World Wide Web (2017) 20:1409–1441
DOI 10.1007/s11280-017-0441-5

Community detection for emerging social networks

Qianyi Zhan[1] · **Jiawei Zhang**[2] · **Philip Yu**[2,3] · **Junyuan Xie**[1]

Received: 21 June 2016 / Revised: 21 December 2016 /
Accepted: 22 January 2017 / Published online: 8 February 2017

Abstract Many famous online social networks, e.g., Facebook and Twitter, have achieved great success in the last several years. Users in these online social networks can establish various connections via both social links and shared attribute information. Discovering groups of users who are strongly connected internally is defined as the community detection problem. Community detection problem is very important for online social networks and has extensive applications in various social services. Meanwhile, besides these popular social networks, a large number of new social networks offering specific services also spring up in recent years. Community detection can be even more important for new networks as high quality community detection results enable new networks to provide better services, which can help attract more users effectively. In this paper, we will study the community detection problem for new networks, which is formally defined as the "New Network Community Detection" problem. New network community detection problem is very challenging to solve for the reason that information in new networks can be too sparse to calculate effective similarity scores among users, which is crucial in community detection. However, we notice that, nowadays, users usually join multiple social networks simultaneously and those

✉ Qianyi Zhan
zhanqianyi@gmail.com

Jiawei Zhang
jzhan9@uic.edu

Philip Yu
psyu@cs.uic.edu

Junyuan Xie
jyxie@nju.edu.cn

[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China

[2] University of Illinois at Chicago, Chicago, IL 60607, USA

[3] Institute for Data Science, Tsinghua University, Beijing, China

who are involved in a new network may have been using other well-developed social networks for a long time. With full considerations of network difference issues, we propose to propagate useful information from other well-established networks to the new network with efficient information propagation models to overcome the shortage of information problem. An effective and efficient method, CAT (Cold stArT community detector), is proposed in this paper to detect communities for new networks using information from multiple heterogeneous social networks simultaneously. Extensive experiments conducted on real-world heterogeneous online social networks demonstrate that CAT can address the new network community detection problem effectively.

Keywords Community detection · Cold start problem · Transfer learning · Data mining

1 Introduction

Clusters in a network are defined as groups of nodes which are strongly connected in the group but loosely connected to nodes in other groups. Depending on specific disciplines, networks studied in clustering problems can be very diverse, which include online social networks, e.g., Twitter and Facebook [40]; biological networks, e.g., between-species interaction networks [37] and protein-protein interaction networks [16]; bibliographic networks, e.g., DBLP [35]; e-commerce networks, e.g., Amazon and Epinions [15]. Discovering clusters of user nodes in social networks can be formally defined as the *community detection* problem [16, 21, 35, 37, 40].

Community detection is a very important problem for online social networks as it is a crucial prerequisite for many concrete social services: (1) better organization of users' friends in online social networks, e.g., Facebook and Twitter, which can be achieved by applying community detection techniques to partition users' friends into different clusters, e.g., schoolmates, family, celebrities, etc.; (2) better group-level recommender systems for users in e-commerce social sites, e.g., Amazon and Epinions, which can be reached by grouping users with similar purchase interests into the same cluster; (3) better identification of influential users [38] in online social networks, which can be attained by selecting the most influential users in each community and these influential users can usually act as the seed users in viral marketing [31].

Meanwhile, witnessing the incredible success of popular online social networks, e.g., Facebook and Twitter, a large number of new social networks offering specific services also spring up overnight to compete for the market share. Generally, new networks are those containing very sparse information and can be (1) the social networks which are newly constructed and start to provide social services for a short period of time; or (2) even more mature ones that start to branch into new geographic areas or social groups [47]. The formal mathematical definition of "*new networks*" and "*developed networks*" is available in Section 3. These new networks can be of a wide variety, which include (1) location-based social networks, e.g., Foursquare and Jiepang; (2) photo organizing and sharing sites, e.g., Pinterest and Instagram; (3) educational social sites, e.g., Stage 32.

Considering its wide applications in various concrete social services, community detection can be more important for new networks because high-quality community detection results enable new networks to provide better services, which can help attract user registration effectively. However community detection in new networks is a novel problem and conventional community detection methods for well-developed networks cannot be

applied directly. In the new network, there is relatively little information about each user, which results in an inability to classify users into communities. Therefore compared with well-developed networks, information in new networks are too sparse to support traditional community detection methods to calculate effective closeness scores and achieve good results. Meanwhile, as proposed in [12, 46, 47, 52], users nowadays usually participate in multiple social networks simultaneously to enjoy more social services. Users who are involved in a new network may have been using other well-developed social networks for a long time, in which they can have plenty of information.

In this paper, we will detect social communities for *new networks* with information propagated across multiple *partially aligned social networks*, which is formally defined as the "*new network community detection*" problem. Especially, when the network is brand new, the problem will be the "*cold start community detection*" problem. *Cold start problem* is most prevalent in *recommender systems* [46], where the system cannot draw any inferences for users or items about which it has not yet gathered sufficient information, but few works have been done on studying the *cold start problem* in clustering/community detection problems. We are the first to propose the concepts of "*new network community detection*" problem and "*cold start community detection*" problem. Meanwhile, we are also the first to study community detection problem across partially aligned heterogeneous networks. The "*new network community detection*" problem and "*cold start community detection*" problem studied in this paper are both novel problems and totally different from other existing works on community detection. A detailed comparison of the "*new network community detection*" problem with some related problems is available in Table 1.

Despite its importance and novelty, the "*new network community detection*" studied in this paper is also very challenging to solve due to the following reasons:

- *network heterogeneity problem*: Proper definition of closeness measure among users with link and attribute information in the heterogeneous social networks is very important for community detection problems.
- *shortage of information*: Community detection for new networks can suffer from the shortage of information problem, similar to the "*cold start problem*" in [46, 47].
- *network difference problem*: Different networks can have different properties. Some information propagated from other well-developed networks can be useful for solving the *new network community detection* problem but some can be misleading on the other hand.
- *high memory space cost*: Community detection across multiple aligned networks can involve too many nodes and connections, which will lead to high space cost.

To solve all the above challenges, a novel community detection method, CAT, is proposed in this paper: (1) CAT introduces a new concept, *intimacy*, to measure the closeness relationships among users with both link and attribute information in online social networks; (2) CAT can propagate useful information from aligned well-developed networks to the new network to solve the shortage of information problem; (3) CAT addresses the network heterogeneity and difference problems with both *micro-level* and *macro-level* control of the link and attribute information proportions, whose parameters can be adjusted by CAT automatically; (4) effective and efficient cross-network information propagation models are proposed in this paper to solve the high space cost problem.

This paper is organized as follows. We first analyze the dataset in Section 2 and formulate the problem in Section 3. Detailed description of the methods is introduced in Sections 4–6.

Table 1 Summary of related problems

Property	New Network Community Detection	Social Influence based Clustering [54]	Clustering with Complete Links & Attributes [53]	Clustering with Incomplete Attributes [35]	Clustering with Incomplete Links [17]
# networks	multiple original networks	multiple decomposed networks	single	single	single
network type	heterogeneous	homogeneous or bipartite	heterogeneous	heterogeneous	heterogeneous
connections across networks	anchor links	connections across decomposed subnetworks	n/a	n/a	n/a
network aligned?	partially aligned	n/a	n/a	n/a	n/a
incomplete?	yes	no	no	yes	yes
incomplete information	both links and attributes	n/a	n/a	attributes only	links only
has cold start problem?	yes	no	no	no	no

In Section 7, we show the experiment results. The related works are given in Section 8. Finally, we conclude the paper in Section 9.

2 Observation

In this section, we analyze the data from both *developed networks*, for example, Twitter and *new networks*, such as Foursquare, and get the following observations related to the community detection.

1. The network structure of the new social network is sparse.

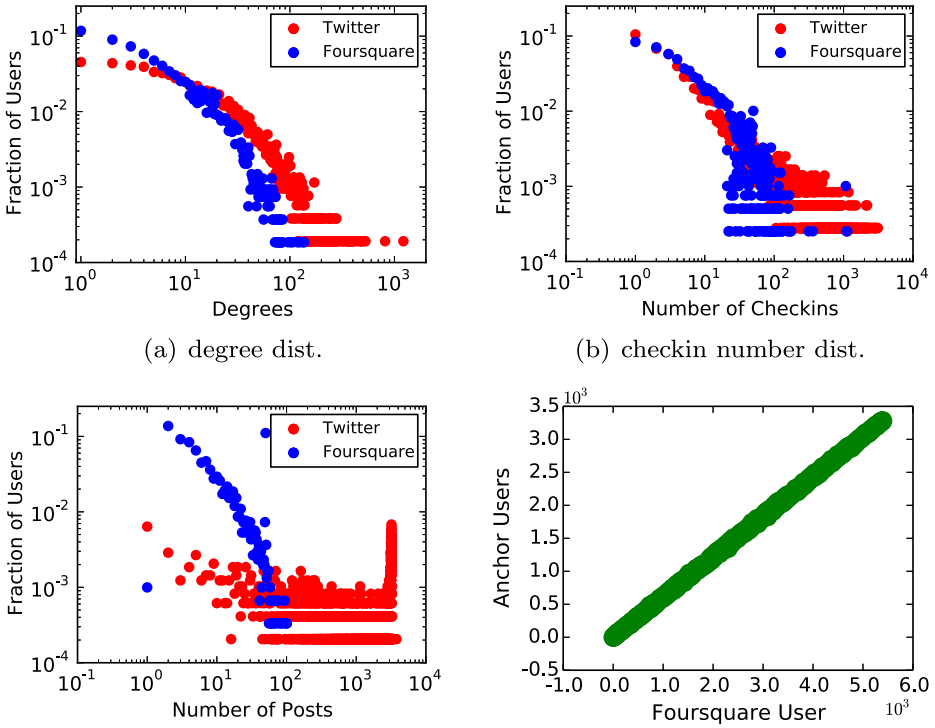
According to the market report from DRM[1], by the end of 2013, the total number of registered users in Foursquare has reached 45 million but these Foursquare users have only post 40 million tips. In other words, each user has posted less than one tip in Foursquare on average. Meanwhile, the 1 billion registered Twitter users have published more than 300 billion tweets by the end of 2013 and each Twitter user has written more than 300 tweets. We also provide a statistics investigation on the datasets, including both Foursquare and Twitter, used in this paper, whose basic information is listed in Table 2. The most straightforward way to show the sparsity of a graph is the *graph density*, which is defined as $D = \frac{|\mathcal{E}|}{|\mathcal{V}|\times(|\mathcal{V}|-1)}$ in a directed graph, we first check this criteria of two networks. Twitter's graph density is 0.6047 %, while this score of Foursquare is 0.2648%. Since the distinction between sparse and dense graphs is rather vague, and depends on the context, in this situation, we can say

[1] http://expandedramblings.com

Table 2 Properties of the Heterogeneous Social Networks

		Network	
	Property	Twitter	Foursquare
# node	user	5,223	5,392
	tweet/tip	9,490,707	48,756
	location	297,182	38,921
# link	friend/follow	164,920	76,972
	write	9,490,707	48,756
	locate	615,515	48,756

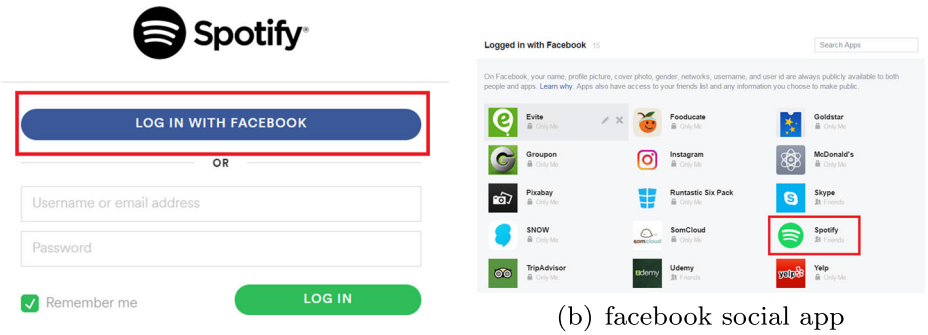
Foursquare is much more sparse than Twitter. We further study the information distribution of two networks, and results are given in Figure 1. As shown in Figure 1a–c, users in Twitter have far more social connections, posts and location check-ins than users in Foursquare. The figures illustrate the shortage of information encountered in community detection problems for new networks can be a serious obstacle for traditional community detection methods to achieve good performance and is urgent to solve.



(a) degree dist. (b) checkin number dist.

(c) post number dist. (d) anchor user dist.

Figure 1 Information and anchor user distributions in Foursquare and Twitter. **a**: social degree distribution, (**b**): number of check-ins distribution, (**c**): number of posts distribution, (**d**): number of anchor users in a random sample of Foursquare users



(a) spotify social login (b) facebook social app

Figure 2 A Social Login Example

2. Different networks are connected by anchor links.

To describe the structure of networks sharing common users, we define the shared users across different networks as the *anchor users*, while the remaining unshared users are formally defined as the *non-anchor users*. Links between accounts of anchor users in partially aligned networks are defined as the *anchor links* and networks partially aligned by the anchor users are named as the *partially aligned networks*. Anchor users are abound in real-world social networks. Most emerging social networks provide users with the option to log in the network with their existing Facebook or Twitter accounts, via which these emerging networks can get aligned with Facebook and Twitter extensively. For example, when users log in Spotify[2], a music, podcast, and video streaming service, they can choose using their Facebook accounts to log in, shown as Figure 2a. On the other hand, when users connect their Facebook accounts with Spotify, Spotify icon will appear on the users' app page, which means their profiles are publicly available to Spotify, shown as Figure 2b. This widely applied technique is called *social login*, which creates anchor links naturally and demonstrate a large amount of networks are connected by anchor links in the real world. We further study the effect of anchor links on the datasets used in this paper. As shown in Figure 1d, we randomly sampled a proportion of users from Foursquare, in which the number of users who are also involved in Twitter accounts for about 70 %. Most online social networks can provide the APIs (Application Programming Interfaces) to allow external applications to retrieve users' information in them. For a new network, if there are other matured networks which share some common user behavior, we may use the knowledge accumulated in the matured network to help mine the new network. For example, Foursquare and Twitter share some common user behavior as people participated in Foursquare often use Twitter to make comments. In fact, our experimental results will validate this is indeed the case and can prove what we claim.

3. Information in different networks have distinct properties.

Though different networks are connected by anchor links, information in them have distinct properties, which need to be considered separately when propagating information across different social networks. To demonstrate such claim, we also analyze the link (social link) and attribute (visited location) information in the aligned network datasets used in

[2]http://www.spotify.com

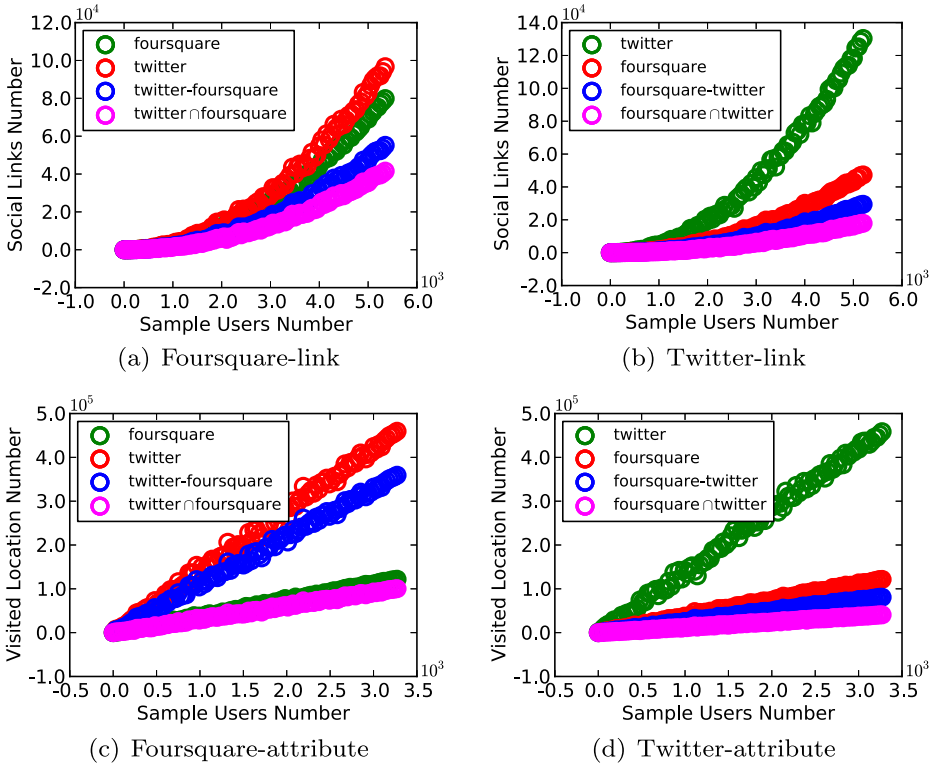


Figure 3 Link and attribute information in aligned networks, where (**a**) and (**c**) show the information of a given Foursquare sample user set; (**b**) and (**d**) show the information of a given Twitter sample user set

this paper, whose results are given in Figure 3. In Figure 3a, for a given random sample of Foursquare users, we extract all the social links among them in Foursquare and Twitter, the unique links in Twitter only, denoted by "twitter - foursquare", and common links existing in both networks, denoted by "twitter ∩ foursquare". As illustrated in Figure 3a, Twitter does share common links with Foursquare but also contains lots of unique links that don't exist in Foursquare. Similar results can be obtained in Figure 3b–d. As a result, propagating useful information but discarding misleading one, which includes both link and attribute information, from other well-developed networks to the new network can be what we desire.

3 Problem formulation

Before introducing the methods, we will give the definitions of many important concepts and the formulation of the new network community detection problem first in this section.

3.1 Terminology definition

Definition 1 Attribute Augmented Heterogeneous Networks: Users in networks can have both link and attribute information and the network studied in this paper are formulated as attribute augmented heterogeneous networks $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$, where $\mathcal{V}$ and $\mathcal{E}$ are

the user set and directed social link set of G respectively. $\mathcal{A} = \{a_1, a_2, \cdots, a_m\}$ is the set of m different kinds of attribute information that users have in G and the i_{th} attribute $a_i \in \mathcal{A}$ can have n_i different values in the network.

Users in social networks, e.g., G, can be correlated with each other closely. In our paper, this correlation is quantified as the *intimacy score* among users and stored in the *intimacy matrix*.

Definition 2 Intimacy Matrix: In the network G, matrix $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is the intimacy matrix among users in $\mathcal{V}$, where $H(i, j)$ is the intimacy score between u_i and u_j. The intimacy score $H(i, j)$ between user $u_i, u_j \in \mathcal{V}$ denotes the probability that u_i is connected with u_j.

Our aim is to solve community detection problem in the new target network with the help of the developed source network. Thus we first provide the definition of new or developed networks, which are based on *average degree*. The *average degree* of a network denotes the average number of edges connected to each node in the network, which can depict the connection density of a network [39, 44].

Definition 3 Average Degree: The *average degree* of network $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ can be defined as $AD(G) = \frac{|\mathcal{E}|}{|\mathcal{V}|}$.

Definition 4 New and Developed Networks: Concepts "new" and "developed" can depict the sparsity of information in networks. In this paper, new networks (or well-developed networks) are defined as networks whose average degree is lower than threshold ϵ_{new} (or larger than threshold ϵ_{dev}). In other words, network $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ is a new network iff $AD(G) < \epsilon_{new}$ and G is a developed network iff $AD(G) > \epsilon_{dev}$.

Next we connect the new and developed heterogeneous network to a pair of partially aligned networks through *anchor links*.

Definition 5 Partially Aligned Attribute Augmented Heterogeneous Networks: A pair of partially aligned attribute augmented heterogeneous networks can be defined as $\mathcal{G} = (G^s, G^t, L^{s,t})$, where G^s is the source attributed augmented heterogeneous network and G^s is the target one. Both G^s and G^t can be formulated as one attribute augmented heterogeneous social network, e.g., $G^t = (\mathcal{V}^t, \mathcal{E}^t, \mathcal{A}^t)$ and its intimacy matrix is $\mathbf{H}$. $L^{s,t}$ is the set of undirected anchor links between G^s and G^t.

Definition 6 Anchor Link: Undirected link (u^s, v^t) is an anchor link between G^s and G^t if $(u^s \in \mathcal{V}^s) \wedge (v^t \in \mathcal{V}^t)$ and u^s, v^t are the accounts of the same anchor user, where $\mathcal{V}^s, \mathcal{V}^t$ are the user sets of networks G^s and G^t respectively.

Users who join G^s and G^t simultaneously can be defined as the *anchor users* between G^s and G^t.

3.2 New network community detection

New Network Community Detection problem aims at partitioning user set $\mathcal{V}^t$ of the new network G^t into K disjoint clusters, $\mathcal{C} = \{C_1, C_2, \cdots, C_K\}$, based on the *intimacy matrix*,

H, where $\bigcup_i^K C_i = \mathcal{V}^t$ and $C_i \cap C_j = \emptyset, \forall i, j \in \{1, 2, \cdots, K\}, i \neq j$. When the target network G^t is brand new, i.e., $\mathcal{E}^t = \emptyset$ and $\mathcal{A}^t = \emptyset$, the problem will be the *cold start community detection* problem.

We study the new network community detection problem based on two real-world *partially aligned networks*: Foursquare and Twitter, whose detailed information is available in Section 7. The method CAT proposed to solve this problem will be introduced in detail in the next three sections. CAT is based on the *intimacy matrix*, which can be efficiently calculated in this paper. Intimacy score in CAT, which is each element in the matrix, captures network distance (factoring into fan out), other attributes, and cross network effect in a unified and coherent way. After that, we will introduce the clustering and parameter self-adjustment method.

4 Intimacy matrix of one network

Our paper aims to solve the new network community detection problem based on the *intimacy matrix* and with the help of another developed network. In this section, we will define the *intimacy scores* and *intimacy matrix* from an information propagation perspective.

4.1 Intimacy matrix of one homogeneous network

Given one homogeneous network, e.g., $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of users and $\mathcal{E}$ is the set of social links among users in $\mathcal{V}$, we can define the adjacency matrix of G to be $\mathbf{A} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, where $A(i, j) = 1$, iff $(u_i, u_j) \in \mathcal{E}$. Meanwhile, via the social links in $\mathcal{E}$, information can propagate among the users within the network, whose propagation paths can reflect the closeness among users [26]. Formally, we define

$$p_{ij} = \frac{A(i, j)}{\sum_n A(n, j)}$$

to be the information *transition probability* from u_i to u_j. It can also be represented by the *transition matrix* $\mathbf{X}(i, j) = p_{ij}$. $\mathbf{X} = \mathbf{A}\mathbf{D}^{-1}$, where $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$, and diagonal matrix $\mathbf{D} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ with value $D(j, j) = \sum_{i=1}^{|\mathcal{V}|} A(i, j)$.

Next, we introduce an information propagation model which can depict the influence diffusion process effectively. The heat diffusion is a physical phenomenon that heat flows from an object with high temperature to another object with low one. The spread of information on social graph resembles the heat diffusion, which experts transfer their influence to other majority.

Definition 7 Heat Diffusion on Social Graph: The model assumes that user $u_i \in \mathcal{V}$ injects a stimulation into network G initially and the information will be propagated to other users in G afterwards. During the propagation process, users receive stimulation from their neighbors and the amount is proportional to the difference of the amount of information reaching the user and his neighbors over their degrees. Let vector $\boldsymbol{f}^{(\tau)} \in \mathbb{R}^{|\mathcal{V}|}$ denote the states of all users in $\mathcal{V}$ at τ, i.e., the proportion of stimulation at users in $\mathcal{V}$ at time τ. The change of stimulation at u_i at time $\tau + \Delta t$ is defined as follows:

$$\frac{f^{(\tau+\Delta t)}(i) - f^{(\tau)}(i)}{\Delta t} = \alpha \sum_{u_j \in \mathcal{V}} p_{ji}(f^{(\tau)}(j) - f^{(\tau)}(i)),$$

where α is the heat diffusion coefficient and can be set as 1. [54]

According to the above *heat diffusion model*, and based on the *transition matrix* $\mathbf{X}$, we define the *social transition probability matrix*.

Definition 8 Social Transition Probability Matrix: The social transition probability matrix of network G can be represented as $\mathbf{Q} = \mathbf{X} - \mathbf{D}$, where $\mathbf{X}$ is the transition matrix defined above and diagonal matrix $\mathbf{D} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ with value $\mathbf{D}(j, j) = \sum_{i=1}^{|\mathcal{V}|} A(i, j)$.

Furthermore, by setting $\Delta t = 1$, denoting that stimulation propagates step by step in a discrete time framework through network, we can rewrite the propagation updating equation as:

$$\boldsymbol{f}^{(\tau)} = \boldsymbol{f}^{(\tau-1)} + \alpha(\mathbf{X} - \mathbf{D})\boldsymbol{f}^{(\tau-1)} = (\mathbf{I} + \alpha\mathbf{Q})f^{(\tau-1)} = (\mathbf{I} + \alpha\mathbf{Q})^{\tau}\boldsymbol{f}^{(0)}.$$

The propagation process will stop when $\boldsymbol{f}^{(\tau)} = \boldsymbol{f}^{(\tau-1)}$, i.e., $(\mathbf{I} + \alpha\mathbf{Q})^{(\tau)} = (\mathbf{I} + \alpha\mathbf{Q})^{(\tau-1)}$.The smallest τ which can stop the propagation is defined as the *stop step*. To obtain the *stop step* τ, we need to keep checking the powers of $(\mathbf{I} + \alpha\mathbf{Q})$ until it does not change as τ increases, which meets the *stop criteria*, and get the final *intimacy matrix*.

Definition 9 Intimacy Matrix: $\mathbf{H} = (\mathbf{I} + \alpha\mathbf{Q})^{\tau} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ is defined as the intimacy matrix of users in $\mathcal{V}$, where τ is the stop step and $H(i, j)$ denotes the intimacy score between u_i and $u_j \in \mathcal{V}$ in the network.

4.2 Intimacy matrix of one attribute augmented heterogeneous network

We have discussed how to get intimacy matrix of a homogeneous network, but real-world social networks usually contain various kinds of information. Thus we extend the model from homogeneous networks to *attribute augmented heterogeneous networks*. They can be formulated as $G = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ as introduced in Section 3. Attribute set $\mathcal{A} = \{a_1, a_2, \cdots, a_m\}$ and $a_i = \{a_{i1}, a_{i2}, \cdots, a_{in_i}\}$ can have n_i different values for $i \in \{1, 2, \cdots, m\}$. An example of attribute augmented heterogeneous network is given in Figure 4, where Figure 4a is the *attribute augmented heterogeneous network*, Figure 4b–d show the attribute information in the network, which include timestamps, text and location checkins. Including the attributes as nodes provides a conceptual framework to handle social links and node attributes in a unified framework.

The connections between users and attributes can be represented as the *attribute adjacency matrix* $\mathbf{A}_i \in \mathbb{R}^{|\mathcal{V}|\times n_i}$. Similar to the social transition probability matrix of homogeneous network, we propose the *attribute transition probability matrix* based on $\mathbf{A}_i$.

Definition 10 Attribute Transition Probability Matrix: We formally define the attribute transition probability matrix from users to attribute a_i to be $\mathbf{R}_i \in \mathbb{R}^{|\mathcal{V}|\times n_i}$, where n_i is the size of attribute a_i's value. For the given user u and one attribute value j, the transition probability is calculated as:

$$\mathbf{R}_i(u, j) = \frac{\mathbf{A}_i(u, j)}{\sum_{n=1}^{|\mathcal{V}|} \mathbf{A}_i(n, j)}.$$

Similarly, we can define the attribute transition probability matrix from attribute a_i to users in $\mathcal{V}$ as $\mathbf{S}_i = \mathbf{R}_i^T$.

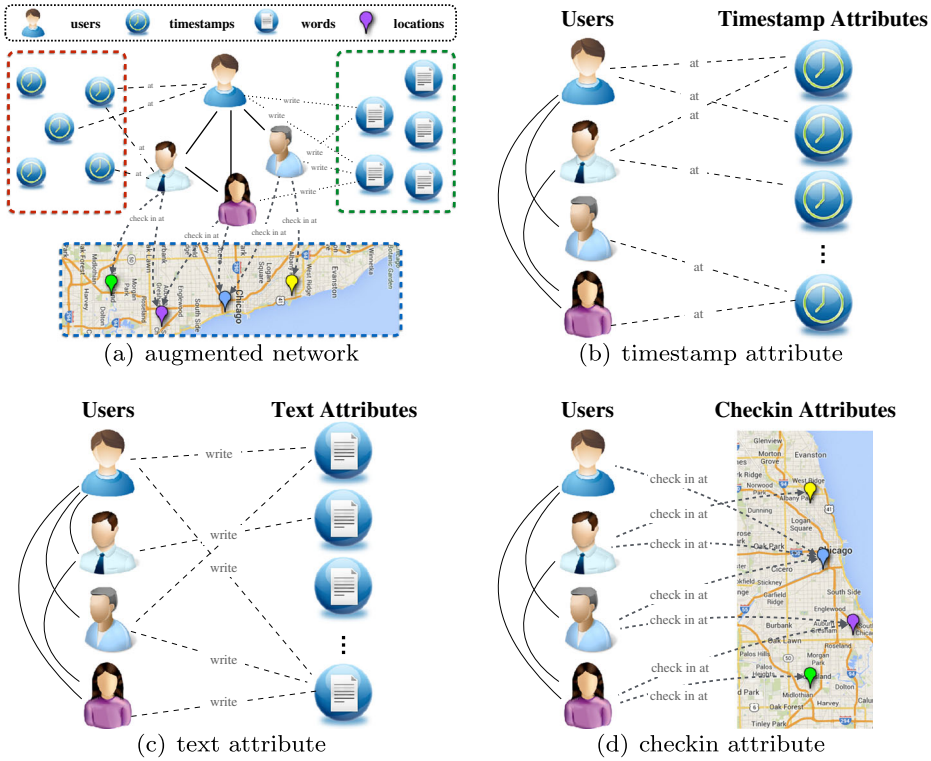


Figure 4 An example of attribute augmented heterogeneous network. **a**: attribute augmented heterogeneous network, (**b**): timestamp attribute, (**c**): text attribute, (**d**): location checkin attribute

Various kinds of attributes comprise a heterogeneous network, therefore we then give different weights to denote their importance: $\omega = \{\omega_0, \omega_1, \cdots, \omega_m\}$, where $\sum_{i=0}^{m} \omega_i = 1.0$, ω_0 is the weight of social link information and ω_i is the weight of attribute a_i, for $i \in \{1, 2, \cdots, m\}$. Together with *attribute transition probability matrix*, we define the *weighted attribute transition probability matrix*.

Definition 11 Weighted Attribute Transition Probability Matrix: Let $n_{aug} = (|\mathcal{V}| + \sum_{i=1}^{m} n_i)$ be the number of all nodes in the augmented network. With weights ω, we define matrix $\tilde{\mathbf{R}} = [\omega_1 \mathbf{R}_1, \cdots, \omega_n \mathbf{R}_n] \in \mathbb{R}^{|\mathcal{V}| \times (n_{aug} - |\mathcal{V}|)}$ to be the weighted attribute transition probability matrix from users to all attributes. Similarly, $\tilde{\mathbf{S}} = \tilde{\mathbf{R}}^T \in \mathbb{R}^{(n_{aug} - |\mathcal{V}|) \times |\mathcal{V}|}$ is the weighted attribute transition probability matrix from all attribute to users.

Furthermore, the *transition probability matrix* of whole attribute augmented heterogeneous network G is defined as $\tilde{\mathbf{Q}}_{aug} \in \mathbb{R}^{n_{aug} \times n_{aug}}$:

$$\tilde{\mathbf{Q}}_{aug} = \begin{bmatrix} \tilde{\mathbf{Q}} & \tilde{\mathbf{R}} \\ \tilde{\mathbf{S}} & \mathbf{0} \end{bmatrix},$$

where $\tilde{\mathbf{Q}} = \omega_0 \mathbf{Q} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is the *weighted social transition probability matrix* of social links in $\mathcal{E}$.

In the real world, heterogeneous social networks usually contain large amounts of attributes, i.e., n_{aug} can be extremely large. The *weighted transition probability matrix*, i.e., $\tilde{\mathbf{Q}}_{aug}$, can be extremely high dimensions and can hardly fit in the memory. As a result, it is impossible to update the matrix until it meets *stop criteria* to obtain the *stop step* and *intimacy matrix*. To solve such problem, we lower dimensional space by applying partitioned block matrix operations with the following **Lemma 1**.

Lemma 1 $(\tilde{\mathbf{Q}}_{aug})^k = \begin{bmatrix} \tilde{\mathbf{Q}}_k & \tilde{\mathbf{Q}}_{k-1}\tilde{\mathbf{R}} \\ \tilde{\mathbf{S}}\tilde{\mathbf{Q}}_{k-1} & \tilde{\mathbf{S}}\tilde{\mathbf{Q}}_{k-2}\tilde{\mathbf{R}} \end{bmatrix}$, $k \geq 2$, *where*

$$\tilde{\mathbf{Q}}_k = \begin{cases} \mathbf{I}, & \text{if } k = 0, \\ \tilde{\mathbf{Q}}, & \text{if } k = 1, \\ \tilde{\mathbf{Q}}\tilde{\mathbf{Q}}_{k-1} + \tilde{\mathbf{R}}\tilde{\mathbf{S}}\tilde{\mathbf{Q}}_{k-2}, & \text{if } k \geq 2 \end{cases}$$

and the intimacy matrix among users in $\mathcal{V}$ *can be represented as*

$$\begin{aligned} \tilde{\mathbf{H}}_{aug} &= \left(\mathbf{I} + \alpha\tilde{\mathbf{Q}}_{aug}\right)^{\tau}(1:|\mathcal{V}|, 1:|\mathcal{V}|) \\ &= \left(\sum_{t=0}^{\tau}\binom{\tau}{t}\alpha^t(\tilde{\mathbf{Q}}_{aug})^t\right)(1:|\mathcal{V}|, 1:|\mathcal{V}|) \\ &= \left(\sum_{t=0}^{\tau}\binom{\tau}{t}\alpha^t\left((\tilde{\mathbf{Q}}_{aug})^t(1:|\mathcal{V}|, 1:|\mathcal{V}|)\right)\right) \\ &= \left(\sum_{t=0}^{\tau}\binom{\tau}{t}\alpha^t\tilde{\mathbf{Q}}_t\right), \end{aligned}$$

where $\mathbf{X}(1:|\mathcal{V}|, 1:|\mathcal{V}|)$ *is a sub-matrix of* $\mathbf{X}$ *with indexes in* $[1, |\mathcal{V}|]$, τ *is the stop step, achieved when* $\tilde{\mathbf{Q}}_\tau = \tilde{\mathbf{Q}}_{\tau-1}$, *i.e., the stop criteria,* $\tilde{\mathbf{Q}}_\tau$ *is called the stationary matrix.*

Proof The lemma can be proved by induction on k [53]. Considering that $(\tilde{\mathbf{R}}\tilde{\mathbf{S}}) \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{V}|}$ can be pre-computed in advance, the space cost of Lemma 1 is $O(|\mathcal{V}|^2)$, $|\mathcal{V}| \ll n_{aug}$. □

Since we are only interested in the *intimacy* and *transition matrices* among users, not those between the augmented items and users, we create a reduced dimensional representation only involving users for $\tilde{\mathbf{Q}}_k$ and $\tilde{\mathbf{H}}$ such that we can capture the effect of "user-attribute" and "attribute-user" transition on "user-user" transition. $\tilde{\mathbf{Q}}_k$ is a reduced dimension representation of $\tilde{\mathbf{Q}}^k_{aug}$, while eliminating the augmented items, it still maintains the "user-user" transitions effectively.

5 Intimacy matrix of aligned networks

We have introduced how to get the *intimacy matrix* $\tilde{\mathbf{H}}$ of one attribute augmented heterogeneous network, however when G^t is new, the matrix among users calculated based on the information in G^t can be very sparse. To solve this problem, we propose to propagate useful information from other well developed aligned networks to the new network in this section.

5.1 Intimacy matrix across aligned networks

Information propagated from other aligned well-developed networks can help solve the shortage of information problem in the new network [46, 47]. However, as proposed in [25], different networks can have different properties and information propagated from other well-developed aligned networks can be very different from that of the new network as well.

To handle this problem, we use weights, $\rho^{s,t}$, $\rho^{t,s} \in [0, 1]$, to control the proportion of information propagated between developed network G^s and new network G^t. If information from G^s is helpful for improving the community detection results in G^t, we can set a higher $\rho^{s,t}$ to propagate more information from G^s. Otherwise, we can set a lower $\rho^{s,t}$ instead. The weights $\rho^{s,t}$ and $\rho^{t,s}$ can be adjusted automatically with method to be introduced in Section 6.

Based on weights $\rho^{s,t}$ and $\rho^{t,s}$, we get the *weighted network transition probability matrix* of G^t and G^s to be

$$\bar{\mathbf{Q}}^t_{aug} = (1-\rho^{t,s})\begin{bmatrix} \tilde{\mathbf{Q}}^t & \tilde{\mathbf{R}}^t \\ \tilde{\mathbf{S}}^t & \mathbf{0} \end{bmatrix}, \qquad \bar{\mathbf{Q}}^s_{aug} = (1-\rho^{s,t})\begin{bmatrix} \tilde{\mathbf{Q}}^s & \tilde{\mathbf{R}}^s \\ \tilde{\mathbf{S}}^s & \mathbf{0} \end{bmatrix},$$

where $\bar{\mathbf{Q}}^t_{aug} \in \mathbb{R}^{n^t_{aug} \times n^t_{aug}}$ and $\bar{\mathbf{Q}}^s_{aug} \in \mathbb{R}^{n^s_{aug} \times n^s_{aug}}$, n^t_{aug} and n^s_{aug} are the numbers of all nodes in G^t and G^s respectively.

To propagate information across networks, we define the *anchor transition matrix*:

Definition 12 Anchor Transition Matrix: Given a pair of partially aligned heterogeneous networks $\mathcal{G} = (G^s, G^t, L^{s,t})$, $\mathbf{T}^{t,s} \in \mathbb{R}^{|\mathcal{V}^t| \times |\mathcal{V}^s|}$ and $\mathbf{T}^{s,t} \in \mathbb{R}^{|\mathcal{V}^s| \times |\mathcal{V}^t|}$ are defined as anchor transition matrices between G^t and G^s. $\mathbf{T}^{t,s}(i, j) = \mathbf{T}^{s,t}(j, i) = 1$, iff $(u^t_i, u^s_j) \in L^{s,t}$, $u^t_i \in \mathcal{V}^t$, $u^s_j \in \mathcal{V}^s$.

Furthermore, the *weighted anchor transition matrices* between G^s and G^t are

$$\bar{\mathbf{T}}^{t,s} = (\rho^{t,s})\begin{bmatrix} \mathbf{T}^{t,s} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \bar{\mathbf{T}}^{s,t} = (\rho^{s,t})\begin{bmatrix} \mathbf{T}^{s,t} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

where $\bar{\mathbf{T}}^{t,s} \in \mathbb{R}^{n^t_{aug} \times n^s_{aug}}$ and $\bar{\mathbf{T}}^{s,t} \in \mathbb{R}^{n^s_{aug} \times n^t_{aug}}$. Nodes corresponding to entries in $\bar{\mathbf{T}}^{t,s}$ and $\bar{\mathbf{T}}^{s,t}$ are of the same order as those in $\bar{\mathbf{Q}}^t_{aug}$ and $\bar{\mathbf{Q}}^s_{aug}$ respectively.

Based on the above definitions, the *transition probability matrix* across aligned networks is defined as

$$\bar{\mathbf{Q}}_{align} = \begin{bmatrix} \bar{\mathbf{Q}}^t_{aug} & \bar{\mathbf{T}}^{t,s} \\ \bar{\mathbf{T}}^{s,t} & \bar{\mathbf{Q}}^s_{aug} \end{bmatrix}$$

where $\bar{\mathbf{Q}}_{align} \in \mathbb{R}^{n_{align} \times n_{align}}$, $n_{align} = n^t_{aug} + n^s_{aug}$ is the number of all nodes across the aligned networks.

According to the definition of intimacy matrix, with $\bar{\mathbf{Q}}_{align}$, we can obtain the *aligned network intimacy matrix* $\bar{\mathbf{H}}_{align}$ of users in G^t to be $\bar{\mathbf{H}}_{align} = (\mathbf{I} + \alpha\bar{\mathbf{Q}}_{align})^\tau(1 : |\mathcal{V}^t|, 1 : |\mathcal{V}^t|)$, where $\bar{\mathbf{H}}_{align} \in \mathbb{R}^{|\mathcal{V}^t| \times |\mathcal{V}^t|}$, τ is the *stop step*.

Meanwhile, the structure of $(\mathbf{I} + \alpha\bar{\mathbf{Q}}_{align})$ can not meet the requirements of Lemma 1 as it does not have a zero square matrix at the bottom right corner. As a result, methods introduced in Lemma 1 cannot be applied. To obtain the *stop step*, we have no choice but to keep calculating powers of $(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align})$ until the *stop criteria* can meet, which can be very time consuming. In this part, we propose to solve the problem with the following Lemma 2.

Lemma 2 *For the given matrix* $(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align})$*, its* k_{th} *power meets*

$$(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align})^k\mathbf{P}=\mathbf{P}\Lambda^k, k\geq 1,$$

matrices $\mathbf{P}$ *and* Λ *contain the eigenvector and eigenvalues of* $(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align})$*. The* i_{th} *column of matrix* $\mathbf{P}$ *is the eigenvector of* $(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align})$ *corresponding to its* i_{th} *eigenvalue* λ_i *and diagonal matrix* Λ *has value* $\Lambda(i,i)=\lambda_i$ *on its diagonal.*

The proof of Lemma 2 can refer to the Appendix. The time cost of calculating Λ^k is $O(n_{align})$, which is far less than that required to calculate $(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align})^k$.

In addition, if $\mathbf{P}$ is invertible, we can have $(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align})^k=\mathbf{P}\Lambda^k\mathbf{P}^{-1}$, where Λ^k has $\Lambda(i,i)^k$ on its diagonal. Thus the intimacy calculated based on *eigenvalue decomposition* is

$$\bar{\mathbf{H}}_{align}=\left(\mathbf{P}\Lambda^{\tau}\mathbf{P}^{-1}\right)(1:|\mathcal{V}^t|,1:|\mathcal{V}^t|).$$

where the stop step τ can be obtained when $\mathbf{P}\Lambda^{\tau}\mathbf{P}^{-1}=\mathbf{P}\Lambda^{\tau-1}\mathbf{P}^{-1}$, i.e., *stop criteria*.

5.2 Approximated intimacy to reduce dimension

Eigendecomposition based method proposed in Lemma 2 enables us to calculate the powers of $(\mathbf{I}+\alpha\mathbf{Q}_{align})$ very efficiently. However, when applying Lemma 2 to calculate the *intimacy matrix* of real-world partially aligned networks, it can suffer from many serious problems. The reason is that the dimension of $(\mathbf{I}+\alpha\mathbf{Q}_{align})$, i.e., $n_{align}\times n_{align}$, is so high that matrix $(\mathbf{I}+\alpha\mathbf{Q}_{align})$ can hardly fit in the memory. To solve that problem, in this part, we propose to calculate the approximated *intimacy matrix* $\bar{\mathbf{H}}_{align}^{approx}$ with less space and time costs instead.

Let's define the transition probability matrices of G^t and G^s to be $\tilde{\mathbf{Q}}_{aug}^t$ and $\tilde{\mathbf{Q}}_{aug}^s$ respectively. By applying **Lemma 1**, we can get their *stop step* and the *stationary matrices* to be τ^t, τ^s, $\tilde{\mathbf{Q}}_{\tau^t}^t$ and $\tilde{\mathbf{Q}}_{\tau^s}^t$ respectively.

Stationary matrices $\tilde{\mathbf{Q}}_{\tau^t}^t$, $\tilde{\mathbf{Q}}_{\tau^s}^t$ together with the *anchor transition matrices* $\mathbf{T}^{t,s}$ and $\mathbf{T}^{s,t}$, can be used to define a low-dimensional *reduced aligned network transition probability matrix*, which only involves users explicitly, while the effect of "attribute-user" or "user-attribute" transition is implicitly absorbed into $\tilde{\mathbf{Q}}_{\tau^t}^t$ and $\tilde{\mathbf{Q}}_{\tau^s}^s$:

$$\bar{\mathbf{Q}}_{align}^{user}=\begin{bmatrix}(1-\rho^{t,s})\tilde{\mathbf{Q}}_{\tau^t}^t & (\rho^{t,s})\mathbf{T}^{t,s}\\ (\rho^{s,t})\mathbf{T}^{s,t} & (1-\rho^{s,t})\tilde{\mathbf{Q}}_{\tau^s}^s\end{bmatrix},$$

where $\bar{\mathbf{Q}}_{align}^{user}\in\mathbb{R}^{(|\mathcal{V}|^t+|\mathcal{V}^s|)^2}$ and $(|\mathcal{V}|^t+|\mathcal{V}^s|)\ll n_{align}$.

Furthermore, with Lemma 2, we can get *approximated aligned network intimacy matrix* of users in G^t based on $\bar{\mathbf{Q}}_{align}^{user}$ to be:

$$\bar{\mathbf{H}}_{align}^{approx}=\left(\mathbf{P}^*(\Lambda^*)^{\tau}(\mathbf{P}^*)^{-1}\right)(1:|\mathcal{V}^t|,1:|\mathcal{V}^t|),$$

where $(\mathbf{I}+\alpha\bar{\mathbf{Q}}_{align}^{user})=\mathbf{P}^*\Lambda^*(\mathbf{P}^*)^{-1}$ and τ is the *stop step*.

5.3 Space and time costs analysis

Let $|\mathcal{V}^t|=n^t$, the size of intimacy matrix $\bar{\mathbf{H}}_{align}$ will be $(n^t)^2$. However, to obtain $\bar{\mathbf{H}}_{align}$, we need to calculate the transition probability matrix $\bar{\mathbf{Q}}_{align}$ in advance, whose size is $(n_{align})^2$.

Space cost In eigendecomposition based method, we have to calculated and store matrices $\bar{\mathbf{Q}}_{align}^{eigen}$, $\mathbf{P}$, $\mathbf{P}^{-1}$, $\Lambda \in \mathbb{R}^{n_{align} \times n_{align}}$, whose space costs are $O(4n_{align}^2)$. However, in the approximation based method, we just need to store matrices $\tilde{\mathbf{Q}}^x \in \mathbb{R}^{n^x \times n^x}$, $\tilde{\mathbb{R}}^x \in \mathbb{R}^{n^x \times (\sum_i n_i^x)}$, $\tilde{\mathbf{S}}^x \in \mathbb{R}^{(\sum_i n_i^x) \times n^x}$, $x \in \{s, t\}$, as well as $\bar{\mathbf{Q}}_{align}^{approx} \in \mathbb{R}^{(n^t+n^s)\times(n^t+n^s)}$, whose space cost will be $O(\max\{(n^t + n^s)^2, n^t(\sum_i n_i^t), n^s(\sum_i n_i^s)\}) < O(4n_{align}^2)$.

Time cost In eigendecomposition based method, the matrix eigendecomposition of $\bar{\mathbf{Q}}_{align}^{eigen}$, inversion $\mathbf{P}^{-1}$ and multiplication of $\mathbf{P}\Lambda^k\mathbf{P}^{-1}$ are all time-consuming operations, whose time costs are $O(kn_{align}^2)$ [42], $O(n_{align}^2 \log(n_{align}))$ [5] and $O(2n_{align}^3)$ respectively. As a result, the time cost of eigendecomposition based method is about $O(2n_{align}^3)$. However, in approximation based methods, we need to apply Lemma 2 to get $\bar{\mathbf{H}}^t$ and $\bar{\mathbf{H}}^t$, whose time cost is

$$O(\max\{\tau((n^t)^3 + (n^t)^2(\sum_i a_i^t)), \tau((n^s)^3 + (n^s)^2(\sum_i a_i^s))\}),$$

which is much smaller than that of eigendecomposition based methods.

6 Clustering and weight self-adjustment

Intimacy matrix $\bar{\mathbf{H}}_{align}$ (or $\bar{\mathbf{H}}_{align}^{approx}$) stores the intimacy scores among users in $\mathcal{V}^t$ and can be used to detect the communities in the network. In this section, we will use two methods to solve the user clustering problem in the target network: *Spectral Clustering* and *Low-Rank Matrix Factorization*. Meanwhile, the weight self-adjustment method is also introduced to get the better community detection result.

6.1 Spectral clustering

The first technique we used for community detection in this paper is *Spectral Clustering*[19, 22], which is a efficient method based on *Laplacian Eigenmaps* [2].

The target network has been model as a graph G^t and its intimacy matrix $\bar{\mathbf{H}}_{align}$ (or $\bar{\mathbf{H}}_{align}^{approx}$), which stores the intimacy scores among users. Let D be the diagonal matrix with the value $d_{ii} = \sum_j \bar{\mathbf{H}}_{align}[j, i]$ on the diagonal. The *Laplacian matrix* $L = D - \bar{\mathbf{H}}_{align}$.

We define:

$$cut(A, B) = \sum_{i \in A, j \in B} \bar{h}_{ij},$$

where $\bar{h}_{ij}$ is the entry of intimacy matrix $\bar{\mathbf{H}}_{align}$. Therefore the clustering problem can be transformed to a mincut problem, which is choosing the partition $\{C_1, \cdots, C_k\}$ to minimizes

$$cut(C_1, \cdots, C_k) = \sum_{i=1}^{k} cut(C_i, \bar{C}_i),$$

where $\bar{C}$ is the complement of a subset C. One of the most common objective functions to address the mincut problem, which also explicitly request that the sets $\{C_1, \cdots, C_k\}$ are "reasonably large", is normalized cut NCUT[33].

$$Ncut(C_1, \cdots, C_k) = \sum_{i=1}^{k} \frac{cut(C_i, \bar{C}_i)}{vol(C_i)}$$

where $vol(C) = \sum_{i \in C} d_{ii}$. The objective function tries to achieve that the clusters are "balanced", as measured by the edge weights.

When finding $k > 2$ clusters, we define the vectors $z_i = (z_{1i}, \cdots, z_{n^t i})$ by

$$z_{ij} = \begin{cases} 1/\sqrt{C_i} & \text{if } i \in C_i \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathbf{Z}$ contains those vectors as columns. We observe that the columns in $\mathbf{Z}$ are orthonormal to each other, thus $\mathbf{Z}'\mathbf{Z} = I$. We can also check $z'\mathbf{D}z = 1$ and $z'\mathbf{L}z = 2Cut(C_i, \bar{C}_i)/vol(C_i)$. Therefore we can rewrite the problem of minimizing NCUT as:

$$\begin{array}{ll} min & Tr(\mathbf{Z}'\mathbf{L}\mathbf{Z}) \\ s.t. & \mathbf{Z}'\mathbf{D}\mathbf{Z} = I \end{array}$$

where $Tr(\cdot)$ denotes the trace of a matrix. The elements of $\mathbf{F}$ are constrained to be discrete values, which makes it a NP-hard discrete optimization problem. The obvious relaxation is to discard the condition on the discrete values in $\mathbf{F}$ and instead allow $f_i \in R$. Thus when we substitute $F = \mathbf{D}^{1/2}\mathbf{Z}$, the optimization function is

$$\begin{array}{ll} min & Tr(\mathbf{F}'\mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}}\mathbf{F}) \\ s.t. & \mathbf{F}'\mathbf{F} = I \end{array}$$

After using eigenvalue decomposition(EVD), we can get the eigenvectors of $\mathbf{L}$ and the simple k-means clustering algorithm has no difficulties to detect the clusters in this new representation.

6.2 Low-rank matrix factorization

The other method is to use the low-rank matrix factorization method proposed in [36] to get the latent feature vectors, $\mathbf{U}$, for each user. To avoid overfitting, we add two regularization terms to the object function as follows:

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V}} & \left\| \bar{\mathbf{H}}_{align} - \mathbf{U}\mathbf{V}\mathbf{U}^T \right\|_F^2 + \theta \left\|\mathbf{U}\right\|_F^2 + \beta \left\|\mathbf{V}\right\|_F^2, \\ & s.t. \mathbf{U} \geq \mathbf{0}, \mathbf{V} \geq \mathbf{0}, \end{aligned}$$

where $\mathbf{U}$ is the latent feature vectors, $\mathbf{V}$ stores the correlation among rows of $\mathbf{V}$, θ and β are the weights of $\|\mathbf{U}\|_F^2$, $\|\mathbf{V}\|_F^2$ respectively.

This object function is hard to solve because it is a great challenge that obtaining the global optimal result for both $\mathbf{U}$ and $\mathbf{V}$ simultaneously. We propose to solve the objective function by fixing one variable, e.g., $\mathbf{U}$, and update another variable, e.g.,$\mathbf{V}$, alternatively [36], whose update equations are as follows:

We can get the Lagrangian function of the object equation as follows:

$$\begin{aligned} \mathcal{F} = & Tr(\bar{\mathbf{H}}_{align}\bar{\mathbf{H}}_{align}^T) - Tr(\bar{\mathbf{H}}_{align}\mathbf{U}\mathbf{V}^T\mathbf{U}^T) \\ & -Tr(\mathbf{U}\mathbf{V}\mathbf{U}^T\bar{\mathbf{H}}_{align}^T) + Tr(\mathbf{U}\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T\mathbf{U}^T) \\ & +\theta Tr(\mathbf{U}\mathbf{U}^T) + \beta Tr(\mathbf{V}\mathbf{V}^T) - Tr(\Theta\mathbf{U}) - Tr(\Omega\mathbf{V}) \end{aligned}$$

where Θ and Ω are the multiplier for the constraint of $\mathbf{U}$ and $\mathbf{V}$ respectively.

By taking derivatives of $\mathcal{F}$ with regarding to $\mathbf{U}$ and $\mathbf{V}$, we can get

$$\begin{aligned}\frac{\partial \mathcal{F}}{\partial \mathbf{U}} &= -2\bar{\mathbf{H}}_{align}^{T}\mathbf{U}\mathbf{V} - 2\bar{\mathbf{H}}_{align}\mathbf{U}\mathbf{V}^{T} + 2\mathbf{U}\mathbf{V}^{T}\mathbf{U}^{T}\mathbf{U}\mathbf{V}^{T} \\ &\quad +2\mathbf{U}\mathbf{V}\mathbf{U}^{T}\mathbf{U}\mathbf{V}^{T} + 2\theta\mathbf{U} - \Theta^{T} \\ \frac{\partial \mathcal{F}}{\partial \mathbf{V}} &= -2\mathbf{U}^{T}\bar{\mathbf{H}}_{align}\mathbf{U} + 2\mathbf{U}^{T}\mathbf{U}\mathbf{V}\mathbf{U}^{T}\mathbf{U} + 2\beta\mathbf{V} - \Omega^{T}\end{aligned}$$

Let $\frac{\partial \mathcal{F}}{\partial \mathbf{U}} = \mathbf{0}$ and $\frac{\partial \mathcal{F}}{\partial \mathbf{V}} = \mathbf{0}$ and use the KKT complementary condition, we can get

$$\mathbf{U}(i,j) \leftarrow \mathbf{U}(i,j)\sqrt{\frac{\left(\bar{\mathbf{H}}_{align}^{T}\mathbf{U}\mathbf{V} + \bar{\mathbf{H}}_{align}\mathbf{U}\mathbf{V}^{T}\right)(i,j)}{\left(\mathbf{U}\mathbf{V}^{T}\mathbf{U}^{T}\mathbf{U}\mathbf{V} + \mathbf{U}\mathbf{V}\mathbf{U}^{T}\mathbf{U}\mathbf{V}^{T} + \theta\mathbf{U}\right)(i,j)}},$$

$$\mathbf{V}(i,j) \leftarrow \mathbf{V}(i,j)\sqrt{\frac{\left(\mathbf{U}^{T}\bar{\mathbf{H}}_{align}\mathbf{U}\right)(i,j)}{\left(\mathbf{U}^{T}\mathbf{U}\mathbf{V}\mathbf{U}^{T}\mathbf{U} + \beta\mathbf{V}\right)(i,j)}}.$$

The low-rank matrix $\mathbf{U}$ captures the information of each users from the intimacy matrix and can be used as latent numerical feature vectors to cluster users in G^t with traditional clustering methods, e.g., Kmeans [8].

Algorithm 1 CAT with Parameter Self-Adjustment

Input: aligned network: $\mathcal{G} = \{\{G^t, G^s\}, \{A^{t,s}, A^{s,t}\}\}$
parameters: ω, ρ, γ, α, β and method type M
Output: community detection results of G^t: $\mathcal{C}$

$\omega_{old} = \omega$, $\rho_{old} = \rho$, $E_{old} = \infty$
for parameter $\delta \in \omega \cup \{\rho\}$ **do**
  while $True$ **do**
    $\delta = (1+\gamma)\delta$ and renormalize ω if$\delta \in \omega$ to get ω_{new}, ρ_{new}
    construct transition probability matrix $\bar{\mathbf{Q}}_{align}$
    if $M = approximation$ **then**
      construct $\bar{\mathbf{Q}}_{align}^{user}$ with $\tilde{\mathbf{Q}}_{\tau^t}^{t}$, $\tilde{\mathbf{Q}}_{\tau^s}^{s}$ calculated according to Lemma 1
      calculate $\bar{\mathbf{H}}_{align}^{approx}$ with $\bar{\mathbf{Q}}_{align}^{user}$ according to Lemma 2
      $\bar{\mathbf{H}}_{align} = \bar{\mathbf{H}}_{align}^{approx}$
    else
      calculate $\bar{\mathbf{H}}_{align}$ with $\bar{\mathbf{Q}}_{align}$ according to Lemma 2
    end if
    get lower-dimensional latent feature vectors $\mathbf{U}$ and $\mathcal{C} = Kmeans(\mathbf{U})$,
    or using spectral clustering to detect communities
    $E_{new} = -\sum_{i=1}^{K} P(i)\log P(i)$, $P(i) = \frac{|U_i|}{\sum_{i=1}^{K}|U_i|}$, $U_i \in \mathcal{C}$
    if $E_{new} < E_{old}$ **then**
      $\omega_{old} = \omega$, $\rho_{old} = \rho$, $E_{old} = E_{new}$
    else
      $\omega = \omega_{old}$, $\rho = \rho_{old}$
      break
    end if
  end while
end for

6.3 Weight self-adjustment

Meanwhile, to handle the *information heterogeneity problem* in each network and the *network difference problem* across networks, we use weights, ω^t, ω^s, $\rho^{t,s}$ and $\rho^{s,t}$, to denote the importance of information in G^t, G^s and that propagated from G^t and G^s respectively. For simplicity, we set $\omega^t = \omega^s = \omega = [\omega_0, \omega_1, \cdots, \omega_m]$ and $\rho^{t,s} = \rho^{s,t} = \rho$ in this paper.

Let $\mathcal{C}$ be the community detection result achieved by CAT in G^t. The optimal result $\mathcal{C}$, evaluated by some metrics, e.g., *entropy* [54], can be achieved with the following equation:

$$\omega, \rho = \min_{\omega,\rho} E(\mathcal{C}).$$

The optimization problem is very difficult to solve. Next, we will propose a method to adjust ω and ρ automatically to enable CAT to achieve better results.

The weight adjustment method used to deal with ω can work as follows: for example, in network G^t, we have relational information and attribute information $\mathcal{E}$ and $\mathcal{A} = \{A_1, A_2, \cdots, A_m\}$, whose weights are initialized to be $\omega = \{\omega_0, \omega_1, \cdots, \omega_m\}$. For $\omega_i \in \omega, i \in \{0, 1, \cdots, m\}$, we keep checking if increasing ω_i by a ratio of γ, i.e., $(1+\gamma)\omega_i$, can improve the performance or not. If so, $(1+\gamma)\omega_i$ after re-normalization is used as the new value of ω_i; otherwise, we restore the old ω_i before increase and study ω_{i+1}. In the experiment, γ is set as 0.05. Similarly, for the weight of different networks, i.e, ρ, we can adjust them with the same methods to find the optimal ρ.

The pseudo code of CAT is available in Algorithm 1.

7 Experiments

To demonstrate the effectiveness of CAT, in this section, we will conduct extensive experiments on two real-world aligned online heterogeneous networks: Foursquare and Twitter.

7.1 Dataset description

Foursquare is a famous location-based social networks offering geographic services [23]. Meanwhile, Twitter is hot social network providing microblogging service, which has totally different characteristics and goals from Foursquare [14]. Both Foursquare and Twitter users can make friends with other users, write online posts, which can have text content, timestamps and location check-ins. However, as illustrated in the following statistical information, the structure of Foursquare and Twitter can be very different.

- **Foursquare**: 5,392 Foursquare users are crawled, who have established 76,972 social links. On average, each Foursquare user has 14 social links. All the 48,756 tips written by these 5,392 Foursquare users are crawled and the average number of tips written by each user is less than 9. Each tip can contain location checkins and the number of location links is 48,756.
- **Twitter**: 5,223 Twitter users and 164,920 social links are crawled. On average, users in Twitter can follow 32 friends, who are more densely connected than those in Foursquare. These 5,223 Twitter user write 9,490,707 tweets in all and each Twitter user has written 1817 tweets, whose number is much larger than that in Foursquare. 615,515 tweets can have location checkins, accounting for 6.49% of the total tweets.

The datasets used in this paper are those proposed in [12, 46, 47], crawled during November, 2012. The anchor links are obtained by crawling the users' Twitter IDs from their Foursquare homepages, whose number is 3,388.

7.2 Experiment settings

In this part, we will introduce the experiment settings in details, which include comparison methods, evaluation metrics and experiment setups.

7.2.1 Comparison methods

We have different implementations of CAT, which are compared with both state-of-art and traditional community detection methods. All the comparison methods can be divided into 3 categories:

Methods with parameter adjustment

- CATS-A (Spectral clustering with exact intimacy matrix and parameter Adjustment): CATS-A can calculate the exact intimacy matrix across aligned attribute augmented networks based on eigenvalue decomposition as proposed in Subsection 5.1, and use spectral clustering to detect communities and adjust parameters ρ and ω automatically.
- CATE-A (matrix factorization with exact intimacy matrix and parameter Adjustment): Similar to CATS-A, CATE-A also obtains the exact intimacy matrix, but it detects communities with matrix factorization method. The parameters are adjusted automatically too.
- CATA-A (matrix factorization with Approximated intimacy matrix and parameter Adjustment): CATA-A is similar to CATE-A except that CATA-A calculate the *intimacy matrix* with the lower-dimensional *reduced aligned network transition probability matrices* method as proposed in Section 5.2.

Methods without parameter adjustment

- CATS (Spectral clustering with exact intimacy matrix): CATS is the same with CATS-A except in CATS, ω and ρ are fixed as $\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ and 0.8 respectively.
- CATE (matrix factorization with exact intimacy matrix): CATE is identical to CATE-A except that in CATE, ω and ρ are fixed as $\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ and 0.8 respectively.
- CATA (matrix factorization with Approximated intimacy matrix): CATA is identical to CATA-A except that in CATA, ω and ρ are fixed as $\{\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\}$ and 0.8 respectively.

Single network clustering methods

- SINFL (Social Influence-based clustering): SINFL proposed in 2013 [54] can detect the communities with the influence matrix calculated based on the new network only.
- NCUT (Normalized Cut): NCUT [32] aiming at minimizing the normalized cut between different clusters can be used to detect the communities based on the influence matrix obtained by SINFL in the new network.
- KMEANS (Kmeans): KMEANS [8] is a traditional clustering methods, which can also detect social communities in online social networks based on the influence matrix obtained by SINFL in the new network.

- LPA (Label Propagation Algorithm): LPA [30] starts from local neighborhood to recognize communities automatically and adopts an asynchronous update strategy where nodes join in groups under their neighbors' choices.
- TopLeader (Top Leaders): TopLeader [10] finds communities according to local leader groups. It gradually associates nodes to the nearest leaders and locally reelects new leaders during each iteration.

7.2.2 Evaluation metrics

Evaluation metrics used to evaluate the performance of all the comparison methods in the experiment include:

- *normalized Davies-Bouldin index* (*ndbi*):

$$ndbi(\mathcal{C}) = \frac{1}{K}\sum_{i=1}^{K}\min_{j\neq i}\frac{d(c_i,c_j)+d(c_j,c_i)}{\sigma_i+\sigma_j+d(c_i,c_j)+d(c_j,c_i)},$$

  where c_i is the centroid of $U_i \in \mathcal{C}$, $d(c_i, c_j)$ is the distance between c_i and c_j, σ_i denotes the average distance between items in U_i and centroid c_i [54].
- *Silhouette index* (*silhouette*):

$$silhouette(\mathcal{C}) = \frac{1}{K}\sum_{i=1}^{K}\left(\frac{1}{|U_i|}\sum_{u\in U_i}\frac{b(u)-a(u)}{\max\{a(u),b(u)\}}\right),$$

  where $a(u) = \frac{1}{|U_i|-1}\sum_{v\in U_i, v\neq u} d(u,v)$ and $b(u) = \min_{j,j\neq i}\left(\frac{1}{|U_j|}\sum_{v\in U_j} d(u,v)\right)$ [18].
- *Entropy* (*E*) [54]:

$$E(\mathcal{C}) = -\sum_{i=1}^{K} P(i)\log P(i), where P(i) = \frac{|U_i|}{|\mathcal{V}|}.$$

7.2.3 Experiment setups

In the experiment, Foursquare and Twitter are regarded as the new network and well developed network respectively. As proposed in [46, 47], to represent different information we know about the new network, we randomly sample a proportion of its information, which include social links, and the attribute information, e.g., location checkins, text used and timestamps, from the new network and delete all the remaining information under the controlled of $\sigma_F(\sigma_T) \in [0, 1]$. Actually, $\sigma_F(\sigma_T)$ can also represent ϵ_{new} defined in Section 3. Take the new network Foursquare as an example, if $\sigma_F = 0.0$, Foursquare is brand new and no information about it exist; if $\sigma_F = 0.8$, 80% of the information is preserved in the network. In addition, considering the abnormally large number of locations and words in

each network, only top 5000 locations that users frequently visited and top 5000 words that users often used in each network are used to construct transition probability matrix. Different methods applied to the new network can obtain the clustering results of the users in it. To check whether these clustering methods can discover the communities in the real world, we evaluate the clustering results based on the similarity matrix among users calculated with original complete social information and the similarity measure used is *Jaccard's Coefficient*. If methods can obtain enough reliable information from the new network or other well-developed networks, then their performance should be very good evaluated by different metrics based on the similarity matrix.

7.3 Experiment results

The experiment results are shown in Tables 3 and 4. Parameter K is fixed as 50 and the ratio of anchor links σ_A is fixed as 0.8 but the information sampling rate (i.e., σ_F) changes with values in $\{0.0, 0.1, \cdots, 1.0\}$ to create severe newness situations and the results are evaluated by metrics: *ndbi*, *entropy* and *silhouette*.

As shown in Table 4, clustering method SINFL, NCUT, KMEANS, LPA and TOPLEADER cannot work when $\sigma_F = 0.0$, where the Foursquare network is brand new and users in it have no information (neither social link nor other attribute). However, CATS-A, CATE-A, CATA-A, CATS, CATE and CATA, based on the intimacy matrix across aligned networks, can still work well. For example, when $\sigma_F = 0.0$, the *ndbi* score of CATE-A is 0.973; the *entropy* is 3.337; the *silhouette* is -0.383, which are very good even compared with algorithms for single network at $\sigma_F = 0.5$.

Compared with CATE (or CATA, CATS), CATE-A (or CATA-A, CATS-A) can perform better in most cases when evaluated by *ndbi*, *silhouette* and *entropy*. For example, when $\sigma_F = 0.8$, the *ndbi* of CATE-A is 0.991, which is 3.6% higher than that of CATE; the *silhouette* of CATE-A is -0.216, which is 20% better than that of CATE. CATE-A (or CATA-A, CATS-A) can always perform better that CATE (or CATA, CATS) for $\sigma_F \in \{0.0, 0.1, \cdots, 1.0\}$ when evaluated by *entropy*, as *entropy* is used as the metric when adjusting the parameters. This shows CATE-A (or CATA-A, CATS-A) is effective in handling network difference to avoid negative transfer as well as information heterogeneity to identify the relevant information.

By comparing CATE, CATA, CATS with CATE-A, CATA-A, CATS-A respectively, methods based on approximated intimacy matrix can achieve very similar results as those based on matrix eigendecomposition. Meanwhile, as shown in Table 3 the memory space and time needed by CATA and CATA-A to calculate $\bar{\mathbf{H}}_{align}^{approx}$ is much less than that of CATE and CATE-A to calculate the exact *intimacy matrix*. So, calculating intimacy matrix with approximation would not harm the performance very much but can do save lots of space and time.

Table 3 Space and time costs in calculating $\bar{\mathbf{H}}_{align}$

		Method	
New network	Cost	Exact	Approx.
Foursquare	space cost(MB)	19526	1627
	time cost(s)	65996.17	6499.97
Twitter	space cost(MB)	23081	2057
	time cost(s)	71128.29	7901.94

Table 4 Community Detection Result of Foursquare

		Information Sampling Rate σ_F										
Measure	Methods	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
ndbi	CATS-A	0.957	0.933	0.951	0.927	0.932	0.952	0.953	0.934	0.956	0.932	0.960
	CATE-A	0.973	0.960	0.970	0.983	0.961	0.957	0.966	0.959	0.991	0.967	0.989
	CATA-A	0.945	0.928	0.920	0.936	0.939	0.954	0.941	0.929	0.945	0.923	0.959
	CATS	0.946	0.930	0.932	0.928	0.921	0.935	0.933	0.938	0.947	0.924	0.961
	CATE	0.961	0.946	0.945	0.940	0.938	0.958	0.952	0.940	0.957	0.939	0.968
	CATA	0.937	0.920	0.914	0.938	0.925	0.928	0.939	0.920	0.929	0.922	0.958
	SINFL	–	0.866	0.865	0.884	0.884	0.906	0.886	0.890	0.900	0.882	0.898
	NCUT	–	0.830	0.867	0.882	0.881	0.884	0.871	0.870	0.874	0.860	0.874
	KMEANS	–	0.823	0.859	0.890	0.879	0.891	0.889	0.885	0.876	0.882	0.885
	LPA	–	0.861	0.868	0.885	0.892	0.910	0.895	0.907	0.894	0.901	0.911
	TOPLEADER	–	0.864	0.889	0.927	0.900	0.904	0.905	0.900	0.904	0.892	0.907
entropy	CATS-A	3.511	3.879	3.703	4.017	4.162	4.127	3.726	4.509	3.418	4.232	3.995
	CATE-A	3.377	2.861	1.725	1.492	3.622	3.545	3.723	3.731	1.399	3.992	2.325
	CATA-A	4.338	4.216	4.201	4.208	4.141	4.145	4.396	4.495	4.371	4.559	4.010
	CATS	3.503	3.997	4.006	4.562	4.381	4.295	4.079	4.526	3.863	4.377	4.002
	CATE	3.407	3.742	3.726	3.751	4.023	3.877	3.846	4.224	2.871	4.002	3.998
	CATA	4.487	4.251	4.619	4.822	4.544	4.371	4.560	4.509	4.543	4.679	4.057
	SINFL	–	4.722	5.122	4.824	5.165	5.018	4.984	5.194	4.946	5.028	4.909
	NCUT	–	5.199	5.126	4.972	5.014	4.972	5.003	5.003	4.994	5.331	4.958
	KMEANS	–	5.501	5.387	5.270	5.191	5.369	5.311	5.411	5.335	5.234	5.683
	LPA	–	4.733	5.127	4.804	5.147	5.038	4.998	5.214	4.936	5.031	4.898
	TOPLEADER	–	4.986	4.888	4.779	4.716	4.859	4.799	4.913	4.831	4.715	5.189

Table 4 (continued)

Measure	Methods	Information Sampling Rate σ_F										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
silhouette	CATS-A	−0.402	−0.189	−0.234	−0.209	−0.218	−0.258	−0.283	−0.289	−0.303	−0.260	−0.296
	CATE-A	−0.383	−0.128	−0.229	−0.188	−0.202	−0.258	−0.241	−0.224	−0.216	−0.252	−0.293
	CATA-A	−0.385	−0.249	−0.241	−0.255	−0.223	−0.254	−0.270	−0.351	−0.364	−0.258	−0.359
	CATS	−0.410	−0.192	−0.225	−0.209	−0.219	−0.324	−0.287	−0.284	−0.325	−0.275	−0.298
	CATE	−0.400	−0.131	−0.194	−0.184	−0.217	−0.259	−0.290	−0.260	−0.267	−0.255	−0.251
	CATA	−0.377	−0.229	−0.234	−0.187	−0.219	−0.347	−0.301	−0.273	−0.284	−0.352	−0.363
	SINFL	–	−0.455	−0.532	−0.517	−0.563	−0.561	−0.556	−0.571	−0.574	−0.561	−0.558
	NCUT	–	−0.411	−0.508	−0.519	−0.518	−0.500	−0.519	−0.512	−0.515	−0.454	−0.427
	KMEANS	–	−0.502	−0.465	−0.500	−0.544	−0.544	−0.557	−0.555	−0.552	−0.559	−0.494
	LPA	–	−0.364	−0.442	−0.424	−0.454	−0.470	−0.446	−0.473	−0.468	−0.482	−0.474
	TOPLEADER	–	−0.451	−0.415	−0.443	−0.487	−0.490	−0.491	−0.487	−0.478	−0.464	−0.482

Similar results can be obtained in Tables 3 and 5 , when Twitter is used as the new network and Foursquare is used as the well developed network.

In sum, parameter adjustment can do improve the clustering results; clustering with approximated intimacy matrix can save lots of memory space but will not harm the clustering results very much; clustering with intimacy matrix across aligned networks can do perform better than those based on intimacy matrix within one single heterogeneous network.

7.4 Parameter analysis

In the experiment, we have two other parameters to analyze, which are the number of clusters K and the ratio of anchor links existing between networks, σ_A. In this part, σ_F and σ_T are both fixed as 0.5.

To show the effects of σ_A, we fix $K = 50$ but change $\sigma_A \in \{0.1, 0.2, \cdots, 1.0\}$. The results are shown in Figure 5. As shown in Figure 5a–c, all these methods' performance will vary as σ_A changes. These methods can achieve the best performance at $\sigma_A = 0.1$ when evaluated by *ndbi*; at $\sigma_A = 0.4$ when evaluated by *entropy* and at $\sigma_A = 1.0$ when evaluated by *silhouette*.

To show the effects of parameter K on clustering results, we fix $\sigma_A = 0.8$ but change K with values in $\{10, 20, \cdots, 100\}$. The results are shown in Figure 6. As shown in Figure 6a–c, different methods can achieve the best performance at different Ks when evaluated by different metrics. For example, in Figure 6a when evaluated by *ndbi*, CATE-A can perform the best at $K = 70$, but CATE performs the best at $K = 90$. In Figure 6b, CATE-A performs the best at $K = 10$ when evaluated by *entropy* and in Figure 6c under the evaluation of *silhouette*, CATE-A can achieve the best performance at $K = 60$.

7.5 Case study

We show a case study to demonstrate the effectiveness of the proposed method CAT in community detection by introducing information from developed network.

In Figure 7, we show a case of three real-world users who have both Twitter and Foursquare accounts. To protect their privacy, we only list their first names. These users are not completely connected in Foursquare, but they follow each other in Twitter, as shown in Figure 7a. By considering this we can complete their social network in Foursquare. In Figure 7b, we show the spatial distribution of different users on both networks. We can see though the spatial distributions of the same user are similar, Twitter can still provide more information. For example, according to Emily's Foursquare check-ins, we may think she is in west coast, but by combining locations in Twitter, we prove that she lives in Chicago. In Figure 7c, we show some frequently used words by the users, and Twitter network can also provide extra information. For example, it seems that Peter does not use Foursquare frequently according to his check-in and text, and it is hard to classify him into a group. While using Twitter information, we discover that he lives in Chicago and likes running and other sports. The information differences between two networks are because Foursquare is a location-based service and many people check in at one place when traveling or first time being there, while Twitter records users' every day life better.

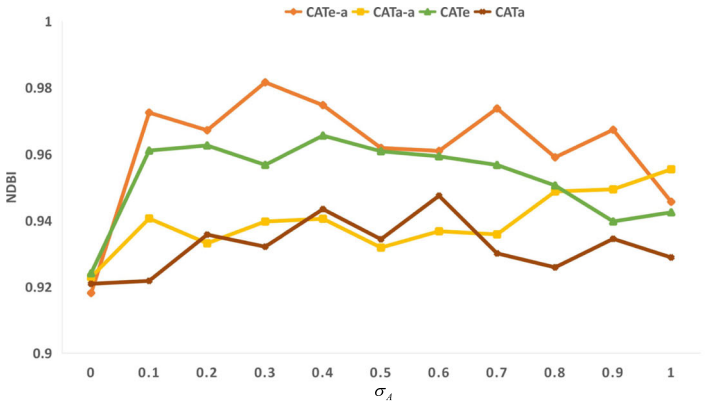
If we only use information in Foursquare, these three users belong to different communities. But the ground truth is that they all live in Chicago and belong to the same running club. After introducing Twitter information, our method CAT obtained the result that they are in the same community.

Table 5 Clustering result of Twitter Network

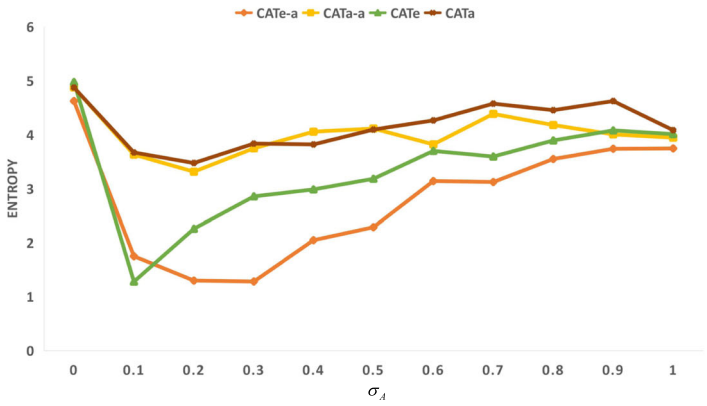
Measure	Methods	Degrees of Newness (remaining information rate) σ_T										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
ndbi	CATS-A	0.971	0.944	0.956	0.963	0.929	0.951	0.947	0.956	0.968	0.953	0.960
	CATE-A	0.973	0.951	0.970	0.983	0.961	0.957	0.966	0.959	0.981	0.957	0.986
	CATA-A	0.945	0.928	0.920	0.916	0.939	0.954	0.941	0.929	0.945	0.923	0.951
	CATS	0.959	0.934	0.937	0.935	0.926	0.931	0.940	0.939	0.945	0.931	0.948
	CATE	0.971	0.946	0.945	0.940	0.938	0.958	0.952	0.940	0.957	0.939	0.968
	CATA	0.947	0.924	0.914	0.938	0.925	0.928	0.939	0.950	0.929	0.925	0.958
	SINFL	–	0.866	0.865	0.884	0.884	0.906	0.886	0.890	0.900	0.882	0.898
	NCUT	–	0.890	0.897	0.892	0.901	0.904	0.902	0.910	0.914	0.950	0.948
	KMEANS	–	0.873	0.909	0.880	0.889	0.891	0.899	0.905	0.906	0.902	0.915
	LPA	–	0.851	0.858	0.875	0.882	0.900	0.885	0.907	0.884	0.891	0.901
	TOPLEADER	–	0.884	0.889	0.897	0.910	0.894	0.905	0.910	0.894	0.882	0.907
entropy	CATS-A	3.289	3.561	2.894	2.365	3.718	3.892	4.023	4.107	3.628	3.992	3.973
	CATE-A	3.007	2.861	1.753	1.492	3.642	3.545	3.723	3.731	2.399	2.994	2.326
	CATA-A	4.507	4.271	4.639	4.842	4.164	4.191	4.580	4.509	4.563	4.699	4.077
	CATS	3.570	3.981	4.011	4.035	4.218	4.106	4.225	4.478	4.420	4.332	4.193
	CATE	3.377	3.742	3.726	3.751	4.023	3.877	3.846	4.224	3.871	4.002	3.998
	CATA	4.358	4.236	4.221	4.228	4.507	4.465	4.416	4.515	4.391	4.579	4.230
	SINFL	–	5.172	5.572	5.274	5.615	5.468	5.434	5.644	5.396	5.478	5.359
	NCUT	–	5.699	5.626	5.472	5.514	5.472	5.503	5.503	5.494	5.831	5.458
	KMEANS	–	5.001	5.887	5.770	5.691	5.869	5.811	5.911	5.835	5.734	6.183
	LPA	–	5.083	5.477	5.154	5.497	5.388	5.348	5.564	5.286	5.381	5.248
	TOPLEADER	–	5.286	5.188	5.079	5.016	5.159	5.099	5.213	5.131	5.015	5.489

Table 5 (continued)

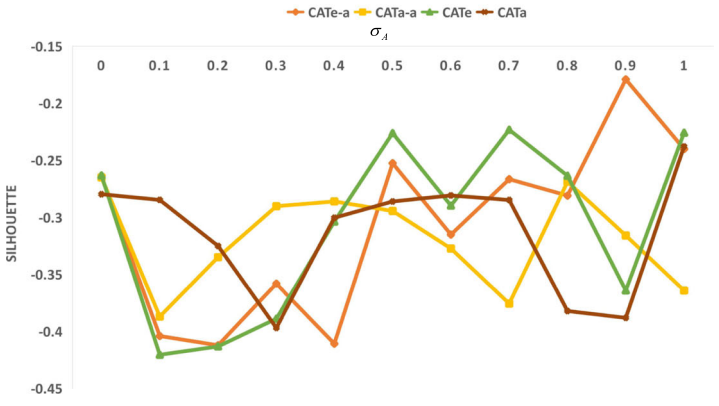
Measure	Methods	Degrees of Newness (remaining information rate) σ_T										
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
silhouette	CATS-A	−0.304	−0.146	−0.208	−0.217	−0.163	−0.169	−0.212	−0.187	−0.179	−0.175	−0.213
	CATE-A	−0.202	−0.128	−0.169	−0.108	−0.144	−0.118	−0.149	−0.150	−0.143	−0.167	−0.174
	CATA-A	−0.305	−0.149	−0.144	−0.175	−0.146	−0.174	−0.190	−0.191	−0.184	−0.168	−0.229
	CATS	−0.311	−0.158	−0.199	−0.238	−0.195	−0.241	−0.227	−0.183	−0.182	−0.209	−0.236
	CATE	−0.302	−0.133	−0.194	−0.184	−0.157	−0.158	−0.202	−0.160	−0.177	−0.185	−0.193
	CATA	−0.312	−0.159	−0.201	−0.277	−0.239	−0.267	−0.215	−0.193	−0.197	−0.272	−0.253
	SINFL	–	−0.155	−0.232	−0.217	−0.263	−0.261	−0.256	−0.271	−0.274	−0.261	−0.358
	NCUT	–	−0.311	−0.308	−0.319	−0.318	−0.300	−0.319	−0.312	−0.315	−0.254	−0.237
	KMEANS	–	−0.262	−0.225	−0.260	−0.304	−0.304	−0.317	−0.315	−0.312	−0.319	−0.254
	LPA	–	−0.194	−0.242	−0.224	−0.254	−0.270	−0.246	−0.273	−0.268	−0.282	−0.343
	TOPLEADER	–	−0.251	−0.215	−0.243	−0.287	−0.290	−0.291	−0.327	−0.328	−0.324	−0.285



(a) evaluated by *ndbi*

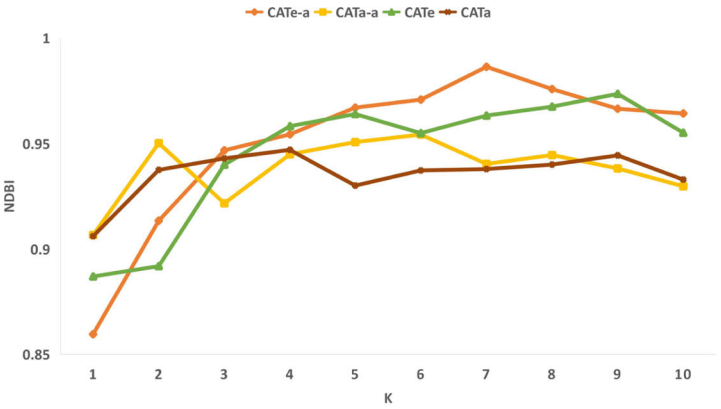


(b) evaluated by *entropy*

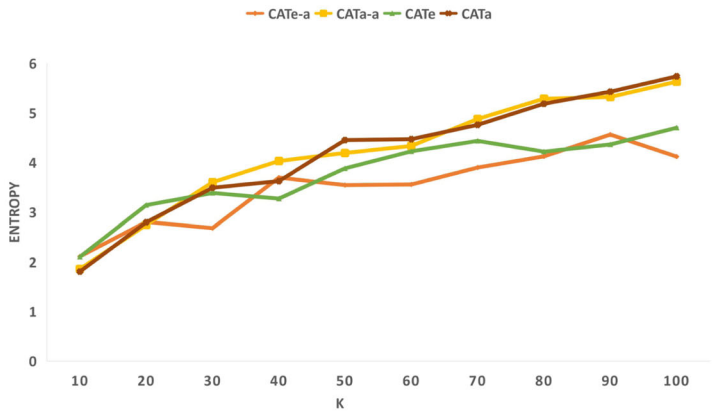


(c) evaluated by *silhouette*

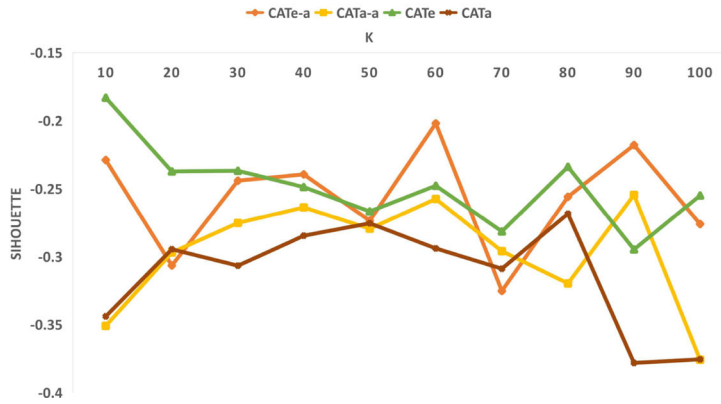
Figure 5 Experiment results with different σ_A



(a) evaluated by *ndbi*



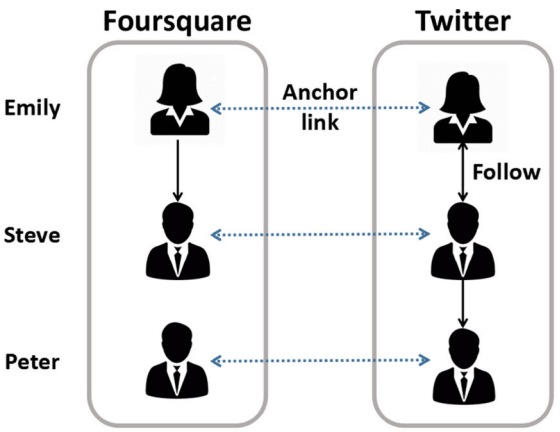
(b) evaluated by *entropy*



(c) evaluated by *silhouette*

Figure 6 Experiment results with different K

Figure 7 Case Study: three real-world users with their social, spatial and text distributions



(a) social

(b) map

	Foursquare	Twitter
Emily	happy(5), kids(4), clean(2)	run(30), enjoy(21), art(8)
Steve	hotel(9), nice(5), weekend(2)	friends(24), coffee(18), gym(13)
Peter	awesome(2), hangout(1)	amazing(9), win(5), run(3)

(c) text

8 Related work

Clustering aims at grouping similar objects in the same cluster and many different clustering methods have also been proposed.The hierarchy clustering algorithms ([4, 6] etc.) form communities in a multi-level structure progressively on the basis of the original graph. This process falls into two types, i.e. agglomeration or division, depending on their construction order of the hierarchy structure. Another type of clustering methods is partition-based methods, which include K-means for instances with numerical attributes [8]. Other clustering methods include density-based clustering methods [1] and fuzzy clustering [7]. Meanwhile, according to the manner that the similarity measure is calculated, the hierarchical clustering methods can be further divided into single-link clustering [34], complete-link clustering [11] and average-link clustering [43].

Clustering has also been widely used to detect communities in networks [41]. Direct partitioning methods separate the entire network into disjoint communities [28, 29]. Clique percolation methods assume communities are constructed by multiple adjacent cliques [13, 24]. Shi et al. introduce the concept of normalized cut and discover that the eigenvectors of the Laplacian matrix provide a solution to the normalized cut objective function [32]. Meila et al. propose to use random to solve the spectral clustering problem [20]. Chakrabarti et al. propose a information theoretic based clustering method in [3].

In recent years, many community detection works have been done on heterogeneous online social networks. Zhou et al. [53] propose to do graph clustering with relational and attribute information simultaneously. Zhou et al. [54] propose a social influence based clustering method for heterogeneous information networks. Some other works have also been done on clustering with incomplete data. Sun et al. [35] propose to study the clustering problem with complete link information but incomplete attribute information. Lin et al. [17] try to detect the communities in networks with incomplete relational information but complete attribute information.

Multiple aligned heterogeneous networks first studied by Kong et al. [12] have become a hot research topic in recent years. Kong et al. [12, 48] are the first to propose the concept of "anchor link", "aligned heterogeneous networks" and study the anchor link prediction problem across aligned networks. Zhang et al. [46, 47, 50, 52] are the first to study link prediction problem for new users with information transferred from other aligned source networks via anchor links. Zhang and Jin et al. [9, 49, 51] also propose to study the community detection problems across aligned networks, where information from all these aligned networks can be transferred to prune and refine the community structures of each network mutually. In addition, Zhan et al. introduce the cross-aligned-network information diffusion problem in [45], where multiple diffusion channels are extracted based on various types of intra and inter network meta paths.

9 Conclusion

In this paper, we have studied the community detection problems for new networks. A novel community detection method, CAT, has been proposed to solve the problem. CAT can calculate the intimacy matrix among users across aligned attribute augmented heterogeneous networks with efficient information propagation model. CAT can handle the network heterogeneity and difference problems very well with micro-level and macro-level controls, whose parameters can be adjusted automatically. Extensive experiments have been done on

real-world partially aligned networks and the results demonstrate effectiveness of CAT in address the new network community detection problem.

Acknowledgments This work is supported in part by NSF through grants IIS-1526499, and CNS-1626432, and NSFC 61672313. It is also supported by the National Key R&D Program of China (Grant No. 2016YFB1001102) and NSFC (Grant No.61375069, 61403156, 61502227) and the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing University.

Appendix

A Proof of Lemma 2

Proof The Lemma can be proved by induction on k [27] as follows:
Base Case: When $k = 1$, let $\mathbf{p}_i$ and λ_i be the i_{th} eigenvector and eigenvalue of matrix $\mathbf{Q}$ respectively, where $\mathbf{Q}\mathbf{p}_i = \lambda_i \mathbf{p}_i$. Organizing all the eigenvectors and eigenvalues of $\mathbf{Q}$ in matrix $\mathbf{P}$ and Λ, we can have $\mathbf{Q}^1\mathbf{P} = \mathbf{P}\Lambda^1$.

Inductive Assumption: When $k = m, m \geq 1$, let's assume the lemma holds when $k = m, m \geq 1$. In other words, the following equation holds:

$$\mathbf{Q}^m\mathbf{P} = \mathbf{P}\Lambda^m.$$

Induction: When $k = m + 1, m \geq 1$,

$$\mathbf{Q}^{(m+1)}\mathbf{P} = \mathbf{Q}\mathbf{Q}^m\mathbf{P} = \mathbf{Q}\mathbf{P}\Lambda^m = \mathbf{P}\Lambda\Lambda^m = \mathbf{P}\Lambda^{(m+1)}.$$

In sum, the lemma holds for $k \geq 1$. □

References

1. Banfield, J., Raftery, A.: Model-based Gaussian and non-Gaussian clustering Biometrics (1993)
2. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput. **15**(6), 1373–1396 (2003)
3. Chakrabarti, D.: Autopart: Parameter-Free Graph Partitioning and Outlier Detection. PKDD (2004)
4. Cimiano, P., Hotho, A., Staab, S.: Comparing Conceptual, Divise and Agglomerative Clustering for Learning Taxonomies from Text. ECAI (2004)
5. Gács, P., Lovász, L.: Complexity of algorithms Lecture Notes (1999)
6. Hastie, T., Tibshirani, R., Friedman, J.: Hierarchical Clustering. The Elements of Statistical Learning (2009)
7. Hopner, F., Hoppner, F., Klawonn, F.: Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition (1999)
8. Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values Data Mining Knowledge Discovery (1998)
9. Jin, S., Zhang, J., Yu, P., Yang, S., Li, A.: Synergistic partitioning in multiple large scale social networks. IEEE BigData (2014)
10. Khorasgani, R.R., Chen, J., Zaïane. O.R.: Top Leaders Community Detection Approach in Information Networks. 4Th SNA-KDD Workshop on Social Network Mining and Analysis, Washington DC. Citeseer (2010)
11. King, B.: Step-Wise Clustering procedures journal of the american statistical association (1967)
12. Kong, X., Zhang, J., Yu, P.: Inferring Anchor Links across Multiple Heterogeneous Social Networks. CIKM (2013)
13. Kumpula, J.M., Kivelä, M., Kaski, K., Saramäki, J.: Sequential algorithm for fast clique percolation. Phys. Rev. E **78**(2), 026109 (2008)
14. Kwak, H., Lee, C., Park, H., Moon, S.: What is twitter, a social network or a news media? WWW (2010)

15. Leskovec, J., Lang, K., Dasgupta, A., Mahoney, M.: Statistical Properties of Community Structure in Large Social and Information Networks. WWW (2008)
16. Lin, C., Cho, Y., Hwang, W., Pei, P., Zhang, A.: Clustering Methods in a Protein-Protein Interaction Network (2007)
17. Lin, W., Kong, X., Yu, P., Wu, Q., Jia, Y., Li, C.: Community detection in incomplete information networks. WWW (2012)
18. Liu, Y., Li, Z., Xiong, H., Gao, X., Wu, J.: Understanding of Internal Clustering Validation Measures. ICDM (2010)
19. Luxburg, U.V.: A tutorial on spectral clustering. Stat. Comput. **17**(4), 395–416 (2007)
20. Meila, M., Shi, J.: A Random Walks View of Spectral Segmentation. AISTATS (2001)
21. Mishra, N., Schreiber, R., Stanton, I., Tarjan, R.: Clustering Social Networks. WAW (2007)
22. Ng, A.Y., Jordan, M.I., Weiss, Y., et al.: On spectral clustering: Analysis and an algorithm. Adv. Neural Inf. Proces. Syst. **2**, 849–856 (2002)
23. Noulas, A., Scellato, S., Mascolo, C., Pontil, M.: An Empirical Study of Geographic User Activity Patterns in Foursquare. ICWSM (2011)
24. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. Nature **435**(7043), 814–818 (2005)
25. Pan, S., Yang, Q.: A survey on transfer learning TKDE (2010)
26. Panigrahy, R., Najork, M., Xie, Y.: How User Behavior is Related to Social Affinity. WSDM (2012)
27. Petersen, P.: Linear algebra (2012)
28. Prat-Pérez, A., Dominguez-Sal, D., Brunat, J.M., Larriba-Pey, J.-L.: Shaping Communities out of Triangles. Proceedings of the 21St ACM International Conference on Information and Knowledge Management, Pages 1677–1681. 1 (2012)
29. Prat-Pérez, A., Dominguez-Sal, D., Larriba-Pey, J.-L.: High Quality, Scalable and Parallel Community Detection for Large Real Graphs. Proceedings of the 23Rd International Conference on World Wide Web, pp. 225–236 (2014)
30. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. Phys. Rev. E **76**(3), 036106 (2007)
31. Richardson, M., Domingos, P.: Mining Knowledge-Sharing Sites for Viral Marketing. KDD (2002)
32. Shi, J., Malik, J.: NorMalized cuts and image segmentation TPAMI (2000)
33. Shi, J., Malik, J.: NorMalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
34. Sneath, P., Sokal, R.: Numerical taxonomy the principles and practice of numerical classification (1973)
35. Sun, Y., Aggarwal, C., Han, J.: Relation strength-aware clustering of heterogeneous information networks with incomplete attributes VLDB (2012)
36. Tang, J., Gao, H., Hu, X., Liu, H.: Exploiting Homophily Effect for Trust Prediction. WSDM (2013)
37. Tobias, J., Planqué, R., Cram, D., Seddon, N.: Species interactions and the structure of complex communication networks. PNAS (2014)
38. Trusov, M., Bodapati, A., Bucklin, R.: Determining influential users in internet social networks journal of marketing research (2010)
39. van Wijk, B., Stam, C., Daffertshofer, A.: Comparing brain networks of different size and connectivity density using graph theory PLos ONE (2010)
40. Wang, L., Lou, T., Tang, J., Hopcroft, J.: Detecting Community Kernels in Large Social Networks. ICDM (2011)
41. Wang, M., Wang, C., Jeffrey, X.Y., Zhang, J.: Community detection in social networks: an in-depth benchmarking study with a procedure-oriented framework. Proc. VLDB Endowment **8**(10), 998–1009 (2015)
42. Wang, S., Zhang, Z., Li, J.: A scalable cur matrix decomposition algorithm: Lower time complexity and tighter bound. CoRR (2012)
43. Ward, J.: Hierarchical grouping to optimize an objective function Journal of the American Statistical Association (1963)
44. Watts, D., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature (1998)
45. Zhan, Q., Zhang, S., Wang, J., Yu, P., Xie, J.: Influence Maximization across Partially Aligned Heterogenous Social Networks. PAKDD (2015)
46. Zhang, J., Kong, X., Yu, P.: Predicting social links for new users across aligned heterogeneous social networks. ICDM (2013)
47. Zhang, J., Kong, X., Yu, P.: Transferring Heterogeneous Links across Location-Based Social Networks. WSDM (2014)
48. Zhang, J., Shao, W., Wang, S., Kong, X., Yu, P.: Pna: Partial network alignment with generic stable matching. IEEE IRI (2015)

49. Zhang, J., Yu, P.: Community Detection for Emerging Networks. SDM (2015)
50. Zhang, J., Yu, P.: Integrated anchor and social link predictions across partially aligned social networks. IJCAI (2015)
51. Zhang, J., Yu, P.: Mcd: Mutual Clustering across Multiple Heterogeneous Networks. IEEE Bigdata Congress (2015)
52. Zhang, J., Yu, P., Zhou, Z.: Meta-Path Based Multi-Network Collective Link Prediction. KDD (2014)
53. Zhou, Y., Cheng, H., Yu, J.: Graph clustering based on structural/attribute similarities VLDB (2009)
54. Zhou, Y., Liu, L.: Social Influence Based Clustering of Heterogeneous Information Networks. KDD (2013)