COSNET: Connecting Heterogeneous Social Networks with Local and Global Consistency

Yutao Zhang*, Jie Tang*, Zhilin Yang*, Jian Pei♯, and Philip S. Yu†‡
*Department of Computer Science and Technology, Tsinghua University
‡Institute for Data Science, Tsinghua University
♯School of Computing Science, Simon Fraser University
†Department of Computer Science, University of Illinois at Chicago
{yt-zhang13,yzl11}@mails.tsinghua.edu.cn, jietang@tsinghua.edu.cn, jpei@cs.sfu.ca, psyu@cs.uic.edu

ABSTRACT

More often than not, people are active in more than one social network. Identifying users from multiple heterogeneous social networks and integrating the different networks is a fundamental issue in many applications. The existing methods tackle this problem by estimating pairwise similarity between users in two networks. However, those methods suffer from potential inconsistency of matchings between multiple networks.

In this paper, we propose COSNET (COnnecting heterogeneous Social NETworks with local and global consistency), a novel energy-based model, to address this problem by considering both local and global consistency among multiple networks. An efficient subgradient algorithm is developed to train the model by converting the original energy-based objective function into its dual form.

We evaluate the proposed model on two different genres of data collections: SNS and Academia, each consisting of multiple heterogeneous social networks. Our experimental results validate the effectiveness and efficiency of the proposed model. On both data collections, the proposed COSNET method significantly outperforms several alternative methods by up to 10-30% ($p \ll 0.001$, t-test) in terms of F1-score. We also demonstrate that applying the integration results produced by our method can improve the accuracy of expert finding, an important task in social networks.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Text Mining; H.2.8 [**Database Management**]: Database Applications—*Data Mining*

General Terms

Algorithms, Experimentation

Keywords

Social network, Network integration, Energy-based model

KDD'15, August 10-13, 2015, Sydney, NSW, Australia.

DOI: http://dx.doi.org/10.1145/2783258.2783268.

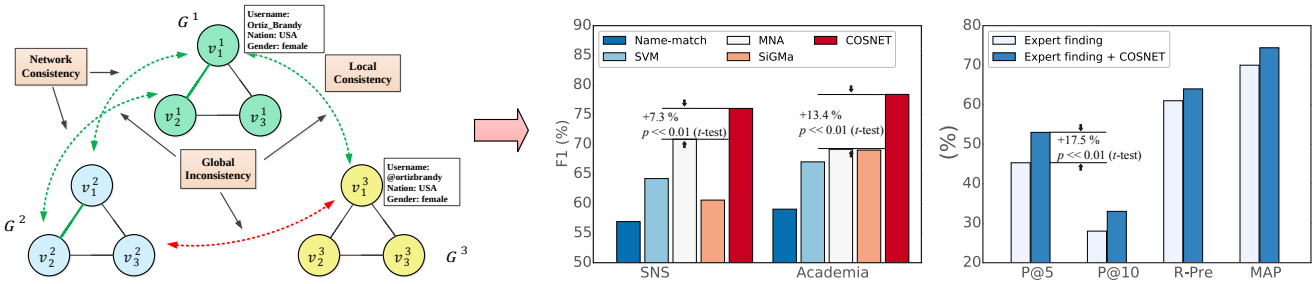
1. INTRODUCTION

We are facing an era of online with offline (OWO)—almost everyone is using online social networks to connect friends or, more generally, to satisfy social needs at different levels [26]. In fact, many users participate in more than one social network, such as public networks and private networks, as well as business networks and family networks. We conducted an interview with 20 graduate students from the authors' lab. Preliminary statistics show that the average number of social networks in which a user participates is eight. The intentions behind these choices are sophisticated. For example, people may be attracted by different functionalities offered by different social networks. A survey[1] in the US shows that two-thirds of online adults (66%) use social media platforms such as Facebook, Twitter, MySpace, or LinkedIn, to stay in touch with current friends, family members, and business partners. Users generate heterogeneous content and also build different ego-networks in different social networks. One interesting and important question is: can we automatically integrate the different heterogeneous social networks together?

The results can benefit many applications in one way or another. For example, if we could correctly integrate different social networks together, we could create an integrated profile for each user, and build a better user interest model. Talentbin[2] uses this idea to integrate professional information of an employer that scattered in different social networks to provide a better view of expertise. We can also leverage the integrated results to help social recommendations [25, 32].

The problem is fundamentally important in social network analysis and is also very challenging. First, users' information in different networks is very unbalanced. Some network may contain rich profile information such as location and interests, while some others may not have any information. Users' behavior is an important clue (also referred to as social fingerprint [40]) to help recognize users in different networks. If we use the similarity $s(u, v), u \in G^1, v \in G^2$ between users to link users from two different networks (G^1 and G^2), the problem can be formalized as an optimization problem: $\max \sum_{u,v} s(u, v)$. The problem can be solved efficiently by using a minimum cost flow algorithm [1, 34]. This formulation focuses on *local consistency* based on user profiles; however, it does not consider the network structure—an identical user in different networks may have similar ego-networks. Finding matches between the users of two networks G^1 and G^2 can be mapped onto the problem of finding bijection between two networks, a challenging problem in graph theory [36]. Approximation

[1]http://www.pewinternet.org/2011/11/15/why-americans-use-social-media/
[2]https://www.talentbin.com/, an online recruitment service.



(a) Example illustration (b) Accuracy performance (c) Application: Expert Finding

Figure 1: (a) Example illustration of global inconsistency when connecting three social networks; (b) Accuracy performance of comparison methods; (c) Application improvement by the proposed model (COSNET).

can be accomplished using a greedy algorithm. In our problem, this formulation aims to achieve network consistency, also referred to as network matching (Cf. § 3). However, if we extend this problem to the setting of multiple networks, e.g., $\{G^1, G^2, \cdots, G^k\}$, only considering local consistency and network consistency is then insufficient. Figure 1 gives an example of connecting three different networks. User v_1^1 in G^1 has high similarities with v_1^2 in G^2 and v_1^3 in G^3 (indicated by the green arrows), while v_1^3 has the highest similarity with v_3^2 in G^2. If we match any two networks independently, it can be easily seen that we will have an inconsistent results: $v_1^1 \leftrightarrow v_1^2$, $v_1^1 \leftrightarrow v_1^3$, and $v_3^2 \leftrightarrow v_1^3$. An ideal solution to the problem is to consider all pieces of information (local, network, and global consistencies) in a unified model and tackle them simultaneously.

Despite several studies on various related topics including entity linking [22, 4, 5, 20, 29], entity resolution [6, 13, 21, 31], and deanonymization [3], the problem of connecting multiple social networks remains largely unsolved. Most existing works focus on estimating pairwise similarity between users from two networks. They ignore either network consistency or global consistency. Some other methods such as [20] and [6] consider the network consistency; however they are still targeting at two networks and do not consider the global consistency among multiple networks.

Challenges and Our Solution. The problem of connecting heterogeneous social networks is non-trivial and poses a set of challenges. First, how to formulate local, network and global consistencies into a principled optimization model is a challenging issue. Moreover, as real networks are becoming larger and larger with millions of nodes, it is important to develop efficient algorithms that can scale up well. In addition, how to quantitatively validate the usefulness of integrated results is also a challenging task.

To address these challenges, in this paper, we conduct a systematic investigation into the problem of connecting multiple social networks. We formally define the problem and develop a general model to support the integration of an arbitrary number of networks. Our contributions can be summarized as follows:

- We propose COSNET (COnnecting heterogeneous Social NETworks with local and global consistency), a novel energy-based model, to formalize our problem as a unified optimization framework.
- Solving the proposed optimization model is an intractable problem. We develop an efficient algorithm by solving the dual form of the original energy-based objective function by means of a subgradient method.
- Our empirical study on two different genres of data collections, SNS and Academia, verifies the effectiveness of the proposed model. SNS consists of several popular social networks including Twitter, Flickr, Myspace, Last.fm, and LiveJournal. The Academia data collection consists of LinkedIn, ArnetMiner, and VideoLectures. Figure 4 shows the performance of different comparison methods on the two data collections. Clearly, the proposed COSNET method performs on average 10-30% better than the comparative methods in terms of F1-score ($p \ll 0.001$, t-test).
- We use expert finding, an important task in social networks, as an application case study to further validate the effectiveness of the proposed method. Figure 1(c) shows the performance of expert finding. When applying the integrated results, it is clear that the performance of expert finding can be improved (+5-10% by Precision@5).

Table 1: Notations.

SYMBOL	DESCRIPTION
$\mathbf{G}$	a set of social networks to be integrated
V	a set of $\|V\| = N$ users
E	a set of relationships between users
$\mathbf{R}$	a $N \times d$ attribute matrix
MG	a matching graph constructed from $\mathbf{G}$
$\mathbf{x}_i \in \mathbf{X}$	the i^{th} user pair
$y_i \in \{+1, 0\}$	the binary indicator representing whether i^{th} user pair in MG is a correct matching or not
$\mathbf{g}_l(.), \mathbf{f}_e(.), \mathbf{f}_t(.)$	a set of feature functions defined in the energy-based model

Organization. The rest of the paper is organized as follows. Section 2 formally defines the problem. Section 3 is devoted to our COSNET model. Section 4 presents experimental results. Section 5 reviews related works and Section 6 concludes the paper.

2. PROBLEM DEFINITION

Let $G = (V, E, \mathbf{R})$ denote a social network, where $V = \{v_1, \cdots, v_N\}$ is a set of $|V| = N$ users and $E \subseteq V \times V$ is a set of relationships between users. Each element $e_{ij} = \langle v_i, v_j \rangle \in E$ represents a directed relationship from user v_i to user v_j. If we consider undirected relationships, then we have $e_{ij} \in E \Leftrightarrow e_{ji} \in E$. Each user v_i is associated with a d-dimensional attribute vector $\mathbf{r}_i$ (the i-th row in $\mathbf{R}$), which can be defined based on the user's profile (e.g., interests or posted tweets). Given this, we define the input of our problem as follows.

Input: The input of our problem consists of a set of m social networks $\mathbf{G} = \{G^1, G^2, \cdots, G^m\}$, where $G^k = (V^k, E^k, \mathbf{R}^k)$

represents the i^{th} social network with $k \in \{1, 2, \cdots, m\}$, and $\mathbf{R}^k$ denotes an $|V^k| \times d$ attribute matrix with an element r_{ij}^k indicating the j^{th} attribute of user v_i^k in G^k. Throughout this paper, we use the integer superscript ^{k} to indicate variables (or notations) associated with the k^{th} social network. Table 1 summarizes the notations used in this paper.

Output: For a user v_i^1 in the network $G^1 \in \mathbf{G}$ and a user v_j^2 in the network $G^2 \in \mathbf{G}$, decide whether v_i^1 and v_j^2 refer to the same user in the offline real world.

A user may have multiple accounts in an online social network, which makes the problem much more complicated. For simplicity, in this work, we assume that one user has at most one account in an online social network.

3. COSNET: THE PROPOSED MODEL

In Section 2, we formulated the problem of connecting multiple networks as a binary classification problem, where our goal is to determine whether a pair of users from different networks refer to the same user. There are a lot of factors we can leverage to improve the classification performance. In this section, we propose an energy-based model [19] to jointly consider local user matching, network structure and global consistency.

Let $X = \{\mathbf{x}_i\}$ be a set of user pairs for classification and $Y = \{y_i\}$ be the set of corresponding binary labels in $\{0, 1\}$. $y_i = 1$ means the user pair $\mathbf{x}_i$ refer to the same user; otherwise not. We can define an energy function $E(Y, X)$ to model the likelihood of a configuration of X and Y. The problem of connecting multiple social networks is to find an optimal configuration $Y^\star$ given the user pairs X:

$$Y^\star = \arg\min_{Y \in \mathcal{Y}} E(Y, X)$$

where $\mathcal{Y}$ is the value space of the variable of Y. In our case, it is binary, i.e., $\mathcal{Y} = \{+1, 0\}^n$, respectively representing whether a candidate matching is correct or not.

Figure 2 gives an example to illustrate how we generate candidate user pairs X. For two input networks G^1 and G^2, each containing three users, we could generate nine user pairs. Each node in Figure 2(b) represents a user pair (candidate matching) and the edge represents the relationship between two connected nodes (the corresponding two candidate matchings). The relationships are generated according to the neighborhood relationships in the original networks. Thus we could construct a matching graph MG (as defined below). The resultant matching graph might be very large, if the input two networks contain many users. We prune the generated matching graph using heuristics. Figure 2(c) shows an example of matching graph after pruning. The red edges indicates the one-to-one matching constraint as we assume that each user has at most one user account in each network.

We will discuss the detailed formulation of the energy function in the following sections.

3.1 Local Matching

First we define some local features for a single user pair $\mathbf{x}_i$. We consider the similarity of user profiles, which is widely used in entity linking [4, 5, 20, 29] and object resolution [6, 13, 31].

Let $\mathbf{g}_l(\mathbf{x}_i, y_i)$ be a vector-valued feature function for encoding the user profile similarity for the user pair $\mathbf{x}_i$. We model the local matching energy function as:

$$E_l(Y, X) = \sum_i \mathbf{w}_l^\intercal \mathbf{g}_l(\mathbf{x}_i, y_i)$$

where $\mathbf{w}_l$ is a model parameter.

3.2 Network Matching

Local matching deals with each pair of users independently, and does not consider the network structure. To leverage the network structure, we should consider "neighborhood-preserving matching". The basic idea is that if user $v_i^1 \in G^1$ is matched onto $v_j^2 \in G^2$, then we also hope that v_i^1's neighbors can be also matched to v_j^2's neighbors in G^2. In graph theory, the problem can be reduced to graph isomorphism, where the objective is to find a structure-preserving bijection between the vertex sets of two graphs [36].

Here we first introduce the definition of a *matching graph*, and then we will show how to incorporate the network matching idea into the energy-based model.

DEFINITION 1 (MATCHING GRAPH). *Let $\mathcal{V}_i^j$ be the j-th vertex of the user pair $\mathbf{x}_i$ where j takes a value in $\{1, 2\}$. Let $\mathcal{N}(v)$ be the neighbour set of user v in a social network. Given a set of social networks $\mathbf{G}$ and user pairs X, we construct a graph $MG = (V_{MG}, E_{MG})$ by the following two steps:*

1. *For each user pair $\mathbf{x}_i \in X$, construct a corresponding node in V_{MG}.*
2. *For two user pairs $\mathbf{x}_i$ and $\mathbf{x}_j$, if $\mathcal{V}_i^1 \in \mathcal{N}(\mathcal{V}_j^1)$ and $\mathcal{V}_i^2 \in \mathcal{N}(\mathcal{V}_j^2)$, then construct an edge in E_{MG} between the nodes $\mathbf{x}_i$ and the nodes $\mathbf{x}_j$.*

The resulting graph MG is a matching graph given X and $\mathbf{G}$.

Given the definition of matching graph, we model the network matching energy function as:

$$E_e(Y, X) = \sum_{\langle \mathbf{x}_i, \mathbf{x}_j \rangle \in E_{MG}} \mathbf{w}_e^\intercal \mathbf{f}_e(y_i, y_j) \tag{1}$$

with

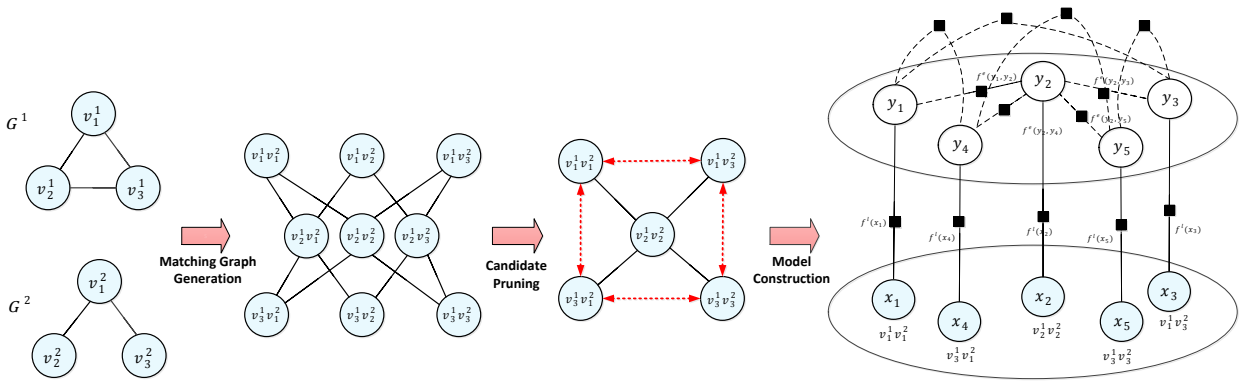
$$\mathbf{f}_e(y_i, y_j) = \begin{cases} (1,0,0)^\intercal & \text{if } y_i = y_j = 0 \\ (0,1,0)^\intercal & \text{if } y_i + y_j = 1 \\ (0,0,1)^\intercal & \text{if } y_i = y_j = 1 \end{cases}$$

where $\mathbf{w}_e$ is a model parameter and $\mathbf{f}_e(\cdot)$ is an indexing function. When y_i and y_j takes different binary values, the function $\mathbf{f}_e$ outputs 1 in one dimension and 0 in others, indicating which combination of y_i and y_j is activated.

Now we explain why the defined energy function in Eq. 1 can leverage the network matching idea. Suppose that the principle of "neighborhood-preserving matching" is effective in the real world datasets. After we learn the model parameter $\mathbf{w}_e$ from the training data, the weight for y_i and y_j being the same will be low. For the test set, if two user pairs are connected in the matching graph, the model energy will be low if they are assigned the same label. Therefore, by the definition of the network matching energy model in Eq. 1, we tend to assign the same labels for neighbors in the matching graph to minimize the model energy.

3.3 Global Consistency

Generally, our energy-based model with local matching and network matching will work well when we have only two social networks to connect. However, when we have more than two networks, we need to address the *inconsistency* problem. For example, given three networks G^1, G^2 and G^3, connecting $\langle v_i^1, v_j^2 \rangle$ and $\langle v_j^2, v_k^3 \rangle$ without connecting $\langle v_k^3, v_i^1 \rangle$ will cause the inconsistency problem, because the connections did not consider the transitivity of user identities. Formally, we define global inconsistency as follows:



(a) Two input networks (b) The generated matching graph (c) Matching graph after pruning (d) The constructed model

Figure 2: Illustration of candidate matching graph generation and the constructed model. (a) two input networks; (b) the generated matching graph; (c) matching graph after pruning; (d) the constructed model.

DEFINITION 2 (GLOBAL INCONSISTENCY). *Given a set of social networks* $\mathbf{G}$*, a set of user pairs* X *and the corresponding labels* Y*, if there exists a sequence of user pairs* $\langle \mathbf{x}_{i_1}, \mathbf{x}_{i_2}, \cdots, \mathbf{x}_{i_n} \rangle$*, such that*

$$\forall i = i_1, i_2, \cdots, i_n, y_i = 1$$

and

$$\forall k = 1, 2, \cdots, n-1, \mathcal{V}^2_{i_k} = \mathcal{V}^1_{i_{k+1}}$$

and

$$\text{For the pair } \langle \mathcal{V}^2_{i_n}, \mathcal{V}^1_1 \rangle, \text{ if the corresponding label } y_j = 0$$

then we say that the assigned labels Y *causes global inconsistency given* $\mathbf{G}$ *and* X*.*

In this section, we add another energy function into our model to penalize global inconsistency. Ideally, we should consider all possible user pairs sequences in X. However, the number of valid user pairs sequences grows exponentially with the number of users in the networks. Therefore, it is infeasible to directly penalize all the sequences. Instead, we approximately only consider sequence of length 3 in our model. Figure 3 gives an example. Suppose we have three networks G^1, G^2, G^3, and three candidate user pairs among them, (v^1_2, v^2_2), (v^1_2, v^3_1), and (v^2_2, v^3_1), the edges between pairs indicates two user pairs share a user. The three pairs form a triadic closure in the matching graph (as shown in Figure 3). Thus, we define a triad-based energy function for each closed triad in the matching graph.

Let T_{MG} be the set of closed triads in the matching graph MG, we can write the energy function for global consistency as follows:

$$E_t(Y, X) = \sum_{c \in T_{MG}} \mathbf{w}_t^\intercal \mathbf{f}_t(Y_c)$$

with

$$\mathbf{f}_t(y_i, y_j) = \begin{cases} (1,0,0,0)^\intercal & \text{if } |Y_c| = 0 \\ (0,1,0,0)^\intercal & \text{if } |Y_c| = 1 \\ (0,0,1,0)^\intercal & \text{if } |Y_c| = 2 \\ (0,0,0,1)^\intercal & \text{if } |Y_c| = 3 \end{cases}$$

where $\mathbf{w}_t$ is a model parameter and $Y_c = (y_i, y_j, y_k)$ are the labels of the triad $c = (\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k)$; $|Y_c|$ represents the number of 1s in the triad; $\mathbf{f}_t$ is an indexing function reflecting whether labels in Y_c are consistent.

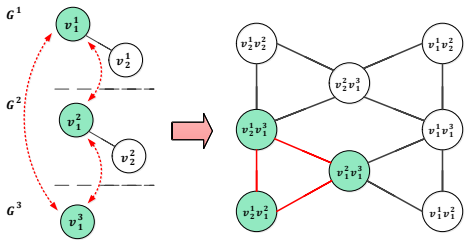


Figure 3: Modeling global consistency with triad-based energy.

3.4 Model Learning

Finally, by combing together local matching, network matching and global consistency, we obtain the following objective function:

$$\begin{aligned} E(Y, X) \quad = \quad & \sum_{\mathbf{x}_i \in V_{MG}} \mathbf{w}_l^\intercal \mathbf{g}_l(\mathbf{x}_i, y_i) + \sum_{\langle \mathbf{x}_i, \mathbf{x}_j \rangle \in E_{MG}} \mathbf{w}_e^\intercal \mathbf{f}_e(y_i, y_j) \\ & + \sum_{c \in T_{MG}} \mathbf{w}_t^\intercal \mathbf{f}_t(Y_c) \end{aligned} \tag{2}$$

Given a labeled training dataset, we aim to learn a parameter configuration $W = (\mathbf{w}_l, \mathbf{w}_e, \mathbf{w}_t)$ from a labeled dataset. We use a maximum-margin method for model learning [15, 35]. Specifically, we first define the distance of two matching configurations (Y and Y') in the energy space as:

$$\begin{aligned} \Delta(Y, Y') \quad = \quad & \sum_{\mathbf{x}_i \in V_{MG}} \delta_l(y_i, y'_i) + \sum_{c \in T_{MG}} \delta_c(Y_c, Y'_c) + \\ & \sum_{\langle \mathbf{x}_i, \mathbf{x}_j \rangle \in E_{MG}} \delta_e(\langle y_i, y_j \rangle, \langle y'_i, y'_j \rangle) \end{aligned}$$

where $\delta_l(y_i, y'_i)$, $\delta_e(\langle y_i, y_j \rangle, \langle y'_i, y'_j \rangle)$, $\delta_l(Y_c, Y'_c)$ respectively represents a distance function between local, network, and global energy functions. Here we use Hamming distance as the distance function.

Therefore, according to the maximum margin theory [12, 35], our objective function (Eq. 2) can be rewritten as a minimization over a regularized hinge loss function:

$$\min_{W} \frac{1}{2}||W||^2 + \mu\xi$$
$$\text{s.t.} \quad E(\hat{Y}, X; W) \leq E(Y, X; W) - \Delta(Y, \hat{Y}) + \xi \tag{3}$$

where ξ is a slack variable to handle non-separable data, by allowing some constraints to be violated (or by introducing an error penalty); μ controls the trade-off between the maximum margin and the error penalty; with the constraint, the model forces to find a parameter setting such that the distance between the manual labeled matching configuration $\hat{Y}$ and a matching configuration Y by at least $\Delta(\hat{Y}, Y)$. The constraint can be redefined as

$$E(\hat{Y}, X; W) - \min_{Y}(E(Y, X; W) - \Delta(Y, \hat{Y})) = \xi$$

Moreover, by redefining the energy function, we can absorb the distance function $\Delta(\hat{Y}, Y)$ into $E(\hat{Y}, X; W)$ and $E(Y, X; W)$ [14]. Finally, the objective function becomes:

$$\min_{W} \frac{1}{2}||W||^2 + \mu(E(\hat{Y}, X; W) - \min_{Y}(E(Y, X; W)) \tag{4}$$

Dual Decomposition. Solving Eq. 4 is an intractable problem. One challenging task is to estimate the energy function $E(.)$. To this end, we use Lagrangian relaxation to decompose the function into a set of easy-to-solve subproblems. Specifically, in our problem, given the input model (Cf. Figure 2), we decompose the graphical structure into a set of function-based subgraphs, each representing a function. Accordingly, we obtain the following decomposition:

$$\begin{aligned} E(Y, X; W) &= \sum_{f \in \mathcal{F}} E_f(Y_f, X_f; W) \\ &= \sum_{f \in \mathcal{F}} \sum_{\mathbf{x}_i \in X_f} (\frac{1}{|\mathcal{F}_i|} \mathbf{w}_l^\intercal \mathbf{g}_l(\mathbf{x}_i, y_i^f) + \mathbf{w}_f^\intercal f(Y_f)) \\ \text{s.t.} \quad & y_i^f = y_i, \ \ \forall f, y_i \in Y_f \end{aligned} \tag{5}$$

where $f \in \mathcal{F}$ is a defined factor function and $\mathcal{F}$ is the space of all defined factor functions; $\mathbf{w}_f$ is the parameter of f; X_f is a set of user pairs related to function f and Y_f is the set of corresponding labels ; y_i^f is a local replica of the variable y_i in the function $f(.)$; and $\mathcal{F}_i$ is the set of functions to which a variable y_i is related.

By imposing the constraint $y_i^f = y_i, \forall f, y_i \in Y_f$, we obtain that minimizing this function is equivalent to minimizing the original energy function Eq. 2. To relax these coupling constraints, we introduce a set of Lagrange multipliers $\boldsymbol{\lambda} = \{\lambda_i^f : f \in \mathcal{F}, y_i \in Y_f\}$

$$\begin{aligned} L(Y, X, \boldsymbol{\lambda}; W) &= \min_{W} \sum_{f \in \mathcal{F}} \left(\sum_{y_i \in Y_f} \frac{1}{|\mathcal{F}_i|} \mathbf{w}_l^\intercal \mathbf{g}_l(\mathbf{x}_i, y_i^f) + \mathbf{w}_f^\intercal f(Y_f) \right) \\ &\quad + \sum_{f \in \mathcal{F}} \sum_{y_i \in Y_f} \lambda_i^f (y_i - y_i^f) \\ &= \min_{W} \sum_{f \in \mathcal{F}} \left(\sum_{y_i \in Y_f} \frac{1}{|\mathcal{F}_i|} \mathbf{w}_l^\intercal \mathbf{g}_l(\mathbf{x}_i, y_i^f) + \mathbf{w}_f^\intercal f(Y_f) + \lambda_i^f y_i^f \right) \\ &\quad + \sum_{f \in \mathcal{F}} \sum_{y_i \in Y_f} \lambda_i^f y_i \end{aligned} \tag{6}$$

The last term is eliminated by setting $\sum_{y_i \in Y_f} \lambda_i^f = 0$. Since $L(Y, X, \boldsymbol{\lambda}; W)$ optimizes over a larger space, it always satisfies $L(Y, X, \boldsymbol{\lambda}; W) \leq E(Y, X; W)$. Hence, it provides a lower bound to the minimum energy of the origin problem. Thus, we form the dual problem to maximize the lower bound by adjusting λ.

$$\begin{aligned} &\max_{\boldsymbol{\lambda}} L(Y, X, \boldsymbol{\lambda}; W) \approx \min_{Y} E(Y, X; W) \\ &\text{s.t.} \sum_{y_i \in Y_i} \lambda_i^f = 0, \ \ \forall f \in \mathcal{F} \end{aligned} \tag{7}$$

Thus, the objective function becomes

$$\begin{aligned} &\min_{W, \boldsymbol{\lambda}} \frac{1}{2}||W||^2 + \mu(E(\hat{Y}, X; W) - \max_{\boldsymbol{\lambda}} L(Y, X, \boldsymbol{\lambda}; W)) \\ &\text{s.t.} \sum_{y_i \in Y_i} \lambda_i^f = 0, \ \ \forall f \in \mathcal{F} \end{aligned} \tag{8}$$

The resulting dual objective function is convex and non-differentiable, hence it can be solved by the projected subgradient method. Specifically, we have two sets of parameters, W and $\boldsymbol{\lambda}$, to estimate. We optimize the dual objective function by alternately updating W and $\boldsymbol{\lambda}$. The subgradients are given by:

$$dW = \frac{\partial}{\partial W} \left(\frac{1}{2}||W||^2 + \mu(E(\hat{Y}, X; W) - \max_{\boldsymbol{\lambda}} L(Y, X, \boldsymbol{\lambda}; W)) \right) \tag{9}$$

$$d\lambda_i^f = \frac{\partial}{\partial \lambda_i^f} L(Y, X, \boldsymbol{\lambda}; W) \tag{10}$$

Algorithm 1 summarizes the procedure for learning the model. The model runs in a semi-supervised fashion, which means that after learning the model, we can also obtain unknown matchings. Specifically, in each iteration, we first apply the learned parameters to infer unknown matchings $Y^\star = \arg\max_Y E(Y, X)$, and then use all the matchings to estimate model parameters W.

Implementation notes. The subgradient method is guaranteed to converge to the optimal solution of the dual problem when the step size satisfies $\lim_{t \to \infty} \eta_t = 0$ and $\sum_{t=0}^{\infty} \eta_t = \infty$ [2]. We set the step size as $\eta_t = \frac{e}{\sqrt{t}}$, where $e > 0$ is a tunable parameter.

ALGORITHM 1: Model Learning

Input: Labeled candidates $(X^l, \hat{Y}^l)$, unlabeled candidates X^u, factor functions $\mathcal{F}$, step-sizes $\{\eta_t\}$
$\boldsymbol{\lambda} \leftarrow \mathbf{0}$;
repeat
  foreach *factor* $f \in \mathcal{F}$ **do**
    // Optimize over subgraphs
    Compute $Y_f{}^\star = \arg\min_{Y_f} E_f(Y_f, X_f; W)$;
  end
  Compute dW based on eq. 9 ;
  // Feature weight update
  $W \leftarrow W - \eta_t \cdot dW$;
  Compute $d\boldsymbol{\lambda}$ based on eq. 10 ;
  // Dual variable update
  $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \eta_t \cdot d\boldsymbol{\lambda}$;
until *convergence*;
Output: The best configuration of unlabeled candidates $Y^{u\star}$;

3.5 Candidate Generation

In practice, the generated candidate matching graph might be very large. Here, we introduce several strategies for generating the candidate matching graph.

Content-based method. A straightforward solution is to use the username as the identification to generate seed matchings. Following [39, 40, 28], we define a similarity function for usernames. Specifically, we tokenize each username u_i into a set of segments $\{u_{i1}, u_{i2}, ...\}$ and then represent each username u_i as a TF-IDF weighted vector $\mathbf{w}_u$. We then calculate the cosine similarity between u_i and u_j base on their segments, i.e. $\text{sim}(u_i, u_j) = \frac{\mathbf{w}_{u_i} \cdot \mathbf{w}_{u_j}}{||\mathbf{w}_{u_i}|| ||\mathbf{w}_{u_j}||}$. We then add the user pairs into the candidate set, if the similarity is above a threshold. To speedup the computation, we also build an inverted index for each username. The similarity is computed only for pairs of usernames that share at least one segment. Finally, a threshold is used to determine whether two usernames belongs to the same user.[3]

Structure-based method. The second method expands candidate matchings base on the network structure. Specifically, based on the seed matchings discovered in the content-based method, we propagate the matching along with neighborhood relationships, and iteratively augment the mapping graph with new candidate matchings if their matching scores exceed a threshold[2]. Similar ideas have been studied in [27, 38].

Hybrid method. We consider a hybrid of the above two methods to generate candidates. The idea is first to use the content and the structure-based methods to generate candidate matchings and then to combine the outputs of the two methods.

4. EXPERIMENTAL RESULTS

All datasets and codes used in this work are publicly available.[4]

4.1 Data Collections

We perform experiments on two data collections: **SNS** and **Academia**. Each data collection consists of several social networks. In the different networks of each data collection, both the users and the meanings of the relationships in the different networks are very different. Table 3 lists statistics of the two data collections.

SNS Network Collection. The SNS data collection consists of five popular online social networking sites: Twitter (TW), LiveJournal (LJ), Flickr (FL), Last.fm (LA), and MySpace (MS).

Twitter: a widely used microblogging service. The Twitter dataset we used in our experiment is obtained from [17], and it consists of 40.1 million user profiles and 1.47 billion following relationships among them. Taking user profiles as network nodes, the following relationships allows us to reconstruct the network within them.

LiveJournal: a free online social network where users can keep a blog, journal, or diary. The dataset was crawled from its website in late 2013, and contains 3,017,286 users and 87,037,567 friend relationships.

Flickr: a popular photo-sharing network that allows users to upload and share photos. The Flickr dataset was crawled in early 2014, and consists of 215,495 individual users and 9,114,557 friend relationships.

Last.fm: a streaming radio service, where users can search for music and get personalized recommendations. Last.fm builds detailed profiles of users' musical tastes and preferences. The dataset was collected in late 2013 and consists of 136,420 users and 1,685,524 following relationships.

[3]We empirically set the threshold as 0.8.

[4]http://aminer.org/cosnet

Table 3: Statistics of the two datasets. SNS consists of five networks and Academia consists of three networks.

Dataset	Network	#Users	#Relationships
SNS	Twitter	40,171,624	1,468,365,182
	LiveJournal	3,017,286	87,037,567
	Flickr	215,495	9,114,557
	Last.fm	136,420	1,685,524
	MySpace	854,498	6,489,736
Academia	LinkedIn	2,985,414	25,965,384
	ArnetMiner	1,053,188	3,916,907
	VideoLectures	11,178	786,353

MySpace: a social networking website for sharing music. The dataset contains 854,498 user profiles and 6,489,736 directed relationships among users.

Academia Network Collection. The Academia data collection consists of three academic or professional social networks: ArnetMiner (AM), LinkedIn (LI) and VideoLectures (VL).

ArnetMiner: an expertise search and mining service for the academic community [33]. We obtained the entire ArnetMiner data up to 2013, which consists of 1,053,188 user profiles and 3,916,907 co-author relationships.

LinkedIn: a professional network, where users can maintain their profiles and social connections. We collected public profiles from LinkedIn. As we cannot crawl user connections on LinkedIn, we pursued another method to construct the network. We consider two profiles to be linked if they were viewed ("co-viewed") by the same user. In this way, we obtained a network of 2,985,414 user profiles and 25,965,384 relationships.

VideoLectures: an open access educational video lectures repository. The lectures are given by distinguished scholars and scientists at academic events like conferences and summer schools. We obtained lecturers' profiles from the website. We defined a connection between two researchers as indicating that they attended a particular event (e.g., KDD 2014). This yielded a network of 11,777 lecturer profiles and 786,353 "co-attendance" relationships.

Ground Truth. It is difficult to obtain a "ground truth" for evaluating the proposed method. For the SNS network collection, we use the linked user accounts dataset from [28, 8] as the "ground truth". The data was originally collected by Perito el. al [28] through Google Profiles service by allowing users to integrate different social network services. For the Academia network collection, we obtained the "ground truth" from ArnetMiner. The ArnetMiner system allows users to connect each author's profile to external social networks such as LinkedIn and Facebook. We chose 10,000 authors from ArnetMiner who were connected with LinkedIn profiles and VideoLectures profiles as the ground truth. More precisely, in each data collection, if a user has accounts in two different networks, then we have a ground truth matching.

Feature Definition. We now define the local feature functions in Section 3. We tried to define them with sufficient generality to allow the model to be easily adapted to different data collections.

Username similarity and uniqueness: We use Jaro-Winkler distance to measure username similarity [9]. Two accounts with similar usernames are likely to belong to the same person. We also measure how unique the usernames are [39, 28, 40, 22]. To estimate the uniqueness of a username, we adopt the language-model based approach from [28]. More specifically, we train a Markov-chain model based on all the usernames from Twitter and LinkedIn

Table 2: Performance comparison of different methods for network integration task. The results are presented jointly and separately for each pair of networks .

	Name-match			SVM			MNA			SiGMa			COSNET			COSNET -		
Network Pair	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Twitter · LiveJournal	77.05	54.77	64.02	61.91	73.92	67.39	70.53	74.54	72.48	91.03	43.31	58.69	77.44	75.83	**76.63**	76.12	75.52	75.82
Twitter · Flickr	82.75	58.69	68.67	60.46	71.72	65.61	77.52	70.59	73.90	92.75	47.69	62.99	81.06	73.82	**77.28**	80.33	73.17	76.58
Twitter · Last.fm	72.21	54.19	61.95	46.39	71.26	56.20	75.09	69.41	72.14	90.21	44.19	59.32	80.03	72.14	**75.89**	77.21	73.85	75.49
Twitter · MySpace	73.29	41.82	53.25	57.95	72.44	41.74	75.16	66.94	70.82	88.29	35.82	50.96	81.46	70.25	**75.45**	78.48	69.56	73.75
LiveJournal · Flickr	87.30	51.89	65.09	59.25	77.53	67.16	75.00	73.25	74.12	93.30	46.89	62.41	83.69	79.38	**81.48**	81.52	78.12	79.78
LiveJournal · Last.fm	86.58	42.46	56.98	44.05	60.50	50.98	73.89	62.47	67.70	91.58	39.46	55.15	77.50	70.93	**74.07**	75.94	69.76	72.73
LiveJournal · MySpace	80.43	36.73	50.44	44.29	57.26	49.95	79.66	58.14	67.22	89.43	32.73	47.92	85.18	65.57	**74.10**	82.48	64.17	72.18
Flickr · Last.fm	95.81	44.90	61.15	78.46	60.00	68.00	82.70	62.44	71.16	96.81	41.90	58.49	88.18	67.84	**76.69**	84.06	66.52	74.27
Flickr · MySpace	96.49	43.65	60.11	66.17	54.26	59.63	87.68	57.79	69.66	98.49	34.65	51.26	90.04	62.63	**73.87**	89.05	61.93	73.06
Last.fm · MySpace	95.65	31.86	47.80	69.92	53.26	60.47	79.53	55.83	65.60	95.65	32.86	48.92	82.27	61.38	**70.31**	81.16	59.26	68.68
Overall	81.18	44.46	56.94	59.79	69.32	64.20	76.05	66.76	70.84	91.27	45.33	60.57	82.05	71.10	**76.04**	76.15	68.43	72.08
ArnetMiner · LinkedIn	59.50	69.94	64.30	67.65	87.19	76.18	78.51	72.14	75.13	88.50	47.39	61.72	86.02	86.74	**86.38**	84.84	83.15	83.98
ArnetMiner · VideoLectures	46.56	84.10	59.94	40.63	86.92	55.37	64.27	83.32	73.96	87.56	58.10	69.85	70.76	89.46	**79.01**	69.19	87.10	77,10
LinkedIn · VideoLectures	11.75	82.22	20.55	30.45	58.89	40.15	36.71	73.33	48.92	85.75	54.22	66.43	50.12	86.67	**65.31**	37.48	79.90	51.02
Overall	48.59	75.22	59.04	52.09	86.73	67.02	62.31	77.65	69.14	87.27	57.08	69.01	70.96	87.62	**78.41**	64.49	80.41	71.58

and assign a uniqueness value $-\log_2(p(u))$ to each username u, where $p(u)$ is the n-gram probability of u.

Profile content similarity: We combine all information related to a profile into one document and convert the document into a bag-of-words vector, where the words are weighted by TF-IDF. The similarity of two profiles is then measured by inner product and cosine distance.

Ego network: To capture the similarity or overlaps between the ego networks of two accounts on different networks, [16] proposed the notion of extended common neighbors and defined it as the number of known anchor links between two ego networks. Three similarity features are defined between two users' ego networks using common neighbors, Jaccard's coefficient and Adamic/Adar score to measure the similarity between ego networks.

Social status: According to our preliminary analysis of real-world social networks, the status of a user in different social networks tends to be consistent. We use pagerank to measure social status. We sort accounts by pagerank score in each social network and we call the top 1% accounts "opinion leaders"[5], the following 10% "middle class", and the rest "ordinary users". Then given a candidate matching (v_i^1, v_j^2).

4.2 Evaluation Metrics and Baselines

To quantitatively evaluate the proposed model, we consider the following performance metrics:

- **Accuracy.** In both data collections, if a method can find a matching between the two networks, we say that the method correctly recognizes a matching; otherwise, we say that the method makes a wrong recognition. We evaluate the comparison methods in terms of Precision, Recall, and F1-Measure.
- **Application improvement.** We apply the obtained matching graph to help expert finding. This will demonstrate how the matching results can benefit other real applications.

Comparison Methods. We compare the following methods for connecting multiple networks.

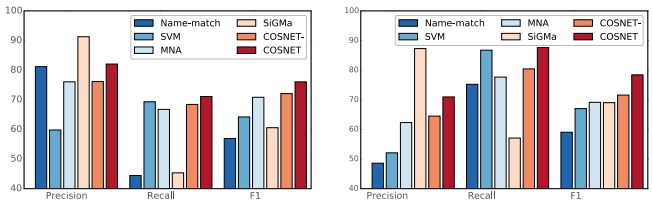
Name-match: This method considers a candidate matching to be correct if and only if the two users have exactly the same name.

SVM: This method formalizes the matching problem as a classification problem. It trains a classification model based on the labeled data, and then applies the classification model to classify whether a candidate matching is correct or not. It uses all local features defined in § 4.1 for training the classification model.

MNA [16]: This method leverages the output score of SVM as the initial pairwise score and then optimize the matching problem by satisfying the one-to-one constraints.

SiGMa [18]: This method was designed to align two knowledge bases, by propagating the confidence score in the matching graph. Specifically, it starts from a seed mapping set and iteratively augments the seed set with new matched pairs. We use the result of the Name-match method (by filtering out the most common names) as the seed set and the output confidence score of SVM as the pairwise similarity for SiGMa.

COSNET: This is the proposed method in the paper, which considers both local consistency and global consistency. To emphasize the importance of global consistency, we also evaluate a variant of the method by removing the global consistency. The variant version is referred to as COSNET$-$.



(a) SNS (b) Academia

Figure 4: Performance of integrating two network collections

4.3 Performance Analysis

We perform experiments on both data collections: SNS and Academia. In our experiments, we randomly partition the ground-truth mappings into five groups and conduct five-fold cross-validation and report the average results.

Prediction Performance. Figure 4 shows the overall performance of the comparison methods on the two data collections. The proposed method achieves clearly better performance than the other comparison methods. In terms of F1-score, COSNET achieves a $10-30\%$ improvement over SVM, MNA, and SiGMa. This

[5]Statistics have shown that 1% of the Twitter users produce 50% of its content [37] and control 25% of the information diffusion [24].

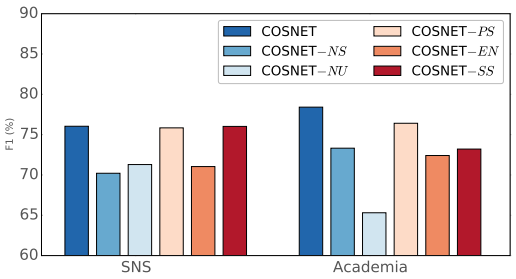


Figure 5: Factor contribution analysis

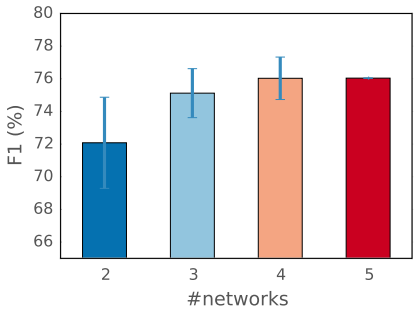
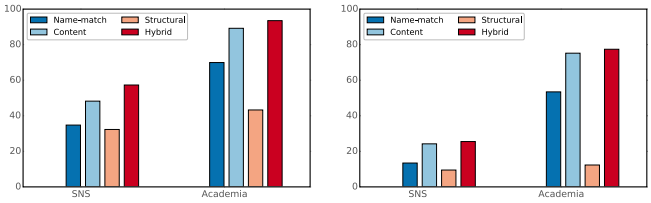


Figure 6: Performance of COSNET when varying the number of networks involved in the SNS data collection.



(a) Coverage rate (b) Negative rate

Figure 7: Performance of candidate generation and pruning.

demonstrates that our method, which combines local matching and global consistency is useful.

We also report the evaluation results for each pair of networks. Table 2 lists the detailed performance of each pair of networks. Again, COSNET achieves the best performance in all tasks. SiGMa results in high precision but suffers from a low recall in the SNS dataset. In the Academia dataset, the situation is better, while overall it still underperforms our proposed method. We deepened the analysis and found that SiGMa is good for matching when the input networks have significant overlap; however it cannot handle the sparsity problem, Thus when the input networks have less overlap, e.g., the SNS dataset, it results in very low recall due to its inability to perform propagation. This confirms the necessity of considering global consistency for connecting multiple networks.

Factor Contribution Analysis. In the COSNET model, we define five local features based on similarity between users: username similarity (US), username uniqueness (UN), profile content similarity (PS), ego network similarity (EN), and social status (SS). Here we examine the contributions of the different local factors. Specifically, we remove one of the five features from the model, and evaluate the deterioration in its prediction performance. Figure 5 shows the F1-score on the two datasets. We see that different local factors contribute differently in the different data collections. The name uniqueness feature is very helpful in the Academia data collection, but not that useful in the SNS data collection. This is probably because the name uniqueness feature can effectively address the name ambiguity problem in the Academia data collection.

Effects of Global Consistency. One important feature of our proposed COSNET model is that it is able to preserve global consistency. We now investigate how global consistency helps improve matching performance. As shown in Figure 4 and Table 2, COSNET− represents the performance of our method without considering global consistency. We see that without considering global consistency, the overall performances drop significantly (-2% and -7%; p-value$\ll 0.01$, t-test) on both data collections. We see that the major problem of COSNET− is a drop in precision. This is reasonable, as global consistency can help improve precision when dealing with more than two networks.

Effects of Number of Social Networks. Intuitively, the more networks involved, the greater the improvements we can obtain by considering the global consistency. To verify the effectiveness of global consistency when dealing with different number of social networks. We evaluate the average performance of COSNET by varying the number of social networks in SNS. Specifically, we first consider two networks. In this case, COSNET degrades to COSNET−. Then we increase the number of networks to three, four, and five, and report the average F1-score of COSNET for matching the multiple networks. Figure 6 shows the average performance of COSNET when varying the number networks involved in the SNS data collection. The results validate our intuition regarding global consistency.

Effects of Candidate Pruning. We evaluate the performance of candidate generation in terms of *coverage rate* (the number of known positive candidates divided by the number of ground-truth mappings) and *negative rate* (the number of known negative candidates divided by the sum of the number of known positive and negative candidates). Figure 7 shows the coverage rate and negative rate in each candidate generation strategy described in § 3.4. In most cases, propagation can significantly increase the coverage of candidates with a reasonable number of errors. We also note that the propagation algorithm failed to output desirable candidates on AM · VL and LI · VL. This might be because the dense co-attendance relationships in VideoLectures cannot reflect real-world relationships among researchers.

4.4 Application Improvement

We turn now to evaluating the performance improvement when applying the model output to expert finding [33]. Recall that the goal of expert finding is to identify persons with some expertise or experience on a specific topic (query) q. The experiment is conducted on an Academia dataset used in [41], which consists of 14,134 researchers. Four-vaule scores (3, 2, 1, and 0) are manually labeled to represent definite expertise, expertise, marginal expertise, and no expertise. All the researchers have accounts on ArnetMiner, and according to our integration result, 5,374 of them have been successfully mapped to the LinkedIn network. We construct an integrated social graph of ArnetMiner and LinkedIn by simply appending the profile content and relationships of the mapped accounts on LinkedIn to the ArnetMiner network. Recall that in LinkedIn, the edges indicates the "co-viewed" relationship between users which can be regarded as indicating that the corresponding users are similar in terms of their expertise. As an example, Philip

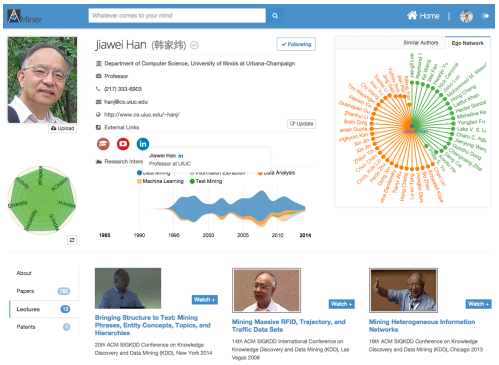


Figure 8: Screenshot of the prototype system.[6]

Yu has co-viewed relationships with Wei Wang, Jian Pei, Charu Aggarwal, Jiawei Han, Hui Xiong, Vipin Kumar, etc. These people are not necessarily coauthors of Philip, but their expertise is similar. Our goal here is to leverage the external network, LinkedIn, to help improve expert finding in ArnetMiner.

To quantify the effectiveness of the integration result, we perform expert finding on both ArnetMiner's coauthor network and the integrated social graph. For fair comparison, we use the same algorithm in [33] which is a combination of language model and random walks. Specifically, we calculate the expertise score of an author for a given query based on language model, and propagate the score in the network using random walks. With the integrated social graph (by matching ArnetMiner and LinkedIn), we have more "co-viewed" relationships to propagate the expertise scores.

We evaluate the performance in terms of precision@5 (P@5), P@10, P@20, R-precision (R-Pre), and mean average precision (MAP) [7]. Figure 1(c) shows the result of expert finding, on the ArnetMiner network only, and on the integrated networks. We see that the external network information from LinkedIn can significantly improve the performance of expert finding (+5-10% by Precision@5). This confirms the effectiveness of connecting different networks in real applications.

4.5 Prototype System

We have developed and deployed a web application for an integrated social network based on the proposed COSNET method. The system integrates LinkedIn and VideoLectures with ArnetMiner. As a result, we have successfully discovered 237,842 matched pairs between LinkedIn and ArnetMiner and 8,932 between VideoLectures and ArnetMiner (representing 85% of the profiles on VideoLectures). We use the profile content from LinkedIn and videos from VideoLectures to enrich the corresponding user profile on ArnetMiner. Figure 8 shows an example author profile on ArnetMiner.[6] The top of the figure shows the integrated external relationships, and the profile content from LinkedIn has been used to enrich the corresponding user profile on ArnetMiner. The bottom lists videos from VideoLectures.

5. RELATED WORK

We briefly review related literature from two aspects: connecting users across social networks, and entity linking.

Connecting Users across Social Networks. An immediate method to connect users in different networks is to leverage usernames [39, 28, 40, 22]. Zafarani et al. [39] were the first to address this problem. Their approach utilized prefix/postfix addition and removal to map usernames from a base community to a target community. Peritio et al. [28] estimated the uniqueness of usernames with a Markov chain model. The recent work by Zafarani et al. [40] conducted a more in-depth investigation of this problem. They defined sophisticated features, such as knowledge limitation and typing patterns, to model the behavior patterns of users in selecting usernames. Liu et al. [22] leveraged rare usernames to create training instances for user identification. Liu et al. [23] explored the social identity linkage problem based on user behavior modeling. They proposed a method to incorporate user attributes, user generated content, and social activities to link user accounts in different social networks.

The above methods only considered local similarity of account pairs. Zhang et al. [16] formulated the problem as an anchor-link prediction task. Their method resolved one-to-one mapping conflicts and issues from encountered structural features, but was limited to pairwise mapping. Tan et al. [30] used hypergraph to model social relationships and proposed a manifold alignment framework to map users from different networks onto a common low-dimension space. Cui et al. [11] studied the problem of finding email correspondents in social networks. Their approach integrated profile similarity and graph-based similarity and successfully found a mapping between the Facebook network and an email network.

Most of the above work focuses on estimating pairwise similarity between users from two networks. However, they do not consider global consistency among mappings of multiple networks.

Entity Linking and Disambiguation. Linking entities from different information sources is an important and extensively studied problem in data integration. Bellare [5] proposed an active sampling approach to improve entity-matching results. Bai et al. [4] leveraged user clicks for automatic seed generation in entity matching. Lacoste et al. [18] proposed a greedy method that successfully aligned two large-scale knowledge bases. Cucerzan et al. [10] presented a large-scale named-entity disambiguation system that leverages information extracted from Wikipedia. Kataria et al. [13] developed an entity-disambiguation framework based on a hierarchical topic model. There is also research studying the privacy issue across multiple social networks. For example, Backstrom et al. [3] presented the processes where one can identify individuals in anonymized networks by either manipulating networks before they are anonymized or by having prior knowledge about certain anonymized nodes. Narayanan et al. [27] successfully deanonymized the interlinks between two real-world social networks based on only the network topology.

6. CONCLUSIONS

In this paper, we study the problem of multiple social network integration. We precisely define the problem, and propose a novel energy-based framework COSNET to address it. We develop an efficient model learning algorithm based on dual decomposition which can be easily parallelized. Our experimental results on two different genres of data sets validate the effectiveness and efficiency of the proposed framework. We further validate the effectiveness of our method by applying the integrated results to support expert finding, an important application.

Acknowledgements. We thank Shlomo Berkovsky, Terence Chen, and Dali Kaafar for sharing the linked accounts data in this research. The work is supported by the National High-tech R&D Program (No. 2014AA015103), National Basic Research Program of China (No. 2014CB340506, No. 2012CB316006), Natural Science Foundation of China (No. 61222212), NSF (CNS-1115234), National Social Science Foundation of China (No. 13&ZD190), the Tsinghua University Initiative

[6]https://aminer.org/profile/jiawei-han/53f42f36dabfaedce54dcd0c

Scientific Research Program (20121088096), a research fund supported by Huawei Inc., and Beijing key lab of networked multimedia.

7. REFERENCES

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] K. M. Anstreicher and L. A. Wolsey. Two "well-known" properties of subgradient optimization. *Mathematical Programming*, 120(1):213–220, 2009.

[3] 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.

[4] X. Bai, F. P. Junqueira, and S. H. Sengamedu. Exploiting user clicks for automatic seed set generation for entity matching. In *KDD'13*, pages 980–988, 2013.

[5] K. Bellare, S. Iyengar, A. G. Parameswaran, and V. Rastogi. Active sampling for entity matching. In *KDD'12*, pages 1131–1139, 2012.

[6] I. Bhattacharya and L. Getoor. Collective entity resolution in relational data. *ACM TKDD*, 1(1):1–36, March 2007.

[7] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR'2004*, pages 25–32, 2004.

[8] W. Chen, Z. Liu, X. Sun, and Y. Wang. A game-theoretic framework to identify overlapping communities in social networks. *DMKD*, 21(2):224–240, 2010.

[9] W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string metrics for matching names and records. In *IJCAI Workshop on IIWeb'03*, pages 73–78, 2003.

[10] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP-CoNLL'07*, volume 6, pages 708–716, 2007.

[11] Y. Cui, J. Pei, G. Tang, W.-S. Luk, D. Jiang, and M. Hua. Finding email correspondents in online social networks. *World Wide Web*, 16(2):195–218, 2013.

[12] R. Herbrich, T. Graepel, and K. Obermayer. *Large margin rank boundaries for ordinal regression*. MIT Press, Cambridge, MA, 2000.

[13] S. Kataria, K. S. Kumar, R. Rastogi, P. Sen, and S. H. Sengamedu. Entity disambiguation with hierarchical topic models. In *KDD'11*, pages 1037–1045, 2011.

[14] N. Komodakis. Efficient training for pairwise or higher order crfs via dual decomposition. In *CVPR'11*, pages 1841–1848, 2011.

[15] N. Komodakis, N. Paragios, and G. Tziritas. Mrf energy minimization and beyond via dual decomposition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2011.

[16] X. Kong, J. Zhang, and S. Y. Philip. Inferring anchor links across multiple heterogeneous social networks. In *CIKM'13*, pages 179–188, 2013.

[17] H. Kwak, C. Lee, H. Park, and S. B. Moon. What is twitter, a social network or a news media? In *WWW'10*, pages 591–600, 2010.

[18] S. Lacoste-Julien, K. Palla, A. Davies, G. Kasneci, T. Graepel, and Z. Ghahramani. Sigma: Simple greedy matching for aligning large knowledge bases. In *KDD'13*, pages 572–580, 2013.

[19] Y. LeCun, S. Chopra, and R. Hadsell. A tutorial on energy-based learning. *2006 CIAR Summer School: Neural Computation & Adaptive Perception*, 2006.

[20] J. Li, J. Tang, Y. Li, and Q. Luo. Rimom: A dynamic multi-strategy ontology alignment framework. *IEEE TKDE*, 21(8):1218–1232, 2009.

[21] Y. Li, C. Wang, F. Han, J. Han, D. Roth, and X. Yan. Mining evidences for named entity disambiguation. In *KDD'13*, pages 1070–1078, 2013.

[22] J. Liu, F. Zhang, X. Song, Y.-I. 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.

[23] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *SIGMOD'14*, pages 51–62, 2014.

[24] T. Lou and J. Tang. Mining structural hole spanners through information diffusion in social networks. In *WWW'13*, pages 837–848, 2013.

[25] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM'08*, pages 931–940, 2008.

[26] A. Maslow. A theory of human motivation. *Psychological Review*, 50(4):370–396, 1943.

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

[28] D. Perito, C. Castelluccia, M. A. Kaafar, and P. Manils. How unique and traceable are usernames? In *Privacy Enhancing Technologies*, pages 1–17, 2011.

[29] W. Shen, J. Wang, P. Luo, and M. Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *KDD'13*, pages 68–76, 2013.

[30] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen. Mapping users across networks by manifold alignment on hypergraph. In *AAAI'14*, pages 159–165, 2014.

[31] J. Tang, A. Fong, B. Wang, and J. Zhang. A unified probabilistic framework for name disambiguation in digital library. *IEEE TKDE*, 24(6):975–987, 2012.

[32] J. Tang, H. Gao, H. Liu, and A. D. Sarma. eTrust: Understanding trust evolution in an online world. In *KDD'12*, pages 253–261, 2012.

[33] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su. Arnetminer: Extraction and mining of academic social networks. In *KDD'08*, pages 990–998, 2008.

[34] W. Tang, J. Tang, T. Lei, C. Tan, B. Gao, and T. Li. On optimization of expertise matching with various constraints. *Neurocomputing*, 76(1):71–83, 2012.

[35] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. *NIPS'04*, 16, 2004.

[36] H. Whitney. Congruent graphs and the connectivity of graphs. *American Journal of Mathematics*, 54(1):150?168, 1932.

[37] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on twitter. In *WWW'11*, pages 705–714, 2011.

[38] L. Yartseva and M. Grossglauser. On the performance of percolation graph matching. In *COSN'13*, pages 119–130, 2013.

[39] R. Zafarani and H. Liu. Connecting corresponding identities across communities. In *ICWSM'09*, pages 354–357, 2009.

[40] R. Zafarani and H. Liu. Connecting users across social media sites: A behavioral-modeling approach. In *KDD'13*, pages 41–49, 2013.

[41] J. Zhang, J. Tang, and J. Li. Expert finding in a social network. In *DASFAA'07*, pages 1066–1069, 2007.