User Identity Linkage by Latent User Space Modelling

Xin Mu*, Feida Zhu♯, Ee-Peng Lim♯, Jing Xiao†, Jianzong Wang†, Zhi-Hua Zhou*
*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
♯School of Information Systems, Singapore Management University, Singapore, 178902
†Ping An Technology (Shenzhen) Co.,Ltd, China
{mux, zhouzh}@lamda.nju.edu.cn, {fdzhu, eplim}@smu.edu.sg,
{xiaojing661, wangjianzong347}@pingan.com.cn

ABSTRACT

User identity linkage across social platforms is an important problem of great research challenge and practical value. In real applications, the task often assumes an extra degree of difficulty by requiring linkage across multiple platforms. While pair-wise user linkage between two platforms, which has been the focus of most existing solutions, provides reasonably convincing linkage, the result depends by nature on the order of platform pairs in execution with no theoretical guarantee on its stability. In this paper, we explore a new concept of "*Latent User Space*" to more naturally model the relationship between the underlying real users and their observed projections onto the varied social platforms, such that the more similar the real users, the closer their profiles in the latent user space. We propose two effective algorithms, a batch model(ULink) and an online model(ULink-On), based on latent user space modelling. Two simple yet effective optimization methods are used for optimizing objective function: the first one based on the constrained concave-convex procedure(CCCP) and the second on accelerated proximal gradient. To our best knowledge, this is the first work to propose a unified framework to address the following two important aspects of the multi-platform user identity linkage problem — (I) the platform multiplicity and (II) online data generation. We present experimental evaluations on real-world data sets for not only traditional pairwise-platform linkage but also multi-platform linkage. The results demonstrate the superiority of our proposed method over the state-of-the-art ones.

Keywords

User identity linkage; Latent User Space; Social network

1. INTRODUCTION

The problem of *User Identity Linkage* (UIL), which aims to identify the accounts of the same user across different social platforms, has recently been attracting an increasing amount of attention and effort due to both the significant research challenges and the immense practical value of the problem. For example, in [11], Liu et al pointed out *completeness*, *consistency* and *continuity* as three major benefits for user profiling from successful user identity linkage, an essential task in today's social-data-enabled business intelligence. In industry, human-centric data fusion from various sources has become a key component for most leading data intelligence companies such as Palantir[1]. In a nutshell, the ability of integrating data across various platforms down to the granularity of individuals lies at the very core of the data-driven analytical paradigm for business and consumer insight.

However, the methodologies and approaches adopted by the existing solutions have so far fallen short of successfully addressing the following two essential characteristics of this problem.

- *Platform Multiplicity*: The power of user identity linkage lies in piecing up information from multiple sources, typically more than two. However, most existing solutions have focused on pair-wise user linkage between two platforms, i.e., identifying the two accounts ID_A and ID_B for the same user on two platforms A and B respectively. For three or more platforms, existing methods would have to first match users between pairs of platforms and then integrate the matching results to derive the final user linkage across all platforms. Since different orders of such pair-wise platform linkage, as demonstrated by our experiments in Section 6, would lead to different final linkage results, it therefore raises serious concern for the result stability, especially when no theoretical bound has been known as yet.

- *Online Data Generation*: The existing approaches examine a snapshot of two platforms at a certain time point and compute the best possible linkage result with the current data. On the other hand, users generate content continually on social platforms. An intelligent linkage algorithm should be able to take advantage of the incremental data updates to continuously improve the linkage quality with much lower computational cost than re-computing everything again from scratch at every data update.

To better address the two above-mentioned challenges, we introduce in this paper a new concept *Latent User Space*, to more naturally model the reality. The main idea is to take advantage of the fact that, after all, underlying all these

[1]https://www.palantir.com/

KDD '16, August 13-17, 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2939672.2939849

different accounts that we try to link, there does exist this real user as a natural person, if these accounts indeed belong to the same user. We call each such an underlying user a *"user-in-itself"*, borrowing inspiration rooted in western Philosophy [2]. Every user-in-itself corresponds to a point in the latent user space. If a real user has accounts on multiple social platforms, each account is deemed simply as a projection of the underlying " user-in-itself", which we may call it the " user-as-observed". More specifically, all that are observed from the " user-as-observed" on a social platform, i.e., profile, behaviour data, contents, etc., are the projection of the "user-in-itself" constrained by the features and structures provided by the platform.

It follows from this model that when we project data from different platforms back to this space, the data points of the same user should be close to each other (ideally, they should be projected back to a single data point). In essence, the more different the two users, the greater the distance between their data points in the latent user space.

Figure 1 gives an illustration with results on real data. We show four real users each with corresponding accounts on two popular Chinese social platforms, Renren and Weibo, denoted as u_i and $v_i, 1 \leq i \leq 4$ (user profile images are blurred for privacy concerns). When their accounts from the two platforms are projected back to the underlying latent user space, it is clear that accounts belonging to the same user would project back to data points that are much closer to each other than data points from accounts belonging to different users (the values along the edges denote the distances between data points in the latent user space, e.g., the distance between u_1 and v_1 is 0.09). The details of distance calculation in the latent user space are given in Section 4.

An important feature of our work is that, compared with previous work on UIL problem, our proposed *Latent User Space* frees the model from focusing on either the design of distance rules or building models depending on specific data forms, but rather on examining the intrinsic structure of user. While latent space has been introduced for analyzing dynamic social networks [19]. to our best knowledge, our work is the first to apply the latent user space for UIL problem across multiple social platforms.

Based on the *Latent User Space* concept, we propose ULink, a multi-platform linking user identity framework based on modeling latent user space, and ULink-On, an online framework for the same task. In ULink framework, we build the *Latent User Space* through projection matrix, and address this problem by jointly optimizing objective function with matching pair information, non-matching pair information and intra-platform relation constraints across different platforms. Inspired by Marginal Structured SVM, two efficient methods based on the concave-convex procedure (CCP) and accelerated proximal gradient(APG) are applied for solving the optimization problem. We further propose an online learning framework(ULink-On) by considering constrain of batch model. We conduct empirical studies on real social network data to show the effectiveness and efficiency of our approach.

[2]A notion in the Philosophy of Immanuel Kant, a "thing-in-itself" is what a thing really is as different from how it appears to us — an object as it would appear to us if we did not have to approach it under the conditions of space and time.

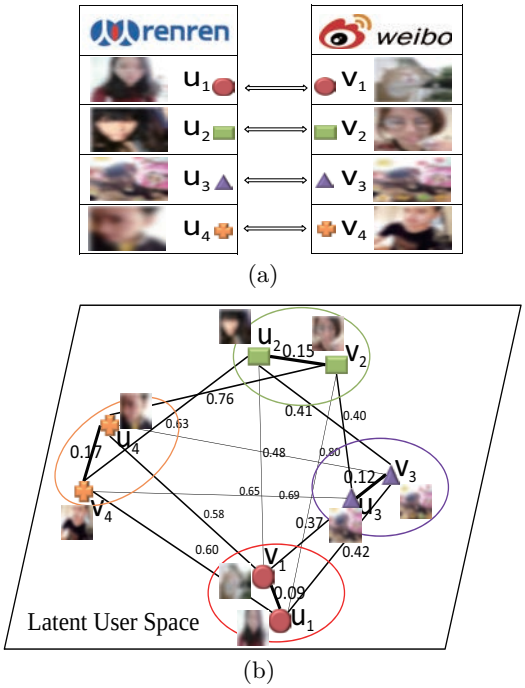


Figure 1: An illustration of latent user space. (a) Four users in Renren and Weibo data. (b) Latent user space

We summarize our key contributions as follows:

- We propose a new model for the multi-platform user identity linkage problem based on a new concept of *Latent User Space*, which more naturally models the relationship among the underlying real user and the various accounts belonging to her on different platforms. It goes beyond pair-wise platform user linkage treating user attributes (user features) as main direction to focusing on the intrinsic structure of the underlying user, which is particularly powerful in linkage settings with multiple platforms.

- To take advantage of the continual online generation of user data on social platforms, we extend our batch framework ULink to propose an online version, called ULink-On, which is able to take advantage of the incremental data updates to continually improve the linkage quality. We also develop efficient optimization for ULink-On.

- We conduct experiments on real-world data sets to comprehensively evaluate the performance of our proposed algorithms. For both pairwise platform and multi-platform settings, our algorithm have consistently outperformed the state-of-the-art existing methods with greater stability. We provide discussions for some important aspects of our framework for future exploration.

The rest of this paper is organised as follows: Section 2 examines the related work. We introduce the proposed framework in Section 4 and Section 5. The experimental evaluation is detailed in Section 6. We also give a discussion in Section 7 and conclude the paper in Section 8.

2. RELATED WORK

A closely-related problem long studied by database community is that of *Record Linkage*, which aims to find records in a data set across different data sources that refer to the same entity. The concept of modern record linkage originated from geneticist Howard Newcombe, who introduced odds ratios of frequencies and the decision rules for delineating matches and non-matches[15, 16]. A large number of algorithms, both supervised and unsupervised, have been developed in recent years to solve the record linkage problem, which can be grouped mainly into two types: deterministic linkage and probabilistic linkage. The former approach, which is often rule-based and strives for exact one-to-one matching of user name and other user attributes[17, 5], usually works well for simple linkage problems or in the presence of special domain knowledge of the matching. Probabilistic linkage [18], on the other hand, assigns probabilistic weighting to records and accepts record pairs with sufficiently high weights as linked pairs. [4] provided the formal mathematical foundations and some theoretical analysis. Despite the similarity with the record linkage problem, the UIL problem that we consider in this paper distinguishes itself with unique characteristics of social data to make possible breakthroughs previously unattainable.

The *User Identity Linkage* problem was initially formalized as connecting corresponding identities across communities in [27], and was addressed with a web-search-based approach. Considering social network diversity and information asymmetry, many early works were proposed based on user information, including user-profile-based, user-generated-content-based and user-behavior-model-based. User-profile-based methods collect tagging information provided by users [7] or user profiles, e.g., user-name, description, location, etc. [24, 10, 29]. User-generated-content-based ones collect personal identifiable information from user personal reading records[1] or user-generated content. User-behavior-model-based methods [28] analyze behavior patterns and build feature models from user names, language and writing styles. As most of these algorithms are often tailored to a particular pattern, they face serious challenges in identifying cross-platform linkage if required data patterns are not available on all platforms.

More recent approaches have been proposed in both supervised and unsupervised learning frameworks. [11] proposed a supervised multi-objective learning framework to link up user accounts of the same natural person across different social network platforms. [9] studied link prediction methods for homogeneous networks based on massive unsupervised link indicators. To solve the collective link identification problem, [30] proposed a unified link prediction framework. [31] studied the multi-network link prediction problem across partially aligned networks with a PU link prediction framework. However, Most existing solutions have focused on pair-wise user linkage between two platforms. Even though a few of them can handle multiple platforms, the computation complexity is too high for practical applications and the models tend to depend on specific data forms, e.g., location and friendship.

Other relevant approaches include *subspace learning-based* approaches [25], an important learning framework in multi-view learning which aims to obtain a latent subspace shared by multiple views by assuming that the input views are generated from this subspace. The structured support vector machine[23] is a machine learning algorithm that generalizes the Support Vector Machine (SVM) classifier. [21] developed a method for structured margin classification, and an online framework was proposed by [14].

Before introducing the detail of our proposed framework, we will give the formal definitions of many important concepts.

Table 1: Notations

SYMBOL	DESCRIPTION
$\mathbb{O}$	The set of real users in LUS
S_i	i^{th} social media platform
$\mathcal{P}_i$	The set of users on S_i
d	The user feature dimension in LUS
o_i	i^{th} user in LUS, $o_i \in R^d$
n_i	The number of users on S_i
m_i	The user feature dimension on S_i
u_j^i	j^{th} user on S_i, $u_j^i \in R^{m_i}$
w_i	The projection matrix for S_i, $w_i \in R^{d \times m_i}$

3. PROBLEM FORMULATION

We formulate our problem in this section by first introducing the concept of *latent user space* as follows.

DEFINITION 3.1. ***[Latent User Space (LUS)]*** *We define the* Latent User Space (LUS) *as a triple* $(\mathbb{O}, \mathbb{A}, \mathbb{D})$ *where* $\mathbb{O} = \{o_1, o_2, \ldots, o_N\}$ *is the set of all* N *real users each corresponding to a natural person,* $\mathbb{A} = (a_1, a_2, \ldots, a_d)$ *denotes the vector of* d *attributes by which every real user is represented, i.e.,* $o_i = (a_1^i, a_2^i, \ldots, a_d^i), 1 \leq i \leq N$, *and* $\mathbb{D}$ *represents the distance function such that* $\mathbb{D}(o_i, o_j)$ *is the distance between any two users* $o_i, o_j \in O$.

We denote a set of e different social media platforms as $\mathcal{S}$ $=\{S_1, S_2, \ldots, S_e\}$, and for each $S_i \in S$, $S_i = (\mathcal{P}_i, \mathcal{F}_i)$ where $\mathcal{P}_i = \{u_1, u_2, \ldots, u_{n_i}\}$ denotes the set of all user accounts on S_i and $\mathcal{F}_i = (f_1, f_2, \ldots, f_{m_i})$ denotes the feature vector to represent each user such that $u_j = (f_1^j, f_2^j, \ldots, f_{m_i}^j)$ for $1 \leq j \leq n_i$.

We refer to every user x in LUS as a "user-in-itself". For any platform S_i, we refer to every user u on S_i as a "user-as-observed", which corresponds to a "user-in-itself" x in LUS through the projection function of S_i as defined below.

DEFINITION 3.2. ***[Projection Function]*** *We denote as* Φ_i *the* projection function *of* S_i *such that for each* $o_j \in O$ *in latent user space, we have* $\Phi_i(o_j) = \Phi_i((a_1^j, a_2^j, \ldots, a_d^j)) = u_k^i$, $u_k^i \in \mathcal{P}_i$. *We also denote as* Φ_i^{-1} *the inverse function of* Φ_i *such that* $\Phi_i^{-1}(\Phi_i(o)) = o$ *holds for all* $o \in O$ *and* $1 \leq i \leq e$.

Notice that in general, the projection function Φ_i is unknown to us for a given social platform S_i. The user identity linkage problem defined for multiple platforms is given as follows. It is clear that definitions for the same problem for two platform case as in [11] is just a special case of this more general definition.

DEFINITION 3.3. ***[Multi-platform User Identity Linkage (MUIL)]*** *Given the latent user space* $(\mathbb{O}, \mathbb{A}, \mathbb{D})$, *a set of* e *social media platforms* $\mathcal{S} = \{S_1, S_2, \ldots, S_e\}$ *where each*

$S_i = (\mathcal{P}_i, \mathcal{F}_i)$, *the problem of* Multi-platform User Identity Linkage (MUIL) *is to find a binary function* f *such that for any given vector* $\vec{u}$ *of user accounts* $\vec{u} = (u^1, u^2, \ldots, u^e), u^i \in \mathcal{P}_i, 1 \leq i \leq e$

$$f(\vec{u}) = \begin{cases} 1, & if \exists\ o \in \mathbb{O},\ s.t.\ u^i = \Phi_i(o),\ 1 \leq i \leq e \\ 0, & otherwise \end{cases}$$

The binary function f as in Definition 3.3 decides perfectly if a set of user accounts on various social platforms correspond to the same real user. In reality, however, such an ideal function is hard to identify as both the latent user space and true projection functions Φ_i are unknown. Our approach in this paper is therefore to turn the MUIL problem into an optimization problem by the intuition that *the more similar the two real users* o_a, o_b *in latent user space, the smaller the distance when they are projected back from the social platforms to the latent user space*, i.e., $\mathbb{D}(\Phi_i^{-1}(\Phi_i(o_a)), \Phi_j^{-1}(\Phi_j(o_b)))$ for all $1 \leq i, j \leq e$. Hence the following optimization version of the MUIL problem.

Given the latent user space $(\mathbb{O}, \mathbb{A}, \mathbb{D})$, a set of e social media platforms $\mathcal{S} = \{S_1, S_2, \ldots, S_e\}$ where each $S_i = (\mathcal{P}_i, \mathcal{F}_i)$, we solve the MUIL problem by finding a set of projection functions $\Phi_i, 1 \leq i \leq e$ such that for any given vector of user accounts $(u^1, u^2, \ldots, u^e), u^i \in \mathcal{P}_i, 1 \leq i \leq e$ corresponding to the same real user, i.e., $\exists o \in \mathbb{O}$ such that $u^i = \Phi_i(o)$ for $1 \leq i \leq e$. We search for projection functions Φ_i for the MUIL problem by minimizing following objective function:

$$\min_{\Phi^{-1}} \sum_{1 \leq i,j \leq e} \mathbb{D}(\Phi_i^{-1}(u^i), \Phi_j^{-1}(u^j)) \tag{1}$$

where u^i and u^j are same user on S_i and S_j.

Considering that fully aligned networks hardly exist in the real world, in this paper, we also adopt the assumption of partially aligned social platforms as proposed in [31]. Table 1 summarizes the notations in this paper.

4. PROPOSED METHOD

4.1 ULink Framework

Eqn.(1) is a direct way to model LUS to obtain inverse projection function Φ^{-1}. We would further consider the user relation in both LUS and the original space in our proposed ULink framework.

Let $\{u_l^i, u_{\rho(l)}^j\}^L$ be a set of same user pairs(matching pairs) for any two social media platforms S_i and S_j, $\rho(\cdot)$ is an index mapping function to represent $\rho(l)^{th}$ user in S_j matching l^{th} user in S_i. Let $\{u_l^i, u_k^j\}^{UL}$ be a set of different user pairs(non-matching pairs). Following the definition 3.3, we aim to obtain all projection matrix w_z for each inverse function projection Φ^{-1} given same user pairs and different user pairs. The proposed framework ULink is to minimize objective function such that

$$\begin{aligned} &\min_{w,\xi} \frac{1}{2}(\sum_{z=1}^{e} ||w_z||_{\mathrm{F}}^2) + C\sum \xi \\ &s.t.\ \ \mathbb{D}(\Phi_i^{-1}(u_l^i), \Phi_j^{-1}(u_k^j)) - \mathbb{D}(\Phi_i^{-1}(u_l^i), \Phi_j^{-1}(u_{\rho(l)}^j)) \\ &\geq B\delta(u_k^j, u_{\rho(l)}^j) - \xi_{k\rho(l)}, \forall i, j, l, k \\ &i, j \in \{1, 2, \cdots, e\}, i \neq j; l \in \{1, 2, \cdots, n_i\}, \\ &k, \rho(l) \in \{1, 2, \cdots, n_j\}, \rho(l) \neq k; \xi \geq 0 \end{aligned} \tag{2}$$

where, e represents the number of social platform, ξ is a slack variable. $\delta(\cdot)$ is a flexible constant which is regarded as intra-platform relation in original space. B and C is the coefficient. Since the positive of the right side of constraint always make same user be close to each other, and different user be separated from each other in LUS. In particular, the greater the value of the difference between users in original space $\delta(\cdot)$, the more apparent this relation.

Specifically in this work, we take the Euclidean distance as the distance function $\mathbb{D}$. i.e., $\mathbb{D}(\Phi_i^{-1}(u_l^i), \Phi_j^{-1}(u_k^j)) = ||u_l^i w_i^T - u_k^j w_j^T||_2^2$. Euclidean distance is also considered for $\delta(\cdot)$ throughout this work, i.e., $\delta(u_k^j, u_{\rho(l)}^j) = ||u_k^j - u_{\rho(l)}^j||_2^2$.

For ease of exposition, we can formulate Eqn.(2) on two platforms. x and y are used for representing the user on two platforms such as u^1 and u^2, $x \in R^{m_1}$, $y \in R^{m_2}$. As mentioned above, $\{x_i, y_k\}^{UL}$ is the set of non-matching pairs, and $\{x_i, y_{\Phi(i)}\}^{L}$ is the set of matching pairs. The Eqn.(2) becomes:

$$\begin{aligned} &\min_{w_1, w_2, \xi} \frac{1}{2}(||w_1||_{\mathrm{F}}^2 + ||w_2||_{\mathrm{F}}^2) + C\sum_i \sum_k \xi_{ik} \\ &s.t.\ \ ||x_i w_1^T - y_k w_2^T||_2^2 - ||x_i w_1^T - y_{\rho(i)} w_2^T||_2^2 \\ &\geq B\delta(y_{\rho(i)}, y_k) - \xi_{ik},\ \forall i, k; \rho(i) \neq k \\ &i \in \{1, 2, ..., N\}, k, \rho(i) \in \{1, 2, ..., M\}, \xi_{\rho(i)k} \geq 0 \end{aligned} \tag{3}$$

where M and N are the number of users on two platforms. Ideally, we should consider all non-matching pairs for modeling. However, this would result in exponential computational cost with the number of non-matching pairs. Therefore, we select a limited number of non-matching pairs as experimental set in this paper. We also give an analysis and discuss some feasible solutions for this problem in Section 7.

For convenience, we combine variables w_1, w_2 to $W = \begin{bmatrix} w_1^T \\ w_2^T \end{bmatrix}$, $W \in R^{(m_1+m_2)\times d}$ and matching pair vector $d^l = \begin{bmatrix} x_i & -y_{\Phi(i)} \end{bmatrix}$, non-matching pair vector $d^{ul} = \begin{bmatrix} x_i & -y_k \end{bmatrix}$, d^{ul}, $d^l \in R^{m_1+m_2}$. Therefore, the optimization problem Eqn.(3) can be rewritten as

$$\begin{aligned} &\min_{W, \xi_{ik}} \ \ \frac{1}{2}||W||_{\mathrm{F}}^2 + C\sum_i \sum_k \xi_{ik} \\ &s.t.\ ||d^{ul}W||_2^2 - ||d^l W||_2^2 \geq B\delta(y_{\rho(i)}, y_k) - \xi_{ik}, \forall i, k; \rho(i) \neq k \\ &i \in \{1, 2, ..., N\}, \rho(i), k \in \{1, 2, ..., M\}, \xi_{ik} \geq 0 \end{aligned} \tag{4}$$

It is a non-trivial task to solve Eqn.(4), because the constrains of Eqn.(4) are no longer convex, and the minimization is not a convex problem. However, it is interesting to note that our objective function is very similar to the state-of-the-art framework structural SVMs[8], which is to learn the classifier w:

$$\begin{aligned} &\min_{w, \xi} \ \ \Omega(w) + C\sum \xi_i \\ &s.t. \quad w^T[\Psi(x_i, \overline{y}_i) - \Psi(x_i, y_i)] \geq \delta(\overline{y}_i, y_i) - \xi_i, \forall i \end{aligned}$$

where, the structured input-output pairs $(x, y) \in X \times Y$, X and Y are the spaces of the input and output variables, $\delta(\cdot)$ is a loss function that quantifies the loss associated with predicting $\overline{y}$ when y is the correct output value. Furthermore, $\Psi(\cdot)$ is a joint feature vector that describes the relationship between input x and structured output y, $\Omega(\cdot)$ is regarded as regular term and ξ_i is a slack variable. Inspired by this work, we adopt two simple yet effective strategies

for handling this optimization problem. One is based on the constrained concave-convex procedure(CCCP) used in [20], and the second is a gradient descent algorithm(accelerated proximal gradient[22]). The details will be given as follows.

4.2 Optimization

Smola et.al. [20] provide a strategy to use the constrained concave-convex procedure for constrained problems. The idea of the concave-convex procedure (CCP) can also be applied to the optimization problem of Eqn.(4).

Denote by f_i, g_j real-valued convex and differentiable functions on a vector space $\mathcal{X}$ for all $i \in \{0, \ldots, n\}$, and let $c_i \in R$ for $i \in \{1, \ldots, n\}$. Then, the Constrained Concave Convex Procedure is defined:

$$\begin{aligned} &\min_x f_0(x) - g_0(x) \\ &s.t. \quad f_i(x) - g_i(x) \leq c_i, \forall i \end{aligned}$$

Denote by $T_n\{f, x\}(x')$ the n^{th} order Taylor expansion of f at location x, that is, $T_1\{f, x\}(x') = f(x) + < x' - x, \nabla f(x) >$. Thus, the above optimization problem can be replaced by finding x_{t+1} as the solution to the convex optimization problem until the convergence of x_t:

$$\begin{aligned} x_{t+1} = &\min f_0(x) - T_1\{g_0, x_t\}(x) \\ &s.t. \quad f_i(x) - T_1\{g_i, x_t\}(x) \leq c_i, \forall i \end{aligned}$$

Note that [20] presents the proof of its convergence, and shows this algorithm can be customized to various cases to efficiently solve the optimization problem.

It is clear that Eqn.(4) satisfies the conditions of Constrained CCP. we define:

$$\begin{aligned} f_0(W) &= \frac{1}{2}||W||_{\mathrm{F}}^2 + C\sum_i \sum_k \xi_{ik} \\ f_i(W) &= B\delta(y_{\rho(i)}, y_k) - \xi_{ik} + ||d^l W||_2^2 \\ g_i(W) &= ||d^{ul} W||_2^2, \quad g_0(W) = 0 \end{aligned} \tag{5}$$

thus, each iteration requires solving the following optimization problem:

$$\begin{aligned} &W_{t+1} = \min_{W, \xi_{ik}} \quad \frac{1}{2}||W||_{\mathrm{F}}^2 + C\sum_i \sum_k \xi_{ik} \\ &s.t. \quad 2 * d^{ul} W_t W^T (d^{ul})^T - d^l W W^T (d^l)^T - d^{ul} W_t W_t^T (d^{ul})^T \\ &\geq B\delta(y_{\rho(i)}, y_k) - \xi_{ik}, \forall i, k; \rho(i) \neq k \\ &i \in \{1, 2, ..., N\}, \rho(i), k \in \{1, 2, ..., M\}, \xi_{ik} \geq 0 \end{aligned} \tag{6}$$

Since Eqn.(6) is a convex optimization problem, a quadratically constrained quadratic program (QCQP) can be used to solve it. We use CVX: Matlab Software for Disciplined Convex Programming[6] to optimize this function. In summary, the sketch of the optimization process is described in Algorithm 1.

Algorithm 1 ULink-CCP

1: **initialize:** W_0 with a random value , B, C - parameters
2: $W_t = W_0$
3: **repeat**
4: find W_{t+1} as the solution of the optimization problem in Eqn.(6)
5: **until** convergence of W_t
6: Obtain w_1 and w_2 by W_{t+1}

Another effective optimal algorithm Accelerated Proximal Gradient (APG)[22] is used for solving our problem as follows.

According to Eqn.(4), we define a symmetric positive semi-definite matrix Q : $Q = WW^T$, $Q \in R^{(m_1+m_2)\times(m_1+m_2)}$. Thus, Eqn.(4) can be transformed to the following problem:

$$\begin{aligned} &\min_{Q, \xi_{ik}} \quad \frac{1}{2}\mathrm{Tr}(Q) + C\sum_i \sum_k \xi_{ik} \\ &s.t. \quad (d^{ul})Q(d^{ul})^T - (d^l)Q(d^l)^T \geq B\delta(y_{\rho(i)}, y_k) - \xi_{ik} \\ &\forall i, k; \rho(i) \neq k, i \in \{1, 2, ..., N\}, \rho(i), k \in \{1, 2, ..., M\} \\ &\xi_{ik} \geq 0 \end{aligned} \tag{7}$$

Note that any feasible(or optimal) solution to Eqn.(7) gives a feasible (or optimal) solution to Eqn.(4), and vice versa[32].

We can apply the accelerated proximal gradient (APG) method[12] to efficiently solve the primal form of Eqn.(7). Let $p(Q) = \frac{1}{2}\mathrm{Tr}(Q)$ and $f(Q) = C\sum_i \sum_k \xi_{ik}$. $\xi_{ik} = \max\{0, B\delta(y_{\Phi(i)}, y_k) + (d^l)Q(d^l)^T - (d^{ul})Q((d^{ul})^T\}$. We define: $F(Q) = f(Q) + p(Q)$. The derivative of f is denoted by ∇f. [26] shows that ∇f is Lipschitz continuous on Q. For any symmetric positive semi-definite matrix Z, consider the following QP problem of $F(Q)$ at Z:

$$\begin{aligned} A\tau(Q; Z) &= f(Z) + < \nabla f(Z); Q - Z > \\ &\quad + \frac{\tau}{2}||Q - Z||_{\mathrm{F}}^2 + p(Q) \\ &= \frac{\tau}{2}||Q - G||_{\mathrm{F}}^2 + p(Q) + f(Z) + \frac{1}{2\tau}||\nabla f(Z)||_{\mathrm{F}}^2 \end{aligned} \tag{8}$$

where $\tau > 0$ is a constant and $G = Z - \frac{1}{\tau}\nabla f(Z)$. To minimize $A\tau(Q; Z)$ w.r.t. Q, it is reduced to following :

$$\arg\min_Q \quad \frac{\tau}{2}||Q - G||_{\mathrm{F}}^2 + p(Q) \tag{9}$$

Thus, take the derivative of the objective function, and get $Q = G - \frac{1}{2\tau}I$. Note that G can be decomposed by SVD as $G = U\overline{G}U^T$, and $Q = U\overline{G}U^T - \frac{1}{2\tau}UU^T$, then $Q = U(\overline{G} - \frac{1}{2\tau}I)U^T$. We use 0 to replace the negative entries in $\overline{G} - \frac{1}{2\tau}$. Finally, the projection matrix W can be obtained by symmetric positive semi-definite matrix Q. Note that convergence criteria for this optimal solution was given in [12], which is a similar algorithm.

5. FROM BATCH TO ONLINE

An intelligent linkage algorithm should be able to take advantage of the incremental data updates to continuously improve the linkage quality. In this section, we extend our batch framework(ULink) to an online learning framework (ULink-On), and formalize our online framework(ULink-On) based on Eqn.(7).

Note that we assume one matching pair $(x_t, y_t)^L$ and one non-matching pair $(x_t, y_t')^{UL}$ would arrive at every time stamp t. As mentioned before, let d_t^l and d_t^{ul} be a pair of same user and a pair of different users at time t. We consider the objective function scale quadratically with ξ as follows:

$$\begin{aligned} Q_{t+1} = &\min_{Q, \xi} \frac{1}{2}||Q - Q_t||_{\mathrm{F}}^2 + \frac{1}{2}C\xi_t^2 \\ &s.t. \quad (d_t^{ul})Q(d_t^{ul})^T - (d^l(t))Q(d_t^l)^T \\ &\geq B\delta(y_t', y_t) - \xi_t \end{aligned} \tag{10}$$

Like Online Passive-Aggressive algorithm[3], the objective function in Eqn.(10) attempts to keep the norm of the change to the parameter vector as small as possible on each update, while incorporating the assumption of LUS.

Before optimize Eqn.(10), we need to initialize a symmetric positive semi-definite matrix Q_t. Thus, the Lagrangian of the optimization problem Eqn.(10) is defined as:

$$\begin{aligned} L(Q,\xi) =& \frac{1}{2}||Q-Q_t||_{\mathrm{F}}^2 + \frac{1}{2}C\xi_t^2 \\ &+ \beta(B\delta(y_t', y_t) - \xi_t \\ &+ (d_t^l)Q(d_t^l)^T - (d_t^{ul})Q(d_t^{ul})^T) \end{aligned} \tag{11}$$

where $\beta \geq 0$ is a Lagrange multiplier. Setting the partial derivatives of L with respect to the elements of Q to zero, this yields:

$$Q = Q_t - \beta H, \quad H = (d_t^l)^T(d_t^l) - (d_t^{ul})^T(d_t^{ul})$$

Setting the partial derivatives of the Lagrangian with respect to ξ and setting that partial derivative to zero :

$$\frac{\partial L(Q,\xi)}{\partial \xi} = C\xi_t - \beta = 0, \quad \xi_t = \frac{\beta}{C}.$$

we can rewrite Eqn(11) as,

$$\begin{aligned} L(Q,\xi) =& \frac{1}{2}||Q_t + \beta H - Q_t||_{\mathrm{F}}^2 + \frac{1}{2}C(\frac{\beta}{C})^2 \\ &+ \beta(B\delta(y_t', y_t) - \frac{\beta}{C} \\ &+ (d_t^l)Q(d_t^l)^T - (d_t^{ul})Q(d_t^{ul})^T) \end{aligned}$$

Setting the derivative β of the above to zero, this yields

$$\beta = \frac{V + B\delta(y_t', y_t)}{2Z - ||H||_{\mathrm{F}}^2 + \frac{1}{C}} \tag{12}$$

where,

$$\begin{aligned} Z &= (d_t^l)H(d_t^l)^T - (d_t^{ul})H(d_t^{ul})^T \\ V &= (d_t^l)Q_t(d_t^l)^T - (d_t^{ul})Q_t(d_t^{ul})^T \end{aligned} \tag{13}$$

As described above, the pseudo-code for this algorithm is given in Algorithm 2

Algorithm 2 ULink-On

Input: d_t^l, d_t^{ul} - pairwise data; B, C - parameters
Output: Q

1: **initialize:** Q_t - symmetric positive semi-definite matrix
2: **for** t=1,2,$\cdots$ **do**
3: Calculate Z and V use (13)
4: Calculate β use (12)
5: Calculate $Q = Q_t - \beta H$, where $H = (d_t^l)^T(d_t^l) - (d_t^{ul})^T(d_t^{ul})$.
6: Q_t=Q
7: **end for**

Note that it is often the case that more than two platforms are involved for user linkage in the real applications. Yet, most previous works have focused on pair-wise user linkage problem. If a third platform is needed to link with the existing platforms, many algorithms may suffer from optimization problem. For proposed batch model (4) and online model (10), combining the alternative optimization technique into the CCP framework can be adopted to handle this problem, i.e., we optimize one variable w_1 by using the fixed other values w_3 and w_2. One salient feature of our model is that we directly connect multiple platforms by considering diverse connection relationship, instead of integrating results from pair-wise connections. Note that the optimization problem has been turned into one with one variable optimization, such that many algorithms can be used to solve this problem. In a nutshell, the sketch of the optimization process for proposed model is easy to be adapted to multiple social platforms, as demonstrated in our experiments with three platforms in Section 6.2.1.

6. EXPERIMENTAL EVALUATION

6.1 Experimental Setup

Data Sets. We use the following four real data sets to assess the performance of all methods in comparison:

- *Weibo* (http://www.weibo.com/): Weibo is one of the most popular Chinese micro-blogging websites with 450 million active users, akin to a hybrid of Twitter and Facebook.
- *Renren* (http://www.renren.com/): Renren is a leading real-name social networking Internet platform in China, often dubbed as the Facebook of China with 162 million registered users.
- *36.cn* (http://www.36.cn/): 36.cn is an online job hunting service in China serving more than 100 thousand businesses with user-uploaded resumes.
- *Zhaopin* (http://www.zhaopin.com/): Zhaopin is another publicly-listed company providing online job hunting service in China with more than 22 million resumes.

For Weibo and Renren, the ground-truth user linkage pairs across these two platforms are manually annotated. For the other three platforms (Renren, 36.cn, and Zhaopin), the ground-truth user linkage across the three platforms are provided by our industrial partner who have access to the users' real names and emails. A summary of the ground truth information is given in Table 2.

Table 2: A summary of cross-platform ground-truth user linkage.

Data Set	Weibo	Renren	36.cn	Zhaopin
Weibo	NA	2186	NA	NA
Renren	2186	NA	11268	9495
36.cn	NA	11268	NA	2698
Zhaopin	NA	9495	2698	NA

Renren & Zhaopin & 36.cn
835

Competing Algorithms. To evaluate the performance of ULink, we chose three state-of-the-art supervised classifiers — HYDRA[11], COSNET[33] and SVM[2] — and one non-parametric method of KNN, explained as follows:

1. **HYDRA** [11]:a large-scale social identity linkage framework via heterogeneous behavior modeling which learns the mapping function by multi-objective optimization incorporating both supervised learning on pair-wise ID linkage information and the cross-platform structure consistency maximization.

Table 3: A summary of user features used for each data set.

Data Set	User Features
Weibo	gender; birthday; location; educational background
Renren	gender; nationality; birthday; location; educational background
36.cn	gender; nationality; birthday; marital status; degree; work experience
Zhaopin	gender; birthday; mailing address; educational background

2. **COSNET** [33]: an algorithm that addresses the UIL problem by considering both local and global consistency (network structure) among multiple networks, which is useful in our setting as the requirement of global consistency of network structure is not satisfied for some data sets. The training set is composed of linked pairs and unlinked pairs. An efficient subgradient algorithm is developed to train the model by converting the original objective function into its dual form.

3. **SVM** [2]:a binary prediction on user pairs using support vector machines on the proposed similarity calculation schemes for pairwise linkage setting. The training data is composed of linked pairs and unlinked pairs, which is represented by 1 and -1 as the label respectively.

4. **KNN**: We use K-Nearest-Neighbor (KNN) as a non-parametric method as follows. When matching users between two platforms S_i and S_j, we take the user feature vectors of both platforms to form a unified feature vector. For each testing user u^i on S_i, we use KNN to generate $k = 5$ nearest users on S_j as matching candidates. The final linkage is the result of majority voting.

5. **ULink-CCP**: our batch model with Constrained Concave Convex Procedure optimization method.

6. **ULink-APG**: our batch model with Accelerated Proximal Gradient Update optimization method.

7. **ULink-On**: our online version of the ULink model.

Experiment Settings. Table 3 lists the information used for each social platform. We adopt the bag-of-words model for raw text data processing, and replace with the value of 0 for missing attributes. All methods are executed in the MATLAB environment with the following implementations: LIBSVM package[2] is used for modeling SVM; The codes for both HYDRA and COSNET are developed based on the original papers. We employ K-Nearest-Neighbor as predictive classifier in LUS.

Experiments are conducted for both pairwise and multi-platform (e.g., three platforms) linkage settings. Each experiment is repeated for 10 times and both the mean and the standard variance of the performance are reported. The ground-truth linked pairs are divided into 5 folds every time, 4 folds being the training set and 1 fold being the testing set. In the training set and testing set, non-matching pairs are randomly sampled by setting two different ratios, 1:5 and 1:10, between the ground-truth matching pairs to non-matching pairs. It is easy to set parameter B by 10^n, $n \in \mathbb{Z}$. A guide of setting d is mentioned in section 7. The coefficient C in our algorithm, SVM and COSNET is selected via cross validation on the training data. For HYDRA, the parameter p, which determines how the model learned approximates the *Utopia* solution, is set as 5 according to the original paper. The two parameters, γ_L and γ_M, which determine the relative importance of the problems in HYDRA framework from a decision maker's perspective, are set by tuning on the validation set. For COSNET, the matching graph is generated with the relation between users.

Evaluation Metrics. A well-established and widely-used evaluation metric in many real user linkage applications is to compare the top-k candidates for user linkage. In this paper, we set $k = 5$ and evaluate all methods by computing top-k precision for each test user as follows:

$$h(x) = \frac{k - (hit(x) - 1)}{k}.$$

where $hit(x)$ represents the position of correct linked user in the returned top-k users. Then precision, represented by the symbol "*hit-precision*", is calculated on N test users by $\frac{\sum^N h(x_i)}{N}$. For example, given the result of top-k users $\{y_1, y_2, .., y_k\}$ for test data x, if y_1 hits ground truth, $hit(x) = 1$, and $h(x) = 1$. Similarly, if y_4 hits the ground truth, $hit(x) = 4$, and $h(x) = \frac{k-3}{k}$. For the multiple platforms, average "*hit-precision*" will be report.

6.2 Experimental Results

We first evaluate our algorithm for the batch data setting in Subsection 6.2.1, including both the pair-wise platform case and multi-platform case, and then for the online data setting in Subsection 6.2.2.

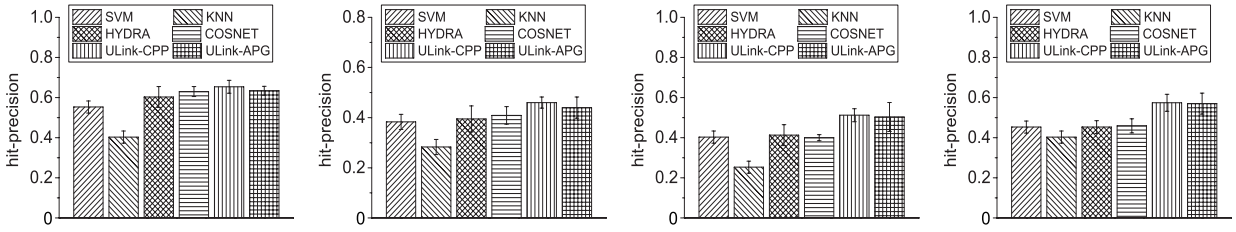
6.2.1 Batch Data Setting

Pairwise Platform Case. This section illustrates the results of the user linkage problem for pairwise platform case on four real-world data sets. Figure 2 and Figure 3 respectively show the performance on different ratio of unlinked pairs and linked pairs.

Summary. Our proposed ULink models — both ULink-CCP and ULink-APG — have consistently produced higher *hit-precision* in all data sets than any other method, with noticeable leading advantage over the rest except for the "Weibo & Renren" case. Among other competing methods, COSNET and HYDRA, both of which are partially based on the structure of SVM each with their own advance, show better performance than SVM in some data sets. While KNN needs no training and runs faster, its performance fell behind others in all sdata sets.

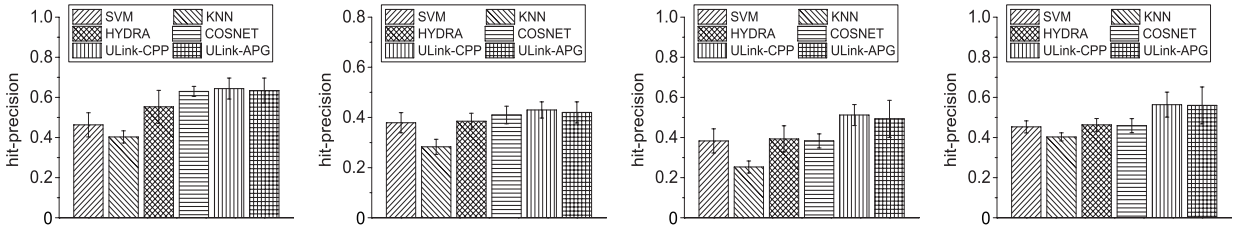
Detailed Analysis.

- HYDRA learns the linkage function via optimizing two objective functions, i.e., the supervised learning using the reliable ground truth, and the structure consistency maximization by modeling the core social network behavior consistency. Its performance, as demonstrated by our experiments, hinges heavily upon the availability of the consistent structure of friendship



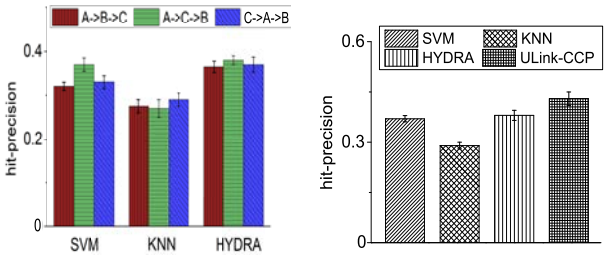
(a) Weibo & Renren (b) Renren & 36.cn (c) Renren & Zhaopin (d) 36.cn & Zhaopin

Figure 2: Pair-wise platform user linkage comparison for batch data setting (with ratio between the ground-truth matching pairs to non-matching pairs being 1:5).



(a) Weibo & Renren (b) Renren & 36.cn (c) Renren & Zhaopin (d) 36.cn & Zhaopin

Figure 3: Pair-wise platform user linkage comparison for batch data setting (with ratio between between the ground-truth matching pairs to non-matching pairs being 1:10).



(a) (b)

Figure 4: User linkage on three platforms. (a)Result of building model with different connection orders. A: Renren, B: 36.cn, C: Zhaopin. (b) Result of user linkage on three platforms.

network across platforms. It performed worse than our ULink model in the three cases where such consistent structure is not available, and excelled for the Renren-Weibo case, the only one where its hypothesis is supported. For example, in online job-hunting data sets of Zhaopin and 36.cn where users are independent and observed social links are weak, the absence of the constraint of structure consistency critical for objective function optimization resulted in its degraded performance similar as *SVM*. On the other hand, it is also worth noting that the pre-computation of affinity scores as a prerequisite for the building model imposes extra computation time upon HYDRA.

- COSNET is found to be in a similar situation as HYDRA. Its performance could rival ours in data sets where the matching graph based on friend relationship is available, as in Renren-Weibo case, yet it fared not as well in all other data sets.

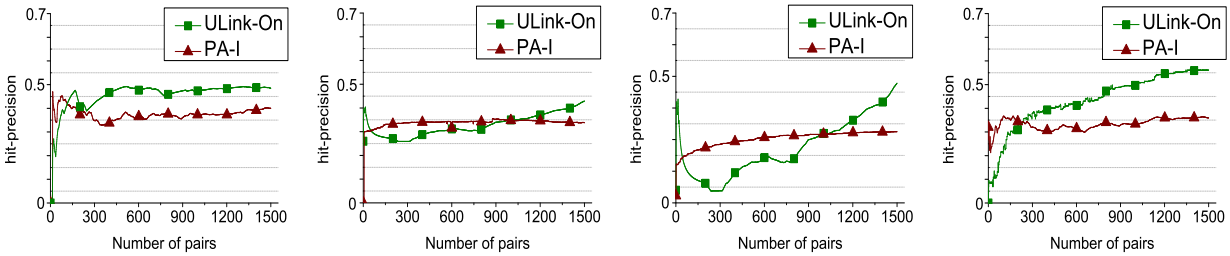
- SVM presents in general an average performance in all cases. The more important reasons why it is not a good choice for the MUIL problem are that, while it enjoys easy deployment for pairwise linkage classification, it suffers from a number of challenging issues including the high computational complexity if using Gaussian kernel, the difficulty in finding right parameters and missing values.

- KNN results in the worst performance in general although it is the simplest and fastest among all with no training required. On the other hand, this illustrates the significance of our proposed concept of latent user space (LUS) because, while applying KNN directly in the space defined by the feature vectors of the linking social platforms has been shown to work poorly, our ULink methods do achieve the best performance by applying KNN in the LUS.

- ULink-CCP and ULink-APG achieve the best performance in most of the data sets, making them in general the best choices for the MUIL problem. The factors of consideration when choosing between them are (I) ULink-APG is a better choice in terms of time complexity when the number of dimensions is high; and (II) The influence of initial condition of ULink-CCP is smaller than ULink-APG.

Multi-Platform Case. We demonstrate in this part why existing solutions suffer from inherent defects when solving the user identity linkage problem on more than two platforms, driving home the importance of a new framework like our proposed ULink. which more naturally models the fundamental structure of the MUIL problem. Figure 4 shows the result of user identity linkage for multi-platform case, i.e., the three platforms of Renren, 36.cn and Zhaopin.

First of all, since existing solutions consider a pair of platforms at a time, one needs to derive the final user linkage result for the three platforms by integrating the results of



(a) Weibo & Renren (b) Renren & 36.cn (c) Renren & Zhaopin (d) 36.cn & Zhaopin

Figure 5: Result of online framework in the different data sets.

two pairwise linkage, i.e., $A \to B$ and $B \to C$. As shown in Figure 4 (a), for different orders of integrating the pairwise linkage, all the final results of each competing algorithm exhibits noticeable inconsistency. This clearly illustrates the limitation of trying to handle the multi-platform case with pairwise linkage approach, an worrying issue particularly important when no theoretical analysis is known as yet on the stability of the final linkage results thus obtained. Notice that the problem only gets exacerbated as the number of platforms involved increases.

Furthermore, in Figure 4 (b), we take the best results among the different ordering for each method to compare with ULink-CCP. In fact, different connection orders has already been considered in our ULink framework, so that ULink-CCP still outperforms all the rest demonstrates that our model not only provides a stable linkage result unavailable from previous methods, but also offers a better one by a model of greater generality. In particular, the hypothesis of structure consistency is hard to be all satisfied for multi-platform case, the performance of HYDRA is therefore similar as SVM. COSNET is not compared due to the unavailability of necessary information for building matching graph.

6.2.2 *Online Data Setting*

In this part we show how our proposed ULink-On model is able to benefit from new linkage information and improve performance in the online data setting. We notice that this is the first time an online model is proposed for the user identity linkage problem, we therefore choose the state-of-the-art online learning algorithm Passive-Aggressive (PA)[3] for comparison [3].

Figure 5 shows that ULink-On is able to take advantage of new input of user pairs from incoming data stream to update and improve model — the hit-precision of ULink-On increases continuously with the increasing number of linkage pairs. In contrast, the performance of PA-I does not exhibit similar improvement. We assume each incoming piece of data contains one linked pair and one unlinked pair, and verify algorithms on fixed test data set.

In a nutshell, two important characteristics of our proposed ULink-On make it particularly useful for the online setting of the MUIL problem where new data input are constantly generated on various social platforms: (I) It has the ability to update model with improved performance with incremental new data input, e.g., one linked pair and one unlinked pair; and (II) It does not need to store a large amount of data for model construction.

[3]The code used is from Online Multiclass Prediction toolbox at http://www.cs.huji.ac.il/ shais/code/

7. DISCUSSION

We discuss two further challenges of the MUIL problem, together with our solution in plan as future work.

(1) One challenge for any learning algorithm to solve the MUIL problem is how to efficiently handle the exponentially large number of known non-matching user pairs. This issue can be addressed in our framework by applying the cutting plane method[8, 23] to the optimization problem — The constraints most violated are iteratively added to the set of cutting planes for model training until convergence. Alternatively, the latest ensemble method *EasyEnsemble*[13] can be used to build *Ensemble Latent User Space* model, which will not ignore useful information by under-sampling, and obtain the final result by majority voting.

(2) The curse of dimensionality has remained a challenging issue hard to be dealt away in the MUIL problem. In our framework, LUS is built through projection matrix with dimensions adjustable according to measures such as the separability of users. In particular, we can use user similarity as a measure to guide the setup of dimensions for a given platform before model training: the higher the user similarity, the larger the value of dimension d.

8. CONCLUSION

This paper introduces the concept of *Latent User Space* to address in a unified ULink framework two important issues not yet sufficiently explored for the MUIL problem, i.e., platform multiplicity and online data generation. The proposed batch framework ULink based on LUS could be easily shifted into online framework ULink-On. Experiments on real-world data sets have demonstrated the effectiveness of both the proposed batch mode algorithm and the online version, with user linkage results outperforming the state-of-the-art existing methods for both pairwise-platform and multi-platform settings.

Our future work would further advance the efficiency and scalability of our proposed framework with improved performance, and explore theoretical foundation for the latent user space model. It is also in our interest to extend the idea of *Latent User Space* to unsupervised learning framework.

Acknowledgment

This research was partially supported by NSFC (61333014, 61321491), 973 Program (2014CB340501), the Collaborative Innovation Center of Novel Software Technology and

Industrialization, Nanjing University; the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative and Pinnacle lab for analytics at Singapore Management University.

9. REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *WSDM*, pages 635–644, 2011.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Trans. Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[3] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006.

[4] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[5] S. J. Grannis, J. M. Overhage, and C. J. McDonald. Analysis of identifier performance using a deterministic linkage algorithm. In *AMIA*, page 305, 2002.

[6] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, Mar. 2014.

[7] T. Iofciu, P. Fankhauser, F. Abel, and K. Bischoff. Identifying users across social tagging systems. In *ICWSM*, 2011.

[8] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 77(1):27–59, 2009.

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

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

[11] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan. Hydra: Large-scale social identity linkage via heterogeneous behavior modeling. In *SIGMOD*, 2014.

[12] W. Liu and I. W. Tsang. Large margin metric learning for multi-label prediction. In *AAAI*, 2015.

[13] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, 39(2):539–550, 2009.

[14] R. McDonald, K. Crammer, and F. Pereira. Online large-margin training of dependency parsers. In *ACL*, pages 91–98, 2005.

[15] H. Newcombe, J. Kennedy, S. Axford, and A. James. Automatic linkage of vital records. *Science*, 130(3381):954–959, 1959.

[16] H. B. Newcombe. *Handbook of record linkage: methods for health and statistical studies, administration, and business*. Oxford University Press, Inc., 1988.

[17] L. Roos and A. Wajda. Record linkage strategies. Part I: Estimating information and evaluating approaches. *Methods of information in medicine*, 30(2):117–123, 1991.

[18] M. Sadinle and S. E. Fienberg. A generalized fellegi–sunter framework for multiple record linkage with application to homicide record systems. *Journal of the American Statistical Association*, 108(502):385–397, 2013.

[19] P. Sarkar and A. W. Moore. Dynamic social network analysis using latent space models. *ACM SIGKDD Explorations Newsletter*, 7(2):31–40, 2005.

[20] A. J. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of International Workshop on Artificial Intelligence and Statistics*, pages 325–332, 2005.

[21] B. Taskar, D. Klein, M. Collins, D. Koller, and C. D. Manning. Max-margin parsing. In *EMNLP*, 2004.

[22] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(15):615–640, 2010.

[23] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, pages 1453–1484, 2005.

[24] J. Vosecky, D. Hong, and V. Y. Shen. User identification across multiple social networks. In *NDT*, pages 360–365, 2009.

[25] C. Xu, D. Tao, and C. Xu. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

[26] G.-X. Yuan, C.-H. Ho, and C.-J. Lin. An improved glmnet for l1-regularized logistic regression. *Journal of Machine Learning Research*, 13(1):1999–2030, 2012.

[27] R. Zafarani and H. Liu. Connecting corresponding identities across communities. In *ICWSM*, 2009.

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

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

[30] J. Zhang and P. S. Yu. Integrated anchor and social link predictions across partially aligned social networks. In *IJCAI*, 2015.

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

[32] Y. Zhang and J. G. Schneider. Maximum margin output coding. In *ICML*, 2012.

[33] Y. Zhang, J. Tang, Z. Yang, J. Pei, and P. S. Yu. Cosnet: Connecting heterogeneous social networks with local and global consistency. In *KDD*, 2015.