{E}(C)|}{|C|}$ (internal consistency only, non-attributed).

— *Cut ratio* [79], $\frac{cut(C)}{|C|(n-|C|)}$, is the fraction of boundary edges over all possible boundary edges (external separability only, non-attributed).

— *Conductance* [5], $\frac{cut(C)}{\min(vol(C), vol(G \backslash C))}$, normalizes the cut by the total volume of C (internal+external quality, non-attributed).
— *Flake-ODF* [21], $\frac{|\{i \in C : |\{(i,j) \in \mathcal{E} : j \in C\}| < k_i/2\}|}{|C|}$, is the fraction of nodes inside a neighborhood that have less than half of their edges pointing inside (internal+external quality, non-attributed).
— *OddBall* [2] uses a linear model to find neighborhoods that deviate in node density (internal consistency only, non-attributed).
— *SODA* [29] finds a max-margin hyperplane that separates connected and disconnected nodes using both structure and attributes. It ranks neighborhoods by the negative margin of this hyperplane (internal+external quality, attributed).
— *AW-NCut* is based on a cluster quality measure proposed in [28] for attributed graphs. It identifies a subspace of attributes for a cluster, which minimizes its weighted normalized cut, where edges are weighted by the similarity of end-nodes on the selected subspace. Subspace selection, however, is *quadratic* in the number of all attributes. Our real-world datasets DBLP, Citeseer, and LastFM have more than 23,000, 206,000, and 3.9 million attributes, respectively, for which [28] is intractable. As such, we consider a simplified version, by using a uniform weight vector over the full attribute space to compute normalized cut (internal+external quality, attributed).

Results are shown in Figure 19. Our normality consistently outperforms all other measures and methods, especially when attributes are perturbed. When structure perturbation is involved, conductance appears to do well, whereas for attribute perturbation SODA is the second best while still much worse than normality. Across perturbation strategies and datasets, on average AMEN outperforms Flake-ODF by 16%, conductance by 18%, AW-NCut by 20%, SODA by 23%, average degree by 24%, cut ratio by 24%, and OddBall by 25%.

7.6 User Study

7.6.1 Experiment Setup. Finally, we evaluate the usability of our proposed interactive visual exploration and summarization approach in terms of effectiveness (i.e., quality of summaries) and efficiency (i.e., time to summarize). To this end, we conduct a user study with five graduate student participants. Each participant is to analyze communities from five different egonetworks from Facebook (datasets D1, . . . ,D5), with varying size $n = \{54, 106, 144, 208, 251\}$, and build a representative summary containing $K = \{5, 10\}$ communities.

The setup is as follows. Each user is first shown Panels 1 and 2 in Figure 5, community exploration and filtering panels, respectively, and asked to interactively select $K = 5$ communities with a goal to achieve as high objective value as possible while giving equal importance to normality, coverage, and diversity (in other words, assuming $\alpha = \beta = \frac{1}{3}$). Each user is then asked to make $K = 10$ selections, keeping α and β the same as before. Users are shown D1, . . . ,D5 consecutively in the same order. We record the (1) avg. normality, coverage, diversity achieved by respective summarization tasks, and (2) time taken to construct each summary (in seconds) per user.

Next, the panels are cleared and datasets D1, . . . ,D5 are shown to each user one by one once again, where this time Panel 3 displaying the algorithm-generated summary is also shown (for the corresponding dataset and K; $\alpha = \beta = \frac{1}{3}$). Each user is then asked to build an alternative summary to the algorithm's, ideally with equally high objective value, where they could use the algorithm output as guidance. Again, the three quantities of interest as well as time-to-summarize are recorded for each alternative summarization task.

Overall, we conduct 100 summarization tasks—using 5 participants × 5 datasets × 2 different K × 2 different settings (with and without algorithmic guidance).

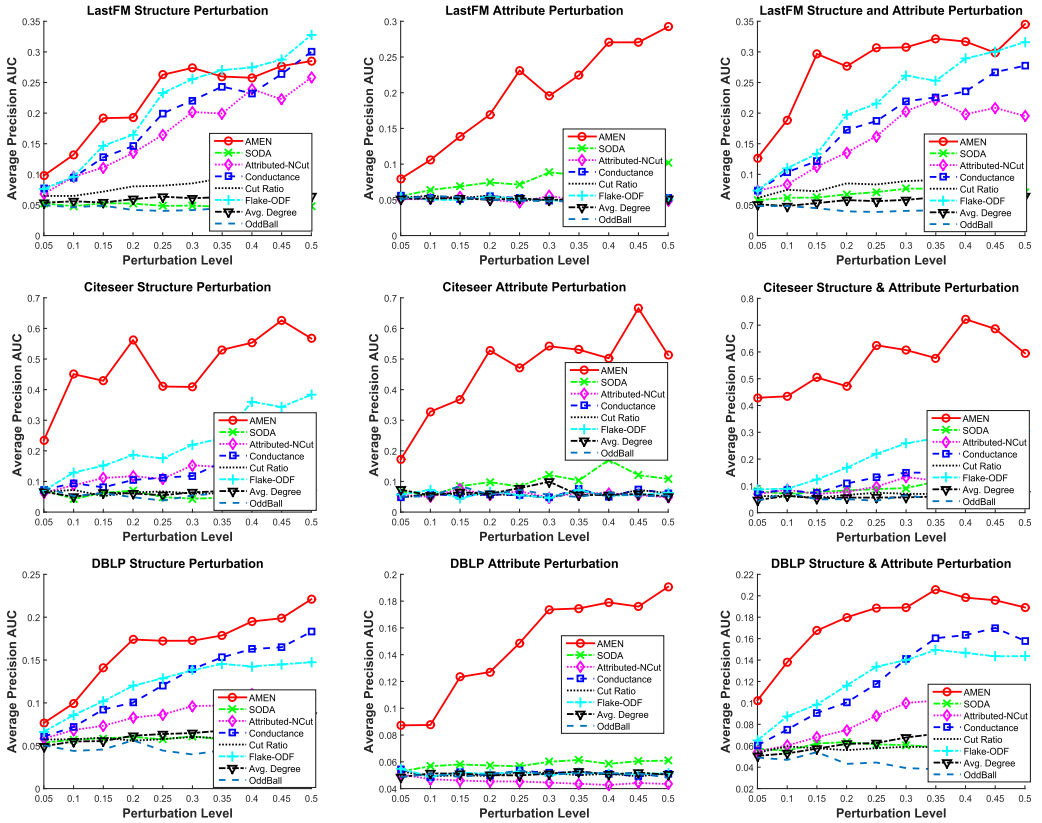


Fig. 19. Anomaly detection results for structure (left), attribute (center), and both (right) anomalies on LastFM (top), Citeseer, and DBLP (bottom).

7.6.2 Experiment Results. Through our user study, we answer four main questions Q1–Q4 that we present in this section.

(Q1) Summarization by visual exploration. *Does interactive visualization help users construct effective summaries, as compared to strawman baselines?*

Our goal is to understand if the users can achieve high values for normality, coverage, and diversity by using our interactive interface to construct their summaries. For comparison, we consider two simple baselines, TopS and TopN, which respectively select K communities with the largest size and largest normality.

Figure 20 shows the quantities achieved on average across users for each dataset and (K) along with those by the baselines. As expected, TopN achieves high normality but poor coverage and TopS gives high coverage but inferior normality. The avg. user finds a well-balanced tradeoff between the quantities. In Figure 21, we show the objective value (weighted sum) achieved under each setting and on average overall. The avg. user outperforms TopS in all and TopN in most cases, with 28.7% and 10.8% relative improvement over these baselines respectively on average (rightmost bars).

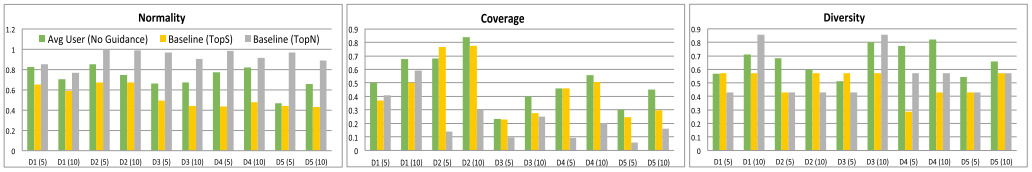


Fig. 20. Average user vs. strawman baselines. (From top to bottom) Normality, coverage, and diversity across summarization tasks <dataset>(K).

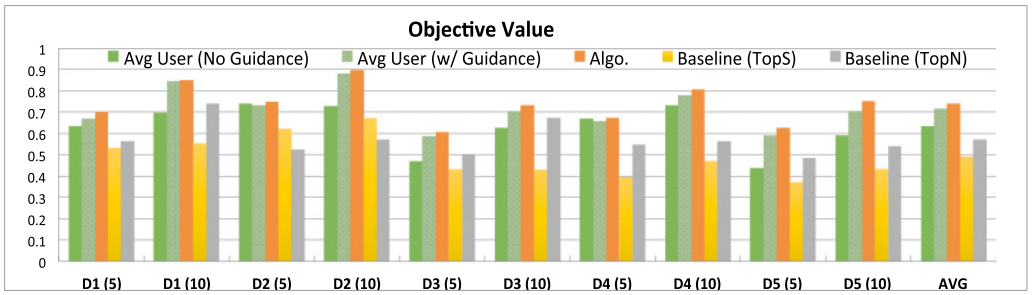


Fig. 21. Objective values achieved by various approaches across summarization tasks <dataset>(K) for $\alpha = \beta = \frac{1}{3}$.

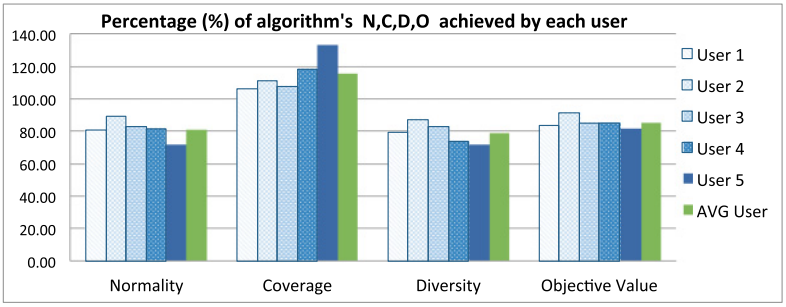


Fig. 22. Percentage normality, coverage, diversity, and overall objective value of the algorithm's as achieved by each user and the avg. user (avg'ed across datasets and (K)).

(Q2) *How close do the summaries by users* **without guidance** *get to the algorithm results (in terms of normality, coverage, diversity, and overall objective value)?*

Next, we aim to understand how the user outcomes compare to those of our summarization algorithm. If the quantities of interest are comparable to the algorithm's, we would conclude that the visual interface is quite useful to the users.

To this end, we compute $\frac{100 q_{user}}{q_{algo}}$, where q correspond to individual quantities $\{N, C, D, O\}$ for normality, coverage, diversity, and overall objective value, respectively. The main finding from Figure 22 is that the users tend to put most emphasis on coverage, and less on normality and diversity. On average, users achieve 115.3% of algorithm's coverage and ≈80% of the normality and coverage. Overall, they reach 85% of the algorithm's overall objective value.

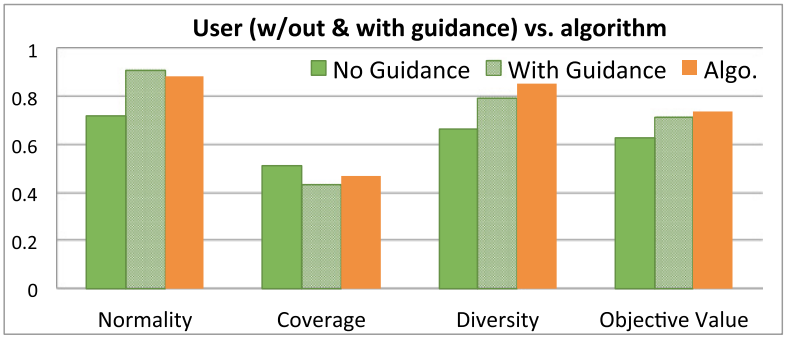


Fig. 23. Comparison of users without and with algorithmic guidance to the algorithm results (avg'ed across users and datasets (K).

Table 3. Percent (%) Improvement in Objective Value by Each User on Each Data/Task *After Algorithmic Guidance*

	D1(5)	D1(10)	D2(5)	D2(10)	D3(5)	D3(10)	D4(5)	D4(10)	D5(5)	D5(10)	AVG
U1	112.59	156.44	99.53	114.31	129.89	130.58	92.20	106.17	170.86	121.08	123.37
U2	91.79	118.14	87.56	102.86	99.19	112.31	92.66	100.00	107.39	117.97	102.99
U3	101.60	112.95	101.30	120.73	140.15	101.75	85.78	96.60	199.57	142.96	120.34
U4	103.98	104.18	100.85	140.65	103.76	105.94	116.86	124.73	110.13	109.13	112.02
U5	117.61	124.02	102.70	129.06	169.17	117.77	105.06	106.17	113.34	109.65	119.45
Avg	105.51	123.15	98.39	121.52	128.43	113.67	98.51	106.73	140.26	120.16	115.63

(Q3) Alternative summarization by algorithmic guidance. *How much* **guidance** *does our summarization algorithm provide users to derive alternative summaries and improve over their earlier results?*

Next, we investigate the effect of the algorithmic guidance on user's summarization behavior and performance. As shown in Figure 23, we find that the users tend to construct alternative summaries with significantly higher normality and diversity than before, and decrease their emphasis on coverage. The guidance helps them obtain (alternative) summaries with higher objective value, which are also nearly as good as the algorithm's on average (also see Figure 21 per task).

Table 3 lists the percentage objective value of their earlier results (without guidance) as achieved after algorithmic guidance. That is, we compute $100\ O_{user}^{(after)}/O_{user}^{(before)}$, for each user and the avg. user per summarization task. We find that users improve their objective value by 102–124%, with an average of 115.63% across tasks.

(Q4) Efficiency. *How long does it take per user on average to construct (i) a summary without guidance, and (ii) alternative summary with guidance?*

Finally, we aim to understand how long users take to build their summaries without guidance, and how their efficiency is affected when they are presented with the algorithm results.

Figure 24 shows that the avg. users spends anywhere from 2.5 to 5 minutes to build a summary without guidance. The longest time is on D1 ($K = 5$), which is the very first dataset/task presented to each user, which can be seen as the warm-up period. Overall, (AVG) is 3.8 minutes across tasks.

When algorithm results per task are also shown to the users, their summarization time drops considerably, to around 3 minutes on average. This is mainly for two reasons. Obviously, the users

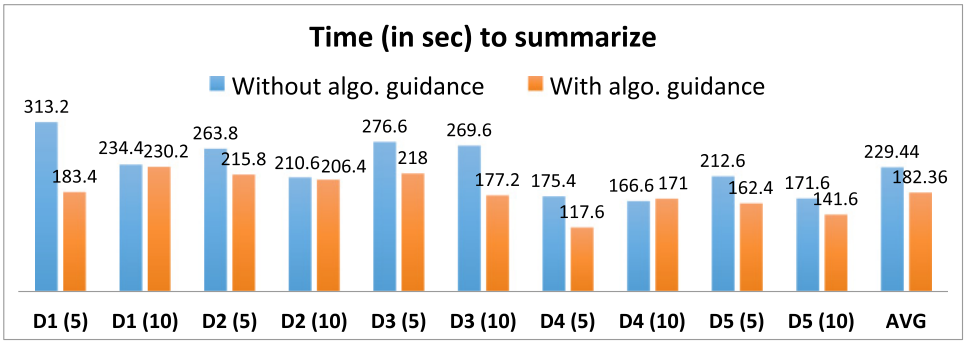


Fig. 24. Time (in seconds) that an average user takes to construct a summary without guidance (blue) and with algorithmic guidance (orange) on each summarization task.

search for circles similar to those selected by the algorithm. A less obvious reason is that the users consider the algorithm results as upper bound, and stop as soon as they build an alternative summary that yields close values (earlier, when the users did not have a "bar" to reach, they were less sure about when to stop).

Remarks. In summary, we find that our visualization interface helps users to construct summaries effectively and efficiently. Their summaries achieve a good tradeoff between the quantities of interest, and reach better objective values than simple baselines. Without guidance, the users tend to put most emphasis on coverage, which they tend to correct for normality and diversity when they are provided with algorithmic guidance. The guidance helps them improve over their earlier outcomes and obtain equally good alternative summaries to the algorithm's. Algorithmic guidance, which serves as a "bar to reach," further improves users' efficiency and help them terminate their search earlier.

8 RELATED WORK

We organize related works into five groups: (1) analysis of community structure, (2) community extraction, (3) anomaly detection in non-attributed and attributed graphs, (4) graph summarization, and (5) other relevant research.

Analysis of community structure. Huang et al. [39] studied the statistical properties of communities in real networks and investigated how they split into communities and how typical community qualities change over a range of size scales. Their findings suggested the absence of large well-defined communities, which has been corroborated in a later study by [24]. Arnaboldi et al. [7] analyzed the structure of ego networks and found that social relationships in online and offline social networks are organized similarly. Akoglu et al. [2] found that egonet characteristics form power-law-like patterns in real networks. Other works studied the structure and dynamics of real-world graphs at large, without specific focus on communities [37, 47]. These works focused mainly on networks with no attributes.

Several works quantify the quality of community structure in non-attributed graphs, such as the modularity measure by [54]. Yang and Leskovec [79] investigated a long list of other such measures and compared their performance based on ground-truth communities. Newman [53] also studied the mixing properties in attributed graphs to quantify the correlations between the attributes of adjacent nodes. High correlation is referred as assortative mixing, which tends to break the network into communities [52]. On similar lines, Silva et al. [68] studied the correlation

between attribute sets and the occurrence of dense subgraphs, called the structural correlation patterns in attributed graphs.

Community extraction. A large body of algorithms aims to optimize modularity for community detection [10, 16, 54, 67]. There also exist well-established algorithms for graph partitioning [32, 69]. Several other works focus on overlapping community detection, including methods that utilize generative models [78], seed set expansion [6, 15, 76], non-negative matrix factorization [36, 80], and label propagation [18]. For other related work we refer to a survey by [77]. These works mainly focus on non-attributed graphs.

More recently, community detection in attributed graphs has attracted considerable attention. Such research focuses on finding communities both dense in structure and coherent in some attribute subspace. Most of these methods extract disjoint communities [4, 23, 27, 44, 86], while several others extend to detecting overlapping attributed communities [22, 46, 61, 81]. Most of these methods focus on internal density, while completely ignoring the community boundary. Moreover, they do not exhibit the notion of "exonerating" edges based on surprise and attribute similarity as in this work.

Graph anomaly detection. Noble and Cook [58] used frequent subgraph mining and information theoretic principles to identify anomalous subgraphs in graphs with a single attribute. Akoglu et al. [2] focused on structural anomalies, where they found and used patterns in egonets to flag the anomalies. Li et al. [40, 41] developed algorithms to find sets of connected nodes in a graph for which a single attribute value is significantly higher than in the neighborhood of the set. Gao et al. [23] formulated a new problem to identify community outliers which deviate in the full attribute space from others that belong to the same community. Perozzi et al. [61] focused on extracting communities that agree on a pre-defined subset of attributes that is inferred from user preference. They also output community outliers that deviate either partially or fully in this subset.

Most similar to ours is the work by [29] on outlier subgraph discovery. Their formulation involves inferring an attribute subspace in which the margin between minimum dissimilarity among disconnected nodes and maximum dissimilarity among connected nodes is maximized. For normal subgraphs, this margin is expected to be large. Our normality formulation is considerably different and yields a much easier optimization problem. Moreover, none of the existing works has a notion of edge "exoneration" as we introduce in this work. For additional related work on graph anomaly mining, we refer to a survey by [3].

Graph summarization. A simple approach to graph summarization is to characterize by summary statistics, such as degree distribution, diameter, triangle counts, and the like [20, 34, 38, 74]. This comes with two drawbacks; first, summary statistics lose most of the structural (or topology) information and second, it is not obvious how to couple such information with the attribute information. A different approach is to embed the input network into a 2-d or 3-d point space, where nodes correspond to points and proximities between the nodes in the graph are preserved (to a large extent) in the Euclidean space. Various graph embeddings are proposed mostly for plain [9, 26, 62, 70] and most recently for attributed networks [11, 31]. Like with many embedding/dimensionality reduction techniques, however, these produce abstract, non-interpretable representations for nodes.

In addition, there is a body of work for large graph summarization [17, 19, 35, 84] (with substructures or motifs) [71, 83] (via grouping or coarsening), querying [13, 82], and interactive visualization and exploration [1, 14, 64]. (For more, we refer to [43, 63].) However, none of these can handle all of the following at the same time: (i) networks with a long list of attributes, (ii) representative and yet succinct summaries, and (iii) interactive visual exploration.

Other related work. Finally, we also point to a few related works in user profiling [42], and attribute inference [66, 85]. The main focus of these works is identifying the values of missing or incorrect attributes for certain nodes in a given network. Common approaches leverage local neighborhood of nodes and the homophily property in social networks. While our work aims to quantify community quality and detect anomalous communities, there exist similarities. For instance, our normality measure may help in attribute inference where missing entries are assigned the values that improve the normality the most. Moreover, our algorithm may reveal communities in which a few nodes deviate from others in their focus attributes, which may indicate incorrect entries. Future work could investigate the connection between the two research areas in more detail.

9 CONCLUSION

We considered the following related problems: how to measure the quality of communities in attributed graphs, extract and summarize communities in a large attributed network, and interactively visualize and explore the communities. We introduced new approaches to these research questions, with three-fold contributions: (1) a *new measure of subgraph quality* for attributed communities called `normality`, (2) a *community extraction* algorithm, and (3) a summarization and visualization approach for attributed graph exploration.

Specifically, we proposed a new measure called `normality` that evaluates quality both internally and externally. Intuitively, a high quality community has members with many surprising edges connecting them that share similar values in a particular attribute subspace, called the *focus* attributes. Moreover, it has either a few edges at the boundary, or many cross-edges can be exonerated as unsurprising and/or dissimilar with respect to the focus. To the best of our knowledge, we are the first to define a formal quality measure for communities in attributed graphs and to provide related solutions for optimizing it (i) when the focus attributes of given communities are unknown, as well as (ii) when both the communities and their focus are latent. Our approach is also the first of its kind in the way it allows for many cross-edges and carefully accounts for them through the notions of surprise and exoneration. In addition, we introduced a new formalism for summarizing attributed graphs through the communities that make up their backbone, with adjustable importance on graph coverage, diversity of attributes, and quality. Finally, we designed and developed an interactive visualization and exploration interface for users to analyze extracted communities, build summaries either from scratch or by building on automatically-generated algorithmic summaries. Experiments on various real-world graphs demonstrate the utility and performance of our `normality` measure and algorithm in finding high and low quality communities, where we outperform other well-established measures and methods. Moreover, extensive user studies show the utility of our interactive exploration tool.

While there are separate related work (in community extraction, graph summarization, and interactive visualization), we uniquely bring those fronts together under a concerted effort and provide an end-to-end analytics solution to sensemaking of attributed networks.

Our work sets out future research directions that include investigating potential applications of network summaries. For instance, ego-network summaries can help with attribute inference [66] and profiling [42], which involve inferring missing attributes of a node (the ego) based on the community characteristics of its neighbors that make up the local network. Egonet summaries can also be used to find similar nodes in large attributed networks with similar community compositions, which can help in role discovery [30].

We publicly share the source code for our community extraction algorithm utilizing the proposed `normality` measure (in Matlab), the summarization algorithm (in Matlab), and our interactive visualization toolkit (in Tableau) for reproducibility, academic and non-commercial use at the following link: http://bit.ly/2wcttyP.

ACKNOWLEDGMENTS

Any conclusions expressed in this material are of the authors' and do not necessarily reflect the views, expressed or implied, of the funding parties. We also thank Rashmi Raghunandan, Shruti Sridhar, and Upasna Suman for their help in the design and implementation of iSCAN, the interactive GUI for community exploration and summarization.

REFERENCES

[1] Leman Akoglu, Duen Horng Chau, U. Kang, Danai Koutra, and Christos Faloutsos. 2012. OPAvion: Mining and visualization in large graphs. In *SIGMOD Conference*. 717–720.
[2] Leman Akoglu, Mary McGlohon, and Christos Faloutsos. 2010. Oddball: Spotting anomalies in weighted graphs. In *PAKDD*. 410–421.
[3] Leman Akoglu, Hanghang Tong, and Danai Koutra. 2014. Graph-based anomaly detection and description: A survey. *DAMI* 28, 4 (2014). DOI: http://dx.doi.org/10.1007/s10618-014-0365-y
[4] Leman Akoglu, Hanghang Tong, Brendan Meeder, and Christos Faloutsos. 2012. PICS: Parameter-free identification of cohesive subgroups in large attributed graphs. In *SDM*. 439–450.
[5] R. Andersen, F. Chung, and K. Lang. 2006. Local graph partitioning using pagerank vectors. In *FOCS*.
[6] Reid Andersen and Kevin J. Lang. 2006. Communities from seed sets. In *WWW*. 223–232.
[7] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Fabio Pezzoni. 2012. Analysis of ego network structure in online social networks.. In *SocialCom/PASSAT*. IEEE, 31–40.
[8] James Bailey. 2013. Alternative clustering analysis: A review. In *Data Clustering: Algorithms and Applications*. CRC Press, 535–550.
[9] Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15, 6 (2003), 1373–1396.
[10] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. L. J. S. Mech. 2008. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 10 (2008), 10008.
[11] Shiyu Chang, Wei Han, Jiliang Tang, Guo-Jun Qi, Charu C. Aggarwal, and Thomas S. Huang. 2015. Heterogeneous network embedding via deep architectures. In *KDD*. 119–128.
[12] Moses Charikar. 2000. Greedy approximation algorithms for finding dense components in a graph. In *APPROX*.
[13] Duen Horng Chau, Christos Faloutsos, Hanghang Tong, Jason I. Hong, Brian Gallagher, and Tina Eliassi-Rad. 2008. GRAPHITE: A visual query system for large graphs. In *ICDM Workshops*. 963–966.
[14] Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. 2011. Apolo: Interactive large graph sensemaking by combining machine learning and visualization. In *KDD*. 739–742.
[15] Aaron Clauset. 2005. Finding local community structure in networks. *Physical Review E* 72 (2005), 6.
[16] A. Clauset, M. E. J. Newman, and C. Moore. 2004. Finding community structure in very large networks. *Physical Review E* 70, 6 (2004), 066111.
[17] Diane J. Cook and Lawrence B. Holder. 1994. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research* 1 (1994), 231–255.
[18] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. 2012. DEMON: A local-first discovery method for overlapping communities. In *KDD*. 615–623.
[19] Cody Dunne and Ben Shneiderman. 2013. Motif simplification: Improving network visualization readability with fan, connector, and clique glyphs. In *CHI*. 3247–3256.
[20] M. Faloutsos, P. Faloutsos, and C. Faloutsos. 1999. On power-law relationships of the internet topology. In *ACM SIGCOMM*. 251–262.
[21] Gary William Flake, Steve Lawrence, and C. Lee Giles. 2000. Efficient identification of web communities. In *KDD*.
[22] Esther Galbrun, Aristides Gionis, and Nikolaj Tatti. 2014. Overlapping community detection in labeled graphs. *Data Mining and Knowledge Discovery* 28, 5–6 (2014), 1586–1610.
[23] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han. 2010. On community outliers and their efficient detection in information networks. In *KDD*. 813–822.
[24] David F. Gleich and C. Seshadhri. 2012. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *KDD*. 597–605.
[25] A. V. Goldberg. 1984. *Finding a Maximum Density Subgraph*. Technical Report CSD-84-171. UC Berkeley.
[26] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.
[27] Stephan Günnemann, Ines Färber, Brigitte Boden, and Thomas Seidl. 2010. Subspace clustering meets dense subgraph mining: A synthesis of two paradigms. In *ICDM*. 845–850.
[28] Stephan Günnemann, Ines Farber, Sebastian Raubach, and Thomas Seidl. 2013. Spectral subspace clustering for graphs with feature vectors. In *ICDM*. IEEE, 231–240.

[29] Manish Gupta, Arun Mallya, Subhro Roy, Jason H. D. Cho, and Jiawei Han. 2014. Local learning for mining outlier subgraphs from network datasets. In *SIAM SDM*. 73–81.
[30] Keith Henderson, Brian Gallagher, Tina Eliassi-Rad, Hanghang Tong, Sugato Basu, Leman Akoglu, Danai Koutra, Christos Faloutsos, and Lei Li. 2012. RolX: Structural role extraction and mining in large graphs. In *KDD*. ACM, 1231–1239.
[31] Xiao Huang, Jundong Li, and Xia Hu. 2017. Accelerated attributed network embedding. In *SDM*. 633–641.
[32] G. Karpis and V. Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* 20, 1 (1998), 359–392.
[33] George Karypis and Vipin Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing* 20 (1998), 359–392.
[34] Danai Koutra, Di Jin, Yuanshi Ning, and Christos Faloutsos. 2015. Perseus: An interactive large-scale graph mining and visualization tool. *PVLDB* 8, 12 (2015), 1924–1927.
[35] Danai Koutra, U. Kang, Jilles Vreeken, and Christos Faloutsos. 2014. VOG: Summarizing and understanding large graphs. In *SDM*. 91–99.
[36] Darong Lai, Xiangjun Wu, Hongtao Lu, and Christine Nardini. 2011. Learning overlapping communities in complex networks via non-negative matrix factorization. *International Journal of Modern Physics C* 22, 10 (2011), 1173–1190.
[37] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. 2005. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *KDD*. ACM, 177–187.
[38] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. 2005. Graphs over time: Densification laws, shrinking diameters and possible explanations. In *KDD*. 177–187.
[39] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. 2008. Statistical properties of community structure in large social and information networks. In *WWW*. 695–704.
[40] Nan Li, Ziyu Guan, Lijie Ren, Jian Wu, Jiawei Han, and Xifeng Yan. 2013. gIceberg: Towards iceberg analysis in large graphs. In *ICDE*. 1021–1032.
[41] Nan Li, Huan Sun, Kyle Chipman, Jemin George, and Xifeng Yan. 2014. A probabilistic approach to uncovering attributed graph anomalies. In *SIAM SDM*. 82–90.
[42] Rui Li, Chi Wang, and Kevin Chen-Chuan Chang. 2014. User profiling in an ego network: Co-profiling attributes and relationships. In *WWW*. 819–830.
[43] Yike Liu, Abhilash Dighe, Tara Safavi, and Danai Koutra. 2016. A graph summarization: A survey. *CoRR abs/1612.04883* (2016).
[44] Bo Long, Zhongfei (Mark) Zhang, Xiaoyun Wu, and Philip S. Yu. 2006. Spectral clustering for multi-type relational data. In *ICML*, vol. 148. 585–592.
[45] Douglas S. Massey and Nancy A. Denton. 1988. The dimensions of residential segregation. *Social Forces* 67, 2 (1988), 218–315.
[46] Julian J. McAuley and Jure Leskovec. 2014. Discovering social circles in ego networks. *ACM Transactions on Knowledge Discovery from Data* 8, 1 (2014), 4:1–4:28.
[47] Mary McGlohon, Leman Akoglu, and Christos Faloutsos. 2008. Weighted graphs and disconnected components: patterns and a generator. In *KDD*. 524–532.
[48] Miller McPherson, Lynn Smith-Lovin, and James M. Cook. 2001. Birds of a feather: Homophily in social networks. *Annual Review of Sociology* 27, 1 (2001), 415–444.
[49] J. Moody. 2001. Race, school integration, and friendship segregation in America. *American Journal of Sociology* 107, 3 (2001), 679–716.
[50] George L. Nemhauser and Laurence A. Wolsey. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research* 3, 3 (1978), 177–188.
[51] M. E. J. Newman and M. Girvan. 2003. Mixing patterns and community structure in networks. In *Statistical Mechanics of Complex Networks*, Vol. 625. 66–87.
[52] M. E. J. Newman. 2002. Assortative mixing in networks. *Physical Review Letters* 89, 20 (2002).
[53] M. E. J. Newman. 2003. Mixing patterns in networks. *Physical Review E* 67 (2003).
[54] M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America* 103, 23 (2006), 8577–8582.
[55] M. E. J. Newman. 2010. *Networks: An Introduction*. Oxford University Press, Oxford; New York.
[56] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. 2001. On spectral clustering: Analysis and an algorithm. In *NIPS*.
[57] Donglin Niu, Jennifer G. Dy, and Michael I. Jordan. 2014. Iterative discovery of multiple alternative clustering views. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 7 (2014), 1340–1353.
[58] Caleb C. Noble and Diane J. Cook. 2003. Graph-based anomaly detection. In *KDD*. ACM, 631–636.
[59] Jeffrey Pattillo, Alexander Veremyev, Sergiy Butenko, and Vladimir Boginski. 2013. On the maximum quasi-clique problem. *Discrete Applied Mathematics* 161, 1–2 (2013), 244–257.

[60] Bryan Perozzi and Leman Akoglu. 2016. Scalable anomaly ranking of attributed neighborhoods. In *SIAM SDM*.
[61] Bryan Perozzi, Leman Akoglu, Patricia Iglesias Sánchez, and Emmanuel Müller. 2014. Focused clustering and outlier detection in large attributed graphs. In *KDD*. 1346–1355.
[62] Bryan Perozzi, Rami Al-Rfou', and Steven Skiena. 2014. DeepWalk: Online learning of social representations. In *KDD*. 701–710.
[63] Robert Pienta, James Abello, Minsuk Kahng, and Duen Horng Chau. 2015. Scalable graph exploration and visualization: Sensemaking challenges and opportunities. In *BigComp*. IEEE Computer Society, 271–278.
[64] Robert Pienta, Minsuk Kahng, Zhiyuan Lin, Jilles Vreeken, Partha Talukdar, James Abello, Ganesh Parameswaran, and Duen Horng Chau. 2017. FACETS: Adaptive local exploration of large graphs. In *SDM*.
[65] Zijie Qi and Ian Davidson. 2009. A principled and flexible framework for finding alternative clusterings. In *KDD*. 717–726.
[66] Eunsu Ryu, Yao Rong, Jie Li, and Ashwin Machanavajjhala. 2013. Curso: Protect yourself from curse of attribute inference: A social network privacy-analyzer. In *DBSocial*. 13–18.
[67] Hiroaki Shiokawa, Yasuhiro Fujiwara, and Makoto Onizuka. 2013. Fast algorithm for modularity-based graph clustering. In *AAAI*.
[68] Arlei Silva, Wagner Meira Jr., and Mohammed J. Zaki. 2012. Mining attribute-structure correlated patterns in large attributed graphs. *PVLDB* 5, 5 (2012), 466–477.
[69] Daniel A. Spielman and Shang-Hua Teng. 2004. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC*. 81–90.
[70] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale information network embedding. In *WWW*. 1067–1077.
[71] Yuanyuan Tian, Richard A. Hankins, and Jignesh M. Patel. 2008. Efficient aggregation for graph summarization. In *SIGMOD*. 567–580.
[72] Amanda L. Traud, Peter J. Mucha, and Mason A. Porter. 2012. Social structure of Facebook networks. *Physica A: Statistical Mechanics and its Applications* 391, 16 (2012), 4165–4180.
[73] Charalampos E. Tsourakakis, Francesco Bonchi, Aristides Gionis, Francesco Gullo, and Maria A. Tsiarli. 2013. Denser than the densest subgraph: Extracting optimal quasi-cliques with quality guarantees. In *KDD*.
[74] Charalampos E. Tsourakakis, U. Kang, Gary L. Miller, and Christos Faloutsos. 2009. DOULION: Counting triangles in massive graphs with a coin. In *KDD*. 837–846.
[75] Tatiana von Landesberger, Arjan Kuijper, Tobias Schreck, Jrn Kohlhammer, Jarke J. van Wijk, Jean-Daniel Fekete, and Dieter W. Fellner. 2011. Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum* 30, 6 (2011), 1719–1749.
[76] Joyce Jiyoung Whang, David F. Gleich, and Inderjit S. Dhillon. 2013. Overlapping community detection using seed set expansion. In *CIKM*. 2099–2108.
[77] Jierui Xie, Stephen Kelley, and Boleslaw K. Szymanski. 2013. Overlapping community detection in networks: The state-of-the-art and comparative study. *ACM Computing Surveys* 45, 4 (2013), 43.
[78] Jaewon Yang and Jure Leskovec. 2012. Community-affiliation graph model for overlapping network community detection. In *ICDM*. 1170–1175.
[79] Jaewon Yang and Jure Leskovec. 2012. Defining and evaluating network communities based on ground-truth. In *ICDM*. 745–754.
[80] Jaewon Yang and Jure Leskovec. 2013. Overlapping community detection at scale: A nonnegative matrix factorization approach. In *WSDM*. 587–596.
[81] Jaewon Yang, Julian J. McAuley, and Jure Leskovec. 2013. Community detection in networks with node attributes. In *ICDM*. 1151–1156.
[82] Shengqi Yang, Yanan Xie, Yinghui Wu, Tianyi Wu, Huan Sun, Jian Wu, and Xifeng Yan. 2014. SLQ: A user-friendly graph querying system. In *SIGMOD*. 893–896.
[83] Ning Zhang, Yuanyuan Tian, and Jignesh M. Patel. 2010. Discovery-driven graph summarization. In *ICDE*. 880–891.
[84] Yang Zhang and Srinivasan Parthasarathy. 2012. Extracting, analyzing and visualizing triangle k-core motifs within networks. In *ICDE*. 1049–1060.
[85] Elena Zheleva and Lise Getoor. 2009. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *WWW*. 531–540.
[86] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. 2009. Graph clustering based on structural/attribute similarities. *Proceedings of the VLDB Endowment* 2, 1 (2009), 718–729.

Received September 2016; revised June 2017; accepted August 2017