

Scalable Heterogeneous Social Network Alignment through Synergistic Graph Partition

Yuxiang Ren

IFM Lab
Department of Computer Science
Florida State University, FL, USA
yuxiang@ifmlab.org

Lin Meng

IFM Lab
Department of Computer Science
Florida State University, FL, USA
lin@ifmlab.org

Jiawei Zhang

IFM Lab
Department of Computer Science
Florida State University, FL, USA
jiawei@ifmlab.org

ABSTRACT

Social network alignment has been an important research problem for social network analysis in recent years. With the identified shared users across networks, it will provide researchers with the opportunity to achieve a more comprehensive understanding of users' social activities both within and across networks. Social network alignment is a very difficult problem. Besides the challenges introduced by the network heterogeneity, the network alignment can be reduced to a combinatorial optimization problem with an extremely large search space. The learning effectiveness and efficiency of existing alignment models will be degraded significantly as the network size increases. In this paper, we focus on studying the scalable heterogeneous social network alignment problem, and propose to address it with a novel two-stage network alignment model, namely Scalable Heterogeneous Network Alignment (*SHNA*). Based on a group of intra- and inter-network meta diagrams, *SHNA* first partitions the social networks into a group of sub-networks synergistically. Via the partially known anchor links, *SHNA* can extract the partitioned sub-network correspondence relationships. Instead of aligning the complete input network, *SHNA* proposes to identify the anchor links between the matched sub-network pairs, while those between the unmatched sub-networks will be pruned to effectively shrink the search space. Extensive experiments have been done to compare *SHNA* with the state-of-the-art baseline methods on a real-world aligned social networks dataset. The experimental results have demonstrated both the effectiveness and efficiency of *SHNA* in addressing the problem.

KEYWORDS

Heterogeneous Network; Network Alignment; Synergistic Partition

ACM Reference Format:

Yuxiang Ren, Lin Meng, and Jiawei Zhang. 2020. Scalable Heterogeneous Social Network Alignment through Synergistic Graph Partition. In *Proceedings of the 31st ACM Conference on Hypertext and Social Media (HT '20)*, July 13–15, 2020, Virtual Event, USA. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3372923.3404799>

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

HT '20, July 13–15, 2020, Virtual Event, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7098-1/20/07...\$15.00

<https://doi.org/10.1145/3372923.3404799>

1 INTRODUCTION

In recent years, a large number of online social networks have appeared, which can provide people with various kinds of services. To enjoy these different services at the same time, users nowadays are usually involved in a number of online social networks simultaneously. For instance, people will join in Facebook¹ to socialize with their friends; use LinkedIn² to establish their professional profile; rely on Twitter³ to access and comment on the latest news information. However, in the real world, these different online social networks are mostly isolated without any knowledge about the shared users among them, which renders the inter-network social network analysis a great challenge.

Recently, some research works have proposed to study the alignment problem [6, 27] across multiple online social networks. The main objective of the social network alignment problem is to uncover the mappings of common users across networks, which are named as the *anchor links* [6] formally. Social network alignment provides researchers with the opportunity to study the users' social activities from a global perspective. By integrating the social activity information from multiple social sites, we can achieve a more comprehensive knowledge about users' social preferences. Meanwhile, via these inferred anchor links, information can also propagate across different social networks to improve the services of different social networks simultaneously.

Formally, given two networks $G^{(1)}$ and $G^{(2)}$ with m and n users respectively, we can denote the number of true anchor links existing between $G^{(1)}$ and $G^{(2)}$ as l . According to [6], the anchor links to be inferred are usually subject to the *one-to-one* cardinality constraint. In other words, each user will be connected by at most one anchor link between the networks, and we can have $l \leq \min(m, n)$. Social network alignment problem aims at identifying these l true anchor links from the $m \times n$ potential anchor links, which will lead to a combinatorial optimization problem of time complexity $O\left(\binom{m \times n}{\min(m, n)}\right)$. Most of the existing network alignment models are mainly proposed based on the complete input network [6, 25, 26], which will become ineffective for large-scale online social networks with a large number of users.

Problem Studied: In this paper, we will study the scalable online social network alignment problem, where each social network studied is of a heterogeneous structure involving multiple types of nodes and links. To address the problem, a reduction of the search space, i.e., these aforementioned $m \times n$ potential anchor links, is

¹<https://www.facebook.com>

²<https://www.linkedin.com>

³<https://twitter.com>

necessary and critical, which can not only improve the learning effectiveness but also significantly lower down the time costs in model learning.

The heterogeneous social network alignment problem is extremely challenging to address due to several different reasons:

- **Heterogeneity:** There exist various types of heterogeneous information in social networks, which can provide critical signals for identifying the common users across networks. Meanwhile, properly handling such heterogeneous information in a unified way is not an easy task.
- **Scalability:** For the large-sized input online social networks, learning efficiency is a crucial factor to consider in the model building. Few of the existing research works have ever studied this problem yet, which remains an open problem.
- **Generalizability:** To ensure the applicability of the proposed model, we need to propose a general learning model that can be extensible to various learning settings. Besides differentiating the non-existing anchor links from the real ones, the model should also incorporate the *one-to-one* cardinality constraint [6] effectively.

To address these challenges aforementioned, we introduce a novel scalable heterogeneous social network alignment framework, namely **Scalable Heterogeneous Network Alignment (SHNA)**, in this paper. To effectively capture the diverse connections among users within and across networks with heterogeneous information, *SHNA* employs a group of *meta diagrams* in this paper. *Meta diagram* is a novel concept proposed in [16], which includes both *meta path* and more complex meta structures to outline the user correlations both within and across heterogeneous networks. As a scalable and general solution, *SHNA* addresses the social network alignment problem via two stages: *network synergistic partition* and *parallel sub-network alignment*. *SHNA* proposes to partition the large-sized input social network data into a group sub-networks with a synergistic network partition method. The partition process needs to take care of both the diverse intra- and inter-network user connections, where the shared users should be partitioned into the groups with correspondence relationships as indicated by the partially known anchor links. Then alignment will be performed between these identified corresponding sub-networks only in the second stage, whose learning results will be fused to recover the complete alignment result of the input networks.

The remaining parts of this paper are organized as follows. In Section 2, we introduce the definitions of several important terminologies and the formal problem statement. Meta diagram which is the basis of features in this paper is introduced in Section 3. Detailed information about the proposed model is provided in Section 4, whose effectiveness and efficiency are verified in Section 5. Related works are discussed in Section 6 and finally in Section 7 we conclude this paper.

2 PROBLEM FORMULATION

2.1 Terminology Definition

The network we study is an *attributed heterogeneous social network*.

Definition 1 (Attributed Heterogeneous Social Network): The *attributed heterogeneous social network* can be represented as $G = (\mathcal{V}, \mathcal{E}, \mathcal{T})$. $\mathcal{V} = \bigcup_i \mathcal{V}_i$ is the set of different nodes, while $\mathcal{E} = \bigcup_i \mathcal{E}_i$

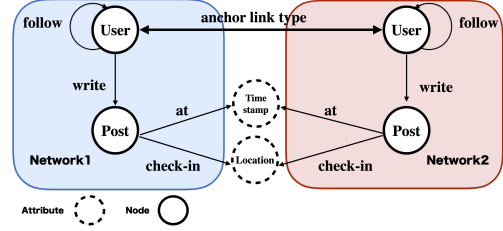


Figure 1: Schema of aligned networks.

represents the set of complex links in the network. Besides, the set $\mathcal{T} = \bigcup_i \mathcal{T}_i$ represents a group of attributes attached to nodes.

Between two *attributed heterogeneous social networks*, if there exist shared users, we define them as *aligned attributed heterogeneous social networks*

Definition 2 (Aligned Attributed Heterogeneous Social Networks): Given the *attributed heterogeneous social networks* $G^{(1)}, G^{(2)}$ and common users are shared between them, we can define them as the *aligned attributed heterogeneous social networks* $\mathcal{G} = ((G^{(1)}, G^{(2)}), \mathcal{A}^{(1,2)})$, and $\mathcal{A}^{(1,2)}$ is the set of undirected anchor links between $G^{(1)}$ and $G^{(2)}$ which connect the common users.

Here, we take two famous online social networks Foursquare and Twitter as an example. We represent them as $\mathcal{G} = ((G^{(1)}, G^{(2)}), \mathcal{A}^{(1,2)})$, where $G^{(1)}$ represents Foursquare and $G^{(2)}$ is Twitter. The Foursquare network $G^{(1)}$ can be represented as $G^{(1)} = (\mathcal{V}^{(1)}, \mathcal{E}^{(1)}, \mathcal{T}^{(1)})$, where $\mathcal{V}^{(1)}$ is the union of $\mathcal{U}^{(1)}$ and $Post^{(1)}$ representing the sets of users and posts in the network respectively. $\mathcal{E}^{(1)} = \mathcal{E}_{u,u}^{(1)} \cup \mathcal{E}_{u,p}^{(1)}$ contains the set of social links among users and the set of write links between users and posts. $\mathcal{T}^{(1)} = \mathcal{T}_l^{(1)} \cup \mathcal{T}_t^{(1)}$ denotes the set of attributes extracted from the posts in $Post^{(1)}$ including location checkins $\mathcal{T}_l^{(1)}$ and timestamps $\mathcal{T}_t^{(1)}$ in this example. The Twitter network can be represented in a similar format as Foursquare, which can be denoted as $G^{(2)} = (\mathcal{V}^{(2)}, \mathcal{E}^{(2)}, \mathcal{T}^{(2)})$. User anchor links in set $\mathcal{A}^{(1,2)}$ connecting to shared users between Foursquare and Twitter can effectively align these two networks together. In the following parts, we illustrate the problem setting and the proposed framework based on the aligned Foursquare and Twitter networks, i.e., $\mathcal{G}$.

2.2 Problem Definition

Given *aligned attributed heterogeneous social networks* $\mathcal{G}$, we can represent all potential anchor links between networks $G^{(1)}$ and $G^{(2)}$ as set $\mathcal{H} = \mathcal{U}^{(1)} \times \mathcal{U}^{(2)}$, where $\mathcal{U}^{(1)}$ and $\mathcal{U}^{(2)}$ denote the user sets in $G^{(1)}$ and $G^{(2)}$ respectively. For the known anchor links, we can group them as a labeled set $\mathcal{D} = \mathcal{A}^{(1,2)}$. The remaining anchor links with unknown labels are those to be inferred, and they can be denoted as the unlabeled set $\mathcal{P} = \mathcal{H} \setminus \mathcal{D}$. Based on both $\mathcal{D}$ and $\mathcal{P}$, we aim at building a mapping function $f: \mathcal{H} \rightarrow \mathcal{Y}$ to infer anchor link labels in $\mathcal{Y} = \{0, +1\}$ subject to the *one-to-one* constraint, where labels +1 and 0 denote the existing and non-existing anchor links respectively.

3 META DIAGRAM

Before introducing the *SHNA* framework, we first introduce *intra-network meta diagram* and *inter-network meta diagram*, which are used to measure the proximity among users in the *SHNA*.

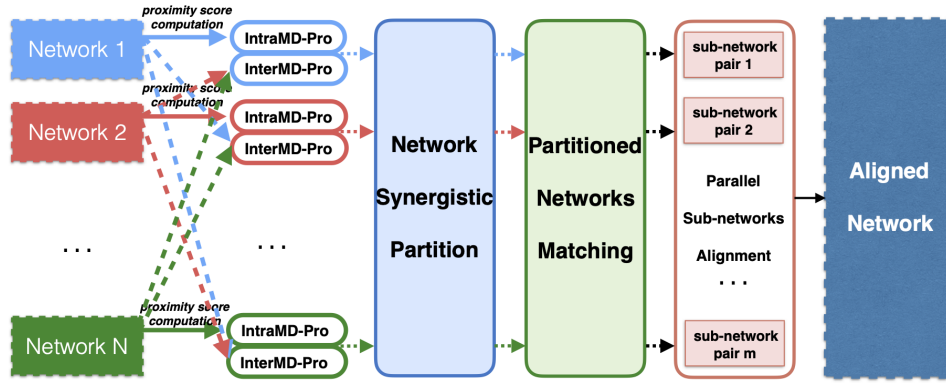


Figure 2: Parallel Implementation of Framework SHNA

3.1 Network Schema

In order to better understand the *aligned attributed heterogeneous social networks*, it is necessary to define a schema-level description. **Definition 3** (Aligned Attributed Heterogeneous Social Network Schema): The schema of the given aligned social networks $\mathcal{G}$ can be represented as $S_{\mathcal{G}} = ((S_{G^{(1)}}, S_{G^{(2)}}), \{\text{anchor}\})$. Here, $S_{G^{(1)}} = (\mathcal{N}_{\mathcal{V}}^{(1)} \cup \mathcal{N}_{\mathcal{T}}^{(1)}, \mathcal{R}_{\mathcal{E}}^{(1)} \cup \mathcal{R}_{\mathcal{A}}^{(1)})$, where $\mathcal{N}_{\mathcal{V}}^{(1)}$ and $\mathcal{N}_{\mathcal{T}}^{(1)}$ denote the set of node types and attribute types in the network, while $\mathcal{R}_{\mathcal{E}}^{(1)}$ represents the set of link types in the network, and $\mathcal{R}_{\mathcal{A}}^{(1)}$ denotes the set of association types between nodes and attributes. In a similar way, the schema of $G^{(2)}$ is $S_{G^{(2)}} = (\mathcal{N}_{\mathcal{V}}^{(2)} \cup \mathcal{N}_{\mathcal{T}}^{(2)}, \mathcal{R}_{\mathcal{E}}^{(2)} \cup \mathcal{R}_{\mathcal{A}}^{(2)})$. We display the schema of the Foursquare and Twitter networks in Figure 1, where the exact node, attribute, and link types can be found intuitively.

3.2 Inter-Network Meta Diagram

The definition of inter-network meta diagram is first proposed in [16]. Based on our own problem, the definition of *inter-network meta diagram* can be presented as follows:

Definition 4 (Inter-Network Meta Diagram): Given *aligned attributed heterogeneous social networks* $S_{\mathcal{G}} = ((S_{G^{(1)}}, S_{G^{(2)}}), \{\text{anchor}\})$. An *inter-network meta diagram* can be formally represented as a directed acyclic subgraph $\Psi^A = (\mathcal{N}_{\Psi}, \mathcal{R}_{\Psi}, N_s, N_t)$, where $\mathcal{N}_{\Psi} \subset (\mathcal{N}_{\mathcal{V}}^{(1)} \cup \mathcal{N}_{\mathcal{V}}^{(2)} \cup \mathcal{N}_{\mathcal{T}}^{(1)} \cup \mathcal{N}_{\mathcal{T}}^{(2)})$ and $\mathcal{R}_{\Psi} \subset (\mathcal{R}_{\mathcal{E}}^{(1)} \cup \mathcal{R}_{\mathcal{E}}^{(2)} \cup \mathcal{R}_{\mathcal{A}}^{(1)} \cup \mathcal{R}_{\mathcal{A}}^{(2)} \cup \{\text{anchor}\})$. N_s, N_t denote the source and target node types from $G^{(1)}$ and $G^{(2)}$ respectively.

The notation, description and physical meanings of *inter-network meta paths* used in this paper are summarized in the first section of Table 1. Because of the problem we try to solve, we are concerned about *inter-network meta diagrams* connecting two users from different networks. We list several *inter-network meta diagram* examples in the second section of Table 1. Now we focus on the Ψ_1^A at first. It is composed of two meta paths which are both P_1^A and represent two users have two followees respectively where there exists an anchor link between these two followees. Ψ_2^A is built by P_5^A and P_6^A which represents two users have posts checking in the same location and at the same time. Ψ_3^A containing 3 inter-network meta paths P_1^A , P_5^A and P_6^A . In a more formal way, we can classify inter-network meta paths as $P_f^A = \{P_1^A, P_2^A, P_3^A, P_4^A\}$ containing the social relationship based inter-network meta paths and $P_a^A = \{P_5^A, P_6^A\}$ representing the sets of the attribute based paths. Besides, we also define that $P^A = P_a^A \cup P_f^A$. We list inter-network meta diagrams used

in SHNA in Table 2. We can represent inter-network meta diagrams as $\Psi^A = P^A \cup \Psi_{f^2}^A \cup \Psi_{a^2}^A \cup \Psi_{f,a}^A \cup \Psi_{f,a^2}^A \cup \Psi_{f^2,a^2}^A$. The physical meanings of notations (e.g. f^2) are also explained in Table 2.

3.3 Intra-Network Meta Diagram

The *intra-network meta diagrams* can be defined in a similar way as *inter-network meta diagrams* in Section 3.2. Their main differences lie in: *inter-network meta diagrams* connect two nodes across two networks but *intra-network meta diagrams* exist in one single network. Formally, we can define *intra-network meta diagrams* as: **Definition 5** (Intra-Network Meta Diagram): Given attributed heterogeneous social network schema $S_G = (\mathcal{N}_{\mathcal{V}} \cup \mathcal{N}_{\mathcal{T}}, \mathcal{R}_{\mathcal{E}} \cup \mathcal{R}_{\mathcal{A}})$. An *intra-network meta diagram* can be defined as a directed acyclic subgraph $\Psi^I = (\mathcal{N}_{\Psi}, \mathcal{R}_{\Psi}, N_s, N_t)$, where $\mathcal{N}_{\Psi} \subset (\mathcal{N}_{\mathcal{V}} \cup \mathcal{N}_{\mathcal{T}})$ and $\mathcal{R}_{\Psi} \subset (\mathcal{R}_{\mathcal{E}} \cup \mathcal{R}_{\mathcal{A}})$, while N_s, N_t denote the source and target node types.

We only consider *intra-network meta diagrams* which $N_s, N_t \in \{U\}$. We list the notation and physical meanings of *intra-network meta paths* used in this paper in the first section of Table 3. Besides, several *intra-network meta diagram* examples are presented in the second section of Table 3. Similar to the *inter-network meta diagrams*, *intra-network meta diagrams* can be represented as $\Psi^I = P^I \cup \Psi_{f^2}^I \cup \Psi_{a^2}^I \cup \Psi_{f,a}^I \cup \Psi_{f,a^2}^I \cup \Psi_{f^2,a^2}^I$.

Meta path [21] is a special type of the meta diagram in the shape of the path. In the following sections, we directly use the term *meta diagram* to refer to both *meta path* and *meta diagram*.

4 PROPOSED METHOD

The structure of SHNA is shown in Figure 2. SHNA is a two-stage framework involving *network synergistic partition* and *parallel sub-network alignment*. *Partitioned networks matching* acts as a bridge between two stages. We will introduce three parts respectively in this section.

4.1 Network Synergistic Partition

The first stage of SHNA is *network synergistic partition*, and we both exploit information within and across networks to obtain the optimal sub-networks. We measure the proximity among users within single network based on *intra-network meta diagrams* and adjust sub-network structures synergistically with the support of *inter-network meta diagrams*.

4.1.1 Intra-Network Meta Diagram based Partition. IntraMD-Pro is defined to measure the proximity among users.

Table 1: Summary of Inter-Network Meta Diagrams.

ID	Notation	Meta Diagram	Semantics
p_1^A	$U \rightarrow U \leftarrow U \leftarrow U$	User $\xrightarrow{follow}$ User $\xleftarrow{anchor}$ User $\xleftarrow{follow}$ User	Common Anchored Followee
p_2^A	$U \leftarrow U \leftarrow U \rightarrow U$	User $\xleftarrow{follow}$ User $\xleftarrow{anchor}$ User $\xrightarrow{follow}$ User	Common Anchored Follower
p_3^A	$U \rightarrow U \leftarrow U \rightarrow U$	User $\xrightarrow{follow}$ User $\xleftarrow{anchor}$ User $\xrightarrow{follow}$ User	Common Anchored Followee-Follower
p_4^A	$U \leftarrow U \leftarrow U \leftarrow U$	User $\xleftarrow{follow}$ User $\xleftarrow{anchor}$ User $\xleftarrow{follow}$ User	Common Anchored Follower-Followee
p_5^A	$U \rightarrow P \rightarrow T \leftarrow P \leftarrow U$	User $\xrightarrow{write}$ Post $\xrightarrow{at}$ Timestamp $\xleftarrow{at}$ Post $\xleftarrow{write}$ User	Common Timestamp
p_6^A	$U \rightarrow P \rightarrow L \leftarrow P \leftarrow U$	User $\xrightarrow{write}$ Post $\xrightarrow{checkin}$ Location $\xleftarrow{checkin}$ Post $\xleftarrow{write}$ User	Common Checkin
$\Psi_1^A(P_1^A \times P_2^A)$	$U \leftrightarrow U \xrightarrow{anchor} U \leftrightarrow U$	User $\xrightarrow{follow}$ User $\xleftarrow{anchor}$ User $\xrightarrow{follow}$ User	Common Aligned Neighbors
$\Psi_2^A(P_5^A \times P_6^A)$	$U \rightarrow P \begin{array}{c} \xrightarrow{\quad} L \xleftarrow{\quad} \\ \xrightarrow{\quad} T \xleftarrow{\quad} \end{array} P \leftarrow U$	User $\xrightarrow{write}$ Post $\xrightarrow{checkin}$ Location $\xleftarrow{checkin}$ Post $\xleftarrow{write}$ User	Common Attributes
$\Psi_3^A(P_1^A \times P_5^A \times P_6^A)$	$U \begin{array}{c} \xrightarrow{\quad} L \xleftarrow{\quad} \\ \xrightarrow{\quad} P \xleftarrow{\quad} \end{array} U$	User $\xrightarrow{follow}$ User $\xleftarrow{anchor}$ User $\xrightarrow{follow}$ User	Common Aligned Neighbor & Attributes

Table 2: Inter-network Meta Diagrams

Set	Physical Meanings
$\Psi_{f,2}^A(P_f^A \times P_f^A)$	Common Aligned Neighbors
$\Psi_{a,2}^A(P_a^A \times P_a^A)$	Common Attributes
$\Psi_{f,a}^A(P_f^A \times P_a^A)$	Common Aligned Neighbor & Attribute
$\Psi_{f,a^2}^A(P_f^A \times P_a^A \times P_a^A)$	Common Aligned Neighbor & Attributes
$\Psi_{f,a^2}^A(P_f^A \times P_f^A \times P_a^A \times P_a^A)$	Common Aligned Neighbors & Attributes

Definition 6 (IntraMD-Pro): Given $\mathcal{D}_{\Psi_i^I}(x, y)$ to represent the set of diagram Ψ_i^I starting from x to y , and $\mathcal{D}_{\Psi_i^I}(x, \cdot)$ to represent the set of diagram Ψ_i^I which go from x to other nodes in the network. The IntraMD-Pro of node pair (x, y) can be defined as

$$\text{IntraMD-Pro}(x, y) = \sum_i \omega_i \left(\frac{|\mathcal{D}_{\Psi_i^I}(x, y)| + |\mathcal{D}_{\Psi_i^I}(y, x)|}{|\mathcal{D}_{\Psi_i^I}(x, \cdot)| + |\mathcal{D}_{\Psi_i^I}(y, \cdot)|} \right),$$

where ω_i is the weight of Ψ_i^I and $\sum_i \omega_i = 1$.

According to [23], the specific values of hyperparameters ω_i can be adjusted automatically by optimizing certain learning objectives, e.g., clustering entropy as used in [23]. We will not elaborate the hyperparameter adjustment algorithm in this paper. What's more, we use $\mathbf{A}_i$ as the *adjacency matrix* which represents Ψ_i^I among users in the network. The proximity score matrix among users of Ψ_i^I can be represented as $\mathbf{S}_i = \mathbf{B}_i \circ (\mathbf{A}_i + \mathbf{A}_i^T)$, where the matrix $\mathbf{B}_i$ represents the out-degree of user x and y , e.g., $\mathbf{B}_i(x, y) = (\sum_m \mathbf{A}_i(x, m) + \sum_m \mathbf{A}_i(y, m))^{-1}$. The $\circ$ represents the Hadamard product. IntraMD-Pro matrix of the network can be represented as:

$$\mathbf{S} = \sum_i \omega_i \mathbf{S}_i = \sum_i \omega_i (\mathbf{B}_i \circ (\mathbf{A}_i + \mathbf{A}_i^T)).$$

We can represent the user-cluster belonging confidence scores as a vector $\mathbf{h}_i = (h_{i,1}, h_{i,2}, \dots, h_{i,k})$, where $h_{i,j}$ denotes the confidence score that $u_i \in \mathcal{U}$ is in the sub-network $U_j \in \mathcal{C}$ ($\mathcal{C}$ is the set of detected clusters), and k is the number of detected communities. Therefore, we can define the partition results of all users in $\mathcal{U}$ as the *user-cluster belonging confidence matrix* $\mathbf{H}$, where $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n]^T$ and $n = |\mathcal{U}|$. We choose to solve the following objective function to minimize the *normalized-cut* (*Ncut*) cost [9, 19] and

achieve the optimal partition result:

$$\begin{aligned} \min_{\mathbf{H}} \quad & \text{Tr}(\mathbf{H}^T \mathbf{L} \mathbf{H}), \\ \text{s.t.} \quad & \mathbf{H}^T \mathbf{D} \mathbf{H} = \mathbf{I}. \end{aligned}$$

where the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{S}$, the 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.

4.1.2 Inter-Network Meta Diagram based Partition. With the help of *inter-network meta diagrams*, we can represent the extra knowledge about the aligned attributed heterogeneous networks from a more complete and convincing view.

Inter-network meta diagrams effectively indicate the closeness among the users across different networks, which can be quantified with the proximity scores in this paper. Given a pair of users $u_x^{(1)}$ and $u_y^{(2)}$, we denote the set of inter-network diagram Ψ_i^A connecting $u_x^{(1)}$ and $u_y^{(2)}$ as $\mathcal{D}_{\Psi_i^A}(u_x^{(1)}, u_y^{(2)})$. Formally, we represent all inter-network meta diagram instances going out from user $u_x^{(1)}$ (or going into $u_y^{(2)}$) as set $\mathcal{D}_{\Psi_i^A}(u_x^{(1)}, \cdot)$ (or $\mathcal{D}_{\Psi_i^A}(\cdot, u_y^{(2)})$). The proximity score between $u_x^{(1)}$ and $u_y^{(2)}$ based on Ψ_i^A can be defined as the following *InterMD-Pro*.

Definition 7 (InterMD-Pro): Based on Ψ_i^A , the proximity between $u_x^{(1)}$ and $u_y^{(2)}$ in $\mathcal{G}$ can be represented as:

$$s_{\Psi_i^A}(u_x^{(1)}, u_y^{(2)}) = \frac{2|\mathcal{D}_{\Psi_i^A}(u_x^{(1)}, u_y^{(2)})|}{|\mathcal{D}_{\Psi_i^A}(u_x^{(1)}, \cdot)| + |\mathcal{D}_{\Psi_i^A}(\cdot, u_y^{(2)})|}.$$

Based on the promixity of every single inter-network meta diagram, InterMD-proximity between users $u_x^{(1)}$ and $u_y^{(2)}$ in $\mathcal{G}$ can be defined as

$$\text{InterMD-Pro}(u_x^{(1)}, u_y^{(2)}) = \sum_i \omega_i (s_{\Psi_i^A}(u_x^{(1)}, u_y^{(2)})).$$

where ω_i is the weight of Ψ_i^A and $\sum_i \omega_i = 1$.

The promixity matrix among all users across networks can be represented as $\mathbf{S}_{\Psi^A} \in \mathbb{R}^{|\mathcal{U}^{(1)}| \times |\mathcal{U}^{(2)}|}$, and $\mathbf{S}_{\Psi^A}(x, y) = \text{InterMD-Pro}(u_x^{(1)}, u_y^{(2)})$. We can correlate users together with their cluster belonging relationships effectively across networks with the matrix $\mathbf{S}_{\Psi^A}$. Given one user $u_l^{(1)}$ in $G^{(1)}$, we are able to calculate the user-cluster belonging confidence scores of $u_l^{(1)}$ with the user-cluster belonging

Table 3: Summary of Intra-Network Meta Diagram.

ID	Notation	Meta Diagram	Semantics
P_1^I	$U \rightarrow U$	User $\xrightarrow{\text{follow}}$ User	Follow
P_2^I	$U \rightarrow U \rightarrow U$	User $\xrightarrow{\text{follow}}$ User $\xrightarrow{\text{follow}}$ User	Follower of Follower
P_3^I	$U \rightarrow U \leftarrow U$	User $\xrightarrow{\text{follow}}$ User $\xrightarrow{\text{follow}^{-1}}$ User	Common Out Neighbor
P_4^I	$U \leftarrow U \rightarrow U$	User $\xrightarrow{\text{follow}^{-1}}$ User $\xrightarrow{\text{follow}}$ User	Common In Neighbor
P_5^I	$U \rightarrow P \rightarrow T \leftarrow P \leftarrow U$	User $\xrightarrow{\text{write}}$ Post $\xrightarrow{\text{at}}$ Timestamp $\xleftarrow{\text{at}}$ Post $\xleftarrow{\text{write}}$ User	Posts Containing Common Timestamps
P_6^I	$U \rightarrow P \rightarrow L \leftarrow P \leftarrow U$	User $\xrightarrow{\text{write}}$ Post $\xrightarrow{\text{checkin}}$ Location $\xleftarrow{\text{checkin}}$ Post $\xleftarrow{\text{write}}$ User	Posts Attaching Common Location Check-ins
$\Psi_1^I(P_1^I \times P_1^I)$	$U \leftrightarrow U$	User $\xrightarrow{\text{follow}}$ User User $\xleftarrow{\text{follow}}$ User	Follower and Followee
$\Psi_2^I(P_5^I \times P_6^I)$	$U \rightarrow P \begin{cases} \rightarrow L \\ \rightarrow T \end{cases} \leftarrow P \leftarrow U$	User $\xrightarrow{\text{write}}$ Post $\xrightarrow{\text{checkin}}$ Location $\xleftarrow{\text{checkin}}$ Post $\xleftarrow{\text{write}}$ User Post $\xrightarrow{\text{at}}$ Timestamp $\xleftarrow{\text{at}}$ Post	Common Attributes
$\Psi_3^I(P_1^I \times P_5^I \times P_6^I)$	$U \rightarrow P \begin{cases} \rightarrow L \\ \rightarrow T \end{cases} \leftarrow P \leftarrow U$	User $\xrightarrow{\text{write}}$ Post $\xrightarrow{\text{checkin}}$ Location $\xleftarrow{\text{checkin}}$ Post $\xleftarrow{\text{write}}$ User Post $\xrightarrow{\text{at}}$ Timestamp $\xleftarrow{\text{at}}$ Post User $\xrightarrow{\text{follow}}$ User	Common Attributes & Follower and Followee

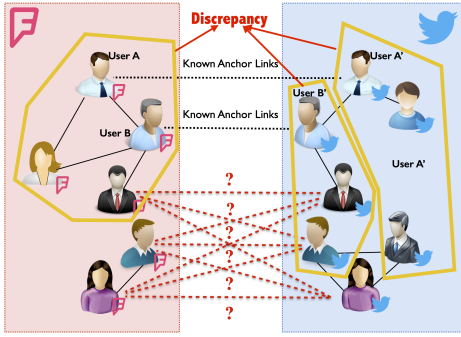


Figure 3: An illustration of Discrepancy. There exists a known anchor link between User A and User A', also User B and User B'. If User A and User B are partitioned into the same sub-network, but User A' and User B' are partitioned into different sub-networks, the discrepancy will arise. Synergistic partition works on eliminating the discrepancy

confidence scores from $G^{(2)}$. Formally, we define *Transition User-cluster Belonging Confidence Scores* as follow:

$$\bar{h}_l^{(1)} = \sum_{j=1}^{|\mathcal{U}^{(2)}|} S_{\Psi A}(l, j) \cdot h_j^{(2)}$$

By maximizing the consensus of partition results based on the *transition user-cluster belonging confidence scores*, we can refine the partition results with information from the other partially aligned network synergistically. In this paper, we will propose the definition of *discrepancy*, which measures how different the shared user pairs are clustered across networks. We provide an illustration about *discrepancy* in Figure 3.

Definition 8 (Discrepancy): Given two users $u_l^{(1)}$ and $u_m^{(1)}$ in $G^{(1)}$, If users $u_l^{(1)}$ and $u_m^{(1)}$ are partitioned into the same sub-network in $G^{(1)}$ but into different sub-networks based on the *transition user-cluster belonging confidence scores* from $G^{(2)}$, then it will lead to the *discrepancy* between the partition results of $u_l^{(1)}$, $u_m^{(1)}$. The confidence that $u_l^{(1)}$ and $u_m^{(1)}$ are in the same sub-network can be denoted as $h_l^{(1)}(h_m^{(1)})^\top$. Formally, the *discrepancy* of $u_l^{(1)}$ and $u_m^{(1)}$ is defined to be $d_{lm}(C^{(1)}) = (h_l^{(1)}(h_m^{(1)})^\top - \bar{h}_l^{(1)}(\bar{h}_m^{(1)})^\top)^2$. Furthermore, the discrepancy of $C^{(1)}$:

$$d(C^{(1)}) = \sum_i^{|\mathcal{U}^{(1)}|} \sum_{j=i+1}^{|\mathcal{U}^{(1)}|} d_{ij}(C^{(1)})$$

With the *user-cluster belonging confidence matrices* $H^{(1)}$ and $H^{(2)}$, the discrepancy of $\mathcal{G}$ is

$$d(C^{(1)}, C^{(2)}) = d(C^{(1)}) + d(C^{(2)}) \\ = \left\| \bar{H}^{(1)} (\bar{H}^{(1)})^\top - H^{(1)} (H^{(1)})^\top \right\|_F^2 + \left\| \bar{H}^{(2)} (\bar{H}^{(2)})^\top - H^{(2)} (H^{(2)})^\top \right\|_F^2.$$

Where $\bar{H}^{(1)} = [\bar{h}_1^{(1)}, \bar{h}_2^{(1)}, \dots, \bar{h}_n^{(1)}]^\top$ and $n = |\mathcal{U}^{(1)}|$, and $\bar{H}^{(2)}$ is the same situation. Besides, $\bar{H}^{(2)} = (S_{\Psi A})^\top H^{(1)}$ and $\bar{H}^{(1)} = S_{\Psi A} H^{(2)}$.

4.1.3 Synergistic Partition of Multiple Networks. By taking both *Intra-Network Meta Diagram based Partition* and *Inter-Network Meta Diagram based Partition* into considerations, the optimal synergistic partition results $C^{(1)}$ and $C^{(2)}$ can be achieved by minimizing both the $Ncut$ costs and the discrepancy simultaneously as follows:

$$\arg \min_{C^{(1)}, C^{(2)}} \alpha \cdot Ncut(C^{(1)}) + \beta \cdot Ncut(C^{(2)}) + \theta \cdot d(C^{(1)}, C^{(2)})$$

where α , β and θ represent the weights of these compositions. We can replace $Ncut(C^{(1)})$, $Ncut(C^{(2)})$, $d(C^{(1)}, C^{(2)})$ with the terms derived before, and the joint objective function can be rewritten as:

$$\min_{H^{(1)}, H^{(2)}} \alpha \cdot \text{Tr}((H^{(1)})^\top L^{(1)} H^{(1)}) + \beta \cdot \text{Tr}((H^{(2)})^\top L^{(2)} H^{(2)}) \\ + \theta \cdot \left\| \bar{H}^{(1)} (\bar{H}^{(1)})^\top - H^{(1)} (H^{(1)})^\top \right\|_F^2 + \theta \cdot \left\| \bar{H}^{(2)} (\bar{H}^{(2)})^\top - H^{(2)} (H^{(2)})^\top \right\|_F^2, \\ s.t. (H^{(1)})^\top D^{(1)} H^{(1)} = I^{(1)}, (H^{(2)})^\top D^{(2)} H^{(2)} = I^{(2)},$$

The joint objective function involves two variables: $H^{(1)}$ and $H^{(2)}$, and the objective is not jointly convex. Besides, the objective function contains complex orthogonality constraints which are numerically expensive to preserve in optimization. In order to preserve constraints in an efficient way during the learning process, we propose to relax the objective function as follows:

$$\min_{H^{(1)}, H^{(2)}} \alpha \cdot Ncut(C^{(1)}) + \beta \cdot Ncut(C^{(2)}) + \theta \cdot d(C^{(1)}, C^{(2)}) \\ + \rho_1 \left\| (H^{(1)})^\top D^{(1)} H^{(1)} - I^{(1)} \right\|_F^2 + \rho_2 \left\| (H^{(2)})^\top D^{(2)} H^{(2)} - I^{(2)} \right\|_F^2,$$

By setting ρ_1 and ρ_2 with large values, e.g., 10^9 , optimizing the above function is (approximately) equivalent to the original objective function. We design an hierarchical alternative variable updating process for solving the problem:

• **Step (1):** Fix $\mathbf{H}^{(2)}$, Update $\mathbf{H}^{(1)}$.

Based on the Gradient Descent, we calculate $\mathbf{H}_k^{(1)}$ representing $\mathbf{H}^{(1)}$ after k descent steps:

$$\mathbf{H}_k^{(1)} = \mathbf{H}_{k-1}^{(1)} - \eta_1 \cdot \nabla \mathcal{L}(\mathbf{H}^{(1)})$$

η_1 is the step length, and the gradient $\nabla \mathcal{L}(\mathbf{H}^{(1)})$ is:

$$\begin{aligned} \nabla \mathcal{L}(\mathbf{H}^{(1)}) = & \frac{\partial \mathcal{L}}{\partial \mathbf{H}^{(1)}} \\ & + \alpha \cdot (\mathbf{L}^{(1)} \mathbf{H}^{(1)} + (\mathbf{L}^{(1)})^T \mathbf{H}^{(1)}) \\ & + 2\theta (\mathbf{X} \mathbf{H}^{(1)} + \mathbf{X}^T \mathbf{H}^{(1)} + 2\mathbf{H}^{(1)} (\mathbf{H}^{(1)})^T \mathbf{H}^{(1)}) \\ & + 2\theta (2\mathbf{S}_{\Psi_A} \mathbf{S}_{\Psi_A}^T \mathbf{H}^{(1)} (\mathbf{H}^{(1)})^T \mathbf{S}_{\Psi_A} \mathbf{S}_{\Psi_A}^T \mathbf{H}^{(1)} - \mathbf{X} \mathbf{H}^{(1)} - \mathbf{X}^T \mathbf{H}^{(1)}) \\ & + 4\rho_1 (\mathbf{D}^{(1)} \mathbf{H}^{(1)} (\mathbf{H}^{(1)})^T \mathbf{D}^{(1)} \mathbf{H}^{(1)} - \mathbf{D}^{(1)} \mathbf{H}^{(1)}) \end{aligned}$$

where $\mathbf{X} = \mathbf{S}_{\Psi_A} \mathbf{Y} \mathbf{S}_{\Psi_A}^T$ and $\mathbf{Y} = \mathbf{H}^{(2)} (\mathbf{H}^{(2)})^T$.

• **Step (2):** Fix $\mathbf{H}^{(1)}$, Update $\mathbf{H}^{(2)}$.

When $\mathbf{H}^{(1)}$ is fixed, the method of updating $\mathbf{H}^{(2)}$ with a fixed $\mathbf{H}^{(1)}$ is almost the same as **Step (1)**:

$$\mathbf{H}_k^{(2)} = \mathbf{H}_{k-1}^{(2)} - \eta_2 \cdot \nabla \mathcal{L}(\mathbf{H}^{(2)})$$

We iteratively operate **Step (1)** and **Step (2)**, and every iteration operates one step descent for $\mathbf{H}^{(1)}, \mathbf{H}^{(2)}$ until convergence.

Based on the learned matrix $\mathbf{H}^{(1)}$, we can learn the clusters of users in network $G^{(1)}$ by applying K-Means algorithms to the learned latent vectors, i.e., rows of matrix $\mathbf{H}^{(1)}$, and the detected clusters can be represented as set $C^{(1)} = \{U_1^{(1)}, U_2^{(1)}, \dots, U_k^{(1)}\}$. For the users within the same cluster, we propose to extract a sub-network formed by these users and other associated nodes/attributes. For instance, based on the cluster $U_l^{(1)} \in C^{(1)}$, we can represent the extracted sub-network as $G_l^{(1)}$. Formally, the set of extracted sub-network from $G^{(1)}$ based on the clustering result $C^{(1)}$ can be represented as $\mathcal{G}^{(1)} = \{G_1^{(1)}, G_2^{(1)}, \dots, G_k^{(1)}\}$. Here, we need to add a remark that the synergistic network partition process involves an iterative variable updating process, which may take some time to converge. Meanwhile, in the real-world application of the proposed model, such a step can be done in an offline manner, where the clustering results can be computed and stored in hard-drive in advance. It will greatly improve the learning efficiency of *SHNA* in aligning the large-scale social networks.

4.2 Partitioned Networks Matching

After partitioning the original networks, it's critical to matching the sub-networks from different networks, which is the prerequisite for the next stage. As a bridge, the matching step should consider not only the object of *network synergistic partition* but also the target of *parallel sub-network alignment*. Here, we propose the sub-network Matching Score as the metric to serve for partitioned networks matching in *SHNA*.

Definition 9 (Matching Score): Given two sub-network $G_i^{(1)}$ and $G_j^{(2)}$, which comes from $\mathcal{G}^{(1)}$ and $\mathcal{G}^{(2)}$ respectively. We define

Matching Score(M-Score) between $G_i^{(1)}$ and $G_j^{(2)}$ as:

$$\text{M-Score}(G_i^{(1)}, G_j^{(2)}) = |\mathcal{A}(G_i^{(1)}, G_j^{(2)})| \cdot \frac{|\mathcal{A}(G_i^{(1)}, G_j^{(2)})|}{|U_i^{(1)}| \cdot |U_j^{(2)}|}$$

where $\mathcal{A}(G_i^{(1)}, G_j^{(2)})$ is the set of known anchor links between $G_i^{(1)}$ and $G_j^{(2)}$, and $U_i^{(1)}, U_j^{(2)}$ are sets of user accounts belongs to $G_i^{(1)}, G_j^{(2)}$. In fact, the second term above is the proportion of known links of all links across $G_i^{(1)}$ and $G_j^{(2)}$. M-Score takes both the number of known anchor users and the performance of pruning negative links into considerations. We can match the sub-networks according to the descending rank of M-Score to achieve the sub-network matching results $\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_s\}$, and $\mathcal{M}_i = \{G_a^{(1)}, G_b^{(2)}\}$. Here, s is a parameter we set corresponding to the *top s* ($s \leq k$) pairs for alignment. Then *SHNA* will start to focus on *parallel sub-network alignment* on $\mathcal{M}$.

4.3 Sub-network Alignment

In this part, we introduce the alignment model for all the sub-network pairs in $\mathcal{M}$. We take $\mathcal{M}_i = \{G_a^{(1)}, G_b^{(2)}\}$ as an example to illustrate the alignment process.

4.3.1 Optimization Objective Function. For all the potential anchor links between $G_a^{(1)}$ and $G_b^{(2)}$ in set $\mathcal{H}$, a set of features will be extracted based on *inter-network meta diagrams*. Formally, the feature vector extracted for the link $l \in \mathcal{H}$ can be represented as vector $\mathbf{x}_l \in \mathbb{R}^f$, where f is the number of types of inter-network meta diagrams. Meanwhile, we can denote the label of link $l \in \mathcal{H}$ as $y_l \in \mathcal{Y}$, and $\mathcal{Y} = \{0, +1\}$, which denotes the existence of anchor link l between the networks. For the existing anchor links in set $\mathcal{D}$, they will be assigned with +1 label; while the labels of anchor links in $\mathcal{P}$ are unknown. All the labeled anchor links in set $\mathcal{D}$ can be represented as a tuple set $\{(\mathbf{x}_l, y_l)\}_{l \in \mathcal{D}}$. The *discriminative* component can effectively differentiate the positive instances from the non-existing ones, which can be denoted as mapping $f(\cdot; \theta_f) : \mathbb{R}^d \rightarrow \{+1, 0\}$ parameterized by θ_f . In this paper, we will use a linear model to fit the link instances, and the *discriminative* model to be learned can be represented as $f(\mathbf{x}_l; \mathbf{w}) = \mathbf{w}^T \mathbf{x}_l + b$, where $\theta_f = [\mathbf{w}, b]$. By adding a dummy feature 1 for all the anchor link feature vectors, we can incorporate bias term b into the weight vector $\mathbf{w}$ and the parameter vector can be denoted as $\theta_f = \mathbf{w}$ for simplicity. The introduced *discriminative* loss function on the labeled set $\mathcal{D}$ can be represented as

$$L(f, \mathcal{D}; \mathbf{w}) = \sum_{l \in \mathcal{D}} (f(\mathbf{x}_l; \mathbf{w}) - y_l)^2 = \sum_{l \in \mathcal{D}} (\mathbf{w}^T \mathbf{x}_l - y_l)^2.$$

Meanwhile, we also propose to utilize the unlabeled anchor links to encourage the learned model can capture the salient structures of all the anchor link instances. Based on the above discriminative model function $f(\cdot; \mathbf{w})$, for an unlabeled anchor link $l \in \mathcal{P}$, we can represent its inferred "label" as $y_l = f(\mathbf{x}_l; \mathbf{w})$. Considering that the result of $f(\cdot; \mathbf{w})$ may not necessary the exact label values in $\mathcal{Y}$, in the *generative* component, we can represent the generated anchor link label as $\text{sign}(f(\mathbf{x}_l; \mathbf{w})) \in \{+1, 0\}$. How to determine its value will be introduced later in the joint function. The loss function introduced in the generative component based on the unlabeled anchor links can be denoted as

$$L(f, \mathcal{P}; \mathbf{w}) = \sum_{l \in \mathcal{P}} (\mathbf{w}^T \mathbf{x}_l - \text{sign}(f(\mathbf{x}_l; \mathbf{w})))^2.$$

As introduced before, the anchor links to be inferred between networks are subject to the *one-to-one* cardinality constraint. Subject to the cardinality constraint, the prediction tasks of anchor links between networks are no longer independent. For instance, if anchor link $(u_i^{(1)}, u_j^{(2)})$ is predicted to be positive, then all the remaining anchor links incident to $u_i^{(1)}$ and $u_j^{(2)}$ in the unlabeled set $\mathcal{P}$ will be negative by default. Viewed in such a perspective, the cardinality constraint on anchor links should be effectively incorporated in model building, which will be modeled as the mathematical constraints on node degrees. To represent the user node-anchor link relationships in networks $G_a^{(1)}$ and $G_b^{(2)}$ respectively, we introduce the user node-anchor link incidence matrices $\mathbf{A}^{(1)} \in \{0, 1\}^{|\mathcal{U}_a^{(1)}| \times |\mathcal{H}|}$, $\mathbf{A}^{(2)} \in \{0, 1\}^{|\mathcal{U}_b^{(2)}| \times |\mathcal{H}|}$. Entry $A^{(1)}(i, j) = 1$ iff anchor link $l_j \in \mathcal{H}$ is connected with $u_i^{(1)}$ in $G_a^{(1)}$, and it is similar for $\mathbf{A}^{(2)}$.

According to the analysis provided before, we can represent the labels of links in $\mathcal{H}$ as vector $\mathbf{y} \in \{+1, 0\}^{|\mathcal{H}|}$, where entry $y(i)$ represents the label of link $l_i \in \mathcal{H}$. Based on the anchor link label vector $\mathbf{y}$, user node-anchor link incidence matrices $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$, the *one-to-one* constraint on anchor links can be denoted as the constraints on node degrees as follows:

$$\mathbf{0} \leq \mathbf{A}^{(1)}\mathbf{y} \leq \mathbf{1}, \text{ and } \mathbf{0} \leq \mathbf{A}^{(2)}\mathbf{y} \leq \mathbf{1}.$$

By combining the loss terms introduced by the labeled and unlabeled anchor links together with the cardinality constraint, we can represent the joint optimization objective function as

$$\begin{aligned} \min_{\mathbf{w}, \mathbf{y}} \quad & L(f, \mathcal{D}; \mathbf{w}) + L(f, \mathcal{P}; \mathbf{w}) + \gamma \cdot \|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y_l \in \{+1, 0\}, \forall l \in \mathcal{H}, \\ & \mathbf{0} \leq \mathbf{A}^{(1)}\mathbf{y} \leq \mathbf{1}, \text{ and } \mathbf{0} \leq \mathbf{A}^{(2)}\mathbf{y} \leq \mathbf{1}. \end{aligned}$$

In fact, we can simplify the loss function as:

$$L(f, \mathcal{D}; \mathbf{w}) + L(f, \mathcal{P}; \mathbf{w}) = L(f, \mathcal{H}; \mathbf{w}) = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2,$$

where matrix $\mathbf{X} = [\mathbf{x}_{l_1}^\top, \mathbf{x}_{l_2}^\top, \dots, \mathbf{x}_{l_{|\mathcal{H}|}}^\top]^\top$ denotes the feature matrix of all the links in the set $\mathcal{H}$.

We use a hierarchical alternative variable updating process to solve the problem:

- **Step (1):** Fix $\mathbf{y}$, Update $\mathbf{w}$.

With $\mathbf{y}$ fixed, the objective function involving $\mathbf{w}$ is:

$$\min_{\mathbf{w}} \frac{c}{2} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 + \frac{1}{2} \|\mathbf{w}\|_2^2.$$

Here, the objective function is a quadratic convex function, and its optimal solution can be represented as

$$\mathbf{w} = c(\mathbf{I} + c\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y},$$

where $c(\mathbf{I} + c\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top$ is a constant matrix. Therefore, the weight vector $\mathbf{w}$ depends only on the $\mathbf{y}$ variable.

- **Step (2):** Fix $\mathbf{w}$, Update $\mathbf{y}$.

With $\mathbf{w}$ fixed, together with the constraint, we know that terms $L(f, \mathcal{D}; \mathbf{w})$, $L(f, \mathcal{P}; \mathbf{w})$ and $\|\mathbf{w}\|_2^2$ are all constant. And the objective function will be

$$\begin{aligned} \min_{\mathbf{y}} \quad & \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2 \\ \text{s.t.} \quad & y_l \in \{+1, 0\}, \forall l \in \mathcal{H}, \\ & \mathbf{0} \leq \mathbf{A}^{(1)}\mathbf{y} \leq \mathbf{1}, \text{ and } \mathbf{0} \leq \mathbf{A}^{(2)}\mathbf{y} \leq \mathbf{1}. \end{aligned}$$

Table 4: Properties of the Heterogeneous Networks

property	network	
	Twitter	Foursquare
# node	user	5,223
	tweet/tip	9,490,707
	location	297,182
# link	friend/follow	164,920
	write	9,490,707

It is an integer programming problem, we use the greedy link selection algorithm proposed in [25] based on values $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$, which can achieve $\frac{1}{2}$ -approximation of the optimal solution.

4.3.2 Parallel Implementation of Sub-network Alignment. The time complexity of these two steps is related to $|\mathcal{H}|$ which is determined by the number of users from two sub-networks. The alignment for all the sub-network pairs in the set $\mathcal{M}$ can be implemented in parallel, so compared with the alignment method conducted in the whole networks directly, *SHNA* has the apparent advantage even counting the time consumption of the *network synergistic partition*. Finally, we have to aggregate alignment results from parallel sub-network alignment in the sub-network pairs. In *SHNA*, we choose to preserve the original results from all sub-network pairs in the set $\mathcal{M}$ as the final result.

5 EXPERIMENTS

5.1 Dataset

Our dataset comes from two real-world heterogeneous networks: *Foursquare* and *Twitter*, which are both famous online social networks. The key statistical data of these two networks is listed in Table 4. Detailed information about the strategy of crawling the dataset can be reached in [6].

5.2 Experimental Settings

5.2.1 Experimental Setup. We can obtain the set of anchor links across Foursquare and Twitter, which will be the positive links. The links between users from Foursquare and Twitter except for anchor links can be treated as negative links. We apply the 2-fold cross-validation to partition the links with the ratio 1 : 1. One fold will be used as the training set and the other one will be treated as the test set. The features depending on the known anchor links like inter-network meta diagrams are extracted only on the basis of the training set. All codes are implemented in Python, and we run the experiments on a Dell PowerEdge T630 Server with 2 20-core Intel CPUs and 256GB memory.

5.2.2 Comparison Methods. Comparison methods in the experiments can be divided into 2 categories according to whether original networks are partitioned or not in building models.

Comparison Methods without partition:

- **IterClip-MD:** *IterClip-MD* extends the cardinality constrained link prediction model in [25] by incorporating meta diagrams.
- **Mna-MD:** *Mna-MD* extends the anchor links prediction model in [6] by incorporating meta diagrams as features.
- **DeepWalk:** A random walk based network embedding method [15]. We utilize it to learn the representation of users merely based on the friendship information and concatenate the representations of two users as the feature of a potential anchor link. Then SVM is trained to predict anchor links based on this feature.

Table 5: Performance comparison of different methods for Network Alignment.

Methods	Metrics			
	Precision	Recall	F1	Time (sec)
$SHNA(\theta = 10)$	0.677 ± 0.002	0.500 ± 0.001	0.575 ± 0.001	7.62
$SHNA(\theta = 80)$	0.691 ± 0.001	0.532 ± 0.011	0.601 ± 0.006	9.12
$SHNA(\theta = 100)$	0.684 ± 0.010	0.515 ± 0.018	0.588 ± 0.015	12.87
<i>SpectralIter</i>	0.481 ± 0.002	0.392 ± 0.013	0.432 ± 0.006	450.14
<i>KmeansIter</i>	0.415 ± 0.003	0.239 ± 0.009	0.303 ± 0.008	361.97
<i>IterClip-MD</i>	0.318 ± 0.004	0.281 ± 0.002	0.298 ± 0.003	49393.81
<i>Mna-MD</i>	0.137 ± 0.008	0.259 ± 0.003	0.178 ± 0.002	6480.38
<i>DeepWalk</i>	0.043 ± 0.001	0.075 ± 0.001	0.054 ± 0.000	18756.13
<i>Metapath2vec</i>	0.071 ± 0.001	0.102 ± 0.002	0.084 ± 0.001	21314.67

- **Metapath2vec**: A meta-path based heterogeneous network embedding method [2]. Similar to *DeepWalk*, we use it to learn the embeddings and predict anchor links with a SVM. We report the best result of all intra-network meta diagrams.

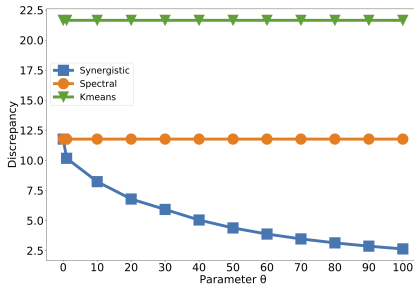
Comparison Methods with partition:

- **SHNA**: *SHNA* is the model proposed in this paper.
- **SpectralIter**: It implements the network partition using spectral clustering, and the sub-network alignment algorithm is the same as *SHNA*.
- **KmeansIter**: In *KmeansIter*, we directly use k-means clustering to partition the networks.

Some recent methods based on graph embedding and structural seeds like [3, 5, 22, 28] are designed for homogeneous graph and attributed networks, which are different from our problem definition. Therefore, we do not include all of them in comparison methods.

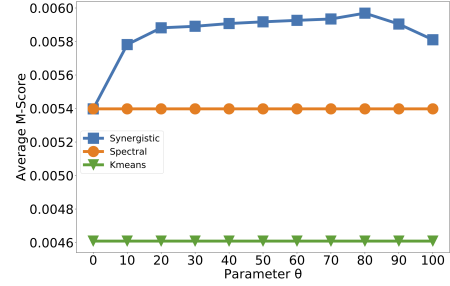
5.3 Experimental Results with Analysis

We will evaluate *network synergistic partition* and *parallel sub-network alignment* respectively together with *partitioned networks matching* that connects them.

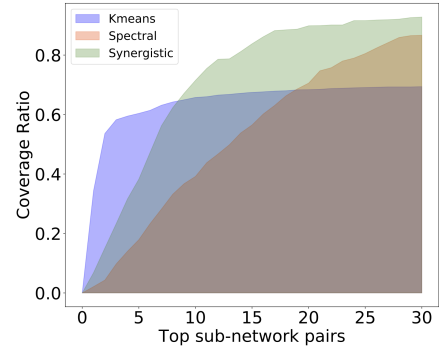
**Figure 4: Discrepancy with different parameter θ**

5.3.1 Network Synergistic Partition. To illustrate the effectiveness of *network synergistic partition*, we evaluate the results of the partition before performing the alignment. First, according to Definition 8, the discrepancy of different partition methods is displayed in Figure 4 which shows that *network synergistic partition* can minimize the discrepancy compared to the other two methods, and the effect of decreasing the discrepancy becomes more apparent as the weight of θ increases.

Besides, according to M-Score defined in Section 4.2, we can observe the average M-Score of Top 20 subnetwork pairs in Figure 5.

**Figure 5: The average M-Score of Top 20 sub-network pairs**

It essentially demonstrates that *network synergistic partition* has the best performance in the task of partitioning according to our requirements. Here we do not apply some classic metrics which are often used to evaluate the clustering result, because in *SHNA*, the partition is used by the next stage in order to better perform the alignment. Conventional metrics for clustering may not be effective here. For example, the partition method obtaining a better result based on conventional clustering metrics in every single network does not guarantee that partitioning multiple networks simultaneously can obtain well-matched sub-networks. From Figure 4 and Figure 5, we can find that as θ rises, the discrepancy is declining, but it does not bring the monotonous rise of the average M-Score. We will make detailed analysis through the discussion on the parameter θ in Section 5.5.

**Figure 6: The ratio of covering unknown anchor links**

5.3.2 Partitioned Networks Matching. Based on the partitioned networks, we can select the optimal sub-network pairs to perform the alignment and reduce the search space by ignoring links not exist in sub-network pairs. We choose the optimal sub-network pairs based on the ranking of M-Score, so to illustrate that our M-Score-based ranking is reliable and effective, we present related experimental results in Figure 6. Here, the x-axis denotes the number of selected top sub-network pairs, i.e., $x = 5$ means top 5 sub-network pairs in the ranking list are selected for alignment. For the y-axis indicator, we use the truth of the test set where the coverage ratio represents the coverage of positive anchor links in the test set. The reason why the coverage ratio is important is that if the positive anchor links in the test set are not included in selected sub-network pairs, there will be no chance to be predicted to positive in the alignment stage. In other words, positive anchor links are pruned as negative links. What needs to be explained is that the truth of the test set is only used for evaluation here. From Figure 6 we can find top

30 sub-network pairs from *network synergistic partition* can cover 92.8% positive anchor links which is higher than two other methods with the ratio 0.866 and 0.693 respectively. It proves not only the matching policy we used is effective which guarantees that 92.8% positive anchor links have the chance to be predicted, but also the effectiveness of *network synergistic partition*. From another perspective,

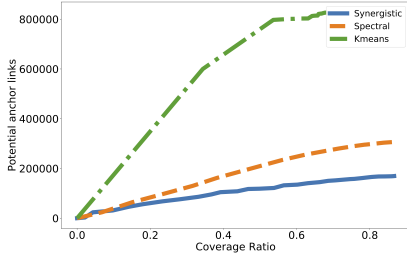


Figure 7: Upward trend of potential anchor links

tive, the upward trend of the number of potential anchor links with the rise of the coverage ratio can also reflect the performance of negative links pruning. We display the correlation in Figure 7, and it is obvious that the increasing rate of *network synergistic partition* is the slowest. It means more impossible and meaningless links are pruned by *SHNA*, which can affect both time complexity and prediction performance badly in the alignment stage.

5.3.3 Parallel Sub-network Alignment. The experimental results of the alignment stage are shown in Table 5. The methods we test in experiments can all output link prediction labels, and we will use F1, Recall and Precision as evaluation metrics. We will not present the metric Accuracy in the tables, because in such a class-imbalance setting of alignment tasks (the number of negative anchor links is much larger than positive links), the value of Accuracy is not so critical in evaluation. Firstly, we focus on the comparison among *SHNA* and *IterClip-MD*. We can find *SHNA* has a distinct advantage over *IterClip-MD* according to all four metrics. It means the alignment task achieves better performance after partition compared with no partition. We insist that *network synergistic partition* not only ensures the scalability but also effectively reduces the search space, that is, pruning impossible and meaningless links. In fact, these links will increase time complexity and affect the alignment stage badly simultaneously. Besides, the comparison among *SHNA*, *DeepWalk* and *Metapath2vec* verifies the effectiveness of inter-network meta diagram based features. It also reminds us that the heterogeneity of social networks needs to be handled in a precise way. Meanwhile, by comparing *SHNA*, *KmeansIter* and *SpectralIter*, we can demonstrate the partition stage is critical to the alignment stage. *SHNA* overperforms other methods significantly which verifies the effectiveness of *network synergistic partition* as well. We can observe that the Recall of *KmeansIter* is lower than *IterClip-MD* which means the partition based on simple k-means will prune lots of positive anchor links and miss them in the final alignment result.

5.4 Time and Convergence Analysis

The number of rounds used to convergence has a significant impact on running time. In Figure 8, we show the label matrix y changes in each iteration when built *SHNA* and *IterClip-MD* respectively. Here, the x axis denotes the iterations, and the y axis denotes the changes of y in sequential iterations i and $i - 1$, i.e., $\Delta y = \|y^i - y^{i-1}\|_1$.

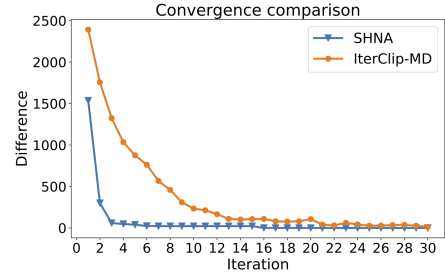


Figure 8: Convergence comparison

Because the iteration steps are executed simultaneously within each subnetwork pair for *SHNA*, we set the changes of y as the sum of the changes of subnetwork pairs, i.e., $\Delta y = \sum_{j=1}^m \|y_j^i - y_j^{i-1}\|_1$. From Figure 8, we can find *SHNA* can reach convergence in much fewer rounds than *IterClip-MD* where *IterClip-MD* needs more than 20 rounds to converge, but *SHNA* converges within 5 rounds.

The alignment time cost of different methods is listed in Table 5. *IterClip-MD* can achieve better prediction results than classic classification methods such as *Mna-MD*, but it costs the longest time. The reason lies in as the size of network increases, the number of rounds required for convergence increases together with each iteration time rises rapidly. *SHNA* has the best performance in the alignment time cost compared with *SpectralIter*, *KmeansIter*, because the partition results of *SpectralIter*, *KmeansIter* are uneven and some subnetworks are very large in size. In conclusion, *SHNA* can deduct the alignment time cost significantly based on faster convergence speed and parallel computing.

5.5 Parameter Analysis

Considering the objective function in Section 4.1.3 is composed of 3 parts, the parameters α , β , and θ reflect the extent to which each part influences the objective function, the proportional relationship among them must be more critical than the numerical values. Therefore, we fix $\alpha = \beta = 1$ because we assume that each network is equally important for partition, and tune θ . From Figure 4, we can observe that the discrepancy keeps monotonous decline with the rising of θ . But combining with Figure 5, the average M-Score gets the highest value when $\theta = 80$ instead of $\theta = 100$ which means the value of discrepancy is not the smaller the better for the synergistic partition. We further observe the result of partition with $\theta = 1000$ and find that in order to make the discrepancy infinitely close to 0, most of the anchor links in the training set are concentrated in one pair of sub-networks, and *IntraMD-Pro* is completely ignored. When θ is small, the result of partitioning will approximate spectral clustering, because *InterMD-Pro* will not work due to the insignificant weight. Further, the results of alignment with different θ can be observed in Figure 9 intuitively. The results show that F1 and Recall obtain the best performance when $\theta = 80$. On the contrary, when the value of θ is too large or too small, the performance becomes worse. In conclusion, θ should be in a suitable interval to make all parts of the objective function contribute to final results.

6 RELATED WORK

Network alignment has become an important research topic in recent years. Network alignment has concrete applications in various areas, e.g., protein-protein-interaction (PPI) and gene regulatory networks alignment in bioinformatics [17], chemical compound matching in chemistry [20], graph matching in combinatorial mathematics [11], figure matching and merging in computer vision [1],

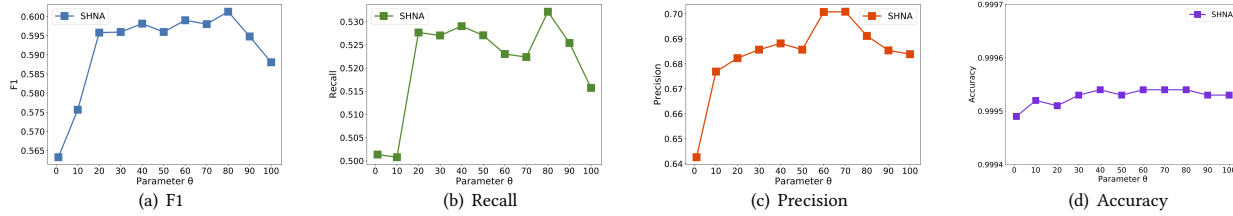


Figure 9: Performance Analysis with different parameter θ

and data schemas matching in data management [12]. In the field of social networks, network alignment provides a powerful tool for information fusion [27]. Zafarani et al. studies the cross-network user matching problem in [24] based on both users relationships and various attribute information. Kong et al. [6] propose to fully align social networks with the heterogeneous link and attribute information simultaneously based on a supervised learning setting. Zhang et al. [25, 26] propose to study the problem based on the PU learning setting. A manifold-based social network alignment method is proposed in [30].

Similarity measurement on heterogeneous networks has been widely studied. Sun introduces the concept of *meta path-based similarity* in [21]. The *meta path* suffers from the disadvantage that cannot describe rich semantics effectively. *Meta structure* [4] is proposed to similarity measure problem, but entities are constrained to the same type. Zhao [29] proposes the concept of *meta graph* for the single non-attribute network and extends the idea to recommendation problems.

Clustering-based community detection in online social networks is also related to our SHNA framework. Many different techniques are proposed to optimize certain measures, e.g., modularity function [14], and normalized cut [19]. A comprehensive survey of correlated techniques used to detect communities is given by Malliaros et al. [10] and a detailed tutorial on spectral clustering is provided by Luxburg [9]. These works are mostly studied based on homogeneous networks. Consensus clustering [7, 8, 13] is a sub-topic under clustering closing to our paper. However, these works mostly aim to find a single consensus clustering from fully mapped clustering solutions. Shao et al. [18] propose MMC which is based on collective spectral clustering with a discrepancy penalty across sources to deal with partially unknown mappings. In comparison, the purpose of partition in our paper is to obtain optimal sub-network matching instead of optimizing the discrepancy merely.

7 CONCLUSION

In this paper, we study the heterogeneous social network alignment problem and propose a novel two-stage framework SHNA to solve it. In order to address the extremely large search space, SHNA partitions the original networks with *network synergistic partition*. A group of inter- and intra-network meta diagrams are defined to constitute heterogeneous features. The metrics *Matching Score* is proposed to obtain optimal sub-network matching results. With the support of the partition stage, not only the search space is greatly reduced, but also the alignment within sub-network pairs can be performed in parallel. Extensive experiments are conducted on real-world networks Foursquare and Twitter. The experiment results demonstrate that SHNA has outstanding performance compared with the state-of-the-art baseline methods in both *network synergistic partition* and *parallel sub-network alignment* stage.

8 ACKNOWLEDGEMENT

This work is partially supported by NSF through grant IIS-1763365 and by FSU.

REFERENCES

- [1] Mohsen Bayati, Margot Gerritsen, David F Gleich, Amin Saberi, and Ying Wang. Algorithms for large, sparse network alignment problems. In *ICDM*, 2009.
- [2] Yuxiao Dong, Nitesh V Chawla, and Ananthram Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 2017.
- [3] Xingbo Du, Junchi Yan, and Hongyuan Zha. Joint link prediction and network alignment via cross-graph embedding. In *IJCAI*, 2019.
- [4] Zhipeng Huang, Yudian Zheng, Reynold Cheng, Yizhou Sun, Nikos Mamoulis, and Xiang Li. Meta structure: Computing relevance in large heterogeneous information networks. In *KDD*, 2016.
- [5] Ehsan Kazemi, S Hamed Hassani, and Matthias Grossglauser. Growing a graph matching from a handful of seeds. *Vldb*, 2015.
- [6] Xiangnan Kong, Jiawei Zhang, and Philip S Yu. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*, 2013.
- [7] Tao Li and Chris Ding. Weighted consensus clustering. In *SDM*, 2008.
- [8] E. Lock and D. Dunson. Bayesian consensus clustering. *Bioinformatics*, 2013.
- [9] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 2007.
- [10] Fragkiskos D Malliaros and Michalis Vazirgiannis. Clustering and community detection in directed networks: A survey. *Physics Reports*, 2013.
- [11] Fredrik Manne and Mahantesh Halappanavar. New effective multithreaded matching algorithms. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, 2014.
- [12] S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In *ICDE*, 2002.
- [13] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 2003.
- [14] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 2004.
- [15] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.
- [16] Yuxiang Ren, Charu C Aggarwal, and Jiawei Zhang. Meta diagram based active social networks alignment. In *ICDE*, 2019.
- [17] B. Seah, S. Showmick, and C. Dewey Jr. Dualaligner: a dual alignment-based strategy to align protein interaction networks. *Bioinformatics*, 2014.
- [18] Weixiang Shao, Jiawei Zhang, Lifang He, and S Yu Philip. Multi-source multi-view clustering via discrepancy penalty. In *IJCNN*, 2016.
- [19] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, 2000.
- [20] Aaron Smalter, Jun Huan, and Gerald Lushington. Gpm: A graph pattern matching kernel with diffusion for chemical compound classification. In *2008 8th IEEE International Conference on Bioinformatics and BioEngineering*, 2008.
- [21] Y. Sun, J. Han, X. Yan, P. Yu, and T. Wu. Paths: Meta path-based top-k similarity search in heterogeneous information networks. *Vldb*, 2011.
- [22] Chenxu Wang, Zhiyuan Zhao, Yang Wang, Dong Qin, Xiapu Luo, and Tao Qin. Deepmatching: A structural seed identification framework for social network alignment. In *ICDCS*, 2018.
- [23] J. Zhang and P. Yu. Community detection for emerging networks. In *SDM*, 2015.
- [24] Reza Zafarani and Huan Liu. Connecting users across social media sites: a behavioral-modeling approach. In *KDD*, 2013.
- [25] Jiawei Zhang, Jianhui Chen, Junxing Zhu, Yi Chang, and Philip S Yu. Link prediction with cardinality constraint. In *WSDM*, 2017.
- [26] Jiawei Zhang and S Yu Philip. Integrated anchor and social link predictions across social networks. In *IJCAI*, 2015.
- [27] Jiawei Zhang and Philip S Yu. *Broad Learning Through Fusions: Applications in Machine Learning*. 2019.
- [28] Jing Zhang, Bo Chen, Xianming Wang, Hong Chen, Cuiping Li, Fengmei Jin, Guojie Song, and Yutao Zhang. Mego2vec: embedding matched ego networks for user alignment across social networks. In *CIKM*, 2018.
- [29] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. Meta-graph based recommendation fusion over heterogeneous information networks. In *KDD*, 2017.
- [30] W. Zhao, S. Tan, Z. Guan, B. Zhang, M. Gong, Z. Cao, and Q. Wang. Learning to map social network users by unified manifold alignment on hypergraph. *IEEE transactions on neural networks and learning systems*, 2018.