Learning to Map Social Network Users by Unified Manifold Alignment on Hypergraph

Wei Zhao, Shulong Tan, Ziyu Guan, Boxuan Zhang, Maoguo Gong, *Senior Member, IEEE*, Zhengwen Cao, and Quan Wang

***Abstract*— Nowadays, a lot of people possess accounts on multiple online social networks, e.g., Facebook and Twitter. These networks are overlapped, but the correspondences between their users are not explicitly given. Mapping common users across these social networks will be beneficial for applications such as cross-network recommendation. In recent years, a lot of mapping algorithms have been proposed which exploited social and/or profile relations between users from different networks. However, there is still a lack of unified mapping framework which can well exploit high-order relational information in both social structures and profiles. In this paper, we propose a unified hypergraph learning framework named unified manifold alignment on hypergraph (UMAH) for this task. UMAH models social structures and user profile relations in a unified hypergraph where the relative weights of profile hyperedges are determined automatically. Given a set of training user correspondences, a common subspace is learned by preserving the hypergraph structure as well as the correspondence relations of labeled users. UMAH intrinsically performs semisupervised manifold alignment with profile information for calibration. For a target user in one network, UMAH ranks all the users in the other network by their probabilities of being the corresponding user (measured by similarity in the subspace). In experiments, we evaluate UMAH on three real world data sets and compare it to state-of-art baseline methods. Experimental results have demonstrated the effectiveness of UMAH in mapping users across networks.**

***Index Terms*— Hypergraph, manifold alignment, social networks, user mapping.**

Manuscript received July 26, 2017; revised October 25, 2017 and February 14, 2018; accepted February 28, 2018. Date of publication April 3, 2018; date of current version November 16, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61672409, Grant 61522206, Grant 61373118, and Grant 61422209, in part by the Major Basic Research Project of Shaanxi Province under Grant 2017ZDJC-31, in part by the Science and Technology Plan Program in Shaanxi Province of China under Grant 2017KJXX-80, and in part by the Fundamental Research Funds for the Central Universities under Grant 20101164837. *(Corresponding author: Ziyu Guan.)*

W. Zhao is with the State Key Laboratory of Integrated Services Networks, School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: ywzhao@mail.xidian.edu.cn).

S. Tan is with Baidu Big Data Lab, Sunnyvale, CA 94089 USA (e-mail: laos1984@gmail.com).

Z. Guan is with the State Key Laboratory of Integrated Services Networks, School of Computer Science and Technology, Xidian University, Xi'an 710071, China, and also with the School of Information and Technology, Northwest University, Xi'an 710127, China (e-mail: ziyuguan@nwu.edu.cn).

B. Zhang and Q. Wang are with the School of Computer Science and Technology, Xidian University, Xi'an 710071, China (e-mail: bxzhang_1@stu.xidian.edu.cn; qwang@xidian.edu.cn).

M. Gong is with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China (e-mail: gong@ieee.org).

Z. Cao is with the School of Information and Technology, Northwest University, Xi'an 710127, China (e-mail: caozhw@nwu.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2812888

I. Introduction

ONLINE social networks have become a popular platform for people to share ideas and communicate. People often have accounts in multiple online social networks, such as BlogCatalog[1], Facebook,[2] Twitter,[3], and among others. However, these networks are maintained by different companies and isolated from one another. That is, user mapping information is missing. Reidentifying users across social networks may be beneficial for many applications. It can enable users to keep up to date with their virtual contacts from different social networks in an integrated environment [1]. It can also help Website owners understand the patterns of user migration which are valuable for retaining and increasing traffic to the Website [2]. Another example happens in transferring user interests across domains or networks. Most of these works are based on user correspondences across networks [3]–[5].

Mapping users across social networks, which is also known as social network deanonymization [6], has attracted researchers' attention recently [1], [3], [7]–[13]. Early work on this topic has explored comparing usernames across different networks and demonstrated that it is a workable and efficient way to map users [7], [8]. However, usernames are not always reliable since a real person would use different usernames in different Websites, and the same username can correspond to different people in different networks [9], [14]. To improve the accuracy of user mapping across networks, more information should be exploited, such as structures of social networks and user profiles (e.g., users' demographic features). Previous work showed that structures of social networks were useful in user mapping [6], [10]. That is because the virtual friends of a natural person are usually a similar group of people in different social networks. User profile is another type of useful information in user mapping [1], [12]. Profile features can help identify the same person since they tend to exhibit similar values in different networks for the same person. For example, to map a user from California in one network, users with the feature "California, USA" in the other network may be more likely to be the corresponding user than others.

It has been shown that the mapping performance based on either social structures or user profiles alone is not satisfactorily accurate [1], [6]. On the one hand, social structures are more or less different from network to network. The social

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

[2]http://www.facebook.com

[3]http://twitter.com

circles of a specific user may vary in different social networks. For example, there may be more real life friends in one's Facebook circle than in his/her Twitter circle. Two users who are friends or join in the same group in one network do not necessarily have social relations in other networks. In one word, the interoperability between social networks should be considered before mapping users by social structures [10]. On the other hand, user profiles in different networks may be heterogeneous, partly missing or with false information. Labitzke *et al.* [15] showed that users tended to publish different pieces of information in different social networks. Hence, it is necessary to combine both types of information in order to boost the mapping performance. Recently, researchers have proposed hybrid methods in this direction [11], [13].

However, previous hybrid methods still have limitations in modeling social and profile relations. First, previous hybrid methods only handled pairwise relations, while the social and profile relations could be high order. For example, social groups or circles are high-order relations involving multiple users; a profile feature value also connects multiple users. Modeling these high-order relations by pairwise edges of ordinary graphs will lose partial information [16], e.g., how many users are involved in a high-order relation, which could be useful for assessing relation importance. Second, different profile features could be of different degrees of usefulness for mapping users, while some traditional hybrid methods [13] treat them equally. For example, users with the same birthday are more likely to be the same person, compared to users with the same gender.

In this paper, we propose a unified hypergraph learning framework named unified manifold alignment on hypergraph (UMAH) for user mapping across social networks. UMAH models social structures and user profile relations in a unified hypergraph. A hypergraph is a generalization of ordinary graph in which the edges, called *hyperedges*, are arbitrary nonempty subsets of the vertex set [17]. Each vertex of the hypergraph corresponds to a user and each hyperedge represents a social or profile relation. By using the hypergraph model, we can accurately capture the high-order relations among users without loss of information. Fig. 1 shows a toy example of the constructed hypergraph, where regions with different colors represent social hyperedges, and rectangles connecting to users represent profile hyperedges. Here, we show only the location feature of users. Each feature value corresponds to a hyperedge. The basic idea of UMAH is to perform manifold alignment [10], [18] according to the social structures of the two networks and use profile relations for calibration. In order to discover the mapping of users between two social networks according to social structures, some training correspondence pairs are needed [6]. These correspondences can be known in different ways, such as by real-name verification or manual labeling. Given a set of training user correspondences, a low-dimensional embedding space is learned by preserving the hypergraph structure as well as the correspondence relations of labeled users. In UMAH, the relative weights of different profile features are learned automatically according to their discriminative power assessed on labeled users. By discriminative power, we mean a good

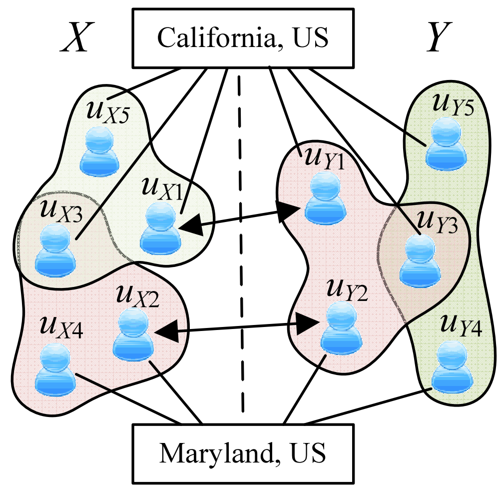


Fig. 1. Example for mapping users across networks. Two types of information are shown: social structures and user features. $\{u_{X1}, u_{Y1}\}$ and $\{u_{X2}, u_{Y2}\}$ are two pairwise correspondences known beforehand.

feature would take the same value for user accounts belonging to the same person while for accounts of different people it tends to take different values. Once the optimal space is learned, for a target user in one network UMAH ranks all the users in the other network by their probabilities of being the corresponding user (measured by cosine similarity in the embedding space).

In [10], a similar hypergraph-based alignment framework called manifold alignment on hypergraph (MAH) was proposed for user mapping across social networks. The key difference between UMAH and MAH is that UMAH models both social and profile relations, while MAH can only generate mapping based on social structures. In summary, main contributions of this paper are as follows.

1) We propose a novel learning framework UMAH to map common users across social networks. This framework can well exploit high-order relations from both social structures and user profiles in each network.
2) We develop a mechanism to learn hyperedge weights automatically.
3) We systematically test UMAH on three real world data sets. Experimental results show that UMAH is effective and outperforms state-of-art baseline methods.

II. Related Work

A. User Mapping Across Social Networks

There have been some research works conducted on the task of user mapping across social networks. Most of them were based on user profile information (usernames are generally treated as one type of user profile information in previous papers). Zafarani and Liu [7] proposed a simple method based on seven hypotheses. For example, a user's profile page usually contains another username which is used in other social networks by the same individual. Carmagnola and Cena [3] introduced a method based on some heuristics to utilize multiple types of profile features, such as username, location, and e-mail address. Based on similar user profile information, some other papers built profile vectors for each user in different networks [1], [19], [20]. They treated each profile feature (e.g., location) as a dimension in the profile vector. Both supervised [19], [20] and unsupervised [1] methods can be applied based on these profile vectors. Iofciu *et al.* [8] tried to map users across social tagging systems by linearly combining similarities of usernames and user tag lists . Their method is ad hoc for specific types of social networks. Our model also utilizes user profile information, but it is general and exploits this kind of information by hypergraphs, rather than profile vectors or heuristics. Hypergraphs can retain high-order profile similarity information.

Very recently, Mu *et al.* [12] proposed a latent space modeling method for user mapping. The main idea was to learn projections from different networks' feature spaces to a common subspace by constraining corresponding users to be nearer than noncorresponding users in the subspace. Although it is a general method, it does not make use of social structures.

Recently, researchers have also explored using social structures for user mapping. Tan *et al.* [10] proposed a hypergraph-based alignment framework for this task. It stemmed from manifold learning and therefore can capture both local and global structures from the data [18], [21]–[23]. However, it did not make use of user profile information. Zhang *et al.* [11] developed an energy-based model called COSNET for mapping users across multiple networks. COSNET exploited both profile features and social structures, and also emphasized global consistency among multiple networks. Nevertheless, COSNET exploited social relations fragmentarily with candidate matching pairs. It failed to capture the global structures of networks. Moreover: 1) the candidate generation process might result in confusing information, e.g., most candidates in neighborhoods of a matching pair are nonmatching pairs, thus contradicting the "neighborhood-preserving matching" principle [11] and 2) the pruning process may discard true positive pairs. Another hybrid method for user mapping was proposed by Zhang and Tong [13]. The method propagated alignment scores from supervised pairs to unsupervised ones via a consistency principle defined according to social relations and attributes. However, it treated all attributes (and social relations) equally, without customizing their importance. Furthermore, it used only one indicator (or one similarity score) to encode all attributes, so it cannot naturally handle multiple types of attributes. One common drawback of the above two hybrid methods is that they cannot capture high-order relations.

B. Deanonymizing Social Networks

From a different point of view, some researchers considered user mapping issues in the contexts of data security and privacy [6], [15], [24]–[27]. Most of these works were focused on problems such as, whether the anonymized social networks are safe to protect user privacy and whether the public information in anonymous systems is sufficient to do deanonymization (i.e., user mapping). Narayanan and Shmatikov [6], [26] found that anonymized networks can be reidentified by only social structures. Our model also utilizes social structures to map users. The difference lies in that the purpose of the methods above is to study whether deanonymization is practicable. Hence, they only need to reidentify a part of users as evidence. However, we aim at mapping all common users across social networks.

C. Applications on Composite Social Networks

At the same time, some researchers introduced some applications across social networks. They considered multiple social networks as a composite social network and proposed methods for personalized recommendation based on it. They overpast the user mapping problem and only used networks with user correspondences known in advance. In this line of research, Pan *et al.* [28] collected a composite social network by built-in sensors in smart phones. It contained multiple networks, such as Bluetooth proximity network and call log network. Then, they used this composite social network to predict mobile application (i.e., app) installation. Similarly, Zhong *et al.* [5] tried to transfer knowledge from a composite social network for user behavior prediction. They used networks from the same Website or the same company to build the composite social network, so user correspondences can be obtained easily.

D. Manifold Alignment and Hypergraph

The method used in this paper is based on the idea of graph-based subspace learning, manifold alignment, and hypergraph learning.

The basic idea of graph-based subspace learning [29]–[34] is to construct an affinity graph as an approximation to the underlying data manifold and then learn a low-dimensional space for the data by preserving the affinity graph structure. Typical algorithms include Laplacian eigenmap [30] and locally linear embedding [29], [33].

Ham *et al.* [18] studied a semisupervised learning algorithm for aligning different data sets characterized by the same underlying manifold. The partial known aligned points of the data sets can be given by coordinates or labeled pairwise correspondences [18], [35]. Xiong *et al.* [36] proposed a manifold alignment method utilizing relative comparison information. Wang and Mahadevan [37], [38] introduced manifold alignment methods based on Procrustes analysis and for the situation without correspondences. In the recent years, manifold learning and alignment have received much attention in the neural networks and learning systems community [39]–[41]. Compared to these previous research works, our key contribution is that we creatively construct a learning system for the social network alignment problem based on the general idea of manifold alignment, widening its applicability in practice.

Recently, there has been a lot of interest in learning with hypergraphs [16], [17], [42]–[46]. Zhou *et al.* [42] developed a general framework which was applicable to classification, clustering, and embedding on hypergraph data. Tan *et al.* [16] proposed a ranking framework on hypergraphs. To make full use of the high-order relations in social networks and user profiles, we propose a new manifold alignment method on hypergraphs in this paper. The main contribution in the technical aspect is that the method can learn edge weights automatically from supervision.

III. Preliminaries

This section describes the notations used throughout this paper and then formally defines the user mapping problem of this paper.

A. Notations

In this paper, we use bold-face upper case letters to denote matrices and use bold-face lower case letters to denote vectors. Let X and Y be two social networks across which we try to map users.[4] For the social structure of each network, we build a hypergraph (called *social hypergraph*) and construct hyperedges corresponding to social relations among users, as shown in Fig. 1. Taking the network X as an example, we build a hypergraph $G^X = (V^X, E^X, w^X)$ for X, 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^X is a weight function defined as $w^X : E^X \to \mathbb{R}$. Each hyperedge $e \in E^X$ is a subset of V^X. The degree of a hyperedge e is defined by $\delta(e) = |e|$, i.e., the cardinality of e. The degree $d(v)$ of a vertex $v \in V^X$ is $d(v) = \sum_{e \in E^X | v \in e} w^X(e)$. 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

$$d(v) = \sum_{e \in E^X} w^X(e) h(v, e) \tag{1}$$

$$\delta(e) = \sum_{v \in V^X} h(v, e). \tag{2}$$

Let $\mathbf{D}_e^X$ and $\mathbf{D}_v^X$ be two diagonal matrices containing hyperedge and vertex degrees, respectively. Let $\mathbf{W}^X$ be a $|E^X| \times |E^X|$ diagonal matrix containing hyperedge weights. Similarly, we have another social hypergraph $G^Y = (V^Y, E^Y, w^Y)$ for Y with the corresponding matrices $\mathbf{H}^Y$, $\mathbf{D}_e^Y$, $\mathbf{D}_v^Y$, and $\mathbf{W}^Y$.

Besides, we build a hypergraph for user profiles across the two social networks X and Y (called *profile hypergraph*). Specifically, each profile feature can take multiple values which we call profile terms in this paper. A hyperedge is built for each unique profile term, such as a location or a school. All users with the same profile term form vertices in the corresponding hyperedge. For the sake of simplicity, we are focused on discrete features in this paper. However, our method can also deal with continuous features by employing proper similarity measures and building hyperedges via proper clustering methods. In practice, we choose only those features shared by different networks. We use superscript F to indicate the profile hypergraph, so we have $G^F = (V^F, E^F, w^F)$ with the corresponding matrices: $\mathbf{H}^F$, $\mathbf{D}_e^F$, $\mathbf{D}_v^F$, and $\mathbf{W}^F$.

The unified hypergraph is constructed based on G^X, G^Y, and G^F. Let $G^U = (V^U, E^U, w^U)$ denote the unified hypergraph. We have $V^U = V^X \cup V^Y$ and $E^U = E^X \cup E^Y \cup E^F$. The weight function $w^U(\cdot)$ is defined so that the weights of different profile features can be customized. The details are in Section V.

We model the user mapping problem in a semisupervised way [47]. So partial user correspondences are needed beforehand. Let l be the set of indices of labeled users in the two networks. The set of labeled correspondences is denoted by: $\{(u_{Xi}, u_{Yi}) | i \in l\}$. This labeled user set is a subset of all the *common users* between the two networks. Without loss of generality, we assume the labeled users are assigned indices 1 to $|l|$ in matrices and vectors related to social hypergraphs (i.e., $l = \{1, 2, \ldots, |l|\}$). We use m and n to denote the set of indices of unlabeled users in X and Y, respectively. For a matrix $\mathbf{M}$ and index sets p and q, we use $\mathbf{M}_{pq}$ to represent the corresponding subblock in $\mathbf{M}$. For vectors, taking an index set as subscript has an analogous meaning.

B. Problem Formulation

Generally speaking, the user mapping problem can be defined by: *given social relations in X and Y, user profiles, and $\{(u_{Xi}, u_{Yi}) | i \in l\}$, find for a user u the users in the other network which are most likely to belong to the same person as u*. In this paper, we formulate it as a ranking problem based on embedding space learning [48]. For a user u in one network, we rank users in the other network according to their probabilities of being the corresponding user of u. Let $\mathbf{f}$ and $\mathbf{g}$ denote real-valued vectors containing embedding coordinates of the users in X and Y, respectively.[5] Then, our problem becomes: given G^U and $\{(u_{Xi}, u_{Yi}) | i \in l\}$, we aim to learn an embedding space specified by $\mathbf{f}$ and $\mathbf{g}$ which best preserves the relations in G^U and $\{(u_{Xi}, u_{Yi}) | i \in l\}$. The intuition is that as follows.

1) By preserving social relations and the supervised correspondence relations, the two networks can be aligned in an optimal sense in the embedding space [10], [18].
2) Preserving profile similarity relationships in the embedding space can provide additional calibration since people tend to have similar profiles in different social Websites.

Once the optimal $\mathbf{f}$ and $\mathbf{g}$ are obtained, for a user u, we can sort users in the other network in descending order of their similarities to u in the embedding space.

IV. Review of Manifold Alignment on Hypergraph

In this section, we briefly review the MAH method [10] which is closely related to our work. MAH performs manifold alignment based on social hypergraphs and partial user correspondences for user mapping across networks. By MAH, users in the two networks can be mapped into a common embedding space. It is intrinsically a regularization framework, forcing vertices (i.e., users) to be near each other in the embedding space if they are contained in many common hyperedges. MAH forces each correspondence user pair (u_{Xi}, u_{Yi}) to be mapped to one point in the learned space, so that effective alignment between the two networks can be established. Let d

[4] Though focused on two networks, the proposed UMAH could also be applied to mapping problems for multiple networks.

[5] Here, we take 1-D embedding spaces for illustration. The solution can be generalized to d-dimensional spaces easily, as will be shown shortly.

be the dimensionality of the learned space. In the following, we will first illuminate MAH with $d = 1$, and then generalize it to $d > 1$.

A. 1-D Case

In this case, $\mathbf{f}$ and $\mathbf{g}$ are $|V^X| \times 1$ and $|V^Y| \times 1$ vectors, respectively. MAH forces the labeled parts of $\mathbf{f}$ and $\mathbf{g}$ to be mapped to the same positions, i.e., $\mathbf{f}_l = \mathbf{g}_l$. For clarity, we use $\mathbf{t}_l$ instead of $\mathbf{f}_l$ and $\mathbf{g}_l$ in the following occasionally. The cost function of $\mathbf{f}$ and $\mathbf{g}$ is defined as follows:

$$\begin{aligned} C(\mathbf{f}, \mathbf{g}) = &\frac{1}{2} \sum_{e \in E^X} \frac{1}{\delta(e)} \sum_{\{v_i, v_j\} \subseteq e} w^X(e) \| f_i - f_j \|^2 \\ &+ \frac{1}{2} \sum_{e \in E^Y} \frac{1}{\delta(e)} \sum_{\{v_i, v_j\} \subseteq e} w^Y(e) \| g_i - g_j \|^2 \\ &\text{s.t.} \quad \mathbf{f}_l = \mathbf{g}_l = \mathbf{t}_l \end{aligned} \tag{3}$$

where the two terms at the right-hand side of the equal sign are smoothness constraints. Note that each hyperedge is normalized by its degree $\delta(e)$, i.e., the number of vertices contained in this hyperedge. In this way, hyperedges with different sizes will be equally treated. This is important since relations (e.g., a user group) involving a large number of users should be general and with little discriminative power. With simple algebraic transformation, the first term can be rewritten as

$$\frac{1}{2} \sum_{e \in E^X} \frac{1}{\delta(e)} \sum_{\{v_i, v_j\} \subseteq e} w^X(e) \| f_i - f_j \|^2 = \mathbf{f}^T \mathbf{L}^X \mathbf{f} \tag{4}$$

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

$$\frac{1}{2} \sum_{e \in E^Y} \frac{1}{\delta(e)} \sum_{\{v_i, v_j\} \subseteq e} w^Y(e) \| g_i - g_j \|^2 = \mathbf{g}^T \mathbf{L}^Y \mathbf{g} \tag{5}$$

where $\mathbf{L}^Y = \mathbf{D}_v^Y - \mathbf{H}^Y \mathbf{W}^Y (\mathbf{D}_e^Y)^{-1} (\mathbf{H}^Y)^T$.

Then, the cost function becomes

$$\begin{aligned} C(\mathbf{f}, \mathbf{g}) &= \mathbf{f}^T \mathbf{L}^X \mathbf{f} + \mathbf{g}^T \mathbf{L}^Y \mathbf{g} \\ &= \mathbf{h}^T \mathbf{L}^Z \mathbf{h} \end{aligned} \tag{6}$$

where $\mathbf{h} = [\mathbf{t}_l^T, \mathbf{f}_m^T, \mathbf{g}_n^T]^T$ and $\mathbf{L}^Z$ has the following structure:

$$\mathbf{L}^Z = \left[\begin{array}{c:c:c} \mathbf{L}_{ll}^X + \mathbf{L}_{ll}^Y & \mathbf{L}_{lm}^X & \mathbf{L}_{ln}^Y \\ \hdashline \mathbf{L}_{ml}^X & \mathbf{L}_{mm}^X & \mathbf{0} \\ \hdashline \mathbf{L}_{nl}^Y & \mathbf{0} & \mathbf{L}_{nn}^Y \end{array} \right]. \tag{7}$$

In order to remove an arbitrary scaling factor (and also maximize the general variance) in the embedding space, we minimize the Rayleigh quotient [18] as follows:

$$\min_{\mathbf{h}} \tilde{C}(\mathbf{h}) = \frac{\mathbf{h}^T \mathbf{L}^Z \mathbf{h}}{\mathbf{h}^T \mathbf{h}} \quad \text{s.t.} \quad \mathbf{h}^T \mathbf{e} = 0 \tag{8}$$

where $\mathbf{e}$ is a vector with all the elements equal to 1. $\mathbf{e}$ is an optimal solution for minimizing the cost function $C(\mathbf{f}, \mathbf{g})$. Nevertheless, this solution maps all the users to one point, meaning it is trivial and should be discarded. The optimization problem of $\tilde{C}(\mathbf{h})$ can be solved by locating the first nontrivial eigenvector of $\mathbf{L}^Z$ when sorting the eigenvectors in ascending order of their corresponding eigenvalues. Alternatively, we can also maximize the global variance in the learned space and the solution can be obtained by solving a generalized eigenvalue problem [10].

B. General d-Dimensional Case

In practice, we usually learn a multidimensional space with $d > 1$. To this end, we define a $|V^X| \times d$ matrix $\mathbf{F} = [\mathbf{f}_1 \mathbf{f}_2 \ldots \mathbf{f}_d]$, a $|V^Y| \times d$ matrix $\mathbf{G} = [\mathbf{g}_1 \mathbf{g}_2 \ldots \mathbf{g}_d]$, and a $(|V^X| + |V^Y| - l) \times d$ matrix $\mathbf{H} = [\mathbf{h}_1 \mathbf{h}_2 \ldots \mathbf{h}_d]$ where vectors $\mathbf{f}_i$, $\mathbf{g}_i$, and $\mathbf{h}_i$ contain users' coordinates on the ith dimension. For each dimension $i \in \{1, 2, \ldots, d\}$, we need to minimize $\tilde{C}(\mathbf{h}_i)$. The cost function then becomes

$$\tilde{C}(\mathbf{H}) = \frac{\sum_{i=1}^d \mathbf{h}_i^T \mathbf{L}^Z \mathbf{h}_i}{\sum_{i=1}^d \mathbf{h}_i^T \mathbf{h}_i} = \frac{tr(\mathbf{H}^T \mathbf{L}^Z \mathbf{H})}{tr(\mathbf{H}^T \mathbf{H})} \tag{9}$$

where $tr(\cdot)$ denotes the trace of a matrix. Then, the optimization problem in (8) becomes

$$\begin{aligned} &\min \tilde{C}(\mathbf{H}) \\ &\quad = \frac{tr(\mathbf{H}^T \mathbf{L}^Z \mathbf{H})}{tr(\mathbf{H}^T \mathbf{H})}, \quad \text{s.t.} \quad \mathbf{h}_i^T \mathbf{e} = 0, \quad \forall i \in \{1, 2, \ldots, d\}. \end{aligned} \tag{10}$$

The solution of (10) is obtained by arranging the first d nontrivial eigenvectors corresponding to the smallest eigenvalues of $\mathbf{L}^Z$ as columns of $\mathbf{H}$. An alternative version can also be derived by maximizing the global variance in the embedding space [10].

After the optimal embedding space is obtained, we can then compute the cosine similarity for any user pair (u_{Xi}, u_{Yj}) and rank users in Y/X for u_{Xi}/u_{Yj} accordingly

$$\text{rel}(u_{Xi}, u_{Yj}) = \frac{\sum_{v=1}^d \mathbf{F}_{iv} \times \mathbf{G}_{jv}}{\sqrt{\sum_{v=1}^d \mathbf{F}_{iv}^2} \times \sqrt{\sum_{v=1}^d \mathbf{G}_{jv}^2}}. \tag{11}$$

V. Unified Manifold Alignment on Hypergraph

MAH only exploits social structures for user mapping. In this section, we present the UMAH framework which can leverage both social structures and user profiles. UMAH also decides the optimal weights for different profile features automatically.

Formally, we define B to be the set of user profile features and $b \in B$ to represent a particular feature. Let E_b^F be the profile hyperedge set corresponding to profile terms belonging to feature b. Thus, we have $E^F = \cup_{b \in B} E_b^F$. The weight function $w^U(\cdot)$ for G^U is defined as follows:

$$w^U(e) = \begin{cases} w_b, & \text{if } \exists b \in B, \ e \in E_b^F \\ 1, & \text{otherwise.} \end{cases} \tag{12}$$

Equation (12) means, we fix social hyperedge weights to 1 and assign a learnable edge weigh w_b to each profile feature b. All hyperedges corresponding to b share the same weight. We do not choose to assign a free weight parameter to each profile term since: 1) it would incur overfitting since some features may contain many terms and 2) different terms of a feature are semantically correlated. In UMAH, we learn the embedding space from the unified hypergraph G^U and labeled correspondences. In the following, we first take the $d = 1$ case for derivation.

A. Objective Formulation

The objective function of UMAH consists of three parts. The first part is similar to (3) in MAH. We force the learned space to preserve the proximity among users conveyed by social hyperedges. In particular, we have

$$\begin{aligned} C_1^U &= \frac{1}{2}\sum_{e\in E^X}\frac{1}{\delta(e)}\sum_{\{v_i,v_j\}\subseteq e} w^U(e)\|f_i - f_j\|^2 \\ &\quad + \frac{1}{2}\sum_{e\in E^Y}\frac{1}{\delta(e)}\sum_{\{v_i,v_j\}\subseteq e} w^U(e)\|g_i - g_j\|^2 \\ &= \mathbf{f}^T\mathbf{L}^X\mathbf{f} + \mathbf{g}^T\mathbf{L}^Y\mathbf{g} \\ &= \mathbf{p}^T\mathbf{L}^{XY}\mathbf{p} \end{aligned} \tag{13}$$

where $\mathbf{p} = [\mathbf{f}^T, \mathbf{g}^T]^T$ is the concatenation vector of $\mathbf{f}$ and $\mathbf{g}$, and $\mathbf{L}^{XY}$ is a block-diagonal concatenation of $\mathbf{L}^X$ and $\mathbf{L}^Y$

$$\mathbf{L}^{XY} = \left[\begin{array}{c:c} \mathbf{L}^X & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{L}^Y \end{array}\right]. \tag{14}$$

The key difference between (13) and (3) is that there is no hard constraint on the coordinates of labeled users. As will be shown shortly, we impose soft constraints on labeled users in order to learn reasonable profile hyperedge weights.

The second part comes from the profile hyperedge constraints. That is, we require users which are similar in profiles (i.e., share many profile hyperedges) to be near each other in the learned space

$$\begin{aligned} C_2^U &= \frac{1}{2}\sum_{b\in B}\sum_{e\in E_b^F}\frac{1}{\delta(e)}\sum_{\{v_i,v_j\}\subseteq e}(w_b)^\beta\|p_i - p_j\|^2 \\ \text{s.t.} &\quad \sum_{b\in B} w_b = \gamma, \quad w_b > 0 \quad \forall b \in B \end{aligned} \tag{15}$$

where $\beta > 1$ is a scale hyperparameter which controls the effective weights of hyperedges and avoids trivial solutions [49], and γ is a hyperparameter representing the total weight of profile features. The γ is used to control the relative importance of profile hyperedges as compared to social hyperedges.

The third part concerns supervision from the labeled users. The purpose of this part is twofold.

1) We encourage each pair of labeled corresponding users to be projected to the same position in the learned space.
2) By constraining the proximity relationships regarding labeled users, we aim to learn $\{w_b\}_{b\in B}$ that best conforms to the supervised proximity.

Intuitively, a good profile feature would take the same value (i.e., profile term) for a pair of corresponding users, while tends to show different values for noncorresponding users. Therefore, the objective terms for this part are

$$\min C_3^U := \left(\frac{1}{2}\sum_{i\in l}\|f_i - g_i\|^2\right). \tag{16}$$

$$\max C_4^U := \left(\sum_{i\in l}\sum_{j\in l, j\neq i}\|f_i - g_j\|^2\right). \tag{17}$$

Equation (16) means, we encourage the labeled pairs of corresponding users to have the same coordinates, respectively, while (17) keeps noncorresponding users far away from each other. As mentioned above, this part of the objective has two purposes: 1) equation (16) induces effective alignment of the two networks and 2) the two terms together provide a guide for learning profile weights. Since the sum of all the profile weights is fixed to γ in (15), features that tend to group corresponding users while avoid encircling noncorresponding users will obtain high weights.

Finally, we synthesize all the four terms C_1^U, C_2^U, C_3^U, and C_4^U to form the optimization problem for UMAH

$$\begin{aligned} \min_{\mathbf{p},\{w_b\}} &\quad \frac{\frac{1}{|E^X|+|E^Y|+|E^F|}\left(C_1^U + C_2^U\right) + \frac{\mu}{|l|}C_3^U}{C_4^U + \mathbf{p}^T\mathbf{p}} \\ \text{s.t.} &\quad \sum_{b\in B} w_b = \gamma \\ &\quad w_b > 0, \quad \forall b \in B \\ &\quad \mathbf{p}^T\mathbf{e} = 0. \end{aligned} \tag{18}$$

Here μ is a hyperparameter for controlling the impact of the supervision term C_3^U and we add $\mathbf{p}^T\mathbf{p}$ in the denominator to also maximize the general variance in the embedding space. We normalize the hypergraph part and the supervision part in the numerator by the numbers of edges (we can also view C_3^U as a graph embedding term, and the number of edges is $|l|$). Note that C_1^U and C_2^U share the same normalization term. This is because they are both a summation over hyperedges [see (13) and (15)]. Hence, we normalize them by the total number of social and profile hyperedges, i.e., $|E^X| + |E^Y| + |E^F|$. The relative importance of different kinds of hyperedges is captured by $\{w_b\}$. This normalization scheme can help find stable μ regardless of data sizes.

B. Optimization

The objective function (18) is a nonconvex problem with multivariables. However, it is convex in $\mathbf{p}$ when $\{w_b\}$ is fixed, and vice versa. Hence, we take an iterative optimization scheme where we optimize the variables in turn in each iteration [49]. Since the objective function (18) is well lower bounded, the iterative optimization will converge after a sufficient number of iterations.

When $\{w_b\}$ is fixed, the constraints related to $\{w_b\}$ in (18) can be ignored. C_2^U can be transformed into the corresponding matrix-vector form

$$C_2^U = \mathbf{p}^T\mathbf{L}^F\mathbf{p} \tag{19}$$

where

$$\mathbf{L}^F = \mathbf{D}_v^F - \mathbf{H}^F\mathbf{W}^F(\mathbf{D}_e^F)^{-1}(\mathbf{H}^F)^T. \tag{20}$$

Moreover, C_3^U and C_4^U can be rewritten as

$$\frac{1}{2}\sum_{i\in l}\|f_i - g_i\|^2 = \mathbf{p}^T\mathbf{L}^C\mathbf{p} \tag{21}$$

$$\sum_{i\in l}\sum_{j\in l, j\neq i}\|f_i - g_j\|^2 = \mathbf{p}^T\tilde{\mathbf{L}}^C\mathbf{p} \tag{22}$$

where $\mathbf{L}^C = \mathbf{D}^C - \mathbf{W}^C$ and $\tilde{\mathbf{L}}^C = \tilde{\mathbf{D}}^C - \tilde{\mathbf{W}}^C$ are graph Laplacian matrices [30] with their corresponding weighted adjacency matrices $\mathbf{W}^C$ and $\tilde{\mathbf{W}}^C$ defined as

$$\mathbf{W}^C = \begin{array}{c} \begin{array}{cccc} \overbrace{\quad}^{|l|} & \overbrace{\quad}^{|V^X|-|l|} & \overbrace{\quad}^{|l|} & \overbrace{\quad}^{|V^Y|-|l|} \end{array} \\ \left[\begin{array}{c:c:c:c} \mathbf{0} & \mathbf{0} & \mathbf{I}_{|l||l|} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{I}_{|l||l|} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] \end{array}$$

$$\tilde{\mathbf{W}}^C = \begin{array}{c} \begin{array}{cccc} \overbrace{\qquad\qquad}^{|l|} & \overbrace{\quad}^{|V^X|-|l|} & \overbrace{\qquad\qquad}^{|l|} & \overbrace{\quad}^{|V^Y|-|l|} \end{array} \\ \left[\begin{array}{c:c:c:c} \mathbf{0} & \mathbf{0} & \mathbf{E}_{|l||l|} - \mathbf{I}_{|l||l|} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{E}_{|l||l|} - \mathbf{I}_{|l||l|} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \hdashline \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] \end{array}$$

where $\mathbf{I}$'s and $\mathbf{E}$'s represent identity matrices and matrices with all the elements equal to 1, respectively, and their sizes are indicated in the subscripts, e.g., $\mathbf{E}_{|l||l|}$ means its size is $|l| \times |l|$. Then, the subproblem for $\mathbf{p}$ can be rewritten as

$$\begin{aligned} &\min_{\mathbf{p}} \frac{\mathbf{p}^T\left[\frac{1}{|E^X|+|E^Y|+|E^F|}(\mathbf{L}^{XY} + \mathbf{L}^F) + \frac{\mu}{|l|}\mathbf{L}^C\right]\mathbf{p}}{\mathbf{p}^T(\tilde{\mathbf{L}}^C + \mathbf{I})\mathbf{p}} \\ &\text{s.t. } \mathbf{p}^T\mathbf{e} = 0. \end{aligned} \tag{23}$$

This is a generalized Rayleigh quotient of the form $\mathbf{q}^T\mathbf{A}\mathbf{q}/\mathbf{q}^T\mathbf{B}\mathbf{q}$ where $\mathbf{A}$ and $\mathbf{B}$ are Hermitian and $\mathbf{B}$ is positive definite [50], [51]. Hence, the optimal $\mathbf{p}$ can be obtained by solving the corresponding generalized eigenvalue problem.

When $\mathbf{p}$ is fixed, we can just focus on C_2^U since the other components do not involve $\{w_b\}$. The objective function of this subproblem is

$$\begin{aligned} &\min_{\{w_b\}} \sum_{b\in B} w_b^\beta M_b \\ &\text{s.t.} \sum_{b\in B} w_b = \gamma \\ &\qquad w_b > 0, \quad \forall b \in B \end{aligned} \tag{24}$$

where $M_b = \sum_{e\in E_b^F}(1/\delta(e))\sum_{\{v_i,v_j\}\subseteq e}\|p_i - p_j\|^2$. By introducing Lagrange multipliers for the constraints and letting the partial derivatives equal 0, we have

$$w_b = \frac{\gamma}{\sum_{b'\in B}\left(\frac{M_b}{M_{b'}}\right)^{\frac{1}{\beta-1}}}. \tag{25}$$

The above derivation of UMAH is based on the $d = 1$ case. For $d > 1$, we optimize the objectives in all the d dimensions of the space. In that case, we define matrices $\mathbf{F}$ and $\mathbf{G}$ as in MAH and let $\mathbf{P} = [\mathbf{p}_1\mathbf{p}_2 \ldots \mathbf{p}_d]$ be the column-wise concatenation of $\mathbf{F}$ and $\mathbf{G}$. The subproblem in (23) then

Algorithm 1 Optimization of UMAH

Input: Unified hypergraph: G^U; user correspondences: l; hyperparameters: β, γ, μ, d
Output: The embedding space: $\mathbf{P} = [\mathbf{F}^T, \mathbf{G}^T]^T$

1 **begin**
2 Construct $\mathbf{L}^{XY}, \mathbf{L}^F, \mathbf{L}^C$ and $\tilde{\mathbf{L}}^C$.
3 Initialize each $w_b = \frac{\gamma}{|B|}$.
4 **repeat**
5 Optimize subproblem (26) with respect to $\mathbf{P}$ while keeping $\{w_b\}_{b\in B}$ fixed.
6 Calculate M_b for each feature b and set w_b according to (25), while keeping $\mathbf{P}$ fixed.
7 **until** *convergence or max no. iterations reached*
8 **end**

becomes

$$\begin{aligned} &\min_{\mathbf{P}} \frac{\mathrm{tr}\left(\mathbf{P}^T\left[\frac{1}{|E^X|+|E^Y|+|E^F|}(\mathbf{L}^{XY} + \mathbf{L}^F) + \frac{\mu}{|l|}\mathbf{L}^C\right]\mathbf{P}\right)}{\mathrm{tr}(\mathbf{P}^T(\tilde{\mathbf{L}}^C + \mathbf{I})\mathbf{P})} \\ &\text{s.t. } \mathbf{p}_i^T\mathbf{e} = 0, \quad \forall i \in \{1, 2, \ldots, d\}. \end{aligned} \tag{26}$$

The subproblem in (24) does not change. The only difference is that Euclidean distance is used to calculate the distances between users' latent representations.

In summary, the optimization algorithm of UMAH is depicted in Algorithm 1. At the beginning, we first initialize $\{w_b\}$ evenly. Then, $\mathbf{P}$ and $\{w_b\}$ are updated in turn, until they converge or the preset maximum number of iterations is reached. Once $\mathbf{F}$ and $\mathbf{G}$ are learned, we can then use the same ranking scheme [(11)] as in MAH for user mapping. Unlike MAH, UMAH requires iterative optimization, so the running time is longer than that of MAH. Fortunately, in experiments we find Algorithm 1 typically converges in 2 or 3 iterations.

C. Time Complexity Analysis

In this section, we analyze the time complexity of Algorithm 1. Since social networks and user profiles are usually sparse, we provide time complexity analysis based on the assumption that all the involved matrices are sparse. We mainly focus on multiplication operations since floating-point computation is much more costly, compared to addition and subtraction.

Algorithm 1 can be divided into two phases. In the first phase, we construct the necessary matrices (line 2). The time cost for computing $\mathbf{L}^X$ is $O(\sum_{e\in E^X}\delta(e)^2)$, where $\delta(e)$ is the degree of e as calculated in (2). The time costs of $\mathbf{L}^Y$ and $\mathbf{L}^F$ can be derived similarly. The calculation time for $\mathbf{L}^C$ and $\tilde{\mathbf{L}}^C$ can be ignored since there is no multiplication operations. In the second phase, the major cost in each iteration is due to the calculation of M_b's and the generalized eigenvectors of (26). The corresponding time costs are $O(\sum_{e\in E^F}\delta(e)(\delta(e) - 1))$ and $O((|V^X| + |V^Y|)^3)$, respectively. Hence, the time cost for the second phase is $O(\tau\ (\sum_{e\in E^F}\delta(e)(\delta(e) - 1) + (|V^X| + |V^Y|)^3))$, where τ denotes the number of iterations. We could use *eigs* in MATLAB to reduce the time cost of the calculation of generalized eigenvectors.

TABLE I
DATA SET INFORMATION

Data set	Twitter-BlogCatalog	Twitter-last.fm	MySpace-last.fm
Users in X	2710	12506	14029
Users in Y	2710	12405	12285
Interop	0.3895	0.1341	0.0913
Profile Features	location, username	location, username	username, age, gender

VI. EXPERIMENTS

In this section, we investigate the use of UMAH for mapping users across networks. Before presenting the experimental results, we introduce the evaluation metrics, data sets, and compared algorithms.

A. Evaluation Metrics

In [10], we proposed a metric called *Interoperability* (abbreviated as Interop) to measure the structural relatedness between two networks. Structural relatedness can greatly affect mapping performance based on social structures, as shown on simulated data sets [10]. Here, we apply Interop on three real world data sets. Interop is defined based on a set of labeled pairs of corresponding users. It measures the fraction of pairwise social relations among the labeled users that exist in both networks

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

where Relations$X$ is the set of pairwise connections (e.g., the two users are friends or in the same group) of the labeled users in network X and Relations$Y$ is that for network Y. Correlations is the intersection of Relations$X$ and Relations$Y$. It is obvious that $0 \leq \text{Interop}(X, Y) \leq 1$.

Regarding evaluation metrics, we use Hit-Precision and mean average precision (MAP) which are well-established and widely used evaluation metrics in real user linkage applications [12]. Hit-Precision@k first calculates position-sensitive top-k precision (we adopt $k = 1, 3, 5$ in this paper) for each test user u as follows:

$$h(u) = \frac{\max(k - (\text{hit}(u) - 1), 0)}{k} \tag{27}$$

where hit(u) represents the position of u's corresponding user in the returned ranking list. It can be easily verified that $h(u)$ is higher with lower hit(u) and its range is $[0, 1]$. Hit-Precision@k is then defined as the average of $h(u)$ over all the N test users: $(\sum_{i=1}^{N} h(x_i)/N)$.

MAP pays more attention to the whole ranking list. A higher MAP means better overall ranking performance. The metric first calculates the average precision for each test user u, and then calculates MAP from all the test users as follows:

$$\text{MAP} = \frac{\sum_{u=1}^{N} \text{AP}(u)}{N} \tag{28}$$

where N is the number of test users and AP(u) is the average precision of user u. In our context, AP(u) takes the reciprocal of the ranking position of the ground truth user in the ranking list. For instance, if a labeled user u in X finds its corresponding user at rank 9, then AP(u) is $1/9$.

B. Data Sets

In the experiments, we employ three real world data sets, Twitter-BlogCatalog, Twitter-last.fm, and MySpace-last.fm, to evaluate UMAH.

1) Twitter-BlogCatalog: We developed a data crawler to crawl data from BlogCatalog. BlogCatalog provided not only directories of blogs, but also an attribute called "my communities" for each user. This attribute enables users to list their corresponding linkages to other online social networks. Hence, we can easily obtain labeled user correspondences across social networks. Specifically, we chose BlogCatalog and Twitter as the two networks, and tried to map users between them. Note that social networks of these two communities are both directed (i.e., following/followed relations). In the crawled users, there were 21 577 users who showed their Twitter usernames and linkages. We further downloaded all their social relations (i.e., followers and followees) and their profile information. To form the final data set for experiments, we filtered users by ensuring that each user in the final data set has at least four friends (followees or followers) in either of the two networks. For user profile information, we collect the username and location information for each user. As mentioned before, user profiles often contain false information or partly missing information with different reasons. In our data set, 6.38% users do not reveal their location information in both networks and 31.03% only publish location information in one network. In the remaining users (62.58%), 22.58% users publish completely a different location information while 25.61% users have partly matched locations (e.g., "*Toronto, ON, Canada*" and "*Toronto, Canada*"); only 14.39% users input exactly the same location information in the two networks.

2) Twitter-Last.fm: This data set is a subset of the social network service data set from [11]. We produced the subset as below: we first searched for users (in Twitter and last.fm) who claimed profile information in the data and then extended the user set by social connections. In this way, we got a subset with 12 506 users from Twitter and 12 405 users from last.fm. In this data set, there are also two types of profile information: username and location. A part of the users in this data set has mapping information.

3) MySpace-Last.fm: This data set was produced similarly as the previous one, but for MySpace and last.fm. There are 14 029 users from Myspace and 12 285 users from last.fm. Three types of profile information: username, age, and gender, were collected. Table I shows basic statistics and the Interop scores for all the three data sets.

In building social hypergraphs, we first constructed one hyperedge for each directed social relation (followed or following). Then, we built a hyperedge that contained all the followers for each user. For example, if u_1, u_2, and u_3 are followers of u_4, we built a hyperedge containing u_1, u_2, and u_3. The motivation was that the followers of the same target user would probably be in the same "group" or "circle." For example, if the target user 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. Considering that the location information is often partly matched, we built a hyperedge for each level of locations in the profile hypergraph. For example, we built three hyperedges for the following location: "*Toronto, ON, Canada*". They were for"*Toronto*", "*ON*," and "*Canada*," respectively. Besides, we built one more profile hyperedge which contained all the users to ensure this hypergraph is connected [42].

For all the data sets, we randomly chose 30% user correspondences for training and used the remaining 70% for test. For methods that have hyperparameters, we further split the training set. Specifically, for methods with hyperparameters 25% user correspondences were for training the model and 5% were for learning the hyperparameters.

C. Compared Algorithms

To demonstrate the effectiveness of the proposed algorithm, we employ four well-designed or state-of-the-art baseline methods.

1) MAH: The first baseline is from [10], which only uses social structure information.

2) MAH-Combine: In this baseline, we first run MAH on the social hypergraphs and the profile hypergraph, respectively. On the social hypergraphs, MAH preserves social relations and the training correspondences, while on the profile hypergraph MAH preserves profile relations and the training correspondences. Then, we get two similarity scores for u_{X_i} and u_{Y_j}: $rel^S(u_{X_i}, u_{Y_j})$ (social) and $rel^F(u_{X_i}, u_{Y_j})$ (profile). The final similarity score used for ranking is a linear interpolation of them

$$\begin{aligned}\text{rel}(u_{Xi}, u_{Yj}) = \alpha \times \text{rel}^S(u_{Xi}, u_{Yj})\\ + (1-\alpha) \times \text{rel}^F(u_{Xi}, u_{Yj})\end{aligned} \quad (29)$$

where $0 \leq \alpha \leq 1$ is the interpolation hyperparameter controlling the relative importance of the two similarity scores.

3) UMAG: The third baseline is similar to our UMAH but based on ordinary graphs. The edge weights for social relations are set as

$$\text{weight}^S(u_i, u_j) = \frac{|R_{u_i} \cap R_{u_j}|}{|R_{u_i}| + |R_{u_j}|} \quad (30)$$

where R_{u_i} and R_{u_j} are relation sets containing u_i and u_j, respectively. The edge weights for profile relations are set as

$$\text{weight}^F(u_i, u_j) = \frac{|F_{u_i} \cap F_{u_j}|}{|F_{u_i}| + |F_{u_j}|} \quad (31)$$

where F_{u_i} and F_{u_j} are profile term sets of u_i and u_j, respectively. Other aspects of the method, including objective function and optimization, are similar to UMAH.

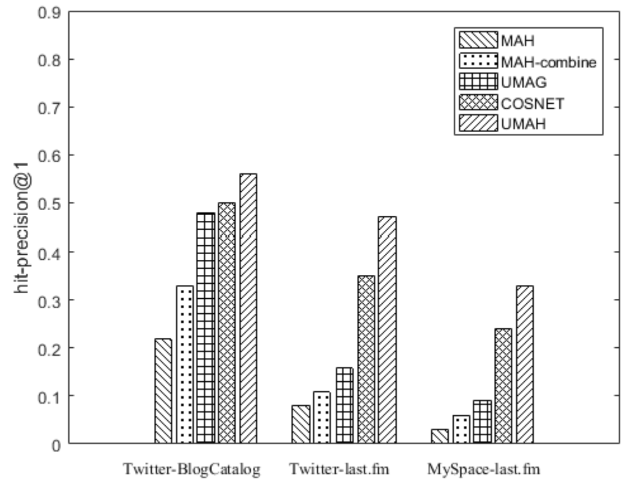


Fig. 2. Hit-Precision@1 on the three data sets for all the five methods.

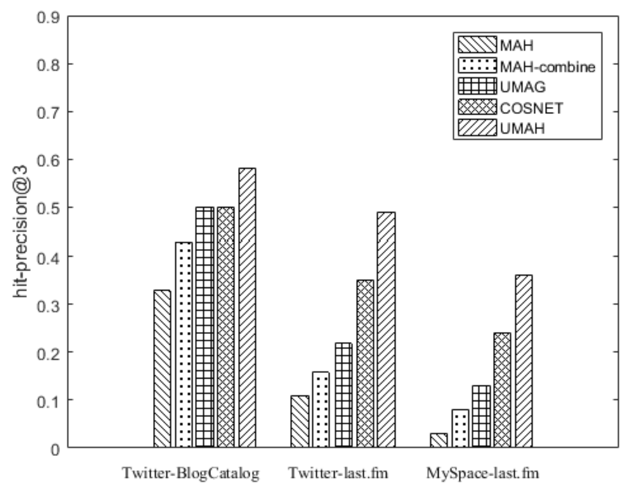


Fig. 3. Hit-Precision@3 on the three data sets for all the five methods.

4) COSNET: This is a state-of-the-art hybrid mapping method which exploits both social relations and profile information [11].

D. Comparison Results

Experimental results by Hit-Precision@k on the three data sets are shown in Figs. 2–4, for $k = 1$, $k = 3$, and $k = 5$, respectively. Results by MAP is shown in Fig. 5. MAH works the worst in all the data sets since it only exploits social relations for user mapping. This is more obvious in the data sets Twitter-last.fm and MySpace-last.fm. Note that the Interop scores of Twitter-last.fm and MySpace-last.fm are much lower than that of Twitter-BlogCatalog (see Table I). MAH's performance on Twitter-last.fm and MySpace-last.fm is much lower than its performance on Twitter-BlogCatalog, which is consistent with Interop. This indicates Interop can be used on real world data to predict mapping performance by using social structures solely. Compared to the baselines, the performance of UMAH is better on all the three data sets. Unified manifold alignment on graph (UMAG) works worse than UMAH because lots of relations in our data sets

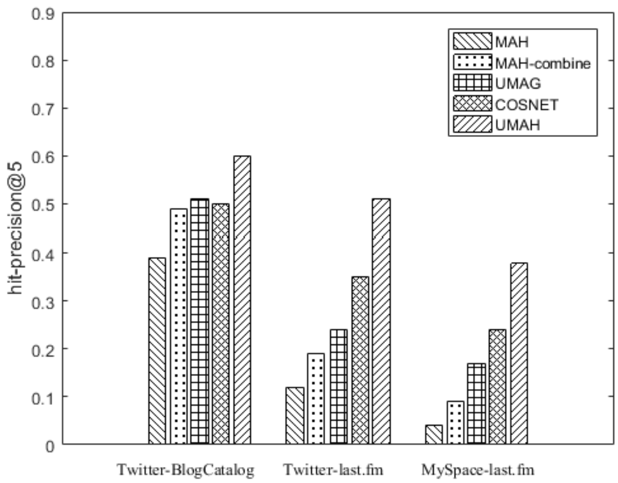


Fig. 4. Hit-Precision@5 on the three data sets for all the five methods.

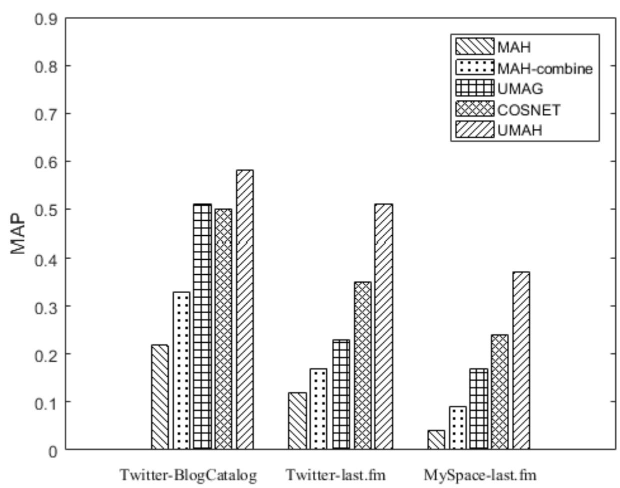


Fig. 5. MAP on the three data sets for all the five methods.

are high-order relations. Using hypergraph to model these relations does not lose information. For example, the degree of a hyperedge (i.e., how many vertices it involves) can reflect the corresponding relation's importance. This information cannot be captured by pairwise modeling. MAH-combine cannot beat UMAH either. This should be because MAH-combine postpones the fusion of the two types of information to the ranking phase, which could be less effective, and it does not customize the weights of profile relations. Regarding COSNET, although it can distinguish the importance of different profile features, it uses pairwise social relations fragmentarily with candidate matching pairs. Hence, COSNET cannot capture the global structures of social networks which would be important for network alignment. In comparison, UMAH stems from manifold learning and therefore can better capture global structures for alignment. Moreover, COSNET cannot well handle high-order relations.

E. Exploring Parameter Setting

There are four hyperparameters in UMAH: d, μ, γ, and β. We study the impact of these hyperparameters in this section. The parameter tuning process is accomplished on the 5% validation set. The results are depicted in Fig. 6, where subfigures in the same row correspond to the same data set and each column corresponds to a hyperparameter.

TABLE II

LEARNED WEIGHTS OF UMAH FOR DIFFERENT PROFILE FEATURES. T-B, T-L, AND M-L REPRESENT TWITTER-BLOGCATALOG, TWITTER-LAST.FM, AND MYSPACE-LAST.FM, RESPECTIVELY

Data set	T-B	T-l	M-l
w for location	1.54	1.29	-
w for username	2.46	1.71	3.95
w for age	-	-	2.14
w for gender	-	-	1.91
w sum (i.e., γ)	4	3	8

For the embedding space dimensionality d, we find that the best value is related to the total number of eigenvectors learned by our method (i.e., $|V^X| + |V^Y|$). The experimental results for tuning d are shown in Fig. 6(a), (e), and (i). Specifically, the best value of d is around $5\% - 10\%$ of the total number of eigenvectors. Too large d would introduce noise and result in an embedding space with too high dimensionality, leading to deteriorated performance [as shown in Fig6(a), (e), and (i)]. Hence, we set d to 400 for Twitter-BlogCatalog, 1200 for Twitter-last.fm, and 2500 for MySpace-last.fm.

Tuning results for μ are shown in Fig. 6(b), (f), and (j). We find the mapping performance is not sensitive to the value of μ, as long as it is large enough. So, we set μ to 30 in all the experiments.

γ and β affect the updating of $\{w_b\}$ in the UMAH optimization. γ is the sum of all the weights of profile features. By examining (25), we can see that a larger β will encourage the distribution of $\{w_b\}$ to be flatter. If γ is fixed and β is small (i.e., close to 1), the algorithm would tend to make some weights dominating, while too large β (e.g., > 10) would fail to discern the importance of different profile features. Hence, in order to obtain proper weights for profile features, we ought to set β to a moderate value. Regarding γ, it represents the total impact of profile features. Intuitively, too small/large values of γ would underemphasize/overemphasize the profile information, as compared to social relations. A moderate value would also achieve the best performance. Note that if we set $\gamma = |B|$ (i.e., the number of profile features), in the even case all profile hyperedge weights are set to 1. Since the weights for social hyperedges are fixed to 1, Setting $\gamma = |B|$ means, we deem profile features and social structures to be equally important. Hence, a moderate value of γ would be near or directly proportional to $|B|$. This gives us a rough notion of what kinds of values are too small/large for γ.

In the tuning process, we first select an intuitive value for β and fix it. Then, we adjust γ to find its best value. In turn, β is tuned. Fig. 6(c), (g), and (k) shows the results of varying parameter γ when β is fixed to 5. Fig. 6(c), (g), and (k) confirms our intuition that moderate values of γ should lead to the best performance. According to the results, we set γ to 4, 3, and 8, for Twitter-BlogCatalog, Twitter-last.fm, and MySpace-last.fm, respectively. Then, we fix γ and tune β. Results are shown in Fig. 6(d), (h), and (l). These results are also consistent with our analysis. For Twitter-BlogCatalog, Twitter-

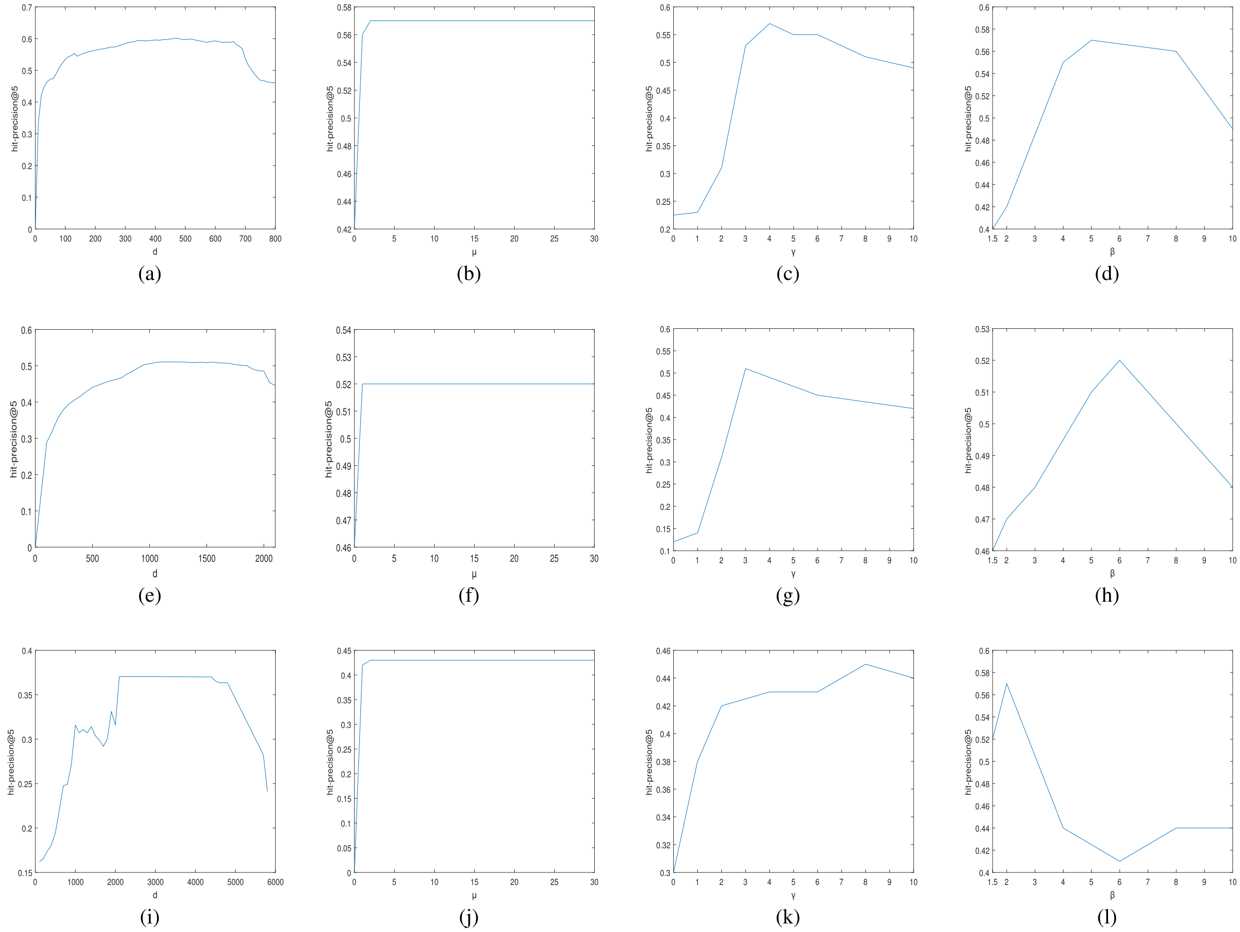


Fig. 6. UMAH performance on the three data sets when varying parameters. (a) Twitter-BlogCatalog-d. (b) Twitter-BlogCatalog-μ. (c) Twitter-BlogCatalog-γ. (d) Twitter-BlogCatalog-β. (e) Twitter-last.fm-d. (f) Twitter-last.fm-μ. (g) Twitter-last.fm-γ. (h) Twitter-last.fm-β. (i) MySpace-last.fm-d. (j) MySpace-last.fm-μ. (k) MySpace-last.fm-γ. (l) MySpace-last.fm-β.

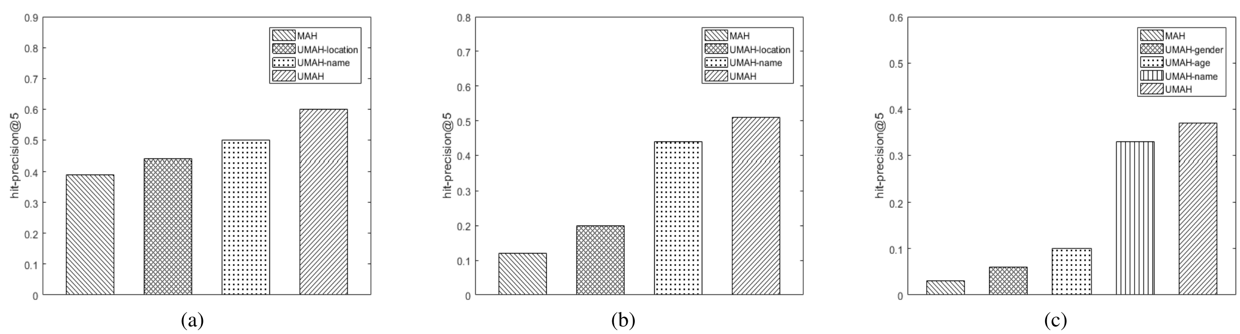


Fig. 7. Contributions of different profile features in the three data sets. (a) Twitter-BlogCatalog. (b) Twitter-last.fm. (c) MySpace-last.fm.

last.fm, and MySpace-last.fm, we get the best performance when $\beta = 5, 6, 2$, respectively.

F. Exploring Contributions of Profile Features

It is also interesting to investigate which profile features are more important in mapping users across networks. We examine the usefulness of each profile feature in this section.

In this experiment, we choose MAH as a baseline which does not use any profile information. Different from UMAH which uses the full profile information, the variants of UMAH, UMAH-location, UMAH-name, UMAH-age, and UMAH-gender, only use one profile feature (i.e., location, username, age, or gender). We show the results in Fig. 7. As can be seen, all of the variants of UMAH work better than MAH. This indicates that every profile feature is useful for mapping users. The contribution of *location* is consistently smaller than that of *username* across the three data sets. It makes sense because many users may live in the same city and share the same location information. Therefore, *location* has less discriminative power than *username*. *Age*, and *gender* are also less important than *username*. UMAH achieves the best performance by optimally combining all the profile features through automatic weighting. We also show UMAH's learned weights for different profile features in Table II. The learned weights correctly reflect the relative importance of these features.

VII. Conclusion

In this paper, we tried to map common users across social networks. To address this problem, we proposed a semi-supervised hypergraph learning framework dubbed UMAH. UMAH combines social structures and user profiles to infer the user correspondence relationships between social networks. Specifically, we first built a unified hypergraph from social relations of the two networks and user profiles. Then, we carried out semisupervised manifold alignment on social structures with some labeled user correspondences, and used profile hyperedges for calibration. The weights of different profile features were learned automatically. A low-dimensional common space was learned for user matching. For a target user u, we used cosine similarity in the common space to assess the probabilities of users in the other network being u's corresponding user. We conducted extensive experiments on three real world data sets to evaluate UMAH. The results showed that UMAH achieved superior performance over state-of-art baseline methods. In this paper, we considered only one type of social relations. However, the model can be extended to incorporate multitype social relations and automatically decide their weights in a similar way as we did for profile relations. We will investigate this issue in the future work.

Acknowledgment

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

[1] J. Vosecky, D. Hong, and V. Y. Shen, "User identification across multiple social networks," in *Proc. 1st Int. Conf. Netw. Digit. Technol. (NDT)*, Ostrava, Czech Republic, Jul. 2009, pp. 360–365.

[2] S. Kumar, R. Zafarani, and H. Liu, "Understanding user migration patterns in social media," in *Proc. 25th Conf. Artif. Intell. (AAAI)*, San Francisco, CA, USA, 2011, pp. 1204–1209.

[3] F. Carmagnola and F. Cena, "User identification for cross-system personalisation," *Inf. Sci.*, vol. 179, nos. 1–2, pp. 16–32, 2009.

[4] B. Cao, N. N. Liu, and Q. Yang, "Transfer learning for collective link prediction in multiple heterogenous domains," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010, pp. 159–166.

[5] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li, "ComSoc: Adaptive transfer of user behaviors over composite social network," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Beijing, China, 2012, pp. 696–704.

[6] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proc. 30th IEEE Symp. Secur. Privacy*, Oakland, CA, USA, May 2009, pp. 173–187.

[7] R. Zafarani and H. Liu, "Connecting corresponding identities across communities," in *Proc. 3rd Int. Conf. Weblogs Social Media (ICWSM)*, San Jose, CA, USA, 2009, pp. 354–357.

[8] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff, "Identifying users across social tagging systems," in *Proc. 5th Int. Conf. Weblogs Social Media (ICWSM)*, Catalonia, Spain, 2011, pp. 522–525.

[9] 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 *Proc. 6th ACM Int. Conf. Web Search Data Mining*, Rome, Italy, 2013, pp. 495–504.

[10] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, and C. Chen, "Mapping users across networks by manifold alignment on hypergraph," in *Proc. AAAI*, 2014, pp. 159–165.

[11] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu, "Cosnet: Connecting heterogeneous social networks with local and global consistency," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1485–1494.

[12] X. Mu, F. Zhu, E.-P. Lim, J. Xiao, J. Wang, and Z.-H. Zhou, "User identity linkage by latent user space modelling," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1775–1784.

[13] S. Zhang and H. Tong, "Final: Fast attributed network alignment," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1345–1354.

[14] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network," in *Proc. 14th Int. Conf. World Wide Web*, Chiba, Japan, 2005, pp. 463–470.

[15] S. Labitzke, I. Taranu, and H. Hartenstein, "What your friends tell others about you: Low cost linkability of social network profiles," in *Proc. 5th SNA-KDD Workshop (SNA-KDD)*, San Diego, CA, USA, 2011, pp. 1–10.

[16] S. Tan, J. Bu, C. Chen, B. Xu, C. Wang, and X. He, "Using rich social media information for music recommendation via hypergraph model," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 7S, no. 1, 2011, Art. No. 22.

[17] S. Agarwal, K. Branson, and S. Belongie, "Higher order learning with graphs," in *Proc. 23rd Int. Conf. Mach. Learn.*, Pittsburgh, PA, USA, 2006, pp. 17–24.

[18] J. Ham, D. D. Lee, and L. K. Saul, "Semisupervised alignment of manifolds," in *Proc. 21st Annu. Conf. Uncertainty Artif. Intell.*, Edinburgh, Scotland, 2005, pp. 120–127.

[19] A. Nunes, P. Calado, and B. Martins, "Resolving user identities over social networks through supervised learning and rich similarity features," in *Proc. 27th Annu. ACM Symp. Appl. Comput.*, Riva del Garda, Italy, 2012, pp. 728–729.

[20] A. Malhotra, L. Totti, W. Meira, Jr., P. Kumaraguru, and V. Almeida, "Studying user footprints in different online social networks," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Istanbul, Turkey, 2012, pp. 1065–1070.

[21] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 321–328.

[22] X. Xu, Z. Huang, L. Zuo, and H. He, "Manifold-based reinforcement learning via locally linear reconstruction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 4, pp. 934–947, Apr. 2017.

[23] D. Tao, L. Jin, Y. Yuan, and Y. Xue, "Ensemble manifold rank preserving for acceleration-based human activity recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1392–1404, Jun. 2016.

[24] D. Frankowski, D. Cosley, S. Sen, L. Terveen, and J. Riedl, "You are what you say: Privacy risks of public mentions," in *Proc. 29th Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Seattle, WA, USA, 2006, pp. 565–572.

[25] L. Backstrom, C. Dwork, and J. Kleinberg, "Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography," in *Proc. 16th Int. Conf. World Wide Web*, Banff, AB, Canada, 2007, pp. 181–190.

[26] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proc. 29th IEEE Symp. Secur. Privacy*, Oakland, CA, USA, May 2008, pp. 111–125.

[27] A. Narayanan and V. Shmatikov, "Myths and fallacies of 'personally identifiable information,'" *Commun. ACM*, vol. 53, no. 6, pp. 24–26, 2010.

[28] W. Pan, N. Aharony, and A. Pentland, "Composite social network for predicting mobile apps installation," in *Proc. 25th Conf. Artif. Intell. (AAAI)*, San Francisco, CA, USA, 2011, pp. 821–827.

[29] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[30] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2001, pp. 585–591.

[31] W. Min, K. Lu, and X. He, "Locality pursuit embedding," *Pattern Recognit.*, vol. 37, no. 4, pp. 781–788, 2004.

[32] X. He, D. Cai, and J. Han, "Learning a maximum margin subspace for image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 2, pp. 189–201, Feb. 2008.

[33] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 342–352, Apr. 2008.

[34] Y. Yuan, L. Mou, and X. Lu, "Scene recognition by manifold regularized deep learning architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 10, pp. 2222–2233, Oct. 2015.

[35] J. H. Ham, D. D. Lee, and L. K. Saul, "Learning high dimensional correspondences from low dimensional manifolds," presented at the 20th Int. Conf. Mach. Learn. (ICML), Washington, DC, USA, 2003.

[36] L. Xiong, F. Wang, and C. Zhang, "Semi-definite manifold alignment," in *Proc. 18th Eur. Conf. Mach. Learn.*, Warsaw, Poland, 2007, pp. 773–781.

[37] C. Wang and S. Mahadevan, "Manifold alignment using procrustes analysis," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, 2008, pp. 1120–1127.

[38] C. Wang and S. Mahadevan, "Manifold alignment without correspondence," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, Pasadena, CA, USA, 2009, pp. 1273–1278.

[39] J. Chen, Z. Ma, and Y. Liu, "Local coordinates alignment with global preservation for dimensionality reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 106–117, Jan. 2013.

[40] X. Li, K. Liu, Y. Dong, and D. Tao, "Patch alignment manifold matting," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7999271/

[41] X. You, W. Ou, C. L. P. Chen, Q. Li, Z. Zhu, and Y. Tang, "Robust nonnegative patch alignment for dimensionality reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 11, pp. 2760–2774, Nov. 2015.

[42] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2006, pp. 1–8.

[43] S. Chen, F. Wang, and C. Zhang, "Simultaneous heterogeneous data clustering based on higher order relationships," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops*, Omaha, NE, USA, 2007, pp. 387–392.

[44] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Las Vegas, NV, USA, 2008, pp. 668–676.

[45] S. R. Bulò and M. Pelillo, "A game-theoretic approach to hypergraph clustering," in *Proc. 22nd Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2009, pp. 1571–1579.

[46] L. An, X. Chen, S. Yang, and X. Li, "Person re-identification by multi-hypergraph fusion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2763–2774, Nov. 2017.

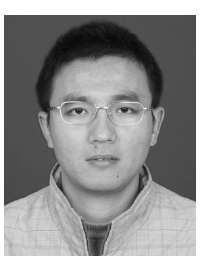
[47] J. Xu, B. Tang, H. He, and H. Man, "Semisupervised feature selection based on relevance and redundancy criteria," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 9, pp. 1974–1984, May 2016.

[48] B. Tang, H. He, Q. Ding, and S. Kay, "A parametric classification rule based on the exponentially embedded family," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 367–377, Feb. 2015.

[49] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, and B. Du, "Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding," *Pattern Recognit.*, vol. 48, no. 10, pp. 3102–3112, Dec. 2015.

[50] P. J. Psarrakos, "Numerical range of linear pencils," *Linear Algebra Appl.*, vol. 317, nos. 1–3, pp. 127–141, 2000.

[51] N. Bebiano, J. da Providência, A. Nata, and J. da Providência, "Generalized Rayleigh quotients and generating vectors," *Linear Multilinear Algebra*, vol. 65, no. 1, pp. 1–23, 2017.



Wei Zhao received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 2002, 2005, and 2015, respectively.

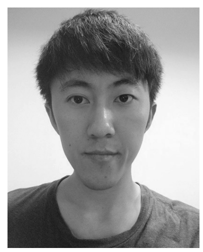
He is currently an Associate Professor with the School of Computer Science and Technology, Xidian University. His current research interests include direction is pattern recognition and intelligent systems, attributed graph mining and search, machine learning, signal processing, and precision guiding technology.

Shulong Tan received the B.S. and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 2008 and 2013, respectively.

From 2013 to 2015, he was a Research Scientist with the University of California at Santa Barbara, Santa Barbara, CA, USA. He is currently a Research Scientist with Baidu Big Data Lab, Sunnyvale, CA, USA. His current research interests include social network mining and recommender systems and text mining.

Ziyu Guan received the B.S. and Ph.D. degrees in computer science from Zhejiang University, Hangzhou, China, in 2004 and 2010, respectively.

From 2010 to 2012, he was a Research Scientist with the University of California at Santa Barbara, Santa Barbara, CA, USA. He is currently a Professor with the School of Information and Technology, Northwest University, Xi'an, China, and the School of Computer Science and Technology, Xidian University, Xi'an. His current research interests include attributed graph mining and search, machine learning, expertise modeling and retrieval, and recommender systems.



Boxuan Zhang received the B.S. degree form Jiangnan University, Wuxi, China, in 2016.

He is currently a Graduate Student with Xidian University, Xi'an, China. His current research interests include machine learning and data mining.

Maoguo Gong (M'07–SM'14) received the B.S. degree in electronic engineering and the Ph.D. degree in electronic science and technology from Xidian University, Xiân, China, in 2003 and 2009, respectively.

Since 2006, he has been a Teacher with Xidian University. In 2008 and 2010, he was promoted to Associate Professor and Full Professor with exceptive admission, Xidian University. His current research interests include computational intelligence, with applications to optimization, learning, data mining, and image understanding.

Zhengwen Cao received the B.S. degree in communication and electronics system from Northwest Polytechnical University, Xi'an, China, in 1996.

She is currently an Associate Professor with the School of Information and Technology, Northwest University, Xi'an. Her current research interests include information processing, light communication, and quantum information technology.

Quan Wang was born in 1970. He received the B.S., M.S., and Ph.D. degrees in computer science and technology from Xidian University, Xi'an, China.

He is currently a Full Professor with Xidian University. His current research interests include input and output technologies and systems, image processing, and image understanding.