Attributed Community Analysis: Global and Ego-centric Views

Xin Huang[†], Hong Cheng[‡], Jeffrey Xu Yu[‡]
[†] *University of British Columbia,* [‡]*The Chinese University of Hong Kong*
xin0@cs.ubc.ca, {hcheng,yu}@se.cuhk.edu.hk

Abstract

The proliferation of rich information available for real world entities and their relationships gives rise to a type of graph, namely attributed graph, where graph vertices are associated with a number of attributes. The set of an attribute can be formed by a series of keywords. In attributed graphs, it is practically useful to discover communities of densely connected components with homogeneous attribute values. In terms of different aspects, the community analysis tasks can be categorized into global network-wide and ego-centric personalized. The global network-wide community analysis considers the entire network, such that community detection, which is to find all communities in a network. On the other hand, the ego-centric personalized community analysis focuses on the local neighborhood subgraph of given query nodes, such that community search. Given a set of query nodes and attributes, community search in attributed graphs is to locally detect meaningful community containing query-related nodes in the online manner. In this work, we briefly survey several state-of-the-art community models based on various dense subgraphs, meanwhile also investigate social circles, that one special kind of communities are formed by friends in 1-hop neighborhood network for a particular user.

1 Introduction

Nowadays with rich information available for real world entities and their relationships, graphs can be built in which vertices are associated with a set of attributes describing the properties of the vertices. The attributed graphs exist in many application domains such as web, social networks, collaboration networks, biological networks and communication networks and so on. Community(cluster), as a group of densely inter-connected nodes sharing similar properties, naturally exists in real-world networks [24]. In this work, we investigate communities in two aspects of global network-wide and ego-centric personalized. From the global network-wide analysis, we study the task of community detection that is to identify all communities in a network [13, 17, 23]. On the other hand, in the ego-centric personalized community analysis, we studied the problem of community search that is to find meaningful communities containing query-related nodes in local subgraph. Since the communities defined by different nodes in a network may be quite different, community search with query nodes opens up the prospects of user-centered and personalized search, with the potential of the answers being more meaningful to a user[9]. Recently, several papers [19, 9, 11, 22, 15, 5, 4, 1] have studied community search on graph structure for ego-centric personalized community analysis.

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

In Section 2, we focus on community detection in attributed graphs. For discovering all communities in attributed graph, [24, 25, 2] model the problem as graph clustering, which aims to partition the graph into several densely connected components with homogeneous attribute values. We proposed a novel graph clustering algorithm, SA-Cluster, which combines structural and attribute similarities through a unified distance measure. SA-Cluster finds all clusters by considering the full attribute space. However, in high-dimensional attributed graphs[10], the high-dimensional clusters are hard to interpret, or there is even no significant cluster with homogeneous attribute values in the full attribute space. If an attributed graph is projected to different attribute subspaces, various interesting clusters embedded in subspaces can be discovered. Therefore, based on the unified distance measure, we extend the method of SA-Cluster to propose a novel cell-based algorithm SCMAG to discover clusters embedded in subspaces, with similar attribute values and cohesive structure[10].

In Section 3, we focus on community search in attributed graphs. Unlike community detection, community search focus on the local neighborhood of given query-related nodes. Given a set of query nodes and attributes, community search on attribute graph is to detect a densely inter-connected communities containing all required query nodes and attributes in the online manner. First, we introduce one of best known query applications on attribute graph as team formation [12, 14, 6]. Team formation is to find a group of individuals satisfying all skilled required in a task with low communication cost. Then we show how to generalize the problem of team formation into community search. Next, we briefly summarize several community models based on various dense subgraphs, such as quasi-clique[4], densest subgraph[22], k-core[19, 15, 5, 1] and k-truss[9, 11]. Finally, we investigate social circles, and analyze its power in social contagion. In social network, for a particular user, social circles are defined as communities in her 1-hop neighborhood network, a network of connections between her friends. The structure of social circles can be modeled as connected component, k-core and k-truss. [20] shows the probability of contagion in social contagion process is tightly controlled by the number of social circles.

2 Community Detection on Attributed Graphs

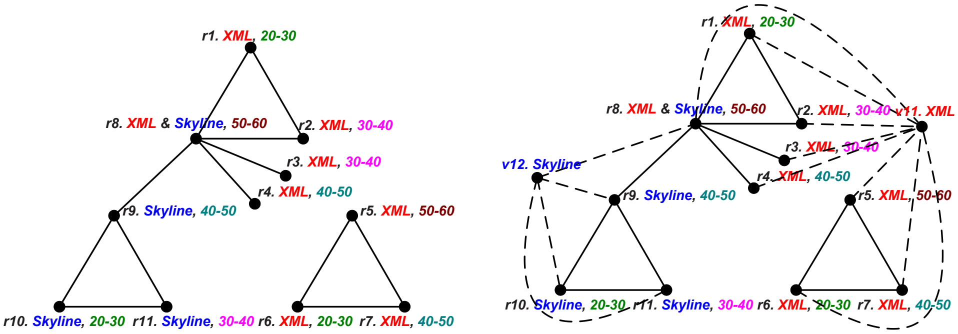
In this section, we study the community detection on attributed graphs, under the semantics of both full attribute space and attribute subspace. We first formulate the problem of graph clustering on attributed graphs by considering both structural connectivity and attribute similarities. Then, we design a unified distance measure to combine structural and attribute similarities. Finally, we briefly review the key ideas of community detection algorithms, as SA-Cluster for graph clustering on full space attributes [24] and SCMAG for graph subspace clustering[10].

2.1 Attributed Graphs

An undirected, unweighted simple graph is represented as $G = (V, E)$ with $|V|$ vertices and $|E|$ edges. When the vertices are associated with attributes, the network structure can be modeled as a new type of attributed graph as follow.

Definition 1 (Attributed Graph): An attributed graph is denoted as $G = (V, E, \Lambda)$, where V is the set of vertices, E is the set of edges, and $\Lambda = \{a_1, \ldots, a_m\}$ is the set of attributes associated with vertices in V for describing vertex properties. A vertex $v \in V$ is associated with an attribute vector $[a_1(v), \ldots, a_m(v)]$ where $a_j(v)$ is a set of attribute values of vertex v on attribute a_j.

Figure 1 shows an example of a coauthor graph where a vertex represents an author and an edge represents the coauthor relationship between two authors. In addition, there are an author ID, research topic and age range associated with each author, which are considered as attributes to describe the vertex properties. For example, the author r_8 works on two topics of XML and Skyline. The problem of community detection is to find all



(a) Attributed Graph

(b) Attribute Augmented Graph

Figure 1: A Coauthor Network with Two Attributes "Research Topic" and "Age Range"

communities on the attributed graph, such as the example in Figure 1(a), based on both structural and attribute similarities. Therefore, we formulate the problem as the graph clustering on attributed graph in the following. **Attributed graph clustering** is to partition an attributed graph G into k disjoint subgraphs $\{G_i = (V_i, E_i, \Lambda)\}_{i=1}^{k}$, where $V = \bigcup_{i=1}^{k} V_i$ and $V_i \bigcap V_j = \emptyset$ for any $i \neq j$. A desired clustering of an attributed graph should achieve a good balance between the following two objectives: (1) vertices within one cluster are close to each other in terms of structure, while vertices between clusters are distant from each other; and (2) vertices within one cluster have similar attribute values, while vertices between clusters could have quite different attribute values.

2.2 Attribute Augmented Graph

In the following, we used an *attribute augmented graph* to represent attributes explicitly as *attribute vertices and edges* proposed by [24].

Definition 2 (Attribute Augmented Graph): Given an attributed graph $G = (V, E, \Lambda)$ with a set of attributes $\Lambda = \{a_1, \ldots, a_m\}$. The domain of attribute a_i is $Dom(a_i) = \{a_{i1}, \ldots, a_{in_i}\}$ with a size of $|Dom(a_i)| = n_i$. An attribute augmented graph is denoted as $G_a = (V \cup V_a, E \cup E_a)$ where $V_a = \{v_{ij}\}_{i=1,j=1}^{m,\ n_i}$ is the set of attribute vertices and $E_a \subseteq V \times V_a$ is the set of attribute edges. An attribute vertex $v_{ij} \in V_a$ represents that attribute a_i takes the j^{th} value. An attribute edge $(v_i, v_{jk}) \in E_a$ iff $a_{jk} \in a_j(v_i)$, i.e., vertex v_i takes the value of a_{jk} on attribute a_j. Accordingly, a vertex $v \in V$ is called a structure vertex and an edge $(v_i, v_j) \in E$ is called a structure edge.

Figure 1(b) is an attribute augmented graph on the coauthor network example. Two attribute vertices v_{11} and v_{12} representing the topics "XML" and "Skyline" are added. Authors with corresponding topics are connected to the two vertices respectively in dashed lines. We omit the attribute vertices and edges corresponding to the age attribute, for the sake of clear presentation. In the attributed graph clustering problem, we need to discuss two main issues: (1) a distance measure, and (2) a clustering algorithm below.

2.3 A Unified Random Walk Distance

We use the neighborhood random walk model on the attribute augmented graph G_a to compute a unified distance between vertices in V. The random walk distance between two vertices $v_i, v_j \in V$ is based on the paths consisting of both structure and attribute edges. Thus it effectively combines the structural proximity and attribute similarity of two vertices into one unified measure. The transition probability matrix P_A on G_a is defined as follows.

A structure edge $(v_i, v_j) \in E$ is of a different type from an attribute edge $(v_i, v_{jk}) \in E_a$. The m attributes in Λ may also have different importance. Therefore, they may have different degree of contributions in random walk distance. Without loss of generality, we assume that a structure edge has a weight of ω_0, attribute edges corresponding to a_1, a_2, $\ldots$, a_m have an edge weight of ω_1, ω_2, $\ldots$, ω_m, respectively. In the following, we will define the transition probabilities between two structure vertices, between a structure vertex and an attribute vertex, and between two attribute vertices. First, the transition probability from a structure vertex v_i to another structure vertex v_j through a structure edge is

$$p_{v_i,v_j} = \begin{cases} \dfrac{\omega_0}{|N(v_i)| * \omega_0 + \omega_1 + \ldots + \omega_m}, & if(v_i, v_j) \in E \\ 0, & otherwise \end{cases} \tag{1}$$

where $N(v_i)$ represents the set of structure vertices connected to v_i.

The transition probability from a structure vertex v_i to an attribute vertex v_{jk} through an attribute edge is

$$p_{v_i,v_{jk}} = \begin{cases} \dfrac{\omega_j}{|N(v_i)| * \omega_0 + \omega_1 + \ldots + \omega_m}, & if(v_i, v_{jk}) \in E_a \\ 0, & otherwise \end{cases} \tag{2}$$

The transition probability from an attribute vertex v_{ik} to a structure vertex v_j through an attribute edge is

$$p_{v_{ik},v_j} = \begin{cases} \dfrac{1}{|N(v_{ik})|}, & if(v_{ik}, v_j) \in E_a \\ 0, & otherwise \end{cases} \tag{3}$$

The transition probability between two attribute vertices v_{ip} and v_{jq} is 0 as there is no edge between attribute vertices.

$$p_{v_{ip},v_{jq}} = 0, \forall v_{ip}, v_{jq} \in V_a \tag{4}$$

The transition probability matrix P_A is a $|V \cup V_a| \times |V \cup V_a|$ matrix, where the first $|V|$ rows and columns correspond to the structure vertices and the rest $|V_a|$ rows and columns correspond to the attribute vertices. For the ease of presentation, P_A is represented as

$$P_A = \begin{bmatrix} P_{V_1} & A_1 \\ B_1 & O \end{bmatrix} \tag{5}$$

where P_{V_1} is a $|V| \times |V|$ matrix representing the transition probabilities defined by Equation (1); A_1 is a $|V| \times |V_a|$ matrix representing the transition probabilities defined by Equation (2); B_1 is a $|V_a| \times |V|$ matrix representing the transition probabilities defined by Equation (3); and O is a $|V_a| \times |V_a|$ zero matrix.

Definition 3 (Random Walk Distance Matrix): Let P_A be the transition probability matrix of an attribute augmented graph G_a. Given L as the length that a random walk can go, $c \in (0, 1)$ as the random walk restart probability, the unified neighborhood random walk distance matrix R_A is

$$R_A = \sum_{l=1}^{L} c(1-c)^l P_A^l \tag{6}$$

2.4 SA-Cluster Algorithm

SA-Cluster adopts the *K-Medoids* clustering framework. After initializing the cluster centroids and calculating the random walk distance at the beginning of the clustering process, it repeats the following four steps until convergence.

1. Assign vertices to their closest centroids;
2. Update cluster centroids;
3. Adjust attribute edge weights $\{\omega_1, \ldots, \omega_m\}$;
4. Re-calculate the random walk distance matrix R_A.

Different from traditional K-Medoids, SA-Cluster has two additional steps (i.e., steps 3-4): in each iteration, the attribute edge weights $\{\omega_1, \ldots, \omega_m\}$ are automatically adjusted to reflect the clustering tendencies of different attributes. Interested readers can refer to [24] for the proposed mechanism for weight adjustment.

The time complexity of SA-Cluster is O($t \cdot L \cdot |V \cup V_a|^3$), where t is the number of iterations in the clustering process, and $O(L \cdot |V \cup V_a|^3)$ is the cost of computing the random walk distance matrix R_A. In order to improve the efficiency and scalability of SA-Cluster, [25] proposes an efficient algorithm Inc-Cluster to incrementally update the random walk distances given the edge weight increments. Complexity analysis shows that Inc-Cluster can improve SA-Cluster by approximately t times. For further speed up Inc-Cluster, [2] designs parallel matrix computation techniques on a multicore architecture.

2.5 Subspace Clustering in High-dimensional Attributed Graphs

Although SA-Cluster can differentiates the importance of attributes with an attribute weighting strategy, it cannot get rid of irrelevant attributes completely, especially when the dimension of attribute is high, i.e., $|\Lambda| = m$ is large. The high-dimensional clusters are hard to interpret, or there is even no significant cluster with homogeneous attribute values in the full attribute space. If an attributed graph is projected to different attribute subspaces, various interesting clusters embedded in subspaces can be discovered which, however, may not exhibit in the full attribute space. In the following, we will study the problem of subspace clustering in high-dimensional attributed graphs. We first define the subspace criterion of good subspace clusters in terms of homogeneous properties and cohesive structure. Then, we propose a novel cell-based subspace clustering algorithm SCMAG.

2.5.1 Criterion of Subspace Clusters

For the discovered clusters embedded in subspaces, should not only have homogeneous attribute values, but also have dense connections, i.e., correspond to communities with homogeneous properties and cohesive structure.
Attribute Criterion. Given a attribute subspace $\mathcal{S} \subseteq \Lambda$, the subspace entropy and interest are defined as follows.

Definition 4 (Subspace Entropy): Given a set of attributes $\mathcal{S} = \{a_1, \ldots, a_k\} \subseteq \Lambda$, the subspace entropy of $\mathcal{S}$ is defined as

$$H(a_1, \ldots, a_k) = - \sum_{A_1 \in Dom(a_1)} \cdots \sum_{A_k \in Dom(a_k)} p(A_1, \ldots, A_k) \log p(A_1, \ldots, A_k) \quad (7)$$

where $p(A_1, \ldots, A_k)$ is the percentage of graph vertices whose attribute value vector is $[A_1, \ldots, A_k]$.

In addition, we want the attributes of a subspace to be correlated. If the attributes are independent of each other, the subspace does not give more information than looking at each attribute independently. We measure the correlation of a subspace $\mathcal{S}$ using mutual information between all individual dimensions of the subspace as below.

$$I(\{a_1, \ldots, a_k\}) = \sum_{i=1}^{k} H(a_i) - H(a_1, \ldots, a_k)$$

We consider a subspace $\mathcal{S} = \{a_1, \ldots, a_k\}$ as an interesting subspace, if $\mathcal{S}$ is more strongly correlated than any of its subsets $\mathcal{S}' \subseteq \mathcal{S}$. To measure the increase in correlation of a subspace, we define the *interest* of a subspace.

Definition 5 (Subspace Interest): Given a set of attributes $\mathcal{S} = \{a_1, \ldots, a_k\} \subseteq \Lambda$, the subspace interest of $\mathcal{S}$ is defined as the minimum increase in correlation of $\mathcal{S}$ over its $(k-1)$-dimensional subsets.

$$interest(a_1, \ldots, a_k) = I(\{a_1, \ldots, a_k\}) - \max_i I(\{a_1, \ldots, a_k\} - \{a_i\})$$

Therefore, a good subspace for clustering should have low subspace entropy and high subspace interest.

Structural Criterion. Given a subspace $\mathcal{S} = \{a_1, \ldots, a_k\}$ and each attribute a_i has n_i values, the k-dimensional space is partitioned to form a grid. The vertices with same attribute vector fall into the same cell of grid, under this k-dimensional space. Thus, for a good space for clustering, we identify the cells with high coverage and connectivity, according to the following definition.

Definition 6 (Coverage and Connectivity): Given a cell u in a subspace, the coverage of u is measured by the number of vertices in u, i.e., $V(u) = |u|$. The connectivity of u is measured by the sum of random walk scores of all pairs of vertices, divided by the cell size

$$D(u) = \frac{\sum_{v_i, v_j \in u} \widetilde{Q}_{VV}(v_i, v_j)}{|u|},$$

where $\widetilde{Q}_{VV}(v_i, v_j)$ is the normalized structural similarity between v_i and v_j.

2.5.2 A review of SCMAG

Based on the criteria for interesting subspace with good clustering tendency and coverage subspace with dense connectivity, the cell-based algorithmic framework of SCMAG is described as follow. We will first find the subspaces with good clustering tendency, and then identify cells in the subspace with high coverage and high connectivity. Adjacent qualified cells will be merged to form a maximal cluster in the subspace.

Follow by the framework of SA-Cluster, we first construct the attribute augmented graph by Definition 2. Then, we use the random walk with restart to unify the structural closeness and attribute similarity into a single measure. Based on the random walk score, we design a novel cell combining strategy on dimensions of attributes. Moreover, to distinguish the multi-values in an attribute, we choose one attribute value with the largest attribute similarity between the value and vertex as the unique one. Thus, each vertex is associated with an attribute vector containing a single value in each attribute. Finally, we iteratively find subspace with low subspace entropy and high subspace interest, and detect clusters by merging adjacent dense cells to satisfy high coverage and dense connectivity. The entire procedure is shown as follow.

1. Construct the attribute augmented graph, and calculate the random walk distance;

2. Identify similar attribute values to be adjacent;

Table 1: Clusters in attribute subspace {Citation, H-index, G-index, Venue} on bibliographic graph, where each vertex represents an author and an edge represents the author collaboration. Each author has 12 attributes, such as Topic, Citation, H-index, Sociability and so on[10].

Cluster 1	**Cluster 2**	**Cluster 3**	**Cluster 4**
Database	**Software Engineering &**	**Hardware &**	**Algorithms &**
	Scientific Computing	**Architecture**	**Theory**
Rakesh Agrawal	C.A.R. Hoare	A. L. Sangiovanni-Vincentelli	Rajeev Motwani
Hector Garcia-Molina	Leslie Lamport	Sharad Malik	Robert E. Tarjan
Jeffrey D. Ullman	Thomas A. Henzinger	Sartaj K. Sahni	Christos Papadimitriou
Jennifer Widom	Rajeev Alur	Lothar Thiele	Prabhakar Raghavan
Christos Faloutsos	David Harel	Sudhakar M. Reddy	David R. Karger
Jim Gray	Joseph Halpern	Jason Cong	Richard M. Karp
David J. DeWitt	Amir Pnueli	Robert Brayton	Jon M. Kleinberg
Michael Stonebraker	Moshe Vardi	Miodrag Potkonjak	Leslie Valiant
Ramakrishnan Srikant	Edmund Clarke	Massoud Pedram	Oded Goldreich
Serge Abiteboul	Robin Milner	Janak H. Patel	Moni Naor

3. Assign vertices into cells of the grid by handling multi-valued attributes;

4. Find good subspaces with low subspace entropy and high subspace interest;

5. Find clusters in the identified subspace by merging adjacent dense cells to satisfy high coverage and dense connectivity;

Case study. Table 1 shows that SCMAG discovers 4 clusters from different research fields on bibliographic graph in the subspace {*Citation, H-index, G-index, Venue*}, and list 10 representative authors in each cluster. The subspace combination of *Citation, H-index* and *G-index* is interesting, as these three attributes are positively correlated – H-index and G-index are computed from citations.

3 Community Search on Attributed Graphs

Given a set of query nodes and attributes, community search on attributed graphs is to detect meaningful community containing query nodes and satisfying attribute constraints in the online manner. As an ego-centric personalized analysis, community search is different from community detection, which focuses on the local neighborhood subgraph of query-related nodes. In the following, we first introduce one of best known query application on attributed graphs as team formation, and show how to generalize it into community search on attributed graphs. Then, we will discuss several state-of-the-art community models based on various dense subgraphs, including special community models of social circles.

3.1 Team Formation

Task-driven Team formation [14]. Assume that in attributed graph $G(V, E, \Lambda)$, each vertex is associated with different skill attributes. Given a task T that requires a set of skills, the problem of team formation is to find a group of individuals $X \subseteq V$ who can function as a team to accomplish task T, such that every required skill in T is exhibited by at least one individual in X. Additionally, the members of team X should define a subgraph or a tree in G with low communication cost. The communication cost measures how effectively the team members can collaborate: the lower the communication cost, the better the quality of the team. [14] measures team communication cost in terms of diameter or spanning tree. We formulate the problem of diameter based team formation as below.

Definition 7 (Graph Diameter): The diameter of a graph G is defined as the maximum length of a shortest path in G, i.e., $\mathsf{diam}(G) = \max_{u,v \in G}\{\mathsf{dist}_G(u, v)\}$, where $\mathsf{dist}_G(u, v)$ is the length of a shortest path between u and v in G.

Definition 8 (Diameter based Team Formation): Given an attributed graph $G(V, E, \Lambda)$ and a task $T = \{w_1, ..., w_k\} \subseteq \bigcup_{a \in \Lambda} Dom(a)$, find a subgraph $H \subseteq G$ such that satisfies

1. $\forall w \in T, \exists v \in H$ and $a \in \Lambda, s.t, w \in a(v)$;
2. $\mathsf{diam}(H)$ is minimized.

This problem has been shown to be NP-complete. However, there exists a 2-approximation algorithm, which can find a subgraph H that satisfies all required skills and has the diameter no greater than 2 times of the optimal one.

3.2 A Formulation of Community Search

In the diameter based team formation, the diameter metric may not measure the communication cost well, because this simple function is instability: a slight change in the graph may result in a radical change in the solution, due to the weak connectivity[6]. Therefore, to enforce the dense connectivity constraints on the formed team is necessary. On the other hand, in some application scenarios, we may need to specify leaders in a team, since leaders need to iteratively communicate with each team member to monitor and coordinate the project[12]. Thus, the given leaders(vertices) must be contained in the reported team. As a result, we can generalize team formation with leader constraints into the problem of diverse attributed community search on attributed graph as follow.

Definition 9 (Diverse Attributed Community Search): Given an attributed graph $G(V, E, \Lambda)$, a set of attribute values $T = \{w_1, ..., w_k\} \subseteq \bigcup_{a \in \Lambda} Dom(a)$ and a set of query nodes $Q \subseteq V(G)$, find a connected subgraph $H \subseteq G$ such that satisfies

1. $Q \subseteq V(H)$;
2. $\forall w \in T, \exists v \in H$ and $a \in \Lambda, s.t, w \in a(v)$;
3. H is densely connected, and the communication cost is minimum.

As we can see, the problem of diverse attributed community search tends to find a densely connected subgraph containing all query nodes and achieving the coverage of diverse attributes, with the minimum communication cost. In Definition 9, either a set of attributes T or a set of query nodes Q can be empty. If the set of attributes are empty as $T = \emptyset$, the problem of community search on attributed graph is equivalent to the problem of find densely connected community in a simple graph $G(V, E)$. If the set of query nodes are empty as $Q = \emptyset$, the problem of community search on attributed graph is equivalent to the problem of team formation without leader constraints in Definition 8.

3.3 Dense Subgraph based Community Models

In this section, we will introduce several novel community models in a simple graph $G(V, E)$ without attributes. These state-of-the-art community models are based on different dense subgraph definitions, such as quasi-clique[4], densest subgraph[22], k-core[19, 15, 5, 1] and k-truss[9, 11]. These community models can be further developed and extended for applying on attributed graph $G(V, E, \Lambda)$.

Quasi-Clique Community. Cui et. al[4] propose a α-adjacency-γ-quasi-k-clique community model. A γ-quasi-k-clique of a simple graph G is defined as a k-node subgraph of G with at least $\lfloor \gamma \frac{k(k-1)}{2} \rfloor$ edges, where $0 \leq \gamma \leq 1$. A γ-quasi-k-clique is a relaxation of a k-clique. Two γ-quasi-k-cliques are α-adjacent and can be union if they share at least α common vertices. Given a query node, the community search problem is to find all α-adjacency-γ-quasi-k-cliques containing it. Several heuristic approaches are proposed for speed up the NP-hard query processing.

Query-biased Densest Subgraph Community. Wu et al. [22] studied the query biased densest connected subgraph (QDC) problem for avoiding subgraphs irrelevant to query nodes in the discovered community. The community is defined based on a connected graph containing given query nodes, and it optimizes a fundamentally different function called query biased edge density, which is calculated as the overall edge weight averaged over the weight of nodes in a community.

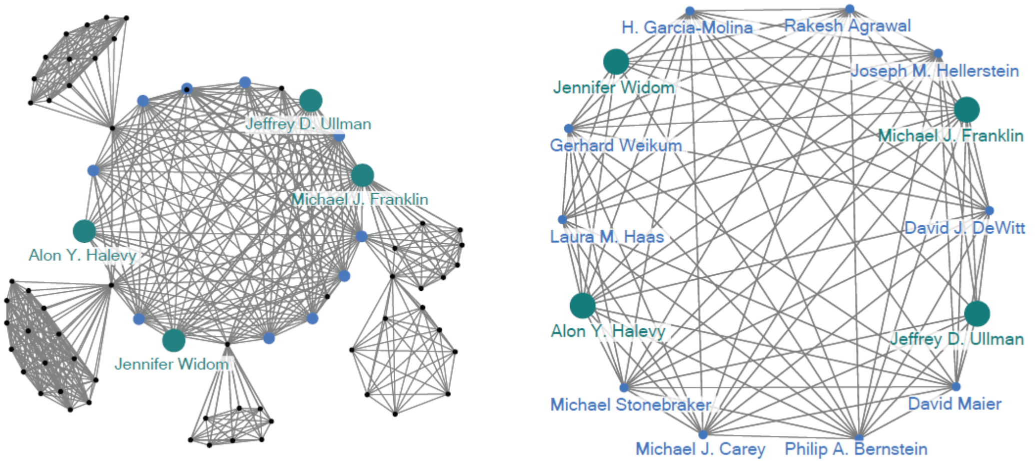
K**-core Community.** Several community models build up on the structure of k-core [19, 15, 5, 1]. A k-core is a subgraph of G that requires each node has at least k neighbors within this subgraph [18]. Sozio et al. [19] proposed a k-core based community model, called Cocktail Party, with the distance and size constraints. Cocktail Party community model finds the k-core with largest k as the density optimization, and uses the furthest query distance as the communication cost function. Cui et al. [5] find a k-core community for a query node using local search. In addition, Li et al. [15] propose influential community model that finds top-r communities with the highest influence scores over the entire graph, without considering query nodes.

K**-truss Community.** A k-truss is a subgraph of G that requires each edge be contained at least (k-2) triangles within this subgraph [3]. In a social network, a triangle indicates two friends have a common friend, which shows a strong relationship among three friends. Intuitively, the more common friends two people have, the stronger their relationship. In a k-truss, each pair of friends is "endorsed" by at least (k-2) common friends[11]. Thus, a k-truss with a large value of k signifies strong inner-connections between members of the subgraph. The community proposed by [9] and [11] both are build upon the connected k-truss. [9] proposes a k-truss community model based on triangle adjacency, to find all overlapping communities of one query node. The closest truss community [11] aims to find a connected k-truss subgraph with the largest k that contains Q, and has the minimum diameter among such subgraphs. Here, the minimum graph diameter is used as the communication cost constraint. In comparison of the k-core community and the k-truss community, conceptually, k-truss is a more cohesive definition than k-core, as k-truss is based on triangles whereas k-core simply considers node degree.

Case study. Figure 2(b) shows a closest truss community [11] detected on DBLP network using the query $Q = \{$"Alon Y. Halevy", "Michael J. Franklin", "Jeffrey D. Ullman", "Jennifer Widom"$\}$ and $T = \emptyset$. It has 14 authors, 81 edges and the edge density of 0.89. The community does not include any authors in a 9-truss [3] that are far away from and loosely connected with queried authors in Figure 2(a), which shows the superiority of closest truss community.

3.4 Social Circles and Social Contagion

In this section, we will study one special kind of community in social networks as social circles. For one query user, social circles are communities in query users 1-hop neighborhood network, a network of connections between her friends. Simply, in terms of graph structure, for a user with a small number of friends, a connected component is strongly enough to represent a social circle; Whereas, for a user with a large number of friends,



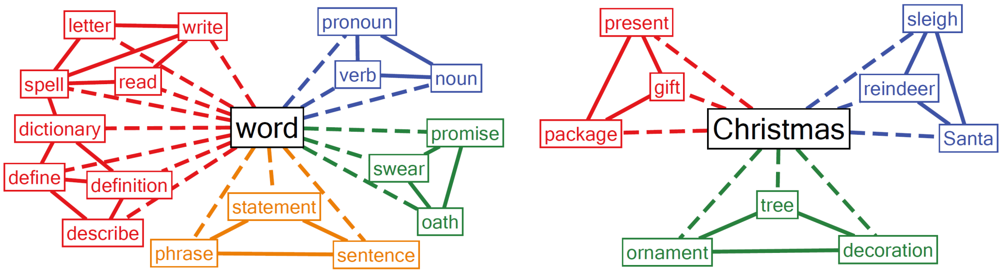
(a) 9-truss [3]

(b) closest truss community [11]

Figure 2: Community search on DBLP network without attributes using query $Q =$ {"Alon Y. Halevy", "Michael J. Franklin", "Jeffrey D. Ullman", "Jennifer Widom"} and $T = \emptyset$

since the structure of her neighborhood network becomes complex, a connected k-core and a connected k-truss as cohesive structure are much better social circle models. Interested readers can refer to more community models on attribute graphs [16, 21].

In the following, we will show how these social circles affects the process of information diffusion on social contagion. Ugander et al. [20] study two social contagion processes in Facebook: the process that a user joins Facebook in response to an invitation email from an existing Facebook user, and the process that a user becomes an engaged user after joining. They find that the probability of contagion is tightly controlled by the number of social circles in a users neighborhood, rather than by the number of friends in the neighborhood. A social circle represents a distinct social context of a user, and the multiplicity of social contexts is termed structural diversity [20]. A user is much more likely to join Facebook and become engaged if he or she has a larger structural diversity, i.e., a larger number of distinct social contexts. [7, 8] studied the problem of find k users with the highest structural diversity in graphs, which can be beneficial to a wide range of application domains, for example, political campaign, the promotion of health practices, marketing, and so on.



(a) Word

(b) Christmas

Figure 3: Top-2 structural diversity based on connected 2-core in word association network. Here "word" and "Christmas" respectively has the top-2 highest structural diversity score as 4 and 3.

Case study. Figure 3 shows that top-2 results in the word association network using connected 2-core as the structural bone of social circles. Two words "word" and "Christmas" have the highest two structural diversity scores of 4 and 3. As we can see, in Figure 3, each vertex in the 2-core component has at least two neighbor words. Specifically, the word "word" in Figure 3 (a) has 4 distinct contexts of associated words with different meanings. For example, {"swear", "oath", "promise"} represent the synonym of "words" as "promise", and {"verb", "noun", "pronoun"} are different types of "word". The word "Christmas" has three distinct contexts of associated words, as shown in Figure 3 (b), {"reindeer", "sleigh", "Santa"} describe the "Santa", {"present", "gift", "package"} represent the "Christmas gifts" and {"tree", "ornament", "decoration"} are related to the "Christmas tree".

4 Future Work and Conclusion

In this paper, we study two problems of community detection and community search in attributed networks, respectively in terms of global network-wide analysis and ego-centric personalized analysis aspects. For community detection, we design a unified distance measure to combine structural and attribute similarities on attribute graphs. Based on that, we propose two community detection algorithms SA-Cluster and SCMAG for respectively considering the full space and subspace of attributes. For community search, we give a formal problem definition of community search on attributed graphs by generalizing from the problem of team formation. Several dense subgraph based community model are surveyed here for a comparison. Since all these dense subgraph based community models only consider structures in simple graphs without attributes, it would be interesting to extend the models and algorithms to attributed graphs for community search. Given the recent surge of interest k-core and k-truss in probabilistic graphs, an exciting question is how k-core and k-truss models generalizes to probabilistic graphs. The challenge is to develop extensions that are widely useful and tractable.

References

[1] N. Barbieri, F. Bonchi, E. Galimberti, and F. Gullo. Efficient and effective community search. *Data Mining and Knowledge Discovery*, 29(5):1406–1433, 2015.

[2] H. Cheng, Y. Zhou, X. Huang, and J. X. Yu. Clustering large attributed information networks: an efficient incremental computing approach. *Data Mining and Knowledge Discovery*, 25(3):450–477, 2012.

[3] J. Cohen. Trusses: Cohesive subgraphs for social network analysis. Technical report, National Security Agency, 2008.

[4] W. Cui, Y. Xiao, H. Wang, Y. Lu, and W. Wang. Online search of overlapping communities. In *SIGMOD*, pages 277–288, 2013.

[5] W. Cui, Y. Xiao, H. Wang, and W. Wang. Local search of communities in large graphs. In *SIGMOD*, pages 991–1002, 2014.

[6] A. Gajewar and A. D. Sarma. Multi-skill collaborative teams based on densest subgraphs. In *SDM*, pages 165–176. SIAM, 2012.

[7] X. Huang, H. Cheng, R.-H. Li, L. Qin, and J. X. Yu. Top-k structural diversity search in large networks. *PVLDB*, 6(13):1618–1629, 2013.

[8] X. Huang, H. Cheng, R.-H. Li, L. Qin, and J. X. Yu. Top-k structural diversity search in large networks. *The VLDB Journal*, 24(3):319–343, 2015.

[9] X. Huang, H. Cheng, L. Qin, W. Tian, and J. X. Yu. Querying k-truss community in large and dynamic graphs. In *SIGMOD*, pages 1311–1322, 2014.

[10] X. Huang, H. Cheng, and J. X. Yu. Dense community detection in multi-valued attributed networks. *Information Sciences*, 314:77–99, 2015.

[11] X. Huang, L. V. Lakshmanan, J. X. Yu, and H. Cheng. Approximate closest community search in networks. *PVLDB*, 9(4):276–287, 2015.

[12] M. Kargar and A. An. Discovering top-k teams of experts with/without a leader in social networks. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 985–994. ACM, 2011.

[13] A. Lancichinetti and S. Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[14] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *KDD*, pages 467–476. ACM, 2009.

[15] R.-H. Li, L. Qin, J. X. Yu, and R. Mao. Influential community search in large networks. *PVLDB*, 8(5), 2015.

[16] J. J. McAuley and J. Leskovec. Learning to discover social circles in ego networks. In *NIPS*, pages 548–556, 2012.

[17] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133, 2004.

[18] S. B. Seidman. Network structure and minimum degree. *Social networks*, 5(3):269–287, 1983.

[19] M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In *KDD*, pages 939–948, 2010.

[20] J. Ugander, L. Backstrom, C. Marlow, and J. Kleinberg. Structural diversity in social contagion. *Proceedings of the National Academy of Sciences*, 109(16):5962–5966, 2012.

[21] Y. Wang and L. Gao. An edge-based clustering algorithm to detect social circles in ego networks. *Journal of computers*, 8(10):2575–2582, 2013.

[22] Y. Wu, R. Jin, J. Li, and X. Zhang. Robust local community detection: On free rider effect and its elimination. *PVLDB*, 8(7), 2015.

[23] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*, pages 587–596, 2013.

[24] Y. Zhou, H. Cheng, and J. X. Yu. Graph clustering based on structural/attribute similarities. *PVLDB*, 2(1):718–729, 2009.

[25] Y. Zhou, H. Cheng, and J. X. Yu. Clustering large attributed graphs: An efficient incremental approach. In *ICDM*, pages 689–698, 2010.