Project Report: The implementation of MNA and a novel model CluITER for Network Alignment

Bowen Li
Florida State University
Tallahassee, FL, USA
bl17j@my.fsu.edu

Yuxiang Ren
IFM Lab
Florida State University
Tallahassee, FL, USA
yren@cs.fsu.edu

ABSTRACT

Online social networks have gained great success in recent years. Some online social networks only involving users and social links among users can be represented as homogeneous networks. Meanwhile, some other social networks containing abundant information, which include multiple kinds of nodes and complex relationships, can be denoted as heterogeneous networks. Network alignment aims at inferring a set of anchor links matching the shared entities between different information networks, which has become a prerequisite step for effective fusion of multiple information sources. Link prediction problems have extensive applications in real-world social networks and many concrete social services can be cast as link prediction tasks, e.g., friend and location recommendations can all be solved as the problem of predicting social links among users and the location links between users and locations. Link prediction problems have been an important research topic for many years and a large number of different methods have been proposed so far. In this report, we study the network alignment problem to fuse online social networks specifically and summaries some existing link prediction methods for both homogeneous and heterogeneous networks. Based on the paper *Inferring Anchor Links across Heterogeneous Social Networks, CIKM'13*, we also implement the progress of link prediction problems across multiple aligned heterogeneous networks and describe the algorithm and implemetation details in the first section of the report. In addition, a novel network alignment model, namely CluITER is introduced in the second part of this report. Model CluITER will use *Mutual Clustering* algorithm to divide a large-scale network into sub-networks and uses *greedy link selection* for anchor link cardinality filtering within every sub-networks, then combines link prediction results in different sub-networks to implement large-scale Network Alignment.

KEYWORDS

Heterogeneous Social Network, Multi-Network, Anchor Links, Cluster Analysis

ACM Reference format:
Bowen Li and Yuxiang Ren. 2018. Project Report: The implementation of MNA and a novel model CluITER for Network Alignment. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 12 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

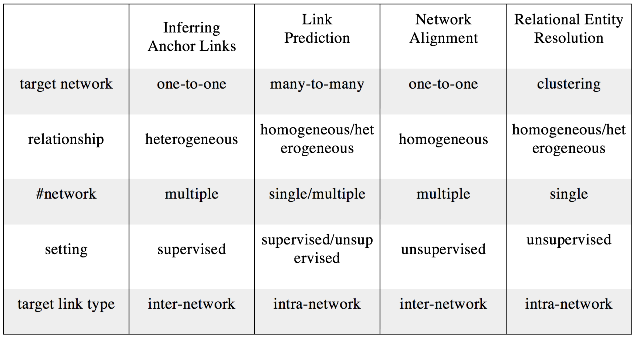
1 INTRODUCTION

Social network analysis[7], especially the link prediction problem in social networks, has been intensively studied in recent years. Typically some similarity measures between pair of nodes are used. Upon whether considering the label information, there are two types of approaches: unsupervised and supervised. Liben-Nowell and Kleinberg[6] developed unsupervised link prediction methods based upon several topological features of a co-author network. Many supervised link prediction methods[3] have also been proposed in recent years, where the features used in unsupervised approaches can be directly used to train a binary classification model for link prediction. Here are two supervised link prediction methods proposed in this paper, Second Order Cone Programming (SOCP) and Pairwise Kernel Approach, which based on vertex attributes and models that use Bayesian concepts. However, the performance of the algorithm entirely depends on the degree to which the network agree to the proposed graph evolution model. There are many other recent efforts on link prediction problem in social networks. Lichtenwalter et. al. [5]have a detailed discussion over different challenges of link prediction problem. Scellato et. al. proposed to use place features for link prediction in location-based social networks. [1] proposed a supervised random walk method for link predictions in social networks. In addition, another line of research works study the link prediction problems on multiple networks or domains. But different with the paper above on link predication and network alignment, for our project, we assume that anchor links are one-to-one relationships among the the two sets of user accounts (i.e., no two edges share a common endpoints) and a small number of anchor links across networks are known beforehand. By explicitly consider the users heterogeneous data within the networks, i.e., social, spatial, temporal and text information, our method can effectively predict the anchor links w.r.t. one-to-one constraint across multiple heterogeneous social networks.

Online social networks usually have very complex network structures, involving different categories of nodes and links. For instance, in online social networks, like Twitter and Foursquare, users can perform various kinds of social activities, e.g., following other users, writing posts. Viewed in such a perspective, their network structures will contain multiple types of nodes and links, i.e., "User",

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn



	Inferring Anchor Links	Link Prediction	Network Alignment	Relational Entity Resolution
target network	one-to-one	many-to-many	one-to-one	clustering
relationship	heterogeneous	homogeneous/het erogeneous	homogeneous	homogeneous/het erogeneous
#network	multiple	single/multiple	multiple	single
setting	supervised	supervised/unsup ervised	unsupervised	unsupervised
target link type	inter-network	intra-network	inter-network	intra-network

Figure 1: Summary of related problems

“Post” (node types), and “Follow”, “Write” (link types). Users’ personal preference may steer their online social activities in making friends and writing posts, and the network structure can provide insightful information for differentiating users between networks. Furthermore, the nodes in online social networks can be also attached with various types of attributes. For example, these written post nodes can contain words, location check-ins and timestamps (attribute types), which can help provide complementary information for inferring users’ language usage, spatial and temporal activity patterns respectively. Based on such an intuition, both the network structure and attribute information should be incorporated in the network alignment model building.

For the existing network alignment models, lots of them are based on supervised learning setting [4], which aim at building classification/regression models with a large set of pre-labeled anchor links to infer the remaining unlabeled ones (where the existing and non-existing anchor links are labeled as positive and negative instance respectively). For the network alignment task, pre-labeled anchor links can provide necessary information for understanding the patterns of aligned user pairs in their information distribution, especially compared with the unsupervised alignment models [2, 12]. However, for the real-world online social networks, cross-network anchor link labeling is not an easy task, since it requires tedious user accounts pairing and manual user background checking, which can be quite time-consuming and expensive. Besides, because the anchor link distribution will be very sparse in real-world social networks, we often need negative sampling for data sets which affect the practicality of the model.

The network alignment problem [2] denotes the task of inferring the set of anchor links [4] between the shared information entities in different networks. Network alignment has concrete applications in real-world, which can be applied to discover the set of shared users between different online social networks [4, 12], identify the common protein molecules between different protein-protein-interaction (PPI) networks [2, 8, 9], and find the mappings of POIs (places of interest) across different traffic networks [12].

In the real-world online social networks, cross-network anchor link labeling is not an easy task, because it requires tedious user accounts pairing and manual user background checking, which can be quite time-consuming and expensive. At the same time, because the anchor link distribution will be very sparse in real-world social networks, we often need negative sampling for data sets which used in previous models which affect the practicality of the models. By this context so far, no research works have studied the heterogeneous network alignment problem based on the combination of spectral clustering and anchor link cardinality filtering. The Anna problem is a novel yet difficult task, and the challenges mainly come from three perspectives, e.g., *network heterogeneity*, *lack of training data*, and *one-to-one constraint.*

- *Network Heterogeneity*: According to the descriptions aforementioned, both the complex network structure and the diverse attributes have concrete physical meanings and can be useful for the social network alignment task. To incorporate such heterogeneous information in model building, a easily extensible approach is required to handle the network structure and attribute information in a unified analytic.
- *Lack of Training Data*: To overcome the lack of training data problem, in Anna, besides the labeled anchor links, it also allows models to query for extra labels of unlabeled instances subject to a pre-specified budget. Both the effective incorporation of information about unlabeled anchor links and a desired query strategy for selecting good unlabeled candidates in the model building are still open problems.
- *One-to-One Cardinality Constraint*: Last but not the least, the anchor link instances to be inferred are not independent actually in the networked data scenario. The *one-to-one* cardinality constraint on anchor links will limit the number of anchor links incident to the user nodes [10], which renders the information of positive and negative anchor links to be imbalanced. For each user, if one incident anchor link is identified to be positive, the remaining incident anchor links will be negative by default. Viewed in such a perspective, positive anchor links bear far more information compared with negative anchor links. Effectively maintaining and utilizing such a constraint on anchor links label query and model building is necessary.

Based on these challenges, we decide to propose a new model CluITER which make use of spectral cluster to divide the large network into multiple sub-networks. Based on the one-to-one constraint, the time complexity will be decreased effectively and the accuracy will be increased if the performance of clustering is good enough. The cluster algorithm in our design will be an iterative precess based on some initial training samples. The core part of CluITER is the cluster algorithm which can effectively reduce the hypothesis space, and it enable use to process sparse networks directly instead of conducting negative sampling in advance. After clustering, CluITER will use *greedy link selection* for anchor link cardinality filtering within every sub-networks and output combined link prediction results from different sub-networks.

In this report, our contributions can be summarized as follows:

- *Implementation of MNA*: Based on the paper [4], we implement the model MNA proposed in this papar along with every baseline method. Extensive comparative experiments are conducted in the same way as the method from the paper [4].
- *The creative model CluITER*: To overcome the lack of shortages we found from MNA, we porpose a novel model CluITER. The description and derivation of CluITER are introduced in great detail.

The report is organized as follows: we introduce the the model *Mna* from [4] which we implement in Section2; we describe the creative model CluITER for link prediction in Section 3;. Finally, we conclude the report along with our achievement in Section4.

2 TEAM WORK DIVISION

In our project, Yuxiang collect the datasets at first. . Bowen extract the feature vector from the data, and sample the data according to experiment setting. After that, Bowen also build up the model proposed in [4] and conduct the related experiments. Finally, we evaluate the result together and write the report. As a Ph.D. student, Yuxiang work on the optimized model in Section 4 and define, interpret and derive the modelCluITER at the theoretical level from a detailed view.

3 THE IMPLEMENTATION OF MNA

In this section, we will give a detailed description about the implementation of the MNA(Multi-Network Anchoring) algorithm. First, we discuss how to extract heterogeneous features across multiple network. Then we discuss that using âĂŸstable matchingâĂŹ to constrain the predicted result to one-one and how to implement the MNA algorithm. After that, we make the experiment related to MNA and evaluate the results.

The datasets of our projects come from Twitter and Foursquare which are famous online social networks right now. They include posts, tips, users follower and following, users' profiles and so on.

3.1 Problem Formulation

We can use $\mathbf{G}_s$ and $\mathbf{G}_t$ stand for the source and target network. These network are both heterogeneous social networks. Each heterogeneous social network is an undirected graph. The source network $\mathbf{G}_s$ = ($\mathbf{V}_s$,$\mathbf{E}_s$) contains different types of nodes and links. $\mathbf{V}_s$ = $\mathbf{U}_s \cup \mathbf{L} \cup \mathbf{T} \cup \mathbf{W}$ is the set of nodes in the source network, which includes four types of nodes. $\mathbf{U}_s$= {u_1^s,u_2^s,...,u_n^s} : the set of user accounts;$\mathbf{L}$ = {l_1,l_2,...} : the set of different locations or places, where users have published their posts at; $\mathbf{T}$ = {t_1,t_2,...} : the set of time slots that users have published posts at; $\mathbf{W}$ = {w_1,w_2,...} : the set of words that users have used in their posts. $\mathbf{E}_s \subset \mathbf{V}_s$ X $\mathbf{V}_s$ is the edges of different types in the heterogeneous social network $\mathbf{G}_s$. $\Gamma_s \subset \mathbf{E}_s$ is the set of user pairs that are friends with each other in network .

For the target network, we define it as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathbf{U}_t$ is the set of user accounts. For the location $\mathbf{L}$, time slots $\mathbf{T}$ and words $\mathbf{W}$, we assume that target network and source network are the same.

Anchor Link Prediction: For the source and target network $\mathbf{G}_s$ and $\mathbf{G}_t$, we can know a small set of known anchor links between the users accounts in two networks before, which can be written A = {(u_i^s,u_j^t), $u_i^s \in u^s$, $u_j^t \in u^t$ }. Because anchor links are one-to-one relationships between user accounts in u^s and u^t, i.e., and no two anchor links share a same user account. (u_i^s,u_j^t) denotes that the two user accounts belong to the same user. So if we want to make anchor link prediction, we just need to know whether there is an anchor link between a pair of user accounts u_i^s and u_i^s, where $u_i^s \in u^s$, $u_j^t \in u^t$.

3.2 Multi-Network Social Features

For users in different social networks, like Twitter and Foursquare, they have different follower and friends. We can extract the social features from these relationships. Most exiting social features focus on single network setting, like Common Neighbors, JaccardâĂŹs coefficient and Adamic/Adar measure, we need to extend them into multiple networks.

Table 1: Properties of the Heterogeneous Networks

		network	
	property	**Twitter**	**Foursquare**
# node	user	500	500
	tweet/tip	1,102,524	5,424
	location	61,119	4,512
# link	friend/follow	6,902	4,512
	write	1,102,524	5,424
	friend/follow	89,021	5,424

Based on users' following on Twitter and Foursquare, we can easily find each user's neighbors which are userâĂŹs follower or following. For two users, one is in Twitter, and the other one is in Foursquare, we need to compare their neighbors. Because we have known some anchor links, we can know the number of their neighbors who are the same users. We use this number as the Extend Common Neighbors.

- *For the Extend Jaccard's coefficient of two users*: Extend Jaccard's coefficient = (Extend Common Neighbors)/(Total numbers of two users' neighbor).
- *For the Extend Adamic/Adar Measure*: Extend Adamic/Adar Measure = $\log^{-1}(Totalnumbersoftwousersfineighbor/2)$

Finally, we get 3 measures to evaluate the social features.

3.2.1 *Spatial distribution features.* We notice that users in different social networks have almost the same location in their profile. We can make use of the similarity between the spatial distributions of two user accounts from different social networks to help locate the same user. By using the users's location information in the dataset, we can certainly judge the distance between two users. Given a certain threshold(in our project, it's 100 mile), we could evaluate the spatial distribution features.

3.2.2 *Temporal distribution features.* We also notice that users in different social networks usually publish posts at similar time slots in real-life, such as hours after working and before sleeping, etc. Such temporal distribution indicates the user's online activity patterns. The temporal distribution of different user accounts can also help us find the anchor links between two networks.

We first need to gather each user's tweets and tips. Base on the time when the tweets and tips are posted, we could get a collection of time for each user. We divide one day into 24 time slots, each slot is one hour. Then we can convert the time collect into 24-dimensional vector, each element stands for the number of tweets or tips in related time slot. We extract similar measures about the spatial distributions for two user accounts: 1) the number of shared time slots when publishing posts; 2) the cosine similarity between the two vectors of temporal activities.

3.2.3 *Text content features.* The text content of posts by users in different social networks can also hint for the anchor links, because different users may have different choices of words in their posts. In order to computer the similarities of 2 users content, we should

Table 2: Performance comparison of different methods for inferring anchor links. We use different imbalance ratios in both training and test sets. (imbalance ration = positive account pairs / negative account pairs).

		Imblance Ratio							
metrics	methods	1	2	3	4	10	20	30	40
Accuracy	MNA	**0.87±0.033**	**0.84±0.021**	**0.79±0.021**	**0.81±0.008**	**0.92±0.006**	**0.95±0.004**	**0.96±0.033**	**0.97±0.024**
	MNA-no	0.68±0.04	0.74±0.020	0.75±0.023	0.77±0.010	0.85±0.005	0.85±0.041	0.87±0.012	0.91±0.013
	Social	0.62±0.025	0.68±0.01	0.70±0.027	0.72±0.011	0.83±0.005	0.89±0.01	0.90±0.054	0.97±0.082
	Spatial	0.64±0.024	0.66±0.006	0.71±0.01	0.77±0.032	0.87±0.002	0.88±0.001	0.90±0.002	0.88±0.007
	Text	0.53±0.071	0.56±0.054	0.53±0.076	0.57±0.089	0.67±0.093	0.68±0.002	0.72±0.013	0.74±0.041
	Time	0.54±0.051	0.65±0.013	0.74±0.014	0.80±0.004	0.90±0.111	0.90±0.087	0.96±0.02	0.96±0.023
F1	MNA	0.77±0.050	**0.7±0.051**	**0.62±0.011**	**0.57±0.098**	**0.37±0.130**	0.27±0.019	0.19±0.023	**0.17±0.024**
	MNA-no	**0.78±0.02**	0.68±0.054	0.51±0.122	0.49±0.116	0.24±0.011	0.21±0.019	0.16±0.489	0.15±0.023
	Social	0.71±0.042	0.69±0.046	0.46±0.149	0.42±0.014	0.26±0.106	0.19±0.016	0.17±0.054	0.11±0.082
	Spatial	0.65±0.045	0.62±0.015	0.49±0.029	0.41±0.016	0.34±0.014	**0.3±0.012**	**0.24±0.022**	0.125±0.019
	Text	0.5±0.042	0.52±0.013	0.51±0.048	0.43±0.054	0.31±0.013	0.17±0.076	0.15±0.034	0.11±0.013
	Time	0.5±0.177	0.51±0.107	0.45±0.098	0.41±0.031	0.32±0.076	0.23±0.078	0.16±0.023	0.13±0.014
Precision	MNA	**0.96±0.04**	**0.92±0.017**	**0.85±0.198**	**0.87±0.142**	**0.6±0.489**	**0.4±0.400**	**0.32±0.490**	**0.31±0.04**
	MNA-no	0.86±0.004	0.83±0.018	0.73±0.234	0.68±0.141	0.5±0.490	0.2±0.015	0.18±0.489	0.16±0.061
	Social	0.81±0.049	0.78±0.056	0.71±0.133	0.53±0.072	0.42±0.564	0.19±0.4	0.18±0.030	0.19±0.082
	Spatial	0.48±0.048	0.58±0.018	0.38±0.033	0.41±0.018	0.34±0.025	0.32±0.024	0.25±0.004	0.14±0.056
	Text	0.55±0.081	0.43±0.012	0.46±0.019	0.36±0.014	0.33±0.032	0.18±0.012	0.14±0.09	0.11±0.123
	Time	0.43±0.227	0.38±0.013	0.33±0.087	0.31±0.077	0.21±0.055	0.18±0.089	0.17±0.04	0.14±0.112
Recall	MNA	**0.64±0.006**	**0.57±0.068**	**0.47±0.088**	**0.42±0.071**	0.285±0.076	**0.25±0.01**	0.21±0.012	**0.2±0.01**
	MNA-no	0.53±0.06	0.51±0.066	0.31±0.078	0.29±0.065	0.2±0.054	0.18±0.065	**0.23±0.13**	0.15±0.087
	Social	0.64±0.073	0.54±0.07	0.35±0.144	0.31±0.07	0.15±0.01	0.14±0.076	0.21±0.012	0.16±0.046
	Spatial	0.51±0.06	0.53±0.016	0.40±0.024	0.32±0.013	**0.38±0.009**	0.21±0.028	0.16±0.015	0.07±0.012
	Text	0.48±0.048	0.41±0.023	0.31±0.027	0.291±0.078	0.17±0.021	0.13±0.078	0.11±0.098	0.11±0.012
	Time	0.51±0.092	0.32±0.212	0.278±0.038	0.31±0.023	0.13±0.013	0.11±0.033	0.07±0.123	0.06±0.041

collect each user's tweets or tips and gather their content into a bag-of-words vector. For the vector, we should delete the punctuation and stop words like 'a', 'the' and so on. Then we weight the vector by TF-IDF.Then for each pair user accounts, we compute two kinds of similarities as features: 1) the inner product of the two vectors; 2) the cosine similarity of the two vectors.

3.3 Inferring anchor links w.r.t. one-to-one constraints

After extracting all the four types of heterogeneous features in the previous section, we can train a binary classifier(we use SVM in our project)for anchor link prediction. However, in the inference processes, the predictions of SVM cannot be directly used as anchor links due to it is designed for constraint-free setting. But the anchor links are certainly one-one relation. Therefore, in our project, we implement MNA(Multiple-Network Anchoring) in infer anchor links with the consideration of one-to-one constrain. The proposed MNA method for anchor link prediction is described as follows.

In each iteration, we first randomly select a free user account from the source network. Then we get the most prefered user node $u_j^t \in u^t$ by $u_i^s \in u^s$ in its preference list $p(u_i^s)$. We then remove u_j^t from the preference list. If u_j^t is also a free account, we add the pair of accounts (u_i^s, u_j^t) into the current solution set. Otherwise, u_j^t is already occupied with u_p^s in current solution set . We then examine the preference of u_j^t. If u_j^t also prefers u_p^s over u_i^s, we remove the pair (u_i^s, u_j^t) by replacing the pair (u_p^s, u_j^t) in the solution set with the it. Otherwise, we start the next iteration to reach out the next free node in the source network. The algorithm stops when all the users in the source network are occupied, or all the preference lists of free accounts in the source network are empty.

3.4 Experiments

3.4.1 *Data Preparation.*

In order to evaluate the performance of the proposed approach for anchor link prediction, we test the algorithm on two real-world social networks as summarized in Table 1. We choose Twitter and Foursquare as our data sources because public tweets and Foursquare tips can be easily collected by their APIs.

- *Foursquare*: The first network we ude is the Foursquare website. We collect a dataset consisting of 500 users and 5,424 ips of these users. For each tip, the location data (latitude and longitude) as well as the timestamp are available. Moreover, Foursquare network also provides data about whether one user is following or a friend of another user. These links can indicate the social relationship among the users.
- *Twitter*: The second network we use is Twitter. We choose 500 users which correspond to the 500 users in Foursqure and 1,102,524 tweets of the users. In Twitter network, all tweets include time data, but just part of them include location data. In total, we have 61,119 tweets with latitude and longitude, which is about 5.5% of all the tweets.

In order to conduct experiments, we pre-process these raw data to obtain the ground-truth of users' anchor links. In Foursquare network, we can collect some users' Twitter IDs in their account pages. We use these information to build the ground-truth of anchor links between user accounts across the two networks. If a Foursquare user has shown his/her Twitter ID in the website, we treat it as an anchor link between this user's Foursquare account and Twitter account.

3.4.2 *Comparative Methods.*

In order to study the effectiveness of the proposed approach, we compare our method with baseline methods. The compared methods are summarized as follows:

- *Multi-Network Anchoring(Mna)*: The proposed method in our project. Mna can explicitly exploit four types of information from both networks to infer anchor links. In addition, Mna incorporates the one-to-one constraint in the inference process.

- *Mna without one-to-one constraints (Mna-no)*: MNA without the one-to-one constraints in the inference step. The label predictions of the base learners are directly used as final predictions for anchor link prediction.
- *Supervised link prediction methods*: In order to verify the effectiveness of different kinds of feature sets, we test supervised link prediction methods using four types of feature sets separately. 'Social' indicates the supervised link prediction method using social features only. 'Spatial' uses only spatial features. 'Time' uses temporal features. 'Text' uses text content features only.

For fair comparisons, SVC from the packet sklearn is used as the base classifier for all the compared methods.

3.4.3 Evaluation. In the experiments, we partition users into 5-fold to make 5-fold cross validation: each fold is used as train set and other rest 4 folds are used as testing data. Then we report the average results and standard deviations bases on 5-fold cross validation.

In real-world social networks, there will always be a small number of known/labeled anchor links. In the first experiment, we study the performance of the proposed MNA method on anchor link prediction with different number of labeled anchor links in each fold. In each round of the cross validation, we randomly sample $\{10, 20, \ldots, 80\}$ users from the training fold, and use them as the labeled anchor links. We set the ratio of negative and positive pairs to 1:1. The results are reported in Table 3. When there are some anchor links known in the two social networks, Multi-Network Anchoring (MNA) method has the best outperforms compared with other baseline methods. And with the increasement of labeled links, the performance of all methods will also increase.

The data samples in real-world social network are usually imbalanced. In the second experiment, we test the performance of MNA with different imbalanced ratio datasets. In each round of the cross validation, we sample pairs of user accounts as the data samples according to different imbalance ratios. Table 2 shows the performances of each of the models. With the increasing of the number of negative pair, the performance of all methods will reduce, but MNA still has the best performance compared with other methods.

Moreover, in order to test the contribution of different type of features, we also tested the performances of baselines with different features. The result is shown in Figure 2. In Figure 2, we can see that MNA has the best performance. And we also notice that the performance of MNA is also better than MNA-no. It shows that by incorporating the one-to-one constraint in the inference process can certainly improve the performance of anchor link prediction.

3.4.4 Conclusion. In the implementation part, Bowen implements the MNA methods for anchor link prediction and makes related experiments to compare the MNA with other method. We studied two real-world social networks: Foursquare and Twitter, finding the anchor link between these two network. By explicitly considering heterogeneous features of users within the networks, i.e., social, spatial, temporal and text information and the one-to-one constraint of the result, the MNA can effectively predict the anchor links w.r.t. one-to-one constraint across multiple heterogeneous social networks and has the best performance than other methods.

4 PROPOSED METHOD CLUITER

In this section, we will introduce the framework CluITER to address the network alignment problem. At first, we will define *intra-network meta diagram* which is used to measure closeness among users in single network and *inter-network meta diagram* which is the basis of promixity features in network alignemnt. After that, we will introduce the innovative model in great detail which actually involves two main steps: (1) mutual clustering across mutiple networks, and (2) network alignment subject to cardinaity constraints.

At first, we need to provide several critical concept definitions formally. Because the problem and scenario are different in this report for our proposed model CluITER comparing with the one in [4], we have to formulate them in a formal way.

4.1 Terminology Definition

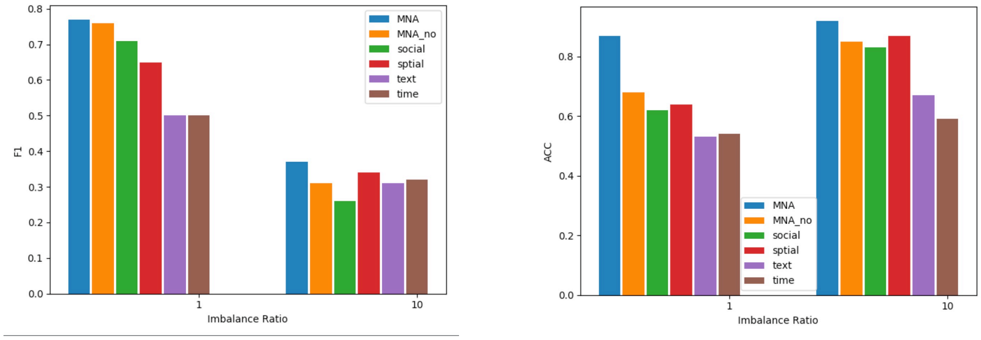
The networks we study can be defined as *attributed heterogeneous social networks.*

Definition 4.1. (Attributed Heterogeneous Social Networks): The *attributed heterogeneous social network* can be represented as $G = (\mathcal{V}, \mathcal{E}, \mathcal{T})$. In this representation, $\mathcal{V} = \bigcup_i \mathcal{V}_i$ is the set of different nodes, while $\mathcal{E} = \bigcup_i \mathcal{E}_i$ represent the set of complex links in the network. Besides, the set $\mathcal{T} = \bigcup_i \mathcal{T}_i$ represents attributes which are associated with $\mathcal{V}_i$, where $\mathcal{V}_i \in \mathcal{V}$, and $\mathcal{T}_i$ denotes the i_{th}-type of attributes.

Among multiple *attributed heterogeneous social networks*, if there exist shared users, we can define them as *aligned attributed heterogeneous social networks*

Definition 4.2. (Aligned Attributed Heterogeneous Social Networks): Given *attributed heterogeneous social networks* $G^{(1)}$, $G^{(2)}$, $\cdots$, and $G^{(n)}$, and common users are shared among them, we can difine them as the *aligned attributed heterogeneous social networks* $\mathcal{G} = \Big((G^{(1)}, G^{(2)}, \cdots, G^{(n)}), (\mathcal{A}^{(1,2)}, \cdots, \mathcal{A}^{(1,n)}, \mathcal{A}^{(2,3)}, \cdots, \mathcal{A}^{(n-1,n)})\Big)$, and $\mathcal{A}^{(i,j)}$ is the set of undirected anchor links between $G^{(i)}$ and $G^{(j)}$ which link common users.

In order to illustrate the definition of *aligned attributed heterogeneous social networks*, we provide an example which contains two *aligned attributed heterogeneous social networks*: Foursquare and Twitter. We can represent it as $\mathcal{G} = ((G^{(1)}, G^{(2)}), \mathcal{A}^{(1,2)})$, where $G^{(1)}$ represents Foursquare and $G^{(2)}$ is Twitter. The Foursquare $G^{(1)}$ 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 $\mathcal{P}^{(1)}$ representing the set of users and the set of posts in the network respectively. $\mathcal{E}^{(1)} = \mathcal{E}^{(1)}_{u,u} \cup \mathcal{E}^{(1)}_{u,p}$ contains the set of social links among users and the set of write links between users and posts. $\mathcal{T}^{(1)} = \mathcal{T}^{(1)}_l \cup \mathcal{T}^{(1)}_t$ denotes various attributes extracted from the posts in $\mathcal{P}^{(1)}$ including location checkins $\mathcal{T}^{(1)}_l$ and timestamps $\mathcal{T}^{(1)}_t$ in this example. The Twitter can be represented in the similar format as Foursquare that is $G^{(2)} = (\mathcal{V}^{(2)}, \mathcal{E}^{(2)}, \mathcal{T}^{(2)})$. User anchor links connecting to



(a) F1

(b) Accuracy

Figure 2: Performance of inferring anchor links with different sets of features.

Table 3: Performance comparison of different methods for inferring anchor links. We use different number of labeled anchor links in the training set

		number of labeled anchor links							
metrics	methods	10	20	30	40	50	60	70	80
Accuracy	MNA	**0.8±0.048**	0.82±0.267	0.813±0.287	0.824±0.223	0.83±0.123	0.85±0.133	**0.87±0.025**	**0.88±0.022**
	MNA-no	0.78±0.034	0.82±0.050	0.80±0.021	0.80±0.070	0.81±0.122	0.86±0.123	0.85±0.029	0.87±0.019
	Social	0.81±0.056	**0.85±0.019**	**0.85±0.04**	**0.85±0.048**	**0.85±0.006**	**0.86±0.003**	0.87±0.011	0.85±0.023
	Spatial	0.74±0.026	0.79±0.013	0.75±0.098	0.76±0.018	0.75±0.013	0.78±0.014	0.77±0.009	0.76±0.012
	Text	0.56±0.025	0.52±0.060	0.53±0.021	0.57±0.065	0.54±0.062	0.58±0.092	0.55±0.06	0.61±0.032
	Time	0.52±0.07	0.48±0.003	0.49±0.073	0.52±0.04	0.51±0.018	0.51±0.009	0.49±0.029	0.49±0.017
F1	MNA	**0.73±0.077**	**0.78±0.029**	**0.78±0.047**	0.79±0.033	0.79±0.020	**0.84±0.020**	**0.86±0.033**	**0.93±0.022**
	MNA-no	0.69±0.004	0.77±0.018	0.78±0.072	0.78±0.022	0.81±0.022	0.84±0.044	0.83±0.035	0.845±0.023
	Social	0.67±0.091	0.71±0.032	0.73±0.027	**0.823±0.020**	**0.83±0.008**	0.82±0.004	0.86±0.014	0.82±0.036
	Spatial	0.64±0.047	0.73±0.012	0.67±0.022	0.68±0.014	0.66±0.023	0.71±0.014	0.69±0.017	0.67±0.016
	Text	0.49±0.061	0.54±0.023	0.54±0.082	0.4±0.157	0.58±0.157	0.68±0.189	0.65±0.171	0.76±0.119
	Time	0.57±0.167	0.37±0.222	0.55±0.172	0.57±0.185	0.54±0.074	0.56±0.232	0.69±0.07	0.67±0.016
Precision	MNA	**0.72±0.33**	**0.76±0.03**	**0.78±0.08**	**0.84±0.034**	**0.87±0.03**	**0.88±0.021**	**0.89±0.032**	**0.93±0.012**
	MNA-no	0.66±0.22	0.67±0.056	0.72±0.098	0.73±0.023	0.73±0.076	0.74±0.032	0.77±0.012	0.81±0.098
	Social	0.57±0.043	0.61±0.02	0.63±0.07	0.68±0.003	0.71±0.008	0.72±0.004	0.72±0.114	0.74±0.146
	Spatial	0.47±0.124	0.47±0.106	0.52±0.098	0.53±0.093	0.53±0.09	0.63±0.032	0.67±0.142	0.71±0.028
	Text	0.6±0.065	0.53±0.058	0.55±0.111	0.59±0.198	0.73±0.223	0.68±0.186	0.65±0.171	0.76±0.119
	Time	0.49±0.085	0.49±0.04	0.48±0.025	0.51±0.050	0.50±0.013	0.5±0.008	0.49±0.035	0.56±0.130
Recall	MNA	**0.59±0.097**	**0.65±0.024**	**0.68±0.078**	**0.71±0.029**	**0.74±0.011**	**0.81±0.02**	**0.87±0.025**	**0.91±0.044**
	MNA-no	0.49±0.076	0.69±0.012	0.67±0.060	0.69±0.077	0.70±0.020	0.745±0.060	0.74±0.041	0.762±0.033
	Social	0.45±0.130	0.48±0.005	0.61±0.038	0.62±0.067	0.72±0.010	0.7±0.06	0.768±0.020	0.72±0.059
	Spatial	0.48±0.050	0.58±0.040	0.51±0.024	0.52±0.03	0.49±0.026	0.55±0.027	0.53±0.020	0.52±0.024
	Text	0.43±0.098	0.65±0.065	0.53±0.056	0.33±0.180	0.33±0.197	0.48±0.137	0.50±0.165	0.57±0.148
	Time	0.43±0.015	0.51±0.04	0.51±0.05	0.52±0.090	0.56±0.03	0.52±0.018	0.53±0.035	0.55±0.130

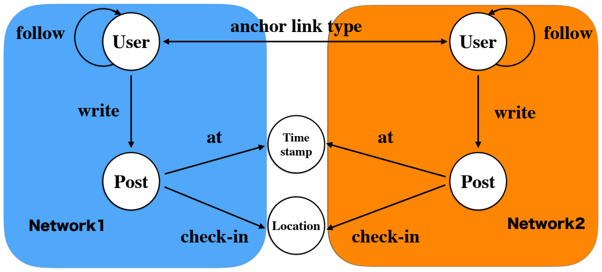


Figure 3: Schema of aligned networks.

shared users between Foursquare and Twitter can make these two networks become *aligned attributed heterogeneous social networks*.

Networks we study in this paper are all *partially aligned* [11, 13] which means user $u_i^{(1)} \in \mathcal{U}^{(1)}$ may not have the corresponding user $u_j^{(2)} \in \mathcal{U}^{(2)}$, where one anchor link connects $u_i^{(1)}$ and $u_j^{(2)}$. We define users like $u_i^{(1)}$ as *non-anchor user*. Besides, we will directly use two *aligned attributed heterogeneous social networks* $\mathcal{G} = ((G^{(1)}, G^{(2)}), \mathcal{A}^{(1,2)})$ to expand the problem definition and the proposed method in the following sections. However, it's natural and simple to apply our proposed method into multiple *aligned attributed heterogeneous social networks* based on the illustrations in this paper.

4.2 Problem Definition

Problem Definition: For the given two *aligned attributed heterogeneous social networks* $\mathcal{G} = ((G^{(1)}, G^{(2)}), \mathcal{A}^{(1,2)})$, the *partitioned heterogeneous network alignment* problem aims to conduct mutual clustering in order to obtain the optimized communities $\{C^{(1)}, C^{(2)}\}$ for $\{G^{(1)}, G^{(2)}\}$. $C^{(1)} = \{U_1^{(1)}, U_2^{(1)}, \ldots, U_k^{(1)}\}$ is a partition of the users set $\mathcal{U}^{(1)}$ in $G^{(1)}$, $k = \left|C^{(1)}\right|$, $U_l^{(1)} \cap U_m^{(1)} = \emptyset, \forall\, l, m \in \{1, 2, \ldots, k\}$ and $\bigcup_{j=1}^{k} U_j^{(1)} = \mathcal{U}^{(1)}$. The clustering is conducted between $G^{(1)}$ and $G^{(2)}$ simultaneously and k is the same in both networks. After mutual clustering, we can get pairs of corresponding users sets

$U_i^{(1)} \in C^{(1)}$ and $U_i^{(2)} \in C^{(2)}$. We can represent all the potential anchor links between networks $G^{(1)}$ and $G^{(2)}$ as set $\mathcal{H} = \bigcup_{j=1}^{k} \mathcal{H}_j$ and $\mathcal{H}_j = \mathcal{U}_j^{(1)} \times \mathcal{U}_j^{(2)}$, where $\mathcal{U}_j^{(1)}$ and $\mathcal{U}_j^{(2)}$ denote the corresponding user sets in $G^{(1)}$ and $G^{(2)}$. We define $\mathcal{Y} = \{0, +1\}$ where $+1$ and 0 denote existing anchor links and non-existing anchor links respectively. In this problem, our goal is to build mapping functions $f_j : \mathcal{H}_j \rightarrow \mathcal{Y}$ to infer anchor link labels in $\mathcal{Y} = \{0, +1\}$ under the *one-to-one* constraint. Finally the set of mapping functions $f = \bigcup_{j=1}^{k} f_j$ will contribute to predict all anchor links for the whole network $\mathcal{G}$ and implement network alignment.

4.3 Network Schema

We have defined *aligned attributed heterogeneous social networks* which can represent most social networks in Section 2. However, in order to better understand the complex *aligned attributed heterogeneous social networks*, it is necessary to define the schema-level description. Therefore, we propose the definition of network schema to describe the meta structure of *aligned attributed heterogeneous social networks*.

Definition 4.3. (Aligned Attributed Heterogeneous Social Network Schema): Given *aligned attributed heterogeneous social networks* $\mathcal{G} = ((G^{(1)}, G^{(2)}), \mathcal{A}^{(1,2)})$, we can represent $\mathcal{G}$ as $S_{\mathcal{G}} = ((S_{G^{(1)}}, S_{G^{(2)}}), \mathcal{L})$, where $\mathcal{L}$ denotes the set of anchor links. $S_{G^{(1)}} = (\mathcal{N}_{\mathcal{V}}^{(1)} \cup \mathcal{N}_{\mathcal{T}}^{(1)}, \mathcal{R}_{\mathcal{E}}^{(1)} \cup \mathcal{R}_{\mathcal{A}}^{(1)})$, where $\mathcal{N}_{\mathcal{V}}^{(1)}$ and $\mathcal{N}_{\mathcal{T}}^{(1)}$ denote the set of node types and attribute types in $G^{(1)}$ seperately. In addition, $\mathcal{R}_{\mathcal{E}}^{(1)}$ and $\mathcal{R}_{\mathcal{A}}^{(1)}$) represent the set of link types and the set of association types between nodes and attributes respectively. The representation is similar for $G^{(2)}$ as $S_{G^{(2)}} = (\mathcal{N}_{\mathcal{V}}^{(2)} \cup \mathcal{N}_{\mathcal{T}}^{(2)}, \mathcal{R}_{\mathcal{E}}^{(2)} \cup \mathcal{R}_{\mathcal{A}}^{(2)})$.

We display the *aligned attributed heterogeneous social networks* in Figure 3, and you can find exact node and link types intuitively as shown in Figure 3.

4.4 Intra-Network Meta Diagram

Definition 4.4. (Intra-Network Meta Path): According to the definition of *aligned attributed heterogeneous social network schema*, given an attributed heterogeneous social network $S_G = (\mathcal{N}, \mathcal{R})$, where $\mathcal{N} = \mathcal{N}_{\mathcal{V}} \cup \mathcal{N}_{\mathcal{T}}$ and $\mathcal{R} = \mathcal{R}_{\mathcal{E}} \cup \mathcal{R}_{\mathcal{A}}$. Path $\mathrm{P}^I = N_1 \xrightarrow{R_1} N_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{n-1}} N_n$ is defined as an *intra-network meta path* with length $n-1$ in the network S_G, where $N_i \in \mathcal{N}, i \in \{1, 2, \cdots, n\}$ and $R_i \in \mathcal{R}, i \in \{1, 2, \cdots, n-1\}$.

Because of the problem we try to solve, we will only care about *intra-network meta paths* which have $N_1, N_n \in \mathrm{U} \wedge N_1 \neq N_n$. In other words, we are concerned about *intra-network meta paths* connecting two different users within one network. Based on the *aligned attributed heterogeneous social network schema*, as shown in Figure 3, the notaion, description and physical meanings of *intra-network meta paths* used in this paper are summarized in the first section of Table 4.

In fact, meta paths can only characterize a small part of rich semantics. For example, we have more than 100 users in the network S_G with check-in records "*u*: (Tallahassee, Apr. 2018)", and only $u_i^{(1)}$, $u_j^{(2)}$ of them have friendship with each other based on meta path P_1, P_5 and P_6, user pair $u_i^{(1)}, u_j^{(2)}$ are much closer than others and are highly likely to be in the same community, since they are friends and check-in the same location at the same time which verify they even attend same activities in daily life. We believe those semantic have the clear gain effect in measure closeness among users. In contrast, if $u_i^{(1)}, u_j^{(2)}$ have no friendship, it perhaps means that they do not have a strong connection, but may just be a coincidence check-in the same place at the same time.In fact, the meta paths can be stacked. In order to characterize richer semantics, we propose the definition of *intra-network meta diagram.*

Definition 4.5. (Intra-Network Meta Diagram): Given an attributed heterogeneous social network $S_G = (\mathcal{N}, \mathcal{R})$, where $\mathcal{N} = \mathcal{N}_{\mathcal{V}} \cup \mathcal{N}_{\mathcal{T}}$ and $\mathcal{R} = \mathcal{R}_{\mathcal{E}} \cup \mathcal{R}_{\mathcal{A}}$. An *inter-network meta diagram* can be formally represented as a directed acyclic subgraph $\Psi^I = (\mathcal{N}_\Psi, \mathcal{R}_\Psi, N_s, N_t)$, where $\mathcal{N}_\Psi \subset \mathcal{N}$ and $\mathcal{R}_\Psi \subset \mathcal{R}$ represents the node, attribute and link types involved, while N_s, N_t denote the source and target node types from network S_G and $N_s \neq N_t$.

Similar to the *intra-network meta paths*, we will only consider *intra-network meta diagrams* which $N_s, N_t \in \{\mathrm{U}\}$. We list several *intra-network meta diagram* examples in the second section of Table 4 which can be represented as $\{\Psi_1^I, \Psi_2^I, \Psi_3^I\}$. Now we focus on the Ψ_1^I at first. It is composed of two meta paths which are both P_1^I and represent two users have the relationship of follower-followee. We insist this followe-followee link is much stronger than one direction link where we can achieve the gain effect. Ψ_2^I is built by P_5^I and P_6^I which represents two users have posts checking in the same location and at the same time. Ψ_3^I containing 3 intra-network meta paths P_1^I, P_5^I and P_6^I respectively. In a more formal way, we can classify intra-network meta paths as P_f containing the social relationship based intra-network meta paths and P representing the sets of the attribute based paths, where $\mathrm{P}_f^I = \{\mathrm{P}_1, \mathrm{P}_2, \mathrm{P}_3, \mathrm{P}_4\}$ and $\mathrm{P}_a^I = \{\mathrm{P}_5, \mathrm{P}_6\}$. Therefore, we can list intra-network meta diagrams as follows:

- $\Psi_{f^2}^I$ $(\mathrm{P}_f^I \times \mathrm{P}_f^I)$: Common Neighbor**s**.
- $\Psi_{a^2}^I$ $(\mathrm{P}_a^I \times \mathrm{P}_a^I)$: Common Attribute**s**.
- $\Psi_{f,a}^I$ $(\mathrm{P}_f^I \times \mathrm{P}_a^I)$: Common Neighbor & Attribute.
- Ψ_{f,a^2}^I $(\mathrm{P}_f^I \times \mathrm{P}_a^I \times \mathrm{P}_a)$: Common Neighbor & Attribute**s**.
- Ψ_{f^2,a^2}^I $(\mathrm{P}_f^I \times \mathrm{P}_f^I \times \mathrm{P}_a^I \times \mathrm{P}_a^I)$: Common Neighbor**s** & Attribute**s**.

Let's take $\Psi_{f,a}^I$ as an example, $\Psi_{f,a}^I = \mathrm{P}_f^I \times \mathrm{P}_a^I = \{\mathrm{P}_i^I \times \mathrm{P}_j^I\}_{\mathrm{P}_i^I \in \mathrm{P}_f^I, \mathrm{P}_j^I \in \mathrm{P}_a^I}$ represents the combination result of meta paths P_i and P_j which are stacked through the common node types shared by them. In this example, node types contain one type from social relationships and the other type from the attributes. The intra-network meta path is the basic element of the intra-network meta diagram and actually is a basic type of the intra-network meta diagram in the shape of path. In the following sections, we will directly use the term *intra-network meta diagrams* to refer to both *intra-network meta paths* and *intra-network meta diagrams.* All the *intra-network meta diagrams* extracted from the social network can be defined as $\Phi_I = \mathrm{P}^I \cup \Psi_{f^2}^I \cup \Psi_{a^2}^I \cup \Psi_{f,a}^I \cup \Psi_{f,a^2}^I \cup \Psi_{f^2,a^2}^I$.

Table 4: Summary of Intra-Network Meta Diagram.

ID	Notation	Meta Diagram	Semantics
P^{I}_{1}	$\mathrm{U} \rightarrow \mathrm{U}$	$\text{User} \xrightarrow{follow} \text{User}$	Follow
P^{I}_{2}	$\mathrm{U} \rightarrow \mathrm{U} \rightarrow \mathrm{U}$	$\text{User} \xrightarrow{follow} \text{User} \xrightarrow{follow} \text{User}$	Follower of Follower
P^{I}_{3}	$\mathrm{U} \rightarrow \mathrm{U} \leftarrow \mathrm{U}$	$\text{User} \xrightarrow{follow} \text{User} \xrightarrow{follow^{-1}} \text{User}$	Common Out Neighbor
P^{I}_{4}	$\mathrm{U} \leftarrow \mathrm{U} \rightarrow \mathrm{U}$	$\text{User} \xrightarrow{follow^{-1}} \text{User} \xrightarrow{follow} \text{User}$	Common In Neighbor
P^{I}_{5}	$\mathrm{U} \rightarrow \mathrm{P} \rightarrow \mathrm{T} \leftarrow \mathrm{P} \leftarrow \mathrm{U}$	$\text{User} \xrightarrow{write} \text{Post} \xrightarrow{at} \text{Timestamp} \xleftarrow{at} \text{Post} \xleftarrow{write} \text{User}$	Posts Containing Common Timestamps
P^{I}_{6}	$\mathrm{U} \rightarrow \mathrm{P} \rightarrow \mathrm{L} \leftarrow \mathrm{P} \leftarrow \mathrm{U}$	$\text{User} \xrightarrow{write} \text{Post} \xrightarrow{checkin} \text{Location} \xleftarrow{checkin} \text{Post} \xleftarrow{write} \text{User}$	Posts Attaching Common Location Check-ins
$\Psi^{I}_{1}(\mathrm{P}^{I}_{1} \times \mathrm{P}^{I}_{1})$	$\mathrm{U} \leftrightarrow \mathrm{U}$	$\begin{tikzcd} \text{User} \arrow[r, bend left, "follow"] & \text{User} \arrow[l, bend left, "follow"] \end{tikzcd}$	Follower and Followee
$\Psi^{I}_{2}(\mathrm{P}^{I}_{5} \times \mathrm{P}^{I}_{6})$	$\begin{tikzcd} & & \mathrm{L} & & \\ \mathrm{U} \arrow[r] & \mathrm{P} \arrow[ur] \arrow[dr] & & \mathrm{P} \arrow[ul] \arrow[dl] & \mathrm{U} \arrow[l] \\ & & \mathrm{T} & & \end{tikzcd}$	$\begin{tikzcd} & & \text{Location} & & \\ \text{User} \arrow[r, "write"] & \text{Post} \arrow[ur, "checkin"] \arrow[dr, "at"'] & & \text{Post} \arrow[ul, "checkin"'] \arrow[dl, "at"] & \text{User} \arrow[l, "write"'] \\ & & \text{Timestamp} & & \end{tikzcd}$	Common Attributes
$\Psi^{I}_{3}(\mathrm{P}^{I}_{1} \times \mathrm{P}^{I}_{5} \times \mathrm{P}^{I}_{6})$	$\begin{tikzcd} & & \mathrm{L} & & \\ \mathrm{U} \arrow[r] \arrow[rrrr, bend left] & \mathrm{P} \arrow[ur] \arrow[dr] & & \mathrm{P} \arrow[ul] \arrow[dl] & \mathrm{U} \arrow[l] \\ & & \mathrm{T} & & \end{tikzcd}$	$\begin{tikzcd} & & \text{Location} & & \\ \text{User} \arrow[r, "write"] \arrow[rrrr, bend left, "follow"] & \text{Post} \arrow[ur, "checkin"] \arrow[dr, "at"'] & & \text{Post} \arrow[ul, "checkin"'] \arrow[dl, "at"] & \text{User} \arrow[l, "write"'] \\ & & \text{Timestamp} & & \end{tikzcd}$	Common Attributes & Follower and Followee

Table 5: Summary of Inter-Network Meta Diagrams.

ID	Notation	Meta Diagram	Semantics
P^{A}_{1}	$\mathrm{U} \rightarrow \mathrm{U} \leftrightarrow \mathrm{U} \leftarrow \mathrm{U}$	$\text{User} \xrightarrow{follow} \text{User} \xleftrightarrow{anchor} \text{User} \xleftarrow{follow} \text{User}$	Common Anchored Followee
P^{A}_{2}	$\mathrm{U} \leftarrow \mathrm{U} \leftrightarrow \mathrm{U} \rightarrow \mathrm{U}$	$\text{User} \xleftarrow{follow} \text{User} \xleftrightarrow{anchor} \text{User} \xrightarrow{follow} \text{User}$	Common Anchored Follower
P^{A}_{3}	$\mathrm{U} \rightarrow \mathrm{U} \leftrightarrow \mathrm{U} \rightarrow \mathrm{U}$	$\text{User} \xrightarrow{follow} \text{User} \xleftrightarrow{anchor} \text{User} \xrightarrow{follow} \text{User}$	Common Anchored Followee-Follower
P^{A}_{4}	$\mathrm{U} \leftarrow \mathrm{U} \leftrightarrow \mathrm{U} \leftarrow \mathrm{U}$	$\text{User} \xleftarrow{follow} \text{User} \xleftrightarrow{anchor} \text{User} \xleftarrow{follow} \text{User}$	Common Anchored Follower-Followee
P^{A}_{5}	$\mathrm{U} \rightarrow \mathrm{P} \rightarrow \mathrm{T} \leftarrow \mathrm{P} \leftarrow \mathrm{U}$	$\text{User} \xrightarrow{write} \text{Post} \xrightarrow{at} \text{Timestamp} \xleftarrow{at} \text{Post} \xleftarrow{write} \text{User}$	Common Timestamp
P^{A}_{6}	$\mathrm{U} \rightarrow \mathrm{P} \rightarrow \mathrm{L} \leftarrow \mathrm{P} \leftarrow \mathrm{U}$	$\text{User} \xrightarrow{write} \text{Post} \xrightarrow{checkin} \text{Location} \xleftarrow{checkin} \text{Post} \xleftarrow{write} \text{User}$	Common Checkin
$\Psi^{A}_{1}(\mathrm{P}^{A}_{1} \times \mathrm{P}^{A}_{2})$	$\mathrm{U} \leftrightarrow \mathrm{U} \xleftrightarrow{anchor} \mathrm{U} \leftrightarrow \mathrm{U}$	$\begin{tikzcd} \text{User} \arrow[r, bend left, "follow"] & \text{User} \arrow[l, bend left, "follow"] \arrow[r, leftrightarrow, "anchor"] & \text{User} \arrow[r, bend right, "follow"'] & \text{User} \arrow[l, bend right, "follow"'] \end{tikzcd}$	Common Aligned Neighbors
$\Psi^{A}_{2}(\mathrm{P}^{A}_{5} \times \mathrm{P}^{A}_{6})$	$\begin{tikzcd} & & \mathrm{L} & & \\ \mathrm{U} \arrow[r] & \mathrm{P} \arrow[ur] \arrow[dr] & & \mathrm{P} \arrow[ul] \arrow[dl] & \mathrm{U} \arrow[l] \\ & & \mathrm{T} & & \end{tikzcd}$	$\begin{tikzcd} & & \text{Location} & & \\ \text{User} \arrow[r, "write"] & \text{Post} \arrow[ur, "checkin"] \arrow[dr, "at"'] & & \text{Post} \arrow[ul, "checkin"'] \arrow[dl, "at"] & \text{User} \arrow[l, "write"'] \\ & & \text{Timestamp} & & \end{tikzcd}$	Common Attributes
$\Psi^{A}_{3}(\mathrm{P}^{A}_{1} \times \mathrm{P}^{A}_{5} \times \mathrm{P}^{A}_{6})$	$\begin{tikzcd} \mathrm{U} \arrow[rrrr, leftrightarrow] & & & & \mathrm{U} \\ & & \mathrm{L} & & \\ \mathrm{U} \arrow[uu] \arrow[r] & \mathrm{P} \arrow[ur] \arrow[dr] & & \mathrm{P} \arrow[ul] \arrow[dl] & \mathrm{U} \arrow[l] \arrow[uu] \\ & & \mathrm{T} & & \end{tikzcd}$	$\begin{tikzcd} & \text{User} \arrow[rr, leftrightarrow, "anchor"] & & \text{User} & \\ & & \text{Location} & & \\ \text{User} \arrow[uur, "follow"] \arrow[r, "write"] & \text{Post} \arrow[ur, "checkin"] \arrow[dr, "at"'] & & \text{Post} \arrow[ul, "checkin"'] \arrow[dl, "at"] & \text{User} \arrow[l, "write"'] \arrow[uul, "follow"'] \\ & & \text{Timestamp} & & \end{tikzcd}$	Common Aligned Neighbor & Attributes

4.5 Inter-Network Meta Diagram

In fact, as paths and diagrams, *inter-network meta paths* and *inter-network meta diagrams* can be defined in a similar way as we define *intra-network meta paths* and *intra-network meta diagrams* in last section. The main difference between them can be found from the terms: *intra-network meta diagrams* exist within one social network, and *inter-network meta diagrams* connect two nodes across two social networks. We will define *inter-network meta diagrams* in the similar way and reflect the difference in the definition as well. We still introduce *intra-network meta paths* first which are the basis of *intra-network meta diagrams.*

Definition 4.6. (Inter-Network Meta Path): Given *aligned attributed heterogeneous social networks* $S_{\mathcal{G}} = ((S_{G^{(1)}}, S_{G^{(2)}}), \mathcal{L})$, we represent $S_{G^{(1)}}$ as $S_{G^{(1)}} = (\mathcal{N}^{(1)}, \mathcal{R}^{(1)})$, where $\mathcal{N}^{(1)} = \mathcal{N}^{(1)}_{\mathcal{V}} \cup \mathcal{N}^{(1)}_{\mathcal{T}}$ and $\mathcal{R}^{(1)} = \mathcal{R}^{(1)}_{\mathcal{E}} \cup \mathcal{R}^{(1)}_{\mathcal{A}}$. For $S_{G^{(2)}}$, the representation is similar. Path $\mathrm{P}^{A} = N_1 \xrightarrow{R_1} N_2 \xrightarrow{R_2} \cdots \xrightarrow{R_{n-1}} N_n$ is defined as an *inter-network meta path* with length $n-1$ between the network $G^{(1)}$ and $G^{(2)}$, where $N_i \in \mathcal{N}^{(1)} \cup \mathcal{N}^{(2)}, i \in \{1, 2, \cdots, n\}$ and $R_i \in \mathcal{R} \cup \mathcal{L}, i \in \{1, 2, \cdots, n-1\}$.

Only paths connecting users across networks are useful for us, in which $N_1, N_n \in \{\mathrm{U}^{(1)}, \mathrm{U}^{(2)}\} \wedge N_1 \neq N_n$. According to the network schema, we extract some *inter-network meta paths* which descriptions are listed in the Table 5. Compared with *intra-network meta paths*, *inter-network meta paths* are used to discribe the similarity between two users across networks instead of the closeness of users in the same network.

However, *inter-network meta paths* also have the disadvantage that can not cover rich semantics as *intra-network meta paths.* We still present an example here. Given two users $u^{(1)}_{i}$ and $u^{(2)}_{j}$ with check-in records "$u^{(1)}_{i}$: (Boston, Apr. 2018), (New York, Jan. 2018), (Los Angeles, May 2018)", and "$u^{(2)}_{j}$: (Los Angeles, Apr. 2018), (Boston, Jan. 2018), (New York, May 2018)" respectively, if the feature just base on *inter-network meta path* P^{A}_{5} and P^{A}_{6}, user pair $u^{(1)}_{i}$, $u^{(2)}_{j}$ have checked-in the same locations for 3 times and 3 posts at the same time. However, according to the records, we observe that they have never been at the same place for the same moments,and this scene reflects that $u^{(1)}_{i}$ and $u^{(2)}_{j}$ have very low similarity. We choose to use *inter-network meta diagrams* which are stacked by *inter-network meta paths* to cover semantics fully.

Definition 4.7. (Inter-Network Meta Diagram): Given *aligned attributed heterogeneous social networks* $S_{\mathcal{G}} = ((S_{G^{(1)}}, S_{G^{(2)}}), \mathcal{L})$, where $S_{G^{(1)}} = (\mathcal{N}^{(1)}, \mathcal{R}^{(1)})$, and $\mathcal{N}^{(1)} = \mathcal{N}^{(1)}_{\mathcal{V}} \cup \mathcal{N}^{(1)}_{\mathcal{T}}$ and $\mathcal{R}^{(1)} = \mathcal{R}^{(1)}_{\mathcal{E}} \cup \mathcal{R}^{(1)}_{\mathcal{A}}$. 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 $\mathcal{N}_{\Psi} \subset \mathcal{N}^{(1)} \cup \mathcal{N}^{(2)}$ and $\mathcal{R}_{\Psi} \subset \mathcal{R} \cup \mathcal{L}$ represent the node, attribute and link types involved, while N_s, N_t denote the source and target node types from $G^{(1)}$ and $G^{(2)}$ and $N_s \neq N_t$.

Several *inter-network meta diagram* examples extracted from Twitter and Foursquare have been shown in the Table 5. We classify inter-network meta paths and represent the stacking process in the same way as inner-network meta paths. All inte-network meta diagrams can be represented and explained according to the discussion in Section 4.4. The complete list of inter-network meta diagrams extracted in this paper are listed as follows:

- $\Psi^{A}_{f^2}$ ($\mathrm{P}^{A}_{f} \times \mathrm{P}^{A}_{f}$): Common Aligned Neighbors.
- $\Psi^{A}_{a^2}$ ($\mathrm{P}^{A}_{a} \times \mathrm{P}^{A}_{a}$): Common Attributes.
- $\Psi^{A}_{f,a}$ ($\mathrm{P}^{A}_{f} \times \mathrm{P}^{A}_{a}$): Common Aligned Neighbor & Attribute.
- Ψ^{A}_{f,a^2} ($\mathrm{P}^{A}_{f} \times \mathrm{P}^{A}_{a} \times \mathrm{P}^{A}_{a}$): Common Aligned Neighbor & Attributes.
- $\Psi^{A}_{f^2,a^2}$ ($\mathrm{P}^{A}_{f} \times \mathrm{P}^{A}_{f} \times \mathrm{P}^{A}_{a} \times \mathrm{P}^{A}_{a}$): Common Aligned Neighbors & Attributes.

We will directly use the term *inter-network meta diagrams* to represent both *inter-network meta paths* and *inter-network meta diagrams*, where $\Phi^{A} = \mathrm{P}^{A} \cup \Psi^{A}_{f^2} \cup \Psi^{A}_{a^2} \cup \Psi^{A}_{f,a} \cup \Psi^{A}_{f,a^2} \cup \Psi^{A}_{f^2,a^2}$.

4.6 Mutual Clustering across Mutiple Networks

The first step of CLUITER is *Mutual Clustering*, and we both exploit information within the network and across networks. To solve the mutual clustering problem and obtain the optimal communities, we make use of *intra-network meta diagrams* to measure the closeness among users in one network and represent the information in other networks with the help of *inter-network meta diagrams* to adjust community strucures mutually.

4.6.1 Intra-Network Meta Diagram based Clustering.

As we describe in Section 4.4, we extract *intra-network meta diagrams* from the social network, and some of them are listed in Table 4. These Intra-network meta diagrams Φ^{I} can represent lots of connections and reflect rich sematics among users in the same social network. Therefore, we can use MDs to measure the closeness among users in a directed heterogeneous social networks, where we name this kind of measurement as IntraMD-Sim(IntraMD-based Similarity).

Definition 4.8. (IntraMD-Sim): We define $\mathcal{D}_i(x \rightsquigarrow y)$ to represent the set of diagrams of Φ^{I}_{i} starting from x to y, and $\mathcal{D}_i(x \rightsquigarrow \cdot)$ to represent the diagrams of Φ^{I}_{i} which go from x to other nodes in the network. The IntraMD-Sim of node pair (x, y) can be defined as

$$\text{IntraMD-Sim}(x, y) = \sum_i \omega_i \left(\frac{|\mathcal{D}_i(x \rightsquigarrow y)| + |\mathcal{D}_i(y \rightsquigarrow x)|}{|\mathcal{D}_i(x \rightsquigarrow \cdot)| + |\mathcal{D}_i(y \rightsquigarrow \cdot)|} \right),$$

where ω_i is the weight of Φ^{I}_{i} and $\sum_i \omega_i = 1$.

What's more, we use $\mathbf{A}_i$ as the *adjacency matrix* which represents Φ^{I}_{i} among users in the network. If there exist k different diagram instances of Φ^{I}_{i} from user x to y, we will record it as $\mathbf{A}_i(x, y) = k$. In this way, the similarity score matrix among users of Φ^{I}_{i} can be represented as $\mathbf{S}_i = \mathbf{B}_i \circ \left(\mathbf{A}_i + \mathbf{A}_i^{T}\right)$, where $\mathbf{A}_i^{T}$ denotes the transpose of $\mathbf{A}_i$, the matrix $\mathbf{B}_i$ which represent the sum of the out-degree of user x and y has values $\mathbf{B}_i(x, y) = (\sum_m \mathbf{A}_i(x, m) + \sum_m \mathbf{A}_i(y, m))^{-1}$. The The $\circ$ symbol represents the Hadamard product of two matrices. IntraMD-Sim matrix of the network which can capture all possible connections among users is represented as follows:

$$\mathbf{S} = \sum_i \omega_i \mathbf{S}_i = \sum_i \omega_i \left(\mathbf{B}_i^{-1} \circ \left(\mathbf{A}_i + \mathbf{A}_i^{T} \right) \right).$$

With the basis of IntraMD-based Similarity, we are able to preserve the network characteristics and cluster the single network independently. For a given network G, let $C = \{U_1, U_2, \ldots, U_k\}$ be the community structures detected from G. We define $\overline{U_i} = \mathcal{U} - U_i$ as complement of set U_i in G. Traditional clustering methods, e.g., *cut* and *normalized-cut* in one single network aim at minimizing the costs to obtain an optimal clustering result. In the clustering method *cut*, the cost is defined as:

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

where $S(u, v)$ denotes the IntraMD-Sim between u, v. Then the *normalized-cut* can be represented as:

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

where $S(U_i, \cdot) = S(U_i, \mathcal{U}) = S(U_i, U_i) + S(U_i, \overline{U}_i)$.

We define $\mathbf{h}_i = (h_{i,1}, h_{i,2}, \ldots, h_{i,k})$, where $h_{i,j}$ denotes the confidence that $u_i \in \mathcal{U}$ is in cluster $U_j \in C$, and k is the number of detected communities as we mentioned before. Therefore, we can define the clustering result of all users in $\mathcal{U}$ as the *result confidence matrix* $\mathbf{H}$, where $\mathbf{H} = [\mathbf{h_1}, \mathbf{h_2}, \ldots, \mathbf{h_n}]^{\mathbf{T}}$ and $n = |\mathcal{U}|$. We can solve the following objective function to minimize the *normalized-cut* cost and achieve the optimal clustering result, and the derivation of this objective equation has been verified by [?]:

$$\min_{\mathbf{H}} \ \mathrm{Tr}(\mathbf{H}^{T}\mathbf{L}\mathbf{H}),$$
$$s.t. \ \mathbf{H}^{T}\mathbf{D}\mathbf{H} = \mathbf{I}.$$

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.

4.6.2 Inter-Network Meta Diagram based Clustering of Mutiple Networks.

With the help of *inter-network meta diagrams*, we can represent the extra information and knowledge about the social network structures across aligned attributed heterogeneous networks which are useful for detecting communities from a more complete and convincing view.

At first, we shall show the way of promiximity feature extraction with *inter-network meta diagrams.* Given a pair of users $u^{(1)}_{i}$ and $u^{(2)}_{j}$, based on inter-network meta diagrams $\Phi^{I}_{k} \in \Phi^{I}$, we can represent the set of inter-network diagram instances connecting $u^{(1)}_{i}$ and $u^{(2)}_{j}$ as $\mathcal{D}_{\Phi^{I}_{k}}(u^{(1)}_{i}, u^{(2)}_{j})$. Users $u^{(1)}_{i}$ and $u^{(2)}_{j}$ can have multiple inter-network meta diagram instances going into/out from them. Formally, we can represent all the inter-network meta diagram instances going out from user $u^{(1)}_{i}$ (or going into $u^{(2)}_{j}$), based on inter-network meta diagram Φ^{I}_{k}, as set $\mathcal{D}_{\Phi^{I}_{k}}(u^{(1)}_{i}, \cdot)$ (or $\mathcal{D}_{\Phi^{I}_{k}}(\cdot, u^{(2)}_{j})$).

The proximity score between $u_i^{(1)}$ and $u_j^{(2)}$ based on inter-network meta diagram Φ_k^I can be defined as *InterMD-Pro*.

Definition 4.9. (InterMD-Pro): Based on inter-network meta diagrams Φ^I, the InterMD-proximity between users $u_i^{(1)}$ and $u_j^{(2)}$ in G can be represented as

$$\text{InterMD-Pro}(u_i^{(1)}, u_j^{(2)}) = \sum_k \omega_k \left(\frac{2|\mathcal{D}_{\Phi_k^I}(u_i^{(1)}, u_j^{(2)})|}{|\mathcal{D}_{\Phi_k^I}(u_i^{(1)}, \cdot)| + |\mathcal{D}_{\Phi_k^I}(\cdot, u_j^{(2)})|} \right).$$

InterMD-Pro considers not only the inter-network meta diagram instances between $u_i^{(1)}$ and $u_j^{(2)}$ but also penalizes those going out from and into $u_i^{(1)}$ and $u_j^{(2)}$ at the same time. With the above definition of InterMD-Pro, we can represent the proximity scores among all users in the network $\mathcal{G}$ based on inter-network meta diagrams Φ^I as matrix $\mathbf{S}_{\Phi^I} \in \mathbb{R}^{|\mathcal{U}^{(1)}| \times |\mathcal{U}^{(2)}|}$, where $S_{\Phi^I}(i,j) = \text{InterMD-Pro}(u_i^{(1)}, u_j^{(2)})$. By minimizing the *discrepancy* of the clustering results based on *InterMD-Pro* in multiple partially aligned networks, we can refine the clustering results with information in other aligned networks mutually.

Given *aligned attributed heterogeneous social networks* $\mathcal{G} = ((G^{(1)}, G^{(2)}), \mathcal{A}^{(1,2)})$, we can represent the clustering results in $G^{(1)}$ and $G^{(2)}$ as $C^{(1)} = \{U_1^{(1)}, U_2^{(1)}, \cdots, U_{k^{(1)}}^{(1)}\}$ and $C^{(2)} = \{U_1^{(2)}, U_2^{(2)}, \cdots, U_{k^{(2)}}^{(2)}\}$ respectively. Let users $u_l^{(1)}$ and $u_m^{(1)}$ come from $G^{(1)}$, and $S_{\Phi^I}(i,j) = \text{InterMD-Pro}(u_i^{(1)}, u_j^{(2)})$. As we define in the Section 4.6.1, $\mathbf{h}_l^{(1)} = (h_{l,1}^{(1)}, h_{l,2}^{(1)}, \ldots, h_{l,k}^{(1)})$, where $h_{l,j}^{(1)}$ denotes the confidence that $u_l^{(1)} \in \mathcal{U}^{(1)}$ is in cluster $U_j^{(1)} \in C^{(1)}$. Thus the confidences that $u_l^{(1)}$ and $u_m^{(1)}$ are in the same cluster can be denoted as $\mathbf{h}_l^{(1)}(\mathbf{h}_m^{(1)})^T$. According the meaning of InterMD-Pro, we can represent the clustering confidence scores in $G^{(1)}$ with the clustering confidence scores in $G^{(2)}$, where the similarity matrix $\mathbf{S}_{\Phi^I}$ acts as a bridge. Formally, we define *Transition Clustering Confidence Scores*

$$\bar{\mathbf{h}}_l^{(1)} = \sum_{k=1}^{|\mathcal{U}^{(2)}|} S_{\Phi^I}(l,k) \cdot \mathbf{h}_k^{(2)}$$

Now we have two confidence scores about the clusting result of user $u_l^{(1)}$: $\mathbf{h}_l^{(1)}$ and $\bar{\mathbf{h}}_l^{(1)}$. If users $u_l^{(1)}$ and $u_m^{(1)}$ are partitioned into the same cluster in $G^{(1)}$ but they are partitioned into different clusters based on the *transition clustering confidence scores* from $G^{(2)}$, then it will lead to a *discrepancy* between the clustering results of $u_l^{(1)}, u_m^{(1)}$. From the opposite direction, the situation is the same that users in $G^{(2)}$ can also achieve *transition clustering confidence scores* from $G^{(1)}$.

Definition 4.10. (Discrepancy): The discrepancy between the clustering results of $u_l^{(1)}$ and $u_m^{(1)}$ across aligned networks $G^{(1)}$ and $G^{(2)}$ is defined as the difference of confidence scores of $u_l^{(1)}$ and $u_m^{(1)}$ being partitioned in the same cluster based on two kind of confidence scores: $\mathbf{h}_l^{(1)}$ and $\bar{\mathbf{h}}_l^{(1)}$, $\mathbf{h}_m^{(1)}$ and $\bar{\mathbf{h}}_m^{(1)}$. Formally, the discrepancy of the clustering results about $u_l^{(1)}$ and $u_m^{(1)}$ is defined to be $d_{lm}(C^{(1)}) = \left(\mathbf{h}_l^{(1)}(\mathbf{h}_m^{(1)})^T - \bar{\mathbf{h}}_l^{(1)}(\bar{\mathbf{h}}_m^{(1)})^T\right)^2$. Furthermore, the discrepancy of $C^{(1)}$ and $C^{(2)}$ will be:

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

$$d(C^{(2)}) = \sum_{i}^{|\mathcal{U}^{(2)}|} \sum_{j=i+1}^{|\mathcal{U}^{(2)}|} d_{ij}(C^{(2)})$$

Based on the discrepancy of $d(C^{(1)})$ and $d(C^{(2)})$, we can represent the discrepancy of the aligned attributed heterogeneous networks $\mathcal{G}$ as $d(C^{(1)}, C^{(2)}) = d(C^{(1)}) + d(C^{(2)})$.

Considering that $G^{(1)}$ and $G^{(2)}$ may have different user number, and minimizing $d(C^{(1)}, C^{(2)})$ will be affected due to the imbalance of the size of mutiple networks. However, we insist the discrepancy from two directions should exert equal influence to clustering result. To solve the problem of imbalance, we propose to minimize the *normalized discrepancy* instead.

Definition 4.11. (Normalized Discrepancy): The normalized discrepancy consider the differences of clustering results in two aligned networks as a fraction of the discrepancy with regard to the number of users in each networks:

$$Nd(C^{(1)}, C^{(2)}) = \frac{d(C^{(1)})}{(|\mathcal{U}^{(1)}|)\,(|\mathcal{U}^{(1)}| - 1)} + \frac{d(C^{(2)})}{(|\mathcal{U}^{(2)}|)\,(|\mathcal{U}^{(2)}| - 1)}.$$

Optimal consensus clustering results of $G^{(1)}$ and $G^{(2)}$ will be $\hat{C^{(1)}}, \hat{C^{(2)}}$:

$$\hat{C}^{(1)}, \hat{C}^{(2)} = \arg\min_{C^{(1)}, C^{(2)}} Nd(C^{(1)}, C^{(2)}).$$

We can represent the normalized-discrepancy objective function with the *clustering results confidence matrices* $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ as well. According to the previous definition of *transition clustering confidence scores*, we can define $\bar{\mathbf{H}}^{(1)} = [\bar{\mathbf{h}}_1^{(1)}, \bar{\mathbf{h}}_2^{(1)}, \ldots, \bar{\mathbf{h}}_n^{(1)}]^T$ and $n = |\mathcal{U}^{(1)}|$, and $\bar{\mathbf{H}}^{(2)}$ is the same situation. In fact, $\bar{\mathbf{H}}^{(2)} = \left(\mathbf{S}_{\Phi^I}\right)^T \mathbf{H}^{(1)}$ and $\bar{\mathbf{H}}^{(1)} = \mathbf{S}_{\Phi^I}\mathbf{H}^{(2)}$. Further, the objective function of inferring clustering confidence matrices, which can minimize the normalized discrepancy can be represented as follows

$$\min_{\mathbf{H}^{(1)}, \mathbf{H}^{(2)}} \left(\frac{\left\| \bar{\mathbf{H}}^{(1)}\left(\bar{\mathbf{H}}^{(1)}\right)^T - \mathbf{H}^{(1)}\left(\mathbf{H}^{(1)}\right)^T \right\|_F^2}{(|\mathcal{U}^{(1)}|)(|\mathcal{U}^{(1)}|-1)} + \frac{\left\| \bar{\mathbf{H}}^{(2)}\left(\bar{\mathbf{H}}^{(2)}\right)^T - \mathbf{H}^{(2)}\left(\mathbf{H}^{(2)}\right)^T \right\|_F^2}{(|\mathcal{U}^{(2)}|)(|\mathcal{U}^{(2)}|-1)} \right),$$

$$s.t.\ \ (\mathbf{H}^{(1)})^T \mathbf{D}^{(1)} \mathbf{H}^{(1)} = \mathbf{I}, (\mathbf{H}^{(2)})^T \mathbf{D}^{(2)} \mathbf{H}^{(2)} = \mathbf{I}.$$

where $\mathbf{D}^{(1)}$, $\mathbf{D}^{(2)}$ are the corresponding diagonal matrices of IntraMD-Sim matrices of networks $G^{(1)}$ and $G^{(2)}$ respectively.

4.6.3 *Joint Clustering of Multiple Networks.*

Taking both *Intra-Network Meta Diagram based Clustering* and *Inter-Network Meta Diagram based Clustering* into considerations, the optimal mutual clustering results $\hat{C}^{(1)}$ and $\hat{C}^{(2)}$ of aligned networks $G^{(1)}$ and $G^{(2)}$ can be achieved as follows:

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

where α, β and θ represents the weights of these compositions. We can replace $Ncut(C^{(1)})$, $Ncut(C^{(2)})$, $Nd(C^{(1)}, C^{(2)})$ with the objective functions derived before, we can rewrite the joint objective function as follows:

$$\min_{\mathbf{H}^{(1)},\mathbf{H}^{(2)}} \quad \alpha \cdot \mathrm{Tr}((\mathbf{H}^{(1)})^T \mathbf{L}^{(1)} \mathbf{H}^{(1)}) + \beta \cdot \mathrm{Tr}((\mathbf{H}^{(2)})^T \mathbf{L}^{(2)} \mathbf{H}^{(2)})$$

$$+ \theta \cdot \left(\frac{\left\| \bar{\mathbf{H}}^{(1)} \left(\bar{\mathbf{H}}^{(1)}\right)^T - \mathbf{H}^{(1)} \left(\mathbf{H}^{(1)}\right)^T \right\|_F^2}{(|\mathcal{U}^{(1)}|)(|\mathcal{U}^{(1)}|-1)} + \frac{\left\| \bar{\mathbf{H}}^{(2)} \left(\bar{\mathbf{H}}^{(2)}\right)^T - \mathbf{H}^{(2)} \left(\mathbf{H}^{(2)}\right)^T \right\|_F^2}{(|\mathcal{U}^{(2)}|)(|\mathcal{U}^{(2)}|-1)} \right),$$

$$s.t. \;\; (\mathbf{H}^{(1)})^T \mathbf{D}^{(1)} \mathbf{H}^{(1)} = \mathbf{I}, (\mathbf{H}^{(2)})^T \mathbf{D}^{(2)} \mathbf{H}^{(2)} = \mathbf{I},$$

where $\mathbf{L}^{(1)} = \mathbf{D}^{(1)} - \mathbf{S}^{(1)}, \mathbf{L}^{(2)} = \mathbf{D}^{(2)} - \mathbf{S}^{(2)}$ and matrices $\mathbf{S}^{(1)}, \mathbf{S}^{(2)}$ and $\mathbf{D}^{(1)}, \mathbf{D}^{(2)}$ are the IntraMD-Sim matrices and their corresponding diagonal matrices.

4.7 Heterogeneous Network Alignment

In this part, we will introduce the *network alignment* model for the anchor link prediction across partitioned networks after *Mutual Clustering* in the first step, which involves 3 main components: (1) *discriminative function* for labeled instances, (2) *generative function* for unlabeled instance, and (3) *one-to-one constraint* modeling component.

4.7.1 Labeled Data Discriminative Loss Function. For all the potential anchor links in set $\mathcal{H}$ (involving both the labeled and unlabeled anchor link instances), a set of features will be extracted based on the meta diagrams introduced before. Formally, the feature vector is based on *inter-network meta diagrams* and its relating proximity as we define in previous content. The feature vector extracted for anchor link $l \in \mathcal{H}$ can be represented as vector $\mathbf{x}_l \in \mathbb{R}^d$ (parameter d denotes the feature size). Meanwhile, we can denote the label of link $l \in \mathcal{L}$ as $y_l \in \mathcal{Y}$ ($\mathcal{Y} = \{0, +1\}$), which denotes the existence of anchor link l between the networks. For the existing anchor links in set $\mathcal{L}_+$, they will be assigned with +1 label; while the labels of anchor links in $\mathcal{U}$ are unknown. All the labeled anchor links in set $\mathcal{L}_+$ can be represented as a tuple set $\{(\mathbf{x}_l, y_l)\}_{l \in \mathcal{L}_+}$. Depending on whether the anchor link instances are linearly separable or not, the extracted anchor link feature vectors can be projected to different feature spaces with various kernel functions $g : \mathbb{R}^d \to \mathbb{R}^k$. For instance, given the feature vector $\mathbf{x}_l \in \mathbb{R}^d$ of anchor link l, we can represent its projected feature vector as $g(\mathbf{x}_l) \in \mathbb{R}^k$. In this paper, the linear kernel function will be used for simplicity, and we have $g(\mathbf{x}_l) = \mathbf{x}_l$ for all the links l.

In the *network alignment* model, the *discriminative* component can effectively differentiate the positive instances from the non-existing ones, which can be denoted as mapping $f(\cdot; \theta_f) : \mathbb{R}^d \to \{+1, 0\}$ 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 $f(\mathbf{x}_l; \mathbf{w}) = \mathbf{w}^\top \mathbf{x}_l + b$, 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. Based on the above descriptions, we can represent the introduced *discriminative* loss function on the labeled set $\mathcal{L}_+$ as

$$L(f, \mathcal{L}_+; \mathbf{w}) = \sum_{l \in \mathcal{L}_+} \left(f(\mathbf{x}_l; \mathbf{w}) - y_l\right)^2 = \sum_{l \in \mathcal{L}_+} (\mathbf{w}^\top \mathbf{x}_l - y_l)^2.$$

4.7.2 Unlabeled Data Generative Loss Function. Meanwhile, to alleviate the insufficiency of labeled data, we also propose to utilize the unlabeled anchor links to encourage the learned model can capture the capture the salient structures of all the anchor link instances. Based on the above discriminative model function $f(\cdot; \mathbf{w})$, for a unlabeled anchor link $l \in \mathcal{U}$, 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 $sign\big(f(\mathbf{x}_l; \mathbf{w})\big) \in \{+1, 0\}$. How to determine its value will be introduced later in Section 4.8. Based on it, the loss function introduced in the generative component based on the unlabeled anchor links can be denoted as

$$L(f, \mathcal{U}; \mathbf{w}) = \sum_{l \in \mathcal{U}} \left(\mathbf{w}^\top \mathbf{x}_l - sign\big(f(\mathbf{x}_l; \mathbf{w})\big)\right)^2.$$

4.7.3 Cardinality Mathematical Constraint. As introduced before, the anchor links to be inferred between networks are subject to the *one-to-one* cardinality constraint. Such a constraint will control the maximum number of anchor links incident to the user nodes in each networks. Subject to the cardinality constraints, the prediction task of anchor links between networks are no longer independent. For instance, if anchor link $(u^{(1)}, v^{(2)})$ is predicted to be positive, then all the remaining anchor links incident to $u^{(1)}$ and $v^{(2)}$ in the unlabeled set $\mathcal{U}$ 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 in this paper. To represent the user node-anchor link relationships in networks $G^{(1)}$ and $G^{(2)}$ respectively, we introduce the user node-anchor link incidence matrices $\mathbf{A}^{(1)} \in \{0, 1\}^{|\mathcal{U}^{(1)}| \times |\mathcal{H}|}, \mathbf{A}(2) \in \{0, 1\}^{|\mathcal{U}^{(2)}| \times |\mathcal{H}|}$ in this paper. Entry $A^{(1)}(i, j) = 1$ iff anchor link $l_j \in \mathcal{H}$ is connected with user node $u_i^{(1)}$ in $G^{(1)}$, and it is similar for $A^{(2)}$.

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 $l_i \in \mathcal{L}$. Depending on which group l_i belongs to, its value has different representations as introduced before $y(i) = +1$, if $l_i \in \mathcal{L}_+$; $y(i) \tilde{y}_{l_i}$, if $l_i \in \mathcal{U}_q$, and $y(i)$ is unknown if $l_i \in \mathcal{U}$. Furthermore, based on the anchor link label vector $\mathbf{y}$, user node-anchor link incidence matrices $\mathbf{A}^{(1)}$ and $\mathbf{A}^{(2)}$, we can represent the the user node degrees in networks $G^{(1)}$ and $G^{(2)}$ as vectors $\mathbf{d}^{(1)} \in \mathbb{N}^{|\mathcal{H}|}$ and $\mathbf{d}^{(2)} \in \mathbb{N}^{|\mathcal{H}|}$ respectively.

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

Therefore, the *one-to-one* constraint on anchor links can be denoted as the constraints on node degrees in $G^{(1)}$ and $G^{(2)}$ as follows:

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

4.8 Joint Optimization Objective Function

Based on the introduction in the previous subsection, by combining the loss terms introduced by the labeled, queried and remaining unlabeled anchor links together with the cardinality constraint, we can represent the joint optimization objective function as

$$\min_{\mathbf{w}, \mathbf{y}} L(f, \mathcal{L}_+; \mathbf{w}) + \alpha \cdot L(f, \mathcal{U}; \mathbf{w}) + \beta \cdot \|\mathbf{w}\|_2^2$$

$$s.t. \;\; y_l \in \{+1, 0\}, \forall l \in \mathcal{U}, y_l = +1, \forall l \in \mathcal{L}_+$$

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

Here, we can see the objective function involve multiple variables, i.e., variable $\mathbf{w}$, label $\mathbf{y}$, and the objective is not jointly convex with regarding these variables. What's more, the inference of the label variable $\mathbf{y}$ is the combinatorial problem, and obtaining their optimal solution will be NP-hard. In this paper, we design an hierarchical alternative variable updating process for solving the problem instead:

(1) fix $\mathbf{y}$, and update $\mathbf{w}$,
(2) fix $\mathbf{w}$, and update $\mathbf{y}$.

Next, we will illustrate the detailed alternative learning algorithm as follows.

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

With $\mathbf{y}$, fixed, we can represent the objective function involving variable $\mathbf{w}$ as

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

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

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

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

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

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

$$\min_{\mathbf{y}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|_2^2$$
$$s.t. y_l = +1, \forall l \in \mathcal{L}_+, \text{ and } \mathbf{0} \leq \mathbf{A}^{(1)}\mathbf{y} \leq \mathbf{1}, \text{ and } \mathbf{0} \leq \mathbf{A}^{(2)}\mathbf{y} \leq \mathbf{1}.$$

It is an integer programming problem, which has been shown to be NP-hard and no efficiently algorithm exists that lead to the optimal solution. In this paper, we will use the greedy link selection algorithm proposed in [10] based on values $\hat{\mathbf{y}} = \mathbf{X}\mathbf{w}$, which has been proven to achieve $\frac{1}{2}$-approximation of the optimal solution.

REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WWW*, page 635âĂŞ644, 2011.

[2] J. Flannick, A. Novak, B. Srinivasan, H. McAdams, and S. Batzoglou. Graemlin: general and robust alignment of multiple large interaction networks. *Genome research*, 2006.

[3] Al Hasan, Mohammad, and Mohammed J. Zaki. *A survey of link prediction in social networks.* Social network data analytics., 2011.

[4] X. Kong, J. Zhang, and P. Yu. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*, 2013.

[5] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM*, page 556âĂŞ559, 2003.

[6] R. Lichtenwlter, J. Lussier, and N. Chawla. New perspectives and methods in link prediction. In *KDD*, page 243âĂŞ252, 2010.

[7] M. Newman. *Clustering and preferential attachments in growing networks.* Physical Review Letters, 2001.

[8] R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *RECOMB*, 2007.

[9] R. Singh, J. Xu, and B. Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 2008.

[10] J. Zhang, J. Chen, J. Zhu, Y. Chang, and P. Yu. Link prediction with cardinality constraint. In *WSDM*, 2017.

[11] J. Zhang, X. Kong, and P. Yu. Transfer heterogeneous links across location-based social networks. In *WSDM*, 2014.

[12] J. Zhang and P. Yu. Multiple anonymized social networks alignment. In *ICDM*, 2015.

[13] J. Zhang, P. Yu, and Z. Zhou. Meta-path based multi-network collective link prediction. In *KDD*, 2014.