Social Network Integration: Towards Constructing the Social Graph

ABSTRACT

In this work, we formulate the problem of social network integration. It takes multiple observed social networks as input and returns an integrated global social graph where each node corresponds to a real person. The key challenge for social network integration is to discover the correspondences or interlinks across different social networks.

We engaged an in-depth analysis across six online social networks, twitter, livejournal, flickr, lastfm and myspace in order to address what reveals users’ social identity, whether the social factors consistent across different social networks and how we can leverage these information to perform integration.

We proposed a unified framework for the social network integration task. It crawls data from multiple social networks and further discovers accounts correspond to the same real person from the obtained networks. We use a probabilistic model to determine such correspondence, it incorporates features like the consistency of social status and social ties across different, as well as one-to-one mapping constraint and logical transitivity to jointly make the prediction. Empirical experiments verify the effectiveness of our method.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Text Mining; H.2.8 [Database Management]: Database Applications; H.4.m [Information Systems]: Miscellaneous

General Terms
Algorithms, Experimentation

Keywords
Social network, Predictive model, Social influence

1. INTRODUCTION

Online social media capture the digital trace of our social life. The diverse functionality of different on-line social media services make us join multiple social networks. We stay in touch with our friends on facebook, follow celebrities on microbloging sites and manage business contacts through linkedin. As a result, our social life has been broken into pieces due to the absent of the interlink between each social network. Is it possible to connect these pieces together, in order to recover the full image of our social life?

The advantage of such an process is multi-folded. For an individual user, managing multiple social network accounts takes a lot of effort. Bridging each isolated social circles of a user could lighten the communication cost and reduce information redundancy. From a service provider’s perspective, aggregating information from different social networks gives a more comprehensive view of each individual user and filters out the potential bias, hence it would benefit many applications. Taking recommendation system as an example, it would be a straightforward solution to cold start problem by leveraging information from the accounts on other

Figure 1: This figure illustrates the process of integrating four observed social networks $G_1, G_2, G_3$ and $G_4$ in order to recover the underlying social structure. Reversely, these social networks are subset of integrated global social graph. Red colored nodes are users that have accounts on multiple social networks and red edges represented co-occurrence social ties across different social networks.
sites of a new user. Moreover the inter-connections between different social networks might reveal interesting yet unobservable phenomenons when we look at an individual social network independently, such as the information diffusion pattern across different social media. However, this could also lead to potential leakage of privacy as it exposes users’ identity without permission.

In this work, we are trying to tackle the problem of social networks integration, where integration means combining the accounts correspond to the same real person across different social networks, in order to get a more complete view of that person’s social life. We want to investigate whether, how and to which extent we can fulfill this task.

Figure 1 gives a toy example illustrating the process of integrating four observed social networks $G_1, G_2, G_3$ and $G_4$ in order to recover the structure of the underlying global social graph $G_0$. The upper side of Figure 1 is the input of our problem: observed social networks. Nodes represent accounts and edges represent social ties within each individual social network. At the bottom shows the output of the problem: the global social graph integrates the four observed social networks. Users who have accounts on multiple social networks are colored with red. Such an intuitive process is non-trivial due to the following reasons. Firstly, lacking of unique identifier. There is no such an attribute that can uniquely identifies a person across different social media, even name won’t satisfy the condition since a single name can be shared by different person and also can be represented in different ways. Secondly, the heterogeneity of different social networks, where the heterogeneity reflected in both topology and semantics on different social networks. As a result, the features of an individual user are not necessarily be consistent across different social networks. The third challenge comes from data availability, because of the privacy policy and the large and the on growing data scale, we can hardly observe the entire network.

Considering the above intuitions, in the paper, we propose a social network integration framework which effectively integrates arbitrary number of social networks. In the framework, the data is obtained a set of ad-hoc crawlers each corresponds to an individual social networks. A probabilistic model is used to discover the accounts on each network that corresponds to the same real person, hence integrates the obtained social network. The result of integration will be further feedback to the crawlers to adjust crawling priority.

In summary, our contributions in this work include:

- We formulate the problem of social networks integration, and discuss associated challenges and potential;
- We propose a unified social network integration framework that takes multiple social networks as input, and returns a integrated social graph.
- Empirical experiments on integrating three large scale social networks, Linkedin, Videolectures and ArnetMiner demonstrate the effectiveness of our framework.

The rest of the paper is organized as follows. Section 2 formally defines the problem of social network integration; Section 3 gives an in-depth analysis on the three social networks; Section 4 is devoted to our social network integration framework; Section 6 analyzes experimental results; Section 7 presents related works and Section 8 concludes this work and discuss future directions.

### Table 1: Symbols and Notations

| $G$ | A set of social networks $G_1, G_2, \ldots, G_K$ |
| $G_0$ | The global social graph given $G$ |
| $v_i^k$ | A given node from $G_i$ |
| $C_{i,j}$ | The candidate set between $G_i$ and $G_j$ |
| $c_{i,j}^k$ | A given corresponding pair in $C_{i,j}$ |
| $O_{i,j}$ | The alignment of $G_i$ and $G_j$ |
| $o_{i,j}^k$ | A given corresponding pair between $G_i$ and $G_j$ |

## 2. PROBLEM DEFINITION

In this section, we give several necessary definitions and present the formulation of social network integration problem.

Given $k$ observed social network $G = \{G_1, G_2, \ldots, G_k\}$. Each social network is an unweighted undirected graph $G_i = (V_i, E_i)$, where $V_i = \{v_i^1, \ldots, v_i^{V_i}\}$ is a set of social network accounts and each edge $(v_i^{k_1}, v_i^{k_2}) \in E_i$ represents a social tie or social interaction between two accounts $v_i^{k_1}$ and $v_i^{k_2}$ within social network $G_i$. Social network integration problem can be divided into the following two sub-tasks:

**Probabilistic Alignment.** An alignment $O$ is a set of triplets $o_{i,j} = (v_i, v_j, p_k)$ each denotes an assertion that $v_i$ from network $G_i$ and $v_j$ from network $G_j$ potentially belongs to the same natural person, where $p_k$ is a real valued probability indicates the confidence. The aim of probabilistic alignment task is to discover such inter-network correspondence within the observed social networks $G$.

**Social Graph Construction.** With the alignment generated at the last step, we further constructing the social graph, where global social graph $G_0(V_0, E_0)$ is the underlying global network structure given a set of observed social network $G_i$. It is represented as a multi-graph, where each node $v_0^i \in V_0$ is a set of node in different networks that corresponds to an individual natural person. An edge $e_0^{k_1,k_2}$ between $v_0^{k_1}$ and $v_0^{k_2}$ will be created if there is an edge between two nodes $v_i^{k_1} \in v_0^{k_1}$ and $v_i^{k_2} \in v_0^{k_2}$ in any social network $G_i$. Given social networks set $G = \{G_i\}$, our goal is to reconstruct the underlying social graph $G_0$ by integrating all the social networks in $G$. For any node $v_0^i$ in $G_0$, we know exactly the set of $\{v_i^{k_1} \in v_0^{k_1}\}$.

## 3. DATA AND OBSERVATIONS

### 3.1 Data Collections

As we addressed in Section 1, to perform network integration, one of the challenges comes from data availability. Obtaining data for social network integration is not an easy task as in most cases, we won’t be able to observe the complete networks due to the size and privacy policy. In this work, we collected data from three social networks: LinkedIn1, VideoLectures2 and ArnetMiner3.

LinkedIn’s “Also-View” network: Linkedin operates the world’s largest professional network on the Internet. It

1http://www.linkedin.com/
2http://www.videolectures.org/
3http://www.arnetminer.org/
has more than 238 million members in over 200 countries and territories\footnote{http://press.linkedin.com/about}. On Linkedin network, part of user information is available for public profiles, such as education, affiliation and industry\footnote{http://videolectures.net/site/about/} (refer to Figure 5). In addition, users could maintain a certain amount of connections, which could be friends, co-workers, etc. However, personal connections are viewed as private information and unavailable to public access. But we discovered through the “Viewers of this profile also viewed...” table at the right side of public profiles. Hence, we construct an undirected network based on the “also-view” links, which can be considered as a k-nearest neighbor network. We crawled approximately 3 million public profiles and more than 25 million “also-view” relationships.

Videolectures’ “Together with” network: Videolecture is an open access educational video lectures repository. The lectures are given by distinguished scholars and scientists at most important events like conference, summer schools and workshops\footnote{http://videolectures.net/site/about/}. Connections between 11777 profiles we crawled are not explicitly provided by Videolecture. However, researcher attending the same venue can view as a kind of social relationship or interaction. Hence we extracted 786353 “Together with” relationships out of all profiles. In Videolecture’s “together with” network, an undirected edge $e = (v_i, v_j)$ would be created if $v_i$ and $v_j$ have at least co-attended a venue.

ArnetMiner’s co-author network: Arnetminer aims to provide comprehensive search and mining services for researcher social networks. We obtained the entire ArnetMiner data up to year 2013, including user profiles and co-author relationships. The details of user profiles ontology are shown in Figure 5. To construct co-author network, we create a undirected edge $e = (v_i, v_j)$ if author $v_i$ and $v_j$ have coauthored a publication. Finally, we obtain a co-author network with more than 1 million nodes and nearly 4 million edges.

The statistics of the three networks are illustrated in Table 2.

| Observed social network | # node | # edge |
|-------------------------|--------|--------|
| Linkedin                 | 2985414| 25965384|
| ArnetMiner              | 1053188| 3916907 |
| Videolecture            | 11178  | 786353  |

Table 2: Data statistics

Crowdsourcing data labeling: To construct training set and evaluate our method, we need ground truth of alignments between different social networks. Labeling such data is a time-consuming and labor intensive work. It requires human annotator to manually searching for the accounts belong to a certain person on different social networking service.

We leveraged crowdsourcing technique to ease the labeling work. An editable column has been provided on academic search service we developed for user to fill in the urls link to the external social network accounts of the correspond-

| Labeled Pairs | $|L(A, *)|$ | $|L(L, *)|$ | $|L(V, *)|$ |
|---------------|---------|---------|---------|
| ArnetMiner    | -       | 7532    | 1470    |
| Linkedin      | 7532    | -       | 410     |
| Videolecture  | 1470    | 410     | -       |

Table 3: Statistics of labeled pairs between different social networks, where $|L(+, +)|$ denotes the number of labeled corresponding pairs between the given two networks.

3.2 Analysis

As the first step, we engage in some high-level investigation of the generality and particularity of different social networks. Hence, we want to answer two questions:

- What reveals our social identity?
- What is consistent across different social networks?

3.2.1 What reveals our social identity?

Name: Username is the identifier of a user on social media sites. We make the assumption that users tend to use similar user name in different social networks. Such an intuition becomes more obvious since we are dealing with real name social networks in this work. However, even if in real name social networks, name wouldn’t necessarily gives away user’s identity since: 1) A name can be shared by different person. Figure 2 shows the statistics of account sharing an individual name in each social network, it is not surprising to see a few names are shared by more than 100 accounts. While for Videolectures network, there is not as much name duplication since the data scale is smaller. 2) A name can be represented in different ways, as in ArnetMiner, many authors’ first name only contains the initial letter. One intuition is that rare used names are likely owned by a single real person. For example, given two accounts both named “Wei Wang”, we wouldn’t have much confidence to claim they belong to the same user since there are so many people using this name. On the contrary, if two accounts share the name “Francis Scott Key Fitzgerald”, it is safe for us to make that assumption. So in a word, we can’t simply integrate accounts with the same name, but at least it gives us some clue.

Profile: Profile information is the description of a person given by itself, which is very informative. Unfortunately, it’s not easy to directly use due to the following two reasons. Firstly, profile ontology on different social networks are heterogeneous. Figure 3 illustrates the profile structure of ArnetMiner and Linkedin. Both ArnetMiner and Linkedin have name, education and locality, but the rest are different. However, in most of the cases, group and skills in Linkedin profile could be mapped to research interests in ArnetMiner; and summary has similar contents in Bio. In addition, homepage in Linkedin is sometimes contained in ArnetMiner’s contact, and the same applies to industry (in
LinkedIn) and work (in ArnetMiner). Hence, information is unstructured and distributed. As for Videolectures, we only have name, affiliation and homepage. As a result, the heterogeneous nature of different social networks makes no guarantee of similar profile structure or structured information, which poses a big challenge to our algorithm design.

Secondly, user provided profiles are noisy and incomplete. According to our statistics, more than half of the LinkedIn users do not have summary, interests, specialties, etc, such a problem is even more serious on ArnetMiner network. In user profiles, some fields can uniquely identify a person, such as Email, homepage url. These information are not commonly available but once we discover two accounts share such kind of attribute, we can safely conclude that they correspond to the same person.

Friends: "Friends make us who we are" [6]. It is possible that the social contacts we created within the social networks expose our identity. We have already mentioned the existence of consistent social ties that co-occurs on different social network. What we want to address here is that different friend reveals our identity to varying degree. Which is quite straightforward since when someone introduced himself as a fan of lady gaga, it almost gives away nothing about his social identity while if he mentioned he is a friend of someone not so famous, at least his identification has been narrowed down to a finite set of person. Further, given two accounts $v^1_i$ and $v^2_j$, if we already known they share a set of friends, can we claim they belongs to the same person? An obvious intuition is that the more shared friends they have, the more likely the accounts belongs to one person, but it’s not the only factor. Imagine if a person is connecting with a strong connected component, according to social balance theory [], the probability of existing another person connecting with that strong connected component is clearly larger than connecting a bunch of person scattered in different communities. Such a intuition can be further confirm by sub-graph frequency of networks[23].

3.2.2 The consistency across different social networks

Social Status: In sociology, social status is the position or rank of a person or group within a social system. Social status indicates the influential ability and the role a user plays in the social network. As an important social factor on every social network, we want to see whether the social status of an individual person is consistent on different social systems. For simplicity, we use degree as the measurement of social status of a user, which suggests that the person with more friends has the higher social status. We ranked the labeled pairs between ArnetMiner network and LinkedIn network by it’s degree on the two network respectively and plotted on a two dimensional space. In Figure 4, X-axis indicates the rank on ArnetMiner and Y-axis for LinkedIn, the color indicates the density of each region.

From Figure 4 we can see that data points tend to appear near the diagonal which suggests to some degree, the social status can be considered as consistent. In addition, we can clearly see the left bottom corner is the densest region which shows the social status of opinion leaders are more consistent than others. This observation is very intuitive since building reputation and becoming an opinion leader on a social network is an expansive process while being an opinion leader on a social network shows the users have enough ability and social capital, therefore he/she is more likely to also be an opinion leader on a different social network.

Social Ties: The semantic of relationships or social ties within different social networks are interpreted differently. As we have addressed in 3.1, the edge in ArnetMiner network represents co-authorship while it is interpreted as business
contact or coworker relationship in Linkedin, and in Videolectures network, it represents two person co-attending a venue. Although the meaning of them are different, intuitively we believe that there will be co-occurrence of social ties across different social networks. The explanation is straight forward, the social tie within a network implies two persons already know each other, so they are more likely to create a relationship in another social network than two random persons. We investigate the consistancy of social tie on the labeled corresponding pairs between ArnetMiner network and Linkedin network. Figure 5 shows the number of co-occurred social tie between these users. Here, we say a social tie is co-occurred if it exists on both social networks. There are 627 co-occurred social ties, while 1670 appear on Linkedin but absent on ArnetMiner and 7773 conversely. Note that we only partially observed the Linkedin network due to the privacy protection, hence, we can safely assume that there will be much more co-occurred social tie if the complete network is available.

4. NETWORK INTEGRATION PIPELINE

Base on the observations above, we proposed a unified framework to integrate multiple observed social networks in order to recover the underlying structure of the global social graph. Figure 7 shows overview workflow of our social network integration framework. The profile accounts crawled from different social networks will be first converted to candidate pairs. Then, the candidate pairs will be feed into a probabilistic model in order to determine the correctness of each candidate pair.

4.1 Candidate Pairs Generation

Finding pairs of accounts on two social networks $G_i(V_i, E_i)$ and $G_j(V_j, E_j)$ that belong to the same natural person, the scale of search space is $O(|V_i| \cdot |V_j|)$, and even larger for integrating more than two social networks. Such a task is intractable for real world networks with millions of nodes. It is obvious that only very few pairs of accounts are actually belong to the same person, hence we need to quickly and roughly filter out those pairs that are unlikely to be true. Therefore, we use a candidate pairs generation step to reduce the search space. A desired candidate generation method should have following characteristics:

- Easy to implement
- Scalable to very large networks.
- Generate small candidate sets meanwhile covers most of the corresponding pairs.

As in , we make the assumption that user tend to use similar user name in different social networks. Such an intuition becomes more obvious since we are dealing with real name social networks in this work. Here we take this heuristic to generate candidate pairs that are potentially belong to an individual natural person.

To tolerate miss spell and different representation of names, we decompose each name into segments and group the accounts sharing at least one segment in the name. To further shrink the size of candidate sets, we compute jaro-winkler distance between the name of accounts within a group. Finally, an appropriate threshold is chosen to trade off between candidate size and coverage rate. Section 6.1 presents empirical analysis on different thresholds. As we can see, our method is easy to implement; scalable to large social networks; and as showing in Section 6.1, it yields high coverage rate of corresponding pairs. In addition, this method fits online streaming data well. Candidate sets could be effectively updated incrementally with the incoming data.

4.2 Learning To Align Corresponding Pairs

With the candidate pairs generated at last step, we need to further determine which of them are truly corresponding pairs, which is a binary classification task to assign the true or false label for each candidate pair.

We formulate the classification problem into a probabilistic graphical model in order to capture the structural correlations and the uncertainty in predictions. Each candidate pair $c^k \in \cup C_{i,j}$ is mapped to an observed variable $x_k$. $Y = \{y_k\}^{\cup C_{i,j}}_k$ are hidden variables that represents the true or false labels of $X = \{x_k\}^{\cup C_{i,j}}_k$. To better incorporate the observations discussed in Section 3.3, we use a factor graph to represent the probabilistic model. We defined two types of feature functions:

- Local feature function: $f(y_k, x_k)$ captures the characteristics between the two accounts $v_i$ and $v_j$ associated with the latent variable $y_k$.
- Correlation feature function: $g(y_k, y_k', G(y_k, y_k'))$ denotes the correlation between two latent variables $y_k$ and $y_k'$.
- Group constraint function: $h(y_k, H(y_k, C_k))$ represents...
the correlation between a latent variable \( y_k \) and a constrained group \( C_k \).

- Triangle feature function: \( q(y_k, y_{k'}, y_{k''}) \) denotes the correlation within the three latent variables \( y_k, y_{k'} \) and \( y_{k''} \) form a triangle closure.

Figure 6 gives a graphical representation of the proposed model. Given the observed social networks \( G \), candidate pair set \( C = \cup C_{i,j} \), we can define the joint distribution over \( Y \) as:

\[
p(Y|G, C) = \prod_k f(x_k, y_k)g(y_k, \mathcal{G}(y_k, y_{k'}))
\]

\[
h(y_k, \mathcal{H}(y_k, C_k))q(y_k, \mathcal{Q}(y_k, y_{k'}, y_{k''}))
\]

(1)

### 4.3 Feature Design

#### Modeling the Uniqueness of Name: As we mentioned in 3.3.1, an unique name are likely to be own by a single natural user. To measure the uniqueness of a name, we use a language model based approach. Given a name \( N = (s_1, s_2, ..., s_n) \), where \( s_i \) is the segment obtained at candidate generation step. We define uniqueness of a name as \( \mathcal{U}(N) = \sum_{i=1}^{n} \log P(s_i) \). For each candidate pair \( c_{i,j}^k = (v_{i}^k, v_{j}^k) \), a feature \( \mathcal{U}(c_{i,j}^k) = (\mathcal{U}(N(v_{i}^k)) + \mathcal{U}(N(v_{j}^k)))/2 \) is assigned to capture whether the two accounts are sharing a unique name.

**Modeling Profile Similarity:** Hence, we categorized user profile fields into Strong Identifier and Weak Identifier. We calculate the edit distance between each shared strong identifier for a given candidate pairs as a set of features. For weak identifiers, we concatenate them into a single document and convert each document into a bag-of-words vector. For each vector, the words are weighted by TF-IDF. The similarity between the profiles of each candidate pairs are measured by inner product and cosine distance.

**Modeling Social Status Shift:** Based on the observation in Section 3.3.1, the pattern of social status shifting across different social networks can give us discriminative power to determine true or false label of candidate pairs. To incorporate this observation into our predictive model, we first determine the social status of each account within it's social network, again, we use degree to measure social status. We sorted accounts by degree in each social networks and we call the top 1% accounts "opinion leaders", the following 10% "middle class", and for the rests, we call them "the messes". Given a candidate pair \( c_{i,j}^k = (v_{i}^k, v_{j}^k) \), where \( v_{i}^k \) is an opinion leader in \( G_i \) and \( v_{j}^k \) is a middle class in \( G_j \), a feature \( \text{fromInG}_i, \text{toMinG}_j \) will be assigned to it.

The features defined above capture the characteristics between within a candidate pairs, hence, we incorporate them as local features, which is defined as an exponential function:

\[
f(y_k, x_k) = \frac{1}{Z_{\alpha}} \sum_a \alpha_a \Phi_d(y_k, x_k)
\]

(2)

Where \( \Phi_d(.) \) is the \( d^{th} \) local feature defined on \( x_k \) with respect to the value of \( y_k \); \( \alpha_d \) is the weight of the feature; \( Z_{\alpha} \) is a normalization factor.

**Modeling Co-occurred Social Ties:** According to Section 3.3.1, a social tie may co-occur on different social networks if both of the two associated users have accounts on those networks. However such a strong signal is not straightforward to capture since we do not know the correspondence between social ties before identifying the users associated with them. Here, we model the co-occurred social ties as the structural correlation between candidate pairs. To make it more intuitive, Figure 7 gives a toy example illustrating this process. On the left side are two networks \( G_1 \) and \( G_2 \). On the right side, each node represents a candidate pair, and each edge represents a potential co-occurred social tie. In the example, we assume any two node from the two network is a candidate pair, hence there are \( 3 \times 3 = 9 \) nodes on the right. We construct the potential co-occurred social ties set \( \text{CST} \) with the following process. For each two candidate pairs \( c^k = (v_{i}^k, v_{j}^k) \) and \( c^{k'} = (v_{i}^k, v_{j}^k') \), we draw an edge between them if there is an edge \( e_{i}^{k_1,k_1'} = (v_{i}^{k_1}, v_{i}^{k_1'}) \) in \( G_1 \) and an edge \( e_{j}^{k_2,k_2'} = (v_{j}^{k_2}, v_{j}^{k_2'}) \) in \( G_2 \) respectively, we say \( e_{i}^{k_1,k_1'} \) and \( e_{j}^{k_2,k_2'} \) are potential co-occurred social ties between \( G_1 \) and \( G_2 \) and there is correlation between \( c^k \) and \( c^{k'} \).

Co-occurred social tie is formulated as a correlation feature which is defined as:

\[
g(y_k, y_{k'}, \mathcal{G}(y_k, y_{k'})) = \frac{1}{Z_{\beta}} e^{\beta g(y_k, y_{k'})}
\]

(3)

\((y_k, y_{k'})\) is an indicator to specify whether there is a potential co-occurred social tie between two candidate pairs, thus, \( g(y_k, y_{k'}) = 1 \) if there is, and 0 otherwise.

**Modeling One-to-one Mapping Constraint:** Users may create alias accounts on social networks, while in most cases users will stick to a single account since managing one account is hard enough. Hence we set the constraint that
Figure 9: Modeling logical transitivity on integrating three social networks $G_1$, $G_2$ and $G_3$

one-to-one mapping is encouraged between two networks. Formally, when integrating network $G_i$ and $G_j$, we say a set of candidate pairs are constrained by $v^k_i$ if all of these candidate pairs contains node $v^k_i$ since only one of them are likely to be true, and we call it a constrained group $C_k$. Figure shows the one-to-one mapping constraint when integrating networks $G_1$ and $G_2$. As an example, the three candidate pairs $(v^1_1, v^1_2), (v^1_2, v^1_3)$ and $(v^2_1, v^2_2)$ connecting by red edges are constrained by $v^1_2$ since all of them are associated with $v^1_2$. We use $MC = \{C_k\}$ to denote all the one-to-one mapping constrained group.

One-to-one mapping constraint feature is defined as

$$h(y_k, \mathcal{H}(y_k, C_i)) = \frac{1}{Z_i}e^{\gamma h(y_k, C_i)} \tag{4}$$

where $C_i \in MC$ is a constrained group that $y_k$ in. $h(y_k, C_i)$ is set to 0 if more than two of variables in $C_i$ labeled true, since it violates the one-to-one mapping constraint, and 1 otherwise.

### Modeling Logical Transitivity

Since the assertion of account A and account B belong to the same user is symmetrical, there is logical transitivity between related mapping. Figure 9 shows the logical transitivity when integrating three social networks $G_1, G_2$ and $G_3$, where each transitivity closure is indicated by an individual color. For instance, the three candidate pairs $(v^2_1, v^2_2), (v^2_1, v^2_3)$ and $(v^3_1, v^3_2)$ forms a transitivity closure, and constrained by the fact that if two of them are labeled true, the other one must be true. Hence when we dealing with more than two social networks, we can to incorporate this kind of constraint to further improve our method. We use $LT$ to represent all such transitivity closures.

Logical transitivity feature is defined as a triangle feature:

$$q(y_k, y'_{k'}, y_{k''}, Q(y_k, y'_{k'}, y_{k''})) = \frac{1}{Z_o}e^{\eta q(y_k, y'_{k'}, y_{k''})} \tag{5}$$

where $y_k, y'_{k'}, y_{k''}$ are nodes forming a logical transitivity closure. $q(y_k, y'_{k'}, y_{k''})$ would be set to 0 if two of them are labeled true and the other one is labeled false since it violates the logical transitivity constraint, otherwise $q(y_k, y'_{k'}, y_{k''})$ is set to 1. As can be seen from Figure 9, logical transitivity connects nodes from different network pairs, thus capturing the correlations across multiple social networks.

### 4.3.1 Learning

Model learning is to estimate the best parameter configuration $\theta = (\lambda, \beta, \gamma, \eta)$ that maximizes the log-likelihood objective function. According to Hammersley-Clifford theorem, we can obtain the following log-likelihood function.

$$O(\theta) = \log P_o(Y|G, C)$$

$$= \sum_{k=1}^{\lvert MC \rvert} \left[ \sum_{d=1}^{\lvert ST \rvert} \alpha_d \Phi(y_k, x_k) \right] + \sum_{(y_k, y_{k'}) \in CST} \beta g(y_k, y_{k'})$$

$$+ \sum_{(y_k, y'_{k'}, y_{k''}) \in LT} \gamma h(y_k, C_i) - \log Z \tag{6}$$

where $Z = Z_1 Z_2 Z_3 Z_4$ is a normalization factor.

For representation simplicity, we concatenate all feature functions for variable $y_k$ as

$$\Psi(y_k) = (\Phi(y_k, x_k)^T, g(y_k, y'_{k'})^T, h(y_k, C_i)^T, q(y_k, y'_{k'}, y_{k''})^T)$$

We adopt gradient decent method to maximize the objective function. The gradient can be calculated as following

$$\frac{\partial O(\theta)}{\partial \theta} = E[\Psi(y_k)] - E_{P_o(Y|G, C)}[\Psi(y_k)] \tag{8}$$

The term $E[\Psi(y_k)]$ in Eq.8 is easy to calculate, but the second term is intractable to directly calculate. In this work, we use Loop Belief Propagation to approximate the marginal probability $p(y_k|\theta)$. Hence the gradient is obtained by summing over all nodes.

### 4.3.2 Inference

Based on the learned parameters $\theta$, unknown labels can be predicted by finding the optimal configuration of $Y$ that maximizes the joint probability in Equation (3), that is,

$$Y^* = \arg \max_{Y^*} P_o(Y|G, C)$$

Utilizing marginal probabilities $p_i|Y, G$ estimated by LBP, whether a node is an overlapping user pair is labeled according to whether it has the largest marginal probability of being labeled as 1 among all the nodes constrained by subgraph constraint factor. The higher the probability is, the more confidence we have in its correctness. For inference, we use the max-sum algorithm (the max version of Eqs. 8
and 9) to find the values of \( Y \) that maximizes the likelihood. This max-sum algorithm is similar with the sum-product algorithm, except for the message passing functions, which calculate the message according to max instead of sum.

4.4 Constructing the Social Graph

Adaptive Data Crawling Strategy Due to the highly dynamic characteristic and the huge data scale, it is almost impossible for us to obtain the entire social network. In our framework, the data are feeding in incrementally by a set of ad-hoc social network crawlers. In traditional social network crawling task, an often used technique is breath-first search which iteratively obtains data of nodes and their neighbors. However such a straightforward crawling strategy is not suitable for social network integration task since it requires overlap of different social networks. Moreover, as we discussed above, friends reveals our identities, it is much easier to determine whether to align a pair of accounts when most of its neighbors has already been aligned. Thus, we prefer to obtain the accounts close to the aligned pairs first.

In the framework, we store the accounts waiting to be crawled in a priority queue. After each time of alignment phase, we iterate all the aligned pairs, for each pair \( c_{i,j} = (v^1_i, v^2_j) \), we give one “credit” score to each of their neighbors \( NB_i(v^1_i) \) and \( NB_j(v^2_j) \). The priority is determined by the received credit of each account. Hence, we can avoid low overlap rate between networks giving return by blind BFS crawling.

Global Social Graph Construction With the alignment output by the probabilistic model, we can finally construct the global social graph. Follow the definition given by Section 2 that global social graph \( G = (V_0, E_0) \) is a multigraph where each node \( v^k \) in \( V_0 \) corresponding to a set of nodes on different networks belonging to an individual natural person. We first group associated aligned pairs into a graph where each node \( v^k \) in \( V_0 \) corresponds to all the nodes within the set. We iteratively incorporate edge for the nodes from each social network and create a new node for any node don’t have correspondence in \( G_0 \). Based on the constructed global social graph, we developed an integrated entity search application which users can search across different social networks. We won’t discuss this part of the work in this paper.

5. EXPERIMENTAL RESULTS

In this section, we first introduce the data sets for experiments, and then present experimental results as well as empirical analysis.

5.1 Experiment Setup

Candidate Pairs Generation We perform candidate pairs generation using the method explained in Section 4. Recall in Section 4 we mentioned the preferred candidate generation method returns candidate pair set with relatively reasonable size meanwhile covers as much corresponding pairs as possible. Hence, to select an appropriate threshold, we need to make a trade off between these two factor. We evaluate the quality of candidate generation on the labeled data. Figure shows the candidate coverage rate and the size of the generated candidate pair set with respected to threshold between each pair of networks. In Figure we can see that the size of candidate size in growing exponentially with the decreasing of threshold. We choose a threshold 0.8 which yields high coverage rate(higher than 98% between ArnetMiner & Videolectures and Linkedin & Videolectures and close to 95% between ArnetMiner & Linkedin) and reasonable candidate set size.

Comparison Methods We compare the following methods with our probabilistic model at corresponding pair alignment phase.

- **Strict Name Matching (SNM).** This method only uses names to estimate the correctness of a given candidate pair. For each candidate pair, it predicts positive if and only if the two associated accounts have the exact same name.
- **Unique Name Matching (UNM).** According to the discussion in Section 4.2.1, this method only returns positive for the candidate pairs with the exact same name \( \mathcal{N} \) meanwhile the uniqueness of the name \( U(\mathcal{N}) > 20 \).
- **Logistic Regression(LR).** This method incorporates all the local features discussed in section 4.2 to train a classification model and the employs the classification model to predict the correctness of candidate pairs.
- **Conditional Random Field (CRF).** It trains a conditional random field [14] with the local features associated with each candidate pair and structural correlations between candidate pairs, where correlations are indicated by the potential persistent social across different social network introduced in Section 4.2.
- **Our Method.** The proposed approach in the paper, which leverages the local features as well as the persistence of social tie, one-to-one mapping constrain and logical transitivity constrain to make the prediction.

Evaluation Measure We evaluate the performance of each method in terms of precision, recall and F1-score.

5.2 Performance Analysis

We compare the performance of the five method for corresponding pairs alignment task. Note that our model is trained over all three social networks ArnetMiner, Linkedin and Videolectures, and making prediction jointly. But for clarity, we present the experimental result for each pair of social networks both separately and combined.

Table 4 shows the empirical results for integrating three social networks: ArnetMiner, Linkedin and Videolectures. According to Table 4, our method significantly outperforms the baselines. The two rule-based baseline methods SNM and UNM only use username to make the prediction. SNM strictly aligns all the candidate pairs with the exact same username. For a commonly used name, it will predict all the candidate pairs sharing that name positive hence yields low precision. UNM considers the uniqueness of name and only considers those candidate pairs sharing a rarely used name as they belong to the same person, which gives a high precision, while recall suffers. However, as a result, the performance of the two simple rule-based method confirms that unique name is a strong indicator for corresponding pairs.

The improvement between CRF and LR verifies the effectiveness of persistent social ties for corresponding pairs alignment and further confirms the intuition that friends reveals our social identity.
Figure 10: Coverage rate and size of the generated candidate pair set w.r.t threshold

Table 4: Performance comparison of different methods for corresponding pair alignment task. The result are presenting combined and separately for the three network pairs.

| Data Set         | Method | Prec. | Rec.  | F1-score |
|------------------|--------|-------|-------|----------|
| ArnetMiner & Linkedin | SNM    | 0.5950| 0.6994| 0.6430   |
|                  | UNM    | 0.9081| 0.5141| 0.6565   |
|                  | LR     | 0.6765| 0.8719| 0.7618   |
|                  | CRF    | 0.8484| 0.8315| 0.8398   |
|                  | Our Method | 0.8602| 0.8674| 0.8638   |
| ArnetMiner & Videolectures | SNM    | 0.4656| 0.8410| 0.5994   |
|                  | UNM    | 0.9192| 0.6034| 0.7286   |
|                  | LR     | 0.4063| 0.8692| 0.5537   |
|                  | CRF    | 0.6919| 0.8710| 0.7710   |
|                  | Our Method | 0.7076| 0.8946| 0.7901   |
| Linkedin & Videolectures | SNM    | 0.1175| 0.8222| 0.2055   |
|                  | UNM    | 0.3045| 0.5889| 0.4015   |
|                  | LR     | 0.1701| 0.7333| 0.2762   |
|                  | CRF    | 0.3748| 0.7990| 0.5102   |
|                  | Our Method | 0.5012| 0.8667| 0.6531   |
| All together     | SNM    | 0.4859| 0.7522| 0.5904   |
|                  | UNM    | 0.8641| 0.5473| 0.6702   |
|                  | LR     | 0.5209| 0.8673| 0.6508   |
|                  | CRF    | 0.6419| 0.8041| 0.7158   |
|                  | Our Method | 0.7096| 0.8762| 0.7841   |

6. RELATED WORKS

Due to the wide potential applications, user identification across multiple social networks has attracted more and more attention recently. Zafarani et al. [24] first addressed the similar problem and proposed a search based attracting. Zafarani et al. [25] takes a behavioral modeling approach models the behavioral pattern of users when selecting username, hence to further determine whether two usernames belong to the same person. Iofciu et al. [10] investigates the user linking problem in within a social tagging system. Liu et al. [17] leverages rare username to conduct training instance for user linking task. Zhang et al. [13] formulates the problem as a anchor link prediction task and leverages temporal and spatio information to make the prediction.

From a different angle, network integration could lead to potential leakage of privacy. A related field is called social network de-anonymization [1, 18]. These works mainly focused on recovering users’ identity from a masked social network. In [2][1], Backstrom et al. present such process where one can identify individuals in these anonymized networks by either manipulating networks before they are nonymized or by having prior knowledge about certain anonymized nodes. Narayanan et al. [18] attempt to de-anonymizing two social networks based only on the network topology.

Network alignment has been studied in many works, which preforms matching between two undirected graphs. Its task is to maximize the number of “overlapped” edges given two undirected graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and a set $L$ containing the feasible matching between $V_1$ and $V_2$. Due to the NP-hard nature of this definition, many different approaches are adopted to relax this constraint, such as Linear Programing[12], Belief Propagation[3] and so on. There also have been some successful applications under this framework, for instance, finding common pathways in biological networks [21, 20] and ontology alignment between Citeseer papers and DBLP papers[9].

Entity disambiguation or entity resolution is also a widely studied problem. Many existing works use Wikipedia for named entities as well as a source for entity disambiguation. Bunescu et al.[4] exploited the Wikipedia (entity pages, redaction pages, categories, and hyperlinks) and built a context-article cosine similarity model. Cucerzan et al. [5] presented a large scale named entity disambiguation system that also employed a huge amount of information extracted from Wikipedia. Other works use DBpedia to construct a graph of linked data entities as a way of disambiguating named entities. Instead of considering only ontology and taxonomy[7, 8], there are attempts to employ all the information from knowledge bases [11, 19]. These works adapted topical models for performing entity disambiguation that considered the context of every word as well as co-occurrence patterns among entities.

Our work is also closely related to relationship mining in social network analysis. One research topic is to predict unknown link in social networks. Liben-Nowell et al. [16] study the unsupervised methods for link prediction. Backstrom et al.[2] proposed a supervised random walk algorithm to estimate the strength of social link. Leskovec et al. [15]
employed a logistic regression model to predict positive and negative links in online social networks. Wenbin et al. [22] propose a Partially-labeled Pairwise Factor Graph Model for learning to infer the type of social ties.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of social network integration. We engaged an in-depth analysis across three social networks, LinkedIn, Videolectures and ArnetMiner and addressed what reveals users’ social identity, whether the social factors consistent across different social networks and how we can leverage this information to perform integration. Base on the observations, we proposed a unified framework for the social network integration task. It crawls data from multiple social networks and further discovers accounts corresponding to the same real person from the obtained networks. We use a probabilistic model to determine such correspondence, it incorporates features like the consistency of social status and social ties across different, as well as one-to-one mapping constraint and logical transitivity to jointly make the prediction. The empirical study shows that our framework can effectively integrate the three social networks which confirms the feasibility of the problem.

The future work falls into two parts. 1) In this work, we are dealing with real name social network, hence we can take the advantages of using name similarity to perform candidate generation in order to shrink the search space. In anonymous social networks, such a process wouldn’t be so straightforward, though we can still use username as a start point [17, 25]. 2) With the integrated global social graph, we want to further investigate several interesting topics such as information diffusion and user behavior pattern across different social networks that only feasible on the global social graph.

8. REFERENCES

[1] L. Backstrom, C. Dwork, and J. M. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In WWW’07, pages 181–190, 2007.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In Proceedings of the fourth ACM international conference on Web search and data mining, pages 635–644, ACM, 2011.

[3] M. Bayati, D. Bleich, A. Saberi, and Y. Wang. Message passing algorithms for sparse network alignment. arXiv preprint arXiv:0907.3338, 2009.

[4] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In Proceedings of EACL, volume 6, pages 9–16, 2006.

[5] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In Proceedings of EMNLP-CoNLL, volume 6, pages 708–716, 2007.

[6] C. Flora. Friendfluence: The Surprising Ways Friends Make Us who We are. Random House Digital, Inc., 2013.

[7] A. Gentle, Z. Zhang, L. Xia, and J. Iria. Semantic relatedness approach for named entity disambiguation. In Digital Libraries, pages 137–148. Springer, 2010.

[8] J. Hassell, B. Aleman-Meza, and I. Arpinar. Ontology-driven automatic entity disambiguation in unstructured text. The Semantic Web-ISWC 2006, pages 44–57, 2006.

[9] W. Hu, Y. Qu, and G. Cheng. Matching large ontologies: A divide-and-conquer approach. Data & Knowledge Engineering, 67(1):140–160, 2008.

[10] T. Iofciu, P. Funkhauser, F. Abel, and K. Bischoff. Identifying users across social tagging systems. In ICWSM’11, pages 1–1, 2011.

[11] S. Kataria, K. Kumar, R. Rastogi, P. Sen, and S. Sengamedu. Entity disambiguation with hierarchical topic models. In Proceedings of KDD, 2011.

[12] G. Klaau. A new graph-based method for pairwise global network alignment. BMC bioinformatics, 10(Suppl 1):S59, 2009.

[13] X. Kong, J. Zhang, and S. Y. Philip. Inferring anchor links across multiple heterogeneous social networks.

[14] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[15] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In Proceedings of the 19th international conference on World wide web, pages 641–650. ACM, 2010.

[16] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7):1019–1031, 2007.

[17] J. Liu, F. Zhang, X. Song, Y.-J. Song, C.-Y. Lin, and H.-W. Hon. What’s in a name?: an unsupervised approach to link users across communities. In WSDM’13, pages 495–504, 2013.

[18] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In IEEE Symposium on Security and Privacy’09, pages 173–187, 2009.

[19] P. Sen. Collective context-aware topic models for entity disambiguation. In Proceedings of the 21st international conference on World Wide Web, pages 729–738. ACM, 2012.

[20] R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In Research in computational molecular biology, pages 16–31. Springer, 2007.

[21] 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, 105(35):12763–12768, 2008.

[22] W. Tang, H. Zhuang, and J. Tang. Learning to infer social ties in large networks. Machine Learning and Knowledge Discovery in Databases, pages 381–397, 2011.

[23] J. Ugander, L. Backstrom, and J. Kleinberg. Subgraph frequencies: mapping the empirical and extremal geography of large graph collections. In Proceedings of the 22nd international conference on World Wide Web, pages 1307–1318. International World Wide Web Conferences Steering Committee, 2013.

[24] R. Zafarani and H. Liu. Connecting corresponding identities across communities. In ICWSM’09, pages 1–1, 2009.
[25] R. Zafarani and H. Liu. Connecting users across social media sites: A behavioral-modeling approach. 2013.
Social Network Integration:
Towards Constructing the Social Graph

ABSTRACT
In this work, we formulate the problem of social network integration. It takes multiple observed social networks as input and returns an integrated global social graph where each node corresponds to a real person. The key challenge for social network integration is to discover the correspondences or interlinks across different social networks.

We engaged an in-depth analysis across six online social networks, twitter, livejournal, flickr, lastfm and myspace in order to address what reveals users’ social identity, whether the social factors consistent across different social networks and how we can leverage these information to perform integration.

We proposed a unified framework for the social network integration task. It crawls data from multiple social networks and further discovers accounts correspond to the same real person from the obtained networks. We use a probabilistic model to determine such correspondence, it incorporates features like the consistency of social status and social ties across different, as well as one-to-one mapping constraint and logical transitivity to jointly make the prediction. Empirical experiments verify the effectiveness of our method.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: Text Mining;
H.2.8 [Database Management]: Database Applications;
H.4.m [Information Systems]: Miscellaneous

General Terms
Algorithms, Experimentation

Keywords
Social network, Predictive model, Social influence

1. INTRODUCTION

Online social media capture the digital trace of our social life. The diverse functionality of different on-line social media services make us join multiple social networks. We stay in touch with our friends on facebook, follow celebrities on microblogging sites and manage business contacts through linkedin. As a result, our social life has been broken into pieces due to the absent of the interlink between each social network. Is it possible to connect these pieces together, in order to recover the full image of our social life?

The advantage of such an process is multi-folded. For an individual user, managing multiple social network accounts takes a lot of effort. Bridging each isolated social circles of a user could lighten the communication cost and reduce information redundancy. From a service provider’s perspective, aggregating information from different social networks gives a more comprehensive view of each individual user and filters out the potential bias, hence it would benefit many applications. Taking recommendation system as an example, it would be a straightforward solution to cold start problem by leveraging information from the accounts on other...
sites of a new user. Moreover the inter-connections between different social networks might reveal interesting yet unobservable phenomenons when we look at an individual social network independently, such as the information diffusion pattern across different social media. However, this could also lead to potential leakage of privacy as it exposes users’ identity without permission.

In this work, we are trying to tackle the problem of social networks integration, where integration means combining the accounts correspond to the same real person across different social networks, in order to get a more complete view of that person’s social life. We want to investigate whether, how and to which extent we can fulfill this task.

Figure 1 gives a toy example illustrating the process of integrating four observed social networks $G_1$, $G_2$, $G_3$ and $G_4$ in order to recover the structure of the underlying global social graph $G_0$. The upper side of Figure 1 is the input of our problem: observed social networks. Nodes represent accounts and edges represent social ties within each individual social network. At the bottom shows the output of the problem: the global social graph integrates the four observed social networks. Users who have accounts on multiple social networks are colored with red. Such an intuitive process is non-trivial due to the following reasons. Firstly, lacking of unique identifier. There is no such an attribute that can uniquely identify a person across different social media, even name won’t satisfy the condition since a single name can be shared by different person and also can be represented in different ways. Secondly, the heterogeneity of different social networks, where the heterogeneity reflected in both topology and semantics on different social networks. As a result, the features of an individual user are not necessarily be consistent across different social networks. The third challenge comes from data availability, because of the privacy policy and the large and the on growing data scale, we can hardly observe the entire network.

Considering the above intuitions, in the paper, we propose a social network integration framework which effectively integrates arbitrary number of social networks. In the framework, the data is obtained a set of ad-hoc crawlers each corresponds to an individual social networks. A probabilistic model is used to discover the accounts on each network that corresponds to the same real person, hence integrates the obtained social network. The result of integration will be further feedback to the crawlers to adjust crawling priority.

In summary, our contributions in this work include:

- We formulate the problem of social networks integration, and discuss associated challenges and potential;
- We propose a unified social network integration framework that takes multiple social networks as input, and returns a integrated social graph.
- Empirical experiments on integrating three large scale social networks, Linkedin, Videolectures and ArnetMiner demonstrate the effectiveness of our framework.

The rest of the paper is organized as follows. Section 2 formally defines the problem of social network integration; Section 3 gives an in-depth analysis on the three social networks; Section 4 is devoted to our social network integration framework; Section 6 analyzes experimental results; Section 7 presents related works and Section 8 concludes this work and discuss future directions.

| $G$ | A set of social networks $G_1$, $G_2$, ..., $G_K$ |
| $G_0$ | The global social graph given $G$ |
| $v_i^k$ | A given node from $G_i$ |
| $C_{i,j}$ | The candidate set between $G_i$ and $G_j$ |
| $c_{i,j}^k$ | A given candidate pair in $C_{i,j}$ |
| $O_{i,j}$ | The alignment of $G_i$ and $G_j$ |
| $o_{i,j}^k$ | A given corresponding pair between $G_i$ and $G_j$ |

Table 1: Symbols and Notations

2. PROBLEM DEFINITION

In this section, we give several necessary definitions and present the formulation of social network integration problem.

Given $k$ observed social network $G = \{G_1, G_2, ..., G_k\}$. Each social network is an unweighted undirected graph $G_i = (V_i, E_i)$, where $V_i = \{v_i^1, ..., v_i^{V_i}\}$ is a set of social network accounts and each edge $(v_i^{k_1}, v_i^{k_2}) \in E_i$ represents a social tie or social interaction between two accounts $v_i^{k_1}$ and $v_i^{k_2}$ within social network $G_i$. Social network integration problem can be divided into the following two sub-tasks:

Probabilistic Alignment. An alignment $O$ is a set of triplets $o_{i,j}^k = (v_i, v_j, p_k)$ each denotes an assertion that $v_i$ from network $G_i$ and $v_j$ from network $G_j$ potentially belongs to the same natural person, where $p_k$ is a real valued probability indicates the confidence. The aim of probabilistic alignment task is to discover such inter-network correspondence within the observed social networks $G$.

Social Graph Construction. With the alignment generated at the last step, we further constructing the social graph, where global social graph $G_0(V_0, E_0)$ is the underlying global network structure given a set of observed social network $G$. It is represented as a multi-graph, where each node $v_0^k \in V_0$ is a set of node in different networks that corresponds to an individual natural person. An edge $e_0^{k_1,k_2}$ between $v_0^{k_1}$ and $v_0^{k_2}$ will be created if there is an edge between two nodes $v_i^{k_1} \in v_0^{k_1}$ and $v_i^{k_2} \in v_0^{k_2}$ in any social network $G_i$. Given social networks set $G = \{G_i\}$, our goal is to reconstruct the underlying social graph $G_0$ by integrating all the social networks in $G$. For any node $v_0^k$ in $G_0$, we know exactly the set of $\{v_i^{k'} \in v_0^k\}$.

3. DATA AND OBSERVATIONS

3.1 Data Collections

As we addressed in Section 1, to perform network integration, one of the challenges comes from data availability. Obtaining data for social network integration is not an easy task as in most cases, we won’t be able to observe the complete networks due to the size and privacy policy. In this work, we collected data from three social networks: Linkedin, VideoLectures and ArnetMiner.

Linkedin’s “Also-View” network: Linkedin operates the world’s largest professional network on the Internet. It

\[ \text{http://www.linkedin.com/} \]

\[ \text{http://www.videolectures.org/} \]

\[ \text{http://www.arnetminer.org/} \]
has more than 238 million members in over 200 countries and territories\(^3\). On Linkedin network, part of user information is available for public profiles, such as education, affiliation and industry (refer to Figure 5). In addition, users could maintain a certain amount of connections, which could be friends, co-workers, etc. However, personal connections are viewed as private information and unavailable to public access. But we discovered through the “Viewers of this profile also viewed...” table at the right side of public profiles. Hence, we construct an undirected network based on the ”also-view” links, which can be considered as a k-nearest neighbor network. We crawled approximately 3 million public profiles and more than 25 million “also-view” relationships.

**Videolectures’ “Together with” network:** Videolectures is an open access educational video lectures repository. The lectures are given by distinguished scholars and scientists at most important events like conference, summer schools and workshops\(^6\). Connections between 11777 profiles we crawled are not explicitly provided by Videolectures. However, researcher attending the same venue can view as a kind of social relationship or interaction. Hence we extracted 786353 “Together with” relationships out of all profiles. In Videolectures’s “together with” network, an undirected edge \(e = (v_i, v_j)\) would be created if \(v_i\) and \(v_j\) have at least co-attended a venue.

**ArnetMiner’s co-author network:** ArnetMiner aims to provide comprehensive search and mining services for researcher social networks. We obtained the entire ArnetMiner data up to year 2013, including user profiles and co-author relationships. The details of user profiles ontology are shown in Figure 5. To construct co-author network, we create a undirected edge \(e = (v_i, v_j)\) if author \(v_i\) and \(v_j\) have coauthored a publication. Finally, we obtain a co-author network with more than 1 million nodes and nearly 4 million edges.

The statistics of the three networks are illustrated in Table 2.

**Crowdsourcing data labeling:** To construct training set and evaluate our method, we need ground truth of alignments between different social networks. Labeling such data is a time-consuming and labor intensive work. It requires human annotator to manually searching for the accounts belong to a certain person on different social networking service.

We leveraged crowdsourcing technique to ease the labeling work. An editable column has been provided on academic search service we developed for user to fill in the urls link to the external social network accounts of the correspond-

| Observed social network | # node | # edge |
|-------------------------|--------|--------|
| Linkedin                | 2985414| 25965384|
| ArnetMiner              | 1053188| 3916907 |
| Videolecture            | 11178  | 786353  |

**Table 2: Data statistics**

| Labeled Pairs | \(|L(A,*)|\) | \(|L(L,*)|\) | \(|L(V,*)|\) |
|---------------|-------|-------|-------|
| ArnetMiner    | -     | 7532  | 1470  |
| Linkedin      | 7532  | -     | 410   |
| Videolecture  | 1470  | 410   | -     |

**Table 3: Statistics of labeled pairs between different social networks, where \(|L(*,*)\)| denotes the number of labeled corresponding pairs between the given two networks.**

3.2 Analysis

As the first step, we engage in some high-level investigation of the generality and particularity of different social networks. Hence, we want to answer two questions:

- What reveals our social identity?
- What is consistent across different social networks?

3.2.1 What reveals our social identity?

**Name:** Username is the identifier of a user on social media sites. We make the assumption that users tend to use similar user name in different social networks. Such an intuition becomes more obvious since we are dealing with real name social networks in this work. However, even if in real name social networks, name wouldn’t necessarily gives away user’s identity since: 1) A name can be shared by different person. Figure 2 shows the statistics of account sharing an individual name in each social network, it is not surprising to see a few names are shared by more than 100 accounts. While for Videolectures network, there is not as much name duplication since the data scale is smaller. 2) A name can be represented in different ways, as in ArnetMiner, many authors’ first name only contains the initial letter . One intuition is that rare used names are likely owned by a single real person. For example, given two accounts both named “Wei Wang”, we wouldn’t have much confidence to claim they belong to the same user since there are so many people using this name. On the contrary, if two accounts share the name ”Francis Scott Key Fitzgerald”, it is safe for us to make that assumption. So in a word, we can’t simply integrate accounts with the same name, but at least it gives us some clue.

**Profile:** Profile information is the description of a person given by itself, which is very informative. Unfortunately, it’s not easy to directly use due to the following two reasons. Firstly, profile ontology on different social networks are heterogeneous. Figure 3 illustrates the profile structure of ArnetMiner and Linkedin. Both ArnetMiner and Linkedin have name, education and locality, but the rest are different. However, in most of the cases, group and skills in Linkedin profile could be mapped to research interests in ArnetMiner; and summary has similar contents in Bio. In addition, homepage in Linkedin is sometimes contained in ArnetMiner’s contact, and the same applies to industry (in

\(^3\)http://press.linkedin.com/about

\(^6\)http://videolectures.net/site/about/
Figure 2: Number of person sharing an individual username on each social network

Figure 3: Heterogeneous profile ontology of ArnetMiner (left) and Linkedin (right).

Figure 4: Users’ social status in Linkedin network and ArnetMiner network

Linkedin) and work (in ArnetMiner). Hence, information is unstructured and distributed. As for Videolectures, we only have name, affiliation and homepage. As a result, the heterogeneous nature of different social networks makes no guarantee of similar profile structure or structured information, which poses a big challenge to our algorithm design.

Secondly, user provided profiles are noisy and incomplete. According to our statistics, more than half of the Linkedin users do not have summary, interests, specialties, etc, such a problem is even more serious on ArnetMiner network. In user profiles, some fields can uniquely identify a person, such as Email, homepage url. These information are not commonly available but once we discover two accounts share such kind of attribute, we can safely conclude that they correspond to the same person.

Friends: “Friends make us who we are” [?]. It is possible that the social contacts we created within the social networks expose our identity. We have already mentioned the existence of consistent social ties that co-occurs on different social network. What we want to address here is that different friend reveals our identity to varying degree. Which is quite straightforward since when someone introduced himself as a fan of lady gaga, it almost gives away nothing about his social identity while if he mentioned he is a friend of someone not so famous, at least his identification has been narrowed down to a finite set of person. Further, given two accounts $v_{k1}^i$ and $v_{k2}^j$, if we already known they share a set of friends, can we claim they belong to the same person? An obvious intuition is that the more shared friends they have, the more likely the accounts belongs to one person, but it’s not the only factor. Imagine if a person is connecting with a strong connected component, according to social balance theory [], the probability of existing another person connecting with that strong connected component is clearly larger than connecting a bunch of person scattered in different communities. Such a intuition can be further confirm by sub-graph frequency of networks[?].

3.2.2 The consistency across different social networks

Social Status: In sociology, social status is the position or rank of a person or group within a social system. Social status indicates the influential ability and the role a user plays in the social network. As an important social factor on every social network, we want to see whether the social status of an individual person is consistent on different social systems. For simplicity, we use degree as the measurement of social status of a user, which suggests that the person with more friends has the higher social status. We ranked the labeled pairs between ArnetMiner network and Linkdin network by it’s degree on the two network respectively and plotted on a two dimensional space. In Figure 4, X-axis indicates the rank on ArnetMiner and Y-axis for Linkedin, the color indicates the density of each region.

From Figure 4 we can see that data points tend to appear near the diagonal which suggests to some degree, the social status can be considered as consistent. In addition, we can clearly see the left bottom corner is the densest region which shows the social status of opinion leaders are more consistent than others. This observation is very intuitive since building reputation and becoming an opinion leader on a social network is an expansive process while being an opinion leader on a social network shows the users have enough ability and social capital, therefore he/she is more likely to also be a opinion leader on a different social network.

Social Ties: The semantic of relationships or social ties within different social networks are interpreted differently. As we have addressed in 3.1, the edge in ArnetMiner network represents co-authorship while it is interpreted as business
contact or coworker relationship in Linkedin, and in Videolecutes network, it represents two person co-attending a venue. Although the meaning of them are different, intuitively we believe that there will be co-occurrence of social ties across different social networks. The explanation is straight forward, the social tie within a network implies two persons already know each other, so they are more likely to create a relationship in another social network than two random persons. We investigate the consistency of social tie on the labeled corresponding pairs between ArnetMiner network and Linkedin network. Figure 5 shows the number of co-occurred social tie between these users. Here, we say a social tie is co-occurred if it exists on both social networks. There are 627 co-occurred social ties, while 1670 appear on Linkedin but absent on ArnetMiner and 7773 conversely. Note that we only partially observed the Linkedin network due to the privacy protection, hence, we can safely assume that there will be much more co-occurred social tie if the complete network is available.

4. NETWORK INTEGRATION PIPELINE

Base on the observations above, we proposed a unified framework to integrate multiple observed social networks in order to recover the underlying structure of the global social graph. Figure 7 shows overview workflow of our social network integration framework. The profile accounts crawled from different social networks will be first converted to candidate pairs. Then, the candidate pairs will be feed into a probabilistic model in order to determine the correctness of each candidate pair.

4.1 Candidate Pairs Generation

Finding pairs of accounts on two social networks \( G_i(V_i, E_i) \) and \( G_j(V_j, E_j) \) that belong to the same natural person, the scale of search space is \( O(|V_i| \cdot |V_j|) \), and even larger for integrating more than two social networks. Such a task is intractable for real world networks with millions of nodes. It is obvious that only very few pairs of accounts are actually belong to the same person, hence we need to quickly and roughly filter out those pairs that are unlikely to be true. Therefore, we use a candidate pairs generation step to reduce the search space. A desired candidate generation method should have following characteristics:

- Easy to implement
- Scalable to very large networks.
- Generate small candidate sets meanwhile covers most of the corresponding pairs.

As in , we make the assumption that user tend to use similar user name in different social networks. Such an intuition becomes more obvious since we are dealing with real name social networks in this work. Here we take this heuristic to generate candidate pairs that are potentially belong to an individual natural person.

To tolerate miss spell and different representation of names, we decompose each name into segments and group the accounts sharing at least one segment in the name. To further shrink the size of candidate sets, we compute jaro-winkler distance between the name of accounts within a group. Finally, an appropriate threshold is chosen to trade off between candidate size and coverage rate. Section 6.1 presents empirical analysis on different thresholds.

As we can see, our method is easy to implement; scalable to large social networks; and as showing in Section 6.1, it yields high coverage rate of corresponding pairs. In addition, this method fits online streaming data well. Candidate sets could be effectively updated incrementally with the incoming data.

4.2 Learning To Align Corresponding Pairs

With the candidate pairs generated at last step, we need to further determine which of them are truly corresponding pairs, which is a binary classification task to assign the true or false label for each candidate pair.

We formulate the classification problem into a probabilistic graphical model in order to capture the structural correlations and the uncertainty in predictions. Each candidate pair \( c_k \in \cup C_{i,j} \) is mapped to an observed variable \( x_k \). \( Y = \{y_k\}_{k=1}^{|\cup C_{i,j}|} \) are hidden variables that represents the true or false labels of \( X = \{x_k\}_{k=1}^{|\cup C_{i,j}|} \). To better incorporate the observations discussed in Section 3.3, we use a factor graph to represent the probabilistic model. We defined two types of feature functions:

- Local feature function: \( f(y_k, x_k) \) captures the characteristics between the two accounts \( v_i \) and \( v_j \) associated with the latent variable \( y_k \).
- Correlation feature function: \( g(y_k, y_{k'}, \mathcal{G}(y_k, y_{k'})) \) denotes the correlation between two latent variables \( y_k \) and \( y_{k'} \).
- Group constraint function: \( h(y_k, \mathcal{H}(y_k, C_k)) \) represents
the correlation between a latent variable \( y_k \) and a constrained group \( C_k \).

- Triangle feature function: \( q(y_k, y_k', y_k'') \) denotes the correlation within the three latent variables \( y_k, y_k' \) and \( y_k'' \) form a triangle closure.

Figure 6 gives a graphical representation of the proposed model. Given the observed social networks \( G \), candidate pair set \( C = \cup C_{i,j} \), we can define the joint distribution over \( Y \) as:

\[
p(Y|G, C) = \prod_k f(x_k, y_k)g(y_k, G, y_{k'})
\]

\[
b(y_k, H(y_k, C_k))q(y_k, Q(y_k, y_{k'}, y_{k''}))
\]

### 4.3 Feature Design

#### Modeling the Uniqueness of Name:

As we mentioned in 3.3.1, a unique name is likely to be owned by a single natural user. To measure the uniqueness of a name, we use a language model based approach. Given a name \( N = (s_1, s_2, \ldots, s_n) \), where \( s_k \) is the segment obtained at candidate generation step. We define uniqueness of a name as:

\[
U(N) = \sum_{k=1}^{n} \log P(s_k)
\]

For each candidate pair \( c_{i,j} = (v_{i}^{k_1}, v_{j}^{k_2}) \), a feature \( U(c_{i,j}) = (U(N(v_{i}^{k_1}))+U(N(v_{j}^{k_2}))) / 2 \) is assigned to capture whether the two accounts are sharing a unique name.

#### Modeling Profile Similarity:

Hence, we categorized user profile fields into Strong Identifier and Weak Identifier. We calculate the edit distance between each shared strong identifier for a given candidate pairs as a set of features. For weak identifiers, we concatenate them into a single document and convert each document into a bag-of-words vector. For each vector, the words are weighted by TF-IDF. The similarity between the profiles of each candidate pairs are measured by inner product and cosine distance.

#### Modeling Social Status Shift:

Based on the observation in Section 3.3.1, the pattern of social status shifting across different social networks can give us discriminative power to determine true or false label of candidate pairs. To incorporate this observation into our predictive model, we first determine the social status of each account within its social network, again, we use degree to measure social status. We sorted accounts by degree in each social networks and we call the top 1% accounts “opinion leaders”, the following 10% “middle class”, and for the rests, we call them “the messes”. Given a candidate pair \( c_{i,j} = (v_{i}^{k_1}, v_{j}^{k_2}) \), where \( v_{i}^{k_1} \) is an opinion leader in \( G_i \) and \( v_{j}^{k_2} \) is a middle class in \( G_j \), a feature from\( \text{In} G_i, \text{To} \text{Min} G_j \) will be assigned to it.

The features defined above capture the characteristics between within a candidate pair, hence, we incorporate them as local features, which is defined as an exponential function:

\[
f(y_k, x_k) = \frac{1}{Z_{\alpha}} \sum_{\alpha} \Phi_{\alpha}(y_k, x_k)
\]

Where \( \Phi_{\alpha}(.) \) is the \( d^\text{th} \) local feature defined on \( x_k \) with respect to the value of \( y_k \); \( \alpha \) is the weight of the feature; \( Z_{\alpha} \) is a normalization factor.

#### Modeling Co-occurred Social Ties:

According to Section 3.3.1, a social tie may co-occur on different social networks if both of the two associated users have accounts on those networks. However such a strong signal is not straightforward to capture since we do not know the correspondence between social ties before identifying the users associated with them. Here, we model the co-occurred social ties as the structural correlation between candidate pairs. To make it more intuitive, Figure 7 gives a toy example illustrating this process. On the left side are two networks \( G_1 \) and \( G_2 \). On the right side, each node represents a candidate pair, and each edge represents a potential co-occurred social tie. In the example, we assume any two node from the two network is a candidate pair, hence there are \( 3 \times 3 = 9 \) nodes on the right. We construct the potential co-occurred social ties set \( \text{CST} \) with the following process. For each two candidate pairs \( c^h = (v_{i}^{k_1}, v_{j}^{k_2}) \) and \( c^k = (v_{i}^{k_1}, v_{j}^{k_2}) \), we draw an edge between them if there is an edge \( e_{i}^{k_1,k_1'} = (v_{i}^{k_1}, v_{i}^{k_1'}) \) in \( G_i \) and an edge \( e_{j}^{k_2,k_2'} = (v_{j}^{k_2}, v_{j}^{k_2'}) \) in \( G_j \), respectively, we say \( e_{i}^{k_1,k_1'} \) and \( e_{j}^{k_2,k_2'} \) are potential co-occurred social ties between \( G_i \) and \( G_j \) and there is correlation between \( c^h \) and \( c^k \).

Co-occurred social tie is formulated as a correlation feature which is defined as:

\[
g(y_k, y_{k'}, G(y_k, y_{k'})) = \frac{1}{Z_{\beta}} e^{g(y_k, y_{k'})}
\]

\((y_k, y_{k'})\) is an indicator to specify whether there is a potential co-occurred social tie between two candidate pairs, thus, \( g(y_k, y_{k'}) = 1 \) if there is, and 0 otherwise.

#### Modeling One-to-one Mapping Constraint:

Users may create alias accounts on social networks, while in most cases users will stick to a single account since managing one account is hard enough. Hence we set the constraint that...
one-to-one mapping is encouraged between the two networks. Formally, when integrating network $G_i$ and $G_j$, we say a set of candidate pairs are constrained by $v^k_i$ if all of these candidate pairs contains node $v^k_i$ since only one of them is likely to be true, and we call it a constrained group $C_k$. Figure 9 shows the one-to-one mapping constraint when integrating networks $G_1$ and $G_2$. As an example, the three candidate pairs $(v_1^2, v_2^2), (v_1^2, v_1^3)$ and $(v_1^3, v_2^3)$ connecting by red edges are constrained by $v_1^2$ since all of them are associated with $v_1^2$. We use $MC = \{C_k\}$ to denote all the one-to-one mapping constrained groups.

One-to-one mapping constraint feature is defined as

$$h(y_k, H(y_k, C_i)) = \frac{1}{Z_y}e^{\gamma h(y_k, C_i)} \quad (4)$$

where $C_i \in MC$ is a constrained group that $y_k$ in. $h(y_k, C_i)$ is set to 0 if more than two of variables in $C_i$ labeled true, since it violates the one-to-one mapping constraint, and 1 otherwise.

Modeling Logical Transitivity: Since the assertion of account A and account B belong to the same user is symmetrical, there is logical transitivity between related mapping. Figure 9 shows the logical transitivity when integrating three social networks $G_1, G_2$ and $G_3$, where each transitivity closure is indicated by an individual color. For instance, the three candidates pairs $(v_1^2, v_2^2), (v_1^2, v_1^3)$ and $(v_2^3, v_1^3)$ forms a transitivity closure, and constrained by the fact that if two of them are labeled true, the other one must be true. Hence when we dealing with more than two social networks, we can to incorporate this kind of constraint to further improve our method. We use $LT$ to represent all such transitivity closures.

Logical transitivity feature is defined as a triangle feature:

$$q(y_k, y_{k'}, y_{k''}, Q(y_k, y_{k'}, y_{k''})) = \frac{1}{Z_q}e^{\eta q(y_k, y_{k'}, y_{k''})} \quad (5)$$

where $y_k, y_{k'}, y_{k''}$ are nodes forming a logical transitivity closure. $q(y_k, y_{k'}, y_{k''})$ would be set to 0 if two of them are labeled true and the other one is labeled false since it violates the logical transitivity constraint, otherwise $q(y_k, y_{k'}, y_{k''})$ is set to 1. As can be seen from Figure 9, logical transitivity connects nodes from different network pairs, thus capturing the correlations across multiple social networks.

4.3.1 Learning

Model learning is to estimate the best parameter configuration $\theta = (\lambda, \beta, \gamma, \eta)$ that maximizes the log-likelihood objective function. According to Hammersley-Clifford theorem, we can obtain the following log-likelihood function.

$$O(\theta) = log P_{\theta}(Y|G, C)$$

$$= \sum_{k=1}^{[\cup C_{k,j}]} \sum_d \alpha_d \Phi(y_k, x_k)$$

$$+ \sum_{(y_k, y_{k'}) \in CST} \beta g(y_k, y_{k'})$$

$$+ \sum_{C_i \in MC} \sum_{y_k \in C_i} \gamma h(y_k, C_i)$$

$$+ \sum_{(y_k, y_{k'}, y_{k''}) \in LT} \eta q(y_k, y_{k'}, y_{k''}) - logZ$$

where $Z = Z_y Z_{\beta} Z_\gamma Z_\eta$ is a normalization factor.

For representation simplicity, we concatenate all feature functions for variable $y_k$ as

$$\Psi(y_k) = \{\Phi(y_k, x_k)^T, g(y_k, y_{k'})^T, h(y_k, C_i)^T, q(y_k, y_{k'}, y_{k''})\}$$

We adopt gradient decent method to maximize the objective function. The gradient can be calculated as following

$$\frac{\partial O(\theta)}{\partial \theta} = E[\Psi(y_k)] - E_{\theta}(Y|G, C)[\Psi(y_k)]$$

(8)

The term $E[\Psi(y_k)]$ in Eq.8 is easy to calculate, but the second term is intractable to directly calculate. In this way, we use Loop Belief Propagation to approximate the marginal probability $p(y_k|\theta)$. Hence the gradient is obtained by summing over all nodes.

4.3.2 Inference

Based on the learned parameters $\theta$, unknown labels can be predicted by finding the optimal configuration of $Y$ that maximizes the joint probability in Equation (3), that is,

$$Y^* = argmax_Y p(Y|G, C)$$

(9)

Utilizing marginal probabilities $p_i|Y, G$ estimated by LBP, whether a node is an overlapping user pair is labeled according to whether it has the largest marginal probability of being labeled as 1 among all the nodes constrained by subgraph constraint factor. The higher the probability is, the more confidence we have in its correctness. For inference, we use the max-sum algorithm (the max version of Eqs. 8
and 9) to find the values of Y that maximizes the likelihood. This max-sum algorithm is similar with the sum-product algorithm, except for the message passing functions, which calculate the message according to max instead of sum.

4.4 Constructing the Social Graph

Adaptive Data Crawling Strategy Due to the highly dynamic characteristic and the huge data scale, it is almost impossible for us to obtain the entire social network. In our framework, the data are feeding in incrementally by a set of ad-hoc social network crawlers. In traditional social network crawling task, an often used technique is breath-first search which iteratively obtains data of nodes and their neighbors. However such a straightforward crawling strategy is not suitable for social network integration task since it requires overlap of different social networks. Moreover, as we discussed above, friends reveals our identities, it is much easier to determine whether to align a pair of accounts when most of its neighbors has already been aligned. Thus, we prefer to obtain the accounts close to the aligned pairs first.

In the framework, we store the accounts waiting to be crawled in a priority queue. After each of time of alignment phase, we iterate all the aligned pairs, for each pair \( c_{i,j} = (v^i_k, v^j_k) \), we give one “credit” score to each of their neighbors \( NB_i(v^i_k) \) and \( NB_j(v^j_k) \). The priority is determined by the received credit of each account. Hence, we can avoid low overlap rate between networks giving return by blind BFS crawling.

Global Social Graph Construction With the alignment output by the probabilistic model, we can finally construct the global social graph. Follow the definition given by Section 2 that global social graph \( G_0 = (V_0, E_0) \) is a multi-graph where each node \( v^k \in V_0 \) corresponding to a set of nodes on different networks belonging to an individual natural person. We first group associated aligned pairs into a set, where relation is defined as two candidate pairs sharing a node(eg. \((v^1_k, v^2_k)\) and \((v^1_j, v^3_j)\)). For each set, we create a node \( v^k \) in \( V_0 \) where \( v^k \) corresponds to all the nodes within the set. We iteratively incorporate edge for the nodes from each social network and create a new node for any node don’t have correspondence in \( G_0 \). Base on the constructed global social graph, we developed an integrated entity search application which users can search across different social networks. We won’t discuss this part of the work in this paper.

5. EXPERIMENTAL RESULTS

In this section, we first introduce the data sets for experiments, and then present experimental results as well as empirical analysis.

5.1 Experiment Setup

Candidate Pairs Generation We perform candidate pairs generation using the method explained in Section 4. Recall in Section 4 we mentioned the preferred candidate generation method returns candidate pair set with relatively reasonable size meanwhile covers as much corresponding pairs as possible. Hence, to select an appropriate threshold, we need to make a trade off between these two factors. We evaluate the quality of candidate generation on the labeled data. Figure shows the candidate coverage rate and the size of the generated candidate pair set with respected to threshold between each pair of networks. In Figure we can see that the size of candidate size in growing exponentially with the decreasing of threshold. We choose a threshold 0.8 which yields high coverage rate(higher than 98% between ArnetMiner & Videolectures and Linkedin & Videolectures and close to 95% between Arnet Miner & Linkedin) and reasonable candidate set size.

Comparison Methods We compare the following methods with our probabilistic model at corresponding pair alignment phase.

- Strict Name Matching (SNM). This method only uses names to estimate the correctness of a given candidate pair. For each candidate pair, it predicts positive if and only if the two associated accounts have the exact same name.
- Unique Name Matching (UNM). According to the discussion in Section 4.2.1, this method only returns positive for the candidate pairs with the exact same name \( \mathcal{N} \) meanwhile the uniqueness of the name \( U(\mathcal{N}) > 20 \).
- Logistic Regression (LR). This method incorporates all the local features discussed in section 4.2 to train a classification model and the employs the classification model to predict the correctness of candidate pairs.
- Conditional Random Field (CRF). It trains a conditional random field [?] with the local features associated with each candidate pair and structural correlations between candidate pairs, where correlations are indicated by the potential persistent social across different social network introduced in Section 4.2.
- Our Method. The proposed approach in the paper, which leverages the local features as well as the persistence of social tie, one-to-one mapping constrain and logical transitivity constrain to make the prediction.

Evaluation Measure We evaluate the performance of each method in terms of precision, recall and F1-score.

5.2 Performance Analysis

We compare the performance of the five method for corresponding pairs alignment task. Note that our model is trained over all three social networks ArnetMiner, Linkedin and Videolectures, and making prediction jointly. But for clarity, we present the experimental result for each pair of social networks both separately and combined.

Table 4 shows the empirical results for integrating three social networks: ArnetMiner, Linkedin and Videolectures. According to Table 4, our method significantly outperforms the baselines. The two rule-based baseline methods SNM and UNM only use username to make the prediction. SNM strictly aligns all the candidate pairs with the exact same name. For a commonly used name, it will predict all the candidate pairs sharing that name positive hence yields low precision. UNM considers the uniqueness of name and only considers those candidate pairs sharing a rarely used name as they belong to the same person, which gives a high precision, while recall suffers. However, as a result, the performance of the two simple rule-based method confirms that unique name is a strong indicator for corresponding pairs.

The improvement between CRF and LR verifies the effectiveness of persistent social ties for corresponding pairs alignment and further confirms the intuition that friends reveals our social identity.
Figure 10: Coverage rate and size of the generated candidate pair set w.r.t. threshold

| Data Set         | Method | Prec. | Rec. | F1-score |
|------------------|--------|-------|------|----------|
| ArnetMiner & Linkedin | SNM    | 0.5950 | 0.6994 | 0.6430   |
|                  | UNM    | 0.9081 | 0.5141 | 0.6565   |
|                  | LR     | 0.6765 | 0.8719 | 0.7618   |
|                  | CRF    | 0.8484 | 0.8315 | 0.8398   |
|                  | Our Method | 0.8602 | 0.8674 | 0.8638   |
| ArnetMiner & Videolectures | SNM    | 0.4656 | 0.8410 | 0.5994   |
|                  | UNM    | 0.9192 | 0.6034 | 0.7286   |
|                  | LR     | 0.4063 | 0.8692 | 0.5537   |
|                  | CRF    | 0.6919 | 0.8710 | 0.7710   |
|                  | Our Method | 0.7076 | 0.8946 | 0.7901   |
| Linkedin & Videolectures | SNM    | 0.1175 | 0.8222 | 0.2655   |
|                  | UNM    | 0.3045 | 0.5889 | 0.4015   |
|                  | LR     | 0.1701 | 0.7333 | 0.2762   |
|                  | CRF    | 0.3748 | 0.7990 | 0.5102   |
|                  | Our Method | 0.5012 | 0.8667 | 0.6531   |
| All together     | SNM    | 0.4859 | 0.7522 | 0.5904   |
|                  | UNM    | 0.8641 | 0.5473 | 0.6702   |
|                  | LR     | 0.5209 | 0.8673 | 0.6588   |
|                  | CRF    | 0.6449 | 0.8041 | 0.7158   |
|                  | Our Method | 0.7096 | 0.8762 | 0.7841   |

Table 4: Performance comparison of different methods for corresponding pair alignment task. The result are presenting combined and separately for the three network pairs.

Finally, due to the extremely imbalance (# negative pairs / # positive pairs is approximately $10^3 : 1$) and lacking of training instance, all the baseline methods yield low precision for the alignment between Linkedin & Videolectures. By encountering logical transitivity between corresponding pairs, we can leverage the information from the other two network pairs to improve the performance, thus, our method significantly outperforms the baseline methods for aligning Linkedin & Videolectures.

6. RELATED WORKS

Due to the wide potential applications, user identification across multiple social networks has attracted more and more attention recently. Zafarani et al. [25] first addressed the similar problem and proposed a search based attracting. Zafarani et al. [25] takes a behavioral modeling approach models the behavioral pattern of users when selecting username, hence to further determine whether two usernames belong to the same person. Iofciu et al. [27] investigates

the user linking problem in within a social tagging system. Liu et al. [28] leverages rare username to conduct training instance for user linking task. Zhang et al. [29] formulates the problem as a anchor link prediction task and leverages temporal and spatio information to make the prediction.

From a different angle, network integration could lead to potential leakage of privacy. A related field is called social network de-anonymization [30, 31]. These works mainly focused on recovering users’ identity from a masked social network. In [2][32], Backstrom et al. present such process where one can identify individuals in these anonymized networks by either manipulating networks before they are nononymized or by having prior knowledge about certain anonymized nodes. Narayanan et al. [33] attempt to de-anonymizing two social networks based only on the network topology.

Network alignment has been studied in many works, which preforms matching between two undirected graphs. Its task is to maximize the number of “overlapped” edges given two undirected graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and a set $L$ containing the feasible matching between $V_1$ and $V_2$. Due to the NP-hard nature of this definition, many different approaches are adopted to relax this constraint, such as Linear Programming[34], Belief Propagation[35] and so on. There also have been some successful applications under this framework, for instance, finding common pathways in biological networks [36, 37] and ontology alignment between Citeseer papers and DBLP papers[38].

Entity disambiguation or entity resolution is also a widely studied problem. Many existing works use Wikipedia for named entities as well as a source for entity disambiguation. Bunescu et al.[39] exploited the Wikipedia (entity pages, redirection pages, categories, and hyperlinks) and built a context-article cosine similarity model. Cucerzan et al. [40] presented a large scale named entity disambiguation system that also employed a huge amount of information extracted from Wikipedia. Other works use DBpedia to construct a graph of linked data entities as a way of disambiguating named entities. Instead of considering only ontology and taxonomy[41, 42], there are attempts to employ all the information from knowledge bases [43] and so on. These works adapted topical models for performing entity disambiguation that considered the context of every word as well as co-occurrence patterns among entities.

Our work is also closely related to relationship mining in social network analysis. One research topic is to predict unknown link in social networks. Liben-Nowell et al. [44] study the unsupervised methods for link prediction. Backstrom et al.[45] proposed a supervised random walk algorithm to estimate the strength of social link. Leskovec et al. [46] em-
ployed a logistic regression model to predict positive and negative links in online social networks. Wenbin et al.[?] propose a Partially-labeled Pairwise Factor Graph Model for learning to infer the type of social ties.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of social network integration. We engaged an in-depth analysis across three social networks, Linkedin, Videolectures and ArnetMiner and addressed what reveals users’ social identity, whether the social factors consistent across different social networks and how we can leverage this information to perform integration. Base on the observations, we proposed a unified framework for the social network integration task. It crawls data from multiple social networks and further discovers accounts corresponding to the same real person from the obtained networks. We use a probabilistic model to determine such correspondence, it incorporates features like the consistency of social status and social ties across different, as well as one-to-one mapping constraint and logical transitivity to jointly make the prediction. The empirical study shows that our framework can effectively integrate the three social networks which confirms the feasibility of the problem.

The future work falls into two parts. 1) In this work, we are dealing with real name social network, hence we can take the advantages of using name similarity to perform candidate generation in order to shrink the search space. In anonymous social networks, such a process wouldn’t be so straightforward, though we can still use username as a start point[? , ?]. 2) With the integrated global social graph, we want to further investigate several interesting topics such as information diffusion and user behavior pattern across different social networks that only feasible on the global social graph.
Social Network Integration: Towards Constructing the Social Graph

ABSTRACT

In this work, we formulate the problem of social network integration. It takes multiple observed social networks as input and returns an integrated global social graph where each node corresponds to a real person. The key challenge for social network integration is to discover the correspondences or interlinks across different social networks.

We engaged an in-depth analysis across six online social networks, twitter, livejournal, flickr, lastfm and myspace in order to address what reveals users’ social identity, whether the social factors consistent across different social networks and how we can leverage these information to perform integration.

We proposed a unified framework for the social network integration task. It crawls data from multiple social networks and further discovers accounts correspond to the same real person from the obtained networks. We use a probabilistic model to determine such correspondence, it incorporates features like the consistency of social status and social ties across different, as well as one-to-one mapping constraint and logical transitivity to jointly make the prediction. Empirical experiments verify the effectiveness of our method.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Text Mining; H.2.8 [Database Management]: Database Applications; H.4.m [Information Systems]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

Social network, Predictive model, Social influence

1. INTRODUCTION

Online social media capture the digital trace of our social life. The diverse functionality of different on-line social media services make us join multiple social networks. We stay in touch with our friends on facebook, follow celebrities on microbloging sites and manage business contacts through linkedin. As a result, our social life has been broken into pieces due to the absent of the interlink between each social network. Is it possible to connect these pieces together, in order to recover the full image of our social life?

The advantage of such an process is multi-folded. For an individual user, managing multiple social network accounts takes a lot of effort. Bridging each isolated social circles of a user could lighten the communication cost and reduce information redundancy. From a service provider’s perspective, aggregating information from different social networks gives a more comprehensive view of each individual user and filters out the potential bias, hence it would benefit many applications. Taking recommendation system as an example, it would be a straightforward solution to cold start problem by leveraging information from the accounts on other
sites of a new user. Moreover the inter-connections between different social networks might reveal interesting yet unobservable phenomenons when we look at an individual social network independently, such as the information diffusion pattern across different social media. However, this could also lead to potential leakage of privacy as it exposes users' identity without permission.

In this work, we are trying to tackle the problem of social networks integration, where integration means combining the accounts correspond to the same real person across different social networks, in order to get a more complete view of that person’s social life. We want to investigate whether, how and to which extent we can fulfill this task.

Figure 1 gives a toy example illustrating the process of integrating four observed social networks $G_1$, $G_2$, $G_3$ and $G_4$ in order to recover the structure of the underlying global social graph $G_0$. The upper side of Figure 1 is the input of our problem: observed social networks. Nodes represent accounts and edges represent social ties within each individual social network. At the bottom shows the output of the problem: the global social graph integrates the four observed social networks. Users who have accounts on multiple social networks are colored with red. Such an intuitive process is non-trivial due to the following reasons. Firstly, lacking of unique identifier. There is no such an attribute that can uniquely identifies a person across different social media, even name won’t satisfy the condition since a single name can be shared by different person and also can be represented in different ways. Secondly, the heterogeneity of different social networks, where the heterogeneity reflected in both topology and semantics on different social networks. As a result, the features of an individual user are not necessarily be consistent across different social networks. The third challenge comes from data availability, because of the privacy policy and the large and the on growing data scale, we can hardly observe the entire network.

Considering the above intuitions, in the paper, we propose a social network integration framework which effectively integrates arbitrary number of social networks. In the framework, the data is obtained a set of ad-hoc crawlers each corresponds to an individual social networks. A probabilistic model is used to discover the accounts on each network that corresponds to the same real person, hence integrates the obtained social network. The result of integration will be further feedback to the crawlers to adjust crawling priority. In summary, our contributions in this work include:

- We formulate the problem of social networks integration, and discuss associated challenges and potential;
- We propose a unified social network integration framework that takes multiple social networks as input, and returns a integrated social graph.
- Empirical experiments on integrating three large scale social networks, Linkedin, Videolectures and ArnetMiner demonstrate the effectiveness of our framework.

The rest of the paper is organized as follows. Section 2 formally defines the problem of social network integration; Section 3 gives an in-depth analysis on the three social networks; Section 4 is devoted to our social network integration framework; Section 6 analyzes experimental results; Section 7 presents related works and Section 8 concludes this work and discuss future directions.

| $G$ | A set of social networks $G_1, G_2, \ldots , G_K$ |
| $G_0$ | The global social graph given $G$ |
| $v^i$ | A given node from $G_i$ |
| $C_{i,j}$ | The candidate set between $G_i$ and $G_j$ |
| $c^i_{i,j}$ | A given candidate pair in $C_{i,j}$ |
| $O_{i,j}$ | The alignment of $G_i$ and $G_j$ |
| $a^i_{i,j}$ | A given corresponding pair between $G_i$ and $G_j$ |

### Table 1: Symbols and Notations

## 2. PROBLEM DEFINITION

In this section, we give several necessary definitions and present the formulation of social network integration problem.

Given $k$ observed social network $G = \{G_1, G_2, \ldots, G_k\}$. Each social network is an unweighted undirected graph $G_i = (V_i, E_i)$, where $V_i = \{v^i_1, \ldots, v^i_{|V_i|}\}$ is a set of social network accounts and each edge $(v^i_{k_1}, v^i_{k_2}) \in E_i$ represents a social tie or social interaction between two accounts $v^i_{k_1}$ and $v^i_{k_2}$ within social network $G_i$. Social network integration problem can be divided into the following two sub-tasks:

**Probabilistic Alignment.** An alignment $O$ is a set of triplets $o_{i,j} = (v_i, v_j, p_k)$ each denotes an assertion that $v_i$ from network $G_i$ and $v_j$ from network $G_j$ potentially belongs to the same natural person, where $p_k$ is a real valued probability indicates the confidence. The aim of probabilistic alignment task is to discover such inter-network correspondence within the observed social networks $G$.

**Social Graph Construction.** With the alignment generated at the last step, we further constructing the social graph, where global social graph $G_0(V_0, E_0)$ is the underlying global network structure given a set of observed social network $G$. It is represented as a multi-graph, where each node $v^0_0 \in V_0$ is a set of node in different networks that corresponds to an individual natural person. An edge $e^0_{k_1,k_2}$ between $v^0_{k_1}$ and $v^0_{k_2}$ will be created if there is an edge between two nodes $v^i_{k_1} \in v^0_{k_1}$ and $v^i_{k_2} \in v^0_{k_2}$ in any social network $G_i$. Given social networks set $G = \{G_i\}_i$, our goal is to reconstruct the underlying social graph $G_0$ by integrating all the social networks in $G$. For any node $v^0_0$ in $G_0$, we know exactly the set of $\{v^i_{k_0} \in v^0_0\}$.

## 3. DATA AND OBSERVATIONS

### 3.1 Data Collections

As we addressed in Section 1, to perform network integration, one of the challenges comes from data availability. Obtaining data for social network integration is not an easy task as in most cases, we won’t be able to observe the complete networks due to the size and privacy policy. In this work, we collected data from three social networks: Linkedin\(^1\), Videolectures\(^2\) and ArnetMiner\(^3\).

**Linkedin’s “Also-View” network:** Linkedin operates the world’s largest professional network on the Internet. It

\(^1\)http://www.linkedin.com/

\(^2\)http://www.videolectures.org/

\(^3\)http://www.arnetminer.org/
Table 2: Data statistics

| Observed social network | # node  | # edge  |
|-------------------------|---------|---------|
| Linkedin                | 2985414 | 25965384 |
| ArnetMiner              | 1053188 | 3916907  |
| Videolecture            | 11178   | 786353   |

Table 3: Statistics of labeled pairs between different social networks, where \([L(A,*)]\) denotes the number of labeled corresponding pairs between the given two networks.

| Labeled Pairs | \([L(A,*)]\) | \([L(L,*)]\) | \([L(V,*)]\) |
|---------------|-------------|-------------|-------------|
| ArnetMiner    | -           | 7532        | 1470        |
| Linkedin      | 7532        | -           | 410         |
| Videolecture  | 1470        | 410         | -           |

has more than 238 million members in over 200 countries and territories\(^5\). On Linkedin network, part of user information is available for public profiles, such as education, affiliation and industry (refer to Figure 5). In addition, users could maintain a certain amount of connections, which could be friends, co-workers, etc. However, personal connections are viewed as private information and unavailable to public access. But we discovered through the “Viewers of this profile also viewed...” table at the right side of public profiles. Hence, we construct an undirected network based on the “also-view” links, which can be considered as a k-nearest neighbor network. We crawled approximately 3 million public profiles and more than 25 million “also-view” relationships.

Videolectures’ “Together with” network: Videolectures is an open access educational video lectures repository. The lectures are given by distinguished scholars and scientists at most important events like conference, summer schools and workshops\(^6\). Connections between 11777 profiles we crawled are not explicitly provided by Videolecture. However, researcher attending the same venue can view as a kind of social relationship or interaction. Hence we extracted 786353 “Together with” relationships out of all profiles. In Videolecture’s “together with” network, an undirected edge \(e = (v_i, v_j)\) would be created if \(v_i\) and \(v_j\) have at least co-attended a venue.

ArnetMiner's co-author network: Arnetminer aims to provide comprehensive search and mining services for researcher social networks. We obtained the entire ArnetMiner data up to year 2013, including user profiles and co-author relationships. The details of user profiles ontology are shown in Figure 5. To construct co-author network, we create a undirected edge \(e = (v_i, v_j)\) if author \(v_i\) and \(v_j\) have coauthored a publication. Finally, we obtain a co-author network with more than 1 million nodes and nearly 4 million edges.

The statistics of the three networks are illustrated in Table 2.

Crowdsourcing data labeling: To construct training set and evaluate our method, we need ground truth of alignments between different social networks. Labeling such data is a time-consuming and labor intensive work. It requires human annotator to manually searching for the accounts belong to a certain person on different social networking service.

We leveraged crowdsourcing technique to ease the labeling work. An editable column has been provided on academic search service we developed for user to fill in the urls link to the external social network accounts of the correspond-

\(^5\)http://press.linkedin.com/about
\(^6\)http://videolectures.net/site/about/
Friends: "Friends make us who we are" [7]. It is possible that the social contacts we created within the social networks expose our identity. We have already mentioned the existence of consistent social ties that co-occurs on different social network. What we want to address here is that different friend reveals our identity to varying degree. Which is quite straightforward since when someone introduced him-
Figure 5: Persistence of social tie between LinkedIn network and ArnetMiner network

contact or coworker relationship in LinkedIn, and in Videolectures network, it represents two person co-attending a venue. Although the meaning of them are different, intuitively we believe that there will be co-occurrence of social ties across different social networks. The explanation is straightforward, the social tie within a network implies two persons already know each other, so they are more likely to create a relationship in another social network than two random persons. We investigate the consistency of social tie on the labeled corresponding pairs between ArnetMiner network and LinkedIn network. Figure 5 shows the number of co-occurred social tie between these users. Here, we say a social tie is co-occurred if it exists on both social networks. There are 627 co-occurred social ties, while 1670 appear on LinkedIn but absent on ArnetMiner and 7773 conversely. Note that we only partially observed the LinkedIn network due to the privacy protection, hence, we can safely assume that there will be much more co-occurred social tie if the complete network is available.

4. NETWORK INTEGRATION PIPELINE

Base on the observations above, we proposed a unified framework to integrate multiple observed social networks in order to recover the underlying structure of the global social graph. Figure 7 shows overview workflow of our social network integration framework. The profile accounts crawled from different social networks will be first converted to candidate pairs. Then, the candidate pairs will be feed into a probabilistic model in order to determine the correctness of each candidate pair.

4.1 Candidate Pairs Generation

Finding pairs of accounts on two social networks \( G_i(V_i, E_i) \) and \( G_j(V_j, E_j) \) that belong to the same natural person, the scale of search space is \( O(|V_i| \cdot |V_j|) \), and even larger for integrating more than two social networks. Such a task is intractable for real world networks with millions of nodes. It is obvious that only very few pairs of accounts are actually belong to the same person, hence we need to quickly and roughly filter out those pairs that are unlikely to be true. Therefore, we use a candidate pairs generation step to reduce the search space. A desired candidate generation method should have following characteristics:

- Easy to implement
- Scalable to very large networks.
- Generate small candidate sets meanwhile covers most of the corresponding pairs.

As in , we make the assumption that user tend to use similar user name in different social networks. Such an intuition becomes more obvious since we are dealing with real name social networks in this work. Here we take this heuristic to generate candidate pairs that are potentially belong to an individual natural person.

To tolerate misspell and different representation of names, we decompose each name into segments and group the accounts sharing at least one segment in the name. To further shrink the size of candidate sets, we compute jaro-winkler distance between the name of accounts within a group. Finally, an appropriate threshold is chosen to trade off between candidate size and coverage rate. Section 6.1 presents empirical analysis on different thresholds.

As we can see, our method is easy to implement; scalable to large social networks; and as showing in Section 6.1, it yields high coverage rate of corresponding pairs. In addition, this method fits online streaming data well. Candidate sets could be effectively updated incrementally with the incoming data.

4.2 Learning To Align Corresponding Pairs

With the candidate pairs generated at last step, we need to further determine which of them are truly corresponding pairs, which is a binary classification task to assign the true or false label for each candidate pair.

We formulate the classification problem into a probabilistic graphical model in order to capture the structural correlations and the uncertainty in predictions. Each candidate pair \( c^k \in \bigcup C_{i,j} \) is mapped to an observed variable \( x_k \). \( Y = \{y_k\}_{k \in \bigcup C_{i,j}} \) are hidden variables that represents the true or false labels of \( X = \{x_k\}_{k \in \bigcup C_{i,j}} \). To better incorporate the observations discussed in Section 3.3, we use a factor graph to represent the probabilistic model. We defined two types of feature functions:

- Local feature function: \( f(y_k, x_k) \) captures the characteristics between the two accounts \( v_i \) and \( v_j \) associated with the latent variable \( y_k \).
- Correlation feature function: \( g(y_k, y_{k'}, G(y_k, y_{k'})) \) denotes the correlation between two latent variables \( y_k \) and \( y_{k'} \).
- Group constraint function: \( h(y_k, H(y_k, C_k)) \) represents...
the correlation between a latent variable $y_k$ and a constrained group $C_k$.

- Triangle feature function: $q(y_k, y_{k'}, y_{k''})$, denotes the correlation within the three latent variables $y_k$, $y_{k'}$ and $y_{k''}$ form a triangle closure.

Figure 6 gives a graphical representation of the proposed model. Given the observed social networks $G$, candidate pair set $C = \cup C_{i,j}$, we can define the joint distribution over $Y$ as:

$$p(Y|G, C) = \prod_k f(x_k, y_k)g(y_k, \mathcal{G}(y_k, y_{k'}))$$

$$h(y_k, \mathcal{H}(y_k, C_k))q(y_k, \mathcal{Q}(y_k, y_{k'}, y_{k''}))$$  \hspace{1cm} (1)

### 4.3 Feature Design

**Modeling the Uniqueness of Name:** As we mentioned in 3.3.1, a unique name is likely to be owned by a single natural user. To measure the uniqueness of a name, we use a language model based approach. Given a name $N = (s_1, s_2, \ldots, s_n)$, where $s_i$ is the segment obtained at candidate generation step. We define uniqueness of a name as $U(N) = \sum_{i=1}^n \log P(s_k)$. For each candidate pair $e_{i,j} = (v_{i,k}^1, v_{j,k}^2)$, a feature $U(e_{i,j}) = U(N(v_{i,k}^1)) + U(N(v_{j,k}^2)) / 2$ is assigned to capture whether the two accounts are sharing a unique name.

**Modeling Profile Similarity:** Hence, we categorized user profile fields into Strong Identifier and Weak Identifier. We calculate the edit distance between each shared strong identifier for a given candidate pairs as a set of features. For weak identifiers, we concatenate them into a single document and convert each document into a bag-of-words vector. For each vector, the words are weighted by TF-IDF. The similarity between the profiles of each candidate pairs are measured by inner product and cosine distance.

**Modeling Social Status Shift:** Based on the observation in Section 3.3.1, the pattern of social status shifting across different social networks can give us discriminative power to determine true or false label of candidate pairs. To incorporate this observation into our predictive model, we first determine the social status of each account within it’s social network, again, we use degree to measure social status. We sort accounts by degree in each social networks and we call the top 1% accounts "opinion leaders", the following 10% "middle class", and for the rests, we call them "the messes". Given a candidate pair $e_{i,j} = (v_{i,k}^1, v_{j,k}^2)$, where $v_{i,k}^1$ is an opinion leader in $G_i$ and $v_{j,k}^2$ is a middle class in $G_j$, a feature from $G_i$ to $G_j$ will be assigned to it.

The features defined above capture the characteristics between within a candidate pairs, hence, we incorporate them as local features, which is defined as an exponential function:

$$f(y_k, x_k) = \frac{1}{Z_{\alpha}} \sum_{d} \alpha_d \Phi_d(y_k, x_k)$$  \hspace{1cm} (2)

Where $\Phi_d(\cdot)$ is the $d^{th}$ local feature defined on $x_k$ with respect to the value of $y_k$; $\alpha_d$ is the weight of the feature; $Z_{\alpha}$ is a normalization factor.

**Modeling Co-occurred Social Ties:** According to Section 3.3.1, a social tie may co-occur on different social networks if both of the two associated users have accounts on those networks. However such a strong signal is not straightforward to capture since we do not know the correspondence between social ties before identifying the users associated with them. Here, we model the co-occurred social ties as the structural correlation between candidate pairs. To make it more intuitive, Figure 7 gives a toy example illustrating this process. On the left side are two networks $G_1$ and $G_2$. On the right side, each node represents a candidate pair, and each edge represents a potential co-occurred social tie. In the example, we assume any two node from the two network is a candidate pair, hence there are $3 \times 3 = 9$ nodes on the right. We construct the potential co-occurred social ties set $CST$ with the following process. For each two candidate pairs $e^k = (v_{i,k}^1, v_{j,k}^2)$ and $e^{k'} = (v_{i',k'}^1, v_{j',k'}^2)$, we draw an edge between them if there is an edge $e_{i,k}\rightarrow e_{i',k'}$ in $G_i$ and an edge $e_{j,k}\rightarrow e_{j',k'}$ in $G_j$, respectively, we say $e_{i,k}$ and $e_{i',k'}$ are potential co-occurred social ties between $G_i$ and $G_j$ and there is correlation between $e^k$ and $e^{k'}$.

![Figure 7: Modeling co-occurred social tie across social networks $G_1$ and $G_2$](image)

Co-occurred social tie is formulated as a correlation feature which is defined as:

$$g(y_k, y_{k'}, \mathcal{G}(y_k, y_{k'})) = \frac{1}{Z_{\beta}} e^{\beta g(y_k, y_{k'})}$$  \hspace{1cm} (3)

$(y_k, y_{k'})$ is an indicator to specify whether there is a potential co-occurred social tie between two candidate pairs, thus, $g(y_k, y_{k'}) = 1$ if there is, and 0 otherwise.

**Modeling One-to-one Mapping Constraint:** Users may create alias accounts on social networks, while in most cases users will stick to a single account since managing one account is hard enough. Hence we set the constraint that
one-to-one mapping is encouraged between two networks. Formally, when integrating network \( G_i \) and \( G_j \), we say a set of candidate pairs are constrained by \( v_i^k \) if all of these candidate pairs contains node \( v_i^k \) since only one of them is likely to be true, and we call it a constrained group \( C_k \). Figure 9 shows the one-to-one mapping constraint when integrating networks \( G_1 \) and \( G_2 \). As an example, the three candidate pairs \((v_1^2, v_2^2), (v_1^3, v_2^3)\) and \((v_1^3, v_2^2)\) connecting by red edges are constrained by \( v_i^2 \) since all of them are associated with \( v_i^2 \). We use \( MC = \{C_k\} \) to denote all the one-to-one mapping constrained group.

One-to-one mapping constraint feature is defined as

\[
h(y_k, H(y_k, C_i)) = \frac{1}{Z_i} e^{\gamma_h(y_k, C_i)}
\]

where \( C_i \in MC \) is a constrained group that \( y_k \) in \( h(y_k, C_i) \) is set to 0 if more than two of variables in \( C_i \) labeled true, since it violates the one-to-one mapping constraint, and 1 otherwise.

Modeling Logical Transitivity: Since the assertion of account A and account B belong to the same user is symmetrical, there is logical transitivity between related mapping. Figure 9 shows the logical transitivity when integrating three social networks \( G_1, G_2 \) and \( G_3 \), where each transitivity closure is indicated by an individual color. For instance, the three candidate pairs \((v_1^2, v_2^2), (v_1^3, v_2^3)\) and \((v_1^3, v_2^2)\) forms a transitivity closure, and constrained by the fact that if two of them are labeled true, the other one must be true. Hence when we dealing with more than two social networks, we can to incorporate this kind of constraint to further improve our method. We use \( LT \) to represent all such transitivity closures.

Logical transitivity feature is defined as a triangle feature:

\[
q(y_k, y_{k'}, y_{k''}, Q(y_k, y_{k'}, y_{k''})) = \frac{1}{Z_{\Phi}} e^{\gamma_q(y_k, y_{k'}, y_{k''})}
\]

where \( y_k, y_{k'}, y_{k''} \) are nodes forming a logical transitivity closure. \( q(y_k, y_{k'}, y_{k''}) \) would be set to 0 if two of them are labeled true and the other one is labeled false since it violates the logical transitivity constraint, otherwise \( q(y_k, y_{k'}, y_{k''}) \) is set to 1. As can be seen from Figure 9, logical transitivity connects nodes from different network pairs, thus capturing the correlations across multiple social networks.

### 4.3.1 Learning

Model learning is to estimate the best parameter configuration \( \theta = (\lambda, \beta, \gamma, \eta) \) that maximizes the log-likelihood objective function. According to Hammersley-Clifford theorem, we can obtain the following log-likelihood function.

\[
\mathcal{O}(\theta) = \log P_{\theta}(Y|G, C)
\]

\[
= \sum_{k=1}^{t} \left[ \sum_{d} \alpha_d \Phi(y_k, x_k) \right] + \sum_{(y_k, y_{k'}) \in \mathcal{G} \cup \mathcal{T} \cup \mathcal{C}} \beta \Phi(y_k, y_{k'})
\]

\[
+ \sum_{c_l \in \mathcal{MC}} \left[ \sum_{y_k \in C_l} \gamma_{h}(y_k, C_l) \right]
\]

\[
+ \sum_{(y_k, y_{k'}, y_{k''}) \in \mathcal{LT}} \eta q(y_k, y_{k'}, y_{k''}) - \log Z
\]

where \( Z = Z_1 Z_2 Z_3 Z_4 \) is a normalization factor.

For representation simplicity, we concatenate all feature functions for variable \( y_k \) as

\[
\Psi(y_k) = \Phi(y_k, x_k)^T, g(y_k, y_{k'})^T, h(y_k, C_l)^T, q(y_k, y_{k'}, y_{k''})
\]

We adopt gradient decent method to maximize the objective function. The gradient can be calculated as following

\[
\frac{\partial \mathcal{O}(\theta)}{\partial \Psi} = E[\Psi(y_k)] - E_{p_{\theta}(Y|G, C)}[\Psi(y_k)]
\]

The term \( E[\Psi(y_k)] \) in Eq.8 is easy to calculate, but the second term is intractable to directly calculate. In this work, we use Loop Belief Propagation to approximate the marginal probability \( p(y_k|\theta) \). Hence the gradient is obtained by summing over all nodes.

### 4.3.2 Inference

Based on the learned parameters \( \theta \), unknown labels can be predicted by finding the optimal configuration of \( Y \) that maximizes the joint probability in Equation (3), that is,

\[
Y^* = \arg \max_{Y} p_{\theta}(Y|G, C)
\]

Utilizing marginal probabilities \( p_{\theta}(Y|G, C) \) estimated by LBP, whether a node is an overlapping user pair is labeled according to whether it has the largest marginal probability of being labeled as 1 among all the nodes constrained by subgraph constraint factor. The higher the probability is, the more confidence we have in its correctness. For inference, we use the max-sum algorithm (the max version of Eqs. 8.

---

**ALGORITHM 1: Learning algorithm.**

**Input:** Observed social networks \( G \) with candidate pairs \( C \), and the learning rate \( \eta \).

**Output:** learned parameters \( \theta \).

\[
\theta \leftarrow 0;
\]

repeat

| foreach \( v_q \in Q \) and \( q \) do |
|-----------------------|
| //Initialization; |
| L ← initialization list; |
| Factor graph \( FG \leftarrow \text{BuildFactorGraph}(C) \); |
| // Learn the parameter \( \theta \) for factor graph model; |
| Calculate \( E_{\theta}(Y|G, C)|\Psi(y_k)| \); |
| foreach \( \theta_i \in \theta \) do |
| Calculate gradient \( \nabla \) according to Eq. 8; |
| Update \( \theta_i^{\text{new}} = \theta^{\text{old}} + \eta \cdot \nabla_i \); |
| end |
| end |

until converge;
and 9) to find the values of $Y$ that maximizes the likelihood. This max-sum algorithm is similar with the sum-product algorithm, except for the message passing functions, which calculate the message according to max instead of sum.

4.4 Constructing the Social Graph

Adaptive Data Crawling Strategy Due to the highly dynamic characteristic and the huge data scale, it is almost impossible for us to obtain the entire social network. In our framework, the data are feeding in incrementally by a set of ad-hoc social network crawlers. In traditional social network crawling task, an often used technique is breath-first search which iteratively obtains data of nodes and their neighbors. However such a straight forward crawling strategy is not suitable for social network integration task since it requires overlap of different social networks. Moreover, as we discussed above, friends reveals our identities, it is much easier to determine whether to align a pair of accounts when most of it’s neighbors has already been aligned. Thus, we prefer to obtain the accounts close to the aligned pairs first.

In the framework, we store the accounts waiting to be crawled in a priority queue. After each of time of alignment phase, we iterate all the aligned pairs, for each pair $\epsilon_{i,j} = (v_i^{k_1}, v_j^{k_2})$, we give one “credit” score to each of their neighbors $NB_i(v_i^{k_1})$ and $NB_j(v_j^{k_2})$. The priority is determined by the received credit of each account. Hence, we can avoid low overlap rate between networks giving return by blind BFS crawling.

Global Social Graph Construction With the alignment output by the probabilistic model, we can finally construct the global social graph. Follow the definition given by Section 2 that global social graph $G_0 = (V_0, E_0)$ is a multigraph where each node $v_i^{k} \in V_0$ corresponding to a set of nodes on different networks belonging to an individual natural person. We first group associated aligned pairs into a set, where relation is defined as two candidate pairs sharing a node(eg. $(v_1^1, v_2^1)$ and $(v_1^2, v_3^2)$). For each set, we create a node $v_0^k \in V_0$ where $v_0^k$ corresponds to all the nodes within the set. We iteratively incorporate edge for the nodes from each social network and create a new node for any node don’t have correspondence in $G_0$. Based on the constructed global social graph, we developed an integrated entity search application which users can search across different social networks. We won’t discuss this part of the work in this paper.

5. EXPERIMENTAL RESULTS

In this section, we first introduce the data sets for experiments, and then present experimental results as well as empirical analysis.

5.1 Experiment Setup

Candidate Pairs Generation We perform candidate pairs generation using the method explained in Section 4. Recall in Section 4 we mentioned the preferred candidate generation method returns candidate pair set with relatively reasonable size meanwhile covers as much corresponding pairs as possible. Hence, to select an appropriate threshold, we need to make a trade off between these two factor. We evaluate the quality of candidate generation on the labeled data. Figure shows the candidate coverage rate and the size of the generated candidate pair set with respected to threshold between each pair of networks. In Figure we can see that the size of candidate size in growing exponentially with the decreasing of threshold. We choose a threshold 0.8 which yields high coverage rate(higher than 98% between ArnetMiner & Videolectures and Linkedin & Videolectures and close to 95% between ArnetMiner & Linkedin) and reasonable candidate set size.

Comparison Methods We compare the following methods with our probabilistic model at corresponding pair alignment phase.

- Strict Name Matching (SNM). This method only uses names to estimate the correctness of a given candidate pair. For each candidate pair, it predicts positive if and only if the two associated accounts have the exact same name.
- Unique Name Matching (UNM). According to the discussion in Section 4.2.1, this method only returns positive for the candidate pairs with the exact same name $\mathcal{N}$ meanwhile the uniqueness of the name $\mathcal{U}(\mathcal{N}) > 20$.
- Logistic Regression(LR). This method incorporates all the local features discussed in section 4.2 to train a classification model and the employs the classification model to predict the correctness of candidate pairs.
- Conditional Random Field (CRF). It trains a conditional random field [?] with the local features associated with each candidate pair and structural correlations between candidate pairs, where correlations are indicated by the potential persistent social across different social network introduced in Section 4.2.
- Our Method. The proposed approach in the paper, which leverages the local features as well as the persistence of social tie, one-to-one mapping constrain and logical transitivity constrain to make the prediction.

Evaluation Measure We evaluate the performance of each method in terms of precision, recall and F1-score.

5.2 Performance Analysis

We compare the performance of the five method for corresponding pairs alignment task. Note that our model is trained over all three social networks ArnetMiner, Linkedin and Videolectures, and making prediction jointly. But for clarity, we present the experimental result for each pair of social networks both separately and combined.

Table 4 shows the empirical results for integrating three social networks: ArnetMiner, Linkedin and Videolectures. According to Table 4, our method significantly outperforms the baselines. The two rule-based baseline methods SNM and UNM only use username to make the prediction. SNM strictly aligns all the candidate pairs with the exact same username. For a commonly used name, it will predict all the candidate pairs sharing that name positive hence yields low precision. UNM considers the uniqueness of name and only considers those candidate pairs sharing a rarely used name as they belong to the same person, which gives a high precision, while recall suffers. However, as a result, the performance of the two simple rule-based method confirms that unique name is a strong indicator for corresponding pairs.

The improvement between CRF and LR verifies the effectiveness of persistent social ties for corresponding pairs alignment and further confirms the intuition that friends reveals our social identity.
6. RELATED WORKS

Due to the wide potential applications, user identification across multiple social networks has attracted more and more attention recently. Zafarani et al. [25] first addressed the similar problem and proposed a search based attracting. Zafarani et al. [25] takes a behavioral modeling approach models the behavioral pattern of users when selecting username, hence to further determine whether two usernames belong to the same person. Iofciu et al. [25] investigates the user linking problem in within a social tagging system. Liu et al. [25] leverages rare username to conduct training instance for user linking task. Zhang et al. [25] formulates the problem as a anchor link prediction task and leverages temporal and spatio information to make the prediction.

From a different angle, network integration could lead to potential leakage of privacy. A related field is called social network de-anonymization [25, 25]. These works mainly focused on recovering users’ identity from a masked social network. In [25, 25], Backstrom et al. present such process where one can identify individuals in these anonymized networks by either manipulating networks before they are nononymized or by having prior knowledge about certain anonymized nodes. Narayanan et al. [25] attempt to de-anonymizing two social networks based only on the network topology.

Network alignment has been studied in many works, which performs matching between two undirected graphs. Its task is to maximize the number of “overlapped” edges given two undirected graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ and a set $L$ containing the feasible matching between $V_1$ and $V_2$. Due to the NP-hard nature of this definition, many different approaches are adopted to relax this constraint, such as Linear Programing[25], Belief Propagation[25] and so on. There also have been some successful applications under this framework, for instance, finding common pathways in biological networks [25, 25] and ontology alignment between Citeseer papers and DBLP papers[25].

Entity disambiguation or entity resolution is also a widely studied problem. Many existing works use Wikipedia for named entities as well as a source for entity disambiguation. Bunescu et al.[25] exploited the Wikipedia (entity pages, redirection pages, categories, and hyperlinks) and built a context-article cosine similarity model. Cucerzan et al.[25] studied problem. Many existing works use Wikipedia for

Finally, due to the extremely imbalance (# negative pairs / # positive pairs is approximately $10^3$ : 1) and lacking of training instance, all the baseline methods yield low precision for the alignment between Linkedin & Videolectures. By encountering logical transitivity between corresponding pairs, we can leverage the information from the other two network pairs to improve the performance, thus, our method significantly outperforms the baseline methods for aligning Linkedin & Videolectures.

Table 4: Performance comparison of different methods for corresponding pair alignment task. The result are presenting combined and separately for the three network pairs.

| Data Set          | SNM | UNM | LR | CRF | Our Method       |
|-------------------|-----|-----|----|-----|------------------|
| ArnetMiner & Linkedin | 0.5950 | 0.6994 | 0.6430 | 0.8641 | 0.9192 |
| ArnetMiner & Videolectures | 0.4656 | 0.8410 | 0.5994 | 0.8719 | 0.7618 |
| Linkedin & Videolectures | 0.1175 | 0.8222 | 0.2655 | 0.8286 | 0.7901 |
| All together      | 0.4859 | 0.7522 | 0.5904 | 0.8461 | 0.6531 |

Figure 10: Coverage rate and size of the generated candidate pair set w.r.t threshold
ployed a logistic regression model to predict positive and negative links in online social networks. Wenbin et al.[?] propose a Partially-labeled Pairwise Factor Graph Model for learning to infer the type of social ties.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we study the problem of social network integration. We engaged an in-depth analysis across three social networks, Linkedin, Videolectures and ArnetMiner and addressed what reveals users' social identity, whether the social factors consistent across different social networks and how we can leverage this information to perform integration. Based on the observations, we proposed a unified framework for the social network integration task. It crawls data from multiple social networks and further discovers accounts corresponding to the same real person from the obtained networks. We use a probabilistic model to determine such correspondence, it incorporates features like the consistency of social status and social ties across different, as well as one-to-one mapping constraint and logical transitivity to jointly make the prediction. The empirical study shows that our framework can effectively integrate the three social networks which confirms the feasibility of the problem.

The future work falls into two parts. 1) In this work, we are dealing with real name social network, hence we can take the advantages of using name similarity to perform candidate generation in order to shrink the search space. In anonymous social networks, such a process wouldn’t be so straightforward, though we can still use username as a start point[?, ?]. 2) With the integrated global social graph, we want to further investigate several interesting topics such as information diffusion and user behavior pattern across different social networks that only feasible on the global social graph.