Overlapping Community Detection in Temporal Text Networks

ABSTRACT

Network is a powerful language to represent relational data. One way to understand network is to analyze groups of nodes which share same properties or functions. The task of discovering such groups is known as community detection. Generally, there are two types of information that can be utilized to fulfill this task, i.e., the link structures and the node attributes. The temporal text network is a special kind of network that contains both sources of information. Typical representatives include online blog networks, the World Wide Web (WWW) and academic citation networks.

In this paper, we study the problem of overlapping community detection in temporal text network. We gather a large set of 32 temporal text networks with reliable ground-truth communities. By examining such networks, we find that a large proportion of edges connect two nodes which share no community in common. This scenario cannot be modeled by practically all existing community detection methods. Besides, we quantitatively analyze how node attributes help to improve the quality of detected communities and discover that nodes in the same community share similar textual contents. Motivated by these empirical observations, we propose MAGIC (Model Affiliation Graph with Interacting Communities), a generative model which captures community interactions and considers the information from both link structures and node attributes. Experimental results show that MAGIC achieves at least 40% relative improvements over 5 state-of-the-art methods in terms of 4 widely-used metrics.

CCS Concepts

• Data Mining → Graph Mining; • Networks → Network Clustering;

Keywords

Community Detection, Text Network, Temporal Text Network, Network Analysis

1. INTRODUCTION

Network can serve as a powerful language to represent relational information among data objects from social, natural and academic domains [31]. One way to understand network is to identify and analyze groups of nodes which share same properties or functions. Such groups of nodes can be from the same organization in social networks [15], proteins with similar functionality in biochemical networks [10], and papers from the same scientific fields in citation networks [19]. The research task of discovering such groups is known as the community detection problem [33]. Traditional methods [15, 20] mainly focus on finding disjoint communities and are all based on a restrictive assumption that each node can only belong to one single community. By relaxing this assumption, the overlapping community detection problem becomes more general and has attracted major attention recently [17, 25].

There are generally two types of information that can be utilized to discover overlapping communities [30]. The first is the link structure, i.e., the presence and absence of edges. Classical methods [17, 2, 1] usually focus on this type of information and aim to extract a group of nodes with more links inside the group than between its members and outside the group [16]. The second type of information is the node attribute, including online profiles of users, pre-existing features of proteins, and textual contents of papers. Due to the prevalent noise in link structures, the approaches for detecting community based on both types of information [19, 30] have gained increasing popularity.

In this paper, we study the problem of overlapping community detection in temporal text networks. A temporal text network is a directed network in which each node has textual content and temporal information. Such networks are ubiquitous in the real world. Typical representatives include online blog networks, the World Wide Web (WWW), email correspondence networks, and academic citation networks. Identification of meaningful communities in temporal text networks provides useful knowledge for subsequent applications such as domain-specific ranking and user-targeted recommendation.

The contributions of our work are three-folded. First, we gather a large collection of 32 temporal text networks with ground-truth communities that are collected from different domains and of varying scales. They enable us to derive insights of community structures and allow us to quantitatively evaluate community detection methods. Second, we study the interactions among ground-truth communities in temporal text networks and discover that a large proportion
of nodes share a link due to community interactions. We also analyze how node attributes help to improve the quality of detected communities and find that nodes in the same communities share similar textual content. Third, based on the empirical observations, we propose MAGIC (Model Affiliation Graph with Interacting Communities), a probabilistic generative model which utilizes all sources of information in the temporal text network and scales to network with millions of nodes.

**Present work: Networks with Ground-Truth Communities.** We generate a large set of 32 temporal text networks with reliable ground-truth communities based on Microsoft Academic Graph (MAG) [22]. The MAG dataset contains over 100 million scientific papers with titles, references, publish time, and sets of “Field of Study” (FoS). Totally, there are over 50,000 FoS labels, organized in a four-level hierarchical manner, starting from top L0 levels such as “Mathematics”, “Physics”, “Computer Science” to middle L1 levels such as “Statistics”, “Quantum mechanics”, “Data mining”, and ending with bottom L3 levels such as “Complex manifold”, “Oseen equations”, and “K-optimal pattern discovery”.

We construct a temporal text network by sampling an academic citation network for each L1 level FoS under Computer Science (CS) field. We further define each FoS label as a ground-truth community since all members (i.e., papers) of the same community are in the same subarea of science and possess the same property. Besides, we treat the publish time and title of each paper as its corresponding temporal and textual attributes. Since the title of each paper has only 6 to 7 words on average, we further enhance the textual content by crawling over 9 million papers in CS Fields using the URLs provided in original MAG datasets. All our datasets and data generation codes are available online[1] for the purpose of reproducible learning.

**Present work: Empirical Observations.** The availability of temporal text networks with ground-truth communities enables us to derive insights of the community structure. In this paper, we study the interactions among ground-truth communities and discover that a large number of nodes share a link because of the community interactions. We find that a large proportion of edges connect a pair of nodes which have no communities in common. Current methods [28, 30, 31] fail to identify such community interactions and fail to model this scenario well.

We also quantitatively analyze how textual contents provide useful information for overlapping community detection. We find that nodes in the same communities share very similar textual content, which is very intuitive.

**Present work: Community Detection in temporal text network.** Based on the above empirical observations, we propose MAGIC (Model Affiliation Graph with Interacting Communities), a generative model which models the probability of an edge between two nodes as a function of the communities they share, the interactions among communities they are affiliated in, and the time information of each node. MAGIC captures community interactions and considers the information from both link structures and node attributes. MAGIC further reduces the noise of missing links by utilizing the time information attached on each node. By fitting MAGIC toward a given temporal text network, we can detect meaningful communities.

Experimental results show that MAGIC achieves at least 40% relative improvements over 5 state-of-the-art methods [17, 2, 28, 30, 31] in terms of several widely-used metrics [11, 14, 28]

**Organization.** The rest of this paper is organized as follows. Section 2 summarizes the related work. Section 3 gives formal definitions of temporal text network and the task of overlapping community detection. Section 4 presents our empirical observations. Section 5 introduces the MAGIC along with its learning method. Finally, we report the experimental results in section 6 and conclude in section 7.

2. RELATED WORK

Overlapping community detection has been extensively investigated in the last decade [25]. Classical methods such as CPM [17], MMSB [2], and LC [1] are mainly based on dense subgraph extraction. For example, CPM aims to find all k-cliques and combine those cliques sharing k – 1 nodes to be communities. Consequently, these methods are not applicable for detecting communities in large-scale networks with millions of nodes.

More recently, a series of affiliation graph models [26, 28, 29, 31] are proposed based on the idea that communities arise due to shared group affiliations [5]. Yang and Leskovec introduced Community-Affiliation Graph Model (AGM) [26] in which nodes are affiliated with latent communities they belong to and links are generated based on node community affiliations. They later relaxed the combinatorial optimization problem of fitting AGM and presented a more scalable model called BIGCLAM [28]. This line of works, however, models the underlying affiliation network as a bipartite graph and assumes each community creates edges independently. Compared to these methods, MAGIC relaxes such assumption and captures community interactions.

Another piece of work which also considers community interactions is BNMTF [34]. This method factorizes the adjacency matrix of network into latent factors which are regarded as communities. However, BNMTF keeps using conventional Euclidean distance and generalized KL-divergence as the objective of matrix factorization, which is not scalable and causes bad interpretability.

Many models also study the problem of overlapping community detection in the context of combining link structure with node attributes [30, 19, 13]. A large catalog of such models are based on topic models [21, 32, 3]. However, these methods do not allow a node to have high membership strength in multiple communities simultaneously and therefore leads to unrealistic assumptions about the structure of community overlaps [29]. To solve this problem, authors in [29] proposed CESNA which is an affiliation graph model based on BIGCLAM and uses a logistic model to generate binary-valued node attributes. CESNA models the generations of node attributes and link structures as two different mechanisms. Compared to CESNA, MAGIC takes a more unified approach to model these two types of information.

3. PROBLEM FORMULATION

In this section, we formalize the problem of overlapping community detection in temporal text networks. We first define the “text network” and “temporal text network”. Then, we discuss a method to explicitly encode text information
in graph and define the “projected temporal text network”. Figure 1 illustrates the relationship and difference among these three types of networks.

Definition 1. (Text Network) A text network is defined as a directed unweighted graph \( G = (V, E) \), where \( V \) is a set of vertices and \( E \) is the set of edges between the vertices. Each vertex \( v \in V \) represents a document and has a sequence of words associated with it. Each edge \( (u, v) \in E \) represents the directed connection between document \( u \) and document \( v \).

The text network captures the relationship among documents and models it explicitly. Such network is ubiquitous in the real world. Online blog networks, email correspondence networks and academic citation networks are some good representatives.

Definition 2. (Temporal Text Network) A temporal text network is a text network with time information, denoted as \( G = (V, E; T) \). In temporal text network, each vertex \( v \in V \) is attached with a timestamp \( t(v) \). Furthermore, a temporal text network is called natural temporal text network if each edge \( (u, v) \) satisfies \( t(u) < t(v) \); otherwise, it is called complex temporal text network.

The temporal text network encodes the time information of each document. We state that most text networks are natural temporal text networks, provided that we give a proper definition of the edge direction. For example, if we define an edge in citation network starting from the cited paper to the citing one, then this network is natural because nobody can cite future papers.

Definition 3. (Projected Temporal Text Network) A projected temporal text network, denoted as \( G^P = (V \cup V_w, E \cup E_{wd}; T \cup T_w) \), is a transformation of original temporal text network \( G = (V, E; T) \). Each additional vertex \( v \in V_w \) represents a word and each additional edge \( (w_i, d_j) \in E_{wd} \) indicates that word \( w_i \) exists in document \( d_j \). We set the timestamps of all word vertices to be zero.

Such projection method is proposed in [23] and proved useful to model document-word dependency. We note that the projected temporal text network captures the document-level word co-occurrences but discards the word frequency. A straightforward method to encode such frequency information is later discussed in the experiment.

Definition 4. (Overlapping Community Detection in Temporal Text Networks) Given a temporal text network \( G = (V, E; T) \), the problem of overlapping community detection in temporal text network is to find a collection of subsets of \( V \) denoted by \( C = \{C_1, \ldots, C_K\} \) such that for each \( C_i \in C \), its induced subgraph \( G[C_i] \) forms a network community. By allowing \( C_i \cap C_j \neq \emptyset \), we can obtain overlapping communities.

Finally, we state that the problem investigated in this paper is overlapping community detection in temporal text networks. The projected temporal text networks only serves as a good proxy for the original network. We will later elaborate their differences and discuss how we detect meaningful communities in the temporal text network by exploiting information in its corresponding projected version.

4. EMPIRICAL OBSERVATION

In the section, we first describe how we generate a large collection of temporal text networks and define reliable ground-truth communities. Then, we present our empirical observations by answering two important questions. How many edges connect two nodes that share no common community? How textual contents improve the quality of detected communities? Finally, we discuss the importance of such findings and how they motivate the development of our model.

4.1 Dataset descriptions

We generate temporal text networks with explicit ground-truth communities based on Microsoft Academic Graph (MAG) [22]. The MAG dataset contains over 100 million scientific papers with their titles, references, publish time, and sets of “Field of Study” (FoS) labels. In total, there are over 50 thousands different FoS labels, organized in a four-level hierarchical manner as demonstrated in Figure 2a. Such FoS labels naturally correspond to ground-truth communities since all members (i.e., papers) of the same community are in the same subarea of science and possess the same
property. Therefore, we define the FoS labels as the ground-truth communities and further treat the publish time and title of each paper as its temporal and textual attributes.

We construct a temporal text network by sampling an academic citation network. To illustrate the sampling process, we take the “Information Retrieval” (IR) field as an example. We consider that a paper is in IR field if it contains at least one FoS label in the set of IR-related FoS labels. A FoS is IR-related if it locates in the FoS tree rooted by the “Field of Study” named “Information Retrieval” (IR), as shown in Figure 2a. Then, we construct a citation network among all these selected papers and delete those with no reference and no citation. We repeat this process for 32 L1 level FoS under Computer Science (CS) field. These networks cover a wide range of domains and the sizes of them ranges from thousands to millions of nodes. Table 1 summarizes the networks we studied.

We note that the authors in [27] also studied networks with ground-truth communities. However, they considered each connected component of the group as a separate ground-truth community, which is unreasonable. To prove this point, we select three datasets in [27], run BIGCLAM [28] on them, and record F1 score during iterations. BIGCLAM is a method proposed by the same authors of [27]. Higher value of F1 score means the detected communities are more accurate. Figure 2b shows the results. When the number of iterations is increasing, the performance of BIGCLAM is actually decreasing. This is totally different from the results reported in [28]. The reason behind this phenomenon is that BIGCLAM adopts a very local community detection method for the model initialization. Therefore, when the ground-truth communities are defined as local connected components, those local methods will take advantage of this point and achieve unreasonable performance due to such bias.

4.2 Empirical Observation

First, we analyze how textual contents help to provide useful information for community detection. For each community, we select two nodes and calculate the Jaccard similarity of their textual contents. The higher this value is, the more similar the textual contents are. We repeat this process for all possible pairs of nodes and get the average Jaccard similarity for that community. Therefore, when the ground-truth communities are defined as local connected components, those local methods will take advantage of this point and achieve unreasonable performance due to such bias.

We can see, the average Jaccard similarity of each community is much higher than that of a randomly selected set. This clearly demonstrates that nodes in the same community have similar textual contents.

Next, we study community interactions by asking the question that how many edges connect two nodes that share no common community? These edges are caused by community interactions and thus we name them interaction edges. They reveal the overall amount of community interactions in each dataset. Results are shown in Figure 4. As we can see, most of networks have more than 20% of edges that are between two nodes with no common community. Besides, the ratio of such edge has an increasing trend with regard to the network size.

We then study such community interactions in a finer granularity. For each edge \((u, v)\), if node \(u\) and \(v\) have some communities in common, we assume this edge is generated only because two nodes share same communities. On the other hand, if node \(u\) and \(v\) have no common community, then this edge must be generated by community interactions. We formalize this idea as following. Let \(C(u)\) denotes the set of communities of node \(u\). For edge \((u, v)\), if \(C(u) \cap C(v) \neq \emptyset\), then for each community \(c \in C(u) \cap C(v)\), we add its Internal Connectivity (IC) by \(\frac{1}{|C(c)||C(c)|}\). Otherwise, if \(C(u) \cap C(v) = \emptyset\), then for each community \(c \in C(u) \cup C(v)\), we add its External Connectivity (EC) by \(\frac{1}{2|C(c)|} \) if \(c \in C(u)\) or \(\frac{1}{2|C(c)|} \) if \(c \in C(v)\). An illustrative example is shown in Figure 5.

After iterating all edges in the network, we can get IC score and EC score of each community. We then define the Interaction Ratio of community \(c\) as \(\frac{EC(c)}{IC(c) + EC(c)}\). Here we ignore all community interactions caused by edges linking two nodes that share the same communities. Take Figure 5 as an example. We ignore possible community interactions between communities \(c_1\) and \(c_3\) or communities \(c_2\) and \(c_4\). We presume the edge \((u, v)\) is generated only due to the Internal connectivity of community \(c_1\) and \(c_3\). Therefore, the Interaction ratio measures the minimum amount of interaction for each community.

As shown in Figure 7, communities have strong interactions. This result is in retrospect, very intuitive. For example, papers in “Information Retrieval” field may adopt the techniques from “Natural Language Processing” papers.
Figure 3: Comparison between the average Jaccard similarity of two randomly selected nodes and that of two nodes from the same community in four temporal text networks.

Figure 4: Ratio of interaction edges in 32 temporal text networks of different scales.

Figure 5: Community affiliation graph. Circles represent nodes in the observed network. Squares represent the latent communities. IC denotes Internal Connectivity and EC denotes External Connectivity.

for semantic search. An algorithm published in a “Machine Learning” conference has its origin from a “Mathematics” problem and been widely used in “Data Mining” field. Consequently, if we find two papers with one in “Data Mining” community and another in “Machine Learning” community, the probability that they share a link should not be modeled as zero, as practically all existing methods do [28, 30, 31]. Instead, we should consider the community interactions and model them explicitly.

5. COMMUNITY DETECTION IN TEMPORAL TEXT NETWORK

Motivated by previous observations, we present MAGIC (Model Affiliated Graph with Interacting Communities), a probabilistic generative model which models the community interactions explicitly. Then, we discuss how MAGIC utilizes the information from both link structures and node attributes. Finally, we explain how to detect overlapping communities in temporal text network by learning MAGIC.

5.1 Model Description

MAGIC is based on the idea that communities arise due to shared group affiliation [5, 6], and views the whole network as a result generated by a variant of the community-affiliation graph model [26]. Same as the original one, MAGIC models the community affiliation strength between each pair of node $u$ and community $c$ with a nonnegative parameter $F_{uc}$ ($F_{uc} = 0$ means node $u$ is definitely not affiliated to community $c$). MAGIC differs mainly in how we model the latent affiliation network. The original community-affiliation graph model treats the affiliation network as a bipartite graph, which fails to capture those important interactions among communities. MAGIC, instead, explicitly models the community interaction strength between every pair of community $c_i$ and $c_j$ with a nonnegative parameter $\eta_{ij}$ ($\eta_{ij} = 0$ indicates community $c_i$ and $c_j$ definitely have no relationship). Finally, we model the influence degree of each community $c_i$ with the parameter $\eta_i$. Such influence degrees measure the probability that two nodes in the same commu-

$
\begin{align*}
EC(c_i) &= 0.25 \\
EC(c_i) &= 0.25 \\
EC(c_i) &= 0.5 \\
IC(c_2) &= 0.5 \\
IC(c_3) &= 0.5 \\
\end{align*}$

$
\begin{align*}
\eta_{ij} &= 0.29 \\
\eta_{ij} &= 0.30 \\
\eta_{ij} &= 0.31 \\
\eta_{ij} &= 0.32 \\
\end{align*}$

\begin{align*}
\text{Interaction ratio of each community in four temporal text networks.}
\end{align*}
Given the parameters, MAGIC generates a link \((u \rightarrow v)\) with the probability \(p(u \rightarrow v)\) defined as follows:

\[
p(u \rightarrow v) = \left(1 - \exp(-\sum_{i,j} F_{ui} \eta_{ij} F_{vj})\right) \delta(u \rightarrow v)
\]

\[= \left(1 - \exp(-F^T_u \eta F_v)\right) \delta(u \rightarrow v),\]

where \(F_u\) is a column vector representing the community affiliation strength for node \(u\), \(\eta\) is the community interaction matrix, and \(\delta(u \rightarrow v)\) is the weighting function defined only on the timestamps of nodes \(u\) and \(v\). The introduction of \(\eta\) and \(\delta\) explicitly models the community interaction and utilizes the node temporal information. In this paper, we mainly focus on the natural temporal text network and thus the weighting function \(\delta\) is defined as:

\[
\delta(u \rightarrow v) = \begin{cases} 
1 & \text{if } t(u) < t(v) \\
0 & \text{otherwise}
\end{cases}
\]

Eq. 2 essentially restricts the generation of an edge starting from a node with early timestamp and ending with a node with later timestamp. This constraint is ubiquitous in the real world. We cannot cite a paper published in future nor forward an unreceived email.

Next, we discuss how MAGIC utilizes the text information. Instead of treating words and documents separately and using different mechanisms to generate them \([30]\), we adopt a more unified approach. We first construct a projected temporal text network corresponding to the original and then applied MAGIC to this projected network. A projected temporal text network is intrinsically a heterogeneous network with two types of nodes — “document-node” and “word-node”. MAGIC treats them in the same way and will learn a feature vector representing the latent community affiliation strength for each document and word.

Finally, MAGIC learns the community affiliation matrix \(F\) and the community interaction matrix \(\eta\) by maximizing the log likelihood of the observed network \(G\):

\[
\hat{F}, \hat{\eta} = \arg\max_{F \geq 0, \eta \geq 0} l(F, \eta),
\]

where nonnegative matrices \(F \in \mathbb{R}^{K \times N}\), \(\eta \in \mathbb{R}^{K \times K}\) and \(K, N\) denote the number of communities and nodes, respectively. The log likelihood can be further written out as below:

\[
l(F, \eta) = \sum_{(u \rightarrow v) \in E} \log(1 - \exp(-F^T_u \eta F_v)) - \sum_{(u \rightarrow v) \not\in E} F^T_u \eta F_v
\]

Notice here we explicitly add the time constraint in the second term so that the absence of an edge \((u \rightarrow v)\) with \(t(u) \geq t(v)\) will not contribute to the likelihood. As demonstrated in Figure 7a, there are two possible reasons for a missing link \((u, v)\). If we find \(t(u) < t(v)\), which means this edge could have been generated, then the absence of such edge can provide some useful information and we define such edge as an unobserved link. Otherwise, if \(t(u) \geq t(v)\), then the absence of this link carries no information because it cannot be generated anyway. Therefore, we define such edge as an impossible link. We can see from Eq. 4 that MAGIC only uses the information derived from observed links and unobserved links.

![Figure 7: (a) Illustration of three types of links. Due to the time order, a missing link can be either an unobserved link or an impossible link. (b) Illustration of two types of neighbors.](image)

### 5.2 Parameter Learning

To solve the optimization problem defined in Eq. 3, we adopt a block coordinate gradient ascent approach. We first update the community affiliation strength \(F_u\) for each node \(u\) with both \(\eta\) and \(F\) for all other nodes \(v \neq u\) fixed. Then, we update the community interaction matrix \(\eta\) with the community affiliation matrix \(F\) fixed. We can see each subproblem is a convex optimization problem which makes efficient algorithm possible.

To update the community affiliation strength \(F_u\) for node \(u\), we solve the following subproblem:

\[
F_u = \arg\max_{F_u \geq 0} l(F_u),
\]

where \(l(F_u)\) is the part of \(l(F, \eta)\) defined in Eq. 4 that involves \(F_u\), i.e.,

\[
l(F_u) = \sum_{v \in \text{in}(u)} \log(1 - \exp(-F^T_u \eta F_u)) - \sum_{v \in \text{out}(u)} F^T_u \eta F_v
\]

\[
+ \sum_{v' \in \text{out}(u)} \log(1 - \exp(-F^T_{u'} \eta F_u)) - \sum_{v' \in \text{out}(u)} F^T_{u'} \eta F_{v'}
\]

where \(\text{in}(u)\) denotes the set of in-neighbors of node \(u\) and \(\text{out}(u)\) denotes the set of out-neighbors of node \(u\). \(N(u)\) is equal to \(\text{in}(u) \cup \text{out}(u)\), as demonstrated in Figure 7b. This subproblem can be further solved by projected gradient ascent \([12]\).

\[
F^\text{new}_{uk} = \max\{0, F^\text{old}_{uk} + \alpha F_{uk} (\nabla l(F_u))_k\},
\]

where \(\alpha\) is the step size computed by backtracking line search \([4]\), and the gradient is:

\[
\nabla l(F_u) = \sum_{v \in \text{in}(u)} \frac{\exp(-F^T_u \eta F_u)}{1 - \exp(-F^T_u \eta F_u)} \eta F_v - \sum_{v \in \text{out}(u)} \eta F_v
\]

\[
+ \sum_{v' \in \text{out}(u)} \frac{\exp(-F^T_{u'} \eta F_u)}{1 - \exp(-F^T_{u'} \eta F_u)} \eta F_{v'} - \sum_{v' \in \text{out}(u)} \eta F_{v'}
\]

After the community affiliation matrix \(F\) updated, we fix \(F\) and update the community interaction matrix \(\eta\). Notice
that \( \eta \) is involved in every term of Eq. 4 and thus we solve it directly.

\[
\eta_{ij}^{\text{new}} \leftarrow \max\{0, \eta_{ij}^{\text{old}} + \alpha_{\eta} (\nabla l(F, \eta))_{ij}\}, \tag{9}
\]

where the step size \( \alpha_{\eta} \) is also calculated by backtracking line search, and the gradient for \( \eta \) is:

\[
\nabla l(F, \eta) = \sum_{(u \rightarrow v) \in E} \frac{\exp(-F^T_u \eta F_v)}{1 - \exp(-F^T_u \eta F_v)} F_u F_v^T - \sum_{(u \rightarrow v) \notin E} F_u F_v^T. \tag{10}
\]

We notice from Eqs. (8) and (10) that direct computations of \( \nabla l(F, \eta) \) and \( \nabla l(F_u) \) take \( O(N) \) and \( O(N^2) \) time, respectively. To reduce the time complexity and increase scalability, we adopt the following tricks:

\[
\eta^T F_u = \sum_{v \in \text{in}(N(u))} \sum_{t(v) < t(u)} \eta_v^T F_v - \sum_{v \in \text{in}(N(u))} \eta_v^T F_v, \tag{11}
\]

\[
\eta^T F_v = \sum_{v' \in \text{out}(N(u))} \sum_{t(v') > t(u)} \eta_v^T F_v' - \sum_{v' \in \text{out}(N(u))} \eta_v^T F_v', \tag{12}
\]

\[
\sum_{(u \rightarrow v) \notin E} F_u F_v^T = \sum_{(u \rightarrow v) \notin E} F_u F_v^T - \sum_{(u \rightarrow v) \notin E} F_u F_v^T. \tag{13}
\]

In this way, we can compute \( \nabla l(F_u) \) in \( O(N(u)) \) by caching the first term in the right hand side of Eqs. (11) and (12), and compute \( \nabla l(F, \eta) \) in \( O(|E|) \) by caching the first term in the right hand side of Eq. 13. We notice that the combined time complexity for updating the whole \( F \) is \( O(|E|) \). Therefore, we conclude that the time complexity of each iteration for MAGIC is \( O(|E|) \).

5.3 Other Issues

5.3.1 Model Initialization

To initialize \( F \), we extend the method in [8] to directed network. The conductance in directed network is also defined in [7]. The in-neighbors \( \text{in}(N(u)) \) of node \( u \) is locally minimal if \( \text{in}(N(u)) \) has lower conductance than all in-neighbors \( \text{in}(N(v)) \) where node \( v \in \text{out}(N(u)) \). For a node \( u' \) belonging to such a locally minimal neighborhood \( k \), we initialize \( F_{u'k} = 1 \), otherwise we let \( F_{u'k} = 0 \). To initialize \( \eta \), we set the entries in the main diagonal as 0.9 and all other entries to be 0.1.

5.3.2 Determining community membership

After learning parameters \( \hat{F} \) and \( \hat{\eta} \), we need to determine the “hard” community membership of each node. We achieve this by thresholding the learned \( \hat{F} \) with a set of \( \{\delta_k\} \), one for each community \( k \). The basic intuition is that if two nodes belong to the same community \( k \), then the probability of having an link between them through community \( k \) is larger than \( 1/N \), where \( N \) is the number of nodes. Following this idea, we can obtain \( \delta_k \) as below:

\[
\delta_k = \sqrt{-\log(1 - l/|N|)} \tag{14}
\]

With \( \{\delta_k\} \) obtained, we consider node \( u \) belonging to community \( k \) if \( F_{uk} \geq \delta_k \).
MMSB (Mixed-Membership Stochastic Block Model): MMSB [2] is originally proposed as a variational inference algorithm for fast approximating posterior inference and has later been applied to detect overlapping communities in [9]. For MMSB, we set the number of communities K to be the same as that in MAGIC. Besides, we note that MMSB returns the stochastic node membership to each community and thus we need to further map it to the “hard” community membership. We follow the convention in [28] and assign a node to a community if its corresponding stochastic membership is non-zero. We use the implementation of MMSB in [9].

BIGCLAM (Cluster Affiliation Model for Big Networks) BIGCLAM [28] is a variant of affiliation graph model for detecting overlapping communities. They used the non-negative matrix factorization method to increase scalability. We use the implementation of BIGCLAM in SNAP.

CSENA (Communities from Edge Structure and Node Attributes) CSENA [30] is another variant of affiliation graph model designed for a network with node attributes. For CSENA, we choose the number of communities to be the same as that in MAGIC and use the implementation of BIGCLAM in SNAP.

MAGIC (Model Affiliation Graph with Interacting Communities) MAGIC is our proposed method for detecting Interacting communities. There are three variants of MAGIC that use different combinations of information sources. We denote MAGIC(all) for the variant that uses both link structures and node attributes, MAGIC(net) for the one that uses only information in link structures, and MAGIC(raw) for the version that ignores edge directions and runs on undirected networks.

Previously BIGCLAM has been shown to outperform NMF [18] [24] and CSENA has been shown to outperform CODICIL [19] and Block-LDA [3]. Therefore, we do not compare with those algorithms.

6.1.3 Metrics

We denote the set of ground-truth communities as C and the set of detected communities as Ĉ. To measure the performance of our model, we select following 4 metrics.

Coverage ratio is the ratio of nodes which can be assigned to at least one community by the model. Intuitively, a model cannot be useful if it can only detects communities for a very few proportion of nodes.

F1 score is the average of the F1 score of the best-matching ground-truth community to reach detected community.

Normalized Mutual Information (NMI) is a measure of similarity borrowed from information theory. Later, it is extended to measure the quality of overlapping communities. Please refer to [11] for details.

NMI-max is a metric based on the NMI described former. It revises some unintuitive behaviors by using a conventional normalization and demonstrates more intuitive behaviors according to the empirical observation. Please refer to [14] for details.

For all 4 metrics higher values mean that the detected communities are more accurate and have better qualities.

6.2 Quantitative Results

6.2.1 Performance on small scale networks

Figure 8 compares the performance of 6 methods on 7 small scale networks in terms of four metrics. For each evaluation metric, we scale it to make sure the best community detection method will get the score 1. Then, we compute the final score for each method by summing up all four normalized scores. This final score is used to compare the composite performance of each method. We can see that MAGIC(all) achieves the best performance in 5 out of 7 networks and MAGIC(net) gets best results in the rest of two. The average composite performance of MAGIC(all) is 3.66, which is 137% higher than BIGCLAM(1.54), 40% higher than CESNA(2.60), 648% higher than CPM(0.45), and 192% higher than MMSB(1.25).

Then, we analyze the results in details. First, we find MAGIC(net) bests MAGIC(all) in first two networks, which are two smallest networks. This is because when networks are small, the noise eliminated by the introduction of node attributes cannot compensate the noise in node attributes themselves. Second, we find that the coverage ratio of CPM is extremely small. This is because when networks are sparse, which is common in the real world, CPM will ignore all ver-
tess which cannot form a clique with size 4. We tried to decease the clique size but only find the composite performance becomes even lower. Finally, we observe that NMI scores for some methods are just zero. This phenomenon is quite common for large networks. When the gap between ground-truth communities and detected communities are too big, the NMI will be set to zero directly [11].

6.2.2 Performance on large scale networks

We further conduct experiments on 9 large scale networks, each network contains at least hundreds of thousands of nodes and the largest one contains nearly 2 million nodes and over 30 million edges. Some methods including CPM and CESNA cannot scale to such big networks. CPM is known for its bad scalability but unfortunately we are not able to run CESNA on large scale networks as well. Surprisingly, the speed of MMSB is not as bad as that reported in [28] and thus we finally compare four baseline methods for large-scale networks including BIGCLAM, MMSB, MAGIC(net) and MAGIC(all).

Table 3 reports the final unnormalized scores. First, we notice that the average coverage ratio of MAGIC(all) is greater than 0.998, which means it can label almost all nodes with at least one community. What’s more, the absolute value of remaining three metrics for MAGIC(all) are 0.17 for F1 score, 0.006 for NMI, and 0.005 for NMI-max, which are the highest among all methods. In summary, our model can obtain higher performance of overlapping community detection in both small scale and large scale networks.

6.3 Effects of Community Interactions

We further analyze how community interactions affect quality of detected communities when different combinations of information sources are used. To achieve this, we introduce a new method called CoDA (Communities through Directed Affiliations) [31], an overlapping community detection method that applies to directed networks. Totally, we have six methods to compare and we divide them into three groups. Table 2 shows the composite performance of these methods.

In the first group, we ignore all node attributes (both temporal information and textual contents) as well as edge directions, and compare the results of BIGCLAM and MAGIC(raw). We can see MAGIC(raw) beats BIGCLAM in six out of seven networks. The average performance of MAGIC(raw) is 2.07, which is about 40% higher that of BIGCLAM (1.48). Such improvements occur in another two groups where we use the edge directions and text information, respectively. We contribute these improvements to the introduction of community interactions as the only variable in all three groups is whether such interactions are considered or not.

7. CONCLUSION

In this paper, we study the problem of community detection in temporal text networks. We generate a large set of 32 temporal text networks with reliable ground-truth communities from varying domains and of different scales. They enable us to quantitatively evaluate community detection methods and study the community structure. We study the interactions among ground-truth communities and discover that a large proportion of nodes share a link due to such community interactions. We also find that nodes in the same community have similar textual contents. Based on these empirical observations, we propose MAGIC, a generative model which explicitly models the community interactions and utilizes the information from both link structures and node attributes. Experimental results on networks with ground-truth communities prove the effectiveness of MAGIC. In the future, we plan to model the time information in finer granularity and study some possible regularizations on the community interaction matrix.

8. REFERENCES

Table 2: Results of six methods on 7 small-scale temporal text networks. Network sizes increase from the left to the right. 'T' denotes we consider a certain type of information and 'F' means we do not. We can see the introduction of community interactions clearly improves the performance, when different combinations of information sources are provided.

<table>
<thead>
<tr>
<th>Method</th>
<th>Directed Text</th>
<th>Community Interaction</th>
<th>Theoretical computer science</th>
<th>Computer graphics images</th>
<th>Computational science</th>
<th>Knowledge management</th>
<th>Computer architecture</th>
<th>Real-time computing</th>
<th>Distributed computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIGCLAM</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>*2.334</td>
<td>1.694</td>
<td>1.232</td>
<td>1.519</td>
<td>0.666</td>
<td>2.318</td>
</tr>
<tr>
<td>MAGIC(raw)</td>
<td>F</td>
<td>F</td>
<td>T</td>
<td>*2.214</td>
<td>*2.857</td>
<td>T</td>
<td>2.158</td>
<td>0.000</td>
<td>*2.116</td>
</tr>
<tr>
<td>CoDA</td>
<td>T</td>
<td>F</td>
<td>F</td>
<td>2.020</td>
<td>1.494</td>
<td>1.407</td>
<td>2.120</td>
<td>1.260</td>
<td>2.293</td>
</tr>
<tr>
<td>CESNA</td>
<td>F</td>
<td>F</td>
<td>F</td>
<td>*3.356</td>
<td>2.892</td>
<td>2.494</td>
<td>2.095</td>
<td>1.302</td>
<td>2.756</td>
</tr>
<tr>
<td>MAGIC(all)</td>
<td>F</td>
<td>T</td>
<td>T</td>
<td>3.237</td>
<td>*2.940</td>
<td>*4.000</td>
<td>*4.000</td>
<td>*3.503</td>
<td>*4.000</td>
</tr>
</tbody>
</table>

Table 3: Results of four methods on 9 large scale temporal text networks. Network sizes increase form the left to the right.

<table>
<thead>
<tr>
<th>Metric</th>
<th>Method</th>
<th>Speech recognition</th>
<th>Computer security</th>
<th>Database</th>
<th>Computer network</th>
<th>Algorithm</th>
<th>Operating system</th>
<th>Programming language</th>
<th>Computer vision</th>
<th>Machine learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coverage ratio</td>
<td>BIGCLAM</td>
<td>0.188</td>
<td>0.116</td>
<td>0.103</td>
<td>0.112</td>
<td>0.069</td>
<td>0.058</td>
<td>0.001</td>
<td>0.032</td>
<td>0.039</td>
</tr>
<tr>
<td></td>
<td>MMSB</td>
<td>0.764</td>
<td>0.761</td>
<td>0.760</td>
<td>0.766</td>
<td>0.854</td>
<td>0.858</td>
<td>0.791</td>
<td>0.725</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAGIC(net)</td>
<td>0.768</td>
<td>0.758</td>
<td>0.367</td>
<td>0.691</td>
<td>0.620</td>
<td>0.552</td>
<td>0.476</td>
<td>0.540</td>
<td>0.495</td>
</tr>
<tr>
<td></td>
<td>MAGIC(all)</td>
<td>0.998</td>
<td>0.998</td>
<td>0.997</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
<td>0.999</td>
</tr>
<tr>
<td>F1 score</td>
<td>BIGCLAM</td>
<td>0.059</td>
<td>0.044</td>
<td>0.043</td>
<td>0.036</td>
<td>0.029</td>
<td>0.030</td>
<td>0.000</td>
<td>0.017</td>
<td>0.021</td>
</tr>
<tr>
<td></td>
<td>MMSB</td>
<td>0.125</td>
<td>0.121</td>
<td>0.114</td>
<td>0.115</td>
<td>0.080</td>
<td>0.094</td>
<td>0.041</td>
<td>0.089</td>
<td>0.091</td>
</tr>
<tr>
<td></td>
<td>MAGIC(net)</td>
<td>0.175</td>
<td>0.175</td>
<td>0.079</td>
<td>0.146</td>
<td>0.113</td>
<td>0.114</td>
<td>0.153</td>
<td>0.115</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MAGIC(all)</td>
<td>0.186</td>
<td>0.216</td>
<td>0.163</td>
<td>*0.191</td>
<td>*0.162</td>
<td>*0.166</td>
<td>*0.148</td>
<td>*0.156</td>
<td>*0.143</td>
</tr>
<tr>
<td>NMI</td>
<td>BIGCLAM</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MMSB</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MAGIC(net)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MAGIC(all)</td>
<td>0.008</td>
<td>0.015</td>
<td>0.003</td>
<td>0.008</td>
<td>0.004</td>
<td>0.004</td>
<td>0.004</td>
<td>0.004</td>
<td>0.004</td>
</tr>
<tr>
<td>NMI-max</td>
<td>BIGCLAM</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MMSB</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MAGIC(net)</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td></td>
<td>MAGIC(all)</td>
<td>0.009</td>
<td>0.010</td>
<td>0.004</td>
<td>0.007</td>
<td>0.002</td>
<td>0.003</td>
<td>0.004</td>
<td>0.003</td>
<td>0.006</td>
</tr>
</tbody>
</table>