Mapping Users across Networks by Manifold Alignment on Hypergraph

Shulong Tan[1], Ziyu Guan[2], Deng Cai[3], Xuzhen Qin[4], Jiajun Bu[4] and Chun Chen[4]
{laos1984, welbyhebei}@gmail.com, dengcai@cad.zju.edu.cn, qinxuzhen@gmail.com, {bjj, chenc}@zju.edu.cn

[1] Department of Computer Science, University of California, Santa Barbara, CA 93106, USA
[2] College of Information and Technology, Northwest University of China, Xi'an 710127, China
[3] State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou 310027, China
[4] Zhejiang Key Laboratory of Service Robot, College of Computer Science, Zhejiang University, Hangzhou 310027,China

Abstract

Nowadays many people are members of multiple online social networks simultaneously, such as Facebook, Twitter and some other instant messaging circles. But these networks are usually isolated from each other. Mapping common users across these social networks will benefit many applications. Methods based on username comparison perform well on parts of users, however they can not work in the following situations: (a) users choose different usernames in different networks; (b) a unique username corresponds to different individuals. In this paper, we propose to utilize social structures to improve the mapping performance. Specifically, a novel subspace learning algorithm, Manifold Alignment on Hypergraph (MAH), is proposed. Different from traditional semi-supervised manifold alignment methods, we use hypergraph to model high-order relations here. For a target user in one network, the proposed algorithm ranks all users in the other network by their possibilities of being the corresponding user. Moreover, methods based on username comparison can be incorporated into our algorithm easily to further boost the mapping accuracy. Experimental results have demonstrated the effectiveness of our proposed algorithm in mapping users across networks.

Introduction

Online social networks have emerged as popular platforms for people to share ideas and carry out communications. Nowadays many of us are members of multiple online social networks in the same time, such as Facebook, Twitter and some other instant messaging circles. However, these networks are generally distributed in different servers and isolated from each other. That is, user mapping information is missing. Benefits of re-identifying users across social networks are multifold: (1) For individual users, it can enable them to keep up-to-date with their virtual contacts from different social networks in an integrated environment (Vosecky, Hong, and Shen 2009). (2) For site owners, it can help them to understand user migration patterns which are of great value to retain and increase site traffic (Kumar, Zafarani, and Liu 2011). (3) It can also help to detect criminals (e.g., cyberbullies and child pornographers) by analyzing user behaviors across networks. (4) Another benefit happens in transferring user interests across networks. Most of these works are based on user correspondences across networks (Carmagnola and Cena 2009; Cao, Liu, and Yang 2010; Zhong et al. 2012).

Mapping or re-identifying users across social networks, which is also known as social network de-anonymization (Narayanan and Shmatikov 2009), becomes a popular research topic recently (Vosecky, Hong, and Shen 2009; Carmagnola and Cena 2009; Zafarani and Liu 2009; Iofciu et al. 2011; Liu et al. 2013). Some papers demonstrate that comparing usernames in different networks is a workable and efficient way to map users across networks (Zafarani and Liu 2009; Iofciu et al. 2011). As reported, about 50% people use the same usernames in different social networks. However, these methods can not work well in all cases. (a) Firstly, many users choose different usernames in different networks, such as using real names in Facebook and using nicknames in Twitter. Reasons may lie in that users' preferred usernames had already been used by other people, or they intentionally used different usernames for online privacy considerations. The study in (Liu et al. 2013) shows that users keep 2-4 usernames in multiple social networks, on average. (b) Secondly, a unique username may correspond to different individuals. Common usernames are often owned by different natural persons (Liu et al. 2013; Bekkerman and McCallum 2005).

To improve the accuracy of mapping users across networks, more information should be exploited, such as structures of social networks. Previous work shows that structures of social networks are useful in user mapping (Narayanan and Shmatikov 2009). That is because the virtual friends of a natural person are usually similar groups of people in different social networks. In this paper, we utilize users' social structures in different networks to help user mapping.

In particular, to map users by exploiting social structures, partial user correspondences are needed beforehand (Narayanan and Shmatikov 2009). These correspondences can be known in different ways, such as by real-name verification or manual labels from users themselves. Then we consider the user mapping task as a potential manifold alignment problem across social structures (i.e., networks). The semi-supervised manifold alignment (Ham, Lee, and Saul 2005) based on traditional graphs is an intuitive choice.

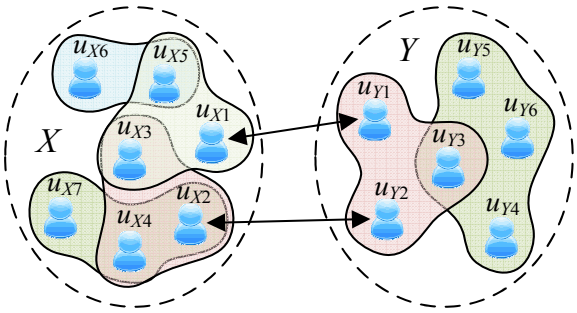


Figure 1: An example for mapping users across networks. In these two networks X and Y, we have two pairwise correspondences known beforehand. The target of user mapping is identifying other unknown user correspondences.

However, there are generally many high-order relations in online social networks, such as multiple users joining in the same interest group or multiple users participating in the same activity. Modeling these high-order relations by an ordinary graph will lose some information (Agarwal, Branson, and Belongie 2006; Zhou, Huang, and Schölkopf 2006; Bu et al. 2010; Tan et al. 2011). In this paper, we choose hypergraph to model social relations. A hypergraph is a generalization of the ordinary graph in which the edges, called *hyperedges*, are arbitrary non-empty subsets of the vertex set (Agarwal, Branson, and Belongie 2006; Chen, Wang, and Zhang 2007). Each vertex of the hypergraph corresponds to a user and the hyperedges are used to model various social relations among users, such as friendships and group memberships, as showed in Figure 1. By using the hypergraph model, we can accurately capture the high-order relations among users without loss of any information. Based on partial pairwise correspondences and social hypergraphs for each social network, we propose a new semi-supervised embedding framework, *Manifold Alignment on Hypergraph (MAH)*, to embed users into a common low dimensional space. Then user mapping can be inferred by comparing distances between users in the embedding space.

Our model outputs user ranking lists which can be easily incorporated into methods based on username comparison by the assumption: the collision rate is much lower in a local area (i.e., top rank positions) than the entire network. That is, in a local area, people often have different names to distinguish themselves from others. In this way, the accuracy of user mapping will be further boosted. Experimental results on real world data have demonstrated the effectiveness of our proposed algorithm in mapping users across networks. Besides, we conduct additional experiments on simulation datasets to investigate the model reliability and settings.

Mapping Users Across Networks

In this section we discuss how to map users across networks by Manifold Alignment on Hypergraph (MAH). We begin with the notation description and the problem definition.

Notation and Problem Definition

Let X and Y be two social networks across which we try to map users. For each network, we build a social hypergraph on users and construct hyperedges corresponding to social relations among users as shown in Figure 1. Taking the network X as an example, we build a social hypergraph $G^X(V^X, E^X, w)$, where V^X is the set of vertices corresponding to users in X, E^X is the set of hyperedges corresponding to social relations in X, and w is a weight function defined as $w : E^X \to \mathbb{R}$. Each hyperedge $e \in E^X$ is a subset of V^X. The degree of a hyperedge e is defined by $\delta(e) = |e|$, that is, the cardinality of e. The degree $d(v)$ of a vertex $v \in V^X$ is $d(v) = \sum_{e \in E^X | v \in e} w(e)$. We define a vertex-hyperedge incidence matrix $\mathbf{H}^X \in \mathbb{R}^{|V^X| \times |E^X|}$ whose entry $h(v, e)$ is 1 if $v \in e$ and 0 otherwise. Then we have: $d(v) = \sum_{e \in E^X} w(e)h(v, e)$ and $\delta(e) = \sum_{v \in V^X} h(v, e)$. Let $\mathbf{D}_e^X$ and $\mathbf{D}_v^X$ be two diagonal matrices consisting of hyperedge and vertex degrees, respectively. Let $\mathbf{W}^X$ be a diagonal matrix containing hyperedge weights. Similarly, we have another hypergraph $G^Y(V^Y, E^Y, w)$ for the network Y with corresponding matrices: $\mathbf{H}^Y$, $\mathbf{D}_e^Y$, $\mathbf{D}_v^Y$ and $\mathbf{W}^Y$.

We model the user mapping problem in a semi-supervised way. So some labeled user correspondences l are needed beforehand. These correspondences are indicated as: $\{u_{Xi}, u_{Yi}\}$, $i \in l$. This labeled user set is a subset of all *common users* between the two networks. Then all users are embedded into a common space based on social hypergraphs. Let $\mathbf{f}$ and $\mathbf{g}$ denote real-valued functions defined on V^X and V^Y respectively. They represent embedding coordinates of each users (i.e., vertices). The first l coordinates in $\mathbf{f}$ and $\mathbf{g}$ correspond to labeled users, which are denoted as $\mathbf{f}_l$ and $\mathbf{g}_l$. The relevance of every pair of users u_{Xi} and u_{Yj}, $rel(u_{Xi}, u_{Yj})$, can be computed by comparing their embedding coordinates, $\mathbf{f}_i$ and $\mathbf{g}_j$. Finally, user mapping can be done by ranking users' relevance. For example, for a user in network X, his/her corresponding user in network Y likely be the most relevance one.

Manifold Alignment on Hypergraph (MAH)

By MAH, users in the two networks can be mapped into a common embedding space. We force to map the two vertices corresponding to the same user (i.e., labeled pairwise correspondences) to the same point in the learned space. Such as the two users in the label $\{u_{X1}, u_{Y1}\}$ will be mapped to one point. In this way, we can fuse the two social hypergraphs together. Then we adopt a similar idea to hypergraph-based subspace learning (Zhou, Huang, and Schölkopf 2006), to learn the optimal space.

1-dimensional Space Learning Let k be the dimensionality of the learned space. We first consider the simplest case, $k = 1$. In this case, $\mathbf{f}$ and $\mathbf{g}$ are $|V^X| \times 1$ and $|V^Y| \times 1$ vectors, respectively. We force $\mathbf{f}_l = \mathbf{g}_l$ here, where the index l corresponds to the labeled user set as mentioned above. In order to express clearly, we use $\mathbf{t}_l$ instead of $\mathbf{f}_l$ and $\mathbf{g}_l$ below, $\mathbf{f}_l = \mathbf{g}_l = \mathbf{t}_l$. The cost function of $\mathbf{f}$ and $\mathbf{g}$ which should be

minimized is defined as follows:

$$C(\mathbf{f},\mathbf{g}) = \frac{1}{2}\sum_{i,j=1}^{|V^X|}\sum_{e\in E^X}\frac{1}{\delta(e)}\sum_{\{v_i,v_j\}\subseteq e} w(e)\left\|f_i - f_j\right\|^2 + \frac{1}{2}\sum_{i,j=1}^{|V^Y|}\sum_{e\in E^Y}\frac{1}{\delta(e)}\sum_{\{v_i,v_j\}\subseteq e} w(e)\left\|g_i - g_j\right\|^2, \quad s.t. \quad \mathbf{f}_l = \mathbf{g}_l = \mathbf{t}_l. \tag{1}$$

This object function is very intuitive. The two terms in the right-hand side are smoothness constraints. The target is, if two vertices are contained in many common hyperedges, then they should be embedded close to each other in the learned space. Users in each closely connected small community will be mapped to points which are near one another. If the two communities (belong to different networks) contain labeled correspondences, then the two communities will mapped close to each other in the learned space. In this way, unlabeled corresponding vertices are mapped to a local area, then user mapping will be more easier, such as by comparing usernames. Because the distinguishability of usernames in local area is much higher as mentioned in the Introduction section. The setting of $\mathbf{f}_l = \mathbf{g}_l$ corresponds to the $\mu \to \infty$ situation in (Ham, Lee, and Saul 2005). We tried to set μ as other values, but with no better results.

With simple algebraic transformations, the first term of the cost function can be rewritten as follows:

$$\begin{aligned}&\frac{1}{2}\sum_{i,j=1}^{|V^X|}\sum_{e\in E^X}\frac{w(e)h(v_i,e)h(v_j,e)}{\delta(e)}\left\|f_i - f_j\right\|^2\\ =&\sum_{i=1}^{|V^X|} f_i^2 d(v_i) - \sum_{i,j=1}^{|V^X|}\sum_{e\in E^X}\frac{f_i w(e)h(v_i,e)h(v_j,e)f_j}{\delta(e)}\\ =&\mathbf{f}^T\mathbf{D}_v^X\mathbf{f} - \mathbf{f}^T\mathbf{H}^X\mathbf{W}^X\mathbf{D}_e^{X-1}\mathbf{H}^{XT}\mathbf{f} = \mathbf{f}^T\mathbf{L}^X\mathbf{f},\end{aligned} \tag{2}$$

where $\mathbf{L}^X = \mathbf{D}_v^X - \mathbf{H}^X\mathbf{W}^X\mathbf{D}_e^{X-1}\mathbf{H}^{XT}$. Similarly, the second term can be rewritten as $\mathbf{g}^T\mathbf{L}^Y\mathbf{g}$, where $\mathbf{L}^Y = \mathbf{D}_v^Y - \mathbf{H}^Y\mathbf{W}^Y\mathbf{D}_e^{Y-1}\mathbf{H}^{YT}$.

Considering the constraint $\mathbf{f}_l = \mathbf{g}_l = \mathbf{t}_l$, the cost function can be further rewritten as:

$$\begin{aligned}C(\mathbf{f},\mathbf{g}) =& \mathbf{t}_l^T\mathbf{L}_{ll}^X\mathbf{t}_l + \mathbf{f}_m^T\mathbf{L}_{ml}^X\mathbf{t}_l + \mathbf{t}_l^T\mathbf{L}_{lm}^X\mathbf{f}_m + \mathbf{f}_m^T\mathbf{L}_{mm}^X\mathbf{f}_m\\ &+\mathbf{t}_l^T\mathbf{L}_{ll}^Y\mathbf{t}_l + \mathbf{g}_n^T\mathbf{L}_{nl}^Y\mathbf{t}_l + \mathbf{t}_l^T\mathbf{L}_{ln}^Y\mathbf{g}_n + \mathbf{g}_n^T\mathbf{L}_{nn}^Y\mathbf{g}_n = \mathbf{h}^T\mathbf{L}^Z\mathbf{h},\end{aligned}$$

where the indices m and n correspond to unlabeled user sets in networks X and Y respectively and $\mathbf{h} = [\mathbf{t}_l^T, \mathbf{f}_m^T, \mathbf{g}_n^T]^T$,

$$\mathbf{L}^Z = \left[\begin{array}{c|c|c} \mathbf{L}_{ll}^X + \mathbf{L}_{ll}^Y & \mathbf{L}_{lm}^X & \mathbf{L}_{ln}^Y \\ \hline \mathbf{L}_{ml}^X & \mathbf{L}_{mm}^X & 0 \\ \hline \mathbf{L}_{nl}^Y & 0 & \mathbf{L}_{nn}^Y \end{array}\right]. \tag{3}$$

In order to remove an arbitrary scaling factor in the embedding space, we minimize the Rayleigh quotient (Ham, Lee, and Saul 2005) as follows:

$$\min_{\mathbf{h}} \tilde{C}(\mathbf{h}) = \frac{\mathbf{h}^T\mathbf{L}^Z\mathbf{h}}{\mathbf{h}^T\mathbf{h}}, \quad s.t. \quad \mathbf{h}^T\mathbf{e} = 0, \tag{4}$$

where $\mathbf{e}$ is the vector with all elements equal to 1. $\mathbf{e}$ is the minimum solution of minimizing the cost function $C(\mathbf{f},\mathbf{g})$. But this solution projects all users onto one point and thus should be removed. So we impose the constraint $\mathbf{h}^T\mathbf{e} = 0$. The optimization problem of $\tilde{C}(\mathbf{h})$ can be solved by finding the eigenvector corresponding to the second smallest eigenvalue of $\mathbf{L}^Z$.

Alternatively, we can maximize the global variance in the learned space instead of maximizing $\mathbf{h}^T\mathbf{h}$ (Guan et al. 2010) and derive the normalized cost function as:

$$\min_{\mathbf{h}} \tilde{C}(\mathbf{h}) = \frac{\mathbf{h}^T\mathbf{L}^Z\mathbf{h}}{\mathbf{h}^T\mathbf{D}_v^Z\mathbf{h}}, \quad s.t. \quad \mathbf{h}^T\mathbf{e} = 0, \tag{5}$$

where where

$$\mathbf{D}_v^Z = \left[\begin{array}{c|c|c} \mathbf{D}_{v,ll}^X + \mathbf{D}_{v,ll}^Y & 0 & 0 \\ \hline 0 & \mathbf{D}_{v,mm}^X & 0 \\ \hline 0 & 0 & \mathbf{D}_{v,nn}^Y \end{array}\right]. \tag{6}$$

Likewise, the optimization problem of $\tilde{C}(\mathbf{h})$ can be solved by finding the generalized eigenvector corresponding to the second smallest eigenvalue of $(\mathbf{L}^Z, \mathbf{D}_v^Z)$.

Generalize to d-dimensional Space In practice, we need to learn a d-dimensional ($d > 1$) representation in order to better capture the relationships between users. To this end, we define a $|V^X| \times d$ matrix $\mathbf{F} = [\mathbf{f}_1\mathbf{f}_2...\mathbf{f}_d]$, a $|V^Y| \times d$ matrix $\mathbf{G} = [\mathbf{g}_1\mathbf{g}_2...\mathbf{g}_d]$ and a $(|V^X| + |V^Y| - l) \times d$ matrix $\mathbf{H} = [\mathbf{h}_1\mathbf{h}_2...\mathbf{h}_d]$ where vectors $\mathbf{f}_i$, $\mathbf{g}_i$ and $\mathbf{h}_i$ contain users' coordinates on the ith dimension. For each dimension $i \in \{1, 2, ..., d\}$, we need to minimize $\tilde{C}(\mathbf{h}_i)$. Therefore, the overall cost function is:

$$\tilde{C} = \frac{\sum_{i=1}^d \mathbf{h}_i^T\mathbf{L}^Z\mathbf{h}_i}{\sum_{i=1}^d \mathbf{h}_i^T\mathbf{h}_i} = \frac{\sum_{i=1}^d (\mathbf{H}^T\mathbf{L}^Z\mathbf{H})_{ii}}{\sum_{i=1}^d (\mathbf{H}^T\mathbf{H})_{ii}} = \frac{tr(\mathbf{H}^T\mathbf{L}^Z\mathbf{H})}{tr(\mathbf{H}^T\mathbf{H})},$$

where $tr(\cdot)$ denotes the trace of a matrix. The solution for this function is similar with the 1-dimensional version.

Mapping Users using MAH

After the embedding space is learned, we can use this common low dimensional embedding space to address the following matching problems: What is the most relevant user u_{Yi} that corresponds to a u_{Xj}? or the most relevant user u_{Xi} that corresponds to a u_{Yj}? (Ham, Lee, and Saul 2005)

We use the cosine similarity to measure the relevance between two users:

$$rel(u_{Xi}, u_{Yj}) = \frac{\sum_{v=1}^d \mathbf{F}_{iv} \times \mathbf{G}_{jv}}{\sqrt{\sum_{v=1}^d \mathbf{F}_{iv}^2} \times \sqrt{\sum_{v=1}^d \mathbf{G}_{jv}^2}}. \tag{7}$$

Then we can rank unlabeled users in network X (or Y) corresponding to a target user in network Y (or X) to find his/her most likely corresponding user.

Note that, in real world social network data, the number of users may be huge (e.g., more than one million). So the eigenvector computation for all users together will be very consuming. To speed up, we can split the user set into small groups by some user features, such as location information. Then user mapping can be done in relatively small groups, such as users from Los Angeles in each social network.

Incorporating Usernames with MAH

User name comparison methods can be incorporated into our proposed model, to further boost user mapping accuracy. In the common sense, names play the role of identification in the local area (e.g., in close social circles or in specific locations). That is, in a local area, people generally have different names to distinguish themselves from others. Based on this assumption, we incorporate user names into MAH by a multi-pass sieve method as follows.

1. For a target user, we scan his/her ranking user list (in the corresponding network) from the top one to the first half position to find the user with the same username. Then we map them directly.
2. If the corresponding user is not found in the first step, we then comparing usernames by adding or deleting prefixes and suffixes. For example, for user u_{Xi}, we scan u_{Xi}'s ranking list from the top one to the first 25% position to find user u_{Yj} satisfying the following condition: the username of u_{Xi} is a substring of u_{Yj}'s username, or the username of u_{Yj} is a substring of u_{Xi}'s user name. If u_{Yj} is found, we map u_{Xi} and u_{Yj} correspondingly.
3. If the corresponding user is not found in above steps, we scan the target user's ranking list from the top one to the first 2% position to find a user satisfying that: the *Levenshtein distance* (i.e., *edit distance*) between the found user's username and the target user's username is less than 4. Then the found user will be considered as the corresponding user of the target user.
4. If the corresponding user is still not found, we consider the user in top one position as the corresponding user.

The different kinds of username similarities are used as constraints here. Higher precision constraints (e.g., with the same username) are used in earlier steps to seek more accurate mapping results. Note that the second step above is more effective than *Levenshtein distance* since users often add prefixes or suffixes on their basic usernames to get new usernames (Narayanan and Shmatikov 2009). Based on simple steps above, we combine usernames with MAH. In practical applications, more kinds of constraints based on usernames can be combined.

Experiments

In this section, we investigate the use of our proposed algorithm for user mapping. We first represent the experimental results on a real world dataset. Then we investigate the model reliability and parameter settings on simulation data.

Compared Algorithms

We design two competitive approaches as baselines in experiments. The first baseline is based on the idea of k nearest neighbors finding (**KNN**). KNN based on social structures is presented as follows: For each unlabeled user in network X (or Y), we first find his/her k nearest labeled neighbors and set pairwise weights (between the target user and a labeled neighbor) as the ratio of the number of common relations between the two users to the number of all common relations between the target user and his/her labeled neighbors. Similarly, we then find k nearest unlabeled neighbors in network Y (or X) for each labeled user and compute every pairwise weight similarly. In this way, we can fuse two networks. Formally, for predicting corresponding users from Y to X, we have two weight matrices $\mathbf{W}_{XX}$ in $(|UserX| - l) \times l$ and $\mathbf{W}_{XY}$ in $l \times (|UserY| - l)$. Then the relevance between every unlabeled user in Y and each unlabeled user in X can be computed by $\mathbf{W}_{XX} \times \mathbf{W}_{XY}$. The situation of predicting corresponding users from X to Y is symmetrical, but using corresponding weight matrices $\mathbf{W}_{YY}$ and $\mathbf{W}_{YX}$. We tune the number of nearest neighbors k to achieve the best performance, in our each experiment.

The second baseline is manifold alignment on traditional graphs (**MAG**). We build a *social graph* for each network by computing user-user pairwise weights as: $weight^S(u_i, u_j) = (|R_{u_i} \cap R_{u_j}|)/(|R_{u_i}| + |R_{u_j}|)$, where R_{u_i} and R_{u_j} are relation sets containing u_i and u_j respectively. Then we use the semi-supervised manifold alignment method to rank users. This method corresponds to the situation of $\mu \to \infty$ in (Ham, Lee, and Saul 2005).

Evaluation Metrics

Taking into account different structures of different social networks in practice, we define a metric to measure the pairwise characteristic across two networks: $Interoperability$ (abbreviated as $Interop$). $Interop$ measures the probability: if two users are closely related in one network and whether they are closely related in another network.

$$Interop(X, Y) = \frac{|Correlations| * 2}{|RelationsX| + |RelationsY|},$$

where $RelationsX$ is the set of direct pairwise connections (e.g., the two users are friends or in the same group) of users in network X and $RelationsY$ is that for network Y. $Correlations$ is the intersection of $RelationsX$ and $RelationsY$. Clearly, $0 \leq Interop(X, Y) \leq 1$.

For the evaluation of user mapping, we use $Precision@t$ as the metric, which is defined as follows:

$$Precision@t = \frac{|CorrUserX@t| + |CorrUserY@t|}{|UnlabeledCommonUsers| * 2},$$

where $|CorrUserX@t|$ is the number of unlabeled users in X for whom the method can find their corresponding users correctly in Y at top t ranks and $|CorrUserY@t|$ is similar but for unlabeled users in Y. $|UnlabeledCommonUsers|$ is the number of all unlabeled common users.

The Dataset and Experimental Results

In this section, we test our model on a real world dataset, Twitter-BlogCatalog dataset. To get data with labeled user correspondences, we develop a data crawler to get data from BlogCatalog [1]. BlogCatalog not only provides directories of blogs, but also provides an attribute called "My Communities" for each user. This attribute enables users to list their corresponding linkages to other online social networks, via

[1] http://www.blogcatalog.com

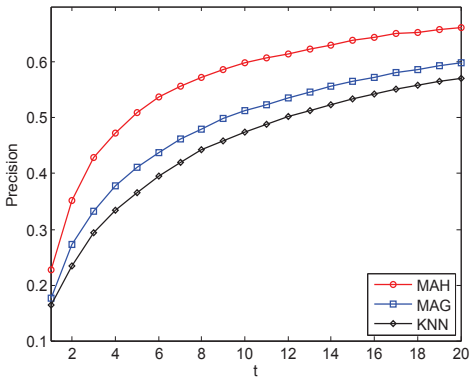


Figure 2: $Precision@t$ on Twitter-BlogCatalog dataset.

which we can get labeled user correspondences across social networks. Specifically, we choose Twitter as a corresponding network, and try to map users across BlogCatalog and Twitter. Note that social networks of these two communities are both directed (i.e., following/followed relations). In our data, 21577 BlogCatalog users also give their Twitter usernames and linkages. We further download all social relations (i.e., following/followed friends) for these users from the two networks. To get our final data for experiments, we filter users by ensuring that each user in the final data has at least 4 friends (following friends or followed friends) in each of the two networks. The final user numbers in the two networks are both 2710. The $Interop$ of this dataset is 0.3895.

In building social hypergraphs, we first construct one hyperedge for each directed social relation (followed or following). Weights for this kind of hyperedges are set to 1. Then we build hyperedges that contains all followers for a corresponding target user. For example, if u_1, u_2 and u_3 are followers of u_4, we build a hyperedge containing u_1, u_2 and u_3. The motivation is that the followers of the same target user would probably be in the same "group" or "circle". For example, if the target user (the central one) is a celebrity, such as Lady Gaga, we may infer that all his/her followers have similar tastes. It is just like that they join the same interest group. If the target user is a common user, his/her followers may also know each other and they may belong to the same circle. Weights for this kind of hyperedges are empirically set to 0.1 since they are not so reliable as true social relations. In real applications, the hypergraph construction method may vary depending on the data content. For example, if we have the high-order interest group information between users, we can model it by hyperedges directly.

Experimental results are shown in Figure 2. This is the situation by using 30% of user correspondences as training data. As can be seen, our method MAH performs better than baselines in all cases of $Precision@t, 1 \leq t \leq 20$. That is because MAH can make full use of high-order relations.

For the experiment of incorporating usernames with MAH, we design two simple baselines: (a) **ExactMatch** maps users with the same usernames (only workable for parts of users) and randomly chooses corresponding users for others. (b) **NameSimi** has similar four steps as our method but does not consider rank positions in each step (i.e., scanning the whole user set in a random order). In the fourth step, it maps users randomly as ExactMatch. We denote our method as **MAH-name** here.

We use $Precision$ (the same with $Precision@1$) as the evaluation metric in this experiment. Table 1 shows the final mapping accuracy of each method. We vary the proportion of training data in this test. The training data does not make sense for baselines. It only means that they work on different user sets. As can be seen, ExactMatch works badly, since a unique username may correspond to different natural persons and there are only a portion of users use the same usernames in different networks. NameSimi works much worse than our method too. The reasons mainly lie in: (a) without considering rank positions, there are too many error mapping in the first three steps; (b) in the fourth step, our method certainly works better than NameSimi, by the ranking result. The performance of our method depends on the training data. However, we find that it can still work satisfactorily with little training data. With increasing training data (up to 50%), our method can get a mapping precision about 85%.

Table 1: Results after Incorporating User Names.

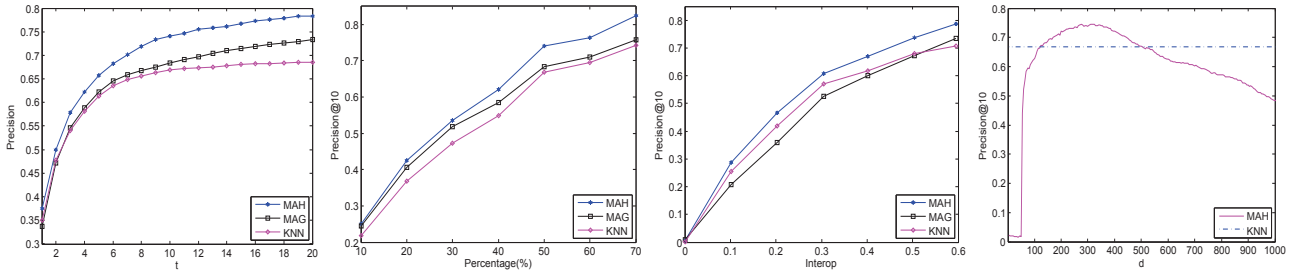
% of Training	20%	30%	40%	50%
ExactMatch	0.5298	0.5240	0.5212	0.5258
NameSimi	0.6654	0.6602	0.6625	0.6665
MAH-name	**0.7319**	**0.7668**	**0.8185**	**0.8530**

Experiments on Simulation Data

To investigate the reliability of our method on different social structures (in $Interop$) and parameter settings, we produce a simulation dataset as follows: we divide one original network into two simulation networks via which we can get user correspondences easily. Specifically, we use data from DBLP (Deng et al. 2011) to construct the original network and consider coauthor relations as social relations. Pairwise simulation networks can be got by dividing all papers (i.e., relations) into two parts randomly. A user can be found from the two parts of data as different papers' author. In this way, we can get all user correspondences across the two simulation networks. Parts of these correspondences can be used for space learning and others would be used for testing. The numbers of users in each simulation network are 2317 and 2327 respectively. The $Interop$ of this dataset is 0.5275. The mapping performance for this dataset is shown in Figure 3 (a). Similar with experiments on Twitter-BlogCatalog data, our method performs better than baselines.

Then we explore the performance for different proportions of training data. In this experiment, we vary the training proportion from 10% to 70% and Figure 3 (b) shows the result ($Precision@10$). As can be seen, more training data clearly benefit user mapping. This is consistent with the common sense. Besides, with more training data, the superiority of MAH is more noticeable.

We also explore the quantified effect of the pairwise char-



(a) (b) (c) (d)

Figure 3: Experimental results on simulation data. (a) $Precision@t$ for the comparing experiment; (b)$Precision@10$ for different proportions of training data; (c) $Precision@10$ for varying $Interop$; (d) $Precision@10$ when varying d.

acteristic $Interop$. To get variants of simulation datasets in different $Interop$ values, we process the simulation dataset above by adding or deleting "Author-Paper" relations. $Interop$ varies from 0 to 0.6. Figure 3 (c) shows $Precision@10$ on these processed simulation datasets. As can be seen, the performance of all three methods strongly depends on the $Interop$ value. It can be concluded that if social structures of the two networks are very different (i.e., with lower $Interop$), it is difficult to map users across social networks only based on social structures.

The important parameter of MAH is the dimensionality d of the learned space. We explore the influence of d on the simulation dateset using $Precision@10$ as the evaluation metric. We also use KNN as the reference in this experiment. The results are showed in Figure 3 (d). When d increases, the precision of MAH increases rapidly. The best value of d is around 350. We find that the best value of d is related to the total number of eigenvectors learned by our method (i.e.,$|V^X|+|V^Y|-l$), based on experiments on simulation datasets of different sizes. Specifically, the best value of d is about 10% of the number of eigenvectors. So we set $d = (|V^X| + |V^Y| - l)/10$ for all other experiments.

Related Work

There has been some research works conducted on the task of user mapping across social networks. Most of them are based on user profile information. Zafarani and Liu propose a simple method based on seven hypotheses. For example, a user's profile page usually contains another username which is used in other social network by the same individual (Zafarani and Liu 2009). Carmagnola and Cena introduce an method based on some heuristics to utilize multiple types of profile attributes, such as username, location and email address (Carmagnola and Cena 2009). Based on similar user profile information, some other papers build profile vectors for each user in different networks (Vosecky, Hong, and Shen 2009; Nunes, Calado, and Martins 2012; Malhotra et al. 2012). They treat each profile field (e.g., location) as a dimension in the profile vector. Both supervised (Nunes, Calado, and Martins 2012; Malhotra et al. 2012) and unsupervised (Vosecky, Hong, and Shen 2009) methods can be applied based on these profile vectors. Besides, Iofciu et al. try to map users across social tagging systems by linearly combining similarities of usernames and user tag lists (Iofciu et al. 2011). This method is dependent on specific types of social networks and not as general as our model. Mapping users only based on user profiles are unreliable, because user profiles in different networks may be heterogeneous, partly missing or with false information. (Labitzke, Taranu, and Hartenstein 2011) shows that users tend to publish different pieces of information in different social networks. So in this paper we propose to exploit social structures to improve the performance. As mentioned above, user profiles, such as usernames, can be integrated with our model easily.

From a different perspective, some researchers follow the user mapping problem for data security and privacy considerations (Frankowski et al. 2006; Backstrom, Dwork, and Kleinberg 2007; Narayanan and Shmatikov 2008; 2009; 2010; Labitzke, Taranu, and Hartenstein 2011). Most of these papers focus on problems: whether anonymized social networks are safe in protecting users' privacy information? and whether the public information in anonymized systems is enough to do de-anonymization? (Narayanan and Shmatikov 2008; 2009) find that anonymized networks can be re-identified by only social structures. Our model also utilize social structures to map users. The difference lies in: the purpose of methods in above papers is studying whether de-anonymization is practicable. So they only need to re-identify a part of users as evidence. However, we aim at mapping all common users across social networks.

Conclusions

In this paper, we try to map common users across social networks. To address this problem, we propose a semi-supervised learning framework to infer the corresponding user in other network for each user in the target network. Specifically, we first build a social hypergraph for each network and then carry out semi-supervised manifold alignment (i.e., Manifold Alignment on Hypergraph, MAH) on social hypergraphs. A low-dimensional common space for all users can be learned. Then the user mapping task can be done by comparing the user relevance. Moreover, methods based on username comparison can be integrated with our algorithm easily to further boost the mapping accuracy. The experimental results show that our model is effective in user mapping across social networks.

Acknowledgments

This work is partially supported by the National Basic Research Program of China (973 Program) under Grant No. 2013CB336500, the Institute for Collaborative Biotechnologies through grant W911NF-09-0001 from the U.S. Army Research Office, National Natural Science Foundation of China (Grant No: 61373118, 61173186, 61222207). The content of the information does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

Agarwal, S.; Branson, K.; and Belongie, S. 2006. Higher order learning with graphs. In *Proc. the 23rd International Conference on Machine Learning*, 17–24.

Backstrom, L.; Dwork, C.; and Kleinberg, J. 2007. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proc. the 16th international conference on World Wide Web*, 181–190.

Bekkerman, R., and McCallum, A. 2005. Disambiguating web appearances of people in a social network. In *Proc. the 14th international conference on World Wide Web*, 463–470.

Bu, J.; Tan, S.; Chen, C.; Wang, C.; Wu, H.; Zhang, L.; and He, X. 2010. Music recommendation by unified hypergraph: combining social media information and music content. In *Proc. the 18th ACM International Conference on Multimedia*, 391–400.

Cao, B.; Liu, N. N.; and Yang, Q. 2010. Transfer learning for collective link prediction in multiple heterogenous domains. In *Proc. the 27th International Conference on Machine Learning*, 159–166.

Carmagnola, F., and Cena, F. 2009. User identification for cross-system personalisation. *Information Sciences* 179(1):16–32.

Chen, S.; Wang, F.; and Zhang, C. 2007. Simultaneous heterogeneous data clustering based on higher order relationships. In *Proc. the 7th IEEE International Conference on Data Mining Workshops*, 387–392.

Deng, H.; Han, J.; Zhao, B.; Yu, Y.; and Lin, C. X. 2011. Probabilistic topic models with biased propagation on heterogeneous information networks. In *Proc. the 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, 1271–1279.

Frankowski, D.; Cosley, D.; Sen, S.; Terveen, L.; and Riedl, J. 2006. You are what you say: privacy risks of public mentions. In *Proc. the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 565–572.

Guan, Z.; Wang, C.; Bu, J.; Chen, C.; Yang, K.; Cai, D.; and He, X. 2010. Document recommendation in social tagging services. In *Proc. the 19th international conference on World Wide Web*, 391–400.

Ham, J.; Lee, D. D.; and Saul, L. K. 2005. Semisupervised alignment of manifolds. In *Proc. of the 21st Annual Conference on Uncertainty in Artificial Intelligence*, 120–127.

Iofciu, T.; Fankhauser, P.; Abel, F.; and Bischoff, K. 2011. Identifying users across social tagging systems. In *Proc. the 5th International Conference on Weblogs and Social Media (ICWSM 2011)*, 522–525.

Kumar, S.; Zafarani, R.; and Liu, H. 2011. Understanding user migration patterns in social media. In *Proc. the 25th Conference on Artificial Intelligence (AAAI)*, 1204–1209.

Labitzke, S.; Taranu, I.; and Hartenstein, H. 2011. What your friends tell others about you: Low cost linkability of social network profiles. In *Proc. the 5th SNA-KDD Workshop 11 (SNA-KDD11)*.

Liu, J.; Zhang, F.; Song, X.; Song, Y.-I.; Lin, C.-Y.; and Hon, H.-W. 2013. What's in a name?: an unsupervised approach to link users across communities. In *Proc. the sixth ACM international conference on Web Search and Data Mining*, 495–504.

Malhotra, A.; Totti, L.; Meira Jr, W.; Kumaraguru, P.; and Almeida, V. 2012. Studying user footprints in different online social networks. In *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'12)*, 1065–1070.

Narayanan, A., and Shmatikov, V. 2008. Robust deanonymization of large sparse datasets. In *Proc. 29th IEEE Symposium on Security and Privacy*, 111–125.

Narayanan, A., and Shmatikov, V. 2009. De-anonymizing social networks. In *Proc. 30th IEEE Symposium on Security and Privacy*, 173–187.

Narayanan, A., and Shmatikov, V. 2010. Myths and fallacies of personally identifiable information. *Communications of the ACM* 53(6):24–26.

Nunes, A.; Calado, P.; and Martins, B. 2012. Resolving user identities over social networks through supervised learning and rich similarity features. In *Proc. the 27th Annual ACM Symposium on Applied Computing*, 728–729.

Tan, S.; Bu, J.; Chen, C.; Xu, B.; Wang, C.; and He, X. 2011. Using rich social media information for music recommendation via hypergraph model. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* 7S(1):22.

Vosecky, J.; Hong, D.; and Shen, V. Y. 2009. User identification across multiple social networks. In *Proc. the 1st International Conference on Networked Digital Technologies, NDT'09*, 360–365.

Zafarani, R., and Liu, H. 2009. Connecting corresponding identities across communities. In *Proc. the 3rd International Conference on Weblogs and Social Media (ICWSM09)*, 354–357.

Zhong, E.; Fan, W.; Wang, J.; Xiao, L.; and Li, Y. 2012. Comsoc: adaptive transfer of user behaviors over composite social network. In *Proc. the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 696–704.

Zhou, D.; Huang, J.; and Schölkopf, B. 2006. Learning with hypergraphs: Clustering, classification, and embedding. In *Advances in Neural Information Processing Systems 19*.