Fusion Learning on Heterogeneous Social Networks

Jiawei Zhang

Abstract Looking from a global perspective, the landscape of online social networks is highly fragmented. A large number of online social networks have appeared, which can provide the users with various types of services. Generally, the information available in the these online social networks is of diverse categories, which can be represented as heterogeneous information networks (HIN) formally. Meanwhile, in such an age of online social media, users usually participate in multiple online social networks simultaneously to enjoy more social networks services, who can act as bridges connecting different networks together. Formally, the online social networks sharing common users are named as the **aligned social networks**, and these shared users who act like anchors aligning the networks are called the **anchor users**. The heterogeneous information generated by users' social activities in the **multiple aligned social networks** provides social network practitioners and researchers with the opportunities to study individual user's social behaviors across multiple social platforms simultaneously.

1 Network Alignment

1.1 Overview

Heterogeneous information networks (HIN) is a very general network representation in the real world and lots of network structured data can be represented as HINs formally, like collaboration networks, online social networks, knowledge base. Meta path first proposed by Sun et al. for heterogeneous information networks in [36] is a powerful tool, which can be applied in link prediction problems [34, 35], clus-

Jiawei Zhang
IFM Lab, Department of Computer Science, Florida State University, FL, USA.
e-mail: jiawei@ifmlab.org

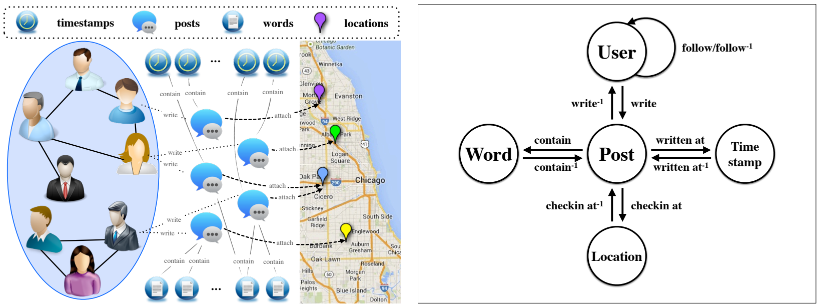
tering problems [36, 33], searching and ranking problems [38, 17] as well as collective classification problem [12] in HINs. However, most of these applications are within one single network only, meta path extracted from which are called the intra-network meta path.

Meanwhile, to enjoy the social network services from multiple online social networks simultaneously, users nowadays are usually involved in multiple online social networks at the same time. Formally, the online social networks sharing common users are named as the **aligned social networks**, and these shared users who act like anchors aligning the networks are called the **anchor users**. Social activity analysis across **aligned social networks** has become a hot research topic in recent years and many pioneer works have been done on this topic. Zhang et al. propose to study the network alignment problem between pairwise fully aligned networks [13], pairwise partially aligned networks [45, 47, 50] and multiple partially aligned networks [49].

Based on the aligned networks, various kinds of application problems have been studied across multiple social platforms, including friend recommendation and social link prediction for new users[43] and emerging networks [44, 52, 47], location recommendation [44], community detection for emerging networks [46] and synergistic clustering across networks [9, 48, 29], information diffusion [40, 41], viral marketing [40], and tipping user identification [41]. To handle the heterogeneous information available across the aligned social networks, the meta path concept is firstly extended to inter-network scenario [52, 45] and applied to address various synergistic knowledge discovery problems across partially aligned heterogeneous social networks, which include network alignment [45], link recommendation [52], community detection [48] and information diffusion [40, 41].

Network alignment problem has been well studied in bioinformatics, e.g., protein-protein interaction (PPI) network alignment [10, 31, 32, 15, 11, 18]. Most network alignment approaches focus on finding approximate isomorphism between two graphs [32, 15, 11]. Because of the intractability of the problem, existing methods usually rely on practical heuristics to solve the problem [11, 18]. Meanwhile, in recent years, some works have been done on aligning social networks [13, 14, 23]. Various network alignment models have been proposed to address the problem, which include the supervised classification based network alignment methods [13, 45], PU (positive and unlabeled) classification based method [47], and unsupervised matrix estimation based methods [49, 50].

In this chapter, we will take heterogeneous social network as an example and introduce the network alignment problem and UNICOAT model studied in [50]. In the **network alignment** problem, we aim at identifying the common users' accounts (i.e., the anchor links) across different social platforms based on the heterogeneous information available in the networks, which includes both the network structure information and various types of attribute information.



(a) HIN (b) Network Schema

Fig. 1 An example of HIN and the corresponding network schema.

1.2 Terminology Definition and Social Meta Path

Before introducing the proposed framework for the network alignment problem, we will first introduce a set of terminologies that will be used both in this section and throughout this chapter, including **heterogeneous information networks**, **multiple aligned social networks**, **anchor links** as well as the **intra-network meta path** and **inter-network meta path**. A set of intra-network and inter-network meta paths will also be introduced, whose notations, representation and physical meanings will be illustrated as follows.

1.2.1 Terminology Definition

As shown in Fig. 1(a), online social networks usually contain heterogeneous information involving different types of nodes, e.g., users, posts, words, timestamps and location checkins, as well as complex links among the nodes, e.g., friendship links among users, write links between users and posts, and the contain/attach links between posts and words, timestamps and checkins. Formally, such a kind of online social network can be represented as the **heterogeneous information networks**.

Definition 1. (**Heterogeneous Information Networks**): A **heterogeneous information network** can be represented as $G = (\mathscr{V}, \mathscr{E})$, where the nodes in set $\mathscr{V} = \bigcup_i \mathscr{V}_i$ and the links in set $\mathscr{E} = \bigcup_i \mathscr{E}_i$ are of different categories respectively.

Users nowadays are usually involved in multiple online social networks simultaneously to enjoy more social network services. Formally, the online social networks sharing common users can be defined as the **multiple aligned social networks** [13], which are connected by the **anchor links** [43] between the accounts of shared users, i.e., the **anchor users** [52].

Definition 2. (**Multiple Aligned Social Networks**): The **multiple aligned social networks** can be represented as $\mathscr{G} = (\{G^i\}_i, \{\mathscr{A}^{(i,j)}\}_{i,j})$, where $G^i = (\mathscr{V}^i, \mathscr{E}^i)$ denotes the i_{th} **heterogeneous information network** and $\mathscr{A}^{(i,j)}$ represents the set of undirected anchor links between networks G^i and G^j.

Definition 3. (**Anchor Link**): Between networks G^i and G^j, the set of undirected anchor links $\mathscr{A}^{(i,j)}$ can be represented as $\mathscr{A}^{(i,j)} = \{(u^i_m, v^j_n) | u^i_m \in \mathscr{U}^i, v^j_n \in \mathscr{U}^i, u^i_m$ and v^j_n are the accounts of the same user$\}$, where $\mathscr{U}^i \subset \mathscr{V}^i$ and $\mathscr{U}^j \subset \mathscr{V}^j$ are the user node sets in networks G^i and G^j respectively.

One way to model the heterogeneous information available across the **multiple aligned social networks** is **meta path** [33, 52, 48], which abstracts the connections among the different categories of nodes as sequences of **link types** connected by the **node types**. For instance, given the social network with its schema shown in Fig. 1, a summary of the intra-network social meta paths extracted from the network is provided in Table 1.

Definition 4. (**Intra-Network Meta Path**): Given a **heterogeneous information network** $G^i = (\mathscr{V}^i, \mathscr{E}^i)$, we can represents its **networks schema** as $S(G^i) = (\mathscr{T}^i, \mathscr{R}^i)$, where $\mathscr{T}^i$ denotes the types of nodes in $\mathscr{V}^i$ and $\mathscr{R}^i$ denotes the types of links in $\mathscr{E}^i$. Formally, based on the **network schema**, we can define the **meta path** as a sequence $P: T^i_1 \xrightarrow{R^i_1} T^i_2 \xrightarrow{R^i_2} \cdots \xrightarrow{R^i_m} T^i_{m+1}$, where $T^i_m \in \mathscr{T}^i$ and $R^i_n \in \mathscr{R}^i$ are the node and link types available in network G^i respectively.

Besides the **intra-network meta paths**, via the anchor links and other shared information entities, nodes across different networks can also get connected by the **inter-network meta paths**.

Definition 5. (**Inter-Network Meta Path**): Given a meta path P consisting of sequences of link types, P is an **inter-network meta path** between networks G^i and G^j iff P involves the node types and link types from the schema of both network G^i and network G^j.

The simplest **inter-network meta path** between networks G^i and G^j will be the **anchor meta path** [45, 52] involving the user node types from G^i and G^j and the anchor link type between G^i and G^j. Some **inter-network meta path** examples are summarized in Table 2.

1.2.2 Social Meta Paths

Meta paths can actually connect various categories of node types from the network, and those starting and ending with user node types are formally named as the **social meta paths** [48] specifically. In this chapter, we will use the Foursquare and Twitter networks as the example of **multiple aligned social networks**, which actually share a large amount of common users. As shown in Fig. 1(a), both the Foursquare and Twitter networks can be represented as a heterogeneous information network $G = (\mathscr{V}, \mathscr{E})$, where the node set $\mathscr{V} = \mathscr{U} \cup \mathscr{P} \cup \mathscr{L} \cup \mathscr{T} \cup \mathscr{W}$ involves the nodes of users, posts, locations, timestamps and words, while the link

Table 1 Summary of Intra-Network Social Meta Paths.

ID	Notation	Intra-Network Social Meta Path	Semantics
1	$\text{U} \rightarrow \text{U}$	$\text{User} \xrightarrow{follow} \text{User}$	Follow
2	$\text{U} \rightarrow \text{U} \rightarrow \text{U}$	$\text{User} \xrightarrow{follow} \text{User} \xrightarrow{follow} \text{User}$	Follower of Follower
3	$\text{U} \rightarrow \text{U} \leftarrow \text{U}$	$\text{User} \xrightarrow{follow} \text{User} \xleftarrow{follow} \text{User}$	Common Out Neighbor
4	$\text{U} \leftarrow \text{U} \rightarrow \text{U}$	$\text{User} \xleftarrow{follow} \text{User} \xrightarrow{follow} \text{User}$	Common In Neighbor
5	$\text{U} \rightarrow \text{P} \rightarrow \text{W} \leftarrow \text{P} \leftarrow \text{U}$	$\text{User} \xrightarrow{write} \text{Post} \xrightarrow{contain} \text{Word} \xleftarrow{contain} \text{Post} \xleftarrow{write} \text{User}$	Posts Containing Common Words
6	$\text{U} \rightarrow \text{P} \rightarrow \text{T} \leftarrow \text{P} \leftarrow \text{U}$	$\text{User} \xrightarrow{write} \text{Post} \xrightarrow{contain} \text{Time} \xleftarrow{contain} \text{Post} \xleftarrow{write} \text{User}$	Posts Containing Common Timestamps
7	$\text{U} \rightarrow \text{P} \rightarrow \text{L} \leftarrow \text{P} \leftarrow \text{U}$	$\text{User} \xrightarrow{write} \text{Post} \xrightarrow{attach} \text{Location} \xleftarrow{attach} \text{Post} \xleftarrow{write} \text{User}$	Posts Attaching Common Location Check-ins

Table 2 Summary of Inter-Network Social Meta Paths.

ID	Notation	Intra-Network Social Meta Path	Semantics
1	$\text{U}^i \rightarrow \text{U}^i \leftrightarrow \text{U}^j \leftarrow \text{U}^j$	$\text{User}^i \xrightarrow{follow} \text{User}^i \xleftrightarrow{Anchor} \text{User}^j \xleftarrow{follow} \text{User}^j$	Inter-Network Common Out Neighbor
2	$\text{U}^i \leftarrow \text{U}^i \leftrightarrow \text{U}^j \rightarrow \text{U}^j$	$\text{User}^i \xleftarrow{follow} \text{User}^i \xleftrightarrow{Anchor} \text{User}^j \xrightarrow{follow} \text{User}^j$	Inter-Network Common In Neighbor
3	$\text{U}^i \rightarrow \text{U}^i \leftrightarrow \text{U}^j \rightarrow \text{U}^j$	$\text{User}^i \xrightarrow{follow} \text{User}^i \xleftrightarrow{Anchor} \text{User}^j \xrightarrow{follow} \text{User}^j$	Inter-Network Common Out In Neighbor
4	$\text{U}^i \leftarrow \text{U}^i \leftrightarrow \text{U}^j \leftarrow \text{U}^j$	$\text{User}^i \xleftarrow{follow} \text{User}^i \xleftrightarrow{Anchor} \text{User}^j \xleftarrow{follow} \text{User}^j$	Inter-Network Common In Out Neighbor
5	$\text{U}^i \rightarrow \text{P}^i \rightarrow \text{L} \leftarrow \text{P}^j \leftarrow \text{U}^j$	$\text{User}^i \xrightarrow{write} \text{Post}^i \xrightarrow{checkin\ at} \text{Location} \xleftarrow{checkin\ at} \text{Post}^j \xleftarrow{write} \text{User}^j$	Inter-Network Common Location Checkins
7	$\text{U}^i \rightarrow \text{P}^i \rightarrow \text{T} \leftarrow \text{P}^j \leftarrow \text{U}^j$	$\text{User}^i \xrightarrow{write} \text{Post}^i \xrightarrow{at} \text{Time} \xleftarrow{at} \text{Post}^j \xleftarrow{write} \text{User}^j$	Inter-Network Common Timestamps
8	$\text{U}^i \rightarrow \text{P}^i \rightarrow \text{W} \leftarrow \text{P}^j \leftarrow \text{U}^j$	$\text{User}^i \xrightarrow{write} \text{Post}^i \xrightarrow{contain} \text{Word} \xleftarrow{contain} \text{Post}^j \xleftarrow{write} \text{User}^j$	Inter-Network Common Words

set $\mathscr{E} = \mathscr{E}_{u,u} \cup \mathscr{E}_{u,p} \cup \mathscr{E}_{p,l} \cup \mathscr{E}_{p,t} \cup \mathscr{E}_{p,w}$ contains the links among users, between users and posts, and those between posts and locations, timestamps, words respectively. The corresponding network schema of the HIN is shown in Fig. 1(b). Based on the network schema, a set of **intra-network social meta paths** can be extracted and defined from the network, which are shown in Table 1.

Besides the **intra-network social meta paths**, in Table 2, we also show a list of **inter-network social meta paths** connecting user node types in networks G^i and G^j respectively. These **inter-network social meta paths** connect user nodes across networks via either the **anchor links** or other common information entities, e.g., location checkins, words and timestamps.

1.3 *Cross-Network Network Alignment*

Formally, given networks $G^1, G^2, \cdots, G^n$ together with information available in them, the **network alignment** problem aims at identifying the anchor link sets $\mathscr{A}^{(1,2)}, \mathscr{A}^{(1,3)}, \cdots, \mathscr{A}^{(n-1,n)}$ between pairwise networks. The set of anchor links to be inferred between networks G^i and G^j can be represented as $\mathscr{A}^{(i,j)}$, which aligns users between networks G^i and G^j. Considering that users in different social networks are associated with both links and attribute information, the quality of the inferred anchor links $\mathscr{A}^{(i,j)}$ can be measured by the costs introduced by such mappings calculated with users' link and attribute information, i.e.,

$$cost(\mathscr{A}^{(i,j)}) = \text{ cost in links } (\mathscr{A}^{(i,j)}) + \alpha \cdot \text{ cost in attributes}(\mathscr{A}^{(i,j)}), \qquad (1)$$

where α denotes the weight of the cost obtained from the attribute information.

1.3.1 Social Structure Information based Network Alignment

Based on the social links among users in both G^i and G^j (i.e., $\mathscr{E}^i_{u,u}$ and $\mathscr{E}^j_{u,u}$ respectively), we can construct the binary **social adjacency matrices** $\mathbf{S}^i \in \mathbb{R}^{|\mathscr{U}^i| \times |\mathscr{U}^i|}$ and $\mathbf{S}^j \in \mathbb{R}^{|\mathscr{U}^j| \times |\mathscr{U}^j|}$ for networks G^i and G^j respectively. Entries in $\mathbf{S}^i$ and $\mathbf{S}^j$ (e.g., $\mathbf{S}^i(p,q)$ and $\mathbf{S}^j(l,m)$) will be assigned with value 1 iff the corresponding social links (u^i_p, u^i_q) and (u^j_l, u^j_m) exist in G^i and G^j, where $u^i_p, u^i_q \in \mathscr{U}^i$ and $u^j_l, v^j_m \in \mathscr{U}^j$ are users in networks G^i and G^j.

Via the inferred anchor links $\mathscr{A}^{(i,j)}$, users as well as their social connections can be mapped between networks G^i and G^j. We can represent the inferred anchor links $\mathscr{A}^{(i,j)}$ with binary **user transitional matrix** $\mathbf{P} \in \mathbb{R}^{|\mathscr{U}^i| \times |\mathscr{U}^j|}$, where the (i_{th}, j_{th}) entry $\mathbf{P}(p,q) = 1$ iff link $(u^i_p, u^j_q) \in \mathscr{A}^{(i,j)}$. Considering that the constraint on anchor links is **one-to-one**, each column and each row of $\mathbf{P}$ can contain at most one entry being assigned with value 1, i.e.,

$$\mathbf{P}\mathbf{1}^{|\mathscr{U}^j| \times 1} \leq \mathbf{1}^{|\mathscr{U}^i| \times 1}, \ \mathbf{P}^\top \mathbf{1}^{|\mathscr{U}^i| \times 1} \leq \mathbf{1}^{|\mathscr{U}^j| \times 1}, \tag{2}$$

where $\mathbf{P}\mathbf{1}^{|\mathscr{U}^j| \times 1}$ and $\mathbf{P}^\top \mathbf{1}^{|\mathscr{U}^i| \times 1}$ can get the sum of rows and columns of matrix $\mathbf{P}$ respectively. Eq. $\mathbf{P}\mathbf{1}^{|\mathscr{U}^j| \times 1} \leq \mathbf{1}^{|\mathscr{U}^i| \times 1}$ denotes that every entry of the left vector is no greater than the corresponding entry in the right vector.

Matrix $\mathbf{P}$ is an equivalent representation of anchor link set $\mathscr{A}^{(i,j)}$. Next, we will infer the optimal **user transitional matrix P**, from which we can obtain the optimal anchor link set $\mathscr{A}^{(i,j)}$.

The optimal anchor links are those which can minimize the inconsistency of mapped social links across networks and the cost introduced by the inferred anchor link set $\mathscr{A}^{(i,j)}$ with the link information can be represented as

$$\text{cost in link}(\mathscr{A}^{(i,j)}) = \text{cost in link}(\mathbf{P}) = \left\| \mathbf{P}^\top \mathbf{S}^i \mathbf{P} - \mathbf{S}^j \right\|_F^2, \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of the corresponding matrix and $\mathbf{P}^\top$ is the transpose of matrix $\mathbf{P}$.

1.3.2 Social Attribute Information based Network Alignment

With these different attribute information (i.e., username, temporal activity and text content), we can calculate the similarities between users across networks G^i and G^j based on the inter-network social meta paths. To measure the social closeness among users across directed heterogeneous information networks, we propose a new closeness measure named **INMP-Sim** (Inter-Network Meta Path based Similarity) as follows.

Definition 6. (**INMP-Sim**): Let $\mathscr{P}_i(x \rightsquigarrow y)$ and $\mathscr{P}_i(x \rightsquigarrow \cdot)$ be the sets of path instances of **inter-network meta paths** # i going from x to y and those going from x to other nodes in the network. The INMP-Sim of node pair (x,y) is defined as

$$\text{INMP-Sim}(x,y) = \sum_i \omega_i \left(\frac{|\mathscr{P}_i(x \rightsquigarrow y)| + |\mathscr{P}_i(y \rightsquigarrow x)|}{|\mathscr{P}_i(x \rightsquigarrow \cdot)| + |\mathscr{P}_i(y \rightsquigarrow \cdot)|} \right), \tag{4}$$

where ω_i is the weight of **inter-network meta paths** # i and $\sum_i \omega_i = 1$.

Formally, we represent such similarity matrix as $\Lambda \in \mathbb{R}^{|\mathscr{U}^i| \times |\mathscr{U}^j|}$, where entry $\Lambda(p,q)$ is the similarity between u^i_p and u^j_q. Similar users across social networks are more likely to be the same user and anchor links $\mathscr{A}_u^{(i,j)}$ that align similar users together should lead to lower cost. In this chapter, the cost function introduced by the inferred anchor links $\mathscr{A}_u^{(i,j)}$ in attribute information is represented as

$$\text{cost in attribute}(\mathscr{A}_u^{(i,j)}) = \text{cost in attribute}(\mathbf{P}) = -\left\|\mathbf{P} \circ \Lambda\right\|_1, \tag{5}$$

where $\|\cdot\|_1$ is the L_1 norm of the corresponding matrix, entry $(\mathbf{P} \circ \Lambda)(i,l)$ can be represented as $P(i,l) \cdot \Lambda(i,l)$ and $\mathbf{P} \circ \Lambda$ denotes the Hadamard product of matrices $\mathbf{P}$ and Λ.

1.3.3 Joint Objective Function for Network Alignment

Both link and attribute information is important for anchor link inference. By taking these two categories of information into consideration simultaneously, we can represent the optimal **user transitional matrix** $\mathbf{P}^*$ which can lead to the minimum cost as follows:

$$\begin{aligned}
\mathbf{P}^* &= \arg\min_{\mathbf{P}} cost(\mathscr{A}_u^{(i,j)}) && (6)\\
&= \arg\min_{\mathbf{P}} \left\| \mathbf{P}^\top \mathbf{S}^i \mathbf{P} - \mathbf{S}^j \right\|_F^2 - \alpha \cdot \left\|\mathbf{P} \circ \Lambda\right\|_1 && (7)\\
s.t. \quad & \mathbf{P} \in \{0,1\}^{|\mathscr{U}^i| \times |\mathscr{U}^j|}, && (8)\\
& \mathbf{P}\mathbf{1}^{|\mathscr{U}^j| \times 1} \leq \mathbf{1}^{|\mathscr{U}^i| \times 1}, \mathbf{P}^\top \mathbf{1}^{|\mathscr{U}^i| \times 1} \leq \mathbf{1}^{|\mathscr{U}^j| \times 1}. && (9)
\end{aligned}$$

The objective function is an constrained $0-1$ integer programming problem, which is hard to address mathematically. Many relaxation algorithms have been proposed so far. For more information about how to resolve the objective function, please refer to [50].

Table 3 Properties of the Heterogeneous 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

1.4 Experiments

To test the effectiveness of the proposed UNICOAT model, in this section, extensive experiments have been done on two real-world partially co-aligned online social networks: Foursquare and Twitter.

1.4.1 Dataset

The social networks dataset used in this chapter are Foursquare and Twitter, which are co-aligned by both users and locations shared between these two networks. These two social network datasets are crawled during November, 2012, whose statistical information is available in Table 3. More detailed descriptions and the crawling method is available in [44, 52].

1.4.2 Performance Evaluation Results

To show the advantages of UNICOAT in addressing the NETWORK ALIGNMENT problem, we compare UNICOAT with many different baseline methods. Considering that no known anchor links are available actually in the NETWORK ALIGNMENT problem, as a result, no existing supervised network alignment methods (e.g., MNA [13]) can be applied. All the comparison methods are based on unsupervised learning settings, which can be divided into 4 categories:
Co-Alignment Methods

- UNICOAT: Method UNICOAT can align two online social networks based on the shared users and locations simultaneously, which consists of two steps: (1) unsupervised potential anchor links inference; (2) co-matching of social networks to prune redundant anchor links to maintain the **one-to-one** constraint.

Bipartite Graph Alignment Methods

Table 4 Performance comparison of different methods for inferring user anchor links (UNICOAT here denotes the first step of UNICOAT only).

measure		θ				
	methods	1	2	3	4	5
AUC	UNICOAT	**0.868**	**0.831**	**0.814**	**0.804**	**0.799**
	BigAlignExt	0.813	0.779	0.759	0.752	0.749
	BigAlign	0.568	0.557	0.555	0.552	0.550
	ISOExt	0.818	0.782	0.762	0.754	0.61
	ISO	0.547	0.529	0.52	0.518	0.516
	RDD	0.531	0.530	0.523	0.514	0.508
Prec@100	UNICOAT	**0.705**	**0.688**	**0.657**	**0.640**	**0.556**
	BigAlignExt	0.587	0.507	0.472	0.434	0.327
	BigAlign	0.347	0.284	0.265	0.228	0.220
	ISOExt	0.427	0.391	0.373	0.352	0.301
	ISO	0.301	0.253	0.225	0.216	0.208
	RDD	0.234	0.228	0.207	0.172	0.127

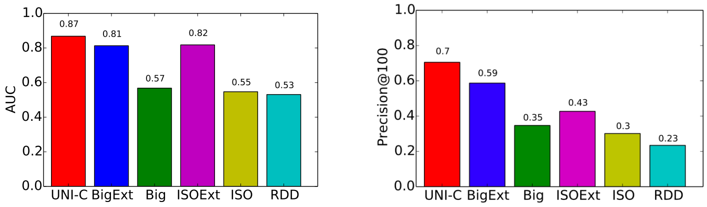
- BigAlign: Method BigAlign is a bipartite network alignment method introduced in [14], which can align two bipartite graphs (e.g., user-product bipartite graph) simultaneously with link information only.
- BigAlignExt: Method BigAlignExt is a bipartite network alignment method. BigAlignExt can align user-location bipartite networks with both location links between users and locations as well as attribute information about users and locations across networks.

Isolated Alignment Methods

- ISO: Method ISO is an unsupervised network alignment method introduced in [14]. ISO merely infers the anchor links only based on the friendship information among users.
- ISOExt: Method ISOExt is an unsupervised network alignment method, which is identical to ISO but utilizes both friendship links among users and attribute information of users.

Traditional Unsupervised Link Prediction Methods

- Relative Degree Distance based Network Alignment: RDD is the heuristics based unsupervised network alignment method introduced in [14] to fill in the initial values of the cross-network transitional matrices, e.g., **P**. For any two users/location $u_l^{(i)}$ and $u_m^{(j)}$ in networks $G^{(i)}$ and $G^{(j)}$, the relative degree distance between them can be represented as $RDD(u_l^{(i)}, u_m^{(j)}) = \left(1 + \frac{|deg(u_l^{(i)}) - deg(u_m^{(j)})|}{(deg(u_l^{(i)}) + deg(u_m^{(j)}))/2}\right)^{-1}$. High relative degree distance denotes lower confidence score of anchor link $(u_l^{(i)}, u_m^{(j)})$.



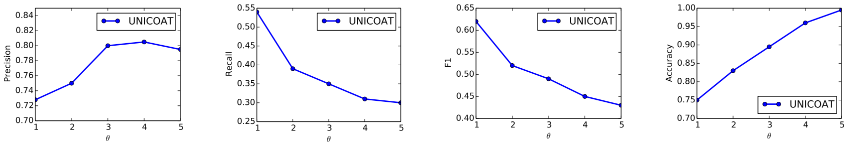
(a) AUC

(b) Prec@100

Fig. 2 Performance of methods without matching in inferring anchor links (UNICOAT here denotes the first step of UNICOAT only).

Methods UNICOAT (the first step), BIGALIGN, BIGALIGNEXT ISO, ISOEXT and RDD can output the confidence scores of potential inferred links but no labels are available, whose performance can be evaluated by metrics like AUC and Precision@100, etc. As to method UNICOAT, links selected finally in the matching are assumed to achieve confidence score 1.0 and label $+1$, while the remaining can achieve confidence score 0.0 and label -1. As a result, UNICOAT can also output the labels of potential anchor links, whose performance can be evaluated by various metrics, e.g., AUC, Precision@100, Precision, Recall, F1 and Accuracy simultaneously.

The experiment results of addressing the NETWORK ALIGNMENT problem are available in Table 4 and Fig. 2. In Fig. 2, we fix $\theta = 1$ and show the results achieved by comparison methods without matching step (i.e., methods UNICOAT (the first step), BIGALIGN, BIGALIGNEXT, ISO, ISOEXT and RDD) evaluated by AUC and Precision@100. Methods ISO and ISOEXT can only be applied to align networks via user generated information. In Fig. 2, we can observe that (1) UNICOAT performs the best among all the comparison methods in inferring anchor links evaluated by both AUC and Precision@100. For example, in Fig. 2, UNICOAT can achieve AUC score of 0.87, which is over 6% better than BIGALIGNEXT and ISOEXT, and 50% higher than the AUC score achieved by BIGALIGN, ISO and RDD. Similar performance of UNICOAT is available in other plots. It demonstrates that utilizing the heterogeneous information in the network to infer anchor links simultaneously can improve the results a lot. (2) BIGALIGNEXT and ISOEXT can achieve better performance than BIGALIGN and ISO. Recalling that methods BIGALIGNEXT and ISOEXT use both the link and attribute information, while BIGALIGN and ISO use the link information. It justifies that the attribute information of both users is helpful for inferring anchor links across networks. (3) By comparing UNICOAT with RDD (i.e., the initialization method of matrices **P** in UNICOAT), we observe that UNICOAT can outperform RDD with significant advantages. It proves the effectiveness of the proposed network co-alignment model, which can obtain better results than the initial value.



(a) Precision (b) Recall (c) F1 (d) Accuracy

Fig. 3 Performance of methods with matching in inferring anchor links (UNICOAT here includes both two steps of UNICOAT).

1.4.3 Sensitivity Analysis

In Fig. 2, parameter θ is fixed as 1. In Table 4, we further change it with values in $\{1,2,3,4,5\}$ by adding more non-anchor users into the network. Generally, with more non-anchor users, the NETWORK ALIGNMENT will become more difficult and the performance of all the methods will degrade, but UNICOAT can achieve the best performance consistently. For example, when $\theta = 5$, the AUC score achieved by UNICOAT in inferring social links is 0.799, which is 6.7%, 45%, 31%, 54.8% and 57.2% higher than that gained by BIGALIGNEXT, BIGALIGN, ISOEXT, ISO and RDD respectively. Similar observations can be obtained from the anchor links inference results evaluated by Precision@100 in Table 4.

In the previous part, we have shown the performance of methods without matching step, while anchor links inferred by which cannot meet the **one-to-one** constraint. Next, we will test the effectiveness of the matching step in pruning the non-existing anchor links and the results achieved by UNICOAT (the second step) are shown in Fig. 3. Parameter θ are assigned with values in $\{1,2,3,4,5\}$. The anchor links inferred by UNICOAT can all meet the one-to-one constraint and are of high quality. For example, when $\theta = 1$, the Precision, Recall, F1 and Accuracy achieved by UNICOAT are 0.73, 0.54, 0.62 and 0.75 respectively in inferring anchor links. As θ increases, Recall and F1 scores achieved by UNICOAT will decrease as it will be more hard to identify the real anchor links among larger number of potential ones. Meanwhile, the Precision and Accuracy of UNICOAT will increase. The potential reason can be due to the class imbalance problem. By adding more non-anchor users to the network, more non-existing anchor links (i.e., the negative class links) will be introduced and UNICOAT can achieve higher Precision and Accuracy by predicting more negative instances correctly.

2 Link Transfer across Aligned Networks

To investigate users' social activities and the propagation of information across different social platforms, several application problems will also be introduce in this chapter after aligning the networks. One important work will be the **link prediction** problems, which aims at infer potential connections among the information entities in the networks. **Link prediction** across the multiple **aligned social networks** is not an easy task, and the heterogeneity of the social networks renders the problem more challenging to solve.

2.1 Overview

Link prediction in social networks first proposed by Liben-Nowell [19] has been a hot research topic and many different methods have been proposed. Liben-Nowell [19] proposes many unsupervised link predicators to predict the social connections among users. Later, Hasan [7] proposes to predict links by using supervised learning methods. An extensive survey of link prediction works is available in [8, 6]. Most existing link prediction works are based on one single network but many researchers start to shift their attention to multiple networks. Dong et al. [4] propose to do link prediction with multiple information sources. Zhang et al. introduce the link prediction problem across aligned networks for new users [43] and emerging networks [44, 47] based on supervised classification models [43] and PU classification models [44, 47] respectively. Depending on the specific application settings, the links to be predicted are usually subject to different **cardinality constraints**, like **one-to-one** [13], **one-to-many** [51] and **many-to-many** [52]. For links with each type of the cardinality constraints, different link prediction models have been proposed already. Zhang et al. propose to unify these different link prediction tasks into a general link prediction problem, and introduce a general model for the problem [42].

In this chapter, we will briefly introduce the multi-network synergistic PU link prediction framework MLI as follows. Given a network screenshot, MLI labels the existing and non-existing social links among users as positive and unlabeled instances respectively, where the unlabeled links involve both positive and negative links at the same time. Therefore, the link prediction prediction task can be transferred into a PU learning task.

2.2 Cross-Network Link Prediction

Formally, given multiple aligned networks $\mathscr{G} = (\{G^1, G^2, \cdots, G^n\}, \{\mathscr{A}^{(1,2)}, \mathscr{A}^{(1,3)}, \cdots, \mathscr{A}^{(n-1,n)}\})$, the objective of the **cross-network link prediction** problem is to infer the potential social connections which will be formed in the near future in networks $G^1, G^2, \cdots, G^n$ respectively.

2.2.1 PU Link Prediction Feature Extraction

Meta paths introduced in the previous sections can actually cover a large number of path instances connecting users across the network. Formally, we denote that node n (or link l) is an instance of node type T (or link type R) in the network as

$n \in T$ (or $l \in R$). Identity function $I(a,A) = \begin{cases} 1, & \text{if } a \in A \\ 0, & \textit{otherwise}, \end{cases}$ can check whether node/link a is an instance of node/link type A in the network. To consider the effect of the unconnected links when extracting features for social links in the network, we formally define the **Social Meta Path based Features** to be:

Definition 7. (**Social Meta Path based Features**): For a given link (u,v), the feature extracted for it based on meta path $P = T_1 \xrightarrow{R_1} T_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{k-1}} T_k$ from the networks is defined to be the expected number of formed path instances between u and v across the networks:

$$x(u,v) = I(u,T_1)I(v,T_k) \sum_{n_1 \in \{u\}, n_2 \in T_2, \cdots, n_k \in \{v\}} \prod_{i=1}^{k-1} p(n_i, n_{i+1}) I((n_i, n_{i+1}), R_i), \tag{10}$$

where $p(n_i, n_{i+1}) = 1.0$ if $(n_i, n_{i+1}) \in E_{u,u}$ and otherwise, $p(n_i, n_{i+1})$ denotes the **formation probability** of link (n_i, n_{i+1}) to be introduced in Sec. 2.2.3.

Based on the above **social meta path based feature** definition and the extracted **intra-network** and **inter-network** meta paths, a set of features can be extracted for user pairs with the information across the aligned networks.

2.2.2 Meta Path based Feature Selection

Meanwhile, information transferred from aligned networks via the features extracted based on the **inter-network social meta path** can be helpful for improving link prediction performance in a given network but can be misleading as well, which is called the **network difference problem**. To solve the **network difference problem**, we propose to rank and select top K features from the feature vector extracted based on the **intra-network** and **inter-network social meta paths**, $\mathbf{x}$, from the multiple **partially aligned heterogeneous networks**.

Let variable $X_i \in \mathbf{x}$ be a feature extracted based on meta paths #i and variable Y be the **label**. $P(Y = y)$ denotes the **prior probability** that links in the training set having label y and $P(X_i = x)$ represents the **frequency** that feature X_i has value x. Information theory related measure **mutual information** (mi) is used as the ranking criteria:

$$mi(X_i) = \sum_x \sum_y P(X_i = x, Y = y) \log \frac{P(X_i = x, Y = y)}{P(X_i = x)P(Y = y)} \tag{11}$$

Let $\bar{\mathbf{x}}$ be the features of the top K **mi** score selected from $\mathbf{x}$. In the next subsection, we will use the selected feature vector $\bar{\mathbf{x}}$ to build a novel PU link prediction model.

2.2.3 PU Link Prediction Method

As introduced at the beginning of this section, from a given network, e.g., G, we can get two disjoint sets of links: connected (i.e., formed) links $\mathscr{P}$ and unconnected links $\mathscr{U}$. To differentiate these links, we define a new concept "**connection state**", z, to show whether a link is connected (i.e., formed) or unconnected in network G. For a given link l, if l is connected in the network, then $z(l) = +1$; otherwise, $z(l) = -1$. As a result, we can have the "**connection states**" of links in $\mathscr{P}$ and $\mathscr{U}$ to be: $z(\mathscr{P}) = \mathbf{+1}$ and $z(\mathscr{U}) = \mathbf{-1}$.

Besides the "**connection state**", links in the network can also have their own "**labels**", y, which can represent whether a link is to be formed or will never be formed in the network. For a given link l, if l has been formed or to be formed, then $y(l) = +1$; otherwise, $y(l) = -1$. Similarly, we can have the "**labels**" of links in $\mathscr{P}$ and $\mathscr{U}$ to be: $y(\mathscr{P}) = \mathbf{+1}$ but $y(\mathscr{U})$ can be either $+1$ or -1, as $\mathscr{U}$ can contain both links to be formed and links that will never be formed.

By using $\mathscr{P}$ and $\mathscr{U}$ as the positive and negative training sets, we can build a **link $\underline{\text{c}}$onnection prediction model** $\mathscr{M}_c$, which can be applied to predict whether a link exists in the original network, i.e., the **connection state** of a link. Let l be a link to be predicted, by applying $\mathscr{M}_c$ to classify l, we can get the **connection probability** of l to be:

Definition 8. (**Connection Probability**): The probability that link l's **connection states** is predicted to be **connected** (i.e., $z(l) = +1$) is formally defined as the **connection probability** of link l: $p(z(l) = +1|\bar{\mathbf{x}}(l))$.

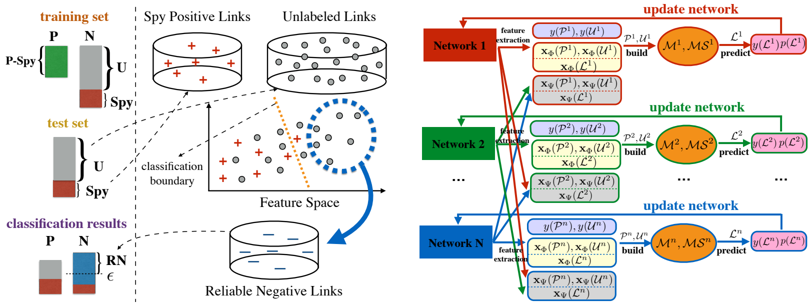
Meanwhile, if we can obtain a set of links that "will never be formed", i.e., "-1" links, from the network, which together with $\mathscr{P}$ ("+1" links) can be used to build a **link $\underline{\text{f}}$ormation prediction model**, $\mathscr{M}_f$, which can be used to get the **formation probability** of l to be:

Definition 9. (**Formation Probability**): The probability that link l's **label** is predicted to be **formed or will be formed** (i.e., $y(l) = +1$) is formally defined as the **formation probability** of link l: $p(y(l) = +1|\bar{\mathbf{x}}(l))$.

However, from the network, we have no information about "links that will never be formed" (i.e., "-1" links). As a result, the **formation probabilities** of potential links that we aim to obtain can be very challenging to calculate. Meanwhile, the correlation between link l's **connection probability** and **formation probability** has been proved in existing works [5] to be:

$$p(y(l) = +1|\bar{\mathbf{x}}(l)) \propto p(z(l) = +1|\bar{\mathbf{x}}(l)). \tag{12}$$

In other words, for links whose **connection probabilities** are low, their **formation probabilities** will be relatively low as well. This rule can be utilized to extract links which can be more likely to be the reliable "-1" links from the network. We propose to apply the the **link connection prediction model** $\mathscr{M}_c$ built with $\mathscr{P}$ and $\mathscr{U}$ to classify links in $\mathscr{U}$ to extract the **reliable negative link set**. Formally, such a


training set
P
N
P-Spy
U
Spy
test set
U
Spy
classification results
P
N
RN
ϵ
Spy Positive Links
Unlabeled Links
classification boundary
Feature Space
Reliable Negative Links
update network
Network 1
feature extraction
$y(\mathcal{P}^1), y(\mathcal{U}^1)$
$\mathbf{x}_\Phi(\mathcal{P}^1), \mathbf{x}_\Phi(\mathcal{U}^1)$
$\mathbf{x}_\Phi(\mathcal{L}^1)$
$\mathbf{x}_\Psi(\mathcal{P}^1), \mathbf{x}_\Psi(\mathcal{U}^1)$
$\mathbf{x}_\Psi(\mathcal{L}^1)$
$\mathcal{P}^1, \mathcal{U}^1$
build
$\mathcal{M}^1, \mathcal{MS}^1$
$\mathcal{L}^1$
predict
$y(\mathcal{L}^1) p(\mathcal{L}^1)$
update network
Network 2
feature extraction
$y(\mathcal{P}^2), y(\mathcal{U}^2)$
$\mathbf{x}_\Phi(\mathcal{P}^2), \mathbf{x}_\Phi(\mathcal{U}^2)$
$\mathbf{x}_\Phi(\mathcal{L}^2)$
$\mathbf{x}_\Psi(\mathcal{P}^2), \mathbf{x}_\Psi(\mathcal{U}^2)$
$\mathbf{x}_\Psi(\mathcal{L}^2)$
$\mathcal{P}^2, \mathcal{U}^2$
build
$\mathcal{M}^2, \mathcal{MS}^2$
$\mathcal{L}^2$
predict
$y(\mathcal{L}^2) p(\mathcal{L}^2)$
...
...
...
update network
Network N
feature extraction
$y(\mathcal{P}^n), y(\mathcal{U}^n)$
$\mathbf{x}_\Phi(\mathcal{P}^n), \mathbf{x}_\Phi(\mathcal{U}^n)$
$\mathbf{x}_\Phi(\mathcal{L}^n)$
$\mathbf{x}_\Psi(\mathcal{P}^n), \mathbf{x}_\Psi(\mathcal{U}^n)$
$\mathbf{x}_\Psi(\mathcal{L}^n)$
$\mathcal{P}^n, \mathcal{U}^n$
build
$\mathcal{M}^n, \mathcal{MS}^n$
$\mathcal{L}^n$
predict
$y(\mathcal{L}^n) p(\mathcal{L}^n)$

(a) PU Link Prediction (b) Multi-PU Link Prediction Framework

Fig. 4 PU Link Prediction Framework across Multiple Aligned Networks.

kind of negative link extraction method is called the **spy technique based reliable negative link extraction**. For more detailed information about method, please refer to [52].

With the extracted **reliable negative link set** $\mathscr{RN}$, we can solve the **PU link prediction** problem with **classification based link prediction methods**, where $\mathscr{P}$ and $\mathscr{RN}$ are used as the positive and negative training sets respectively. Meanwhile, when applying the built model to predict links in $\mathscr{L}^i$, the optimal labels, $\hat{\mathscr{Y}}^i$, of $\mathscr{L}^i$, should be those which can maximize the following **formation probabilities**:

$$\hat{\mathscr{Y}}^i = \arg\max_{\mathscr{Y}^i} p(y(\mathscr{L}^i) = \mathscr{Y}^i | G^1, G^2, \cdots, G^k) \tag{13}$$

$$= \arg\max_{\mathscr{Y}^i} p(y(\mathscr{L}^i) = \mathscr{Y}^i | \bar{\mathbf{x}}(\mathscr{L}^i)) \tag{14}$$

where $y(\mathscr{L}^i) = \mathscr{Y}^i$ represents that links in $\mathscr{L}^i$ have labels $\mathscr{Y}^i$.

2.2.4 Multi-Network Link Prediction Framework

Method proposed in [52] is a general link prediction framework and can be applied to predict social links in n **partially aligned networks** simultaneously. When it comes to n partially aligned network, the optimal labels of potential links $\{\mathscr{L}^1, \mathscr{L}^2, \cdots, \mathscr{L}^n\}$ of networks $G^1, G^2, \cdots, G^n$ will be:

$$\hat{\mathscr{Y}}^1, \hat{\mathscr{Y}}^2, \cdots, \hat{\mathscr{Y}}^n \tag{15}$$

$$= \arg\max_{\mathscr{Y}^1, \mathscr{Y}^2, \cdots, \mathscr{Y}^n} p(y(\mathscr{L}^1) = \mathscr{Y}^1, y(\mathscr{L}^2) = \mathscr{Y}^2, \cdots, y(\mathscr{L}^n) = \mathscr{Y}^n | G^1, G^2, \cdots, G^n) \tag{16}$$

The above target function is very complex to solve and we propose to obtain the solution by updating one variable, e.g., $\mathscr{Y}^1$, and fix other variables, e.g.,

$\mathscr{Y}^2, \cdots, \mathscr{Y}^n$, alternatively with the following equation [44]:

$$\begin{cases} (\hat{\mathscr{Y}}^1)^{(\tau)} & = \arg\max_{\mathscr{Y}^1} p(y(\mathscr{L}^1) = \mathscr{Y}^1 | G^1, \cdots, G^n, (\hat{\mathscr{Y}}^2)^{(\tau-1)}, (\hat{\mathscr{Y}}^3)^{(\tau-1)}, \cdots, (\hat{\mathscr{Y}}^n)^{(\tau-1)}) \\ (\hat{\mathscr{Y}}^2)^{(\tau)} & = \arg\max_{\mathscr{Y}^2} p(y(\mathscr{L}^2) = \mathscr{Y}^2 | G^1, \cdots, G^n, (\hat{\mathscr{Y}}^1)^{(\tau)}, (\hat{\mathscr{Y}}^3)^{(\tau-1)}, \cdots, (\hat{\mathscr{Y}}^n)^{(\tau-1)}) \\ & \cdots\cdots \\ (\hat{\mathscr{Y}}^n)^{(\tau)} & = \arg\max_{\mathscr{Y}^n} p(y(\mathscr{L}^n) = \mathscr{Y}^n | G^1, \cdots, G^n, (\hat{\mathscr{Y}}^1)^{(\tau)}, (\hat{\mathscr{Y}}^2)^{(\tau)}, \cdots, (\hat{\mathscr{Y}}^{(n-1)})^{(\tau)}) \end{cases} \tag{17}$$

The structure of the link prediction framework is shown in Fig. 4(b). When predicting social links in network G^i, we can extract features based on the **intra-network social meta path** extracted from G^i and those extracted based on the **inter-network social meta path** across G^1, G^2, $\cdots$, G^{i-1}, G^{i+1}, $\cdots$, G^n for links in $\mathscr{P}^i$, $\mathscr{U}^i$ and $\mathscr{L}^i$. Feature vectors $\mathbf{x}(\mathscr{P})$ and $\mathbf{x}(\mathscr{P})$ as well as the labels, $y(\mathscr{P})$, $y(\mathscr{U})$, of links in $\mathscr{P}$ and $\mathscr{U}$ are passed to the PU link prediction model $\mathscr{M}^i$ and the meta path selection model $\mathscr{M}\mathscr{S}^i$. The formation probabilities of links in $\mathscr{L}^i$ predicted by model $\mathscr{M}^i$ will be used to update the network by replace the weights of $\mathscr{L}^i$ with the newly predicted formation probabilities. The initial weights of these potential links in $\mathscr{L}^i$ are set as 0 (i.e., the **formation probability** of links mentioned in Definition 11). After finishing these steps on G^i, we will move to conduct similar operations on G^{i+1}. We iteratively predict links in G^1 to G^n alternatively in a sequence until the results in all of these networks converge.

2.3 Experiments

To test the effectiveness of the proposed MLI framework, in this section, extensive experiments have been done on two real-world partially co-aligned online social networks dataset introduced in the previous section.

2.3.1 Performance Evaluation Results

To show the advantages of MLI, we compare MLI with many other baseline methods, which include:

- MLI: Method MLI is the multi-network link prediction framework, which can predict social links in multiple online social networks simultaneously. The features used by MLI are extracted based on the meta paths selected from Φ and Ψ across aligned networks.
- LI: Method LI (Link Identifier) is identical to MLI except that LI predict the formation of social links in each network independently.
- SCAN: Method SCAN (Cross Aligned Network link prediction) proposed in [43, 44] is similar to MLI except that (1) SCAN predicts social links in each network independently; (2) features used by SCAN are those extracted based on meta paths Φ and Ψ_1 without meta path selection.

Table 5 Performance comparison of different methods for inferring social links for Foursquare and Twitter of different remaining information rates. The anchor link sample rate ρ^A is set as 1.0.

			Remaining information rates ρ^F of Foursquare.							
network	measure	methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Foursquare	AUC	MLI	**0.677±0.023**	**0.776±0.011**	**0.844±0.008**	**0.887±0.005**	**0.906±0.003**	**0.912±0.005**	**0.912±0.003**	**0.916±0.004**
		LI	0.573±0.019	0.68±0.023	0.806±0.01	0.853±0.004	0.866±0.003	0.874±0.007	0.881±0.003	0.878±0.005
		SCAN	0.549±0.009	0.56±0.009	0.662±0.03	0.745±0.009	0.786±0.014	0.804±0.01	0.812±0.005	0.82±0.004
		SCANT	0.5±0.083	0.503±0.007	0.613±0.012	0.739±0.008	0.764±0.013	0.787±0.007	0.8±0.006	0.81±0.007
		SCANS	0.524±0.013	0.524±0.017	0.524±0.012	0.524±0.005	0.524±0.002	0.524±0.01	0.524±0.003	0.524±0.005
	Accuracy	MLI	**0.632±0.01**	**0.692±0.007**	**0.755±0.005**	**0.769±0.004**	**0.779±0.002**	**0.798±0.006**	**0.799±0.004**	**0.797±0.005**
		LI	0.568±0.013	0.624±0.053	0.699±0.004	0.722±0.006	0.761±0.01	0.782±0.01	0.789±0.005	0.791±0.006
		SCAN	0.558±0.007	0.6±0.006	0.683±0.071	0.714±0.009	0.721±0.007	0.736±0.007	0.75±0.008	0.765±0.009
		SCANT	0.491±0.019	0.568±0.004	0.65±0.008	0.685±0.007	0.714±0.007	0.727±0.009	0.736±0.012	0.747±0.003
		SCANS	0.548±0.011	0.548±0.055	0.548±0.007	0.548±0.008	0.548±0.007	0.548±0.01	0.548±0.003	0.548±0.006
	F1	MLI	**0.644±0.01**	**0.695±0.022**	**0.722±0.013**	**0.742±0.005**	**0.761±0.005**	**0.789±0.006**	**0.783±0.005**	**0.786±0.006**
		LI	0.63±0.017	0.635±0.015	0.66±0.007	0.684±0.01	0.715±0.016	0.753±0.014	0.764±0.007	0.766±0.009
		SCAN	0.6±0.02	0.609±0.006	0.614±0.031	0.632±0.018	0.645±0.018	0.676±0.016	0.701±0.01	0.726±0.013
		SCANT	0.534±0.196	0.559±0.004	0.565±0.016	0.584±0.011	0.645±0.011	0.674±0.016	0.696±0.019	0.712±0.01
		SCANS	0.56±0.016	0.56±0.041	0.56±0.015	0.56±0.015	0.56±0.013	0.56±0.013	0.56±0.005	0.56±0.01
Twitter	AUC	MLI	**0.884±0.004**	**0.891±0.003**	**0.915±0.003**	**0.917±0.003**	**0.923±0.002**	**0.929±0.003**	**0.927±0.003**	**0.937±0.003**
		LI	0.841±0.003	0.847±0.002	0.852±0.003	0.862±0.002	0.873±0.002	0.884±0.003	0.894±0.003	0.904±0.003
		SCAN	0.801±0.003	0.814±0.002	0.819±0.003	0.817±0.002	0.819±0.002	0.823±0.003	0.831±0.002	0.837±0.003
		SCANT	0.802±0.002	0.802±0.002	0.802±0.002	0.802±0.002	0.802±0.002	0.802±0.002	0.802±0.002	0.802±0.002
		SCANS	0.508±0.002	0.543±0.002	0.584±0.003	0.631±0.001	0.653±0.002	0.666±0.003	0.673±0.003	0.686±0.003
	Accuracy	MLI	**0.92±0.003**	**0.927±0.002**	**0.927±0.003**	**0.929±0.004**	**0.93±0.003**	**0.932±0.003**	**0.936±0.003**	**0.936±0.004**
		LI	0.899±0.004	0.904±0.004	0.908±0.004	0.913±0.002	0.916±0.003	0.918±0.003	0.918±0.003	0.92±0.004
		SCAN	0.831±0.005	0.835±0.003	0.837±0.006	0.842±0.001	0.844±0.002	0.848±0.004	0.848±0.002	0.849±0.004
		SCANT	0.827±0.003	0.827±0.003	0.827±0.003	0.827±0.003	0.827±0.003	0.827±0.003	0.827±0.003	0.827±0.003
		SCANS	0.568±0.004	0.577±0.003	0.585±0.002	0.587±0.002	0.591±0.003	0.594±0.003	0.596±0.003	0.598±0.004
	F1	MLI	**0.804±0.002**	**0.808±0.002**	**0.809±0.003**	**0.811±0.003**	**0.812±0.003**	**0.818±0.003**	**0.826±0.003**	**0.826±0.004**
		LI	0.776±0.005	0.785±0.005	0.792±0.005	0.8±0.003	0.804±0.003	0.808±0.003	0.809±0.003	0.811±0.004
		SCAN	0.682±0.006	0.686±0.004	0.69±0.006	0.699±0.001	0.703±0.003	0.707±0.004	0.709±0.002	0.711±0.005
		SCANT	0.683±0.003	0.683±0.003	0.683±0.003	0.683±0.003	0.683±0.003	0.683±0.003	0.683±0.003	0.683±0.003
		SCANS	0.53±0.006	0.546±0.006	0.559±0.004	0.564±0.004	0.571±0.004	0.575±0.004	0.581±0.004	0.583±0.005

- SCAN-S: Method SCAN-S (SCAN with Source Network) proposed in [43, 44] is identical to SCAN except that the features used by SCAN-S are those extracted based on Ψ_1 without meta path selection.
- SCAN-T: Method SCAN-T (SCAN with Target Network) proposed in [43, 44]) is identical to SCAN except that the features used by SCAN-S are those extracted based on Φ without meta path selection.

The social links in both Foursquare and Twitter are used as the ground truth to evaluate the prediction results. SVM [3] with linear kernel and optimal parameters is used as the base classifier of all comparison methods. **Accuracy**, **AUC** and **F1** score are used as the evaluation metrics in the experiments.

To denote different degrees of network newness, in Table 5, we fix ρ^T as 0.8 but changes ρ^F within $\{0.1, 0.2, \cdots, 0.8\}$. Table 5 has two parts: the upper part is the link prediction results in Foursquare and the lower part is that in Twitter, as MLI is an integrated PU link prediction framework. The link prediction results in each part are evaluated by different metrics: **AUC**, **Accuracy** and **F1**. As shown in Table 5, MLI can outperform all other comparison methods consistently for $\rho^F \in \{0.1, 0.2, \cdots, 0.8\}$ in both Foursquare network and Twitter network. For example, in Foursquare when $\rho^F = 0.5$, the **AUC** achieved by MLI is about 5% better than LI, 15% better than SCAN, 19% better than SCAN-T and 73% better than SCAN-

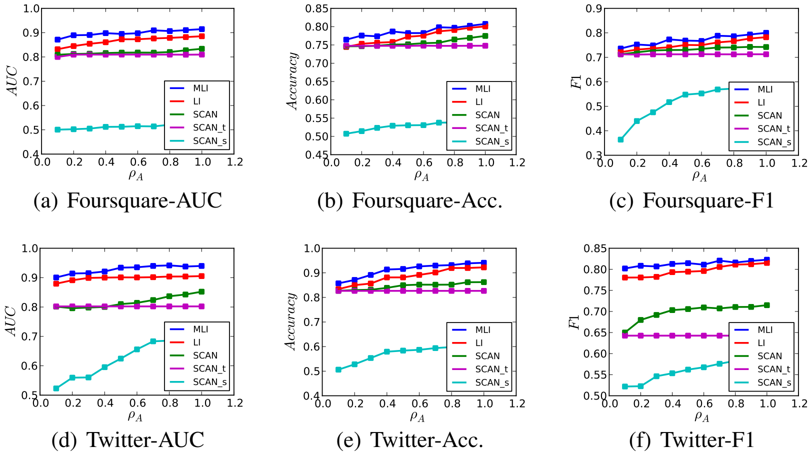
Table 6 Performance comparison of different methods for inferring social links for Foursquare and Twitter of different remaining information rates. The anchor link sample rate ρ^A is set as 1.0.

			Remaining information rates ρ^T of Twitter							
network	measure	methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
Foursquare	AUC	MLI	**0.862±0.003**	**0.867±0.004**	**0.87±0.003**	**0.873±0.005**	**0.885±0.003**	**0.891±0.003**	**0.895±0.004**	**0.916±0.004**
		LI	0.831±0.005	0.834±0.004	0.846±0.004	0.853±0.005	0.855±0.005	0.867±0.004	0.868±0.005	0.87±0.005
		SCAN	0.81±0.007	0.81±0.008	0.812±0.005	0.817±0.007	0.816±0.01	0.815±0.007	0.822±0.006	0.82±0.004
		SCANT	0.81±0.007	0.81±0.007	0.81±0.007	0.81±0.007	0.809±0.007	0.809±0.007	0.81±0.007	0.81±0.007
		SCANS	0.504±0.007	0.51±0.003	0.511±0.005	0.516±0.005	0.522±0.004	0.53±0.005	0.53±0.004	0.53±0.005
	Accuracy	MLI	**0.78±0.003**	**0.786±0.005**	**0.789±0.004**	**0.794±0.005**	**0.793±0.004**	**0.789±0.004**	**0.796±0.005**	**0.797±0.005**
		LI	0.745±0.011	0.762±0.005	0.768±0.007	0.772±0.007	0.777±0.008	0.783±0.008	0.789±0.006	0.791±0.006
		SCAN	0.749±0.007	0.754±0.006	0.754±0.007	0.757±0.006	0.758±0.007	0.761±0.008	0.763±0.009	0.765±0.009
		SCANT	0.748±0.003	0.748±0.003	0.747±0.003	0.748±0.003	0.748±0.003	0.748±0.003	0.748±0.003	0.747±0.003
		SCANS	0.692±0.011	0.717±0.008	0.725±0.008	0.746±0.008	0.741±0.006	0.746±0.004	0.75±0.007	0.758±0.006
	F1	MLI	**0.768±0.004**	**0.774±0.005**	**0.778±0.006**	**0.784±0.006**	**0.785±0.005**	**0.777±0.004**	**0.785±0.006**	**0.786±0.006**
		LI	0.721±0.02	0.734±0.01	0.734±0.012	0.736±0.012	0.744±0.012	0.755±0.011	0.764±0.01	0.766±0.009
		SCAN	0.717±0.01	0.718±0.007	0.714±0.009	0.715±0.009	0.718±0.011	0.72±0.012	0.721±0.013	0.726±0.013
		SCANT	0.713±0.01	0.712±0.01	0.712±0.01	0.713±0.01	0.713±0.01	0.712±0.01	0.713±0.01	0.712±0.01
		SCANS	0.509±0.02	0.514±0.014	0.524±0.014	0.529±0.013	0.54±0.009	0.542±0.007	0.559±0.012	0.559±0.01
Twitter	AUC	MLI	**0.837±0.004**	**0.858±0.004**	**0.905±0.005**	**0.926±0.003**	**0.924±0.002**	**0.932±0.003**	**0.934±0.002**	**0.937±0.003**
		LI	0.772±0.009	0.829±0.008	0.871±0.009	0.887±0.002	0.887±0.002	0.897±0.003	0.899±0.003	0.904±0.003
		SCAN	0.706±0.008	0.771±0.012	0.799±0.009	0.817±0.002	0.819±0.002	0.829±0.003	0.83±0.003	0.834±0.003
		SCANT	0.555±0.133	0.678±0.006	0.753±0.044	0.754±0.019	0.764±0.014	0.781±0.004	0.794±0.003	0.802±0.002
		SCANS	0.687±0.008	0.687±0.002	0.687±0.005	0.687±0.002	0.687±0.002	0.687±0.004	0.687±0.003	0.687±0.003
	Accuracy	MLI	**0.821±0.005**	**0.864±0.001**	**0.892±0.008**	**0.914±0.004**	**0.925±0.002**	**0.926±0.004**	**0.936±0.002**	**0.936±0.004**
		LI	0.706±0.002	0.834±0.011	0.877±0.003	0.898±0.005	0.912±0.001	0.92±0.004	0.924±0.002	0.92±0.004
		SCAN	0.594±0.006	0.716±0.009	0.781±0.005	0.801±0.003	0.823±0.002	0.831±0.004	0.842±0.002	0.849±0.004
		SCANT	0.547±0.062	0.645±0.038	0.723±0.048	0.786±0.004	0.8±0.002	0.815±0.005	0.824±0.002	0.827±0.003
		SCANS	0.59±0.009	0.59±0.007	0.59±0.004	0.59±0.004	0.59±0.002	0.59±0.004	0.59±0.003	0.59±0.004
	F1	MLI	**0.713±0.009**	**0.762±0.005**	**0.791±0.006**	**0.81±0.004**	**0.81±0.002**	**0.819±0.004**	**0.821±0.002**	**0.826±0.004**
		LI	0.651±0.006	0.671±0.023	0.749±0.014	0.779±0.007	0.801±0.003	0.813±0.005	0.818±0.003	0.811±0.004
		SCAN	0.6±0.017	0.633±0.023	0.657±0.013	0.684±0.004	0.703±0.004	0.714±0.005	0.716±0.002	0.711±0.005
		SCANT	0.552±0.113	0.574±0.016	0.604±0.031	0.618±0.003	0.63±0.001	0.641±0.004	0.67±0.002	0.686±0.003
		SCANS	0.575±0.025	0.575±0.016	0.575±0.005	0.575±0.006	0.575±0.004	0.575±0.004	0.575±0.003	0.575±0.005

S; the **Accuracy** achieved by MLI is about 2.3% better than LI, 8% better than SCAN, 9.1% higher than SCAN-T and over 40% higher than SCAN-S; the **F1** of MLI is 6.4% higher than LI, 18% higher than SCAN and SCAN-T and 36% higher than SCAN-S. When $\rho^F = 0.5$, the link prediction results of MLI in Twitter are also much better than all other baseline methods. For instances, in Twitter the **AUC** of MLI is 0.923 ± 0.002, which is about 6% better than LI, over 13% better than SCAN, SCAN-T and over 40% better than SCAN-S. Similar results can be obtained when evaluated by **Accuracy** and **F1**.

In Table 6, we fix $\rho^F = 0.8$ but change ρ^T with values in $\{0.1, 0.2, \cdots, 0.8\}$. Similar to the results obtained in Table 5 where ρ^F varies, MLI can beat all other methods in both Twitter and Foursquare when the degree of newness of the Twitter network changes.

MLI can perform better than LI in both Foursquare and Twitter, which shows that predicting social links in multiple networks simultaneously in MLI framework can do enhance the results in both networks; the fact that LI can beat SCAN shows that features extracted based on cross network meta paths can do transfer useful information for both anchor and non-anchor users; SCAN works better than both SCAN-T and SCAN-S denotes that link prediction with information in two networks simultaneously is better than that with information in one single network.



(a) Foursquare-AUC (b) Foursquare-Acc. (c) Foursquare-F1

(d) Twitter-AUC (e) Twitter-Acc. (f) Twitter-F1

Fig. 5 Effects of anchor link ratio ρ^A on prediction results in different networks evaluated by different metrics.

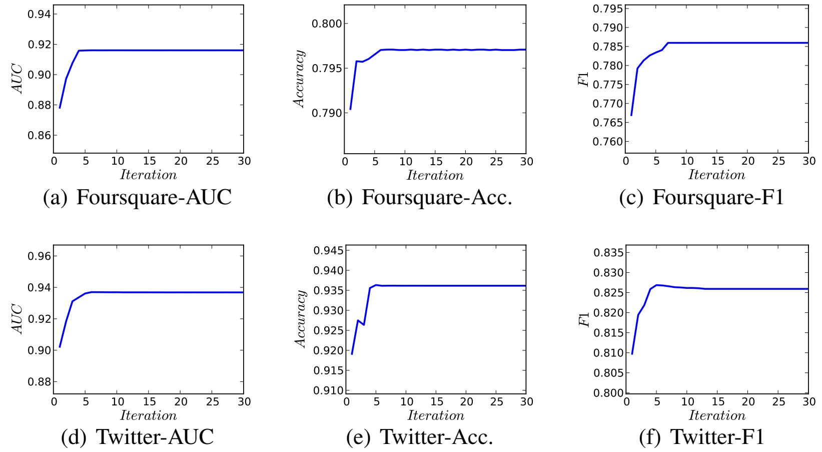
2.3.2 Parameter Analysis

An important parameter that can affect the performance of all these methods is the rate of anchor links existing across networks. In this part, we will analyze the effects of the anchor link rate, $\rho^A \in [0, 1.0]$. To exclude other parameters' interference, we fix ρ^F and ρ^T as 0.8 but change ρ^A with values in $\{0.1, 0.2, \cdots, 1.0\}$ and study the link prediction results in both Foursquare and Twitter under the evaluation of **AUC**, **Accuracy** and **F1**. The results are shown in Fig. 5.

As shown in Fig. 5, where Fig. 5(a)-5(c) are the link prediction results in Foursquare and the Fig. 5(d)-5(f) are those in Twitter, almost all the methods can perform better as ρ^A increases, except SCAN-T as it only utilizes information in the target network only. It shows that with more anchor links, MLI, LI, SCAN and SCAN-S can transfer much more information from other aligned source networks to the target network to enhance the results. In addition, MLI can work better than LI consistently as ρ^A varies, which can show the effectiveness of MLI in dealing with networks with different ratios of anchor links

2.3.3 Convergence Analysis

MLI need to predict the links in all the aligned networks alternatively and iteratively until convergence. In this part, we will analyze whether MLI can converge as this process continues. We show the link prediction results achieved by MLI in both Foursquare and Twitter under the evaluation of **AUC**, **Accuracy** and **F1** when ρ^F, ρ^T and ρ^A are all set as 0.8 in Fig. 6. Fig. 6(a)-6(c) are the results in Foursquare network from iteration 1 to iteration 30 and Fig. 6(d)-6(f) are those in Twitter net-



(a) Foursquare-AUC (b) Foursquare-Acc. (c) Foursquare-F1

(d) Twitter-AUC (e) Twitter-Acc. (f) Twitter-F1

Fig. 6 Convergence analysis in different networks under the evaluation of different metrics.

work. As shown in these figures, results achieved by MLI can converge in less than 10 iterations in both Foursquare and Twitter evaluated by all these three metrics.

3 Synergistic Network Community Detection

3.1 Overview

Clustering is a very broad research area, which includes various types of clustering problems, e.g., consensus clustering [22, 21], multi-view clustering [1, 2], multi-relational clustering [37], co-training based clustering [16], at the same time. Clustering based community detection in online social networks is a hot research topic and many different models have already been proposed to optimizing certain evaluation metrics, e.g., modularity function [26], and normalized cut [30]. A detailed survey about existing community detection works is available in [25, 24]. Meanwhile, based on the information available in multiple aligned networks, Jin [9], Zhang et al. [48] and Shao et al. [29] propose to do synergistic community detection across multiple aligned social networks. Via the anchor links, Zhang et al. also propose to transfer information from developed networks to detect social community structures in emerging networks in [46].

The goal of **cross-network community detection** is to distill relevant information from another social network to compliment knowledge directly derivable from each network to improve the clustering or community detection, while preserving the distinct characteristics of each individual network. To solve the Mutual Clustering problem, a novel community detection method, MCD, is proposed in [48].

By mapping the social network relations into a heterogeneous information, the proposed method in [48] uses the concept of social meta path to define closeness measure among users. Based on this similarity measure, the proposed method [48] can preserve the network characteristics and utilize the information in other networks to refine community structures mutually at the same time. In this section, we will introduce the mutual community detection framework proposed in [48] briefly.

3.2 Cross-Network Community Detection

Given multiple aligned networks $\mathscr{G} = (\{G^1, G^2, \cdots, G^n\}, \{\mathscr{A}^{(1,2)}, \mathscr{A}^{(1,3)}, \cdots, \mathscr{A}^{(n-1,n)}\})$, the **cross-network community detection** problem aims at detecting the community structures of networks $G^1, G^2, \cdots, G^n$ respectively.

3.2.1 Network Characteristic Preservation Clustering

Clustering each network independently can preserve each networks characteristics effectively as no information from external networks will interfere with the clustering results. Partitioning users of a certain network into several clusters will cut connections in the network and lead to some costs inevitably. Optimal clustering results can be achieved by minimizing the clustering costs.

Let $\mathbf{A}_i$ be the **adjacency matrix** corresponding to the **intra-network meta path** # i among users in the network and $\mathbf{A}_i(m,n) = k$ iff there exist k different path instances of **intra-network meta path** # i from user m to n in the network. Furthermore, the similarity score matrix among users of meta path # i can be represented as $\mathbf{S}_i = \left(\mathbf{D}_i + \bar{\mathbf{D}}_i\right)^{-1} \left(\mathbf{A}_i + \mathbf{A}_i^T\right)$, where $\mathbf{A}_i^T$ denotes the transpose of $\mathbf{A}_i$, diagonal matrices $\mathbf{D}_i$ and $\bar{\mathbf{D}}_i$ have values $\mathbf{D}_i(l,l) = \sum_m \mathbf{A}_i(l,m)$ and $\bar{\mathbf{D}}_i(l,l) = \sum_m (\mathbf{A}_i^T)(l,m)$ on their diagonals respectively. The meta path based similarity matrix of the network which can capture all possible connections among users is represented as follows:

$$\mathbf{S} = \sum_i \omega_i \mathbf{S}_i = \sum_i \omega_i \left(\left(\mathbf{D}_i + \bar{\mathbf{D}}_i\right)^{-1} \left(\mathbf{A}_i + \mathbf{A}_i^T\right) \right). \tag{18}$$

For a given network G, let $\mathscr{C} = \{U_1, U_2, \ldots, U_k\}$ be the community structures detected from G. Term $\overline{U_i} = \mathscr{U} - U_i$ is defined to be the complement of set U_i in G. Various cost measure of partition $\mathscr{C}$ can be used, e.g., **cut** and **normalized cut**:

$$cut(\mathscr{C}) = \frac{1}{2} \sum_{i=1}^{k} S(U_i, \overline{U_i}) = \frac{1}{2} \sum_{i=1}^{k} \sum_{u \in U_i, v \in \overline{U_i}} S(u,v), \tag{19}$$

$$Ncut(\mathscr{C}) = \frac{1}{2} \sum_{i=1}^{k} \frac{S(U_i, \overline{U_i})}{S(U_i, \cdot)} = \sum_{i=1}^{k} \frac{cut(U_i, \overline{U_i})}{S(U_i, \cdot)}, \tag{20}$$

where $S(u,v)$ denotes the similarity between u,v and $S(U_i,\cdot) = S(U_i,\mathscr{U}) = S(U_i,U_i) + S(U_i,\overline{U}_i)$.

For all users in $\mathscr{U}$, their clustering result can be represented in the **result confidence matrix H**, where $\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_n}]^{\mathbf{T}}$, $n = |\mathscr{U}|$, $\mathbf{h}_i = (h_{i,1}, h_{i,2}, \ldots, h_{i,k})$ and $h_{i,j}$ denotes the confidence that $u_i \in \mathscr{U}$ is in cluster $U_j \in \mathscr{C}$. The optimal $\mathbf{H}$ that can minimize the normalized-cut cost can be obtained by solving the following objective function:

$$\min_{\mathbf{H}} \ \mathrm{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}), \tag{21}$$

$$s.t. \ \ \mathbf{H}^T \mathbf{D} \mathbf{H} = \mathbf{I}. \tag{22}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{S}$, diagonal matrix $\mathbf{D}$ has $\mathbf{D}(i,i) = \sum_j \mathbf{S}(i,j)$ on its diagonal, and $\mathbf{I}$ is an identity matrix.

3.2.2 Discrepancy based Clustering of Multiple Aligned Networks

Besides the shared information due to common network construction purposes and similar network features [46], anchor users can also have unique information (e.g., social structures) across aligned networks, which can provide us with a more comprehensive knowledge about the community structures formed by these users. Meanwhile, by maximizing the consensus (i.e., minimizing the "**discrepancy**") of the clustering results about the anchor users in multiple partially aligned networks, we refine the clustering results of the anchor users with information in other aligned networks mutually. We can represent the clustering results achieved in G^i and G^j as $\mathscr{C}^i = \{U_1^i, U_2^i, \cdots, U_{k^i}^i\}$ and $\mathscr{C}^j = \{U_1^j, U_2^j, \cdots, U_{k^j}^j\}$ respectively.

Let u_p and u_q be two anchor users in the network, whose accounts in G^i and G^j are u_p^i, u_p^j, u_q^i and u_q^j respectively. If users u_p^i and u_q^i are partitioned into the same cluster in G^i but their corresponding accounts u_p^j and u_q^j are partitioned into different clusters in G^j, then it will lead to a **discrepancy** between the clustering results of u_p^i, u_p^j, u_q^i and u_q^j in aligned networks G^i and G^j.

Definition 10. (**Discrepancy**): The discrepancy between the clustering results of u_p and u_q across aligned networks G^i and G^j is defined as the difference of confidence scores of u_p and u_q being partitioned in the same cluster across aligned networks. Considering that in the clustering results, the confidence scores of u_p^i and u_q^i (u_p^j and u_q^j) being partitioned into k^i (k^j) clusters can be represented as vectors $\mathbf{h}_p^i$ and $\mathbf{h}_q^i$ ($\mathbf{h}_p^j$ and $\mathbf{h}_q^j$) respectively, while the confidences that u_p and u_q are in the same cluster in G^i and G^j can be denoted as $\mathbf{h}_p^i(\mathbf{h}_q^i)^T$ and $\mathbf{h}_p^j(\mathbf{h}_q^j)^T$. Formally, the discrepancy of the clustering results about u_p and u_q is defined to be $d_{p,q}(\mathscr{C}^i,\mathscr{C}^j) = \left(\mathbf{h}_p^i(\mathbf{h}_q^i)^T - \mathbf{h}_p^j(\mathbf{h}_q^j)^T\right)^2$ if u_p, u_q are both anchor users; and $d_{p,q}(\mathscr{C}^i,\mathscr{C}^j) = 0$ otherwise. Furthermore, the discrepancy of $\mathscr{C}^i$ and $\mathscr{C}^j$ will be:

$$d(\mathscr{C}^i, \mathscr{C}^j) = \sum_{p}^{n^i} \sum_{q}^{n^j} d_{p,q}(\mathscr{C}^i, \mathscr{C}^j), \tag{23}$$

where $n^i = |\mathscr{U}^i|$ and $n^j = |\mathscr{U}^j|$.

However, considering that $d(\mathscr{C}^i, \mathscr{C}^j)$ is highly dependent on the number of anchor users and anchor links between G^i and G^j, minimizing $d(\mathscr{C}^i, \mathscr{C}^j)$ can favor highly consented clustering results when the anchor users are abundant but have no significant effects when the anchor users are very rare. To solve this problem, we propose to minimize the **normalized discrepancy** instead.

Definition 11. (**Normalized Discrepancy**) The normalized discrepancy measure computes the differences of clustering results in two aligned networks as a fraction of the discrepancy with regard to the number of anchor users across partially aligned networks:

$$Nd(\mathscr{C}^i, \mathscr{C}^j) = \frac{d(\mathscr{C}^i, \mathscr{C}^j)}{\left(\left|A^{(i,j)}\right|\right)\left(\left|A^{(i,j)}\right| - 1\right)}. \tag{24}$$

Optimal consensus clustering results of G^i and G^j will be $\hat{\mathscr{C}}^i, \hat{\mathscr{C}}^j$:

$$\hat{\mathscr{C}}^i, \hat{\mathscr{C}}^j = \arg\min_{\mathscr{C}^i, \mathscr{C}^j} Nd(\mathscr{C}^i, \mathscr{C}^j). \tag{25}$$

Similarly, the normalized-discrepancy objective function can also be represented with the **clustering results confidence matrices $\mathbf{H}^i$** and $\mathbf{H}^j$ as well. Meanwhile, considering that the networks studied in this chapter are partially aligned, matrices $\mathbf{H}^i$ and $\mathbf{H}^j$ contain the results of both anchor users and non-anchor users, while non-anchor users should not be involved in the discrepancy calculation according to the definition of discrepancy. After pruning the non-anchor users from the confidence matrices, we can represent the pruned confidence matrices as $\bar{\mathbf{H}}^i$ and $\bar{\mathbf{H}}^j$.

Furthermore, the objective function of inferring clustering confidence matrices, which can minimize the normalized discrepancy can be represented as follows

$$\min_{\mathbf{H}^i, \mathbf{H}^j} \frac{\left\|\bar{\mathbf{H}}^i \left(\bar{\mathbf{H}}^i\right)^T - \bar{\mathbf{H}}^j \left(\bar{\mathbf{H}}^j\right)^T\right\|_F^2}{\left\|\mathbf{T}^{(i,j)}\right\|_F^2 \left(\left\|\mathbf{T}^{(i,j)}\right\|_F^2 - 1\right)}, \tag{26}$$

$$s.t. \;\; (\mathbf{H}^i)^T \mathbf{D}^i \mathbf{H}^i = \mathbf{I}, (\mathbf{H}^j)^T \mathbf{D}^j \mathbf{H}^j = \mathbf{I}. \tag{27}$$

where $\mathbf{D}^i$, $\mathbf{D}^j$ are the corresponding diagonal matrices of similarity matrices of networks G^i and G^j respectively.

3.2.3 Joint Optimization Objective Function

Taking both of these two issues into considerations, the optimal mutual clustering results $\hat{\mathscr{C}}^i$ and $\hat{\mathscr{C}}^j$ of aligned networks G^i and G^j can be achieved as follows:

$$\arg \min_{\mathscr{C}^i,\mathscr{C}^j} \alpha \cdot Ncut(\mathscr{C}^i) + \beta \cdot Ncut(\mathscr{C}^j) + \theta \cdot Nd(\mathscr{C}^i,\mathscr{C}^j) \tag{28}$$

where α, β and θ represents the weights of these terms and, for simplicity, α, β are both set as 1.

By replacing $Ncut(\mathscr{C}^i)$, $Ncut(\mathscr{C}^j)$, $Nd(\mathscr{C}^i,\mathscr{C}^j)$ with the objective equations derived above, we can rewrite the joint objective function as follows:

$$\min_{\mathbf{H}^i,\mathbf{H}^j} \alpha \cdot \mathrm{Tr}((\mathbf{H}^i)^T \mathbf{L}^i \mathbf{H}^i) + \beta \cdot \mathrm{Tr}((\mathbf{H}^j)^T \mathbf{L}^j \mathbf{H}^j) + \theta \cdot \frac{\left\| \bar{\mathbf{H}}^i \left(\bar{\mathbf{H}}^i\right)^T - \bar{\mathbf{H}}^j \left(\bar{\mathbf{H}}^j\right)^T \right\|_F^2}{\left\|\mathbf{T}^{(i,j)}\right\|_F^2 \left(\left\|\mathbf{T}^{(i,j)}\right\|_F^2 - 1 \right)}, \tag{29}$$

$$s.t. \;\; (\mathbf{H}^i)^T \mathbf{D}^i \mathbf{H}^i = \mathbf{I}, (\mathbf{H}^j)^T \mathbf{D}^j \mathbf{H}^j = \mathbf{I}, \tag{30}$$

where $\mathbf{L}^i = \mathbf{D}^i - \mathbf{S}^i$, $\mathbf{L}^j = \mathbf{D}^j - \mathbf{S}^j$ and matrices $\mathbf{S}^i$, $\mathbf{S}^j$ and $\mathbf{D}^i$, $\mathbf{D}^j$ are the similarity matrices and their corresponding diagonal matrices defined before.

The objective function is a complex optimization problem with orthogonality constraints, which can be very difficult to solve because the constraints are not only non-convex but also numerically expensive to preserve during iterations. Please refer to [48] for more information about the solution to the objective function.

3.3 Experiments

To test the performance of the MCD model in detecting the communities across multiple aligned social networks, extensive experiments have been done on the aligned social networks data set: Foursquare and Twitter. The experimental results will be illustrated as follows.

3.4 Performance Evaluation Results

The comparison methods used in the experiments can be divided into three categories,

Mutual Clustering Methods

- MCD: MCD is the mutual community detection method, which can detect the communities of multiple aligned networks with consideration of the connections and characteristics of different networks. Heterogeneous information in multiple aligned networks are applied in building MCD.

Multi-Network Clustering Methods

- SICLUS: the clustering method proposed in [39, 46] can calculate the similarity scores among users by propagating heterogeneous information across views/networks. We extend the method proposed in [39, 46] and propose SICLUS to calculate the intimacy scores among users in multiple networks simultaneously, based on which, users can be grouped into different clusters with clustering models based on intimacy matrix factorization as introduced in [46]. Heterogeneous information across networks is used to build SICLUS.

Isolated Clustering Methods, which can detect communities in each isolated network:

- NCUT: NCUT is the clustering method based on normalized cut proposed in [30]. Method NCUT can detect the communities in each social network merely based on the social connections in each network in the experiments.
- KMEANS: KMEANS is a traditional clustering method, which can be used to detect communities [28] in social networks based on the social connections only in the experiments.

The evaluation metrics applied can be divided into two categories: Quality Metrics and Consensus Metrics.

Quality Metrics: 4 widely and commonly used quality metrics are applied to measure the clustering result, e.g., $\mathscr{C} = \{U_i\}_{i=1}^{K}$, of each network.

- **normalized-dbi** [39]:

$$ndbi(\mathscr{C}) = \frac{1}{K} \sum_i \min_{j \neq i} \frac{d(c_i, c_j) + d(c_j, c_i)}{\sigma_i + \sigma_j + d(c_i, c_j) + d(c_j, c_i)}, \quad (31)$$

  where c_i is the centroid of community $U_i \in \mathscr{C}$, $d(c_i, c_j)$ denotes the distance between centroids c_i and c_j and σ_i represents the average distance between elements in U_i and centroid c_i. (Higher ndbi corresponds to better performance).
- **entropy** [39]: $H(\mathscr{C}) = -\sum_{i=1}^{K} P(i) \log P(i)$, where $P(i) = \frac{|U_i|}{\sum_{i=1}^{K} |U_i|}$. (Lower entropy corresponds to better performance).
- **density** [39]: $dens(\mathscr{C}) = \sum_{i=1}^{K} \frac{|E_i|}{|E|}$, where E and E_i are the edge sets in the network and U_i. (Higher density corresponds to better performance).
- **silhouette** [20]:

$$sil(\mathscr{C}) = \frac{1}{K} \sum_{i=1}^{K} \left(\frac{1}{|U_i|} \sum_{u \in U_i} \frac{b(u) - a(u)}{\max\{a(u), b(u)\}}\right), \quad (32)$$

  where $a(u) = \frac{1}{|U_i|-1} \sum_{v \in U_i, u \neq v} d(u, v)$ and $b(u) = \min_{j, j \neq i} \left(\frac{1}{|U_j|} \sum_{v \in U_j} d(u, v)\right)$. (Higher silhouette corresponds to better performance).

Consensus Metrics: Given the clustering results $\mathscr{C}^{(1)} = \{U_i^{(1)}\}_{i=1}^{K^{(1)}}$ and $\mathscr{C}^{(2)} = \{U_i^{(2)}\}_{i=1}^{K^{(2)}}$, the consensus metrics measuring the how similar or dissimilar the anchor users are clustered in $\mathscr{C}^{(1)}$ and $\mathscr{C}^{(2)}$ include:

- **rand** [27]: $rand(\mathscr{C}^{(1)},\mathscr{C}^{(2)}) = \frac{N_{01}+N_{10}}{N_{00}+N_{01}+N_{10}+N_{11}}$, where $N_{11}(N_{00})$ is the numbers of pairwise anchor users who are clustered in the same (different) community(ies) in both $\mathscr{C}^{(1)}$ and $\mathscr{C}^{(2)}$, $N_{01}(N_{10})$ is that of anchor users who are clustered in the same community (different communities) in $\mathscr{C}^{(1)}$ but in different communities (the same communities) in $\mathscr{C}^{(2)}$. (Lower rand corresponds to better performance).
- **variation of information** (vi) [27]: $vi(\mathscr{C}^{(1)},\mathscr{C}^{(2)}) = H(\mathscr{C}^{(1)}) + H(\mathscr{C}^{(2)}) - 2mi(\mathscr{C}^{(1)},\mathscr{C}^{(2)})$. (Lower vi corresponds to better performance).
- **mutual information** [27]: $mi(\mathscr{C}^{(1)},\mathscr{C}^{(2)}) = \sum_{i=1}^{K^{(1)}} \sum_{j=1}^{K^{(2)}} P(i,j) \log \frac{P(i,j)}{P(i)P(j)}$, where $P(i,j) = \frac{|U_i^{(1)} \cap_{\mathscr{A}} U_j^{(2)}|}{|\mathscr{A}|}$ and $|U_i^{(1)} \cap_{\mathscr{A}} U_j^{(2)}| = \left|\{u|u \in U_i^{(1)}, \exists v \in U_i^{(2)}, (u,v) \in \mathscr{A}\}\right|$ [13]. (Higher mi corresponds to better performance).
- **normalized mutual information** [27]: $nmi(\mathscr{C}^{(1)},\mathscr{C}^{(2)}) = \frac{mi(\mathscr{C}^{(1)},\mathscr{C}^{(2)})}{\sqrt{H(\mathscr{C}^{(1)})H(\mathscr{C}^{(2)})}}$. (Higher nmi corresponds to better performance).

The experiment results are available in Tables 7-8. To show the effects of the anchor links, we use the same networks but randomly sample a proportion of anchor links from the networks, whose number is controlled by $\sigma \in \{0.1, 0.2, \cdots, 1.0\}$, where $\sigma = 0.1$ means that 10% of all the anchor links are preserved and $\sigma = 1.0$ means that all the anchor links are preserved.

Table 7 displays the clustering results of different methods in Foursquare and Twitter respectively under the evaluation of ndbi, entropy, density and silhouette. As shown in these two tables, MCD can achieve the highest ndbi score in both Foursquare and Twitter for different sample rate of anchor links consistently. The entropy of the clustering results achieved by MCD is the lowest among all other comparison methods and is about 70% lower than SICLUS, 40% lower than NCUT and KMEANS in both Foursquare and Twitter. In each community detected by MCD, the social connections are denser than that of SICLUS , NCUT and KMEANS. Similar results can be obtained under the evaluation of silhouette, the silhouette score achieved by MCD is the highest among all comparison methods. So, MCD can achieve better results than modified multi-view and isolated clustering methods under the evaluation of **quality metrics**.

Table 8 shows the clustering results on the aligned networks under the evaluation of consensus metrics, which include rand, vi, nmi and mi. As shown in Table 8, MCD can perform the best among all the comparison methods under the evaluation of consensus metrics. For example, the rand score of MCD is the lowest among all other methods and when $\sigma = 0.5$, the rand score of MCD is 20% lower than SICLUS, 72% lower than NCUT and KMEANS. Similar results can be obtained for other evaluation metrics, like when $\sigma = 0.5$, the vi score of MCD is about half of the the score of SICLUS; the nmi and mi score of MCD is the triple of that ofKMEANS.

Table 7 Community Detection Results of Foursquare and Twitter Evaluated by Quality Metrics.

			remaining anchor link rates σ									
network	measure	methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Foursquare	ndbi	MCD	**0.927**	**0.924**	**0.95**	**0.969**	**0.966**	**0.961**	**0.958**	**0.954**	**0.971**	**0.958**
		SICLUS	0.891	0.889	0.88	0.877	0.894	0.883	0.89	0.88	0.887	0.893
		NCUT	0.863	0.863	0.863	0.863	0.863	0.863	0.863	0.863	0.863	0.863
		KMEANS	0.835	0.835	0.835	0.835	0.835	0.835	0.835	0.835	0.835	0.835
	entropy	MCD	**1.551**	**1.607**	**1.379**	**1.382**	**1.396**	**1.382**	**1.283**	**1.552**	**1.308**	**1.497**
		SICLUS	4.332	4.356	4.798	4.339	4.474	4.799	4.446	4.658	4.335	4.459
		NCUT	2.768	2.768	2.768	2.768	2.768	2.768	2.768	2.768	2.768	2.768
		KMEANS	2.369	2.369	2.369	2.369	2.369	2.369	2.369	2.369	2.369	2.369
	density	MCD	**0.216**	**0.205**	**0.196**	0.163	**0.239**	**0.192**	**0.303**	**0.198**	0.170	**0.311**
		SICLUS	0.116	0.121	0.13	0.095	0.143	0.11	0.13	0.12	0.143	0.103
		NCUT	0.154	0.154	0.154	0.154	0.154	0.154	0.154	0.154	0.154	0.154
		KMEANS	0.182	0.182	0.182	**0.182**	0.182	0.182	0.182	0.182	**0.182**	0.182
	silhouette	MCD	**-0.137**	**-0.114**	**-0.148**	**-0.156**	**-0.117**	**-0.11**	**-0.035**	**-0.125**	**-0.148**	**-0.044**
		SICLUS	-0.168	-0.198	-0.173	-0.189	-0.178	-0.181	-0.21	-0.195	-0.167	-0.18
		NCUT	-0.34	-0.34	-0.34	-0.34	-0.34	-0.34	-0.34	-0.34	-0.34	-0.34
		KMEANS	-0.297	-0.297	-0.297	-0.297	-0.297	-0.297	-0.297	-0.297	-0.297	-0.297
Twitter	ndbi	MCD	**0.962**	**0.969**	**0.955**	**0.969**	**0.97**	**0.958**	**0.952**	**0.96**	**0.946**	**0.953**
		SICLUS	0.815	0.843	0.807	0.83	0.826	0.832	0.835	0.808	0.812	0.836
		NCUT	0.759	0.759	0.759	0.759	0.759	0.759	0.759	0.759	0.759	0.759
		KMEANS	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761	0.761
	entropy	MCD	**2.27**	**2.667**	**2.48**	**2.381**	**2.43**	**2.372**	**2.452**	**2.459**	**2.564**	**2.191**
		SICLUS	4.780	5.114	5.066	4.961	4.904	4.866	5.121	4.629	4.872	5.000
		NCUT	3.099	3.099	3.099	3.099	3.099	3.099	3.099	3.099	3.099	3.099
		KMEANS	3.245	3.245	3.245	3.245	3.245	3.245	3.245	3.245	3.245	3.245
	density	MCD	**0.14**	0.097	**0.142**	0.109	**0.15**	**0.158**	**0.126**	**0.149**	**0.147**	**0.164**
		SICLUS	0.055	0.017	0.044	0.026	0.04	0.062	0.016	0.044	0.045	0.02
		NCUT	0.107	0.107	0.107	0.107	0.107	0.107	0.107	0.107	0.107	0.107
		KMEANS	0.119	**0.119**	0.119	**0.119**	0.119	0.119	0.119	0.119	0.119	0.119
	silhouette	MCD	**-0.137**	**-0.179**	**-0.282**	**-0.175**	**-0.275**	**-0.273**	**-0.248**	**-0.269**	**-0.266**	**-0.286**
		SICLUS	-0.356	-0.322	-0.311	-0.347	-0.346	-0.349	-0.323	-0.363	-0.345	-0.352
		NCUT	-0.424	-0.424	-0.424	-0.424	-0.424	-0.424	-0.424	-0.424	-0.424	-0.424
		KMEANS	-0.406	-0.406	-0.406	-0.406	-0.406	-0.406	-0.406	-0.406	-0.406	-0.406

As a result, MCD can achieve better performance than both modified multi-view and isolated clustering methods under the evaluation of **consensus metrics**.

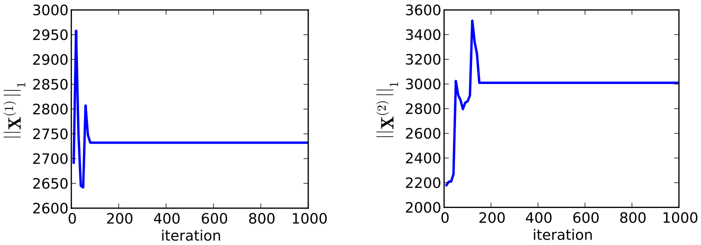
According to the results shown in Tables 7-8, we observe that the performance of MCD doesn't varies much as σ changes. The possible reason can be that, in method MCD, normalized clustering discrepancy is applied to infer the clustering confidence matrices. As σ increases in the experiments, more anchor links are added between networks, part of whose effects will be neutralized by the normalization of clustering discrepancy and doesn't affect the performance of MCD much.

3.5 Convergence Analysis

MCD can compute the solution of the optimization function with Curvilinear Search method, which can update matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$ alternatively. This pro-

Table 8 Community Detection Results of Foursquare and Twitter Evaluated by Consensus Metrics.

		remaining anchor link rates σ									
measure	methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
rand	MCD	**0.095**	**0.099**	**0.107**	**0.138**	**0.116**	**0.121**	**0.132**	**0.106**	**0.089**	0.159
	SICLUS	0.135	0.139	0.144	0.148	0.142	0.14	**0.132**	0.132	0.144	**0.141**
	NCUT	0.399	0.377	0.372	0.4	0.416	0.423	0.362	0.385	0.362	0.341
	KMEANS	0.436	0.387	0.4	0.358	0.403	0.363	0.408	0.365	0.35	0.363
vi	MCD	**3.309**	**4.052**	**4.058**	**3.902**	**4.038**	**4.348**	**3.973**	**3.944**	**4.078**	**2.911**
	SICLUS	7.56	8.324	8.414	8.713	8.756	8.836	8.832	8.621	8.427	8.02
	NCUT	5.384	5.268	5.221	4.855	5.145	5.541	5.909	5.32	5.085	5.246
	KMEANS	5.427	5.117	5.355	5.326	5.679	5.944	5.452	5.567	5.513	4.686
nmi	MCD	0.152	**0.152**	**0.149**	**0.141**	**0.149**	**0.156**	**0.142**	**0.158**	**0.147**	0.146
	SICLUS	**0.172**	0.097	0.081	0.06	0.056	0.069	0.078	0.093	0.105	**0.149**
	NCUT	0.075	0.074	0.111	0.108	0.109	0.099	0.05	0.036	0.042	0.106
	KMEANS	0.008	0.047	0.048	0.054	0.048	0.028	0.047	0.014	0.067	0.119
mi	MCD	0.756	**0.611**	**0.4**	0.258	**0.394**	**0.431**	**0.381**	**0.533**	**0.697**	0.689
	SICLUS	**0.780**	0.446	0.367	**0.277**	0.258	0.325	0.374	0.44	0.489	**0.698**
	NCUT	0.188	0.181	0.261	0.232	0.252	0.243	0.138	0.092	0.111	0.31
	KMEANS	0.02	0.112	0.119	0.135	0.127	0.078	0.119	0.038	0.194	0.314



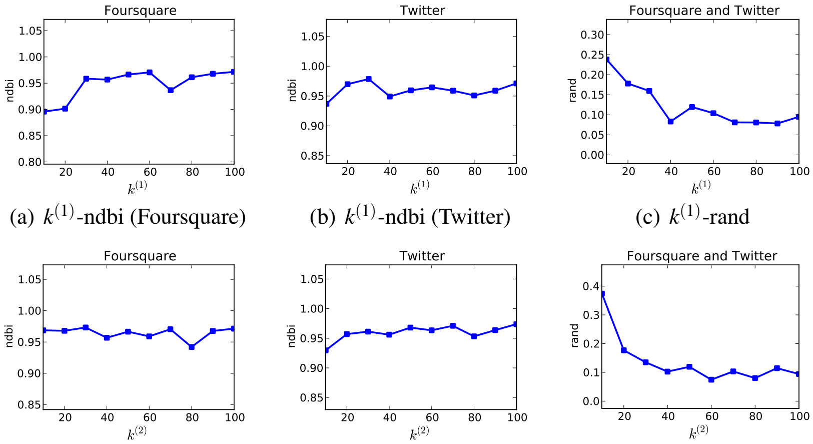
(a) $\left\|\mathbf{X}^{(1)}\right\|_1$ (b) $\left\|\mathbf{X}^{(2)}\right\|_1$

Fig. 7 $\left\|\mathbf{X}^{(1)}\right\|_1$ and $\left\|\mathbf{X}^{(2)}\right\|_1$ in each iteration.

cess will continue until convergence. To check whether this process can stop or not, in this part, we will analyze the convergence of $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$. In Fig. 7, we show the L^1 norm of matrices $\mathbf{X}^{(1)}$ and $\mathbf{X}^{(2)}$, $\left\|\mathbf{X}^{(1)}\right\|_1$ and $\left\|\mathbf{X}^{(2)}\right\|_1$, in each iteration of the updating algorithm, where the L^p norm of matrix $\mathbf{X}$ is $\|\mathbf{X}\|_p = (\sum_i \sum_i {X_{ij}}^p)^{\frac{1}{p}}$. As shown in Fig. 7, both $\left\|\mathbf{X}^{(1)}\right\|_1$ and $\left\|\mathbf{X}^{(2)}\right\|_1$ can converge in less than 200 iterations.

3.6 Parameter Analysis

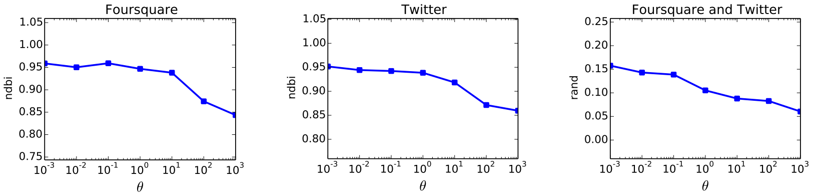
In method MCD, we have three parameters: $k^{(1)}$, $k^{(2)}$ and θ, where $k^{(1)}$ and $k^{(2)}$ are the numbers of clusters in Foursquare and Twitter networks respectively, while



(a) $k^{(1)}$-ndbi (Foursquare) (b) $k^{(1)}$-ndbi (Twitter) (c) $k^{(1)}$-rand

(d) $k^{(2)}$-ndbi (Foursquare) (e) $k^{(2)}$-ndbi (Twitter) (f) $k^{(2)}$-rand

Fig. 8 Analysis of parameters $k^{(1)}$ and $k^{(2)}$.



(a) θ-ndbi (Foursquare) (b) θ-ndbi (Twitter) (c) θ-rand

Fig. 9 Analysis of parameter θ.

θ is the weight of the normalized discrepancy term in the object function. In the pervious experiment, we set $k^{(1)} = 50$, $k^{(2)} = 50$ and $\theta = 1.0$. Here we will analyze the sensitivity of these parameters in details.

To analyze $k^{(1)}$, we fix $k^{(2)} = 50$ and $\theta = 1.0$ but assign $k^{(1)}$ with values in $\{10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. The clustering results of MCD with different $k^{(1)}$ evaluated by *ndbi* and *rand* metrics are given in Fig. 8(a)-8(c). As shown in the figures, the results achieved by MCD are very stable for $k^{(1)}$ with in range $[40, 100]$ under the evaluation of *ndbi* in both Foursquare and Twitter. Similar results can be obtained in Fig. 8(c), where the performance of MCD on aligned networks is not sensitive to the choice of $k^{(1)}$ for $k^{(1)}$ in range $[40, 100]$ under the evaluation of both *rand*. In a similar way, we can study the sensitivity of parameter $k^{(2)}$, the results about which are shown in Fig. 8(d)-8(f).

To analyze the parameter θ, we set both $k^{(1)}$ and $k^{(2)}$ as 50 but assign θ with values in {0.001, 0.01, 0.1, 1.0, 10.0, 100.0, 1000.0}. The results are shown in Fig. 9, where when θ is small, e.g., 0.001, the *ndbi* scores achieved by MCD in both Foursquare and Twitter are high but the *rand* score is not good (*rand* is inversely proportional). On the other hand, large θ can lead to good *rand* score but bad *ndbi* scores in both Foursquare and Twitter. As a result, (1) large θ prefers consensus results, (2) small θ can preserve network characteristics and prefers high quality results.

4 Conclusion and Future Works

In this chapter, we have introduced several research works across multiple **aligned social networks**, including the **network alignment** problem, **link transfer** problem and **community detection** problem. The problems introduced in this chapter are all very important for many concrete real-world social network applications and services. Several nontrivial algorithms have been proposed to resolve these problems respectively, whose performance are evaluated with several real-world datasets

Besides the works introduced in this chapters, many other research problems have been studied across the aligned social networks, like **network embedding**, **information diffusion**, **viral marketing** and **tipping user detection**.

There are also several interesting directions for further research in the domain of **social network fusion learning studies**:

- **Multiple Aligned Social Sites**: Existing aligned network studies mainly focus on studying two aligned networks. Meanwhile, when it comes to **multiple aligned networks** (more than two), many of the studied problems will encounter many new challenges, e.g., the balance of information from different sites, constraints introduced by the multiple sources (e.g., on anchor links).
- **Large Scale Networks**: Most of the introduced methods and models work very well for small-sized social networks, but when it comes to the large scale networks they will suffer from the high time complexity problem a lot. Extending and generalize the existing models to the scalable version will be an interesting direction.
- **Domain Difference Problem**: Many of the existing cross-network studies tackle the domain difference problem in a very simple way, e.g., the meta path selection in **link prediction**, and meta path weighting in **community detection** and **information diffusion**. A more general and effective method to handle the domain difference problem is still an open problem so far.

References

1. S. Bickel and T. Scheffer. Multi-view clustering. In *ICDM*, 2004.
2. X. Cai, F. Nie, and H. Huang. Multi-view k-means clustering on big data. In *IJCAI*, 2013.
3. C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
4. Y. Dong, J. Tang, S. Wu, J. Tian, N. Chawla, J. Rao, and H. Cao. Link prediction and recommendation across heterogeneous social networks. In *ICDM*, 2012.

5. C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *KDD*, 2008.
6. L. Getoor and C. P. Diehl. Link mining: A survey. *SIGKDD Explorations Newsletter*, 2005.
7. M. Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *SDM*, 2006.
8. M. A. Hasan and M. J. Zaki. A survey of link prediction in social networks. In *Social Network Data Analytics*. Springer, 2011.
9. S. Jin, J. Zhang, P. Yu, S. Yang, and A. Li. Synergistic partitioning in multiple large scale social networks. In *IEEE BigData*, 2014.
10. G. Klau. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, 2009.
11. G. Klau. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, 2009.
12. X. Kong, P. Yu, Y. Ding, and D. Wild. Meta path-based collective classification in heterogeneous information networks. In *CIKM*, 2012.
13. X. Kong, J. Zhang, and P. Yu. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*, 2013.
14. D. Koutra, H. Tong, and D. Lubensky. Big-align: Fast bipartite graph alignment. In *ICDM'13*, 2013.
15. O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj. Topological network alignment uncovers biological function and phylogeny. *Journal of The Royal Society Interface*, 2010.
16. A. Kumar and H. Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, 2011.
17. Y. Li, C. Shi, P. Yu, and Q. Chen. Hrank: A path based ranking method in heterogeneous information network. In F. Li, G. Li, S. Hwang, B. Yao, and Z. Zhang, editors, *Web-Age Information Management*. 2014.
18. C. Liao, K. Lu, M. Baym, R. Singh, and B. Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 2009.
19. D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, 2003.
20. Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu. Understanding of internal clustering validation measures. In *ICDM*, 2010.
21. E. F. Lock and D. B. Dunson. Bayesian consensus clustering. *Bioinformatics*, 2013.
22. A. Loureno, S. R. Bul, N. Rebagliati, A. L. N. Fred, M. A. T. Figueiredo, and M. Pelillo. Probabilistic consensus clustering using evidence accumulation. *Machine Learning*, 2013.
23. C. Lu, H. Shuai, and P. Yu. Identifying your customers in social networks. In *CIKM*, 2014.
24. U. Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.
25. F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks: A survey. *CoRR*, abs/1308.0971, 2013.
26. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 2004.
27. N. Nguyen and R. Caruana. Consensus clusterings. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, ICDM '07, pages 607–612, 2007.
28. G. Qi, C. Aggarwal, and T. Huang. Community detection with edge content in social media networks. In *ICDE*, 2012.
29. W. Shao, J. Zhang, L. He, and P. Yu. Multi-source multi-view clustering via discrepancy penalty. In *IJCNN*, 2016.
30. J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 2000.
31. R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *RECOMB*, 2007.
32. R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *RECOMB*, 2007.
33. Y. Sun, C. Aggarwal, and J. Han. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *VLDB*, 2012.

34. Y. Sun, R. Barber, M. Gupta, C. Aggarwal, and J. Han. Co-author relationship prediction in heterogeneous bibliographic networks. In *ASONAM*, 2011.
35. Y. Sun, J. Han, C. Aggarwal, and N. Chawla. When will it happen?: relationship prediction in heterogeneous information networks. In *WSDM*, 2012.
36. Y. Sun, J. Han, X. Yan, P. Yu, and T. Wu. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. In *VLDB*, 2011.
37. X. Yin, J. Han, and P. Yu. Crossclus: user-guided multi-relational clustering. *Data Mining and Knowledge Discovery*, 2007.
38. X. Yu, Y. Sun, B. Norick, T. Mao, and J. Han. User guided entity similarity search using meta-path selection in heterogeneous information networks. In *CIKM*, 2012.
39. R. Zafarani and H. Liu. Connecting users across social media sites: A behavioral-modeling approach. In *KDD*, 2013.
40. Q. Zhan, J. Zhang, S. Wang, P. Yu, and J. Xie. Influence maximization across partially aligned heterogenous social networks. In *PAKDD*. 2015.
41. Q. Zhan, J. Zhang, P. Yu, and J. Xie. Discover tipping users for cross network influencing. In *IRI*, 2016.
42. J. Zhang, J. Chen, J. Zhu, Y. Chang, and P. Yu. Link prediction with cardinality constraint. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, WSDM '17, 2017.
43. J. Zhang, X. Kong, and P. Yu. Predicting social links for new users across aligned heterogeneous social networks. In *ICDM*, 2013.
44. J. Zhang, X. Kong, and P. Yu. Transferring heterogeneous links across location-based social networks. In *WSDM*, 2014.
45. J. Zhang, W. Shao, S. Wang, X. Kong, and P. Yu. Pna: Partial network alignment with generic stable matching. In *IRI*, 2015.
46. J. Zhang and P. Yu. Community detection for emerging networks. In *SDM*, 2015.
47. J. Zhang and P. Yu. Integrated anchor and social link predictions across partially aligned social networks. In *IJCAI*, 2015.
48. J. Zhang and P. Yu. Mcd: Mutual clustering across multiple social networks. In *IEEE BigData Congress*, 2015.
49. J. Zhang and P. Yu. Multiple anonymized social networks alignment. In *ICDM*, 2015.
50. J. Zhang and P. Yu. Pct: Partial co-alignment of social networks. In *WWW*, 2016.
51. J. Zhang, P. Yu, and Y. Lv. Organizational chart inference. In *KDD*, 2015.
52. J. Zhang, P. Yu, and Z. Zhou. Meta-path based multi-network collective link prediction. In *KDD*, 2014.