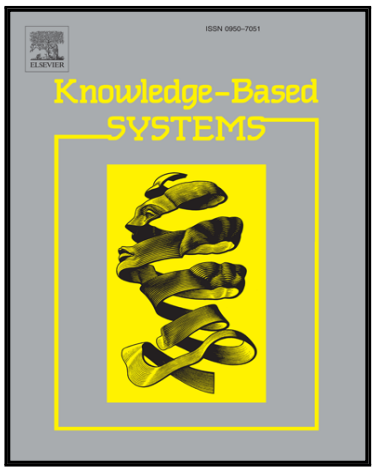


Adaptive Community Detection Incorporating Topology and Content in Social Networks

Meng Qin , Di Jin , Kai Lei , Bogdan Gabrys , Katarzyna Musial

PII: S0950-7051(18)30388-5
DOI: https://doi.org/10.1016/j.knosys.2018.07.037
Reference: KNOSYS 4437

To appear in: *Knowledge-Based Systems*

Received date: 18 December 2017
Revised date: 12 June 2018
Accepted date: 27 July 2018

Please cite this article as: Meng Qin , Di Jin , Kai Lei , Bogdan Gabrys , Katarzyna Musial , Adaptive Community Detection Incorporating Topology and Content in Social Networks, *Knowledge-Based Systems* (2018), doi: https://doi.org/10.1016/j.knosys.2018.07.037

Adaptive Community Detection Incorporating Topology and Content in Social Networks

Meng Qin[1], Di Jin[2,*], Kai Lei[1,*], Bogdan Gabrys[3], Katarzyna Musial[3]

[1] Shenzhen Key Lab for Information Centric Networking and Blockchain Technology,
School of Electronic and Computer Engineering,
Peking University, Shenzhen, China
[2] School of Computer Science and Technology,
Tianjin University, Tianjin, China
[3] Advanced Analytics Institute, School of Software
University of Technology Sydney, Australia
mengqin_az@foxmail.com, jindi@tju.edu.cn, leik@pkusz.edu.cn, {bogdan.gabrys,
katarzyna.musial-gabrys}@uts.edu.au

Abstract. In social network analysis, community detection is a basic step to understand the structure and function of networks. Some conventional community detection methods may have limited performance because they merely focus on the networks' topological structure. Besides topology, content information is another significant aspect of social networks. Although some state-of-the-art methods started to combine these two aspects of information for the sake of the improvement of community partitioning, they often assume that topology and content carry similar information. In fact, for some examples of social networks, the hidden characteristics of content may unexpectedly mismatch with topology. To better cope with such situations, we introduce a novel community detection method under the framework of non-negative matrix factorization (NMF). Our proposed method integrates topology as well as content of networks and has an adaptive parameter (with two variations) to effectively control the contribution of content with respect to the identified mismatch degree. Based on the disjoint community partition result, we also introduce an additional overlapping community discovery algorithm, so that our new method can meet the application requirements of both disjoint and overlapping community detection. The case study using real social networks shows that our new method can simultaneously obtain the community structures and their corresponding semantic description, which is helpful to understand the semantics of communities. Related performance evaluations on both artificial and real networks further indicate that our method outperforms some state-of-the-art methods while exhibiting more robust behavior when the mismatch between topology and content is observed.

Keywords: Social network analysis; Community detection; Semantic description; Non-negative matrix Factorization; Robustness

This paper is an abridged version of [36].

* Corresponding author.

1 Introduction

As a common example of complex systems, social networks can be represented as graphs with sets of nodes and edges, in which community is a significant substructure. Generally, typical communities can be described as subsets of nodes within which the connections are dense, but between which are sparse [1]. The identification of communities can help finding groups of users with similar interests, backgrounds or purposes, which can effectively support the advanced applications of social networks such as recommender systems, sentiment analysis and user profiling. As a result, community detection is a basic and essential step in social network analysis [2].

Since community is originally defined based on linkage structure, conventional community detection methods tend to purely focus on the network's topology [1,16,17,18,19]. However, these methods may have limited performance, because in social networks there is a complicated relationship between topology and content (which is another significant source of information). For example, users may choose different communities for different interests or reasons, which is more likely to be reflected in the content posted by users rather than the linkage structure. Moreover, two users in the social network may also belong to the same community even though they have not had any direct contact [5] (but they may have similar interests which can be encoded in content).

In order to compensate for the possible deficiencies of only using the topological information, one can supplement it by incorporating network's attribute. In practice, content can be effectively used to improve community partition results since it provides orthogonal information to network topology [4]. Moreover, the content information can also help semantically annotate the communities during the process of community detection [5,32]. For the above reasons, some state-of-the-art community detection methods have started to integrate these two aspects of the social networks' related information [3,23,24,25,26].

Although the content in social networks may contain additional characteristics that cannot be reflected in networks' topology, most of the existing community detection methods which incorporate both aspects of the information are based on the assumption that the topology is compatible with the content. In other words, they assume that both structure and content carry the same information from the perspective of network analysis. In fact, there is no evidence that such two sources of information must share the same characteristics in any case, and the *mismatch* between them is common in real social networks. For instance, according to [3], in Twitter the real social relationship reflects the community structure more directly than the content, as the messages generated by users are very diverse. Moreover, our pre-experiment presented in Section 4.3 also indicates the *mismatch* between topology and content. When these two types of information do not *match* well, methods attempting to integrate the topology and content may lack the required robustness and often lead to poor results.

In spite of the *mismatch* between topology and content, our pre-experiment also shows that if we effectively control the contribution of content, the community structures can still be accurately extracted. Based on such conclusion, we propose a novel Adaptive Semantic Community Detection (ASCD) method, which integrates topological structure and node attributes of social networks under the framework of

non-negative matrix factorization (NMF) [6]. Different from some other state-of-the-art methods integrating these two sources of information, our method obtains better robustness, as we introduce a novel adaptive parameter to control the trade-off between topology and content according to the identified *mismatch* degree. Especially for the adaptive parameter, we have designed two variations, which are respectively based on (i) a monotone decreasing function with *arctan* as the core part and (ii) the normalized mutual information (NMI) [7] between topology and content. By utilizing such two different variations of the adaptive parameter, we have finally deduced two derivative forms of our ASCD model named as ASCD-ARC and ASCD-NMI. Moreover, we also derive an effective solving strategy for our method to facilitate the extraction of communities as well as their semantic interpretation at the same time.

Additionally, as the overlapping community structures, where each node in the network can simultaneously belongs to multiple communities, are more ubiquitous in reality [8], we introduce an extended overlapping community detection algorithm based on our method's original disjoint community partition result (where each node would only belong to one community). Different from some other threshold-based overlapping community discovery methods [14], our extended algorithm can directly extract the overlapping community structures from the disjoint community partition result derived by our method without adjusting any additional thresholds.

More importantly, the experiment on a set of artificial networks reveals that our ASCD method exhibits a more robust behavior in presence of *mismatch* between topology and content. The case study on a real social network [9] demonstrates that the proposed method is also able to simultaneously obtain the corresponding semantic descriptions when communities are partitioned. Finally, the evaluation on real network datasets [10,11,12,13] for both disjoint and overlapping community detection also show that our method outperforms some state-of-the-art approaches.

The rest of this paper is organized as follow. We first discuss related work in Section 2 and give formal definitions about community detection as well as the *mismatch* effect in Section 3. Then, in Section 4 we present our ASCD method in details. Specifically, in this section we first model the network's topology and content respectively. By using a simplified unified model, we introduce a pre-experiment to illustrate the *mismatch* effect in real social networks. Finally, we propose our ASCD method with two different variations (ASCD-ARC and ASCD-NMI) and then derive the algorithms for disjoint and overlapping community detection. Thereafter, four evaluation experiments are described in Section 5, including: (i) a parametric analysis about the appropriate setting of the model's hyper-parameters; (ii) an evaluation on artificial networks about the *attribute refining* (AR) and the *mismatch* effect; (iii) a case study concerned with semantic description; and (iv) a performance evaluation on real social networks for both disjoint and overlapping community detection.

2 Related Work

For the last few decades, several methods have been proposed for the task of discovering the community structure in networks. A comprehensive review of different community detection methods can be found in [2], [14] and [35]. Especially,

[15] gives an overview about some state-of-the-art methods that incorporate both topology and content to extract communities in attributed networks.

Based on the definition of community in complex networks introduced in [1], conventional methods mainly explore the topological structure of the network to achieve the community partition. For example, on the basis of edge betweenness and modularity optimization respectively, methods proposed in [1] and [16] both consider community detection as a graph cut problem. Also, for the strategy of modularity maximization, authors of [17] introduce a fast hierarchical clustering method. In [18], the optimization of modularity is transformed into a relaxed spectral problem, and a novel spectral clustering method is then proposed to extract community structures in the network. Moreover, for some other model-based methods, authors of [19] propose the well-known stochastic block model and it is utilized in [20] to infer the community partition of uncertain networks, while in [21] and [22], the identification of community is modeled under the framework of non-negative matrix factorization (NMF) [6]. However, such conventional methods may have limited room for the improvement of community partition as they only focus on the structural information of the network while entirely neglecting the content information. In fact, content can provide additional information that cannot be encoded in topology.

Therefore, some of the recently developed state-of-the-art methods use both topology and content to detect communities. For instance, the method proposed in [23] partitions community structure by defining a uniform signal strength which fuses the link strength with the content similarity between each pair of nodes. In [24], the authors introduce an effective algorithm with two main phases, including a hill-climbing phase to explore the structural information of the network and a description induction phase to adjust the community partition in a supervised way. Research presented in [3] tries to combine topology with content by using graph regularization method. With regard to some probabilistic methods, authors of [25] propose a joint generative model to combine the links and content in networks while authors of [26] considering the integration of topology and content from the perspective of the discriminant model. Moreover, a general framework for graph clustering (namely the community detection problem) in attributed networks is proposed in [27] from the perspective of Bayesian model. By incorporating topological and content information, such state-of-the-art methods only aim at improving the community partition. In order to understand the semantic of certain communities, additional steps still need to be taken to infer relevant attributes for each community. But method introduced in [5], which combines the network's topology and content based on NMF, can simultaneously obtain the community partition and the corresponding semantic description for each community.

However, most methods that incorporate topology and content are based on the *match* assumption. In other word, they tend to simply assume that the topology and the content share the similar information, but such assumption is not sufficient. As stated in [4], the *mismatch* between these two types of information is common in real social networks. Especially, when the *mismatch* effect occurs, methods with the *match* assumption may result in poor community partition, and this is the problem which forms the main focus of this paper.

3 Problem Definition

Definition of Community Detection Incorporating Topology and Content.
We use a 4-tuple $G=(V,E,W,A)$ to represent a network with node attributes, where $V=\{v_1,\cdots,v_N\}$ is the set of nodes, $E=\{(v_i,v_j)|v_i,v_j\in V,i\neq j\}$ is the set of edges, $W=\{w_1,\cdots,w_M\}$ is the set of attributes, and $A=\{a(v_i),\cdots,a(v_N)\}$ represent the attributes of each node. Especially, $a(v_i)\in W$ is the set of node *i*'s attributes.

Given the above network, a typical community detection process is to partition the node set V into K subsets (communities) $C=\{C_1,\cdots,C_K\}$ according to the linkage structure E and the node attributes A such that: (i) within each subset the linkage is dense and the content is similar; but (ii) between any subsets the linkage is relatively loose and the content is distinct. For any $r\neq k$ $(1\leq r,k\leq K)$, if $C_r\bigcap C_k=\varnothing$, then we called the above process as *disjoint community detection*. If there exist $r\neq k$ $(1\leq r,k\leq K)$ that satisfy $C_r\bigcap C_k\neq\varnothing$, we called the process as *overlapping community detection*.

In this paper, we utilize a label sequence $L=\{l_1,\cdots,l_N\}$ to represent the partition result of disjoint community detection, where l_i is the community label of node *i*, (each node in the network can only belong to single community). While we use a set $L=\{L_1,\cdots,L_K\}$ to describe the overlapping partition result, where L_r is the node set of community *r*'s node members.

Since the purpose of community detection is to discover substructures that correspond to distinct groups or organizations in real social networks, in this paper, we evaluate the community partition given by a certain method by comparing the correspondence between the partition result and the ground-truth (provided by the testing dataset). Usually, better correspondence means better performance.

Definition of Match (Mismatch) between Topology and Content.
If we partition the node set V into K subsets (communities) respectively according to the linkage and the node content, we can obtain two clustering structures with $T=\{T_1,\cdots,T_K\}$ corresponding to topology and $S=\{S_1,\cdots,S_K\}$ corresponding to content, where nodes in the same cluster *r* (T_r or S_r) tend to have similar property (dense linkages or similar content). Therefore, an appropriate definition of the *match* degree (*mismatch* degree) between topology and content can be described as the correspondence between the two clustering structures represented by T and S, in which better correspondence represents higher *match* degree (lower *mismatch* degree).

4 The Model

Generally, we consider the case of an undirected and unweighted network G with N nodes. Taking the content information into account, we assume the nodes attributes and communities' semantic description are represented by the Bag-of-Words model with M keywords in total. We use an *adjacency matrix* $\mathbf{A} \in \Re^{N \times N}$ to represent the network connectivity, where $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 1$ when there is an edge between node i and j, and otherwise $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 0$. To represent content of individual nodes, we introduce a *node attribute matrix* $\mathbf{C} \in \Re^{N \times M}$, where $\mathbf{C}_{is} = 1$ when node i's attribute has keyword s, and otherwise $\mathbf{C}_{is} = 0$. As in our method, the number of communities needs to be set in advance, so we generally assume there are K communities in the network.

4.1 Modeling Topological Structure

For an appropriate representation of the community partitions, we introduce a *community membership matrix* $\mathbf{X} \in \Re^{N \times K}$, where $\mathbf{X}_{ir}$ is defined as the propensity that node i belongs to community r. Since the number of edges between any pair of nodes is either 0 or 1, the expectation of the number of edges between nodes i and j in community r is $\mathbf{X}_{ir}\mathbf{X}_{jr}$. Accordingly, $\sum_{r=1}^{K} \mathbf{X}_{ir}\mathbf{X}_{jr}$ represents the expected number of edges between such pair of nodes in the network. In fact, the expectation should be as close as possible to the real value of $\mathbf{A}_{ij}$, so we have the following objective function related to the topological structure:

$$\arg\min_{\mathbf{X}} \left\| \mathbf{A} - \mathbf{X}\mathbf{X}^T \right\|_F^2 \quad \text{s.t.} \quad \mathbf{X}_{ir} \geq 0 . \tag{1}$$

Note that when the optimal solution of the above NMF problem is obtained, one can directly use $\mathbf{X}$ to extract the corresponding disjoint community partition by assigning the column index with maximum propensity value in the i-th ($1 \leq i \leq N$) row of $\mathbf{X}$ to be node i's community label.

4.2 Modeling Content Information

In order to give the corresponding semantic description of each community, we define a *community attribute matrix* $\mathbf{Y} \in \Re^{M \times K}$, where $\mathbf{Y}_{sr}$ represents the propensity that the community r can be described by the keyword s. Similarly, from the perspective of a generative model, the expectation that the node i belongs to the community r and can be described by the keyword s is $\mathbf{X}_{ir}\mathbf{Y}_{sr}$, so $\sum_{r=1}^{K} \mathbf{X}_{ir}\mathbf{Y}_{sr}$ represents the expectation that node i's content has the keyword s in the network. Since the real value of $\mathbf{C}_{is}$ should also be as close as possible to such expectation value, we then have the following objective function related to the content information:

$$\arg\min_{\mathbf{X},\mathbf{Y}} \left\| \mathbf{C} - \mathbf{X}\mathbf{Y}^T \right\|_F^2 \quad \text{s.t.} \quad \mathbf{X}_{ir} \geq 0, \quad \mathbf{Y}_{sr} \geq 0. \tag{2}$$

On the other hand, the same as the modeling process in [5], if the semantic description of the node i is highly similar to that of the community r, then the node i should have high propensity of belonging to the community r (which is based on the *match* assumption). Namely, the value of $\sum_{s=1}^{M} \mathbf{C}_{is}\mathbf{Y}_{sr}$ should be as close as possible to $\mathbf{X}_{ir}$. Therefore, we can formulate another objective function as:

$$\arg\min_{\mathbf{X},\mathbf{Y}} \left\| \mathbf{X} - \mathbf{C}\mathbf{Y} \right\|_F^2 \quad \text{s.t.} \quad \mathbf{X}_{ir} \geq 0, \quad \mathbf{Y}_{sr} \geq 0. \tag{3}$$

Although the premise of (3) that topology *match* with content may not be valid, and it is also not strictly an NMF problem, since the *base matrix* **C** in (3) is the known quantity when a network is given (the *base matrix* and the *coefficient matrix* are defined in [6]), we still use it as a candidate for further discussion.

According to our definition in Section 3, for the semantic description, the keywords used to describe the same community should be semantically similar, but the description between different communities should be distinguishable. To satisfy such property, we introduce another sparsity penalty for **Y**, based on the work of Sparse Non-Negative Matrix Factorization (SNMF) [28].

However, the objective function in (2) is not suitable to introduce such sparsity item for **Y**. As **X** and **Y** are both the unknown quantities that need to be determined, the sparsity of the *coefficient matrix* **Y** is closely related to the value of the *base matrix* **X**. In this case, besides **Y**, additional effort should be taken to ensure the sparsity of **X**, which may result in unnecessary complexity to the model.

But for (3), the *base matrix* **C** is the known quantity and is sparse as it's used to describe individual node attributes. Under such circumstance, to satisfy the sparsity constraint of the *community attribute matrix* **Y**, we only need to add a 1-norm sparsity penalty item for **Y** with the following formula:

$$\arg\min_{\mathbf{X},\mathbf{Y}} \left\| \mathbf{X} - \mathbf{C}\mathbf{Y} \right\|_F^2 + \lambda \left\| \mathbf{Y}(:,r) \right\|_1^2 \quad \text{s.t.} \quad \mathbf{X}_{ir} \geq 0, \quad \mathbf{Y}_{sr} \geq 0, \tag{4}$$

where λ is a non-negative parameter to adjust the contribution of the sparsity item. For the above objective function, when the optimal solution is obtained, besides extracting the community partition result from **X**, the semantic description of community r can also be generated by extracting the top l words (with top l propensity values) in the r-th column of **Y**.

4.3 The Pre-experiment for Mismatch Effect (Based on A Simplified Model)

The Simplified Unified Model.

Based on the models derived above, in this section, we introduce a simplified unified model that integrates topology as well as content to conduct a pre-experiment about the *mismatch* effect, and the experiment results will form the basic concept of our formal unified model, which will be introduced in Section 4.4.

To construct the simplified unified model, we combine (1) with (3) to achieve the following optimization problem with its objective function:

$$\arg\min_{\mathbf{X},\mathbf{Y}} \alpha \left\| \mathbf{A} - \mathbf{X}\mathbf{X}^T \right\|_F^2 + \left\| \mathbf{X} - \mathbf{C}\mathbf{Y} \right\|_F^2 \quad \text{s.t.} \quad \mathbf{X}_{ir} \geq 0, \quad \mathbf{Y}_{sr} \geq 0. \tag{5}$$

where α is the parameter to adjust the trade-off between the first and second term. Especially, in the first term, $\mathbf{X}$ corresponds to the topological structure. In the second term, $\mathbf{Y}$ corresponds to the content information while $\mathbf{X}$ plays the role of bridging such two aspects. Our goal is to find the optimal $\mathbf{X}$ and $\mathbf{Y}$ that minimize the objective function (5).

The general solution strategy of an NMF problem is to properly initialize the unknown quantities and then use certain rules of iteration to continuously update their values until convergence. Moreover, for most NMF unified models, before formally updating, additional initialization steps usually need to be taken to speed up convergence and help avoid getting the local minimum solution. Hence, to obtain the solution of the simplified model defined in (5), we need to first initialize the unknown quantities $\mathbf{X}$ and $\mathbf{Y}$ respectively.

For the initialization of the *community membership matrix* $\mathbf{X}$, we utilize (1) as the objective function. Note that such initialization problem is also an NMF problem, and we can directly use the following updating rule provided by the supplementary information of [29] to update the value of $\mathbf{X}$ in each iteration:

$$\mathbf{X}_{ir} \leftarrow \mathbf{X}_{ir} \frac{(\mathbf{A}\mathbf{X})_{ir}}{(\mathbf{X}\mathbf{X}^T\mathbf{X})_{ir}}. \tag{6}$$

To initialize $\mathbf{X}$, we first set its entries to be random nonnegative values, and use (6) to update it until converges.

Similarly, we initialize the *community attribute matrix* $\mathbf{Y}$ by solving the following optimization problem (7), where we replace the $\mathbf{X}$ which encoded the information about topology in (2) with another matrix $\mathbf{Z} \in \Re^{N \times K}$ to eliminate the influence of topological structure.

$$\arg\min_{\mathbf{Y},\mathbf{Z}} \left\| \mathbf{C} - \mathbf{Z}\mathbf{Y}^T \right\|_F^2 \quad \text{s.t.} \quad \mathbf{Z}_{ir} \geq 0, \quad \mathbf{Y}_{sr} \geq 0. \tag{7}$$

Under such circumstance, $\mathbf{Z}_{it}$ represents the propensity that the node i belongs to the "topic" t, so we name it as the *"topic" membership matrix*. As (7) is also a standard NMF problem, after randomly setting $\mathbf{Z}$ and $\mathbf{Y}$ with nonnegative values, we adopt the following updating rules given by [34] to respectively update their values in turn:

$$\mathbf{Z}_{ir} \leftarrow \mathbf{Z}_{ir} \frac{(\mathbf{C}\mathbf{Y})_{ir}}{(\mathbf{Z}\mathbf{Y}^T\mathbf{Y})_{ir}}, \quad \mathbf{Y}_{sr} \leftarrow \mathbf{Y}_{sr} \frac{(\mathbf{C}^T\mathbf{Z})_{sr}}{(\mathbf{Y}\mathbf{Z}^T\mathbf{Z})_{sr}}. \tag{8}$$

When update $\mathbf{Z}$ we fix the value of $\mathbf{Y}$, and it's the same for the updating of $\mathbf{Y}$.

For the formal solving strategy of the unified model (5) which is not convex, we use the same solution strategy as in [29] to, in turn, take the following two steps. First,

we fix the value of **Y** and update **X** with rule (9). Secondly, we fix **X** and update **Y** with rule (10). (Note that that the derivation processes of (9) and (10) are similar to (18), which will be discussed later in Section 4.4.)

$$\mathbf{X}_{ir} \leftarrow \mathbf{X}_{ir} \frac{\left(2\alpha\mathbf{A}\mathbf{X}+\mathbf{C}\mathbf{Y}\right)_{ir}}{\left(2\alpha\mathbf{X}\mathbf{X}^{T}\mathbf{X}+\mathbf{X}\mathbf{Y}^{T}\mathbf{Y}\right)_{ir}} \tag{9}$$

$$\mathbf{Y}_{sr} \leftarrow \mathbf{Y}_{sr} \frac{\left(\mathbf{C}^{T}\mathbf{X}\right)_{sr}}{\left(\mathbf{Y}\mathbf{X}^{T}\mathbf{X}\right)_{sr}} \tag{10}$$

Basic Proof-of-Concept for the Mismatch Effect.
Based on the unified model introduced above, we designed the following pre-experiment to illustrate the different influences of semantic information with different trade-offs between topology and content.

In the pre-experiment, we applied the unified model (5) to two real network datasets: *Cornell*[1] and *Facebook* [10] (the details about the datasets can be found in Section 5.1). Moreover, we adopt normalized mutual information (NMI) [7] between the label sequences given by the community partition result and the dataset's ground-truth as the evaluation metric, and higher NMI value means better community partition result.

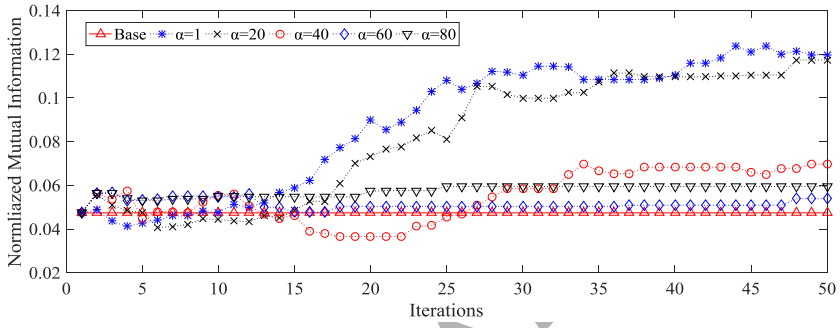
We executed above initialization step of **X** 10 times and adopted the NMI value with respect to the initialization setting that resulted in the minimum value of (1) as the baseline. To highlight the influence of different trade-offs between topology and content, we first set α=1 to assign the same contribution of the topology and the content parts in the unified model (5). Then, we gradually increase the contribution of topology by respectively setting $\alpha = 20$, α=40, α=60 and α=80.

For each parameter setting, we used the solution strategy introduced above (namely (9) and (10)) to get the optimal solution. Especially, for the first 50 iterations, we calculated and recorded the NMI value in each iteration according to current **X**. Finally, we obtained the convergence curves of the NMI for both (a) ***Cornell*** and (b) ***Facebook*** datasets shown in Figure 1.

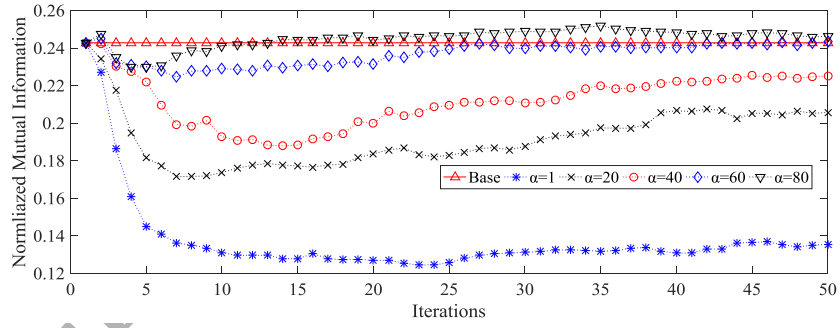
For the result of *Cornell* shown in Figure 1 (a), when α=1 and $\alpha = 20$, the NMI values both gradually increase in the first several iterations and finally stabilize at a level that outperforms the baseline. Moreover, when we set α=40, α=60 and α=80, the corresponding NMI values do not have obvious increase and all of them stay at a level slightly better than the baseline. Note that such result is in line with the expectation of most state-of-the-art community detection methods that incorporate topology and content. The community partition of the *Cornell* dataset is further improved with the addition of content information, and the only difference among the convergence curves (excluding the baseline) is due to the different setting of the hyper-parameter α. In other words, under certain circumstances, the integration of topology and content can really help extract better community structures corresponding to real groups or organizations in social networks.

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

On the other hand, with regard to the result of *Facebook* in Figure 1 (b), when α=1, the corresponding NMI value decreases dramatically in the first iterations, and finally stays at a level that is much lower than the baseline. Such decrease of the NMI reflects the fact that the introduction of content in the unified model (5) has a very negative impact on the extracted community structure, which can be seen as a typical example of the *mismatch* between topology and content. Nevertheless, when $\alpha>1$, although we can still observe a decrease of the NMI value in the first iterations, the degree of decrease is smaller than that when α=1, and the larger the value of α the less the NMI will decrease. Especially when α=40, α=60 and α=80 the corresponding NMI values all gradually recover to a level near to the baseline.



(a) ***Cornell***



(b) ***Facebook***

Fig. 1. NMI convergence curves of the unified model (5) for the (a) ***Cornell*** and (b) ***Facebook*** datasets with respect to α=1, α=2, α=40, α=60 and α=80. For the result of (a) ***Cornell*** the NMI values for all the parameter settings of α finally converge to levels larger than the baseline, which is in line with the expectation of some community detection incorporating topology and content. For the result of (b) ***Facebook***, when α=1 the NMI value dramatically decreases and finally converges to a low level having a big gap to the desirable baseline, which can be seen as a typical example of the *mismatch* between topology and content. When $\alpha>1$, although the NMI value decreases in the first iterations, it finally recovers to a higher level.

Note that Figure 1 do not contain all the details about the experimental result since the updating processes with respect to different parameter setting (except for the baseline) haven't completely converged in the first 50 iterations. For each updating

process with a certain set of α, we continuously recorded the NMI value in each iteration until it converged. In the pre-experiment on *Facebook* (Figure 1 (b)), the NMI value of the baseline was **0.2428**. For different parameter settings of α, the converged NMI values were respectively **0.1350** (α=1), **0.2093** (α=20), **0.2647** (α=40), **0.2613** (α=60) and **0.2633** (α=80). Especially, when α is set to be a relative large value (such α=40, α=60 and α=80), the converged NMI value could even slightly outperform the baseline level rather than just recovering to the baseline. In other words, it is still possible for the unified model to achieve a result relatively better than the baseline even when the *mismatch* occurs (if the contribution of topology is large enough).

Given the results in Figure 1 (b), we conclude that despite the *mismatch* between the topology and content, we can still control the negative impact resulted from the incorporation of content if we appropriately set the trade-off between them. More importantly, the greater the degree of *mismatch*, the greater contribution the topology part of the objective function should have.

4.4 The Formal ASCD Model and Algorithm (Based on the Mismatch Effect)

Based on the above conclusion we have reached in Section 4.3, we can now propose our formal unified model in this subsection.

The Adaptive Parameters.
As we respectively initialize **X** and **Y** in the solving process of (5), and use (3) but not (2) as the content part of the uniform model (5), when the initialization step finishes or after each iteration, (2)'s value can reflect the degree of *mismatch* between topology and content, namely the larger the value of (2), the greater the degree of *mismatch*. Because the (2)'s value is also related to the size of the network, we introduce the following formula to get an average value of (2), which can eliminate the effect of the network size:

$$d(\mathbf{X},\mathbf{Y}) = \sqrt{\left\|\mathbf{C}-\mathbf{X}\mathbf{Y}^T\right\|_F^2 \Big/ (NM)}. \quad (\mathbf{11})$$

Then we combine (11) with a monotonically decreasing function of *arctan* within the interval of [0, 1] to derive the following adaptive penalty parameter, which will be set as the coefficient of the content part in our model:

$$f(\mathbf{X},\mathbf{Y}) = -2\arctan\left(\delta \cdot d(\mathbf{X},\mathbf{Y})\right)/\pi + 1, \quad (\mathbf{12})$$

where δ is the parameter to adjust the value of $d(\mathbf{X},\mathbf{Y})$. In fact, the value of $f(\mathbf{X},\mathbf{Y})$ is inversely proportional to the degree of *mismatch* between topology and content. In other word, if the *mismatch* degree is large, the parameter $f(\mathbf{X},\mathbf{Y})$'s value will be small, and the contribution of the content in the model can be effectively controlled. On the other hand, when the degree of *mismatch* is small, the value of (12) will be relatively large, so the content part can fully play its role.

Furthermore, different from the adaptive parameter (12), which focuses on the *mismatch* degree between topology and content, we also design another type of such parameter from the perspective of *match* degree.

In the process of initializing the *community attribute matrix* **Y** (with (7) as the objective function and (8) as the updating rules) in the simplified model (5), we introduced another *"topic" membership matrix* **Z** to replace the *community membership matrix* **X**, in order to eliminate the influence of topology given by **X**. In fact, the original **X** and **Z** can be used to extract the clustering structure of topology and content. By utilizing the same method of extracting each node's community label from matrix **X**, we can also get the "topic" label of each node from matrix **Z**. If we respectively organize the community labels and the "topic" labels of each node as two label sequences according to the node index, then the NMI [7] between such two label sequences may reflect the similarity between the clustering structures of topology and content, which can directly reflect the *match* degree. Namely, larger NMI value means better match correspondence between topology and content. Although the *"topic" membership matrix* **Z** may not be used and updated in the process of solving the unified model, we can still derive another matrix $\mathbf{Z}' = \mathbf{CY}$ after each iteration to represent a similar clustering structure of content. In this way, we can still evaluate the *match* degree in a certain iteration by similarly calculating the NMI value.

Therefore, based on the NMI value's directly proportional relation to the *match* degree, we introduce the following adaptive parameter:

$$f\left(\mathbf{X},\mathbf{Y}\right) = \delta \cdot \mathrm{NMI}\left(\mathrm{s}\left(\mathbf{X}\right), \mathrm{s}\left(\mathbf{CY}\right)\right) = \delta \cdot \mathrm{NMI}\left(\mathrm{s}\left(\mathbf{X}\right), \mathrm{s}\left(\mathbf{Z}\right)\right), \tag{13}$$

where $\mathrm{NMI}\left(\mathrm{s}_1, \mathrm{s}_2\right)$ is the function to calculate the NMI between label sequence s_1 and s_2, $\mathrm{s}\left(\mathbf{M}\right)$ is the function to extract the top label sequence form matrix **M**, and δ is the parameter to control the value of NMI. Based on the above definition, relative small *mismatch* degree (large *match* degree) may result in a relative large value of (13), so the content part can fully play it role of improving the community partition. When the *mismatch* degree is large (the *match* degree is small), the value of (13) will be small, so the contribution of the content can be effectively controlled.

The Objective Function (Based on the Adaptive Parameters).
By adding additional adaptive parameter and the sparse item discussed in (4), we have formulated the following objective function of our Adaptive Semantic Community Detection (ASCD) method:

$$\arg\min_{\mathbf{X}\geq 0, \mathbf{Y}\geq 0} \left\|\mathbf{A} - \mathbf{X}_{(t)}\mathbf{X}_{(t)}^{T}\right\|_F^2 + f \cdot \left\|\mathbf{X}_{(t)} - \mathbf{CY}_{(t)}\right\|_F^2 + \lambda \sum_{r=1}^{K} \left\|\mathbf{Y}_{(t)}\left(:, r\right)\right\|_1^2, \tag{14}$$

where f is the simplified representation of $f\left(\mathbf{X}_{(t-1)}, \mathbf{Y}_{(t-1)}\right)$, and the subscript of $\mathbf{X}_{(t)}$ and $\mathbf{Y}_{(t)}$ represents their corresponding value in the *t*-th iteration. Especially, subscript 0 infers the value after initialization, and *t* in (14) should start from 1. Note that (14) is only defined for the *t*-th iteration in our ASCD model, since the value of f is dependent on $\mathbf{X}_{(t-1)}$ and $\mathbf{Y}_{(t-1)}$ which may differ in different iterations.

In this paper, we use notation f to represent the adaptive parameter in a general form, since the corresponding solving strategy is independent of the type of adaptive parameter. For the convenience of further discussion, we use ASCD-ARC to represent the condition that we use (12) as the adaptive parameter, while use ASCD-NMI to represent the condition when (13) is utilized.

Basic ASCD Algorithm for Disjoint Community Detection.
In order to achieve the solution of (14), we adopt the block coordinate descent approach as in [29] to solve such non-convex problem and take the following two steps in turns.

1) Updating **X** with **Y** fixed

First, we fix the value of **Y** and solve the following optimization problem only related to **X**:

$$\arg\min_{\mathbf{X}\geq 0} \mathrm{O}(\mathbf{X}) = \left\|\mathbf{A}-\mathbf{X}\mathbf{X}^T\right\|_F^2 + f\cdot\left\|\mathbf{X}-\mathbf{C}\mathbf{Y}\right\|_F^2. \tag{15}$$

By using the property that $\|\mathbf{M}\|_F^2 = \mathrm{tr}\left(\mathbf{M}\mathbf{M}^T\right)$, we can get the partial derivative of $\mathrm{O}(\mathbf{X})$ with respect to **X**:

$$\frac{\partial O(\mathbf{X})}{\partial \mathbf{X}} = \left(2f\mathbf{X}+4\mathbf{X}\mathbf{X}^T\mathbf{X}\right)-\left(2f\mathbf{C}\mathbf{Y}+4\mathbf{A}\mathbf{X}\right) = [\cdot]_+ - [\cdot]_-. \tag{16}$$

Here we use a simplified notation $[\cdot]_+$ to represent all the terms with positive coefficient, and use $[\cdot]_-$ to represent those with negative coefficient. In the case of (16), we have $[\cdot]_+ = 2f\mathbf{X}+2\mathbf{X}\mathbf{X}^T\mathbf{X}$ and $[\cdot]_- = 2f\mathbf{C}\mathbf{Y}+4\mathbf{A}\mathbf{X}$.

Based on the standard gradient descent method, we can derive the following additive updating rule:

$$\begin{aligned}\mathbf{X}_{ir} &\leftarrow \mathbf{X}_{ir} - \eta_{ir}\left(\left(2f\mathbf{X}+4\mathbf{X}\mathbf{X}^T\mathbf{X}\right)-\left(2f\mathbf{C}\mathbf{Y}+4\mathbf{A}\mathbf{X}\right)\right)\\ &= \mathbf{X}_{ir} - \eta_{ir}\left([\cdot]_+ - [\cdot]_-\right)\end{aligned}, \tag{17}$$

where η_{ir} is the learning rate. Similar to the derivation discussed in [30], if we appropriately set $\eta_{ir} \leftarrow \mathbf{X}_{ir}/\left([\cdot]_+\right)_{ir}$, then we can derive the following multiplicative updating rule:

$$\left(\mathbf{X}_{(t+1)}\right)_{ir} \leftarrow \left(\mathbf{X}_{(t)}\right)_{ir}\frac{\left([\cdot]_-\right)_{ir}}{\left([\cdot]_+\right)_{ir}} = \left(\mathbf{X}_{(t)}\right)_{ir}\frac{\left(f\cdot\mathbf{C}\mathbf{Y}_{(t)}+2\mathbf{A}\mathbf{X}_{(t)}\right)_{ir}}{\left(f\cdot\mathbf{X}_{(t)}+2\mathbf{X}_{(t)}\mathbf{X}_{(t)}^T\mathbf{X}_{(t)}\right)_{ir}}. \tag{18}$$

2) Updating **Y** with **X** fixed

Next, we fix **X** and derive the updating rule of **Y**. Similarly, we get the following objective function only related to **Y**:

$$\arg\min_{\mathbf{Y}\geq 0} \mathrm{O}(\mathbf{Y}) = f \cdot \left\|\mathbf{X} - \mathbf{C}\mathbf{Y}\right\|_F^2 + \lambda \sum_{j=1}^{K} \left\|\mathbf{Y}(:, j)\right\|_1^2 . \quad (19)$$

For the sparse item of 1-norm in (19), we used the same method of solving the SNMF problem as in [28] to derive another equivalent objective function:

$$\mathrm{O}(\mathbf{Y}) = f \cdot \left\| \begin{bmatrix} \mathbf{X} \\ \mathbf{0}_{1\times K} \end{bmatrix} - \begin{bmatrix} \mathbf{C} \\ \sqrt{\lambda}\mathbf{e}_{1\times M} \end{bmatrix} \mathbf{Y} \right\|_F^2 = f \cdot \left\|\mathbf{X}' - \mathbf{C}'\mathbf{Y}\right\|_F^2 , \quad (20)$$

where $\mathbf{0}_{1\times K}$ is a K-dimensional row vector whose elements all equal 0, while $\mathbf{e}_{1\times M}$ is an M-dimensional row vector whose elements are all 1. We then derive the partial derivative of $\mathrm{O}(\mathbf{Y})$ with respect to $\mathbf{Y}$:

$$\frac{\partial \mathrm{O}(\mathbf{Y})}{\partial \mathbf{Y}} = 2f \cdot \mathbf{C}'^T\mathbf{C}'\mathbf{Y} - 2f \cdot \mathbf{C}'^T\mathbf{X}' . \quad (21)$$

Finally, by using the same derivation method as in (18), we derive the following updating rule for $\mathbf{Y}$:

$$\left(\mathbf{Y}_{(t+1)}\right)_{sr} \leftarrow \left(\mathbf{Y}_{(t)}\right)_{sr} \frac{\left([\cdot]_-\right)_{sr}}{\left([\cdot]_+\right)_{sr}} = \left(\mathbf{Y}_{(t)}\right)_{sr} \frac{\left(\mathbf{C}'^T\mathbf{X}'_{(t)}\right)_{sr}}{\left(\mathbf{C}'^T\mathbf{C}'\mathbf{Y}_{(t)}\right)_{sr}} . \quad (22)$$

Therefore, we can in turns take the two steps introduced above and respectively use (18) and (22) to update the value of $\mathbf{X}$ and $\mathbf{Y}$, until they converge. For the convergence criterion, we adopt the strategy based on the relative error of the value of objective function (14). In each iteration, we record current value of (14) and calculate the relative error with respect to last iteration. If the relative error is smaller than a pre-set threshold (such as 10^{-6}), we determine that the model has converged. Otherwise, the updating process should continue.

Especially, for the inequality constraint defined in the NMF problem (in our model the constraints are $\mathbf{X}_{ir} \geq 0$ and $\mathbf{Y}_{sr} \geq 0$), [30] has proved that if we initialize $\mathbf{X}$ and $\mathbf{Y}$ as non-negative matrixes, then the non-negativity of them can be guaranteed by using their corresponding multiplicative updating rules ((18) and (22)). Furthermore, as the *adjacency matrix* $\mathbf{A}$ and the *node attribute matrix* $\mathbf{C}$ may be sparse matrixes, the multiplicative updating rule can also be computed efficiently in each iteration.

Based on the above discussion, we can now conclude the basic ASCD algorithm for disjoint community detection in the form of pseudo-code in Table 1.

The result of the proposed algorithm is two-fold: we can not only extract the community structures from $\mathbf{X}$, but also can obtain the most relevant keywords for each community according to $\mathbf{Y}$. In other words, the ASCD method has the powerful capacity to simultaneously obtain the community partition result and the corresponding semantic interpretation.

Table 1. The pseudo-code of the ASCD algorithm for disjoint community detection, where *N*, *M*, *K* are respectively the number of nodes, attributes and communities, while **A**, **C**, **X**, **Y** and **Z** are respectively the *adjacency matrix*, the *node attribute matrix*, the *community membership matrix*, the *community attribute matrix* and the auxiliary *"topic" membership matrix*.

ASCD Algorithm for Disjoint Community Detection

Input: *N*, *M*, *K*, **A**, **C**

Output: **X**, **Y**

Randomly set **X**, **Y** and **Z** to be non-negative values

while (1) **not** *converge* //Initialize **X**

Update **X** via (6)

end while

while (7) **not** *converge* //Initialize **Y**

Update **Y** and **Z** in turn via (8)

end while

while (14) **not** *converge* //Update the value of **X** and **Y**

Calculate $f(\mathbf{X},\mathbf{Y})$ via (12) (ASCD-ARC) or (13) (ASCD-NMI)

Update **X** via (18)

Update **Y** via (22)

end while

Verifying the Algorithm's Convergence.

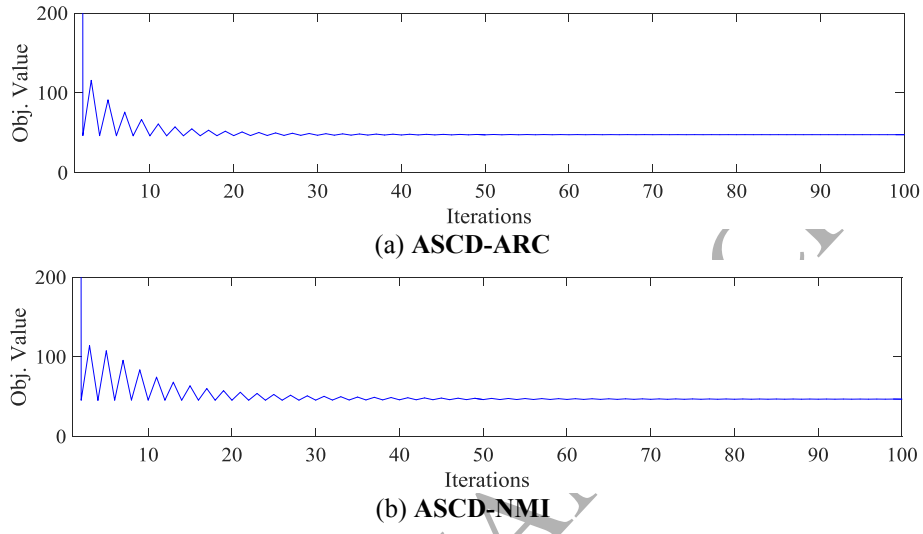
For the algorithm introduced above, if the value of the adaptive parameter *f* is fixed in all iterations, the convergence of the updating rules ((18) and (22)) can be ensured, because it's based on the standard gradient descent process. But for a non-fixed value of *f*, the convergence of our algorithm needs to be specially discussed.

We applied the two derivative forms of our ASCD method (ASCD-ARC and ASCD-NMI) to an artificial network that we used in the artificial network analysis experiment (which will be introduced in Section 5.1). For both ASCD-ARC and ASCD-NMI, we respectively recorded the values of objective function (14) in each iteration to draw two convergence curves, which are shown in Figure 2 (where (a) and (b) are respectively the curve of ASCD-ARC and ASCD-NMI).

As shown in Figure 2, for both ASCD-ARC and ASCD-NMI methods, the values of the objective function converge fast straight from the beginning and ASCD-ARC converges faster than ASCD-NMI. Moreover, for each iteration, we also recorded the relative error of objective function (14) (corresponding to last iteration). With regard to ASCD-ARC, the relative error reached the precision of 10^{-4} after about the first 70 iterations, and the precision reached 10^{-6} after about the first 130 iterations. For ASCD-NMI, the precision of relative error reached 10^{-4} after about the first 110 iterations, and after about the first 250 iterations the precision reached the level of 10^{-6}. As a further verification, we also applied the two derivative forms of our ASCD

method to other artificial and real networks (whose details can be found in Section 5.1), and the convergence we observed was similar to the results shown above.

Therefore, the convergence of the above algorithm (with updating rules (18) and (22)) can still be ensured, even though the value of parameter *f* is non-fixed.



(a) **ASCD-ARC**

(b) **ASCD-NMI**

Fig. 2. The convergence curves of objective function (14)'s value for (a) **ASCD-ARC** and (b) **ASCD-NMI**. The curves in (a) and (b) both converge fast in the first serval iterations. Therefore, although the parameter *f* is non-fixed in the unified model (14), the convergence of our algorithm can still be guaranteed.

Extended ASCD Algorithm for Overlapping Community Detection.
The basic ASCD algorithm introduced above can only extract the disjoint community structures in a certain network, where each node can only be the member of single community. However, overlapping community detection is also another significant task of social network analysis, in which each node in the network can be assigned multiple (one or more) community labels.

In fact, the above algorithm does not fully utilize the topological information encoded in the *community membership matrix* **X**, as we only focus on the element with maximum propensity value in each row. By further mining the information encoded in **X**, we can also introduce an extended algorithm to meet the application requirements of discovering overlapping communities.

To determine the possible community labels of node *i*, we divide the elements of the *i*-th row into two disjoint sets, and respectively name them as the "accepted" set and the "rejected" set. When an element in the *i*-th row of **X** is partitioned into the "accepted" set, we tend to accept the column index of such element as one of the community labels of node *i*. But when the element is in the "rejected" set, we tend to reject its column index as the corresponding community label. Hence, the key of our extended overlapping community detection algorithm is to determine the *boundary* between the "accepted" set and the "rejected" set for each row.

If we sort a row of **X** in descending order and list the elements from left to right, the elements close to left are more likely to be partitioned into the "accepted" set, and for those which are close to right are more likely in the "rejected" set. Hence, the *boundary* of such two parts should be at the position that has the maximum difference in the values of propensity.

Table 2. The pseudo-code of ASCD extended algorithm for overlapping community detection, where N and K are respectively the number of nodes and communities, while **X** is the *community membership matrix* and L_r is the *node membership set* of community r.

ASCD Extended Algorithm for Overlapping Community Detection

```
Input: N, K, X
Output: L={L_1, L_2, …, L_K}
  //Initialize the node membership set of each community
  for each community r
     L_r ←null
  end for
  for each node i
     sort X(i,:) by descending order
     //s and t are two auxiliary sequences
     assign the sorted result to s
     assign the corresponding row index sequence to t
     //Find the boundary between "accepted" and "rejected" parts
     pos ← 1 //Current boundary position
     max_dif ← (s(2)-s(1)) //Current maximum difference in s
     for r from 2 to (K-1)
        cur_dif ←(s(r+1)-s(r))
        if cur_dif > max_dif
           max_dif ← cur_dif
           pos ← r
        end if
     end for
     for r from 1 to pos //Update the node membership set L
        add node i into Lt(r)
     end for
end for
```

According to the definition in Section 3, to represent overlapping community partition result, one can maintain a *node membership set* for each community, which can be notated as $L=\{L_1,\cdots,L_K\}$ where L_r is the node set of community r's

members. During the process of discovering overlapping communities, if node *i* is a member of community *r*, then we add it into the set L_r.

As a conclusion, we use pseudo-code to describe our extended algorithm of overlapping community detection in Table 2.

5 Experimental Evaluation

In this section, a series of experiments are conducted to comprehensively evaluate our method's performance and the organization is as follow. Section 5.1 summarizes all the datasets and evaluation metrics we used in the following experiments. Section 5.2 introduces a parametric analysis experiment to discuss the recommended setting of our model's hyper-parameters. Section 5.3 discusses the results of the artificial analysis, including the special analysis about the *attribute-refining* (AR) effect, which can be used to further improve the community partition, and the general evaluation for our method's capacity to resist the mismatch effect. Thereafter, Section 5.4 introduces a case study to demonstrate our model's ability to simultaneously obtain the community partition result and the corresponding semantic description. Finally, Section 5.5 presents the results of the performance evaluation on several real network datasets for both disjoint and overlapping community detection.

5.1 Datasets and Evaluation Metrics

The Artificial Networks.
In order to generally evaluate our method's performance and robustness (such as the capacity to resist the *mismatch* effect) under an artificially controllable circumstance, we used the method proposed in [1] to generate the topological structure of a sort of random networks and used a binomial distribution to generate the corresponding content.

For the topology, we generated 128 nodes and evenly partitioned them into 4 communities, so each community has 32 nodes. For each node, we randomly generate 8 edges (z_{in}) that connected to other nodes in the same community, and also randomly generate 8 edges (z_{out}) connecting to those in different communities, which can be seen as the noise of topology. Therefore, each node has 16 edges ($z = z_{\text{in}} + z_{\text{out}}$) in total.

For the content information, we set the total number of attributes (keywords) to be 128, and assumed that there were 4 topics which had a one-to-one correlation to the 4 communities. Furthermore, we set the total number of attributes of each node to be 32 ($h = h_{\text{in}} + h_{\text{out}}$), and only 24 (h_{in}) of them were relevant to the node's topic, so there remained 8 attributes (h_{out}) that were irrelevant to the topic, which can be seen as the noise of content. For each node, we generated a 128-dimensional binary vector, in which the values of only 32 elements were one and others were zero. With respect to an arbitrary node *i*, if it belonged to community *r*, we set the values of elements in its

attribute vector whose indexes ranged from $(r-1)\times h$ to $r\times h$ to be one with the probability of h_{in}/h, while we set the rest elements whose indexes were not in the range of $\left[(r-1)\times h,\ r\times h\right]$ to be one with the probability of $h_{\text{out}}/(3h)$.

Because of the one-to-one correlation between community and topic, the clustering structures of topology and content are consistent. To simulate the *mismatch* between topology and content, we randomly select a certain proportion of nodes to swap their attribute vectors, in which we introduced a parameter ρ_{mis} to represent such proportion. In fact, ρ_{mis} controls the degree of *mismatch* between topology and content. Namely, the larger the value of ρ_{mis} is, the larger the degree of this mismatch will be. Especially, when $\rho_{\text{mis}}=1$, all nodes in the network may be randomly selected to swap their attribute vectors, which represents the maximal degree of mismatch we can control. Hence, we set ρ_{mis} to be values from 0 to 1 with step size of 0.1.

Note that we randomly selected nodes, attribute elements and attribute vectors to be swapped according to certain probability distributions during the generative process. In order to achieve a relatively stable result, we finally generated 50 different networks for each setting of ρ_{mis} (with $11\times 50=550$ networks in total).

The Real Network Datasets.

Besides the artificial networks, we also collected and preprocessed 13 real social network datasets with topological structure and attribute information for our evaluation experiments, in which 12 of them provide the ground-truth about the number of communities and the corresponding community membership, while the rest one doesn't have related ground-truth.

Moreover, 8 of the 12 datasets with ground-truth were utilized for the evaluation of disjoint community detection, because their ground-truths are disjoint, in which each node only have single community label. On the other hand, the ground-truths of the rest 4 datasets are overlapping, since each node in the network can simultaneously be the member of multiple (one or more) communities according to the ground-truth.

As a summary, we list the details of the above 13 real network datasets in Table 3, where *N*, *E*, *M* and *K* are the number of nodes, edges, keywords and communities after preprocessing, while *G* is the type of ground-truth. For the ground-truth type, we use "*No*" to represent the datasets without ground-truth, while those with disjoint and overlapping ground-truth are respectively notated as "*D*" and "*O*".

In the experiment, we used the *last.fm* dataset [9] to illustrate our method's capacity to simultaneously derive the partition of communities and the corresponding semantic description. The dataset was collected from an online music platform *last.fm* (http://www.lastfm.com) which includes users' friendships (topology) and interest tags (content). The friend relationship is represented as the undirected and unweighted edges in the network. After preprocessing, we derived a network of 1,829 nodes with 12,712 undirected edges and a total of 9,749 keywords. Sine the dataset didn't have the ground-truth about the number of communities (*K*), we used the Louvain algorithm [17] (which can automatically determine the number of communities and

the corresponding community membership with optimal modularity) to determine the appropriate value of K, and we finally set K=38.

Table 3. Details of the real social network datasets we utilized in the experiment, where N, E, M and K are the number of nodes, edges, keywords and communities respectively, while G is the ground-truth type. "*No*", "*D*" and "*O*" are used to described the dataset without ground-truth, with disjoint ground-truth and with overlapping ground-truth.

Datasets	***N***	***E***	***M***	***K***	***G***	**Brief Description** [9,10,11,12,13]
Last.fm	1,892	12,712	9,749	38	*No*	Dataset collected from an online music platform
Cornell	195	304	1,703	5	*D*	Subnetworks of four American universities in the *WebKB* dataset
Texas	187	187		5		
Washington	230	446		5		
Wisconsin	265	530		5		
Cora	2,708	5,429	1,433	7		A citation network
Citeseer	3,312	4,732	3,703	6		A citation network
Twitter	171	796	578	8		Largest subnetwork of *Twitter* dataset from *SNAP*
Facebook	1,045	26,749	576	10		Largest subnetwork of *Facebook* dataset from *SNAP*
Enron	974	1,557	15,382	13	*O*	*Enron* mail dataset
Reddit25	1,314	1,339	4,616	3		*Reddit* dataset with time slice of 2012-8-25
Reddit26	1,590	1,714	5,055	3		*Reddit* dataset with time slice of 2012-8-26
Reddit27	2,143	2,290	6,635	3		*Reddit* dataset with time slice of 2012-8-27

In the evaluation experiment of disjoint community detection, we applied our new method to 8 real social network datasets with disjoint ground-truth. In the 8 datasets, *Cornell*, *Texas*, *Washington* and *Wisconsin* are 4 subnetworks of American universities in the *WebKB* dataset[1]. *Cora* and *Citeseer* are two citation networks whose source details can be found in [11], while *Twitter* and *Facebook* are two largest subnetworks of the datasets named as ego-Twitter and ego-Facebook from the Stanford Large Network Dataset Collection (SNAP) [10]. As these datasets provide clear formats about the networks' topology and attribute, we directly utilized them as the input of a certain method after some simple preprocessing.

For the detection of overlapping communities, we selected two real network datasets (*Enron* [12] and *Reddit* [13]) with edge-induced content and overlapping node-induced ground-truth as the testing datasets. Since the type of content (such as node-induced content edge-induced content) is not our major concern in this paper (but we intend to consider the edge-induced content in our future work), we assume the content of an edge is shared by the edge's two nodes and finally converted the edge-induced content into the node-induced form.

Enron is a labeled subnetwork of the Enron corporation's email system. For each email in the dataset, we extracted the topological relations according to the addresses of the sender and the receiver(s) (an email can be sent to multiple users), and we considered the 13 categories of "primary topics" (which are manually annotated according to the content of emails) as the ground-truth. Since the content and ground-truth are both edge-induced, we further transformed them into the node-induced forms by simultaneously assigning each email's content and labels to its sender and

[1] http://www.cs.cmu.edu/afs/cs/project/theo-20/www/data/

receiver(s). Finally, we derived a network with node attribute and overlapping node-induced ground-truth.

The *Reddit* dataset contains posts and comments of three sub-forums in the website of Reddit (www.reddit.com) from 2012-8-25 to 2012-8-31. In Reddit, a user can choose a sub-forum (such as Movies and Politics) to post content, and other users can then post comments on such content. Therefore, we extracted the topological relations according to the ID of the post author and the comment authors. Furthermore, the sub-forums in which a user posts content (comments) can be seen as the ground-truth, and we further found that such ground-truth was overlapping (namely there exist users who have post comment in multiple sub-forums). For the edge-induced content, we also converted it into the node-induced form by simultaneously assigning the content to the nodes with respect to the post author and the comment authors.

What's more, in order to have more datasets to be evaluated, we extracted three time slices (2012-8-25, 2012-8-26 and 2012-8-27) of *Reddit* to get 3 different subnetworks, so we finally have 4 testing datasets (*Enron*, *Reddit25*, *Reddit26* and *Reddit27*) for the evaluation of overlapping community detection.

Evaluation Metric for Disjoint Community Detection.

To evaluate the disjoint community partition result given by a certain method, we used the normalized mutual information (NMI) [7] and accuracy (AC) [7] as the evaluation metrics by comparing the method's result with the corresponding ground-truth. According to [7], NMI can be used to measure the similarity between two sets of clusters, while AC represents the percentage of correct labels obtained by the method to be evaluated. Generally, larger NMI or AC means better correspondence between the communities extracted and the groups (or organizations) in real networks.

Let L be the community label sequence given by the method to be evaluated and R be the corresponding sequence with respect to dataset's ground-truth. Also, we let the subscript i of R_i and L_i represent the community label of node i. The definitions of NMI and AC can be described as follows:

$$\mathrm{NMI}(R,\ L)=\frac{2\,\mathrm{MI}(R,\ L)}{\mathrm{H}(R)+\mathrm{H}(L)}, \tag{23}$$

$$\mathrm{AC}(R,\ L)=\frac{\sum_{i=1}^{N}\delta\left(R_i,\mathrm{map}(L_i)\right)}{N}. \tag{24}$$

In (23), $\mathrm{MI}(R,L)=\sum_{r\in R}\sum_{l\in L}\frac{n_{r,l}}{n}\log\frac{n\times n_{r,l}}{n_r\times n_l}$ is the *mutual information* between R and L, while $\mathrm{H}(S)=-\sum_{s\in S}\frac{n_s}{n}\log\frac{n_s}{n}$ is the *entropy* of the label sequence S. With regard to (24), $\mathrm{map}(\cdot)$ is a mapping function of the Kuhn-Munkres algorithm [31] to map a sequence to another equivalent form.

Evaluation Metric for Overlapping Community Detection.
Different from the representation of disjoint community partition, in the experiment of overlapping community detection, we maintained a *node membership set* for each community, namely $L=\{L_1,\cdots,L_K\}$, to represent the overlapping partition result. Specially, if node *i* belongs to community *r*, then *i* is an element of the node set L_r.

Based on the above definition, we used the evaluation methods proposed in [32], which generalized the F-score and Jaccard metrics to the scenario of overlapping community detection with the following expression:

$$\frac{1}{2|L|}\sum_{L_i\in L}\max_{R_j\in R}\varphi\left(L_i,R_j\right)+\frac{1}{2|R|}\sum_{R_j\in R}\max_{L_i\in L}\varphi\left(L_i,R_j\right), \quad (25)$$

where $\varphi\left(L_i,R_j\right)$ is the similarity measure (F-Score or Jaccard), and *L* as well as *R* are respectively the overlapping partition results given by the method to be evaluated and the testing dataset's ground-truth. Moreover, L_i is the *node membership set* of community *i* given by *L*, while R_j is community *j*'s *node membership set* according to *R*. Generally, the larger value of the generalized F-score (or Jaccard) means better performance of overlapping community partition.

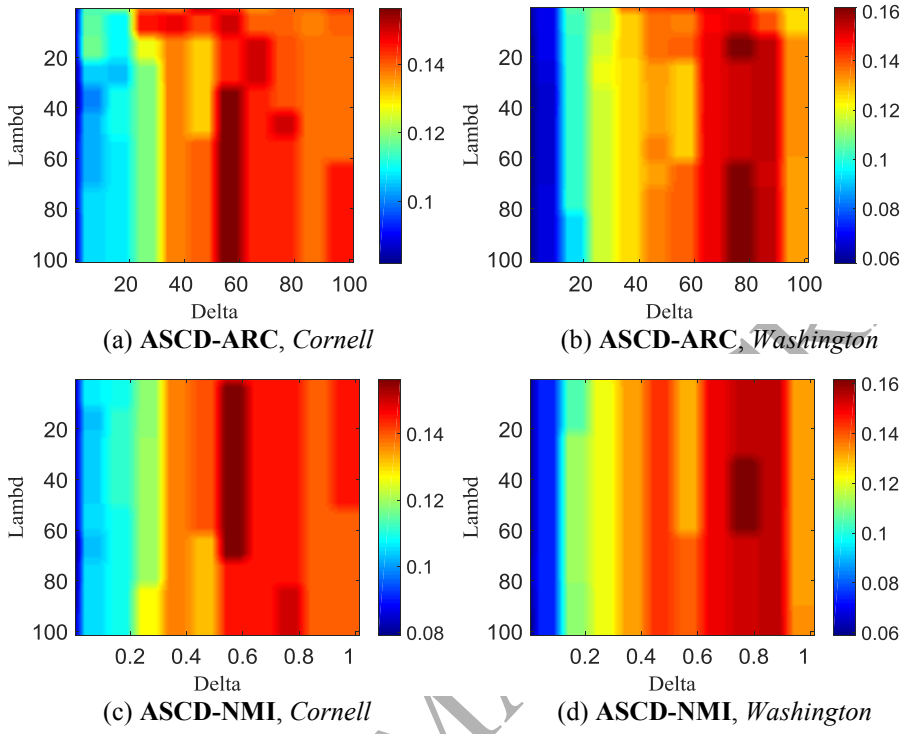
5.2 Parameter Adjustment Analysis

As δ and λ are the hyper-parameters of our ASCD model, different settings of them may lead to different results. For both ASCD-ARC and ASCD-NMI, we use the parameter λ to control the sparsity item of **Y** in the objective function (14). With regard to δ, we utilize it to respectively adjust the value of $d(\mathbf{X},\mathbf{Y})$ in (12) (for ASCD-ARC) and NMI in (13) (for ASCD-NMI).

So as to further analyze the effect of the hyper-parameters, we respectively applied ASCD-ARC and ASCD-NMI to some real network datasets used for the evaluation of disjoint community detection. Also, we adopted NMI as the metric to evaluate the results with respect to different parameter settings.

For ASCD-ARC, we varied the value of δ and λ from 1 to 100 with the step size of 10. As for ASCD-NMI, we used the same adjustment setting of λ as in the analysis of ASCD-ARC, but we set δ within the range of [0, 1] with step size of 0.1. The results of different datasets to which we had applied our ASCD method gave similar results. Here, we select the results of two of them (*Cornell* and *Washington*) as the illustration to draw the corresponding heat maps, which are shown in Figure 3. In each heat map, color close to red indicates a relative high NMI value while color close to blue represent low NMI level.

As shown in Figure 3, compared to λ, the final results of both ASCD-ARC and ASCD-NMI are more sensitive to δ, which can highlight the significance of the adaptive parameter in our ASCD model.



(a) **ASCD-ARC**, *Cornell* (b) **ASCD-ARC**, *Washington*

(c) **ASCD-NMI**, *Cornell* (d) **ASCD-NMI**, *Washington*

Fig. 3. The heat maps of the parameter adjustment analysis for both **ASCD-ARC** and **ASCD-NMI** with *Cornell* and *Washington* as the illustrated datasets and NMI as the metric. Different color corresponds to different NMI, and color close to red indicates high value. For both ASCD-ARC and ASCD-NMI, the results are more sensitive to δ rather than λ.

More importantly, the results shown above also indicate the fact that it is hard to find a fixed setting of δ and λ that can ensure our ASCD method to get the best result for any datasets, but we can still give a recommended setting of such two hyper-parameters. With regard to the ASCD-ARC, we suggest to set λ to be a value in the set of $\{1, 50, 100\}$ and set δ to be a value in the set of $\{0.1, 0.2, \cdots, 1.0\}$. As for ASCD-NMI, we also recommend to set λ as a value in $\{1, 50, 100\}$ and properly tune $\delta \in \{1, 10, \cdots, 100\}$.

Besides the setting of the hyper-parameters, we also need to consider the effect of the initialization. Since the solution strategy of the ASCD method is based on the gradient descend method, it's possible for the final result to converge to a local minima solution with a certain initialization setting of **X** and **Y**. For the sake of a relatively favorable result, we suggest to initialize **X** and **Y** (respectively according to (6) and (8)) at least 10 times, and use the initialization setting with minimum value of the objective function for the initialization process (with respect to (1) and (7)).

5.3 Artificial Network Analysis

Analyzing the Attribute Refining (AR) Effect (Based on the Artificial Networks). When the basic ASCD algorithm converges, an intuitive method to obtain the partition result is to directly utilize $\mathbf{X}$ to extract the community label of each node. In fact, we can also generate a new matrix $\mathbf{X}$ by using $\mathbf{X}=\mathbf{CY}$, since we try to minimize $\|\mathbf{X}-\mathbf{CY}\|_F^2$ when modeling the content information in the objective function (14).

Compared to the former, the latter way may incorporate more attributes to the final community partition, and we also found that when topology and content match well, we can obtain better community structures by using $\mathbf{X}=\mathbf{CY}$, so we name the latter way as *attribute refining* (AR). However, similar to the conclusion of our pre-experiment in Section 4.3, when the *mismatch* between topology and content is serious, AR may result in poor community partition result, as it introduces irrelevant content information to the final result. For our ASCD method, we aim to achieve better community discovery result by utilizing AR when topology *match* well with content, but if there is a serious *mismatch* between them, we tend to directly extract community structures from the original $\mathbf{X}$. In other words, we need to determine whether to utilize AR according to the *mismatch* degree.

Inspired by the introduction of the adaptive parameter (13) (with respect to ASCD-NMI), when the basic ASCD algorithm converges, the NMI value between the label sequences respectively extracted from the original $\mathbf{X}$ and $\mathbf{X}=\mathbf{CY}$ can be a suitable numerical feature, and we name such value as the AR-NMI. Therefore, only when the AR-NMI is large enough, which means the topology and content match well, we tend to use AR.

To further illustrate such effect, we compared the results with and without AR by applying both ASCD-ARC and ASCD-NMI to the artificial networks introduced above. For each value of ρ_{mis}, we recorded the corresponding AR-NMIs for both ASCD-ARC and ASCD-NMI, and we also adopted the NMI between the community discovery result and the ground-truth as the evaluation metric. With regard to the two hyper-parameters of two different forms of ASCD, we used the recommended setting with (δ=50, γ=1.0) for ASCD-ARC and (δ=0.5, γ=1.0) for ASCD-NMI. Finally, we respectively ran the algorithm of ASCD-ARC and ASCD-NMI on the artificial networks, and got the average result shown in Table 4, where we use NAR to represent the results of the methods without AR.

In Table 4, the results of ASCD-ARC and ASCD-NMI are similar. Methods with AR achieve much better performance than those without AR when $\rho_{mis}=0.0$. At this moment, the AR-NMIs of both ASCD-ARC and ASCD-NMI reaches a level of more than 0.6. However, as ρ_{mis} increases, the performance of methods with AR dramatically decrease, which is in line with our previous discussion. On the other hand, the performance of the methods without AR is relatively poor when $\rho_{mis}=0.0$ in comparison to those with AR, but the values of NMI keep at a relatively high level of 0.65 when $\rho_{\text{mis}}>0.0$. Hence, we tend to utilize AR when $\rho_{\text{mis}}=0.0$, as the

performance can be further improved with the help of AR, but we should reject to use such effect when $\rho_{\text{mis}} > 0.0$, because it may introduce irrelevant content information.

Table 4. The analysis of **AR** by applying **ASCD-ARC** and **ASCD-NMI** in the artificial networks with NMI as the metric. **AR** represents the methods with *attribute refining* and **NAR** represents those without **AR**. ρ_{mis} is the parameter to adjust the *mismatch degree* and **AR-NMI** is the numerical feature we need to track. As ρ_{mis} increases, the **AR-NMI**s for both **ASCD-ARC** and **ASCD-NMI** decrease, and NMI of methods with **AR** dramatically decrease while NMI of those without **AR** steadily keep at about 0.65. Only when ρ_{mis}=0.0, methods with **AR** have much better performance than those without **AR**.

ρ_{mis}	**ASCD-ARC**			**ASCD-NMI**		
	AR-NMI	**NAR**	**AR**	**AR-NMI**	**NAR**	**AR**
0.0	0.6487	0.6631	**0.8722**	0.6474	0.6622	**0.8729**
0.1	0.4754	**0.6575**	0.5880	0.4733	**0.6562**	0.5875
0.2	0.3723	**0.6593**	0.4261	0.3734	**0.6584**	0.4281
0.3	0.3159	**0.6556**	0.3338	0.3171	**0.6565**	0.3357
0.4	0.3032	**0.6576**	0.2963	0.3063	**0.6602**	0.2975
0.5	0.2918	**0.6524**	0.2651	0.2926	**0.6558**	0.2684
0.6	0.2802	**0.6609**	0.2593	0.2831	**0.6633**	0.2614
0.7	0.2734	**0.6556**	0.2447	0.2751	**0.6573**	0.2467
0.8	0.2630	**0.6504**	0.2338	0.2624	**0.6505**	0.2331
0.9	0.2614	**0.6565**	0.2343	0.2648	**0.6595**	0.2381
1.0	0.2655	**0.6540**	0.2340	0.2632	**0.6550**	0.2353

Moreover, for a criterion of determining whether to utilize the AR, a threshold of AR-NMI need to be further discussed. Namely, when current AR-NMI is larger than the threshold, one can utilize AR to extract better community structures, but when the AR-NMI is less than such value, one should reject the AR effect. For the result shown in Table 4, we can conclude that the threshold of the artificial network is about 0.5. Besides, we also studied the threshold of other real social network datasets with ground-truth introduced in Section 5.1 by comparing the NMI values corresponding to the results with and without AR. Finally, we found that the threshold of different datasets may be different. For example, the thresholds of *Cornell*, *Texas*, *Washington*, *Wisconsin*, *Cora* and *Citeseer* are about 0.1, but for *Facebook* and *Twitter* the corresponding values are about 0.6. In other word, the value of such threshold is related to concrete datasets, but it's independent of the *mismatch* degree. Therefore, to recommend a value of the threshold of AR-NMI is another challenging problem.

However, we can still reach the conclusion that such threshold is related to the topological clustering structures of a certain dataset. In fact, the parameter z_{out} we utilized to generate the artificial networks can control the quality of the clustering structures of topology, namely the larger the value of z_{out}, the more noise of topology in the network.

For a further study of such artificial network, we used the same fixed setting of h_{in} and h_{out} with the networks introduced above, while setting the value of z_{out} to be 4

and 12 (we have already had the results with respect to $z_{out} = 8$) to generate another two sorts of artificial networks. By using the same analysis method, we found that as the increase of z_{out} (from 4 to 12 with step size of 4), the quality of the topological clustering structures became poorer and the threshold of AR-NMI decreased, where the thresholds were respectively **0.9** ($z_{out} = 4$), **0.5** ($z_{out} = 8$) and **0.2** ($z_{out} = 12$).

On the other hand, we also considered the quality of the clustering structures of content, where we fixed the value of z_{out} to be 8 and respectively set h_{in} to be 16 as well as 8 (we have already had the results with respect to $h_{in} = 24$), but we found that the thresholds were all about 0.5. Therefore, the threshold of AR-NMI is related to the topological clustering structures (but independent to content according to our obersvation). For the convenience of understanding, we conclude different settings of z_{in} , z_{out} , h_{in} as well as h_{out} with the corresponding threshold values in Table 5.

Table 5. The analysis of AR-NMI's threshold by adjusting the quality of clustering structures of topology and content in the artificial networks. As the value of z_{out} gradually increase (with h_{in} (h_{out}) fixed), more noise is added into the clustering structure of topology, and the threshold gradually decrease. However, the thresholds keep as the same, even though increasingly noise is added into the content' clustering structure. Therefore, the threshold is related to the clustering structure of topology.

Parameter Settings	h_{in}=24, h_{out}=8		
	z_{in}=12, z_{out}=4	z_{in}=8, z_{out}=8	z_{in}=4, z_{out}=12
Threshold Value	0.9	0.5	0.2
Parameter Settings	z_{in}=8, z_{out}=8		
	h_{in}=24, h_{out}=8	h_{in}=16, h_{out}=16	h_{in}=8, h_{out}=24
Threshold Value	0.5	0.5	0.5

In a conclusion, for a certain network, to determine whether to utilize the AR effect, additional measure is needed to evaluate the quality of the dataset's clustering structures. However, when such additional evaluation is costly, we recommend to omit the process of determining AR-NMI's threshold and directly refuse to utilize the AR effect for the sake of a relatively robust result.

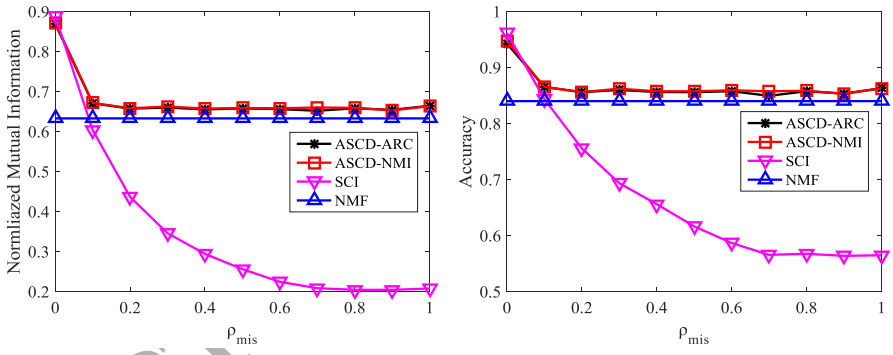
Robustness Evaluation (Based on the Artificial Networks).
Under an artificially controllable circumstance, we also evaluated our method's performance on the above artificial networks to generally illustrate its capacity to resist the *mismatch* effect.

In this experiment, we used the initialization result of **X** as the baseline (notated as the NMF method), since it's only related to topology. We also used the SCI method [5], which integrates topology and content, as the comparative method. Because we evenly divided the nodes in the network into 4 communities, we can directly get the ground-truth of each node's community label. Therefore, we adopted NMI and AC as the evaluation metrics. Especially, for the result of ASCD-ARC and ASCD-NMI, we determined whether to utilize AR according to the criterion proposed above. Finally,

for each setting of ρ_{mis}, we got the average result of the 50 artificial networks which is shown in Figure 4.

As shown in Figure 4, both metrics indicate similar results, and the metric curves of both ASCD-ARC and ASCD-NMI are almost the same. When $\rho_{\text{mis}} = 0$, as we utilized AR to extract the community structures, the performance of our ASCD method is competitive to the SCI method and much better than the baseline method. However, with the increase of ρ_{mis}, the performance of SCI seriously deteriorates and finally there is a big gap between the baseline level for large ρ_{mis} values. In other words, methods (such as SCI) that give the same weight to topology and content may perform even worse than those which only depend on topology. On the other hand, the performance of both ASCD-ARC and ASCD-NMI is at a level that outperforms the baseline, even when *mismatch* between topology and content is large.

In conclusion, by adding the adaptive parameter to control the trade-off between topology and content, our ASCD method robustly combines such two sources of information and maintains a high level of performance even when a significant *mismatch* between the topology and content exists. More importantly, when the two types of information *match* well in the network, one can further improve the community partition by utilizing AR, having a competitive performance to other methods that incorporate topology and content.



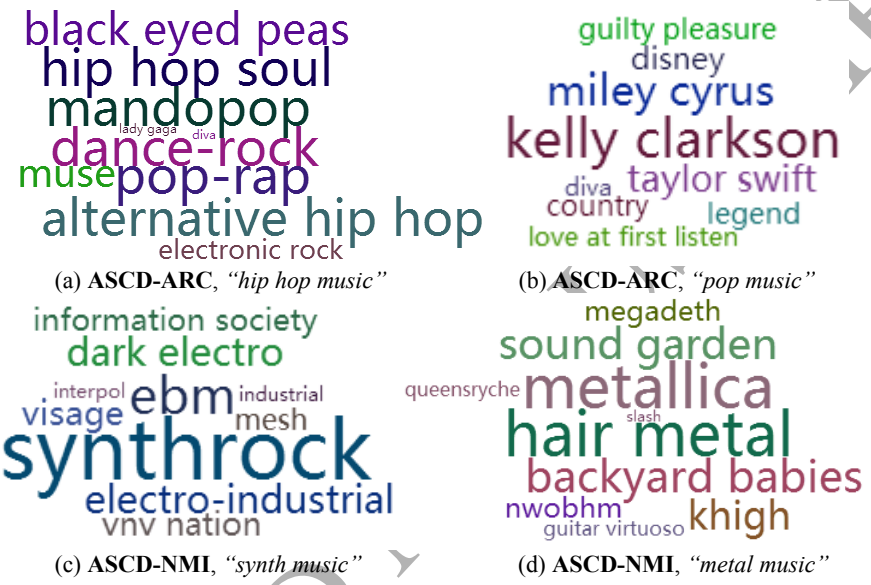
(a) **NMI** (b) **AC**

Fig. 4. Performance comparison of **NMF**, **SCI**, **ASCD-ARC** and **ASCD-NMI** for ρ_{mis} varies from 0 to 1, with (a) **NMI** and (b) **AC** as the metrics. **NMF** is the baseline, which depends on topology alone. **SCI** is the comparative method that integrates topology and content. **ASCD-ARC** and **ASCD-NMI** are our methods. ρ_{mis} is the parameter to control the degree of *mismatch*. Our methods steadily keep at a level that outperforms the baseline, but the comparative method seriously deteriorates as ρ_{mis} increases, which means both **ASCD-ARC** and **ASCD-NMI** can achieve a robust performance even when the *mismatch* is serious.

5.4 The Case Study for Semantic Description

We specially conducted a case study to illustrate our method's powerful capacity to derive corresponding semantic descriptions as soon as the partition of communities is done, in which we used *last.fm* [9] as the testing dataset.

For the hyper-parameters of ASCD-ARC, we used the recommended setting of δ=1.0 and λ=1.0. With regard to ASCD-NMI, we set δ=1.0 and λ=1.0. We applied such two methods to the *last.fm* dataset and selected the top 10 key words for each community to draw the corresponding word clouds. Finally, we display 4 examples in Figure 5, with the first two corresponding to ASCD-ARC and the last two corresponding to ASCD-NMI. So as to analyze the semantic relation among the top words in a word cloud, we also used them as the query keywords of Wikipedia (https://en.wikipedia.org) to further looked up related materials.



(a) **ASCD-ARC**, *"hip hop music"* (b) **ASCD-ARC**, *"pop music"*

(c) **ASCD-NMI**, *"synth music"* (d) **ASCD-NMI**, *"metal music"*

Fig. 5. Four examples of semantic description of **ASCD-ARC** (**(a)** and **(b)**) and **ASCD-NMI** (**(c)** and **(d)**). The four communities are respectively with the topic of **(a)** *"hip hop music"*, **(b)** *"pop music"*, **(c)** *"synth music"* and **(d)** *"metal music"*.

For the community in Figure 5 (a), the main topic is *hip hop music*, as the keywords "hip hop soul" and "alternative hip hop" are directly related to it. Moreover, "black eyed peas" is an American hip hop music group. According to the materials posted in Wikipedia, "mandopop" (which is short for mandarin popular music) and "pop-rap" are both musical genres originated from hip hop.

Community shown in Figure 5 (b) may be related to *pop music* and *female pop singers*, because "kelly clarkson", "miley cyrus" and "taylor swift" are all famous American female rock pop and country pop singers. And "guilty pleasure" may refer to the album of Ashley Tisdale, who is also known as a female rock pop singer.

In the community of Figure 5 (c), *synth music* may be a distinct topic, just as "synthrock" is the most significant keyword. Originated from synth-pop, "electro-industrial" is a music genre drawing on "ebm", and "dark electro" is a derivative form of such genre. Furthermore, "information society", "vnv nation" and "visage" are all typical synthpop bands.

With regard to the community of Figure 5 (d), the topic is *metal music*, which is mainly indicated by "hair metal" and "metallica". In addition, "sound garden",

"megadeth" and "queensryche" are all successful American band with different subgenre of metal music. And "nwobhm" may be the abbreviation of new wave of British heavy mental with respect to the query result of Wikipedia.

In conclusion, the four communities shown in Figure 5 have distinct semantic descriptions with respect to a certain topic. By selecting the top keywords of a community, one can easily understand a community's semantic. Therefore, our ASCD method has the powerful capacity to obtain corresponding semantic description as soon as the partition of communities is finished.

5.5 Real Social Network Evaluation

Evaluation of Disjoint Community Detection.
In the evaluation experiment of disjoint community detection, we applied ASCD-ARC and ASCD-NMI to 8 real social network datasets with disjoint ground-truth, including *Cornell*, *Texas*, *Washington*, *Wisconsin*, *Cora*, *Citeseer*, *Twitter* and *Facebook* [10,11].

For comparison, we utilized the initialization setting of **X** (notated as NMF) and other 4 state-of-the-art methods, including DCSBM [19], BLOCK-LDA [25], PC-LDC [26] and SCI [5], as the baseline. In the 5 comparative methods, NMF and DCSBM are those only depend on structural information of the network, while BLOCK-LDA, PC-LDC as well as SCI are all existing methods that integrate topology and content. Especially, DCSBM, BLOCK-LDA and PC-LDC are probability-based, where DCSBM and BLOCK-LDA are two generative model, while PC-LDC is a discriminant model. Moreover, SCI is an NMF-based hybrid method.

We adopted the default parameter settings of DCSBM, BLOCK-LDA and PC-LDC. For SCI, there was a parameter adjustment process to get the best result. As for both ASCD-ARC and ASCD-NMI, we adjusted the hyper-parameters (δ and λ), and finally adopted the parameter setting with minimum value of the objective function (14). Moreover, we also selectively utilized AR to achieve better performance when the AR-NMI was large enough.

For each dataset, we set the number of communities K according to the ground-truth and used NMI as well as AC as the metrics. The evaluation results of disjoint community detection are shown in Table 6, where the best metric values are in **bold** and the second-best are **underlined**.

As shown in Table 6, for the metric of **AC**, **ASCD-ARC** and **ASCD-NMI** are respectively performs the best and second best on 5 of the 8 datasets (*Cornell*, *Washington*, *Wisconsin*, *Cora* and *Citeseer*). For the rest 3 datasets, **ASCD-NMI** also performs the best on *Texas* and second-best on *Facebook*, while **ASCD-ARC** performs the second-best on *Twitter*.

Moreover, for the metric of **NMI**, **ASCD-ARC** has the best performance on 5 datasets (*Cornell*, *Cora*, *Citeseer*, *Twitter* and *Facebook*) and has the second-best performance on the rest 3 datasets. For the datasets on which ASCD-ARC performs the second-best, **ASCD-NMI** has the best performance, while on the datasets where ASCD-ARC performs the best **ASCD-NMI** has the second-best performance. Namely, our method performs best on the 8 datasets compared to the other methods.

For the sake of a better measure of the improvement degree, we calculated the percentage difference for the performance of both **ASCD-ARC** and **ASCD-NMI** (for **AC** and **NMI**) compared to the next best method. The result is shown in Table 7, where the maximum performance improvement for each method is in **bold**.

Table 6. The evaluation result of disjoint community detection on 8 real network datasets with **AC** and **NMI** as the metrics. For each dataset, the best performance value is in bold and the second-best is underlined. In majority of cases, our ASCD-ARC and ASCD-NMI performs the best or second-best.

Metrics	Methods	Datasets							
		Cornell	**Texas**	**Washington**	**Wisconsin**	**Cora**	**Citeseer**	**Twitter**	**Facebook**
AC	*NMF*	0.3692	0.4866	0.4304	0.3925	0.4453	0.2622	0.4621	0.3559
	DCSBM	0.3795	0.4809	0.3180	0.0314	0.3848	0.2657	**0.6049**	0.4519
	BLOCK-LDA	0.4615	0.5410	0.3917	0.4962	0.2552	0.2435	0.3580	0.3766
	PCL-DC	0.3026	0.3880	0.2995	0.3015	0.3408	0.2485	0.5679	0.4038
	SCI	0.4564	0.6230	0.5115	0.5038	0.4062	0.2798	0.5062	**0.5104**
	ASCD-ARC	**0.5128**	0.6096	0.5261	0.5358	0.4826	**0.3884**	0.5758	0.4391
	ASCD-NMI	0.4872	**0.6310**	**0.5348**	**0.5396**	**0.5041**	0.3154	0.5682	0.4745
NMI	*NMF*	0.0759	0.0868	0.0545	0.0670	0.2665	0.0613	0.6355	0.5529
	DCSBM	0.0969	0.1665	0.0987	0.0314	0.1707	0.0413	0.5748	0.4338
	BLOCK-LDA	0.0681	0.0421	0.0369	0.1009	0.0141	0.0242	0.0000	0.0928
	PCL-DC	0.0723	0.1037	0.0566	0.0501	0.1754	0.0299	0.5265	0.3863
	SCI	0.1144	0.1784	0.1237	0.1704	0.1926	0.0488	0.4300	0.3001
	ASCD-ARC	**0.1840**	0.2025	0.1814	0.1989	**0.3337**	**0.0805**	**0.6689**	**0.5841**
	ASCD-NMI	0.1643	**0.2264**	**0.1838**	**0.2078**	0.3055	0.0690	0.6666	0.5829

Table 7. The percentage difference for the performance of ASCD-ARC and ASCD-NMI compared to the next best method in Table 6. For each method, the best performance improvement is in bold. Our method achieves a maximum performance improvement of **38.83%** for **AC** and **60.84%** for **NMI**.

Metrics	Methods	Datasets							
		Cornell	**Texas**	**Washington**	**Wisconsin**	**Cora**	**Citeseer**	**Twitter**	**Fackbook**
AC	**ASCD-ARC**	11.12%	-	3.85%	6.35%	8.38%	**38.81%**	-	-
	ASCD-NMI	5.57%	1.28%	4.46%	7.1%	**13.21%**	12.72%	-	-
NMI	**ASCD-ARC**	**60.84%**	13.51%	46.65%	16.73%	25.22%	31.32%	5.26%	3.12%
	ASCD-NMI	43.62%	26.91%	**48.59%**	21.95%	14.63%	12.56%	4.89%	3.00%

As shown in Table 7, the **ASCD-ARC** method has the best performance improvement (compared to the next best method excluding ASCD-NMI) of **38.81%** (*Citeseer*) for the metric of **AC** and **60.84%** (*Cornell*) for **NMI**. On the other hand, the corresponding best performance improvements of **ASCD-NMI** are **13.21%** (*Cora*) for **AC** and **48.59%** (*Washington*) for **NMI**. Hence, our method has a maximum performance improvement of about **38.83%** for **AC** and **60.84%** for **NMI**.

In conclusion, in the evaluation of disjoint community detection, the ASCD method has a better performance than other state-of-art methods.

Evaluation of Overlapping Community Detection.

In the evaluation experiment of overlapping community detection, we utilized 4 datasets with overlapping ground-truth as the testing datasets, which are *Enron*, *Reddit25*, *Reddit26* and *Reddit27* [12,13].

As for comparative method, we used the NMF method introduced in the evaluation of disjoint community discovery and other three state-of-the-art overlapping community detection methods, including BigCLAM [21], CESNA [32] and Circles [33]. Within these methods, NMF and BigCLAM use merely the topological strucutre of the network, while CESNA as well as Circles are methods incorporating topology and content. Especially, CESNA and Circles are two typical generative probabilistic methods, but BigCLAM is an NMF-based method.

With regard to ASCD-ARC and ASCD-NMI (also for the NMF method), we applied the extended algorithm to the original **X** with best disjoint community partition result to extract the corresponding overlapping community structures.

Moreover, we used the generalized F-score and Jaccard [32] as the metrics, and the corresponding result is shown in Table 8, where for each dataset the best metric value is in **bold** and the second-best is **underlined**.

In Table 8, both metrics give similar results. For all the testing datasets, ASCD-ARC performs the second-best while ASCD-NMI has the best performance.

Furthermore, we also calculated the percentage difference for the performance of ASCD-ARC and ASCD-NMI compared to the next best method in Table 8. The corresponding results are shown in Table 9, where the best performance improvement for each method is in **bold**.

Table 8. The evaluation of overlapping community detection on 4 real network datasets with generalized **F-Score** and **Jaccard** as the metrics. For each dataset, the best performance value is in bold and the second-best is underlined. Our ASCD-ARC and ASCD-NMI methods performs the best or second-best on all the datasets.

Metrics	Methods	Datasets			
		Enron	**Reddit25**	**Reddit26**	**Reddit27**
F-Socre	*NMF*	0.4764	0.5859	0.5549	0.5865
	BigCLAM	0.1890	0.2036	0.2429	0.1781
	CESNA	0.3015	0.3488	0.3396	0.2781
	Circles	0.4522	0.5023	0.5108	0.5255
	ASCD-ARC	<u>0.5307</u>	<u>0.6181</u>	<u>0.5836</u>	<u>0.6218</u>
	ASCD-NMI	**0.5382**	**0.6339**	**0.5866**	**0.6810**
Jaccard	*NMF*	0.3255	0.4371	0.4249	0.4406
	BigCLAM	0.1092	0.1263	0.1633	0.1058
	CESNA	0.2021	0.2593	0.2153	0.1715
	Circles	0.3211	0.3609	0.3728	0.3820
	ASCD-ARC	<u>0.3727</u>	<u>0.4829</u>	<u>0.4497</u>	<u>0.4801</u>
	ASCD-NMI	**0.3793**	**0.4912**	**0.4507**	**0.5487**

Table 9. The percentage difference for the performance of **ASCD-ARC** and **ASCD-NMI** compared to the next best method in Table 8. For each method, the best performance improvement is in bold. Our method achieves a maximum performance improvement of **16.11%** for generalized **F-Score** and **16.53%** for generalized **Jaccard**.

Metrics	Methods	Datasets			
		Enron	**Reddit25**	**Reddit26**	**Reddit27**
F-Socre	**ASCD-ARC**	**11.40%**	5.50%	5.17%	6.02%
	ASCD-NMI	12.97%	8.19%	5.71%	**16.11%**
Jaccard	**ASCD-ARC**	**14.50%**	10.48%	5.84%	3.95%
	ASCD-NMI	**16.53%**	12.38%	6.07%	10.81%

In Table 9, **ASCD-ARC** has the best performance improvement of **11.40%** (*Enron*) for generalized **F-Score** and **14.50%** (*Enron*) for generalized **Jaccard** compared to the next best method (excluding the ASCD-NMI). With regard to ASCD-NMI, the corresponding best performance improvements are respectively **16.11%** (*Reddit27*) for generalized **F-Score** and **16.53%** (*Enron*) for generalized **Jaccard**. Therefore, in the evaluation of overlapping community detection, our method obtains a maximum performance improvement of **16.11%** for generalized **F-Score** and **16.53%** for generalized **Jaccard**.

In summary, by applying the extended algorithm to the original disjoint community partition result, our method, in majority of cases, have better performance than other comparative methods.

As a conclusion for the evaluation of both disjoint and overlapping community detection, because of the integration of topological and content information, the ASCD method has resulted in better performance than methods using only topological information. More importantly, by adding the adaptive parameter (with two different forms) to resist the *mismatch* effect, our new method performs better than most of the state-of-the-art methods incorporating topology and content.

6 Conclusion

In this paper, we proposed a novel ASCD model under the framework of NMF while introducing two different forms of the adaptive parameter to be applied in such model. Moreover, as we also introduced an extended algorithm to extract overlapping community structures, the proposed method can meet the application requirements of both disjoint and overlapping community detection. Our approach integrates the network's topology and content while taking the *mismatch* between them into account. In comparison to conventional methods that merely consider topology, our method is capable of improving the community detection accuracy as it further integrates additional information of content. Furthermore, when comparing to other state-of-the-art methods combining topology and content, our method also achieves better performance, because we introduced a novel adaptive parameter with two different forms to effectively control the contribution of content information. More importantly, the new proposed method also has the powerful ability to simultaneously obtain the community partitions and corresponding semantic description.

However, for real social networks, content may have various forms, where node attributes we considered in this paper is just one typical form. In fact, content related to edges is common in social networks. For example, in an e-mail system (like the *Enron* dataset [12] introduced in Section 5.1), the content of an email is related to the edge connecting from the sender to the receiver. Moreover, edge-based community partition can also help discover overlapping communities [30]. Therefore, we intend to further consider the comprehensive effect of incorporating both node-induced and edge-induced attributes in the process of community detection in our future work.

7 Acknowledgments

This work has been financially supported by the National Natural Science Foundation of China (61502334, 61303110) and the Shenzhen Key Fundamental Research Projects (JCYJ20151030154330711).

References

[1] M. Girvan, M. E. Newman, Community structure in social and biological networks, Proceedings of the national academy of sciences 99(12) (2002) 7821-7826.

[2] A. Alamsyah, B. Rahardjo, Kuspriyanto, Community detection methods in social network analysis, Journal of Computational & Theoretical Nanoscience, 20(1) (2014) 250-253.

[3] Y. Pei, N. Chakraborty, K. Sycara, Nonnegative matrix tri-factorization with graph regularization for community detection in social networks, In Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015, June.

[4] D. He, Z. Feng, D. Jin, X. Wang, W. Zhang, Joint Identification of Network Communities and Semantics via Integrative Modeling of Network Topologies and Node Contents, In Thirty-First AAAI Conference on Artificial Intelligence, 2017, February.

[5] X. Wang, D. Jin, X. Cao, L. Yang, W. Zhang, Semantic community identification in large attribute networks, Thirtieth AAAI Conference on Artificial Intelligence , AAAI Press, 2016, pp. 265-271.

[6] D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature, 401(6755) (1999) 788-791.

[7] H. Liu, Z. Wu, X. Li, D. Cai, T. S. Huang, Constrained nonnegative matrix factorization for image representation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(7) (2012) 1299-1311.

[8] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature, 435(7043) (2005) 814.

[9] I. Cantador,P. Brusilovsky, T. Kuflik, Second workshop on information heterogeneity and fusion in recommender systems (HetRec2011), ACM Conference on Recommender Systems, Recsys 2011, 2011, October, Vol.92, pp. 387-388.

[10] J. Leskovec, R. Sosič, Snap: A general-purpose network analysis and graph-mining library, ACM Transactions on Intelligent Systems and Technology (TIST) 8(1) (2016) 1.

[11] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, T. Eliassi-Rad, Collective classification in network data, AI magazine 29(3) (2008) 93.

[12] Z. Zhao, S. Feng, Q. Wang, J. Z. Huang, G. J. Williams, J. Fan, Topic oriented community detection through social objects and link analysis in social networks, Knowledge-Based Systems 26 (2012) 164-173.
[13] C. D. Wang, J. H. Lai, S. Y. Philip, NEIWalk: Community discovery in dynamic content-based networks, IEEE transactions on knowledge and data engineering 26(7) (2014) 1734-1748.
[14] H. Fani, E. Bagheri, Community detection in social networks, Encyclopedia with Semantic Computing & Robotic Intelligence 01(1) (2017) 1630001.
[15] I. Falih, N. Grozavu, R. Kanawati, Y. Bennani, Community detection in Attributed Network, Companion of the the Web Conference, 2018, pp. 1299-1306.
[16] M. E. Newman, Fast algorithm for detecting community structure in networks, Physical review E 69(6) (2004) 066133.
[17] V. D. Blondel, J. L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, Journal of statistical mechanics: theory and experiment, 2008(10) (2008) P10008.
[18] S. White, P.Smyth, A spectral clustering approach to finding communities in graphs, In Proceedings of the 2005 SIAM international conference on data mining, Society for Industrial and Applied Mathematics, 2005, April, pp. 274-285.
[19] B. Karrer, M. E. Newman, Stochastic blockmodels and community structure in networks, Physical Review E 83(1) (2011) 016107.
[20] T. Martin, B. Ball, M. E. Newman, Structural inference for uncertain networks, Physical Review E 93(1) (2016) 012306.
[21] J. Yang, 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, ACM, 2013, February, pp. 587-596.
[22] F. Wang, T. Li, X. Wang, S. Zhu, C. Ding, Community discovery using nonnegative matrix factorization, Data Mining and Knowledge Discovery 22(3) (2011) 493-521.
[23] Y. Ruan, D. Fuhry, S. Parthasarathy, Efficient community detection in large networks using content and links, In Proceedings of the 22nd international conference on World Wide Web, ACM. ,2013, May, pp. 1089-1098.
[24] S. Pool, F. Bonchi, M. V. Leeuwen, Description-driven community detection, ACM Transactions on Intelligent Systems and Technology (TIST) 5(2) (2014) 28.
[25] R. Balasubramanyan, W. W. Cohen, Block-LDA: Jointly modeling entity-annotated text and entity-entity links, In Proceedings of the 2011 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2011, April, pp. 450-461.
[26] T. Yang, R. Jin, Y. Chi, 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, ACM, 2009, June, pp. 927-936.
[27] Z. Xu, Y. Ke, Y. Wang, H. Cheng, J. Cheng, A model-based approach to attributed graph clustering, ACM SIGMOD International Conference on Management of Data, ACM, 2012, pp. 505-516.
[28] J. Kim, H. Park, Sparse nonnegative matrix factorization for clustering, Georgia Institute of Technology, 2008.
[29] D. Jin, B. Gabrys, J. Dang, Combined node and link partitions method for finding overlapping communities in complex networks, Scientific reports 5 (2015).
[30] G. J. Qi, C. C. Aggarwal, T. Huang, Community detection with edge content in social media networks, International conference on data engineering, international conference on data engineering, 2012, 41(4), pp. 534-545.
[31] H. Zhu, D. Liu, S. Zhang, Y. Zhu, L. Teng, S. Teng, Solving the many to many assignment problem by improving the kuhn–munkres algorithm with backtracking, Theoretical Computer Science 618(C) (2016) 30-41.

[32] J. Yang, J. Mcauley, J. Leskovec, Community Detection in Networks with Node Attributes, International Conference on Data Mining, IEEE, 2013, pp. 1151-1156.
[33] J. Mcauley, J. Leskovec, Discovering social circles in ego networks, ACM Transactions on Knowledge Discovery from Data (TKDD) 8(1) (2014) 4.
[34] D. D. Lee, H. S. Seung, Algorithms for Non-negative Matrix Factorization, neural information processing systems, 2001, pp. 556-562.
[35] S. Fortunato, D. Hric, Community detection in networks: a user guide, Physics Reports 659 (2016) 1-44.
[36] M. Qin, D. Jin, D. He, B. Gabrys, K. Musial, Adaptive Community Detection Incorporating Topology and Content in Social Networks, IEEE/acm International Conference, 2017, pp. 675-682.