Mining Query-Based Subnetwork Outliers in Heterogeneous Information Networks

Honglei Zhuang*, Jing Zhang†, George Brova*, Jie Tang†, Hasan Cam‡, Xifeng Yan§, Jiawei Han*
*Department of Computer Science, University of Illinois at Urbana-Champaign
†Department of Computer Science and Technology, Tsinghua University
‡US Army Research Lab
§Computer Science Department, University of California at Santa Barbara
{hzhuang3, brova2, hanj}@illinois.edu, zhangjing12@mails.tsinghua.edu.cn,
jietang@tsinghua.edu.cn, hasan.cam.civ@mail.mil, xyan@cs.ucsb.edu

***Abstract*—Mining outliers in a heterogeneous information network is a challenging problem: It is even unclear what should be outliers in a large heterogeneous network (*e.g.*, outliers in the entire bibliographic network consisting of authors, titles, papers and venues). In this study, we propose an interesting class of outliers, *query-based subnetwork outliers*: Given a heterogeneous network, a user raises a query to retrieve a set of task-relevant subnetworks, among which, subnetwork outliers are those that significantly deviate from others (*e.g.*, outliers of author groups among those studying "topic modeling"). We formalize this problem and propose a general framework, where one can query for finding subnetwork outliers with respect to different semantics. We introduce the notion of subnetwork similarity that captures the proximity between two subnetworks by their membership distributions. We propose an outlier detection algorithm to rank all the subnetworks according to their outlierness without tuning parameters. Our quantitative and qualitative experiments on both synthetic and real data sets show that the proposed method outperforms other baselines.**

I. Introduction

Outlier detection [10], [2], that is, uncovering objects, data points, or vertices that significantly deviate from others, is a critical task in data mining, owing to its broad applications. With the advent of heterogeneous information networks, it is natural to examine the problem of outlier mining in such networks. Unfortunately, this is a challenging problem since it is even unclear what should be the outliers in a large heterogeneous network. For example, given a bibliographic heterogeneous network consisting of authors, papers, titles and venues, what type of entities, such as objects, relationships, or subnetworks, should be outliers? Even confining the outliers to authors, it is still unclear by which standard authors should be distinguished from others: by their coauthorship, or by the venues where they publish papers. Based on our reasoning, we believe a user is usually interested in studying and comparing subnetworks of certain properties. For example, a user may be interested in studying coauthor subnetworks for those who study "topic modeling" and finding unusual author subnetworks (*i.e.*, outliers) that deviate substantially from others.

This leads to a new notion of outliers, *query-based subnetwork outliers* in a heterogeneous information network. Given such a network, a user can pose query at will to retrieve a set of task-relevant subnetworks, among which, subnetwork outliers are those that significantly deviate from others (*e.g.*, outliers of author groups among those studying "topic modeling").

Notice that a subnetwork outlier is different from an individual outlier. Even if every member in a subnetwork is individually normal, the subnetwork as a whole can still be an outlier. For instance, thousands of Amazon shoppers who purchase pressure cookers are not considered as outliers. However, a group of users buying pressure cookers within a short duration, sharing the same zipcode, and/or purchasing fertilizer simultaneously, can be suspicious (*e.g.*, Boston Marathon bombers). Thus, mining such subnetwork outliers is a new, challenging problem with broad applications.

Figure 1 illustrates a motivating example of our research problem. Suppose we are given a heterogeneous travel network, which consists of numerous travelers and their booked flights and hotels. The booking relations are represented by edges in the heterogeneous network. An analyst may pose a query to check only those whose destination is Sochi (2014 Winter Olympics city), which leads to the extraction of certain flights (*e.g.*, the red solid circle in left figure), with the results of retrieved subnetworks shown in the middle figure by blue dash circles. In the right figure, a traveler subnetwork is identified as an outlier (red solid circle), since they travel all the way together but have no traceable hotel information, which significantly distinguish themselves from other subnetworks.

The query-based subnetwork outlier detection problem in heterogeneous networks poses several unique challenges. First, we need to design a flexible interaction model for users to pose queries in order to retrieve an interesting set of subnetworks. Second, we need to formalize how to judge whether a subnetwork is an outlier subnetwork, which is a fundamentally different and more challenging problem from individual outlier detection. Third, with different queries, outliers can be defined rather differently. It is challenging to design a general algorithm that can adapt itself to different queries and accurately identify the outlier subnetworks.

In this paper, we formalize the query-based subnetwork outlier detection problem in a heterogeneous network; we then introduce a new notion of subnetwork similarity to measure the proximity between two subnetworks based on their membership distribution; we finally propose an outlier detection method to calculate the outlierness and output a ranked list of subnetworks. Experimental results on both synthetic and real data sets show that the proposed method can outperform the baselines in terms of AUC (of ROC curves) and MAP.

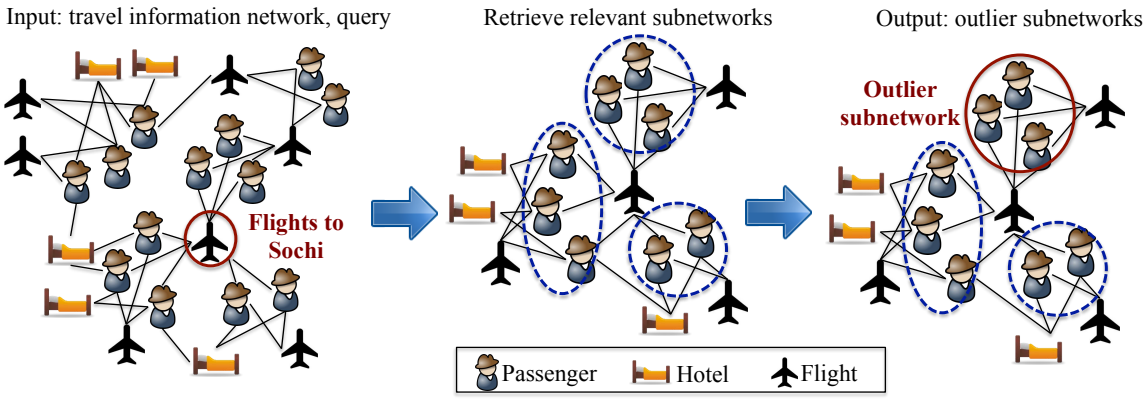

Fig. 1. Uncover potential terrorist ring by detecting subnetwork outliers in a travel information network. The left figure shows the input: a heterogeneous network containing traveler, flight and hotel information, as well as a query (find traveler subnetwork outliers flying to Sochi). The middle figure presents the candidate subnetworks, represented by blue dashed circles. The right figure illustrates a subnetwork outlier, represented by the red solid circle.

II. Related Work

Outlier detection. There has been numerous studies exploring the field of outlier detection. A number of surveys [2], [10] provide thorough summarization of existing outlier detection techniques. Network-based outlier detection is a specific type of outlier analysis. Some methods focus on detecting vertex outliers based on network structure [18], [5], [6], while others also regard the whole network as outliers [9]. Noble *et al.* [17] propose to discover the graph-based outliers based on minimum description length. There are studies on subnetwork outlier detection [16], [22]. However, they do not explore a query-based outlier detection scenario. Gupta *et al.* [8] also propose a definition of subgraph outliers in heterogeneous networks in a query-based framework, where queries can only extract isomorphic subgraphs. The query-based framework proposed in this paper is more flexible.

Network similarity. There are many different similarity or distance functions to measure the proximity of two vertices in a network. Jeh *et al.* [11] propose SimRank to recursively define a similarity measure for two vertices in a network. Similar ways to define a similarity include [14] and [12]. Sun *et al.* [19] propose PathSim for vertices in a heterogeneous information network. Some other vertex similarity measures are summarized in [13]. But it is not straightforward to extend vertex similarity measures to subnetwork similarity measures.

There are also several attempts to develop similarity measures that are capable of measuring network proximity. Eiter *et al.* [3] propose several distance measures for sets of points but not vertices in networks. Tong *et al.* [21] defines a group-to-group proximity but in a homogeneous network.

III. Overview

A. Concepts and Notations

We start with recalling the concept of a heterogeneous information network.

Definition 1: **Heterogeneous information network.** A heterogeneous information network is an information network with multiple types of vertices. Without loss of generality, it can be defined as a directed network $G = (\mathcal{V}, E; \phi, \mathcal{A})$ where $\mathcal{V}$ is the set of vertices, and E is the set of edges. There is a vertex/edge type mapping function $\phi : \mathcal{V} \to \mathcal{A}$ or $E \to \mathcal{A}$ where $\mathcal{A}$ is the set of types, *i.e.* each vertex $v \in \mathcal{V}$ or edge $e \in E$ belongs to a particular type in $\mathcal{A}$. For undirected cases, an undirected edge can be viewed as two symmetric directed edges. When there is only one type, *i.e.* $|\mathcal{A}| = 1$, the network reduces to a homogeneous information network.

A typical example is a bibliographic information network, with four types of vertices: paper (P), venue (V), author (A), and term (T). Directed edges can be defined between different types of vertices according to their relationships.

Definition 2: **Meta-path.** In a heterogeneous network G, a meta-path is an ordered sequence of vertex types, denoted by $\mathcal{P} = (T_1 T_2 \dots T_l)$, where $T_x \in \mathcal{A}$. We say an instantiation of $\mathcal{P}$ is a path in G, denoted by $p = (v_1 v_2 \dots v_l)$, satisfying $\phi(v_x) = T_x, \forall x = 1, 2, \dots, l$. In addition, we denote the set of all the path instances instantiated by meta-path $\mathcal{P}$ between vertices v_i and v_j as $\pi_{\mathcal{P}}(v_i, v_j)$.

B. Problem Definition

We define our research problem of query-based subnetwork outlier detection, and describe how can we specify a query and retrieve relevant subnetworks.

Problem 1: **Subnetwork outlier detection.** Given a heterogeneous information network $G = (\mathcal{V}, E; \phi, \mathcal{A})$ and a query q, where $\phi : \mathcal{V} \to \mathcal{A}$ specifies the type of each vertex. Our objective is to identify a set of outlier subnetworks $\mathcal{S}_\omega \subset \mathcal{S}(q)$, where $\mathcal{S}(q)$ is the set of subnetworks $\{S_i \subset \mathcal{V}\}_{i=1,\dots,k}$ relevant to q; and the subnetworks in $\mathcal{S}_\omega$ significantly deviate from subnetworks in $\mathcal{S}(q) \setminus \mathcal{S}_\omega$.

Query definition and execution. We introduce a simple but effective definition. A query consists of 1) $\mathcal{V}_q \subset \mathcal{V}$ as a set of queried vertices; 2) $T_S \in \mathcal{A}$ which indicates the vertex type of desired subnetworks; 3) meta-path $\mathcal{P}_Q$ and $\mathcal{P}_S$. $\mathcal{P}_Q$ is given to specify the semantics of query vertices, and $\mathcal{P}_S$ is given to specify the semantics of candidate vertices. For example, if one wants to find author subnetworks relevant to venue "KDD", where venues and authors can be characterized by papers, a query can be formed as $q = (\mathcal{V}_q = \{\text{"KDD"}\}, T_S = A, \mathcal{P}_Q = (VP), \mathcal{P}_S = (AP))$

We retrieve subnetworks by finding any groups of vertices that are both mutually highly connected as well as highly connected to the query vertices. To be concrete, we denote

the set of vertices reachable from v_i via paths instantiated by meta-path $\mathcal{P}$ as $\nu_{\mathcal{P}}(v_i) = \{v_j | \pi_{\mathcal{P}}(v_i, v_j) \neq \emptyset\}$. We extract all the subnetworks S satisfying

$$\left| \bigcap_{u \in \mathcal{V}_q} \nu_{\mathcal{P}_Q}(u) \bigcap_{v \in S} \nu_{\mathcal{P}_S}(v) \right| \geq \theta$$

where θ is a given threshold. We also remove all the subnetworks that are a proper subset of any other retrieved subnetworks from the results. Thereby we obtain a set of subnetworks relevant to query q, denoted by $\mathcal{S}(q)$.

In practice, this simple methodology can easily generate reasonable subnetworks relevant to the given query. Some of the results can be seen in Table II. Other methods for retrieving relevant subnetworks from a query [7] can be easily plugged in our framework without much modification.

IV. Subnetwork Similarity

In this section, we introduce a subnetwork similarity measure to examine the similarity between two subnetworks in heterogeneous networks. The intuition is, for two subnetworks, if they are composed of members from similar communities and with similar authorities, they should be regarded as similar regardless of their network size.

Balance mapping similarity (BMSim). We define the balance mapping similarity between two subnetworks S_1 and S_2. We first construct a bipartite with vertices in S_1 and S_2 and edges between every pair of $v_1^i \in S_1$ and $v_2^j \in S_2$ weighted by a certain vertex similarity of v_1 and v_2 in the original G. Here we employ PathSim [19], which is defined as:

$$PathSim_{\mathcal{P}}(v_i, v_j) = \frac{2|\pi_{\mathcal{P}}(v_i, v_j)|}{|\pi_{\mathcal{P}}(v_i, v_i)| + |\pi_{\mathcal{P}}(v_j, v_j)|} \quad (1)$$

Without loss of generality, we assume $|S_1| \geq |S_2|$. We try to map each member of S_1 to a similar member in S_2, while trying to keep the frequency of members in S_2 being mapped as balanced as possible. To be concrete, we define a balance mapping between S_1 and S_2 as a set of vertex pairs $M \subset S_1 \times S_2$, which satisfies that $\forall v_1^i \in S_1$, $\left|\{v_1^i | (v_1^i, v_2^j) \in M\}\right| = 1$, and $\forall v_2^j \in S_2$, $1 \leq \left|\{v_2^j | (v_1^i, v_2^j) \in M\}\right| < 1 + \frac{|S_1|}{|S_2|}$, *i.e.* each vertex in S_1 must be mapped to exactly one vertex in S_2, and each vertex in S_2 has to be connected to at least one but no more than a certain upper limit number of vertices in S_1. The similarity is measured by the optimal balance mapping M^* with the maximum sum of edge weights, normalized by the cardinality of S_1:

$$\sigma_{BM}(S_1, S_2) = \frac{1}{|S_1|} \max_{M} \sum_{(x_i, y_j) \in M} PathSim(v_1^i, v_2^j) \quad (2)$$

where M is a balance mapping satisfying the conditions above. To obtain the maximum weighted balance mapping, it is straightforward to convert this formulation into a minimum cost maximum flow problem by constructing a network flow graph, and solved in polynomial time [3].

Comparison with existing measures. We compare the similarity measure to other existing measures. A simple strategy to calculate subnetwork similarity, called Average Subnetwork Similarity (AvgSim), simply takes the average similarity over all the pairs of vertices from S_1 and S_2 respectively. More precisely,

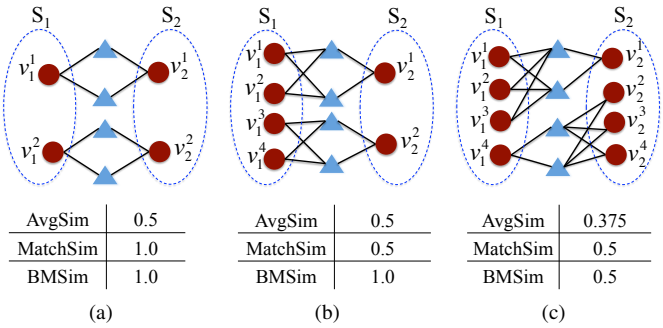

Fig. 2. Comparing different similarity measures.

$$\sigma_{Avg}(S_1, S_2) = \frac{\sum_{v_1^i \in S_1, v_2^j \in S_2} PathSim(v_1^i, v_2^j)}{|S_1| \times |S_2|} \quad (3)$$

Another subnetwork similarity that can be employed is Maximum Matching Similarity (MatchSim), adapted from [14][1]. It is calculated by finding a maximum weighted matching in a bipartite constructed in a similar way to BMSim, where a matching between S_1 and S_2 is defined as a set of pairs of vertices from each subnetwork without intersecting vertices. More precisely,

$$\sigma_{Match}(S_1, S_2) = \frac{1}{|S_1|} \max_{M'} \sum_{(x_i, y_j) \in M'} PathSim(v_1^i, v_2^j) \quad (4)$$

where M' is a matching between S_1 and S_2.

In Figure 2, we show several examples to compare all the similarity measures above. Suppose each circle represents a type of vertex (denoted as A), and each triangle represents another type of vertex (denoted as B). We use the meta-path ABA to calculate the PathSim between vertex pairs of type A. In Figure 2(a), vertices in both subnetworks share the same neighborhood, and therefore S_1 and S_2 should be considered identical. However, AvgSim compares every possible pair of vertices and yields a similarity of 0.5, which violates our intuition. In Figure 2(b), subnetwork S_1 is larger than S_2, but its membership distribution is still identical to S_2. However, MatchSim yields the similarity of 0.5, which also violates our intuition. In Figure 2(c), the two subnetworks have totally different membership distributions. Thus BMSim is calculated as 0.5, which correctly reflects our intuition.

V. Ranking Outlier Subnetworks

In this section, we introduce an algorithm to rank all the subnetworks in $\mathcal{S}(q)$ by their outlierness. The basic idea follows the clustering-based outlier detection philosophy, *i.e.* trying to cluster all the subnetworks, and those subnetworks that are less similar to any clusters have higher outlierness. The algorithm is summarized in Algorithm 1.

The clustering algorithm is similar to affinity propagation [4], but with multiple similarity measures. Suppose there

[1]Note the original paper does not use PathSim, and cannot be directly applied on heterogeneous networks. We make some slight modifications to the original MatchSim formalization.

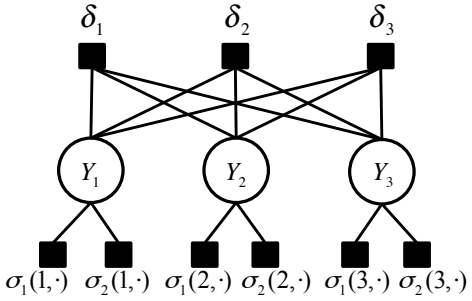

Fig. 3. An example graphical representation of the factor graph model. Random variables Y_i are represented by circles; constraint function $\delta_i(\mathbf{Y})$ and similarity measure $\sigma_m(i,\cdot)$ are represented by squares.

are in total k subnetworks $S_1, \ldots, S_k$ as input. Based on μ predefined different types of meta-paths, we can calculate μ different similarity measures between subnetworks, represented by $\sigma_1(S_i, S_j), \ldots, \sigma_\mu(S_i, S_j)$. We use random variable Y_i to represent whether subnetwork S_i is an outlier or belongs to a cluster represented by a "representative subnetwork". Each Y_i can take a value y_i from $\{0, 1, \ldots, k\}$, where $y_i = 0$ represents that S_i is an outlier subnetwork, and $1 \leq y_i \leq k$ indicates the index of the "representative subnetwork" of the cluster it belongs to. We denote the vector $(Y_1, \ldots, Y_k)$ as $\mathbf{Y}$. Our objective is to find a configuration for $\mathbf{Y}$ such that the following objective function can be maximized:

$$\mathcal{O}(\mathbf{Y}, \mathbf{w}) = \prod_{i=1}^{k} \exp\left[\sum_{m=1}^{\mu} w_m^\beta \sigma_m(i, Y_i)\right] \tag{5}$$

with the constraint that

$$\forall i : Y_i = j \Longrightarrow Y_j = j \text{ and } \sum_{m=1}^{\mu} w_m = 1$$

i.e. if any subnetwork S_j is identified by other subnetwork as a representative, it cannot recognize other subnetworks but itself as its representative. Notice that there are two special cases in the similarity function $\sigma_m(\cdot,\cdot)$. First, $\sigma_m(i,i)$ should be used to indicate how well a subnetwork serves as a representative subnetwork. Without any prior knowledge, we can simply set all the $\sigma_m(i,i)$ as the average value of elements of $\sigma_m(i,j)$ where $i \neq j$ and $(1 \leq i, j \leq k)$. Second, $\sigma_j(i,0)$ should be defined as a threshold similarity to determine outliers. $\mathbf{w} = (w_1, \ldots, w_\mu)$ is a weighting vector to leverage the importance of different similarity measures, with a given factor $0 < \beta < 1$ indicating the discrimination extent of the weighting vector. Similar formalization is also used in [1].

We take the logarithm of the objective function, and combine the first constraint into the objective function:

$$\log \mathcal{O}(\mathbf{Y}, \mathbf{w}) = \sum_i \delta_i(\mathbf{Y}) + \sum_i \sum_m w_m^\beta \sigma_m(i, Y_i)$$

where $\delta_i(\mathbf{Y}) = -\infty$ when there exists $Y_j = i$ and $Y_i \neq i$; otherwise $\delta_i(\mathbf{Y}) = 0$. Thereby we can construct a factor graph model to optimize the objective function. An example is illustrated in Figure 3 ($k = 3, \mu = 2$).

To maximize the objective function with respect to the given constraints, we optimize parameter $\mathbf{w}$ and $\mathbf{Y}$ respectively. At each iteration, we first hold the value of $\mathbf{w}$ and optimize $\mathbf{Y}$, then keep the configuration of $\mathbf{Y}$ fixed and optimize $\mathbf{w}$.

Inferring Y. We use a loopy belief propagation similar to [4] to optimize $\mathbf{Y}$. We denote messages sent from Y_i to Y_j as $r_{i\to j}$ and messages sent from Y_j to Y_i as $a_{i\leftarrow j}$. In addition, we need to introduce an auxiliary node Y_0. Messages sending from Y_i to Y_0 are denoted by $r_{i\to 0}$ and messages received by Y_i from Y_0 are $a_{i\leftarrow 0}$. All the random variables send $r_{i\to j}$ to other variables and receive $a_{i\leftarrow j}$ from other variables iteratively until the objective function converges. The updating rules can be written as:

$$r_{i\to j} = \sum_m w_m^\beta \sigma_m(i,j) - \max_{j'\neq j}\left[\sum_m w_m^\beta \sigma_m(i,j') + a_{i\leftarrow j'}\right]$$

$$a_{i\leftarrow j} = \begin{cases} 0, & j = 0 \\ \sum_{i'\neq i} \max\left(0, r_{i'\to j}\right), & j = i \\ \min\left[0, r_{j\to j} + \sum_{i'\notin\{i,j\}} \max\left(0, r_{i'\to j}\right)\right], & 0 < j \neq i \end{cases}$$

After the updating process converges, we can determine the optimal configuration for $\forall i > 0$ by:

$$\hat{Y}_i = \arg\max_{y_i}\left[a_{i\leftarrow j} + \sum_m w_m^\beta \sigma_m(i, y_i)\right]$$

$$= \begin{cases} 0, & \max_{j\neq 0}\left[a_{i\leftarrow j} + \sum_m w_m^\beta \sigma_m(i,j)\right] < \sum_m w_m^\beta \sigma_m(i,0); \\ i, & i = \arg\max_j\left[a_{i\leftarrow j} + \sum_m w_m^\beta \sigma_m(i,j)\right]; \\ \arg\max\limits_{j':\hat{Y}_{j'}=j'}\left[a_{i\leftarrow j'} + \sum_m w_m^\beta \sigma_m(i,j')\right], \text{otherwise.} \end{cases}$$

i.e. we first determine all the outlier subnetworks with $\hat{Y}_i = 0$; then we determine all the representative subnetworks with $\hat{Y}_i = i$; we finally assign the representative subnetwork for the rest subnetworks by the last equation above.

Learning w. The second step is to learn the weighting vector $\mathbf{w}$ for different similarity measures derived from different meta-paths. We aim to find configuration of $\mathbf{w}$ to maximize the objective function, while holding the configuration of $\mathbf{Y}$. Omitting the delta function since at this step the first constraint is always satisfied. By using a Lagrange multiplier, the optimal weights can be updated by:

$$w_m = \frac{\left[\sum_i \sigma_m(S_i, Y_i)\right]^{\frac{1}{1-\beta}}}{\sum_{m'}\left[\sum_i \sigma_{m'}(S_i, Y_i)\right]^{\frac{1}{1-\beta}}} \tag{6}$$

We iteratively update the configuration $\mathbf{Y}$ and weighting vector $\mathbf{w}$ till the objective function converges.

To rank the outlier subnetworks, since we judge whether a subnetwork is an outlier subnetwork by $\sum_m w_m^\beta \sigma_m(i,0) - \max_{j\neq 0}\left[a_{i\leftarrow j} + \sum_m w_m^\beta \sigma_m(i,j)\right]$, and $\sum_m w_m^\beta \sigma_m(i,0)$ remains the same for all the subnetworks when we have no prior knowledge, we can calculate the "outlierness" as:

$$\Omega(S_i) = -\max_{j\neq 0}\left[a_{i\leftarrow j} + \sum_m w_m^\beta \sigma_m(i,j)\right] \tag{7}$$

By ranking subnetwork S_i with respect to $\Omega(S_i)$, we can output the desired ranked list of outliers.

VI. Experiments

To validate the effectiveness of our proposed framework on outlier subnetwork detection, we apply our method on a synthetic data set and several real data sets.

Input: G, $\mathcal{S} = \{S_i\}_{i=1,\ldots,k}$, $\{\mathcal{P}_m\}_{m=1,\ldots,\mu}$, β
Output: Outlierness for all $\Omega(S_i)$

```
// Calculate similarity measures;
forall the P_m do
    forall the S_i, S_j ∈ S do
        Calculate σ_m(S_i, S_j) according to Equation (2);
// Calculate outlierness scores;
w_m ← 1/μ, ∀m = 1, ..., μ;
a_{i←j} = 0, ∀1 ≤ i, j ≤ k ;
repeat
    repeat
        Update r_{i→j}, ∀1 ≤ i ≤ k, 0 ≤ j ≤ k;
        Update a_{i←j}, ∀1 ≤ i ≤ k, 0 ≤ j ≤ k;
    until converged;
    Estimate Ŷ_i, ∀1 ≤ i ≤ k;
    Update w_m, ∀m = 1, ..., μ;
until converged;
Calculate Ω(S_i), ∀1 ≤ i ≤ k;
```

Algorithm 1: Calculate outlierness for subnetworks.

A. Data Sets

We employ a synthetic data set and two real data sets in our experiments.

Synthetic. We first generate a synthetic network by a graph partition model [15] with slight modification. There are n vertices in the vertex set $\mathcal{V}$. There is a coloring assignment function $\psi : \mathcal{V} \rightarrow 2^{\mathcal{C}} \setminus \emptyset$ to assign each vertex a coloring. A vertex v_i is also associated with a type denoted by $\phi(v_i)$. For any two vertices $v_i, v_j \in \mathcal{V}$, we generate an edge between them with probability p if $\psi(v_i) \cap \psi(v_j) \neq \emptyset$; otherwise with probability $q \ll p$. In our experiments, we set n as $1,000$, $|\mathcal{C}|$ as 3 and $|\mathcal{A}|$ as 2. We also set $p = 0.1$ and $q = 0.001$. The function ψ and ϕ for each vertex is randomly determined.

Bibliography. We use a bibliographic heterogeneous information network data generated from ArnetMiner[2] [20]. It has $2,244,018$ publications and $1,274,360$ authors in different fields of computer science. We construct the network by introducing four different types of vertices, corresponding to papers (P), authors (A), venues (V), and terms (T) respectively. The edges involved include paper-author (written-by), paper-venue (published-in) and paper-terms (title-containing).

Patent. We collect a subset of US patents data[3]. The data set consists of 1,000,000 patents, with 970,869 inventors and 96,161 assignees (companies). There are 6 types of vertices: patent (P), inventor (I), assignee (A), term (T), keyword (K), and class (C). Each patent can be associated with several inventors, a few assignees, several terms in its title, some keywords, and a set of classes.

B. Experiment Setup

Comparison methods. There are several baseline methods we can compare with.

- *Individual (Ind).* We perform our proposed algorithm on individual vertices with the PathSim similarity measure and calculate the subnetwork outlierness as the average of members' outlierness scores.

[2]http://arnetminer.org/AMinerNetwork

[3]http://www.uspto.gov/patents/resources/classification/index.jsp

TABLE I. PERFORMANCE COMPARISON (%).

Data set	Synthetic			Bibliography		
Measure	P@5	MAP	AUC	P@5	MAP	AUC
Ind	60.00	66.61	85.00	28.00	24.82	59.91
IndNB	0.00	17.43	76.26	8.00	16.87	55.67
NB	75.00	75.76	93.68	28.00	30.20	67.87
AvgSim	65.00	74.52	95.42	**44.00**	40.47	75.01
MatchSim	76.00	85.54	98.99	**44.00**	40.70	76.24
BMSim	**84.00**	**92.04**	**99.50**	**44.00**	**45.05**	**79.55**

- *Neighborhood-based (NB).* An alternative way to find outlier subnetworks is to use a topic-model-like algorithm by inserting an additional "outlier" topic [5]. By regarding each subnetwork as a document and the neighbors of vertices in the subnetwork as words, we can estimate the probability of a subnetwork belonging to the "outlier" topic as its outlierness.
- *Individual Neighborhood-based (IndNB).* We perform the neighborhood-based method on individual vertices and use the average of outlierness scores of subnetwork members as subnetwork outlierness.

We can also plug in different similarity measures into the $\sigma_j(\cdot,\cdot)$ function in our proposed ranking algorithm, and check which similarity measure is performs better.

- *BMSim.* Our proposed similarity in Equation (2).
- *AvgSim.* Similarity measure replaced by AvgSim defined by in Equation (3).
- *MatchSim.* Similarity measure replaced by MatchSim [14] defined in Equation (4).

Settings. In Synthetic data set, we set $\mathcal{A} = \{A, B\}$. For each query, we directly select 100 subnetworks consisting of type A vertices. Among these subnetworks, 95% are generated by drawing vertices from a "normal" coloring distribution, while another 5% are generated from an "outlier" coloring distribution. We generate 5 different sets of subnetworks for Synthetic data set. We use meta-path ABA to calculate the similarity in our experiments.

In Bibliography data set, we raise five queries q, sharing the same $T_S = A$, the same $\mathcal{P}_S = (AP)$ and the same $\mathcal{P}_Q = (TP)$, but with different keywords $\mathcal{V}_q$. We use "frequent pattern mining", "topic model", "social influence", "named entity recognition", and "transfer learning" in our experiments. We choose 4 types of meta-paths to calculate the similarity: APA, $APAPA$, $APVPA$, and $APTPA$.

In Patent data set, we aim to find outlier subnetworks of assignees (companies), while querying by terms. We tried a list of queries with $\mathcal{P}_S = (API)$ and $\mathcal{P}_Q = (TPI)$. We use meta-paths $APIPA$, $APTPA$, $APAPA$, $APKPA$, and $APCPA$ to calculate the subnetwork similarity.

For all the data sets, β is set to 0.5. For real data sets, we pick the largest threshold θ that guarantees more than 50 subnetworks can be retrieved.

Evaluation. For Synthetic data set, since we manually insert outliers into the data set, the ground-truth is known. For Bibliography data set, we are able to label the outlier subnetworks. We evaluate our outlier ranking results by the mean average precision (MAP), and area under the ROC curve (AUC).

TABLE II. CASE STUDY RESULTS. THE LEFT COLUMN SHOWS THE OUTLIER SUBNETWORKS RANKED AS TOP-5 BY THE PROPOSED METHOD BUT NOT BY THE BASELINE; THE RIGHT COLUMN SHOWS THE OUTLIER SUBNETWORKS RANKED AS TOP-5 BY THE BASELINE BUT NOT BY THE PROPOSED METHOD.

Data set / Query	Comparing results	
	$\mathcal{S}^{\omega}_{BMSim} \setminus \mathcal{S}^{\omega}_{Ind}$	$\mathcal{S}^{\omega}_{Ind} \setminus \mathcal{S}^{\omega}_{BMSim}$
Bibliography / "topic model"	**Ankur Moitra; Sanjeev Arora; Rong Ge** *They have many theoretical papers in STOC, FOCS, etc.*	**Khoat Than; Tu Bao Ho** *They are interested in machine learning and data mining.*
	Andrea Tagarelli; Giovanni Ponti *Giovanni Ponti is interested in economics.*	**Zhongzhi Shi; Huifang Ma; Zhixin Li** *They are interested in machine learning, neural computing and data mining.*
	$\mathcal{S}^{\omega}_{BMSim} \setminus \mathcal{S}^{\omega}_{NB}$	$\mathcal{S}^{\omega}_{NB} \setminus \mathcal{S}^{\omega}_{BMSim}$
Patent / "rechargeable battery"	**LSI Logic Corporation; Symbios Logic Inc.** *LSI designs semiconductors and software to accelerate storage; Symbios is a manufacturer of storage systems.*	**Eltech Systems Corporation; Diamond Shamrock** *Eltech provides solutions for electrochemical industries; Diamond Shamrock produces basic chemical products.*
	Advanced Technology; Teledyne Industries, Inc. *Advanced Technology is an environmental technology company; Teledyne produces digital imaging and engineered systems.*	**Oronzio de Nora; Diamond Shamrock** *Oronzio de Nora is a provider of electrochemical technologies; Diamond Shamrock produces basic chemical products.*

C. Experimental Results

Performance comparison. Table I shows a comparison for different methods in terms of their average performance over multiple queries in both Synthetic and Bibliography data sets. Our proposed method outperforms other baselines in both data sets. In Synthetic data set, our proposed approach achieves almost perfect performance ($> 99\%$ in AUC and $> 90\%$ in MAP), while the two baselines based on individual outlier detection (Ind and IndNB) achieve relatively poor performance. This verifies our claim that subnetwork outlier detection is a problem more challenging than individual outlier detection and cannot directly be solved by traditional methods. The results on Bibliography data set also show that our proposed method outperforms the baselines, with a fairly high performance (about 80% in AUC and 45% in MAP).

Case study. We conduct a case study to compare our method with several baseline methods on different data sets (Cf. Table II). In Bibliography data set, we compare BMSim with Ind baseline on a selected query. We find that our method is able to find interesting outlier subnetworks such as a theory research group (Moitra *et al.*) publishing papers about "topic model", which is very different from other retrieved subnetworks from data mining or machine learning areas. However, the results returned by Ind baseline is not satisfying. For Patent data set, we compare BMSim with the NB baseline. It shows that NB prioritizes a subnetwork of electrochemical companies, which is fairly normal for the query "rechargeable battery". This is probably because NB fails to characterize "normal stereotype" of subnetworks when there are many big companies with a great variety of business involved (Hitachi, Mitsubishi, *etc.*), as they introduce significant noise. In comparison, our proposed method can still tell interesting outliers (*e.g.* LSI and Symbios).

VII. CONCLUSION

In this work, we present a novel problem of query-based subnetwork outlier detection in a heterogeneous information network. We design a similarity measure which can capture the proximity between membership distributions of two subnetworks and propose a novel outlier detection method to rank the subnetworks by their outlierness. The subnetwork outlier detection can trigger various applications, such as identification of gangs of terrorists hidden in a large social network. Also, it is worth exploring several extensions of this problem, *e.g.* to apply the algorithm on dynamic heterogeneous networks.

Acknowledgements. Research was sponsored in part by the Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA) and W911NF-11-2-0086, the Army Research Office under Cooperative Agreement No. W911NF-13-1-0193, National Science Foundation IIS-1017362, IIS-1320617, and IIS-1354329, HDTRA1-10-1-0120, and MIAS, a DHS-IDS Center for Multimodal Information Access and Synthesis at UIUC. Jie Tang and Jing Zhang are supported by the National High-tech R&D Program (No. 2014AA015103) and Natural Science Foundation of China (No. 61222212).

REFERENCES

[1] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

[2] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, 2009.

[3] T. Eiter and H. Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133, 1997.

[4] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007.

[5] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. In *KDD*, pages 813–822, 2010.

[6] M. Gupta, J. Gao, Y. Sun, and J. Han. Integrating community matching and outlier detection for mining evolutionary community outliers. In *KDD*, pages 859–867, 2012.

[7] M. Gupta, J. Gao, X. Yan, H. Cam, and J. Han. Top-k interesting subgraph discovery in information networks. In *ICDE*, 2014.

[8] M. Gupta, A. Mallya, S. Roy, J. H. Cho, and J. Han. Local learning for mining outlier subgraphs from network datasets. In *SDM*, 2014.

[9] S. Hirose, K. Yamanishi, T. Nakata, and R. Fujimaki. Network anomaly detection based on eigen equation compression. In *KDD*, pages 1185–1194, 2009.

[10] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[11] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *KDD*, pages 538–543, 2002.

[12] R. Jin, V. E. Lee, and H. Hong. Axiomatic ranking of network role similarity. In *KDD*, pages 922–930, 2011.

[13] D. Liben-Nowell and J. M. Kleinberg. The link-prediction problem for social networks. *JASIST*, 58(7):1019–1031, 2007.

[14] Z. Lin, M. R. Lyu, and I. King. Matchsim: a novel neighbor-based similarity measure with maximum neighborhood matching. In *CIKM*, pages 1613–1616, 2009.

[15] F. McSherry. Spectral partitioning of random graphs. In *FOCS*, pages 529–537, 2001.

[16] M. Mongiovi, P. Bogdanov, and R. Ranca. Netspot: Spotting significant anomalous regions on dynamic networks. In *SDM*, 2013.

[17] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *KDD*, pages 631–636, 2003.

[18] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM*, pages 418–425, 2005.

[19] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, pages 992–1003, 2011.

[20] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD*, pages 990–998, 2008.

[21] H. Tong, C. Faloutsos, and Y. Koren. Fast direction-aware proximity for graph mining. In *KDD*, pages 747–756, 2007.

[22] R. Yu, X. He, and Y. Liu. Glad: group anomaly detection in social media analysis. In *KDD*, pages 372–381, 2014.