On Community Outliers and their Efficient Detection in Information Networks[*]

Jing Gao[†], Feng Liang[†], Wei Fan[‡], Chi Wang[†], Yizhou Sun[†], and Jiawei Han[†]
[†]University of Illinois at Urbana-Champaign, IL USA
[‡]IBM T. J. Watson Research Center, Hawthorn, NY USA
[†]{jinggao3,liangf,chiwang1,sun22,hanj}@illinois.edu, [‡]weifan@us.ibm.com

ABSTRACT

Linked or networked data are ubiquitous in many applications. Examples include web data or hypertext documents connected via hyperlinks, social networks or user profiles connected via friend links, co-authorship and citation information, blog data, movie reviews and so on. In these datasets (called "information networks"), closely related objects that share the same properties or interests form a community. For example, a community in blogsphere could be users mostly interested in cell phone reviews and news. Outlier detection in information networks can reveal important anomalous and interesting behaviors that are not obvious if community information is ignored. An example could be a low-income person being friends with many rich people even though his income is not anomalously low when considered over the entire population. This paper first introduces the concept of community outliers (interesting points or rising stars for a more positive sense), and then shows that well-known baseline approaches without considering links or community information cannot find these community outliers. We propose an efficient solution by modeling networked data as a mixture model composed of multiple normal communities and a set of randomly generated outliers. The probabilistic model characterizes both data and links simultaneously by defining their joint distribution based on hidden Markov random fields (HMRF). Maximizing the data likelihood and the posterior of the model gives the solution to the outlier inference problem. We apply the model on both synthetic data and DBLP data sets, and the results demonstrate importance of this concept, as well as the effectiveness and efficiency of the proposed approach.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms

Keywords

outlier detection, community discovery, information networks

1. INTRODUCTION

Outliers, or anomalies, refer to aberrant or interesting objects whose characteristics deviate significantly from the majority of the data. Although the problem of outlier detection has been widely studied [6], most of the existing approaches identify outliers from a global aspect, where the entire data set is examined. In many scenarios, however, an object may only be considered abnormal in a specific context but not globally [25, 29]. Such contextual outliers are sometimes more interesting and important than global outliers. For example, 20 Fahrenheit degree is not a global outlier in temperature, but it represents anomalous weather in the spring of New York City.

In this paper, we study the problem of finding contextual outliers in an "information network". Networks have been used to describe numerous physical systems in our everyday life, including Internet composed of gigantic networks of webpages, friendship networks obtained from social web sites, and co-author networks drawn from bibliographic data. We regard each node in a network as an object, and there usually exist large amounts of information describing each object, e.g. the hypertext document of each webpage, the profile of each user, and the publications of each researcher. The most important and interesting aspect of these datasets is the presence of links or relationships among objects, which is different from the feature vector data type that we are more familiar with. We refer to the networks having information from both objects and links as information networks. Intuitively, objects connected via the network have many interactions, subsequently share mutual interests, and thus form a variety of communities in the network [11]. For example, in a blogsphere, there could be

[*]Research was sponsored in part by Air Force Office of Scientific Research MURI award FA9550-08-1-0265, the U.S. National Science Foundation under grants IIS-09-05215 and DMS-0732276, and by the Army Research Laboratory under Cooperative Agreement Number W911NF-09-2-0053 (NS-CTA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

KDD'10, July 25–28, 2010, Washington, DC, USA.

financial, literature, and technology cliches. Taking communities as contexts, we aim at detecting outliers that have non-conforming patterns compared with other members in the same community.

Example: Low-income person with rich friends

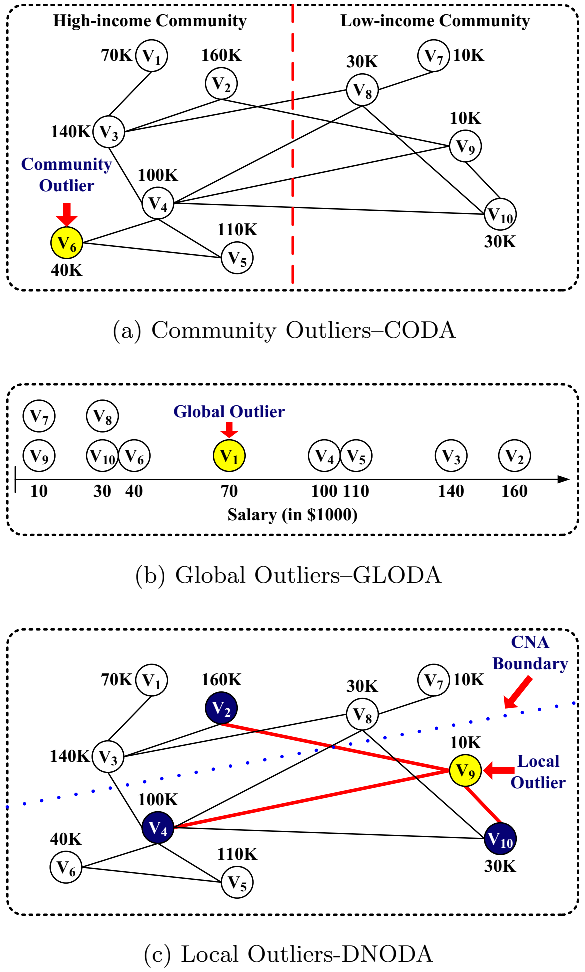
A friend network is shown in Figure 1(a), where each node denotes a person, and a link represents the friend relationship between two persons. Each person's annual salary is shown as a number attached to each node. There obviously exist two communities, high-income (v_1,v_2,v_3,v_4,v_5) and low-income (v_7,v_8,v_9,v_{10}). Interestingly, v_6 is an example of community outliers. It is only linked to the high-income community (70 to 160K), but has a relatively low income (40K). This person could be a rising star in the social network, for example, a young and promising entrepreneur, or someone who may settle down in a rich neighborhood. Another example is a co-author network. Researchers are linked through co-authorship, and texts are extracted from publications for each author in bibliographic databases. A researcher who does independent research on a rare topic is an outlier among people in his research community, for example, a linguistic researcher in the area of data mining. Additionally, an actor cooperation network can be drawn from movie databases where actors are connected if they co-star a movie. Exceptions can be found when an actor's profile deviates much from his co-star communities, such as a comedy actor co-starring with lots of action movie stars.

Limitation of Traditional Approaches

Identifying community outliers is a non-trivial task. First, if we conduct outlier detection only based on each object's information, without taking network structure into account, the identified outliers would only be "global" outliers. As shown in Figure 1(b), v_1 is a global outlier with 70K deviating from the other salary amounts in the "low-income person with rich friends" example. We call this method GLobal Outlier Detection Algorithm (**GLODA**). Secondly, when only "local" information (i.e., information from neighboring nodes) is considered, the identified node is just significantly away from its adjacent neighbors. It is a "local" outlier, not necessarily a "community" outlier. As illustrated in Figure 1(c), v_9 is a local outlier because his salary is quite different from those of his direct friends (v_2, v_4 and v_{10}). The corresponding algorithm is denoted as Direct Neighbor Outlier Detection Algorithm (**DNODA**).

In detecting community outliers, both the information at each individual object and the one in the network should be taken into account simultaneously. A naive solution is to first partition the network into several communities using network information [24, 14], and then within each community, identify outliers based on the object information. This two-stage algorithm is referred to as Community Neighbor Algorithm (**CNA**). The problem with such a two-stage approach is that communities discovered using merely network information may not make much sense. For example, partitioning the graph in Figure 1(c) along the dotted line minimizes the number of normalized cuts, and thus the line represents the boundary between two communities identified by CNA. However, the resulting two communities have wide-spread income levels, and thus it does not make much sense to detect outliers in such two communities.

Therefore, we propose to utilize both the network and data information in an integrated solution, in order to improve community discovery and find more meaningful outliers. The algorithm we developed is called community outlier detection algorithm (**CODA**). With the proposed method, the network in Figure 1(a) will be divided by the dashed line, and v_6 is detected as the community outlier. In many applications, no matter the network is dense or sparse, there is ambiguity in community partitions. This is particularly true for very large networks, since information from both nodes and links can be noisy and incomplete. Consolidating information from both sources can compensate missing or incomplete information from one side alone and is likely to yield a better solution.



(a) Community Outliers–CODA

(b) Global Outliers–GLODA

(c) Local Outliers-DNODA

Figure 1: Comparison of Different Types of Outliers

Some clustering methods (**CLA** for short) have been developed to group nodes in an information network into communities using both data and link information [17, 32, 30]. Those methods, however, are not designed for outlier detection. The reason is that they are proposed under the assumption that there are no outliers. It is well-known that outliers can highly affect the formation of communities. Different from those methods, the proposed approach combines, instead of separating, outlier detection and community mining into a unified framework. As summarized in Table 1, both GLODA and DNODA only use part of the available in-

Table 1: Summary of Related Work

Algorithms	Tasks	Information Sources
GLODA DNODA	**global** outlier detection **local** outlier detection	data of objects data and direct neighbors
CNA	find communities then detect outliers	use data and links **separately**
CLA	clustering in information networks	use data and links **together**

formation, whereas the other two approaches consider both data and links. However, CNA utilizes the two information sources separately, and CLA is used to conduct clustering, instead of outlier detection.

Summary of the Proposed Approach

In this paper, we propose a probabilistic model for community outlier detection in information networks. It provides a unified framework for outlier detection and community discovery, integrating information from both the objects and the network. The information collected at each object is formulated as a multivariate data point, generated by a mixture model. We use K components to describe normal community behavior and one component for outliers. Distributions for community components are, but not limited to, either Gaussian (continuous data) or multinomial (text data), whereas the outlier component is drawn from a uniform distribution. The mixture model induces a hidden variable z_i at each object node, which indicates its community. Then inference on z_i's becomes the key in detecting community outliers. We regard the network information as a graph describing the dependency relationships among objects. The links from the network (i.e., the graph) are incorporated into our modeling via a hidden Markov random field (HMRF) on the hidden variable z_i's. We motivate an objective function from the posterior energy of the HMRF model, and find its local minimum by using an Iterated Conditional Modes (ICM) algorithm. We also provide some methods for setting the hyper-parameters in the model. Moreover, the proposed model can be easily generalized to handle a variety of data as long as a distance function is defined.

A summary of this paper is as follows:

- Finding community outliers is an important problem but has not received enough attention in the field of information network analysis. To the best of our knowledge, this is the first work on identifying community outliers by analyzing both the data and links simultaneously.

- We propose an integrated probabilistic model to interpret normal objects and outliers, where the object information is described by some generative mixture model, and network information is encoded as spatial constraints on the hidden variables via a HMRF model.

- Efficient algorithms based on EM and ICM algorithms are provided to fit the HMRF model as well as inferring the hidden label of each object.

- We validate the proposed algorithm on both synthetic and real data sets, and the results demonstrate the advantages of the proposed approach in finding community outliers.

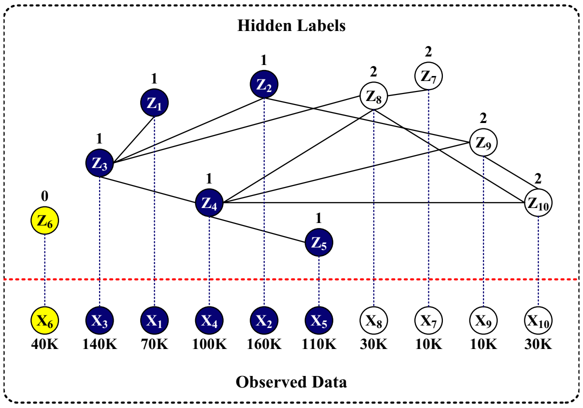


Figure 2: Community Outlier Detection Model

2. COMMUNITY OUTLIER DETECTION

Community outliers can be defined in various ways. We define it based on a generative model unifying data and links. Based on the definition, we discuss the specific models for continuous data and text data. Table 2 summarizes some important notations used in the paper.

2.1 Outlier Detection via HMRF

The problem is defined as follows: suppose we have an information network denoted as a graph $G = (V, W)$, where V denotes a set of objects $\{v_1, \ldots, v_M\}$, and W represents the links between each pair of objects. Specifically, the input include:

- $S = \{s_1, \ldots, s_M\}$ where s_i is the data associated with object v_i.

- W is the symmetric $M \times M$ adjacency matrix of the network where w_{ij} ($w_{ij} \geq 0$) is the weight of the link between the two objects v_i and v_j. If $w_{ij} > 0$, v_i and v_j are connected.

Let $I = \{1, \ldots, M\}$ be the set of indices of the M objects. The objective is to derive the anomalous subset $\{i : v_i$ is a contextual outlier with respect to S and W, $i \in I\}$.

Next, we discuss how to formulate this using HMRF model. Mathematically, a HMRF model is characterized by the following:

Observed data

$X = \{x_1, \ldots, x_M\}$ is a set of random variables. Each random variable x_i generates the data s_i associated with the i-th object.

Hidden labels

$Z = \{z_1, \ldots, z_M\}$ is the set of hidden random variables, whose values are unobservable. Each variable z_i indicates the community assignment of v_i. Suppose there are K communities, then $z_i \in \{0, 1, \ldots, K\}$. If $z_i = 0$, v_i is an outlier. If $z_i = k$ ($k \neq 0$), v_i belongs to the k-th community.

Neighborhood system

The links in W induce dependency relationships among the hidden labels, with the rationale that if two objects v_i and v_j are linked on the network (i.e., they are neighbors), then they are more likely to belong to the same community

Table 2: Important Notations

Symbol	Definition
$I = \{1, \ldots, i, \ldots, M\}$	the indices of objects
$V = \{v_1, \ldots, v_M\}$	the set of objects
$S = \{s_1, \ldots, s_M\}$	the given attribute values of the objects
$W_{M \times M} = [w_{ij}]$	the given link structure, w_{ij}-the link strength between objects v_i and v_j
$Z = \{z_1, \ldots, z_M\}$	the set of random variables for hidden labels of the objects
$X = \{x_1, \ldots, x_M\}$	the set of random variables for observed data
$N_i \quad (i \in I)$	the neighborhood of object v_i
$1, \ldots, k, \ldots, K$	the indices of normal communities
$\Theta = \{\theta_1, \ldots, \theta_K\}$	the set of random variables for model parameters
$\theta_k = \{\mu_k, \sigma_k^2\}$	the parameters of the k-th normal community (continuous data): μ_k-mean, σ_k^2-variance
$\theta_k = \{\beta_{k1}, \beta_{k2}, \ldots, \beta_{kT}\}$	the parameters of the k-th normal community (text data)
$\beta_{kl} \quad (l = 1, \ldots, T)$	the probability of observing the l-th word in the k-th community (text data)

(i.e., z_i and z_j are likely to have the same value). However, since outliers are randomly generated, the neighbors of an outlier are not necessarily outliers. So we adjust the neighborhood system as the following:

$$N_i = \begin{cases} \{j; w_{ij} > 0, i \neq j, z_j \neq 0\} & z_i \neq 0 \\ \phi & z_i = 0. \end{cases}$$

Here N_i stands for the set of neighbors of object v_i. When $z_i \neq 0$, i.e., v_i is not an outlier, the neighborhood of v_i contains its normal neighbors in G. In contrast, v_i's neighborhood is empty if it is an outlier ($z_i = 0$).

Conditional independence

The set of random variables X are conditionally independent given their labels:

$$P(X = S|Z) = \prod_{i=1}^{M} P(x_i = s_i|z_i).$$

Normal Communities and Outliers

We assume that the k-th normal community ($k \neq 0$) is characterized by a set of parameters θ_k, i.e.,

$$P(x_i = s_i|z_i = k) = P(x_i = s_i|\theta_k).$$

Quite differently, the outliers follow a uniform distribution, i.e.,

$$P(x_i = s_i|z_i = 0) = \rho_0$$

where ρ_0 is a constant. Let $\Theta = \{\theta_1, \ldots, \theta_K\}$ be the set of all parameters describing the normal communities.

Dependency between hidden variables

The random field defined over the hidden variables Z is a Markov random field, where the Markov property is satisfied:

$$P(z_i|z_{I-\{i\}}) = P(z_i|z_{N_i}) \quad z_i \neq 0.$$

It indicates that the probability distribution of z_i depends only on the labels of v_i's neighbors in G if z_i corresponds to a normal community. If $z_i = 0$, v_i is an outlier and is not linked to any other objects in the random field, and thus we set $P(z_i = 0) = \pi_0$ where π_0 is a constant. According to the Hammerskey-Clifford theorem [3], an MRF can equivalently be characterized by a Gibbs distribution:

$$P(Z) = \frac{1}{H_1} \exp(-U(Z)) \tag{1}$$

where H_1 is a normalizing constant, and $U(Z) = \sum_{c \in C} V_c(Z)$, the potential function, is a sum of clique potentials $V_c(Z)$ over all possible cliques ($c \in C$) in G. Since outliers are stand-alone objects (their links in G are ignored in the model), we define the potential function only on the neighborhood of normal objects:

$$U(Z) = -\lambda \sum_{w_{ij} > 0, z_i \neq 0, z_j \neq 0} w_{ij} \delta(z_i - z_j) \tag{2}$$

where λ is a constant, $w_{ij} > 0$ denotes that there is a link connecting the two objects v_i and v_j, and both z_i and z_j are non-zero. The δ function is defined as $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise. The potential function suggests that, if v_i and v_j are normal objects, they are more likely to be in the same community when there exists a link connecting them in G, and the probability becomes higher if their link w_{ij} is stronger.

Figure 2 shows the HMRF model for the example in Figure 1(a). The top layer represents the hidden variables $\{z_1, \ldots, z_{10}\}$. It has the same topology as the original network G except that the neighborhood of z_6 is now empty because it is an outlier. Given $z_i = k$, the corresponding data value is generated according to the parameter θ_k. The bottom layer is composed of the data values (salaries) of the objects. In this example, two communities are formed, and objects in the same community are strongly linked in the top layer, as well as having similar values in the bottom layer. When considering both data and link information, we cannot assign v_6 to any community (linked to community 1 but its value is closer to community 2), and thus regard it as a community outlier.

2.2 Modeling Continuous and Text Data

In the proposed model, the probability of hidden variables is modeled by Eq. (1) and Eq. (2), and the outliers are generated by a uniform distribution. However, given the hidden variable $z_i \neq 0$, the probability distribution of x_i can be modeled in various ways depending on the format it is taking. In this part, we discuss how $P(x_i = s_i|z_i)$ ($z_i \neq 0$) is modeled when s_i is continuous or a text document, the two major types of data we encounter in applications. Extensions to general cases are discussed in Section 4.

Continuous Data

For the sake of simplicity, we assume that the data S are 1-dimensional real numbers. Extensions to model multi-dimensional continuous data are straightforward. We propose to model the normal points in S by a Gaussian mixture due to its flexibility in approximating a wide range of continuous distributions. Parameters needed to describe the k-th community are the mean μ_k and variance σ_k^2: $\theta_k =$

$\{\mu_k, \sigma_k^2\}$. Given the model parameter $\Theta = (\theta_1, \ldots, \theta_K)$, if $z_i = k \in \{1, \ldots, K\}$, the logarithm of the conditional likelihood $\ln P(x_i = s_i | z_i = k)$ is:

$$\ln P(x_i = s_i | z_i = k) = -\frac{(s_i - \mu_k)^2}{2\sigma_k^2} - \ln \sigma_k - \ln \sqrt{2\pi}. \quad (3)$$

Text Data

Suppose each object v_i is a document that is comprised of a bag of words. Let $\{w_1, w_2, \ldots, w_T\}$ be all the words in the vocabulary, and each document is represented by a vector $s_i = (d_{i1}, d_{i2}, \ldots, d_{iT})$, where d_{il} denotes the count of word w_l in v_i. Now the parameter characterizing each normal community is $\theta_k = \{\beta_{k1}, \beta_{k2}, \ldots, \beta_{kT}\}$ where $\beta_{kl} = P(w_l | z_i = k)$ is the probability of seeing word w_l in the k-th community. Given that a document v_i is in the k-th community, its word counts s_i follow a multinomial distribution, and thus $\ln P(x_i = s_i | z_i = k)$ is defined as:

$$\ln P(x_i = s_i | z_i = k) = \sum_{l=1}^{T} d_{il} \ln P(w_l | z_i = k) = \sum_{l=1}^{T} d_{il} \ln \beta_{kl}. \quad (4)$$

3. FITTING COMMUNITY OUTLIER DETECTION MODEL

In the HMRF model for outlier detection we discussed in Section 2, both the model parameters Θ and the set of hidden labels Z are unknown. In this section, we present the method to infer the values of hidden variables (Section 3.1) and estimate model parameters (Section 3.2).

3.1 Inference

We first assume that the model parameters in Θ are known, and discuss how to obtain an assignment of the hidden variables. The objective is to find the configuration that maximizes the posterior distribution given Θ. We then discuss how to estimate Θ and Z simultaneously in Section 3.2.

In general, we seek a labeling of the objects, $Z = \{z_1, \ldots, z_M\}$, to maximize the posterior probability (MAP):

$$\hat{Z} = \arg\max_{Z} P(X = S|Z)P(Z).$$

We use the Iterated Conditional Modes (ICM) algorithm [4] to solve this MAP estimation problem. It adopts a greedy strategy by calculating local minimization iteratively and the convergence is guaranteed after a few iterations. The basic idea is to sequentially update the label of each object, keeping the labels of the other objects fixed. At each step, the algorithm updates z_i given $x_i = s_i$ and the other labels by maximizing $P(z_i | x_i = s_i, z_{I-\{i\}})$, the conditional posterior probability. Next we discuss the two scenarios separately when z_i takes non-zero or zero values.

If $z_i \neq 0$, we have

$$P(z_i | x_i = s_i, z_{I-\{i\}}) \propto P(x_i = s_i | Z) P(Z).$$

As discussed in Eq. (1) and Eq. (2), the probability distribution of Z is given by

$$P(Z) \propto \exp\Big(\lambda \sum_{w_{ij}>0, z_i \neq 0, z_j \neq 0} w_{ij} \delta(z_i - z_j)\Big).$$

Algorithm 1 Updating Labels

Input: set of data S, adjacency matrix W, set of model parameters Θ, number of clusters K, link importance λ, threshold a_0, initial assignment of labels $Z^{(1)}$;
Output: updated assignment of labels Z;
Algorithm:

```
Randomly set $Z^{(0)}$
$t \leftarrow 1$
while $Z^{(t)}$ is not close enough to $Z^{(t-1)}$ do
  $t \leftarrow t + 1$
  for $i = 1; i <= M; i++$ do
    update $z_i^{(t)} = k$ which minimizes $U_i(k)$ in Eq. (6).
return $Z^{(t)}$
```

In $P(z_i | x_i = s_i, z_{I-\{i\}})$, the links that involve objects other than v_i are irrelevant, and thus

$$P(z_i | x_i = s_i, z_{I-\{i\}}) \propto P(x_i = s_i | z_i) \cdot \exp\Big(\lambda \sum_{j \in N_i} w_{ij} \delta(z_i - z_j)\Big)$$

where only the links between v_i and its neighbors in N_i are taken into account. We take logarithm of the posterior probability, and then transform the MAP estimation problem to the minimization of the conditional posterior energy function:

$$U_i(k) = -\ln P(x_i = s_i | z_i = k) - \lambda \sum_{j \in N_i} w_{ij} \delta(k - z_j).$$

If $z_i = 0$, v_i has no neighbors, and thus

$$P(z_i | x_i = s_i, z_{I-\{i\}}) \propto P(x_i = s_i | z_i = 0) P(z_i = 0) = \exp(-U_i(0)) \quad (5)$$

with

$$U_i(0) = -\ln(\rho_0 \pi_0) = a_0.$$

Therefore, to find z_i that maximizes $P(z_i | x_i = s_i, z_{I-\{i\}})$, it is equivalent to minimizing the posterior energy function: $\hat{z}_i = \arg\min_k U_i(k)$ where

$$U_i(k) = \begin{cases} -\ln P(x_i = s_i | z_i = k) - \lambda \sum_{j \in N_i} w_{ij} \delta(k - z_j) & k \neq 0 \\ a_0 & k = 0 \end{cases} \quad (6)$$

As can be seen, λ is a predefined hyper-parameter that represents the importance of the network structure. $\ln P(x_i = s_i | z_i = k)$ is defined in Eq. (3) and Eq. (4) for continuous and text data respectively. To minimize $U_i(k)$, we first select a normal cluster k^* such that $k^* = \arg\min_k U_i(k) (k \neq 0)$. Then we compare $U_i(k^*)$ with $U_i(0)$, which is a predefined threshold a_0. If $U_i(k^*) > a_0$, we set $\hat{z}_i = 0$, otherwise $\hat{z}_i = k^*$. As shown in Algorithm 1, we first initialize the label assignment for all the objects, and then repeat the update procedure until convergence. At each run, the labels are updated sequentially by minimizing $U_i(k)$, which is the posterior energy given $x_i = s_i$ and the labels of the remaining objects.

3.2 Parameter Estimation

In Section 3.1, we assume that Θ is known, which is usually unrealistic. In this part, we consider the problem of estimating unknown Θ from the data. Θ describes the model that generates S, and thus we seek to maximize the data likelihood $P(X = S|\Theta)$ to obtain $\hat{\Theta}$. However, because both the hidden labels and the parameters are unknown and they are inter-dependent, it is intractable to directly maximize the

Algorithm 2 Community Outlier Detection

```
Input: set of data $S$, adjacency matrix $W$, number of clusters $K$, link importance $\lambda$, threshold $a_0$;
Output: set of outliers;
Algorithm:
  Initialize $Z^0$,$Z^1$ randomly
  $t \leftarrow 1$
  while $Z^{(t)}$ is not close enough to $Z^{(t-1)}$ do
    M-step: Given $Z^{(t)}$, update the model parameters $\Theta^{(t+1)}$ according to Eq. (8) and Eq. (9) (continuous data), or Eq. (10) (text data).
    E-step: Given $\Theta^{(t+1)}$, update the hidden labels as $Z^{(t+1)}$ using Algorithm 1.
    $t \leftarrow t + 1$
  return the indices of outliers: $\{i : z_i^{(t)} = 0, i \in I\}$
```

data likelihood. We view it as an "incomplete-data" problem, and use the expectation-maximization (EM) algorithm to solve it.

The basic idea is as follows. We start with an initial estimate $\Theta^{(0)}$, then at E-step, calculate the conditional expectation $Q(\Theta|\Theta^{(t)}) = \sum_Z P(Z|X, \Theta^{(t)}) \ln P(X, Z|\Theta)$, and at M-step, maximize $Q(\Theta|\Theta^{(t)})$ to get $\Theta^{(t+1)}$ and repeat. In the HMRF outlier detection model, we can factorize $P(X, Z|\Theta)$ as $P(X|Z, \Theta)P(Z)$, and since $P(Z)$ is not related to Θ, we can regard the corresponding terms as a constant in Q. Similarly, the outlier component does not contribute to estimation of Θ neither, and thus $\sum_{i=1}^{n} P(z_i = 0|x_i = s_i) \ln P(x_i = s_i|z_i = 0)$ can also be absorbed into the constant term H_2:

$$Q = \sum_{i=1}^{M} \left(\sum_{k=1}^{K} P(z_i = k|x_i = s_i, \Theta^{(t)}) \ln P(x_i = s_i|z_i = k, \Theta) \right) + H_2. \tag{7}$$

We approximate $P(z_i = k|x_i = s_i, \Theta^{(t)})$ using the estimates obtained from Algorithm 1, where $P(z_i = k^*|x_i = s_i, \Theta^{(t)}) = 1$ if $k^* = \arg\min_k U_i(k)$, and 0 otherwise.

Specifically, for continuous data, we maximize Q to get the mean and variance of each normal community $k \in \{1, \ldots, K\}$, where $\ln P(x_i = s_i|z_i = k, \Theta)$ is defined in Eq. (3):

$$\mu_k^{(t+1)} = \frac{\sum_{i=1}^{M} P(z_i = k|x_i = s_i, \Theta^{(t)}) s_i}{\sum_{i=1}^{M} P(z_i = k|x_i = s_i, \Theta^{(t)})}, \tag{8}$$

$$(\sigma_k^{(t+1)})^2 = \frac{\sum_{i=1}^{M} P(z_i = k|x_i = s_i, \Theta^{(t)})(s_i - \mu_k)^2}{\sum_{i=1}^{M} P(z_i = k|x_i = s_i, \Theta^{(t)})}. \tag{9}$$

Similarly, for text data, based on Eq. (4), as well as the constraints that $\sum_{l=1}^{T} \beta_{kl} = 1$ $(k = 1, \ldots, K)$, we have:

$$\beta_{kl}^{(t+1)} = \frac{\sum_{i=1}^{M} P(z_i = k|x_i = s_i, \Theta^{(t)}) d_{il}}{\sum_{l=1}^{T} \sum_{i=1}^{M} P(z_i = k|x_i = s_i, \Theta^{(t)}) d_{il}} \tag{10}$$

for $k = 1, \ldots, K$ and $l = 1, \ldots, T$.

In summary, the community outlier detection algorithm works as follows. As shown in Algorithm 2, we begin with some initial label assignment of the objects. In the M-step, the model parameters are estimated by maximizing the Q function based on the current label assignment. In the E-step, we run Algorithm 1 to re-assign the labels to the objects by minimizing $U_i(k)$ for each node v_i sequentially. The E-step and M-step are repeated until convergence is achieved, and thus the outliers are the nodes that have 0 as the estimated labels. Note that the running time is linear in the number of edges. It is not worse than any baseline that uses links because each edge has to be visited at least once. For dense graphs, the time can be quadratic in the number of objects. However, in practice, we usually encounter sparse graphs, on which the method runs in linear time and can scale well.

4. DISCUSSIONS

To use the community outlier detection algorithm more effectively, the following questions need to be addressed: 1) How to set the hyper parameters? 2) What is a good initialization of the label assignment Z? 3) Can the algorithm be applied to any type of data?

Setting Hyper-parameters

We need users' input on three hyper-parameters: threshold a_0, link importance λ, and the number of components K. Intuitively, a_0 controls the percentage of outliers r discovered by the algorithm. We will expect a large number of outliers if a_0 is low and few outliers if a_0 is high. Therefore, we can transform the problem of setting a_0, which is difficult, to an easier problem to choose the percentage of outliers r. To do this, in Algorithm 1, we first let $\hat{z_i} = \arg\min_k U_i(k)(k \neq 0)$ for each $i \in I$, and sort $U_i(\hat{z_i})$ for $i = 1, \ldots, M$ and select the top r percent as outliers.

$\lambda > 0$ represents our confidence in the network structure where we put more weights on the network and less weights on the data if λ is set higher. Therefore, if λ is lower, the outliers found by Algorithm 2 is more similar to the results of detecting outliers merely based on nodes information. On the other hand, a higher λ makes the network structure play a more important role in community discovery and outlier detection. It is obvious that if we set λ to be extremely high, and the graph is connected, then every node will turn out to have the same label. To avoid such cases, we set an upper bound λ^C so that for any $\lambda > \lambda^C$, the results contain empty communities. With this requirement, we show that the proposed algorithm is not sensitive to λ in Section 5.

K is a positive integer, denoting the number of normal communities. In principle, it controls the scale of the community, and thus a small K leads to "global" outliers, whereas the outliers are determined locally if lots of communities are formed (i.e., large K). Many techniques have been proposed to set K effectively, for example, Bayesian information criterion (BIC), Akaike information criteria (AIC) and minimum description length (MDL). In this paper, we use AIC to set the number of normal communities. It is defined as:

$$AIC(\Delta) = 2b - 2\ln P(X|\Delta) \tag{11}$$

where Δ denotes the set of hyper-parameters and b is the number of parameters to be estimated (model complexity). Since $P(X|\Delta)$ is hard to obtain, in the proposed algorithm, we use $P(X|\hat{Z}, \Delta)$ to approximate it by assuming that $\hat{Z}$ is the true configuration of the hidden variables.

Initialization

Good initialization is essential for the success of the proposed community outlier detection algorithm, otherwise the algorithm can get stuck at some local maximum. Instead of starting with a random initialization, we initialize Z by clustering the objects without assigning any outliers. Although this may affect the estimation of the model parameters at

Table 3: Comparison of Precisions on Synthetic Data

Precisions		$K = 5$				$K = 8$			
		GLODA	DNODA	CNA	CODA	GLODA	DNODA	CNA	CODA
$M = 1000$	$r = 1\%$	0.0143	0.0714	0.5429	**0.6286**	0.0571	0.0571	0.4429	**0.7429**
	$r = 5\%$	0.0867	0.2600	0.6930	**0.8106**	0.0688	0.1554	0.5723	**0.6565**
$M = 2000$	$r = 1\%$	0.0118	0.0111	0.1007	**0.6565**	0.0395	0.0170	0.1536	**0.4974**
	$r = 5\%$	0.0567	0.1779	0.4645	**0.6799**	0.0649	0.1341	0.4944	**0.7047**
$M = 5000$	$r = 1\%$	0.0061	0.0041	0.0510	**0.3714**	0.0163	0.0000	0.0204	**0.5347**
	$r = 5\%$	0.0496	0.1134	0.1854	**0.7302**	0.0565	0.0646	0.1602	**0.7926**

the first iteration, it can gradually get corrected while we update Z and nominate outliers in the E-step. To overcome the barrier of local maximum, we repeat the algorithm multiple times with different initialization and choose the one that maximizes the likelihood.

Extensions

We have provided models for continuous and text data, which already covers lots of applications. Here, we discuss extension of the proposed approach to more general data formats by using a distance function. In general, we let the center of each community μ_k to be the parameter characterizing the community, and define $D(s_i, \mu_k)$ to be the distance in feature values from any object s_i to the center of the k-th community μ_k. For $k = 1, \ldots, K$, we then define $P(x_i = s_i | z_i = k)$ in terms of the distance function:

$$P(x_i = s_i | z_i = k) \propto \exp(-D(s_i, \mu_k))$$

which suggests that given v_i is from a normal community, the probability of $x_i = s_i$ increases as the distance from s_i to μ_k gets closer. For example, we can choose D to be a distance function from the class of Bregman divergence [1], which is a generalization from the Euclidean distance and is known to have a strong connection to exponential families of distributions.

5. EXPERIMENTS

The evaluation of community outlier detection itself is an open problem due to lack of groundtruths for community outliers. Therefore, we conduct experiments on synthetic data to compare detection accuracy with the baseline methods, and evaluate on real datasets to validate that the proposed algorithm can detect community outliers effectively[1].

5.1 Synthetic Data

In this part, we describe the experimental setting and results on the synthetic data.

Data Generation

We generate networked data through two steps. First, we generate synthetic graphs, which follow the properties of real networks–they are usually sparse, follow power law's degree distributions, and consist of several communities. The links of each object follow Zipf's law, i.e., most of the nodes have very few links, and only a few nodes connect to many nodes. We forbid self-links and remove the nodes that have no links. Secondly, we infer the label of each node following the proposed generative model, and sample a continuous number based on the label of each node. The configuration parameters to describe $P(X|Z)$ include the number of communities K and the percentage of outliers r. We draw the mean of each community uniformly from [-10,10], let the standard deviation be $10/K$, and generate random numbers using Gaussian probability density.

Baseline Methods

As discussed in Section 1, we compare the proposed community outlier detection algorithm (**CODA**) with the following outlier detection methods:

- **GLODA**: This baseline looks at the data values only. We use the popular outlier detection algorithm LOF [5] to detect "global" outliers without taking the network structure into account.
- **DNODA**: This method only considers the values of each object's direct neighbors in the graph. We define the outlier score as:

$$\frac{\sum_{j \in N_i} D(s_i, s_j)}{|N_i|} \tag{12}$$

  where D is the Euclidean distance function. N_i contains all the direct neighbors of v_i in the graph: $N_i = \{j : w_{ij} > 0, i \neq j\}$. If s_i is significantly different from the data of v_i's direct neighbors, it is considered an outlier.
- **CNA**: In this approach, we partition the graph into K communities using clustering algorithms [13], and define outliers as the objects that have significantly different values compared with the other objects in the same community. Therefore, the outlier score is calculated in the same way as in Eq. (12). But here, N_i stands for the whole community: $N_i = \{j : z_i = z_j, i \neq j\}$ where z_i is the community label derived from the clustering of the network structure.

Empirical Results

In experimental studies, we make each baseline method detect the same number of outliers as that of the groundtruths. To achieve this, we simply sort the outlier scores obtained by the three baseline methods in descending order, and take the top r percent as outliers. Then we use **precision**, also known as **true positive rate**, as the evaluation metric. It is defined as the percentage of correct ones in the set of outliers identified by the algorithm. We vary the scale of the network to have 1000, 2000 and 5000 nodes respectively. We set the number of clusters K to be either 5 or 8, and the percentage of outliers r to be either 1% or 5%.

For each parameter setting, we randomly generate 10 sets of networked data, and report the average precisions of all the methods in Table 3. It is clear that GLODA fails to find most of community outliers because the method completely ignores the network structure information. The approach that only checks the direct neighbors of each object to determine outliers (DNODA) also has a low precision. On

[1] http://ews.uiuc.edu/∼jinggao3/kdd10coda.htm

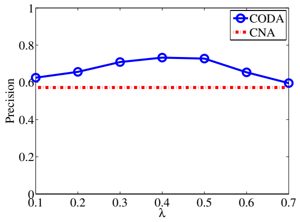


Figure 3: Sensitivity

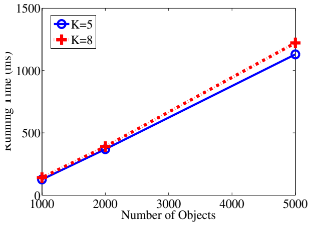


Figure 4: Running Time

the other hand, if we first discover the communities, and then identify outliers based on the peers in the community, the precision is improved as shown in the method CNA. The proposed CODA algorithm further increases the precision by modeling both data and link information. We can observe the consistent improvements where the margin of precision increase is from 8% to 60%.

Sensitivity

Figure 3 shows the performance of the CODA algorithm when we vary λ from 0.1 to 0.7, as illustrated using the solid line. The dotted line represents the performance of the best baseline method CNA applied on the same data set. The results are obtained on the synthetic data with 1000 objects, 5 communities and 1% community outliers. It is clear that in spite of slight changes caused by parameter variation, the proposed method improves over the best baseline method. We let $\lambda = 0.2$ to get the experimental results shown in Table 3.

Time Complexity

Suppose the number of objects is M, and the number of edges is E. In M-step, we need to visit all the objects to calculate the model parameters, so the time complexity is $O(M)$. In E-step, for each object v_i, the posterior energy function U_i has to aggregate the effect of the labels of v_i's neighbors to compute $P(Z)$. Therefore, in principle, the time of the E-step is $O(E)$. Real network is usually sparse, and thus the computation time of the proposed approach can be linear in the number of objects. Figure 4 presents the average running time of the CODA algorithm on the synthetic data. We generate sparse networks using power law distribution where the number of edges grow linearly, and thus the running time is linear in the number of objects.

5.2 DBLP

DBLP[2] provides bibliographic information on major computer science journals and proceedings. We extract two subnetworks from the DBLP data: a conference relationship network and a co-authorship network.

Sub-network of Conferences

In the conference relationship network, we use 20 conferences from four research areas as the nodes of the graph, and construct a similarity graph based on the 20 nodes. Suppose there are L authors, then each conference has a $L \times 1$ vector A_i, whose l-th entry is the number of times the l-th author publishes in the i-th conference. We use cosine similarity to represent the link weight between two conferences:

$$w_{ij} = cos(A_i, A_j) = \frac{A_i \cdot A_j}{||A_i||||A_j||}. \tag{13}$$

This suggests that the conferences that attract the same set of authors have strong connections, and such conferences may form a research community. Additionally, we have a document attached to each node, which contains all the published titles in the conference. We conduct the community outlier detection algorithm on this network to obtain the outlier that has a different research theme compared with the other conferences containing similar researchers.

From this dataset, we find the following communities:

- **Database:** ICDE, VLDB, SIGMOD, PODS, EDBT
- **Artificial Intelligence:** IJCAI, AAAI, ICML, ECML
- **Data Mining:** KDD, PAKDD, ICDM, PKDD, SDM
- **Information Analysis:** SIGIR, WWW, ECIR, WSDM

The community outliers detected by the proposed algorithm include **CVPR** and **CIKM**. Clearly, CVPR is more likely to fall into the AI area because researchers in CVPR will often attend IJCAI, AAAI, ICML and ECML. However, although people in computer vision utilize many general artificial intelligence methods, there exist unique computer vision techniques, such as segmentation, object tracking, and image modeling. Therefore, CVPR represents a community outlier in this problem. On the other hand, CIKM has a wide-spread scope, and attracts people from information analysis, data mining, and database areas. Apparently, it has a different research theme from that of any conference in these areas, and thus represents a community outlier as well.

Sub-network of Authors

We extract a co-authorship network, which contains the authors publishing in the 20 conferences mentioned above from DBLP. We select the top 3116 authors with the highest number of publications in these conferences, and use them as nodes of the network[3]. If two researchers have co-authored papers, there is an edge connecting them in the graph. The weight of the edge is the number of times two researchers have collaborated. We run the CODA algorithm on this co-author network to identify communities and community outliers. The top-10 frequent words occurring in each community identified by the algorithm are shown in Table 4. It is obvious that we can discover four research communities

Table 4: Top Words in Communities

Communities	Keywords
Data Mining	frequent dimensional spatial association similarity pattern fast sets approximate series
Database	oriented views applications querying design access schema control integration sql
Artificial Intelligence	reasoning planning logic representation recognition solving problem reinforcement programming theory
Information Analysis	relevance feature ranking automatic documents probabilistic extraction user study classifiers

[2] http://www.informatik.uni-trier.de/∼ley/db/

[3] This is a sub-network of the original DBLP network. There could have some information loss in the co-authorship relationships.

Researchers & Collaborators	**Research Interests**
Dennis Shasha **DB** 19 **DM** 6	biological computing, pattern recognition, querying in trees and graphs, pattern discovery in time series, cryptographic file systems, database tuning
Craig A. Knoblock **IA** 4 **AI** 4 **DM** 1 **DB** 1	planning, machine learning, constraint reasoning, semantic web, information extraction, gathering, integration, mediators, wrappers, source modeling, record linkage, mashup construction, geospatial and biological data integration
Eric Horvitz **IA** 9 **AI** 4	human decision making, computational models of reflection, action with applications in time-critical decision making, scientific exploration, information retrieval, and healthcare
Sourav S. Bhowmick **IA** 8 **DM** 2 **DB** 2	blogs, social media analysis; web evolution, evolution, graph mining; social networks, XML storage, query processing, usability of XML/graph databases, indexing and querying graphs, predictive modeling, comparison of molecular networks, multi-target drug therapy
Timothy W. Finin **IA** 6 **AI** 1	social media, the semantic web, intelligent agents, pervasive computing
Jack Mostow **AI** 3 **IA** 2	focuses on using computers to listen to children read aloud while other interests include machine learning, automated replay of design plans, and discovery of search heuristics
Terrance E. Boult **AI** 2 **IA** 1	vision and security including video surveillance systems, biometrics, biometric fusion, supporting trauma treatment, steganalysis, network security, detection of chemical and biological weapons
Jayant R. Kalagnanam **DB** 3 **AI** 2 **IA** 1	decision support, optimization, economics and their applications to electronic commerce
Ken Barker **IA** 2 **AI** 2 **DB** 1	knowledge representation and reasoning, knowledge acquisition, natural language processing
Dimitris Achlioptas **AI** 4	threshold phenomena in random graphs and random formulas, applications of embeddings and spectral techniques in machine learning, algorithmic analysis of massive networks

Figure 5: Community Outliers in DBLP co-authors

in this co-author network: Database (**DB**), Artificial Intelligence (**AI**), Data Mining (**DM**), and Information Analysis (**IA**).

Outliers in this sub-network somehow represent researchers who are conducting research on some different topics from his collaborators and peer researchers in the community. To illustrate the effectiveness of the proposed algorithm, we check the research interests listed on the homepages of the researchers identified by the CODA algorithm. In Figure 5, we show each researcher's name together with the number of his collaborators in each of the four communities (DB, AI, DM, and IA) in the first column. Their research interests are shown in the second column. As can be seen, these researchers indeed studied something different from his collaborators and the majority of the communities. For example, Jayant R. Kalagnanam mainly focuses on electronic commerce, which is a less popular topic among his collaborators in Database, Artificial intelligence and Information Analysis areas. Jack Mostow has focused on using computers to listen to children read aloud, which is a less studied research theme in Artificial Intelligence and Information Analysis. Through this example, we demonstrate that the proposed CODA algorithm has the ability of detecting outliers that deviate from the rest of the community.

6. RELATED WORK

Outlier detection, sometimes referred to as anomaly or novelty detection, has received considerable attention in the field of data mining [6]. Recently, people began to study how to identify anomalies within a specific context. These methods are able to detect interesting outliers or anomalies which cannot be found by existing outlier detection algorithms from a global view. Specifically, the pre-defined contextual attributes include spatial attributes [23, 27], neighborhoods in graphs [26], and some contextual attributes [25]. When there is no a priori contextual information available, Wang et al. propose to simultaneously explore contexts and contextual outliers based on random walks [29]. The proposed community outlier problem differs from these papers in that we use communities in networks as contexts, and they are inferred based on both *data* and *link* information.

Outlier detection in data without considering contexts is called *global outlier detection*. Existing methods detect anomalies based on how far their distances [16], densities [5], statistical distributions [21] deviate from the rest of the data. The proposed model shares some common properties with existing methods. For example, we assume that outliers are far from any clusters and are uniformly distributed [7]. On the other hand, we may refer to outliers identified in network structures purely by link analysis as *structural outliers* [31]. There are also works devoting to finding unusual sub-graph patterns in networks [22]. Clearly, these types of outliers are not the same as the community outliers defined in this paper. In general, outlier detection is *unsupervised*, i.e., the task is to identify something novel or anomalous without the aid of labeled data. There exist some semi-supervised outlier detection approaches that take labeled examples as prior knowledge of label distribution [33, 28, 8]. In this paper, we aim at *unsupervised* outlier detection on networked data requiring no labeled data.

In recent years, many methods have been developed to discover clusters or communities in networks [11]. At first, community discovery is conducted on links only without consulting objects' information. Such techniques find communities as strongly connected sub-networks by similarity computation [15, 12] or graph partitioning [24, 20, 14]. Later, it was found that utilizing both link and data information leads to the discovery of more accurate and meaningful communities [17, 32, 30]. Some relational clustering methods [9, 19] fall into this category when both attributes of objects and relationships between objects are considered. Among various techniques, Markov random field [18, 34] is commonly used to model the structural dependency among random variables and has been successfully applied to many applications, such as image segmentation. More generally, relational learning explores use of link structure in inference and learning problems [10]. Moreover, some semi-supervised clustering techniques based on must-links and cannot-links [2] can be used to discover communities on networked data as well, where network structures provide must-links. As shown in the experiments, separating community discovery and outlier detection cannot work as well as our unified model because absorbing outliers into normal communities affect the profiling of normal communities, and in turn degrade the performance of the second stage outlier detection.

7. CONCLUSIONS

In this paper, we discuss a new outlier detection problem in networks containing rich information, including data about each object and relationships among objects. We detect outliers within the context of communities such that the identified outliers deviate significantly from the rest of the community members. We propose a generative model called CODA that unifies both community discovery and outlier detection in a probabilistic formulation based on hidden Markov random fields. We assume that normal objects form K communities and outliers are randomly generated. The data attributes associated with each object are modeled using mixture of Gaussian distributions or multinomial distributions, whereas links are used to calculate prior distributions over hidden labels. We present efficient algorithms based on ICM and EM techniques to learn model parameters and infer the hidden labels of the community outlier detection model. Experimental results show that the proposed CODA algorithm consistently outperforms the baseline methods on synthetic data, and also identifies meaningful community outliers from the DBLP network data.

8. REFERENCES

[1] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh. Clustering with bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[2] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proc. of KDD'04*, pages 59–68, 2004.

[3] J. Besag. Spatial interaction and the statistical analysis of lattic systems. *Journal of the Royal Statistical Society, Series B*, 36(2):192–236, 1974.

[4] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society, Series B*, 48(3):259–302, 1986.

[5] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proc. of SIGMOD'00*, pages 93–104, 2000.

[6] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. Technical Report 07-017, University of Minnesota, 2007.

[7] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proc. of ICML '00*, pages 255–262, 2000.

[8] J. Gao, H. Cheng, and P.-N. Tan. a novel framework for incorporating labeled examples into anomaly detection. In *Proc. of SDM'06*, pages 594–598, 2006.

[9] R. Ge, M. Ester, B. J. Gao, Z. Hu, B. Bhattacharya, and B. Ben-Moshe. Joint cluster analysis of attribute data and relationship data: The connected k-center problem, algorithms and applications. *ACM Transactions on Knowledge Discovery from Data*, 2(2):1–35, 2008.

[10] L. Getoor and B. Taskar. *Introduction to statistical relational learning*. MIT Press, 2007.

[11] M. Girvan and M. Newman. Community structure in social and biological networks. 99(12):7821–7826, 2002.

[12] G. Jeh and J. Widom. Simrank: a measure of structural-context similarity. In *Proc. of KDD'02*, pages 538–543, 2002.

[13] G. Karypis. Cluto - family of data clustering software tools. http://glaros.dtc.umn.edu/gkhome/views/cluto.

[14] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of ACM*, 46(5):604–632, 1999.

[16] E. M. Knorr, R. T. Ng, and V. Tucakov. Distance-based outliers: algorithms and applications. *The VLDB Journal*, 8(3-4):237–253, 2000.

[17] H. Li, Z. Nie, W.-C. Lee, L. Giles, and J.-R. Wen. Scalable community discovery on textual data with relations. In *Proc. of CIKM'08*, pages 1203–1212, 2008.

[18] S. Z. Li. *Markov random field modeling in computer vision*. Springer-Verlag, 1995.

[19] B. Long, Z. M. Zhang, and P. S. Yu. A probabilistic framework for relational clustering. In *Proc. of KDD'07*, pages 470–479, 2007.

[20] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[21] M. Markou and S. Singh. Novelty detection: a review-part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[22] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *Proc. of KDD'03*, pages 631–636, 2003.

[23] S. Shekhar, C.-T. Lu, and P. Zhang. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proc. of KDD'01*, pages 371–376, 2001.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[25] X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, 2007.

[26] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Prof. of ICDM'05*, pages 418–425, 2005.

[27] P. Sun and S. Chawla. On local spatial outliers. In *Proc. of ICDM'04*, pages 209–216, 2004.

[28] A. Vinueza and G. Grudic. Unsupervised outlier detection and semi-supervised learning. Technical Report CU-CS-976-04, University of Colorado at Boulder, 2004.

[29] X. Wang and I. Davidson. Discovering contexts and contextual outliers using random walks in graphs. In *Proc. of ICDM'09*, pages 1034–1039, 2009.

[30] X. Wang, N. Mohanty, and A. McCallum. Group and topic discovery from relations and their attributes. In *Proc. of NIPS'06*, pages 1449–1456, 2006.

[31] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. Scan: a structural clustering algorithm for networks. In *Proc. of KDD'07*, pages 824–833, 2007.

[32] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proc. of KDD'09*, pages 927–936, 2009.

[33] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *Proc. of CVPR'05*, pages 611–618, 2005.

[34] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, 2001.