Accelerated Local Anomaly Detection via Resolving Attributed Networks

Ninghao Liu[1], Xiao Huang[1], Xia Hu[1,2]
[1]Department of Computer Science and Engineering, Texas A&M University
[2]Center for Remote Health Technologies and Systems, Texas A&M Engineering Experiment Station
{nhliu43, xhuang, xiahu}@tamu.edu

Abstract

Attributed networks, in which network connectivity and node attributes are available, have been increasingly used to model real-world information systems, such as social media and e-commerce platforms. While outlier detection has been extensively studied to identify anomalies that deviate from certain chosen background, existing algorithms cannot be directly applied on attributed networks due to the heterogeneous types of information and the scale of real-world data. Meanwhile, it has been observed that local anomalies, which may align with global condition, are hard to be detected by existing algorithms with interpretability. Motivated by the observations, in this paper, we propose to study the problem of effective and efficient local anomaly detection in attributed networks. In particular, we design a collective way for modeling heterogeneous network and attribute information, and develop a novel and efficient distributed optimization algorithm to handle large-scale data. In the experiments, we compare the proposed framework with the state-of-the-art methods on both real and synthetic datasets, and demonstrate its effectiveness and efficiency through quantitative evaluation and case studies.

1 Introduction

Anomalies refer to the noteworthy objects with patterns or behaviors that significantly deviate from the background. Anomaly detection has been intensively studied for various applications, such as spam detection [Yang *et al.*, 2012], fraud detection [Rayana and Akoglu, 2015], events detection [Chen *et al.*, 2016] and computer security [Cheng *et al.*, 2016]. Successful anomaly detection plays a critical role in many information systems towards achieving a secure cyberspace.

Due to the data characteristics in many real-world information systems, anomaly detection faces new challenges. First, attributed networks [Huang *et al.*, 2017a; 2017b; Akoglu *et al.*, 2015] have been increasingly used in modeling complex real-world information systems. Different from plain graphs, in attributed networks, both connectivity information and node properties are available. For example, in social networks with friendship relations as links, users can also be characterized by their interests, work background, and user-generated posts. Unfortunately, existing anomaly detection algorithms usually focus on either attribute information [Xiong *et al.*, 2011; Zhao and Fu, 2015; Jiang *et al.*, 2015] or network structure [Xu *et al.*, 2007], or do not provide much interpretation for anomalies beyond visualization [Tong and Lin, 2011; Akoglu *et al.*, 2012]. Furthermore, real-world attributed networks usually contain large-scale data instances and high-dimensional features. Given the heterogeneous and large-scale attributed networks, existing anomaly detection methods cannot be simply applied.

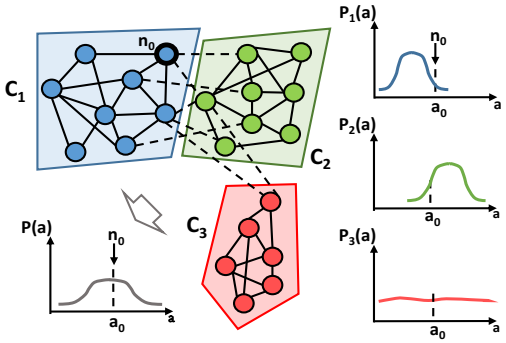

Figure 1: A toy network of three communities with known attribute distributions and an anomaly node n_0.

In addition, it has been observed that some local anomalies well align with the global condition and cannot be easily detected. Take the attributed network in Fig. 1 as an example. There are three clusters in the network. Each node has one continuous numerical attribute a, and $p_c(a)$ represents the attribute distribution of the instances in community c. By only looking at node n_0 in the overall distribution $p(a)$, it is unlikely to identify it as an anomaly. However, by collectively examining the network structure and attribute information of the node, we can clearly observe that n_0 is an anomaly, as it is inconsistent with other nodes in C_1 according to the distribution in $p_1(a)$. This phenomenon can also be observed in health provider networks, where the records declared by an anomaly provider may look normal among all entities, but its fraud activities can be revealed by a refined inspection in the local community [Akoglu *et al.*, 2015]. Thus, motivated by these observations, we propose to investigate the problem of local anomaly detection in large-scale attributed networks.

To deal with the aforementioned challenges, in this paper, we propose to collectively make use of network and attribute information for effective and efficient anomaly detection. Specifically, we investigate how to extract insightful information from heterogeneous data sources as the basis for spotting local abnormal nodes, as well as design an accelerated algorithm to deal with large-scale data. The contributions are presented as follows:

- An integrated anomaly detection framework for attributed networks is proposed. Network structure and node attribute information are jointly considered for network clustering and attribute distributions estimation. The normality score of a node is measured within its local context.
- The model is interpretable as we are able to identify the abnormal attributes of anomalies and measure their degrees of abnormality.
- A novel parallel algorithm is developed to accelerate the optimization process to resolve networks, especially when the given network is sparse in terms of links or attributes.
- The effectiveness and efficiency of our framework are demonstrated through experiments on real-world and synthetic networks, as well as case studies.

2 Local Anomaly Detection Framework

In this paper, we propose a novel framework for Accelerated Local Anomaly Detection (ALAD) in attributed networks. The notations used in this paper are introduced as follows. We use bold uppercase letters (e.g. $\mathbf{A}$) to denote matrices. We represent the $(n,m)^{th}$ entry of a matrix as $\mathbf{A}_{nm}$, the n^{th} row of a matrix as $\mathbf{A}_{n*}$, and the m^{th} column as $\mathbf{A}_{*m}$. The $(i,j)^{th}$ block of a matrix is represented using $\mathbf{A}^{i,j}$. We denote the ℓ_2-norm of a vector as $\|\cdot\|_2$, and Frobenius norm of a matrix as $\|\cdot\|_F$. Let $\mathcal{N}=\{\mathcal{V},\mathbf{G},\mathbf{A}\}$ be a target attributed network, where $\mathcal{V}$ is a set of N nodes, and $\mathbf{G}\in\mathbb{R}_{\geq 0}^{N\times N}$ is the weighted adjacency matrix. $\mathbf{G}_{nm}$ equals to the positive edge weight if there is a link between nodes n and m, and 0 otherwise. $\mathbf{A}\in\mathbb{R}_{\geq 0}^{N\times K}$ represents the node attribute matrix, where $\mathbf{A}_{n*}$ describes the attributes associated with node n. We assume that the network is undirected, and all attribute values are non-negative.

2.1 Problem Statement

Anomalies are defined as the objects that "arouse suspicions that it was generated by a different mechanism" [Hawkins, 1980]. According to the definition, there are three questions to be answered for designing a detection method: (1) where is the local context in which an object locates; (2) what is the probable data generation mechanism in such local context; (3) how to measure the suspicion of current object with respect to the contextual mechanism. Under this problem setting, each object n can be characterized using two types of features [Chandola *et al.*, 2009]: 1) **context features** (w_n) which determine the location, group or genre of an object, e.g., the spatial coordinates of a city, or the category labels of a commercial product; 2) **content features** (a_n) which describe the attributes, behaviors or properties of an object, e.g., the demographics of a city, or the user reviews of a product. The neighbors of object n can be discovered by comparing w_n with other $w_{n'}$, $n'\neq n$. In attributed networks, the information for determining the context features of a node is not limited to its connection status. The attribute information may also affect the role of a node in its local context. Let h_c denote the **attribute distributions** measured from $\{a_{n_1},a_{n_2},...\}$ where objects $n_1,n_2,...$ belong to the same context c. The suspicious score of object n being an anomaly with respect to context c can thus be formulated as the disparity between a_n and h_c. Many traditional anomaly detection algorithms, either explicitly or implicitly, have adopted the three factors above [Gao *et al.*, 2010; Liang and Parthasarathy, 2016; Perozzi *et al.*, 2014; Liang *et al.*, 2017]. In some cases, w_n and a_n refer to the same feature set from homogeneous data sources [Breunig *et al.*, 2000; Jiang *et al.*, 2015].

Based on the terminologies described above, we formally define the problem as follows: Given the attributed network $\mathcal{N}$, we aim at developing a framework to identify anomalies as nodes o whose content features a_o significantly deviate from the feature distribution h_c of their context localized by the context features w_o of o.

2.2 Context Extraction and Summarization from Heterogeneous Data

For the given network $\mathcal{N}$, each node n is characterized by both $\mathbf{G}_{n*}\in\mathbb{R}_{\geq 0}^{N}$ and $\mathbf{A}_{n*}\in\mathbb{R}_{\geq 0}^{K}$, which cannot be directly concatenated as w_n due to their heterogeneity and high dimensionality. In fact, the number of factors that determine the dependencies in a network is limited [Li *et al.*, 2017]. Therefore, we would like to first transform the original features to a low-dimensional vector $\mathbf{W}_{n*}\in\mathbb{R}_{\geq 0}^{C}$. Then a straightforward solution for anomaly detection is to apply heterogeneous network embedding methods [Chang *et al.*, 2015] to build $\mathbf{W}$, perform clustering on nodes, estimate the data distribution on each cluster (e.g., take the mean values), and at last find anomalies. This pipeline, however, is limited by several problems. The noise in attributes may affect the accuracy of distribution estimation. Moreover, it relies on the performance of specific embedding methods, while the superposition of embedding and clustering also significantly increases the computational cost. The anomaly detection step tends to be a byproduct of embedding and clustering methods.

Because of the reasons above, we propose to map the data to a C-dimensional latent space in which each $\mathbf{W}_{nc}$ already corresponds to a group or topic for $c\in[1,C]$. In the latent space, taking the attribute matrix $\mathbf{A}$ as an example, the property of each node can be seen as a weighted linear composition of the aspects from C groups, i.e., $\mathbf{A}_{n*}\approx\sum_{c=1}^{C}\mathbf{W}_{nc}\mathbf{H}_{c*}$, where $\mathbf{W}_{nc}$ measures how close is node n affiliated with cluster c, and $\mathbf{H}_{ck}$ indicates the degree to which the attribute k is associated with cluster c [Xu *et al.*, 2003]. In another word, $\mathbf{W}_{n*}$ represents the context features of object n, while $\mathbf{H}_{c*}$ encodes the attribute distributions of cluster c. The problem of learning latent factors, i.e. approximating $\mathbf{A}\approx\mathbf{WH}$, can be solved via non-negative matrix factorization (NMF) [Lee and Seung, 2001]. Here $\mathbf{W}\in\mathbb{R}_{\geq 0}^{N\times C}$, $\mathbf{H}\in\mathbb{R}_{\geq 0}^{C\times K}$, and C is the number of latent factors.

Despite the heterogeneity of the two information sources $\mathbf{G}$ and $\mathbf{A}$, both of their row dimensions represent network nodes, which provides the basis for joint consideration. Motivated by symmetric nonnegative matrix factorization [Kuang *et al.*, 2012], we incorporate links information in $\mathbf{G}$ and propose the objective function formally as below:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{H}} \ & \|\mathbf{G}-\mathbf{W}\mathbf{W}^T\|_F^2+\alpha\|\mathbf{A}-\mathbf{W}\mathbf{H}\|_F^2 \\ & +\gamma(\|\mathbf{W}\|_F^2+\|\mathbf{H}\|_F^2), \\ \text{s.t.} \ & \mathbf{W}\geq 0, \mathbf{H}\geq 0. \end{aligned} \tag{1}$$

Two nodes i and j that are closely correlated in the same context will have similar representation vectors $\mathbf{W}_{i*}$ and $\mathbf{W}_{j*}$. The regularization terms $\|\mathbf{W}\|_F^2$ and $\|\mathbf{H}\|_F^2$ are used to avoid overfitting and the affect of data noise. The elements in $\mathbf{W}$ and $\mathbf{H}$ are constrained to be non-negative, not only because $\mathbf{G}$ and $\mathbf{A}$ are non-negative, but also to achieve interpretability. As the solution for NMF is not unique, we further normalize the factor matrices as below:

$$\mathbf{H}_{c,k}\leftarrow \mathbf{H}_{c,k}/\sqrt{\sum_k \mathbf{H}_{c,k}^2}, \quad \mathbf{W}_{n,c}\leftarrow \mathbf{W}_{n,c}\sqrt{\sum_k \mathbf{H}_{c,k}^2}. \tag{2}$$

The proposed model has several nice properties. First, NMF based clustering is semantically interpretable [Kuang *et al.*, 2012]. Second, the model does not have a strong assumption such as positive semidefinite on the network, so it is general to various real-world applications. Third, by applying the bi-factorization scheme instead of its tri-factorization counterpart [Long *et al.*, 2006], we could avoid convoluted interactions between parameters, thus enabling to develop efficient optimization algorithms.

2.3 Local Suspiciousness Evaluation

The suspiciousness of a node should be evaluated within a proper context. As a node n may belong to several groups simultaneously, its suspiciousness can be measured as:

$$s(n)=p(n;c)\,s(n,c)=\tilde{\mathbf{W}}_{n,c}\,s(n,c), \tag{3}$$

where $\tilde{\mathbf{W}}_{n,c}=\mathbf{W}_{n,c}/\sum_c \mathbf{W}_{n,c}$ is interpreted as the likelihood that node n belongs to group c [Shashanka *et al.*, 2008].

After locating nodes in the latent space, we want to compute how significantly node n deviates from its background condition. To uniformly handle different types of distributions (e.g., Gaussian or Bernoulli) in real-world systems, we transform all types of attributes to categorical ones [Catlett, 1991], so that $\mathbf{A}_{nk}$ counts the occurrence of attribute k on node n. As mentioned before, the suspiciousness of node n can be measured by the disparity between a_n and h_c. Here we take content features $a_n=\mathbf{A}_{n*}$ and distribution vector $h_c=\mathbf{H}_{c*}$. Some commonly used metrics for quantifying the difference between two vectors include Euclidean distance $\|\mathbf{A}_{n*}-\mathbf{H}_{c*}\|_2$, cosine similarity $cos(\mathbf{A}_{n*},\mathbf{H}_{c*})$ and Kullback Leibler (KL) divergence $D(\mathbf{A}_{n*}\|\mathbf{H}_{c*})$. In many real-world networks, the attributes are expected to be high-dimensional and sparse, so Euclidean distance and KL divergence may not perform well [Zimek *et al.*, 2012]. Here we use cosine similarity to calculate the normality score, i.e. inverse of suspiciousness score, between the two vectors:

$$s(n,c)=cos(\mathbf{A}_{n*},\mathbf{H}_{c*})=\frac{<\mathbf{A}_{n*},\mathbf{H}_{c*}>}{\|\mathbf{A}_{n*}\|_2\|\mathbf{H}_{c*}\|_2}. \tag{4}$$

The intuition behind the score is that, if there is little overlap between the attributes of a node n and the distribution of its local context, then the node should be viewed as abnormal.

After computing the normality score for each node, we can have a ranked list in which nodes are sorted in ascending order with respect to their scores, where nodes ranked higher are considered as anomalies.

3 Accelerated Optimization

Typical optimization algorithms for NMF, such as multiplicative update [Lee and Seung, 2001] and stochastic gradient descent (SGD) [Koren *et al.*, 2009], are computationally expensive and cannot be simply applied to our problem. The former is costly for each round of parameter update, while the latter uses only a small number of data instances at each iteration. In this section, we will introduce a distributed algorithm for accelerated optimization.

3.1 Parallel Optimization Scheme

In parallel algorithms, concurrent implementations are required to be independent of each other to avoid interference. To have a clearer view of the objective function, as well as to schedule the updates of parameter batches, we rewrite Eq. (1) in the block-wise summation form as follows,

$$\begin{aligned} & L_{\mathbf{G},\mathbf{A}}(\mathbf{W},\mathbf{H}) \\ & =\sum_{i,j}(\|\mathbf{G}^{i,j}-\mathbf{W}^i\mathbf{W}^{j^T}\|_F^2+\frac{\gamma}{2B}\|\mathbf{W}^i\|_F^2+\frac{\gamma}{2B}\|\mathbf{W}^j\|_F^2) \\ & +\sum_{i,k}(\,\alpha\sum_l\|\mathbf{A}^{i,k}-\mathbf{W}^i\mathbf{H}^k\|_F^2+\frac{\gamma}{B}\|\mathbf{H}^k\|_F^2) \\ & =\sum_{\{i,j,k\}}(\underbrace{\|\mathbf{G}^{i,j}-\mathbf{W}^i\mathbf{W}^{j^T}\|_F^2}_{L_{\mathbf{G}^{i,j}}}+\frac{\gamma}{2B}\|\mathbf{W}^i\|_F^2+\frac{\gamma}{2B}\|\mathbf{W}^j\|_F^2 \\ & +\frac{\alpha}{2}\underbrace{\|\mathbf{A}^{i,k}-\mathbf{W}^i\mathbf{H}^k\|_F^2}_{L_{\mathbf{A}^{i,k}}}+\frac{\alpha}{2}\underbrace{\|\mathbf{A}^{j,k}-\mathbf{W}^j\mathbf{H}^k\|_F^2}_{L_{\mathbf{A}^{j,k}}}+\frac{\gamma}{B}\|\mathbf{H}^k\|_F^2) \\ & =\sum_{\{i,j,k\}}L_{i,j,k}(\mathbf{W}^i,\mathbf{W}^j,\mathbf{H}^k), \end{aligned} \tag{5}$$

where B represents the number of splits on each dimension, and the superscripts denote the position of each block. An intuitive illustration of problem segmentation is shown in Fig.2. To approximate $\mathbf{G}^{i,j}\in\mathbb{R}^{N_i\times N_j}$, we only need to update $\mathbf{W}^i$ and $\mathbf{W}^j$, while the local factorization of $\mathbf{A}^{i,k}$ only involves $\mathbf{W}^i$ and $\mathbf{H}^k$. The two dimensions of $\mathbf{G}$ are identically partitioned due to its symmetry. In practice, $\mathbf{G}$ and $\mathbf{A}$ can be segmented into nonuniform blocks, depending on the entry density in different matrix regions and computing resources available on different machines in a distributed system.

We design a parallel mini-batch SGD algorithm to efficiently solve the problem. To generalize, let $L(\theta)$ denote the objective function, where $\theta=\{\mathbf{W},\mathbf{H}\}$ represents the parameters to learn. $L(\theta)=\sum_{\mathcal{U}\in\{i,j,k\}}L_{\mathcal{U}}(\theta)$, in which $\mathcal{U}$ represents an **instances set** $\{\mathbf{G}^{i,j},\mathbf{A}^{i,k},\mathbf{A}^{j,k}\}$ across the two matrices. Starting from initial θ_0, traditional mini-batch stochastic gradient descent refines the parameter by iterating the update: $\theta_{t+1}=\theta_t-\epsilon_t\nabla L_{\mathcal{U}}(\theta_t)$, where $-\nabla L_{\mathcal{U}}(\theta)$ is the steepest-descent direction of $L(\theta)$ over the data samples in $\mathcal{U}$.

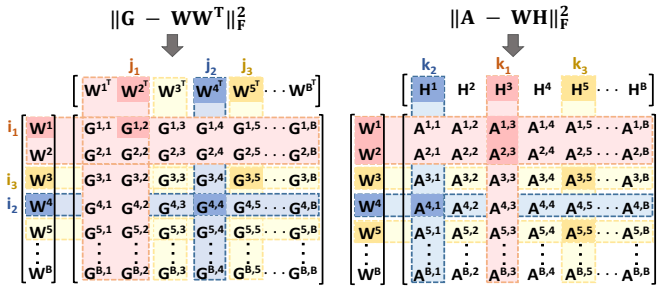

Figure 2: Matrices segmentation and parallel optimization scheme. There are three interchangeable sets $\{\mathbf{G}^{1,2}, \mathbf{A}^{1,3}, \mathbf{A}^{2,3}\}$, $\{\mathbf{G}^{4,4}, \mathbf{A}^{4,1}, \mathbf{A}^{4,1}\}$ and $\{\mathbf{G}^{3,5}, \mathbf{A}^{3,5}, \mathbf{A}^{5,5}\}$.

An example of instances set can be found in in Fig.2, where $\mathcal{U} = \{\mathbf{G}^{1,2}, \mathbf{A}^{1,3}, \mathbf{A}^{2,3}\}$ makes $\mathbf{W}^1, \mathbf{W}^2, \mathbf{H}^3$ to be updated. However, sequential SGD does not fully utilize the computation bandwidth. Motivated by the concept of instance-level interchangeability [Gemulla *et al.*, 2011], we propose the definition of **interchangeable sets** as below:

Definition 1. *$\mathcal{U}_1$,$\mathcal{U}_2$ are interchangeable sets concerning a loss function L if any two instances $u_1 \in \mathcal{U}_1$ and $u_2 \in \mathcal{U}_2$ are interchangeable, where u_1, u_2 are interchangeable if*

$$\begin{aligned} \nabla L_{u_1}(\theta) &= \nabla L_{u_1}(\theta - \epsilon \nabla L_{u_2}(\theta)), \\ \nabla L_{u_2}(\theta) &= \nabla L_{u_2}(\theta - \epsilon \nabla L_{u_1}(\theta)). \end{aligned} \tag{6}$$

Interchangeable sets can be processed in parallel without interference. Two sets $\mathcal{U}_1 = \{\mathbf{G}^{i_1,j_1}, \mathbf{A}^{i_1,k_1}, \mathbf{A}^{j_1,k_1}\}$ and $\mathcal{U}_2 = \{\mathbf{G}^{i_2,j_2}, \mathbf{A}^{i_2,k_2}, \mathbf{A}^{j_2,k_2}\}$ are guaranteed to be interchangeable if $i_1 \neq i_2, j_1 \neq j_2, k_1 \neq k_2, i_1 \neq j_2$ and $i_2 \neq j_1$. To fully utilize the computation resources and to guarantee interchangeability between instance sets, the parallel optimization scheme with three interchangeable sets is shown in Fig.2. Instance sets in different colors can be processed in parallel. Picking one data block will lock all other blocks in the same row or column at current iteration. The detailed optimization process can be found from line 2 $\sim$ 9 in Alg. 1. After obtaining the factor matrices $\mathbf{W}$ and $\mathbf{H}$, we calculate the suspiciousness score of each node within various groups, and return the sorted node list where anomalies are ranked higher than normal objects (from line 10 $\sim$ 15 in Alg. 1).

3.2 Efficient Computation for Sparse Data

Many real-world networks are sparse, meaning that the number of links is usually very small [Zhang and Yu, 2015]. As a result, the gradient computation involving the blocks in $\mathbf{G}$ can be processed more efficiently as below:

$$\begin{aligned} \frac{\partial L_{\mathbf{G}^{i,j}}(\mathbf{W}^i, \mathbf{W}^j)}{\partial \mathbf{W}^i} &= (\mathbf{G}^{i,j} - \mathbf{W}^i \mathbf{W}^{j^T}) \mathbf{W}^j \\ &= (\mathbf{0}_{\mathbf{G}} + \mathbf{S}_{\mathbf{G}} - \mathbf{W}^i \mathbf{W}^{j^T}) \mathbf{W}^j \\ &= -\mathbf{W}^i (\mathbf{W}^{j^T} \mathbf{W}^j) + \sum_{\{r,c,g\} \in \mathbf{S}_{\mathbf{G}}} g \mathbf{e}_r \mathbf{W}^j_{c*} \end{aligned} \tag{7}$$

where $\mathbf{e}_r$ is the column vector of length N_i with one 1 at position r and 0s otherwise. Here $\mathbf{G}^{i,j}$ is separated into two parts, a matrix of all zeros $\mathbf{0}_{\mathbf{G}} \in \mathbb{R}^{N_i \times N_j}$ and a sparse matrix $\mathbf{S}_{\mathbf{G}} = \{(r_1, c_1, g_1), (r_2, c_2, g_2), ...\}$ in the form of the

Algorithm 1: Accelerated Local Anomaly Detection in Attributed Networks

Input: $\mathbf{G}$, $\mathbf{A}$, C, α, γ.
Output: A node list l sorted based on normality score.

1 Initialize $\mathbf{W}$, $\mathbf{H}$, and segment $\mathbf{G}$, $\mathbf{A}$, $\mathbf{W}$ and $\mathbf{H}$;
2 **while** not converged **do**
3 Randomly generate a set of 3-tuples: $\mathcal{S} = \{(i_1, j_1, k_1), (i_2, j_2, k_2), ...\}$ where any two instance sets are interchangeable;
4 **for** $(i, j, k) \in \mathcal{S}$ in parallel **do**
5 $\mathbf{W}'^i \leftarrow \mathbf{W}^i - \epsilon_t \nabla_{\mathbf{W}^i} L_{i,j,k}$
6 $\mathbf{W}'^j \leftarrow \mathbf{W}^j - \epsilon_t \nabla_{\mathbf{W}^j} L_{i,j,k}$ (if $i \neq j$)
7 $\mathbf{H}^k \leftarrow \mathbf{H}^k - \epsilon_t \nabla_{\mathbf{H}^k} L_{i,j,k}$
8 $\mathbf{W}^i \leftarrow \mathbf{W}'^i$, $\mathbf{W}^j \leftarrow \mathbf{W}'^j$ (if $i \neq j$)
9 Non-negativity projection for $\mathbf{W}^i$, $\mathbf{W}^j$ and $\mathbf{H}^k$;
10 Normalize $\mathbf{W}$ and $\mathbf{H}$ according to Eq. (2);
11 **for** $c = 1 : C$ **do**
12 Find members in group c: $group^c = \{n_1^c, n_2^c, ...\}$, so that $\tilde{\mathbf{W}}_{n,c} \geq \tau$ for $n \in group^c$, where τ is the threshold;
13 Compute local normality score $s(n, c)$, $n \in group^c$;
14 Compute $s(n)$ according to Eq. (3) as the global normality score for each $n \in [1, N]$;
15 Return the sorted list of nodes $l = \{n_1, n_2, ..., n_N\}$ so that $s(n_1) \leq s(n_2) \leq ... \leq s(n_N)$.

$(row, column, value)$ list. It is convenient for processing large sparse datasets since $\mathbf{G}^{i,j}$ can be stored in its sparse form throughout the computation. To approximate the matrix in which zero entries are dominant, we can randomly sample only a portion of zero entries as $\mathbf{0}'_{\mathbf{G}}$ for parameters update at each iteration [Devooght *et al.*, 2015], so that:

$$\begin{aligned} \frac{\partial L_{\mathbf{G}^{i,j}}(\mathbf{W}^i, \mathbf{W}^j)}{\partial \mathbf{W}^i} = &- \sum_{\{r',c',0\} \in \mathbf{0}'_{\mathbf{G}}} (\mathbf{W}^i_{r'*} \mathbf{W}^{j^T}_{c'*}) \mathbf{e}_{r'} \mathbf{W}^j_{c'*} \\ &+ \sum_{\{r,c,g\} \in \mathbf{S}_{\mathbf{G}}} (g - \mathbf{W}^i_{r*} \mathbf{W}^{j^T}_{c*}) \mathbf{e}_r \mathbf{W}^j_{c*}. \end{aligned} \tag{8}$$

If we set the size of $\mathbf{0}'_{\mathbf{G}}$ to be comparable to that of $\mathbf{S}_{\mathbf{G}}$, then for each task, the time complexity for parameter update is linear to the number of links included in $\mathbf{G}^{i,j}$, i.e. $\mathcal{O}(\|\mathbf{G}^{i,j}\|_0)$, where $\|\cdot\|_0$ means the 0-norm of a matrix. Similar strategies can be applied for factorizing blocks in $\mathbf{A}$ if it is sparse, then the time complexity for its update is $\mathcal{O}(\|\mathbf{A}^{i,k}\|_0 + \|\mathbf{A}^{j,k}\|_0)$. Therefore, the time complexity of each iteration is $\mathcal{O}(\|\mathbf{G}^{i,j}\|_0 + \|\mathbf{A}^{i,k}\|_0 + \|\mathbf{A}^{j,k}\|_0)$, which is linear to the number of data instances.

4 Experiments

In this section, we conduct experiments for evaluating the effectiveness and efficiency of ALAD compared with other baseline methods. We also perform case studies to illustrate how ALAD achieves interpretability.

4.1 Datasets

The experiments are conducted on both synthetic and real datasets. We generate a series of synthetic attributed networks with known clusters and ground-truth anomalies. The network generation algorithm is based on the partition

model [Perozzi *et al.*, 2014]. Once the total number of nodes N, the number of clusters C and the number of members in each cluster N_C are determined, the adjacency matrix is partitioned into blocks. Let p_{ij} be the link density for $(i,j)^{th}$ block, where $p_{ii} > p_{ij}(i \neq j)$, so that diagonal blocks correspond to actual clusters. The number of connections per node is constrained to follow Zipf's law. We only consider undirected networks here. Unless otherwise stated, we set $p_{i,i} = 0.2$ and $p_{i,j} = 0.5p_{i,i}/C$. To make the problem closer to real situations, we let each community i have a probability of 0.2 to have an exceptionally high connection rate ($p'_{ij} = 40p_{ij}$) to another community j. Each cluster is assigned with a set of key attributes, and nodes within the same cluster are more likely to have similar attribute patterns. Only categorical attributes are considered. For attribute k, its value is drawn from a Bernoulli distribution with success probability of $p_{c,k}$ for cluster c. Key attributes are assigned with larger $p_{c,k}$ value. We set 10% of attributes to be key attributes, and $p_{c,k} = 0.3$ for key attributes and $p_{c,k} = 0.02$ for others. We generate five networks in the experiment. We fix $N = 10{,}000$, $C = 100$, and vary $K = \{100, 200, 300, 400, 500\}$. In each network, 100 anomaly nodes are randomly generated by distorting their attributes.

Besides synthetic networks, we employ four real-world datasets: Disney, Books [Muller *et al.*, 2013], PolBlog [Perozzi *et al.*, 2014] and DBLP [Silva *et al.*, 2012]. *Disney* and *Books* are co-purchase networks extracted from Amazon. *PolBlog* is a citation network of online blogs. *DBLP* is an academic co-authorship network. The first two are used for quantitative comparison between our model and baseline methods, while the rest are applied in case studies. The statistics of these datasets are listed in Table 1. r_a denotes the ratio of anomalies. Numerical attributes are transformed into categorical ones in preprocessing [Catlett, 1991].

4.2 Baseline Methods

The proposed anomaly detection algorithm ALAD is compared with the following baseline methods:

- **CODA** is a community-based anomaly detection algorithm on attributed networks using Hidden Markov Random Fields (HMRF) [Gao *et al.*, 2010].
- **LOF** identifies contextual outliers using only attribute information based on the k-Nearest Neighbors algorithm [Breunig *et al.*, 2000].
- **GOutRank** measures anomaly score of each node based on the statistics of its cluster [Muller *et al.*, 2013].
- **NrMF** applies NMF for low-rank approximation, and uses residuals to measure anomaly score [Tong and Lin, 2011]. Here we use the sum of row-residuals as the anomaly score of each node.
- **GLOB** is the non-clustering version of ALAD, which detects global anomalies instead of local ones.
- **ALAD_TS** is a two-stage variation of ALAD, where we first partition the network only using **G** and then detect outliers in each community using **A**. The h_c of each cluster is the averaged attribute value over all members.

For each method, nodes are ranked in ascending order according to their normality (or abnormal) scores as the output, where a decent detection model would be able to rank the true anomalies earlier (or later) than normal ones. To evaluate the performance quantitatively, we use the Area Under Precision-Recall Curve (AUC-PR) as the metric.

Dataset	N	K	$\lvert\mathcal{E}\rvert$	r_a
Disney	124	347	334	0.048
Books	1,418	1,943	3,695	0.020
PolBlog	362	44,839	2,576	-
DBLP	108,030	23,285	276,658	-

Table 1: Details of the real-world datasets.

	CODA	LOF	GOutRank	NrMF	GLOB	ALAD_TS	ALAD
Disney	0.202	0.309	0.303	0.274	0.144	0.304	0.336
Books	0.058	0.019	0.020	0.052	0.045	0.054	0.061

Table 2: Algorithm performance on the real-world datasets.

4.3 Effectiveness Evaluation

We first analyze the anomaly detection accuracy on synthetic attributed networks as shown in Fig.3. Some observations can be drawn as follows:

- ALAD outperforms baseline methods in most cases. The approaches that utilize both links and attributes information (CODA, NrMF, ALAD_TS, ALAD) generally have better performance, which validates the importance of applying heterogeneous information sources.
- The advantage of ALAD and NrMF is more obvious as the number of attributes increases because of: (1) the promotion of the attributes' roles in discovering neighborhood; (2) the attenuated effect of noise from sampling process so that more accurate community structures are extracted.
- The performance of GLOB is worse than that of most other methods which perform anomaly detection regionally in networks. It thus justifies the importance of identifying anomalies within certain structural or semantic groups.
- The proposed ALAD is better than its two-stage variation, which validates the effectiveness of combining heterogeneous information for jointly determining neighborhoods and their internal attribute distributions.

The experimental results of different methods on real-world networks are presented in Table 2. Similar observations can also be drawn as synthetic data experiments. For examples, approaches accumulating the effect of individual attributes have better performance, and ALAD is better than its two-stage variation. The former phenomenon is more obvious in Books dataset where the network size is relatively larger, which is in accordance with the perception limitation of human labellers since it is more difficult for people to get a panoramic view as the network size grows.

To further investigate the significance of involving links in solving our problem, we remove the adjacency matrix from the datasets, and apply CODA, NrMF and ALAD on only the attribute matrix. Fig. 4 indicates the difference between jointly using heterogeneous information sources and using only attribute information. The absence of structural information weakens the capability of all detectors as reflected by lower AUC scores. It is also worth noting that the performances of CODA and ALAD are similar when only considering attribute information, which is reasonable since they are relevant with the K-means algorithm on attributes in this case.

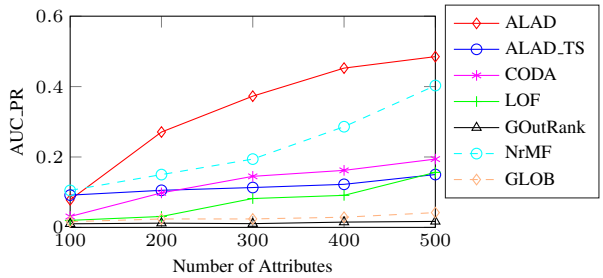

Figure 3: Detection performance on synthetic datasets.

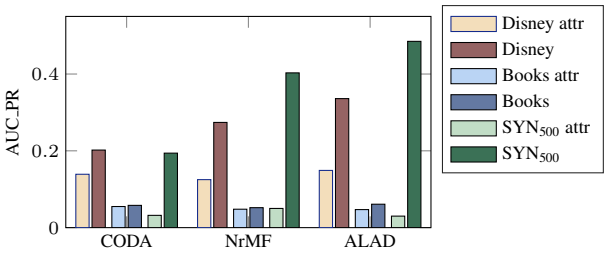

Figure 4: The role of links in identifying anomalies.

	CODA	ALAD_SEQ	ALAD
Disney	2	109	31
Books	107	1,079	270
PolBlog	267	3,242	427
Synthetic_{500}	10,820	8,750	1,350
DBLP	-	23,497	2,550

Table 3: Running time (in seconds) of different methods.

Anomaly (.com)	Abnormal Keywords
thepatriette	whoo, niece, hoo, wishes, dancing, married, wedding
bronxpundit.blog-city	shaolin, fairytale, monks, kant, hamburgers, cuisine
opiniontimes	photon, quantum, mechanics, planet, physics, atomic
ewackos.blogspot	restaurateurs, horsemeat, eatery, chargrilled, steaks

Table 4: Community anomalies in PolBlog dataset.

Anomaly	Key Normal Attributes
A. Sharma: convert, current, output, multipl, inductor, control	wavelet, compress, imag, transform, base, lossless, denois, filter, code,
Marcel Dasen: distribut, router, multicast, activ, architectur, video	semant, servic, mine, inform, integr, extract, page, databas, queri
Andy C. Hung: error, pyramid, quantiz, compress, resili, imag	local, reliabl, system, multi, mobil, multicor, hoc, schedul, ad
G. Manimaran: constrain, syn, flood, tree, multicast, mitig	3d, structur, mdgrape, comput, tflop, molecular, protein, gordon, amyloid

Table 5: Community anomalies in DBLP dataset.

4.4 Efficiency of ALAD

In this section, we study the efficiency of our model. The experiments are run on a Linux machine with 8 Intel i7 CPUs with 3.40GHz. The multiprocessing module in Python is used for parallel data processing. Although theoretically the distributed implementation of optimization processes can reduce the time of each iteration, we can benefit from it only when the computational load per task is not too small that the communication overheads start to dominate.

To evaluate the efficiency of our framework, we compare ALAD with its sequential version ALAD_SEQ and CODA. LOF is not considered because it does not leverage structural information. GOutRank and NrMF are not included because they rely on the clustering result of ALAD. For ALAD_SEQ, different blocks are processed sequentially. When implementing ALAD, we set $B = 8$ for Disney and Books, $B = 16$ for Polblog and Synthetic, and $B = 24$ for DBLP. The synthetic attributed network with 500 attributes is used. The running times of each method on different datasets are shown in Table 3. We do not obtain reasonable results of CODA on DBLP. The result shows that, when the dataset is relatively small, CODA is advantageous in speed since they do not have communication costs. However, as the data size grows larger, ALAD becomes more advantageous than the baseline methods. The speed-up effect of parallel implementation, especially when computational cost on each task is dominant over the system overhead.

4.5 Case Studies

The first case study we consider is on PolBlog dataset. As shown in Table 4, four sampled anomalies are listed for explanations. The attributes of each node are the words appeared in the text samples of blog websites. Here abnormal keywords refer to those with low frequency in the whole dataset corpus, but appear in the anomaly blogger. Therefore, the words of anomalies listed are lack of political flavor. Some of their posts actually focus on other topics such as wedding description (thepatriette.com), physics (opiniontimes.com) or food (ewackos.blogspot.com), which are not commonly discussed in political events. The result of the second case study on DBLP is shown in Table 5. We run ALAD on the network to detect research communities and internal outliers. The top frequent technical terms occurring in each community are shown in the right column. The anomalies are shown in the left column followed by their representative keywords. The anomalies are researchers whose research topic differs from those of their peers in the same community. For example, the papers of Marcel Dasen included in the dataset focus on computer networks, which is a less studied topic in information retrieval community. Therefore, ALAD can effectively summarize the topic distributions of network communities, and spot anomalies with different attribute patterns.

5 Conclusions and Future Work

In this paper, we introduce an effective and efficient framework for identifying anomalies in attributed networks, where heterogeneous information sources are collectively used for determining refined neighborhood and locally assessing the normality of internal node entities. We develop a mini-batch SGD based method to accelerate the optimization and handle large datasets in real-world scenarios. Experiments on real and synthetic datasets indicate the effectiveness and efficiency of our approach. The future extensions of this work include studying the correlation between different attributes, dealing with dynamic networks that evolve over time, as well as detecting anomaly clusters instead of individuals.

Acknowledgments

The work is, in part, supported by DARPA (#N66001-17-2-4031, #W911NF-16-1-0565) and NSF (#IIS-1657196). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

[Akoglu *et al.*, 2012] L. Akoglu, H. Tong, B. Meeder, and C. Faloutsos. Pics: Parameter-free identification of cohesive subgroups in large attributed graphs. *ICDM*, 2012.

[Akoglu *et al.*, 2015] L. Akoglu, H. Tong, and D. Koutra. Graph-based anomaly detection and description: a survey. *Data Min. Knowl. Discov.*, 2015.

[Breunig *et al.*, 2000] M. Breunig, H. Kriegel, R. Ng, and J. Sander. LOF: identifying density-based local outliers. *SIGMOD*, 2000.

[Catlett, 1991] J. Catlett. On changing continuous attributes into ordered discrete attributes. *EWSL*, 1991.

[Chandola *et al.*, 2009] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *CSUR*, 2009.

[Chang *et al.*, 2015] S. Chang, W. Han, J. Tang, G. Qi, C. Aggarwal, and T. Huang. Heterogeneous network embedding via deep architectures. *KDD*, 2015.

[Chen *et al.*, 2016] T. Chen, L. Tang, Y. Sun, Z. Chen, and K. Zhang. Entity embedding-based anomaly detection for heterogeneous categorical events. *IJCAI*, 2016.

[Cheng *et al.*, 2016] W. Cheng, K. Zhang, H. Chen, G. Jiang, Z. Chen, and W. Wang. Ranking causal anomalies via temporal and dynamical analysis on vanishing correlations. *KDD*, 2016.

[Devooght *et al.*, 2015] R. Devooght, N. Kourtellis, and A. Mantrach. Dynamic matrix factorization with priors on unknown values. *KDD*, 2015.

[Gao *et al.*, 2010] J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. *KDD*, 2010.

[Gemulla *et al.*, 2011] R. Gemulla, E. Nijkamp, P. Haas, and Y. Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. *KDD*, 2011.

[Hawkins, 1980] D. Hawkins. *Identification of outliers*, volume 11. Springer, 1980.

[Huang *et al.*, 2017a] X. Huang, J. Li, and X. Hu. Accelerated attributed network embedding. 2017.

[Huang *et al.*, 2017b] X. Huang, J. Li, and X. Hu. Label informed attributed network embedding. 2017.

[Jiang *et al.*, 2015] M. Jiang, A. Beutel, P. Cui, B. Hooi, S. Yang, and C. Faloutsos. A general suspiciousness metric for dense blocks in multimodal data. *ICDM*, 2015.

[Koren *et al.*, 2009] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 2009.

[Kuang *et al.*, 2012] D. Kuang, C. Ding, and H. Park. Symmetric nonnegative matrix factorization for graph clustering. *SDM*, 2012.

[Lee and Seung, 2001] D. Lee and S. Seung. Algorithms for non-negative matrix factorization. *NIPS*, 2001.

[Li *et al.*, 2017] J. Li, H. Dani, X. Hu, and H. Liu. Radar: Residual analysis for anomaly detection in attributed networks. 2017.

[Liang and Parthasarathy, 2016] J. Liang and S. Parthasarathy. Robust contextual outlier detection: Where context meets sparsity. 2016.

[Liang *et al.*, 2017] J. Liang, P. Jacobs, and S. Parthasarathy. Seano: Semi-supervised embedding in attributed networks with outliers. *arXiv preprint*, 2017.

[Long *et al.*, 2006] B. Long, Z. Zhang, X. Wu, and P. Yu. Spectral clustering for multi-type relational data. *ICML*, 2006.

[Muller *et al.*, 2013] E. Muller, P. Sánchez, Y. Mulle, and K. Bohm. Ranking outlier nodes in subspaces of attributed graphs. *ICDEW*, 2013.

[Perozzi *et al.*, 2014] B. Perozzi, L. Akoglu, P. Iglesias Sánchez, and E. Müller. Focused clustering and outlier detection in large attributed graphs. *KDD*, 2014.

[Rayana and Akoglu, 2015] S. Rayana and L. Akoglu. Collective opinion spam detection: Bridging review networks and metadata. 2015.

[Shashanka *et al.*, 2008] M. Shashanka, B. Raj, and P. Smaragdis. Probabilistic latent variable models as nonnegative factorizations. *Comput Intell Neurosci*, 2008.

[Silva *et al.*, 2012] A. Silva, W. Meira Jr, and M. Zaki. Mining attribute-structure correlated patterns in large attributed graphs. *VLDB*, 2012.

[Tong and Lin, 2011] H. Tong and C. Lin. Non-negative residual matrix factorization with application to graph anomaly detection. *SDM*, 2011.

[Xiong *et al.*, 2011] L. Xiong, X. Chen, and J. Schneider. Direct robust matrix factorizatoin for anomaly detection. *ICDM*, 2011.

[Xu *et al.*, 2003] W. Xu, X. Liu, and Y. Gong. Document clustering based on non-negative matrix factorization. *SIGIR*, 2003.

[Xu *et al.*, 2007] X. Xu, N. Yuruk, Z. Feng, and T. Schweiger. Scan: a structural clustering algorithm for networks. *KDD*, 2007.

[Yang *et al.*, 2012] C. Yang, R. Harkreader, J. Zhang, S. Shin, and G. Gu. Analyzing spammers' social networks for fun and profit: a case study of cyber criminal ecosystem on twitter. *WWW*, 2012.

[Zhang and Yu, 2015] J. Zhang and P. Yu. Integrated anchor and social link predictions across social networks. *IJCAI*, 2015.

[Zhao and Fu, 2015] H. Zhao and Y. Fu. Dual-regularized multi-view outlier detection. *IJCAI*, 2015.

[Zimek *et al.*, 2012] A. Zimek, E. Schubert, and H. Kriegel. A survey on unsupervised outlier detection in high-dimensional numerical data. *Statistical Analysis and Data Mining*, 2012.