

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

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'16 October 24-28, 2016, Indianapolis, USA

© 2016 ACM. ISBN 978-1-4503-2138-9.

DOI: 10.1145/1235

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 users 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 online¹ 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,

¹We mask the exact URL to achieve anonymity and will provide it in the final version.

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

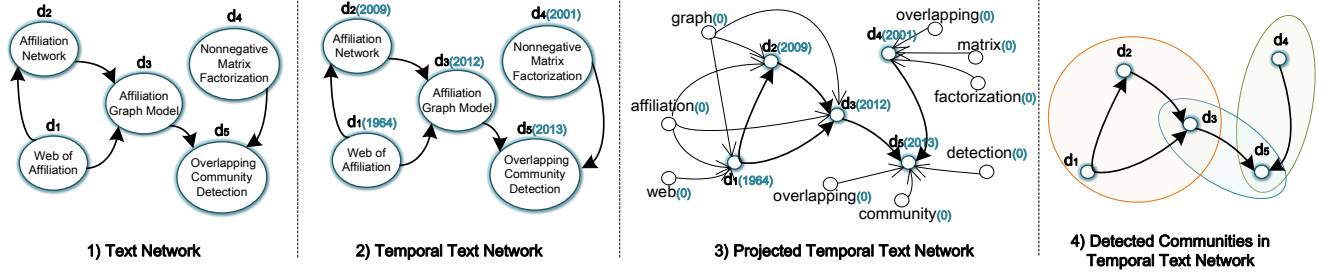


Figure 1: Illustration of three different networks. Blue-colored numbers in parentheses indicate the time information of each node. Clearly, there are two types of information in the temporal text network, i.e., the link structure and the node attribute. The projected temporal text network represents the text information in a more explicit manner and serve as a good proxy for its corresponding temporal text network.

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 zero².

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

²The exact number of this value is actually not important, as long as it is less than the earliest timestamp of all documents.

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 $C = \{C_1, \dots, C_K\}$ such that for each $C_i \in C$, its induced subgraph $G[C_i]$ forms a network community³. 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

³An induced subgraph $G[C_i]$ is a graph whose vertex set is C_i and whose edge set consists of all the edges in E that have both endpoints in C_i .

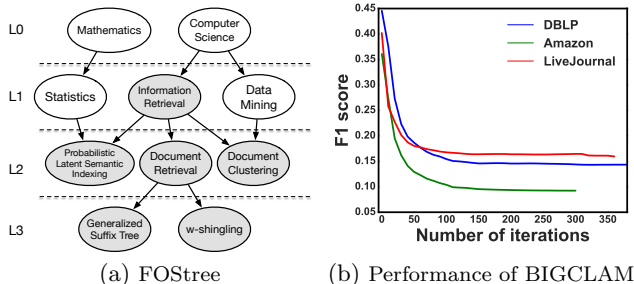


Figure 2: (a) A toy example of Field of Study (FoS) Tree. Each *IR-related* FoS is labeled with a shaded node. (b) Performance of BIGCLAM on three networks with ground-truth communities used in [27].

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 of 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. We compare this value with the average Jaccard similarity of a randomly selected set which has the same size of that community. Results are shown in

Name	N	E	C	A
Computer Hardware	28.8k	45.7k	5.5k	9.2
Knowledge Management	35.9k	104k	4.4k	5.6
Information Retrieval	45.5k	164k	4.3k	5.7
Speech Recognition	245k	863k	12.1k	6.9
Data Mining	312k	858k	14.9k	7.1
Operating System	777k	2.6M	20k	7.4
Artificial Intelligence	1.08M	3.66M	21.9k	6.6
Computer Vision	1.16M	4.31M	19.6k	7.1
Machine Learning	1.51M	7.08M	21.3k	6.2
Bioinformatics	1.84M	16.1M	17.8k	5.8

Table 1: **Dataset Description.** Selected 10 representatives of 32 networks. N: number of nodes, E: number of edges, C: number of communities, A: community memberships per node. M denotes a million and k denotes one thousand. Totally, there are **13 small networks** with size less than 100k nodes, **14 medium networks** with size between 100k nodes and 1M nodes, and **5 large networks** with size larger than 1M nodes.

Figure 3. As 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 another 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 $\frac{1}{|C(u) \cap C(v)|}$. 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 $\frac{1}{2 \times |C(u)|}$ if $c \in C(u)$ or $\frac{1}{2 \times |C(v)|}$ 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_2 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

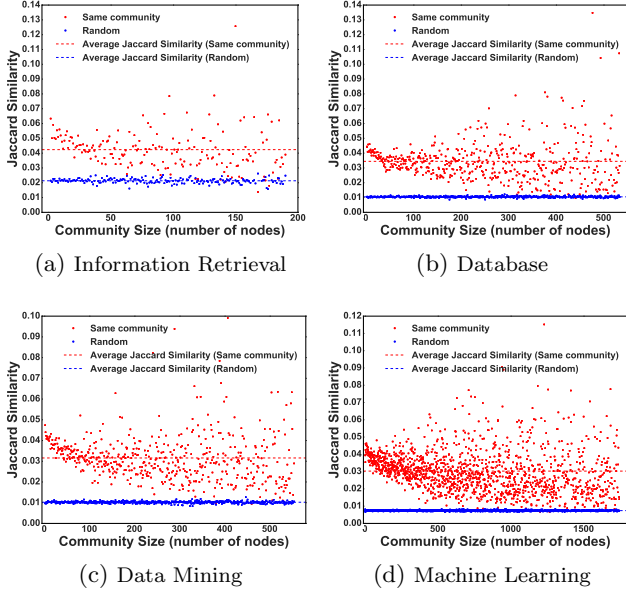


Figure 3: Comparison between the average Jaccard similarity of two randomly selected nodes and that of two nodes from the same community in four temporal text networks.

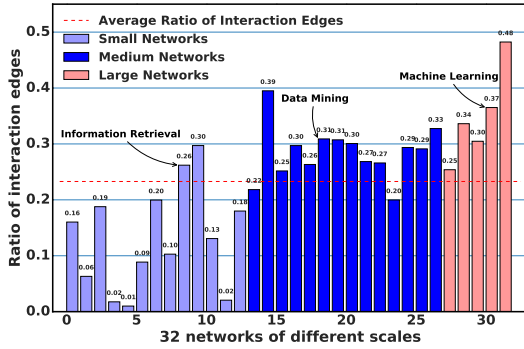


Figure 4: Ratio of interaction edges in 32 temporal text networks of different scales.

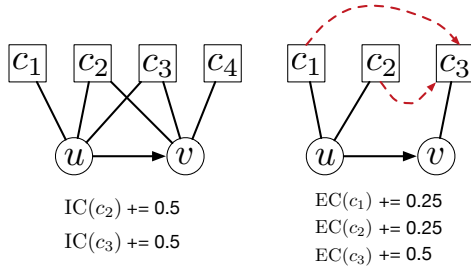


Figure 5: Community affiliation graph. Circles represent nodes in the observed network. Squares represent the latent communities. IC denotes *Internal Connectivity* and EC denotes *External Connectivity*.

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. Con-

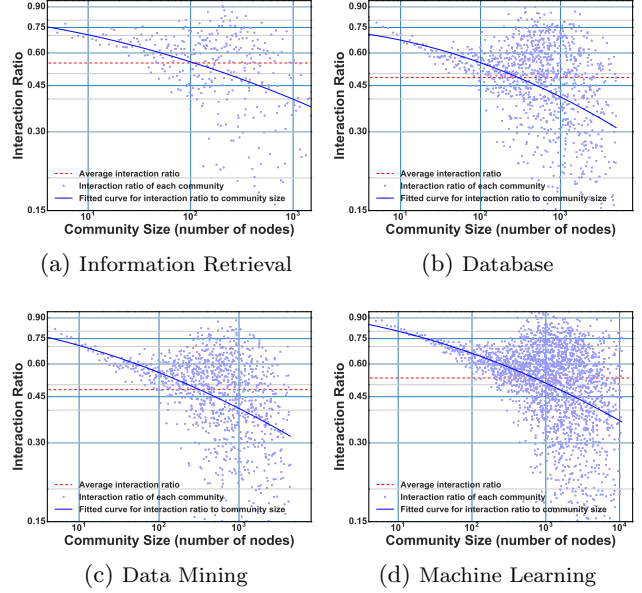


Figure 6: Interaction ratio of each community in four temporal text networks.

sequently, 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 η_{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 η_{ii} . Such influence degrees measure the probability that two nodes in the same commu-

nity are connected and thus they are generally different for each community.

Given the parameters, MAGIC generates a link ($u \rightarrow v$) with the probability $p(u \rightarrow v)$ defined as follows:

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

$$= \left(1 - \exp(-\mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_v) \right) \delta(u \rightarrow v),$$

where \mathbf{F}_u is a column vector representing the community affiliation strength for node u , $\boldsymbol{\eta}$ 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 $\boldsymbol{\eta}$ and δ explicitly models the community interaction and utilities the node temporal information. In this paper, we mainly focus on the natural temporal text network and thus the weighting function δ is defined as:

$$\delta(u \rightarrow v) = \begin{cases} 1 & \text{if } t(u) < t(v) \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

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 use different mechanisms to generate them [30], we adopt a more unified approach. We first construct a projected temporal text network corresponding to the original one 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 \mathbf{F} and the community interaction matrix $\boldsymbol{\eta}$ by maximizing the log likelihood of the observed network G :

$$\hat{\mathbf{F}}, \hat{\boldsymbol{\eta}} = \underset{\mathbf{F} \geq 0, \boldsymbol{\eta} \geq 0}{\operatorname{argmax}} l(\mathbf{F}, \boldsymbol{\eta}), \quad (3)$$

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

$$l(\mathbf{F}, \boldsymbol{\eta}) = \sum_{(u \rightarrow v) \in E} \log(1 - \exp(-\mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_v)) - \sum_{\substack{(u \rightarrow v) \notin E \\ t(u) < t(v)}} \mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_v \quad (4)$$

Notice here we explicitly add the time constraint in the second term so that the absence of an edge (u, 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 provides 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*.

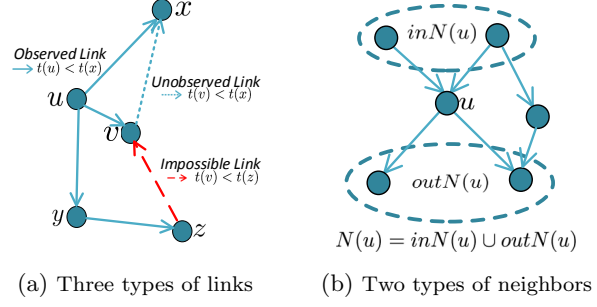


Figure 7: (a) Illustration of three types of links. Due to the time order, a missing link can be either an *unobserved link* or an *impossible link*. (b) Illustration of two types of neighbors.

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 \mathbf{F}_u for each node u with both $\boldsymbol{\eta}$ and \mathbf{F}_v for all other nodes $v \neq u$ fixed. Then, we update the community interaction matrix $\boldsymbol{\eta}$ with the community affiliation matrix \mathbf{F} fixed. We can see each subproblem is a convex optimization problem which makes efficient algorithm possible.

To update the community affiliation strength \mathbf{F}_u for node u , we solve the following subproblem:

$$\hat{\mathbf{F}}_u = \underset{\mathbf{F}_u \geq 0}{\operatorname{argmax}} l(\mathbf{F}_u), \quad (5)$$

where $l(\mathbf{F}_u)$ is the part of $l(\mathbf{F}, \boldsymbol{\eta})$ defined in Eq. 4 that involves \mathbf{F}_u , i.e.,

$$l(\mathbf{F}_u) = \sum_{v \in inN(u)} \log(1 - \exp(-\mathbf{F}_v^T \boldsymbol{\eta} \mathbf{F}_u)) - \sum_{\substack{v \notin N(u) \\ t(v) < t(u)}} \mathbf{F}_v^T \boldsymbol{\eta} \mathbf{F}_u \\ + \sum_{v' \in outN(u)} \log(1 - \exp(-\mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_{v'})) - \sum_{\substack{v' \notin N(u) \\ t(v') > t(u)}} \mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_{v'}, \quad (6)$$

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

$$F_{uk}^{new} \leftarrow \max\{0, F_{uk}^{old} + \alpha_{F_u} (\nabla l(\mathbf{F}_u))_k\}, \quad (7)$$

where α_{F_u} is the step size computed by backtracking line search [4], and the gradient is:

$$\nabla l(\mathbf{F}_u) = \sum_{v \in inN(u)} \frac{\exp(-\mathbf{F}_v^T \boldsymbol{\eta} \mathbf{F}_u)}{1 - \exp(-\mathbf{F}_v^T \boldsymbol{\eta} \mathbf{F}_u)} \boldsymbol{\eta}^T \mathbf{F}_v - \sum_{\substack{v \notin N(u) \\ t(v) < t(u)}} \boldsymbol{\eta}^T \mathbf{F}_v \\ + \sum_{v' \in outN(u)} \frac{\exp(-\mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_{v'})}{1 - \exp(-\mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_{v'})} \boldsymbol{\eta} \mathbf{F}_{v'} - \sum_{\substack{v' \notin N(u) \\ t(v') > t(u)}} \boldsymbol{\eta} \mathbf{F}_{v'}. \quad (8)$$

After the community affiliation matrix \mathbf{F} updated, we fix \mathbf{F} and update the community interaction matrix $\boldsymbol{\eta}$. Notice

Algorithm 1 Parameter Learning for MAGIC

- 1: **Input:** $G = (V, E; T)$: the original temporal text network, max_iter : maximum number of iterations.
 - 2: **Output:** \mathbf{F} , the community affiliation matrix, and $\boldsymbol{\eta}$, the community interaction matrix.
 - 3: Determine the number of communities K .
 - 4: Initialize \mathbf{F} and $\boldsymbol{\eta}$.
 - 5: **repeat**
 - 6: **for** $u = 1, 2, \dots, N$ **do**
 - 7: Calculate $\nabla l(\mathbf{F}_u)$ based on Eq. (8)
 - 8: Calculate α_{F_u} using backtracking line search
 - 9: Update \mathbf{F}_u based on Eq. (7)
 - 10: **end for**
 - 11: Calculate $\nabla_{\boldsymbol{\eta}} l(\mathbf{F}, \boldsymbol{\eta})$ based on Eq. (10)
 - 12: Calculate $\alpha_{\boldsymbol{\eta}}$ using backtracking line search
 - 13: Update $\boldsymbol{\eta}$ based on Eq. (9)
 - 14: **until** convergence or max_iter is reached
 - 15: Return parameters \mathbf{F} and $\boldsymbol{\eta}$
-

that $\boldsymbol{\eta}$ is involved in every term of Eq. 4 and thus we solve it directly.

$$\eta_{ij}^{new} \leftarrow \max\{0, \eta_{ij}^{old} + \alpha_{\boldsymbol{\eta}} (\nabla_{\boldsymbol{\eta}} l(\mathbf{F}, \boldsymbol{\eta}))_{ij}\}, \quad (9)$$

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

$$\nabla_{\boldsymbol{\eta}} l(\mathbf{F}, \boldsymbol{\eta}) = \sum_{(u \rightarrow v) \in E} \frac{\exp(-\mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_v)}{1 - \exp(-\mathbf{F}_u^T \boldsymbol{\eta} \mathbf{F}_v)} \mathbf{F}_u \mathbf{F}_v^T - \sum_{\substack{(u \rightarrow v) \notin E \\ t(u) < t(v)}} \mathbf{F}_u \mathbf{F}_v^T. \quad (10)$$

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

$$\sum_{\substack{v \notin N(u) \\ t(v) < t(u)}} \boldsymbol{\eta}^T \mathbf{F}_v = \sum_{t(v) < t(u)} \boldsymbol{\eta}^T \mathbf{F}_v - \sum_{v \in inN(u)} \boldsymbol{\eta}^T \mathbf{F}_v, \quad (11)$$

$$\sum_{\substack{v' \notin N(u) \\ t(v') > t(u)}} \boldsymbol{\eta} \mathbf{F}_{v'} = \sum_{t(v') > t(u)} \boldsymbol{\eta} \mathbf{F}_{v'} - \sum_{v' \in outN(u)} \boldsymbol{\eta} \mathbf{F}_{v'}, \quad (12)$$

$$\sum_{\substack{(u \rightarrow v) \notin E \\ t(u) < t(v)}} \mathbf{F}_u \mathbf{F}_v^T = \sum_{\substack{(u \rightarrow v) \\ t(u) < t(v)}} \mathbf{F}_u \mathbf{F}_v^T - \sum_{\substack{(u \rightarrow v) \in E \\ t(u) < t(v)}} \mathbf{F}_u \mathbf{F}_v^T. \quad (13)$$

In this way, we can compute $\nabla l(\mathbf{F}_u)$ in $O(N(u))$ by caching the first term in the right hand side of Eqs. (11) and (12), and compute $\nabla_{\boldsymbol{\eta}} l(\mathbf{F}, \boldsymbol{\eta})$ 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 \mathbf{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 \mathbf{F} , we extend the method in [8] to directed network. The conductance in directed network is also defined in [7]. The in-neighbors $inN(u)$ of node u is *locally minimal* if $inN(u)$ has lower conductance than all in-neighbors $inN(v)$ where node $v \in outN(u)$. 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 $\boldsymbol{\eta}$, 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 $\hat{\mathbf{F}}$ and $\hat{\boldsymbol{\eta}}$, we need to determine the ‘‘hard’’ community membership of each node. We achieve this by thresholding the learned $\hat{\mathbf{F}}$ 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 δ_k as below:

$$\delta_k = \sqrt{-\frac{\log(1 - 1/N)}{\eta_{kk}}}. \quad (14)$$

With $\{\delta_k\}$ obtained, we consider node u belonging to community k if $F_{uk} \geq \delta_k$.

5.3.3 Choosing the number of communities

We use the method in [2] to choose the number of communities K . Specifically, we reserve 20% of links for validation and learn the model parameters with the remaining 80% of links for different K . After that, we use the learned parameters to predict the links in validation set and select the K with the maximum prediction score as the number of communities. The whole process of parameter learning is described in Algorithm 1.

6. EXPERIMENTS

In this section, we proceed to evaluate the effectiveness of proposed MAGIC method for overlapping community detection. The experiments are set up as the following.

6.1 Experiment Setup

6.1.1 Dataset

We evaluate our model using two categories of networks with ground-truth communities. The first category is the temporal text networks defined in section 4, and the second one is the networks without textual and temporal information. For the first category, we randomly select 7 small networks with less than 100,000 nodes and 9 large networks which are of size from hundreds of thousands of nodes to millions of nodes. We perform text normalization by removing stop words and stemming on top of the original data. For the second category, we just delete the node attributes in networks. In all networks, each node is assigned to at least one community and thus all quantitative metrics are applicable.

6.1.2 Compared methods

We compare MAGIC with another 4 baseline methods. Two of them are representatives of methods based on dense subgraph extraction and another two are representatives of methods based on affiliation graph model.

CPM (Cliques Percolation Method) CPM [17] constructs the communities from k -cliques which is a complete subgraph of k nodes. Two k -cliques are defined as adjacent if they share $k - 1$ nodes. A community is further defined as a union of all adjacent k -cliques. For CPM, we set the clique size $k = 4$ and use the implementation in the Stanford Network Analysis Platform (SNAP)⁴.

⁴<https://github.com/snap-stanford/snap>

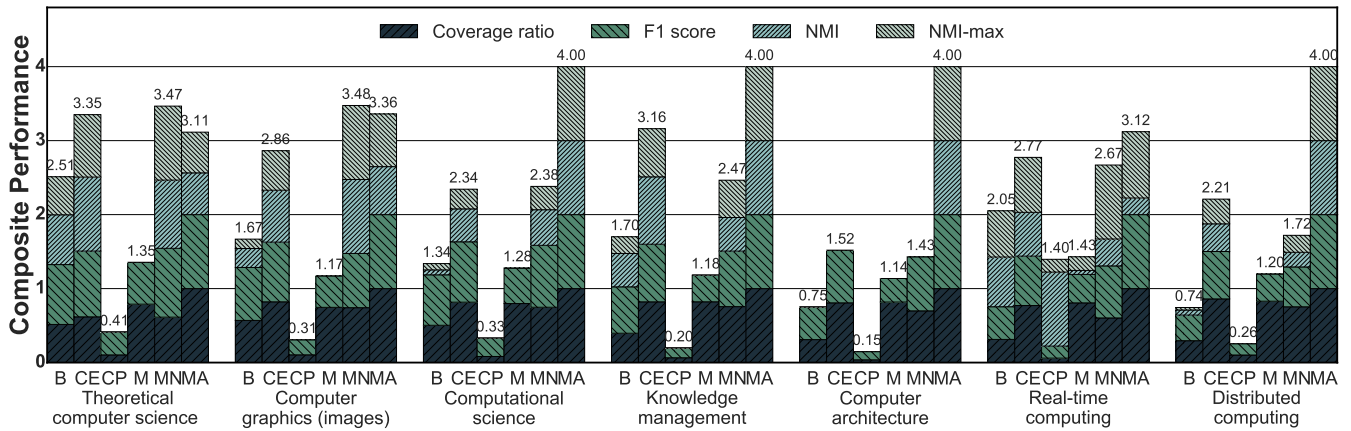


Figure 8: The composite performance of 6 methods on 7 small scale temporal text networks. B: BIGCLAM; CE: CESNA, CP: CPM, M: MMSB, MN: MAGIC(net), MA: MAGIC(all).

MMSB (Mixed-Membership Stochastic Block Model): MMSB [2] is originally proposed as a variational inference algorithm for fast approximating posterior inference and later has 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]⁵.

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 \hat{C} . 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. Please refer to [28] for details.

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-

⁵MMSB: <https://github.com/premgopalan/svinet>

texes which cannot form a clique with size 4. We tried to decrease 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 than 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

- [1] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.
- [2] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. In *Advances in Neural Information Processing Systems*, pages 33–40, 2009.
- [3] R. Balasubramanyan and W. W. Cohen. Block-lda: Jointly modeling entity-annotated text and entity-entity links. In *SDM*, volume 11, pages 450–461. SIAM, 2011.
- [4] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.
- [5] R. L. Breiger. The duality of persons and groups. *Social forces*, 53(2):181–190, 1974.
- [6] S. Fortunato and M. Barthelemy. Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41, 2007.
- [7] S. Galhotra, A. Bagchi, S. Bedathur, M. Ramanath, and V. Jain. Tracking the conductance of rapidly evolving topic-subgraphs. *Proc. VLDB Endow.*, 8(13):2170–2181, Sept. 2015.
- [8] D. F. Gleich and C. Seshadhri. Vertex neighborhoods, low conductance cuts, and good seeds for local community methods. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 597–605. ACM, 2012.
- [9] P. K. Gopalan and D. M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 110(36):14534–14539, 2013.
- [10] N. J. Krogan, G. Cagney, H. Yu, G. Zhong, X. Guo, A. Ignatchenko, J. Li, S. Pu, N. Datta, A. P. Tikuisis, et al. Global landscape of protein complexes in the yeast *saccharomyces cerevisiae*. *Nature*, 440(7084):637–643, 2006.
- [11] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015, 2009.
- [12] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.
- [13] L. Liu, L. Xu, Z. Wangy, and E. Chen. Community detection based on structure and content: A content propagation perspective. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 271–280, Nov 2015.
- [14] A. F. McDaid, D. Greene, and N. Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*, 2011.
- [15] M. E. Newman. Detecting community structure in networks. *The European Physical Journal B-Condensed Matter and Complex Systems*, 38(2):321–330, 2004.
- [16] M. E. Newman. Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582, 2006.
- [17] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
- [18] I. Psorakis, S. Roberts, M. Ebden, and B. Sheldon. Overlapping community detection using bayesian non-negative matrix factorization. *Physical Review E*,

Method	Directed	Text	Community Interaction	Theoretical computer science	Computer graphics images	Computational science	Knowledge management	Computer architecture	Real-time computing	Distributed computing
BIGCLAM	F	F	F	2.498	*1.694	1.232	1.519	0.656	2.138	0.857
MAGIC(raw)	F	F	T	*3.343	1.605	*1.895	*2.334	*1.149	*2.780	*1.627
CoDA	T	F	F	2.020	1.494	1.407	2.120	1.260	2.294	1.531
MAGIC(net)	T	F	T	*3.451	*3.501	*2.111	*2.214	*1.272	*2.462	*2.116
CESNA	F	T	F	*3.336	2.892	2.094	2.679	1.362	2.758	2.857
MAGIC(all) ⁶	F	T	T	3.237	*2.940	*4.000	*4.000	*4.000	*3.503	*4.000

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.

Metric	Method	Speech recognition	Computer security	Database	Computer network	Algorithm	Operating system	Programming language	Computer vision	Machine learning
Coverage ratio	BIGCLAM	0.188	0.116	0.103	0.112	0.069	0.058	0.001	0.032	0.059
	MMSB	0.764	0.761	0.760	0.756	0.766	0.854	0.858	0.751	0.775
	MAGIC(net)	0.788	0.758	0.367	0.691	0.620	0.552	0.476	0.540	0.495
	MAGIC(all)	*0.998	*0.998	*0.997	*0.999	*0.999	*0.997	*0.999	*0.999	*0.999
F1 score	BIGCLAM	0.059	0.044	0.043	0.036	0.029	0.030	0.000	0.017	0.021
	MMSB	0.125	0.121	0.107	0.115	0.080	0.054	0.041	0.089	0.097
	MAGIC(net)	0.175	0.169	0.079	0.146	0.137	0.147	0.118	0.118	0.116
	MAGIC(all)	*0.186	*0.216	*0.163	*0.191	*0.162	*0.166	*0.148	*0.156	*0.143
NMI	BIGCLAM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	MMSB	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	MAGIC(net)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	MAGIC(all)	*0.008	*0.015	*0.003	*0.008	*0.004	*0.004	*0.003	*0.004	*0.004
NMI-max	BIGCLAM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	MMSB	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	MAGIC(net)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	MAGIC(all)	*0.009	*0.010	*0.004	*0.007	*0.002	*0.003	*0.004	*0.003	*0.006

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

- 83(6):066114, 2011.
- [19] Y. Ruan, D. Fuhry, and S. Parthasarathy. Efficient community detection in large networks using content and links. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1089–1098. International World Wide Web Conferences Steering Committee, 2013.
- [20] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.
- [21] J. Shen, Z. Song, S. Li, Z. Tan, Y. Mao, L. Fu, L. Song, and X. Wang. Modeling topic-level academic influence in scientific literatures. In *Workshops at the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [22] A. Sinha, Z. Shen, Y. Song, H. Ma, D. Eide, B.-J. P. Hsu, and K. Wang. An overview of microsoft academic service (mas) and applications. In *Proceedings of the 24th International Conference on World Wide Web, WWW ’15 Companion*, pages 243–246, New York, NY, USA, 2015. ACM.
- [23] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174. ACM, 2015.
- [24] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding. Community discovery using nonnegative matrix factorization. *Data Mining and Knowledge Discovery*, 22(3):493–521, 2011.
- [25] J. Xie, S. Kelley, and B. K. Szymanski. Overlapping community detection in networks: The state-of-the-art and comparative study. *Acm computing surveys (csur)*, 45(4):43, 2013.
- [26] J. Yang and J. Leskovec. Community-affiliation graph model for overlapping network community detection. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 1170–1175. IEEE, 2012.
- [27] J. Yang and J. Leskovec. Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics, MDS ’12*, pages 3:1–3:8, New York, NY, USA, 2012. ACM.
- [28] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 587–596. ACM, 2013.
- [29] J. Yang and J. Leskovec. Structure and overlaps of ground-truth communities in networks. *ACM Trans. Intell. Syst. Technol.*, 5(2):26:1–26:35, Apr. 2014.
- [30] J. Yang, J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *Data mining (ICDM), 2013 IEEE 13th international conference on*, pages 1151–1156. IEEE, 2013.
- [31] J. Yang, J. McAuley, and J. Leskovec. Detecting cohesive and 2-mode communities in directed and undirected networks. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 323–332. ACM, 2014.
- [32] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 927–936. ACM, 2009.
- [33] H. Zhang, I. King, and M. R. Lyu. Incorporating implicit link preference into overlapping community detection. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI’15*, pages 396–402. AAAI Press, 2015.
- [34] Y. Zhang and D.-Y. Yeung. Overlapping community detection via bounded nonnegative matrix tri-factorization. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 606–614. ACM, 2012.