11-2019

AD-Link: An adaptive approach for user identity linkage

Xin MU
Singapore Management University, xinmu@smu.edu.sg

Wei XIE
Singapore Management University, weixie@smu.edu.sg

Ka Wei, Roy LEE
Singapore Management University, roylee@smu.edu.sg

Feida ZHU
Singapore Management University, fdzhu@smu.edu.sg

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

AD-Link: An Adaptive Approach for User Identity Linkage

Xin Mu*, Wei Xie†, Roy Ka-Wei Lee‡, Feida Zhu† and Ee-Peng Lim†
*National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
mux@lamda.nju.edu.cn
†School of Information Systems, Singapore Management University, Singapore
{wei.xie.2012, fdzhu, eplim}@smu.edu.sg
‡Department of Computer Science, University of Saskatchewan, Saskatchewan, Canada
roylee@cs.usask.com

***Abstract*—User identity linkage (UIL) refers to linking accounts of the same user across different online social platforms. The state-of-the-art UIL methods usually perform account matching using user account's features derived from the profile attributes, content and relationships. They are however static and do not adapt well to fast-changing online social data due to: (a) new content and activities generated by users; as well as (b) new platform functions introduced to users. In particular, the importance of features used in UIL methods may change over time and new important user features may be introduced. In this paper, we proposed AD-Link, a new UIL method which (i) learns and assigns weights to the user features used for user identity linkage and (ii) handles new user features introduced by new user-generated data. We evaluated AD-Link on real-world datasets from three popular online social platforms, namely, Twitter, Facebook and Foursquare. The results show that AD-Link outperforms the state-of-the-art UIL methods.**

***Keywords*-user identity linkage; user data growing; user attribute weight;**

I. Introduction

A 2014 survey[1] conducted by *Pew Research Center* has shown that more than half of the Internet users use two or more online social network platforms (OSNs). The use of multiple platforms calls for more comprehensive, and hopefully more accurate, user profiling based on users' content or behavioral data across multiple OSNs [1]–[3]. Such derived user profiles can enhance the effectiveness of recommender systems and help them cope with the cold start problem for new users of an OSN platform who have been active on another OSN platform [4], [5]. To tap on all these opportunities, we need to first tackle the user identity linkage (UIL) problem, which aims to link accounts of the same user across OSNs.

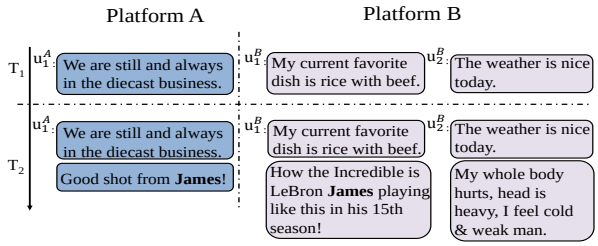
The UIL problem has been widely studied [6]–[8]. However, most solutions assume static data sets and do not adapt to the growth of data. This growth of user-generated data in OSNs can be attributed to (a) users' generation of new content and activity data, and (b) newly added OSN functions enabling users to generate new data. Such data growth brings challenges to the UIL problem.

[1]http://www.pewinternet.org/2015/01/09/social-media-update-2014/

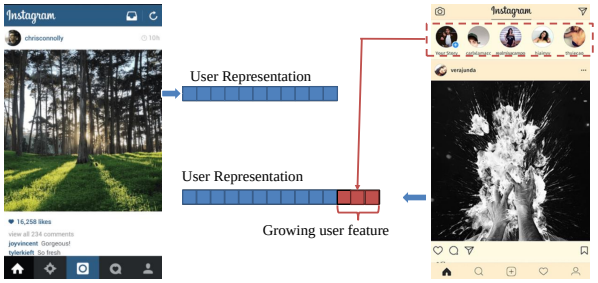
Changes in feature importance. Firstly, most of the existing UIL solutions assign fixed weights to user features (e.g., generated content) to optimize the linkage accuracy [6], [7], [9]. This approach does not adapt to changes in the relative importance of user features as user-generated data grows. Fig. 1(a) illustrates such an example where we attempt to link a user account u_1^A from platform A with some user account from platform B, and u_1^A and u_1^B belong to the same underlying user. At time T_1, we are unable to identify between u_1^B and u_2^B the one that matches with user u_1^A, based on the content posted by the users. Yet at a subsequent time T_2, the similarity between new content by u_1^A and u_1^B suggests a higher chance of linkage between u_1^A and u_1^B.

Expansion in user features. Secondly, OSN platforms constantly introduce new platform functions to attract new users and retain the current ones. In many cases, the new platform functions solicit new user behaviors and data, contributing to an expansion of user features. For instance, in Fig. 1(b), *Instagram* introduced an interactive function called '*stories function*' in 2016, allowing users to post images and photos which expire and disappear after a certain time duration. This new platform function expands the user feature dimensions (e.g., the story images posted by the user). Ideally, the UIL solution should consider this new piece of data and expand the user features for user identity linkage. To the best of our knowledge, there are no existing UIL methods designed to address this challenge yet.

In this paper, we propose a novel framework, AD-Link, to address the above two challenges. AD-Link (i) learns the weights of heterogeneous user features used in linking user identities across multiple OSN platforms, and (ii) addresses the change of user feature data and expansion of user features by incremental learning of model due to added data and the added user features respectively. Our proposed model builds on linking users in *latent user space* (LUS), a concept which was shown to work well in UIL problem [8]. However, unlike the previous works, AD-Link models the LUS through optimizing a linear combination of multiple user attributes with known matching and non-matching pairs of user identities. To address the expansion of user features, AD-Link introduces a new loss function,



(a) Content growing.



(b) User feature expansion.

Figure 1. An illustration of user data growing.

which considers both the existing optimal model and new feature data. Concave-Convex procedure (CCCP) [10] and the accelerated proximal gradient (APG) [11] are applied to solve the optimization problem. Finally, we conduct extensive experiments on real-world datasets and demonstrate the effectiveness and efficiency of our proposed framework.

We summarize our key contributions as follows:

• The proposed AD-Link model learns from a linear combination of user attributes of matching and non-matching user identity pairs so as to learn the projections and the weights of heterogeneous user attributes. Through the assignment of the attribute weights, the proposed method can better model the importance of user attributes.

• AD-Link is able to handle the expansion of user features by introducing a new loss function that balances the new feature data and the existing optimal model. We also employ efficient optimization for this new objective function.

• We benchmark AD-Link against state-of-the-art UIL methods by conducting extensive set of experiments on real-world datasets from three OSN platforms. The results show the effectiveness of the proposed method.

II. Related Work

The problem of *User Identity Linkage* has attracted many research works in recent years [12]. Most of the existing methods address this problem through two main steps: Representation and Modeling. The first step is to represent user identity as a vector of user attribute (user name, profile, timeline, and network, etc.). For example, a user account can be represented by her text content in the form a user feature vector and be compared by pair-wise similarity vector by TF-IDF model [13], word embedding technique [14] or similarity function [15], etc. For instance, Liu et al. [16] proposed a method to perform UIL using *user-name*. Liu et al. [6] proposed a latent topic model which modeled users' tweets for UIL. The trajectory can be extracted from time-stamped location data and modeled to capture the users' activities in [17]. The basic idea of using network information is that if user u_1 is matched onto u_2, then we also hope that u_1's neighbors can also be matched to u_2's neighbors. Zhang et al. [7] used *matching graph* to model the network matching energy function and consider global consistency in the learning model.

Once the representation step is completed, the second step is to employ a learning method to solve the UIL problem. We would like to conclude existing methods into three categories, i.e., supervised, semi-supervised, and unsupervised models. Researchers have given much attention to supervised and semi-supervised learning frameworks. In the supervised approach, UIL can be simplified as a typical binary classification problem. Labeled matching pairs and non-matching pairs are regarded as positive instances and negative instances, respectively, a binary classifier can be trained, such as SVM or logistic regression, for linkage prediction. In [9], Zafarani et al. employed a Bayesian approach for user identification. The recent work by Mu et al. [8] introduced a supervised linkage model through learning a map function from the observed data on the various social platforms to a common space. Semi-supervised methods usually take into account both labeled and unlabeled user identities, and unknown user identity pairs are predicted during the learning process [7], [18]. Furthermore, a multi-objective learning framework was proposed to link user accounts of the same person across different social networks [6]. Unsupervised model is performed with only unlabeled data. CNL [15] is an unsupervised and collective method to link user accounts across heterogeneous social networks, which incorporates heterogeneous attributes and social features unique to social network users.

Creating models to cope with environment changes [19], is widely studied in the machine learning and data mining community. For example, incremental feature [20], [21] is one of the feature evolution scenario. Other relevant problems and approaches include *link prediction problem* [22] which is to predict missing or future formed links in different social networks (homogeneous or heterogeneous social networks); *subspace learning-based* approaches [23], e.g., 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; and *learning distance metric* [24] which is a framework for learning a distance metric in feature space.

III. Preliminaries

We define the User identity linkage (UIL) problem as follows: Given two social networks $\mathcal{G}_A$ and $\mathcal{G}_B$, and user

identities u_i^A and u_j^B from $\mathcal{G}_A$ and $\mathcal{G}_B$ respectively, find a function f to predict whether u_i^A and u_j^B belongs to same real person such that: $f(u_i^A, u_j^B) = 1$, when u_i^A and u_j^B belong to the same person, and 0 otherwise. When $f(u_i^A, u_j^B) = 1$, we call (u_i^A, u_j^B) a matching user identity pair. Conversely, when $f(u_i^A, u_j^B) = 0$, we call (u_i^A, u_j^B) a non-matching user identity pair.

Generally, user identity includes multiple user attributes such as profile, network, timeline content, etc. User feature vector is one of existing user representation techniques, which maps original information into an n-dimensional numerical vector. In this paper, we consider there exist R user attributes. The user identity u^A is denoted as $u^A = \{\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^R\}$, where $\boldsymbol{x}^r \in \mathbb{R}^{d_r^A}$ is the r^{th} user attribute representation and d_r^A is the dimension of r^{th} user attribute. Similarly, let $u^B = \{\boldsymbol{y}^1, \boldsymbol{y}^2, \cdots, \boldsymbol{y}^R\}$, $\boldsymbol{y}^r \in \mathbb{R}^{d_r^B}$ be a user identity in the platform $\mathcal{G}_B$. We define a user identity triplet as $\{u_i^A, u_i^B, \tilde{u}_i^B\} = \{\boldsymbol{x}_i^r, \boldsymbol{y}_i^r, \tilde{\boldsymbol{y}}_i^r\}_{r=1}^R$, where u_i^A and u_i^B are same person from two social platforms $\mathcal{G}_A$ and $\mathcal{G}_B$, u_i^B and $\tilde{u}_i^B$ are different persons from social platform $\mathcal{G}_B$.

In a recent study, Mu et al. [8] introduced Latent User Space (LUS) to address the UIL problem. The basic idea is that original data is projected from different platforms to a common LUS, where the data points of the matching pair are close to each other in the LUS, while the different user identities are "pushed" further apart. Solving the UIL problem thus turns into an optimization problem by finding a set of projection functions Φ as follows:

$$\min_{\Phi} \quad \mathbb{D}(\Phi^A(u^A), \Phi^B(u^B)), \tag{1}$$

where $\mathbb{D}(u_i, u_j)$ is the distance between any two user identities.

Inspired by this idea, we propose a novel adaptive user linkage framework named AD-Link, which builds on LUS by (a) assigning the weight to different user attributes, and (b) handling user feature expansion by considering both the model before and after addition of new user features. The details will be given as follows.

IV. Proposed Method

A. AD-Link for weighting user attributes

In this subsection, we introduce the proposed method AD-Link for weighting user attributes. AD-Link includes a weight α_r for each user attribute r and a distance difference objective function $D(\cdot)$. We further employ the Concave-Convex Procedure (CCCP) optimization technique to learn the user attribute projection Φ_r and the weight α_r for each attribute r. Building on the idea of LUS, we define a distance function follows:

$$\begin{aligned} D(u_i^A, u_i^B, \tilde{u}_i^B) = \sum_{r=1}^{R} \alpha_r (\mathbb{D}(\Phi_r^A(\boldsymbol{x}_i^r), \Phi_r^B(\boldsymbol{y}_i^r)) - \\ \mathbb{D}(\Phi_r^A(\boldsymbol{x}_i^r), \Phi_r^B(\tilde{\boldsymbol{y}}_i^r))), \quad \text{with } \sum_{r=1}^{R} \alpha_r = 1, \alpha_r > 0. \end{aligned} \tag{2}$$

In (2), we use a user identity triple to define distance difference as a sum of attribute weighted difference between "projected" known matching pair distance and "projected" known non-matching pair distance. The sum of the attribute weights is equal to one. Specifically, in this work, we take Euclidean distance as the distance function, i.e., $\mathbb{D}(\boldsymbol{x}_i, \boldsymbol{y}_i) = ||\boldsymbol{x}_i - \boldsymbol{y}_i||_2^2$. The projection function is defined as $\Phi_r(\boldsymbol{x}^r) = \boldsymbol{x}^r w^r$, where $w^r \in \mathbb{R}^{d^r \times d^{LUS}}$ is a projection matrix for r^{th} attribute, and d^{LUS} is the dimension of LUS. Suppose we now have N triplets, the proposed framework AD-Link is to minimize the following objective function:

$$\begin{aligned} \min_{\Phi, \alpha} \quad & \sum_{i=1}^{N} D(u_i^A, u_i^B, \tilde{u}_i^B) + C\Omega(\Phi) \\ s.t. \quad & \sum_{r=1}^{R} \alpha_r = 1, \alpha_r > 0, \end{aligned} \tag{3}$$

where the projection matrix Φ is penalized by the norm $\Omega(\cdot)$, C is the coefficient.

As shown in (3), the matching users will be close to each other in the LUS, while the non-matching users will be pushed further apart. Also, for each user attribute, the smaller the distance difference between the matching and non-matching pairs, the larger α_r should be (i.e., the more important the attribute is). In addition, the constraint on the weight α_r is similar to ℓ_1 norm, which will force some α_r to be zero if there are a large number of user attributes.

For ease of exposition, Eqn. (3) can be rewritten as

$$\begin{aligned} \min_{\{w_A^r, w_B^r, \alpha^r\}_{r=1}^R} & \sum_{i=1}^{N} \sum_{r=1}^{R} \alpha_r (||\boldsymbol{x}_i^r w_A^r - \boldsymbol{y}_i^r w_B^r||_2^2 - ||\boldsymbol{x}_i^r w_A^r - \\ & \tilde{\boldsymbol{y}}_i^r w_B^r||_2^2) + C(\sum_{r=1}^{R} ||w_A^r||_{\mathrm{F}}^2 + \sum_{r=1}^{R} ||w_B^r||_{\mathrm{F}}^2) \\ s.t. \quad & \sum_{r=1}^{R} \alpha_r = 1, \alpha_r > 0, \end{aligned} \tag{4}$$

where $||\cdot||_{\mathrm{F}}$ is the Frobenius norm.

Optimization. Inspired by this optimization technique [25], [26], we reformulate (4) as the following:

$$\min_{\alpha^r} \quad P(\alpha^r) \quad \text{ such that } \quad \sum_{r=1}^{R} \alpha_r = 1, \alpha_r > 0. \tag{5}$$

where,

$$\begin{aligned} P(\alpha^r) = \min_{w_A^r, w_B^r} \sum_{i=1}^{N} \sum_{r=1}^{R} \alpha_r (||\boldsymbol{x}_i^r w_A^r - \boldsymbol{y}_i^r w_B^r||_2^2 - \\ ||\boldsymbol{x}_i^r w_A^r - \tilde{\boldsymbol{y}}_i^r w_B^r||_2^2) + \sum_{r=1}^{R} ||w_A^r||_{\mathrm{F}}^2 + \sum_{r=1}^{R} ||w_B^r||_{\mathrm{F}}^2. \end{aligned} \tag{6}$$

We adopt a simple yet effective strategy that combines the alternative optimization technique into the "Concave-Convex Procedure" (CCCP) framework [10], [27], i.e., we consider one projection w_A^r as the variable and the others as fixed values. The CCCP can solve a difference of convex functions programming as a sequence of convex programming. Each iteration of the CCCP procedure approximates the concave part by its tangent and minimizes the resulting convex

function: $\theta^{t+1} = \arg\min_\theta(J_{vex}(\theta) + J'_{cav}(\theta^t)\cdot\theta)$, where $J'_{cav}(\theta^t)$ is a derivation of $J_{cav}(\theta)$. As a result, problem (6) can be reformulated as:

$$\begin{aligned}\mathcal{L}(w_A^r) = &\min_{w_A^r} ||w_A^r||_{\mathrm{F}}^2 + \sum_{i=1}^{N}\alpha_r(||x_i^r w_A^r - y_i^r w_B^r||_2^2 \\ &- 2*\alpha_r x_i^r w_A^r(x_i^r w_A^r(t) - \tilde{y}_i^r w_B^r)).\end{aligned} \tag{7}$$

After the decomposition of the objective function, Eqn.(7) is a convex optimization problem, which can be solved by a quadratically constrained quadratic program (QCQP) can be used to solve it.

So far, we can get each optimal projection matrix w through the above process. In the second step, the optimization problem (5) becomes a linear combination of convex problem when (7) is considered, which can be solved by a optimization software in each iteration such as CVX [28]. The procedure stops when the terminating condition is satisfied, i.e., a predetermined maximum number of iterations.

B. AD-Link for user feature expansion

Before introducing the details in this subsection, we define the user features expansion as follows:

Definition 4.1: **[User Feature Expansion (UFE)]** Let $\boldsymbol{x}_t \in \mathbb{R}^{d_t}$ be a user feature vector in the time period t and $\boldsymbol{x}_{t+1} \in \mathbb{R}^{d_{t+1}}$ be a user feature vector at time period $t+1$. The expansion occurs at time period $t+1$ if $d_{t+1} > d_t$.

In the time period t, which we call *initial stage*, a $model_t$ is built on the current data from Twitter and Facebook platforms (represented in blue and red respectively). In the time period $t+1$, which we call *expansion stage*, user feature vector $x^{Twitter}$ might grow due to new interaction function. $model_t$ is unable to utilize these new features to improve user identity linkage. Therefore, we need a timely updated model which utilizes the new features and retain the good matching results from the existing $model_t$. Note that any existing UIL models can be applied in the *initial stage* (I-Stage), as it is a traditional UIL problem. In this section, we will focus on addressing the *expansion stage* (E-Stage).

A straightforward approach to this problem is to learn a new model based on data with expansion features. However, this solution suffers from some deficiencies. First, when new features are created, there might be few data samples described by these features, and thus, the samples are insufficient for learning a strong model. Second, the previous model already learned is not incorporated to take advantage of the data collection effort. Our approach in this paper, therefore, builds a new model based on the existing model together with the limited data described by new features.

As only one specific attribute will be discussed in the following, we omit symbol 'r' for clarity. Let $\{\boldsymbol{x}_{i,t},\ \boldsymbol{y}_{i,t},\ \tilde{\boldsymbol{y}}_{i,t}\}$ be a triplet in the E-Stage, where $\boldsymbol{x}_{i,t} \in \mathbb{R}^{d_t^A}$, $\boldsymbol{y}_{i,t}$ and $\tilde{\boldsymbol{y}}_{i,t} \in \mathbb{R}^{d_t^B}$. Let $\{\boldsymbol{x}_{i,t-1}, \boldsymbol{y}_{i,t-1}, \tilde{\boldsymbol{y}}_{i,t-1}\}$ be a triplet in the I-Stage, where $\boldsymbol{x}_{i,t-1} \in \mathbb{R}^{d_{t-1}^A}$, $\boldsymbol{y}_{i,t-1}$ and $\tilde{\boldsymbol{y}}_{i,t-1} \in \mathbb{R}^{d_{t-1}^B}$, $d_{t-1}^A \le d_t^A$, $d_{t-1}^B \le d_t^B$. We consider the function projection in the E-Stage as $\Phi_t(\boldsymbol{x}_{i,t}) = \boldsymbol{x}_{i,t}\cdot w_t$, where $w_t \in \mathbb{R}^{d_t \times d^{LUS}}$ is a projection matrix.

To address the user feature expansion in the E-Stage, we introduce a new loss function ℓ_a such that the distance between matching users in I-Stage, are similar in the new projected LUS. We define ℓ_a as: $\ell_a(\Phi_{t-1}^A, \Phi_{t-1}^B, \Phi_t^A, \Phi_t^B) = \mathbb{D}(\Phi_t^A(\boldsymbol{x}_{i,t}), \Phi_t^B(\boldsymbol{y}_{i,t})) - \mathbb{D}(\Phi_{t-1}^A(\boldsymbol{x}_{i,t-1}), \Phi_{t-1}^B(\boldsymbol{y}_{i,t-1}))$. Note that when we minimize this difference between new projections Φ_t and existing projections Φ_{t-1}, it allows the new projections Φ_t still keeping the same user identities close to each other.

We therefore search for new projection functions in the E-Stage to solve the following minimization problem:

$$\min_{\Phi_t, \Phi_{t-1}} \quad \lambda_1\ell_h + \lambda_2\ell_a + C(\Omega(\Phi_t) + \Omega(\Phi_{t-1})), \tag{8}$$

where the loss functions $\ell_h(\Phi_t^A, \Phi_t^B) = \mathbb{D}(\Phi_t^A(\boldsymbol{x}_{i,t}), \Phi_t^B(\boldsymbol{y}_{i,t})) - \mathbb{D}(\Phi_t^A(\boldsymbol{x}_{i,t}), \Phi_t^B(\tilde{\boldsymbol{y}}_{i,t}))$, $\Omega(\cdot)$ is regularization term, λ_1, λ_2 and C are the coefficients. Note that ℓ_h is of the same structure as in (2), which encourages the matching users close to each other in the LUS, while the non-matching users will be pushed further apart. We also provide the discussion of parameters λ_1 and λ_2 in Section V-C. For ease of exposition, Eqn.(8) can be rewritten as:

$$\begin{aligned}\min_{w_t, w_{t-1}} \lambda_1 \sum_{i=1}^{N}(&||\boldsymbol{x}_{i,t}w_{1,t} - \boldsymbol{y}_{i,t}w_{2,t}||_2^2 - ||\boldsymbol{x}_{i,t}w_{1,t} - \\ &\tilde{\boldsymbol{y}}_{i,t}w_{2,t}||_2^2) + \lambda_2 \sum_{i=1}^{N}(||\boldsymbol{x}_{i,t}w_{1,t} - \boldsymbol{y}_{i,t}w_{2,t}||_2^2 - \\ &||\boldsymbol{x}_{i,t-1}w_{1,t-1} - \boldsymbol{y}_{i,t-1}w_{2,t-1}||_2^2) + \\ &C\sum_{j=1}^{2}(||w_{j,t}||_{\mathrm{F}}^2 + ||w_{j,t-1}||_{\mathrm{F}}^2).\end{aligned} \tag{9}$$

Optimization. To deal with this optimization problem, we employ an efficient optimal algorithm called Accelerated Proximal Gradient (APG) [8], [11], [29] together with the alternative optimization technique. The details will be given as follows.

For convenience, we combine variables $w_{1,t}$, $w_{2,t}$ to $W_t = \begin{bmatrix} w_{1,t} \\ w_{2,t} \end{bmatrix}$, $W_t \in \mathbb{R}^{(d_t^A + d_t^B)\times d^{LUS}}$. We define a symmetric positive semidefinite matrix Q_t : $Q_t = W_t(W_t)^T, Q_t \in \mathbb{R}^{(d_t^A+d_t^B)\times(d_t^A+d_t^B)}$. The matching pair vector is denoted by $p_{i,t} = \begin{bmatrix} x_{i,t} & -y_{i,t} \end{bmatrix}$, non-matching pair vector $\tilde{p}_{i,t} = \begin{bmatrix} x_{i,t} & -\tilde{y}_{i,t} \end{bmatrix}$, $p_{i,t},\ \tilde{p}_{i,t} \in \mathbb{R}^{(d_t^A+d_t^B)}$. Similarly, we define $p_{i,t-1}$, $\tilde{p}_{i,t-1}$ and Q_{t-1} are for user data in the I-Stage. Thus, Eqn.(9) can be transformed to the following problem:

$$\begin{aligned}\min_{Q_t, Q} \sum_{i=1}^{N}(&\lambda_1(\mathrm{Tr}(Q_t A_{i,t}) - \mathrm{Tr}(Q_t \tilde{A}_{i,t})) + \lambda_2(\mathrm{Tr}(Q_t A_{i,t}) - \\ &\mathrm{Tr}(Q_{t-1}A_{i,t-1}))) + C(\mathrm{Tr}(Q_t) + \mathrm{Tr}(Q_{t-1})),\end{aligned} \tag{10}$$

where $\mathrm{Tr}(\cdot)$ is the trace, $A_{i,t} = (p_{i,t})^T p_{i,t}$, $\tilde{A}_{i,t} = (\tilde{p}_{i,t})^T \tilde{p}_{i,t}$, $A_{i,t-1} = (p_{i,t-1})^T p_{i,t-1}$.

Note that any feasible solution to (10) gives a feasible (or optimal) solution to (9), and vice versa [30]. We can apply the Accelerated Proximal Gradient (APG) method to solve the primal form of (10) to alternatively obtain the optimal Q_t and Q_{t-1}, i.e., we first use a fixed value Q_{t-1} to calculate Q_t in (10), then fix Q_t to get the new Q_{t-1}.

In the following, we briefly describe the optimization procedure to derive Q_t, and the similar process can be conducted to get Q_{t-1}. The detailed sketch of APG can be found in [11].

Let $F(Q_t, Q_{t-1}) = f(Q_t, Q_{t-1}) + r(Q_t, Q_{t-1})$, where $r(Q_t, Q_{t-1}) = C(\mathrm{Tr}(Q_t) + \mathrm{Tr}(Q_{t-1}))$, $f(Q_t, Q_{t-1}) = \sum_{i=1}^{N}(\lambda_1(\mathrm{Tr}(Q_t A_{i,t}) - \mathrm{Tr}(Q_t \tilde{A}_{i,t})) + \lambda_2(\mathrm{Tr}(Q_t A_{i,t}) - \mathrm{Tr}(Q_{t-1} A_{i,t-1})))$, Set the partial derivatives of f with respect to the elements of Q_t and Q_{t-1}, $\nabla f^1(Q_t) = (\lambda_1 + \lambda_2)A_{i,t} - \lambda_1(\tilde{A}_{i,t})$, $\nabla f^2(Q_{t-1}) = -\lambda_2(A_{i,t-1})$.

Given Q_{t-1}, for any symmetric positive semidefinite matrix Z, we define the following QP problem of $F(Q_t, Q_{t-1})$ at Z:

$$\begin{aligned} A\tau(Q_t; Z) =& f(Z, Q_{t-1}) + < \nabla f^1(Z); Q_t - Z > \\ &+ \frac{\tau}{2}||Q_t - Z||_{\mathrm{F}}^2 + r(Q_t, Q_{t-1}) \\ =& \frac{\tau}{2}||Q_t - G||_{\mathrm{F}}^2 + r(Q_t, Q_{t-1}) \\ &+ f(Z, Q_{t-1}) + \frac{1}{2\tau}||\nabla f^1(Z)||_{\mathrm{F}}^2, \end{aligned} \tag{11}$$

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

$$\arg\min_{Q_t} \quad \frac{\tau}{2}||Q_t - G||_{\mathrm{F}}^2 + r(Q_t, Q_{t-1}). \tag{12}$$

We thus take the derivative of the objective function, and get $Q_t = G - \frac{C}{\tau}I$. Note that G can be decomposed by SVD as $G = U\overline{G}V^T$, $Q_t = U\overline{G}V^T - \frac{C}{\tau}UU^T$, and $Q_t = U(\overline{G} - \frac{C}{\tau}I)U^T$. Zero is used to replace the negative entries in $\overline{G} - \frac{C}{\tau}I$. When we get optimal Q_t, the same procedure is conducted to get optimal Q_{t-1}. Finally, the projection matrix W_t can be obtained by matrix Q_t, and this procedure can be conducted again for other user attributes.

So far, the new linkage model is ready for linking the same accounts in this stage. The same procedure further can be applied when the new feature is emerging in the next E-Stage. Note that though the increased dimensions of user feature are not consistent, i.e., $d_{t+1}^A \neq d_{t+1}^B$, the data will be finally projected to the same dimension, because LUS is built through projection matrix with adjustable dimension.

V. Experiment

In this section, we conduct experiments to evaluate AD-Link and several other state-of-the-art methods. The experiments consist of two parts: (1) the effectiveness in different time periods in Section V-B; and (2) the ability to handle user feature expansion in Section V-C.

A. Experimental Setup

Data Sets. We evaluate our proposed model using data sets from three popular OSNs, namely, *Twitter*, *Facebook* and *Foursquare*. We first gathered a set of Singapore-based Twitter users who declared Singapore as a location in their user profiles. From Singapore-based Twitter users, we retrieve a subset of Twitter users who declared their Facebook or Foursquare accounts in their short bio description. In total, we gathered 3,739 Facebook-Twitter (FB-TW) matching pairs and 5,982 Foursquare-Twitter (FQ-TW) matching pairs. User attributes include user name, screen name, tweets, bio information and network. Note that for network attribute, we retrieve the links among users in the (FB-TW) and (FQ-TW), and the network information is preprocessed using the "deepwalk"[2]. The text content is preprocessed using the "word2vec"[3], and the name information is preprocessed using the name-embedding approach in [31].

Methods for Comparison. We compare AD-Link with the below state-of-the-art user linkage methods: 1. **HYDRA** [6]: A linkage function learns by multi-objective optimization incorporating both supervised learning on user identity linkage information and the cross-platform structure consistency maximization. 2. **COSNET** [7]: Links user identities by considering distance-based profile features and network features, i.e., both local and global consistency. 3. **ULink** [8]: A supervised method which learns projection functions to map the original feature space to a latent user space across the social platforms. The linkage finally is calculated in the latent user space. 4. **SVM** [32]: A binary prediction on user pairs using support vector machines. The training data is composed of matching pairs and non-matching pairs, which is represented by 1 and -1 as the label respectively. 5. **KNN**: Uses K-Nearest-Neighbor (KNN) to generate k nearest user identities on $\mathcal{G}_B$ as matching candidates.

Experiment Settings. All methods except COSNET (C++) are executed in the MATLAB environment with the following implementations: LIBSVM package [32] is used for modeling SVM; the codes for HYDRA are developed based on the original papers; ULink and COSNET use the codes as released by the corresponding authors.

The coefficient C in our algorithm, SVM and ULink is selected via cross-validation on the training data. Parameter B in ULink is set according to the value mentioned in [8]. All parameters in COSNET are set by default. For HYDRA, the parameter p, which determines how the learned model 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 the HYDRA framework from a decision maker's perspective, are set by tuning on the validation set. The dimension of LUS d^{LUS} in ULink and AD-Link is set as 300 according

[2]https://github.com/phanein/deepwalk

[3]https://radimrehurek.com/gensim/index.html

Table I
RESULTS ON FB-TW DATA SET.

	2014-2015			2014-2017		
Method	*Precision@1*	*Precision@10*	*MRR*	*Precision@1*	*Precision@10*	*MRR*
KNN	0.569±0.011	0.618±0.010	0.601±0.010	0.571±0.013	0.620±0.012	0.600±0.012
SVM	0.591±0.026	0.656±0.024	0.613±0.022	0.603±0.024	0.640±0.025	0.637±0.020
HYDRA	0.652±0.036	0.732±0.030	0.673±0.031	0.652±0.031	0.711±0.035	0.671±0.036
COSNET	**0.696±0.040**	0.754±0.033	0.703±0.032	0.699±0.032	0.758±0.037	0.708±0.031
ULink	0.683±0.021	0.760±0.022	0.709±0.021	0.687±0.022	0.758±0.020	0.710±0.024
AD-Link	0.690±0.014	**0.781±0.012**	**0.717±0.019**	**0.701±0.013**	**0.783±0.019**	**0.719±0.020**

Table II
RESULTS ON FQ-TW DATA SET.

	2016			2016-2017		
Method	*Precision@1*	*Precision@10*	*MRR*	*Precision@1*	*Precision@10*	*MRR*
KNN	0.537±0.010	0.621±0.015	0.575±0.011	0.538±0.011	0.623±0.013	0.580±0.012
SVM	0.539±0.017	0.682±0.013	0.584±0.012	0.532±0.012	0.688±0.017	0.581±0.011
HYDRA	0.546±0.023	0.692±0.020	0.617±0.018	0.543±0.021	0.693±0.023	0.615±0.015
COSNET	0.546±0.013	0.699±0.017	0.624±0.012	0.545±0.011	0.698±0.013	0.626±0.011
ULink	0.542±0.021	0.701±0.020	0.639±0.015	0.547±0.022	0.699±0.022	0.635±0.017
AD-Link	**0.559±0.024**	**0.721±0.025**	**0.645±0.022**	**0.551±0.022**	**0.729±0.026**	**0.649±0.021**

to the guideline in [8]. We take the Euclidean distance as the distance function. 70% of the ground-truth matching pairs are allocated for training, the remaining 30% are as test data set. In the training set, non-matching pairs are randomly sampled by setting ratios, 1:10, between the ground-truth matching pairs to non-matching pairs.

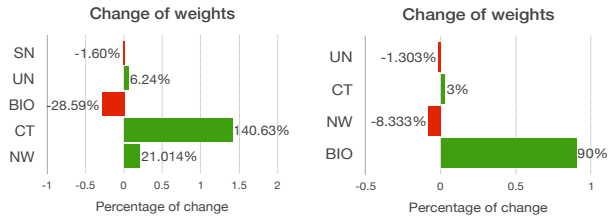
Evaluation Metrics. We employ two evaluation metrics. One is $precision@K = \frac{\sum^{N} hit(x_i)}{N}$, where $hit(x_i) = 1$ if correct linked user is in the returned top K candidates, otherwise $hit(x_i) = 0$. We use $K = 1$ and $K = 10$ in this paper. The other is mean reciprocal rank (*MRR*), which is the mean of reciprocal rank of ground truth target user identity in the user identity rank list returned by the UIL method. $MRR = \frac{1}{N}\sum^{N}\frac{1}{Rank_{x_i}}$, where $Rank_{x_i}$ refers to the rank position of the truth target user identity.

B. Change in User Attributes Importance

Setting. We first crawled 2014∼2017 user data from *Facebook* and *Twitter*, and crawled 2016∼2017 user data from *Foursquare* and *Twitter*. Then, we separate each pairwise data set into two groups according to the year, as shown in Table I and Table II. The experiment is repeated for ten times, and the average result of ten trials is reported.

Results. As can be seen in Tables I and II, AD-Link has consistently produced higher *Precision@10* and *MRR* than other methods. SVM, which considers pair similarity performs worse than AD-Link. COSNET and HYDRA, both of which consider user network graph and profile features, show better performance than directly calculating the distance in the original space. The closest contender ULink, which is also based on LUS without weighting for user attributes, is worse than AD-Link. While KNN needs no training and runs faster, its performance fell behind others in these data sets.

Specifically, the method COSNET and HYDRA learn the linkage function via modeling the text contents similarity and social network behavior. COSNET, which further considers the global consistency of network, could perform



(a) FB-TW. (b) FQ-TW.

Figure 2. The percentage changes in the user attribute weights across the two time periods. (user name(UN); screen name(SN); bio information(BIO); content (CT); network(NW).)

better than HYDRA. However, both of them still performed worse than AD-Link. One possible reason is that whenever network data is sparse, they cannot capture network information to improve the performance or suffer from unreliable similarity information. SVM suffers from a number of challenges including high computational complexity when using RBF kernel, difficulty in finding the right parameters and attributes importance. ULink is similar to our framework, but it achieved worse results than AD-Link. This explains that the idea of weighting attributes is important in the real problem. KNN, which directly calculates the distance in the space defined by the feature vectors of the linking social platforms, has been shown to work poorly.

In addition, it also worth noting that the performance of ULink and HYDRA on FB-TW data set declined slightly in the second time period, and that explains models might be affected by data growing. AD-Link achieved the best performance and remained robust in these data sets, making it, in general, the best choice for this problem.

We also empirically examined the changes in user attribute weights across the two time periods in Fig. 2. Generally, we observed that the user attribute weights increase and decrease in various magnitude. For instance, we observed that the importance of using a content attribute for user linkage increases by 140% in the 2014-2017 time period

in the FB-TW dataset. Conversely, in the same time period and dataset, the importance of using bio information attribute decreases by over 28%. Similar observations are also made for the FQ-TW dataset. AD-Link is more robust and can capture these salient changes in user attribute weights, thus outperforming the state-of-the-art methods in user linkage.

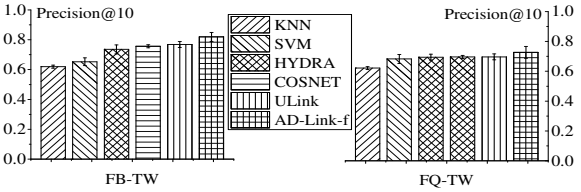
C. User Feature Expansion

We demonstrate in this part, existing solutions suffer from inherent defects under user feature expansion, driving home the importance of a new framework like our proposed AD-Link, which more naturally handles user feature expansion. In the following, we call the proposed method under user feature expansion AD-Link-f.

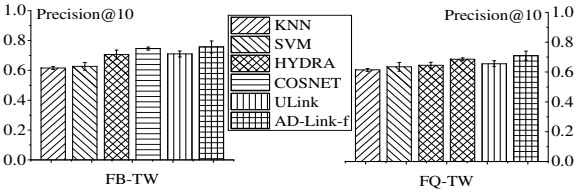
Setting. In the following experiments, two groups as mentioned in the previous subsection are respectively regarded as 'I-Stage' and 'E-Stage', e.g., in FB-TW data set, the time period 2014∼2015 as I-Stage and 2014∼2017 as E-Stage. In the E-Stage, we select one user attributes from bio information, post content, network as the expansion feature, and feature vector will be 100 dimensions more than that in the I-Stage. Furthermore, available the number of ground-truth matching pairs in the E-Stage is decreased as demonstrated in Fig. 3. As all comparison are not designed to handle the feature expansion, they will be retrained in E-Stage. For our proposed method, we use these limited data and existing model in the I-Stage to train the model. We use $Precision@10$ as the evaluation metric. The experiment in each setting is repeated for 10 times, and both the mean and the standard variance of the performance are reported.

Results. The results are shown in Fig. 3. AD-Link-f produced better performance than other methods, especially when the number of matching pairs is limited (e.g., Fig.3(c)). AD-Link-f can take advantage of the existing model and new data to build linkage model in the E-Stage and converges very fast, as shown in Fig. 4. Retraining a new model like HYDRA, ULink, etc., is a straightforward idea to adapt to a new situation, but those solutions suffer from some shortcomings: 1) when new features just emerge, there are few data samples described by these features. They thus produce poor results due to a lack of training data. For example, the performance of all methods in Fig. 3(c) is worse than them in Fig. 3(a); and 2) the abandon of the existing model is a big waste since it takes efforts on modeling. Despite COSNET, which considers both unlabelled and labeled user pairs in the training process, it still performed worse than AD-Link-f. In short, AD-Link-f is a good choice for user feature expansion, especially when only limited data are available.

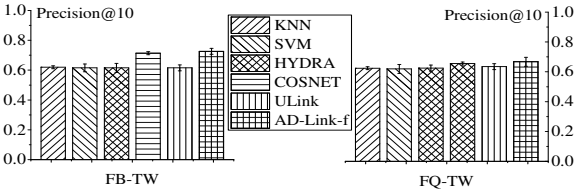
Convergence and parameters analysis. We validate the convergence of AD-Link-f under the 500 matching pairs available scenario. Fig. 4(a) shows that: First, AD-Link-f converges finally; Second, this method converges fast benefited from the optimization APG. We also study the influences of two major parameters, i.e., λ_1 and λ_2 in Fig. 4(b). It can be found that AD-Link-f performs well when ratio is set as around $\lambda_1 = 0.7$ and $\lambda_2 = 0.3$. In particular, this ratio can be used to guide the setup in other data sets.



(a) 3000 matching pairs available.

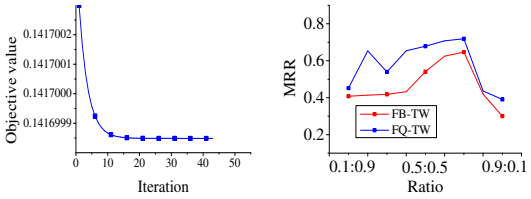


(b) 1500 matching pairs available.



(c) 500 matching pairs available.

Figure 3. Results on user feature expansion.



(a) AD-Link-f Convergence. (b) Sensitivity tests. The x-axis is the ratio between λ_1 and λ_2.

Figure 4. Convergence and parameters analysis.

VI. Conclusion

This paper addresses the problem of growing user data in UIL. We proposed a general framework AD-Link, which learns and assigns different weights to different features, and handles new user features as they emerge from the OSN platforms. We have evaluated AD-Link by conducting experiments on real-world data sets. The experiment results have demonstrated the superiority of AD-Link over the state-of-the-art methods. For future works, we will further advance the efficiency and scalability of our proposed framework. We will also explore adapting AD-Link to the online learning scenario for UIL. It is also our interest to extend the theoretical foundation AD-Link and other proposed models.

ACKNOWLEDGMENT

This research was supported by the National Research Foundation, Prime Minister's Office, Singapore under its International Research Centres in Singapore Funding Initiative.

REFERENCES

[1] J. Tang, L. Yao, D. Zhang, and J. Zhang, "A combination approach to web user profiling," *TKDD*, vol. 5, no. 1, pp. 2:1–2:44, 2010.

[2] R. K.-W. Lee, T.-A. Hoang, and E.-P. Lim, "On analyzing user topic-specific platform preferences across multiple social media sites," in *WWW*, 2017.

[3] ——, "Discovering hidden topical hubs and authorities across multiple online social networks," *IEEE TKDE*, 2019.

[4] Z. Deng, J. Sang, and C. Xu, "Personalized video recommendation based on cross-platform user modeling," in *ICME*, 2013, pp. 1–6.

[5] R. K.-W. Lee and D. Lo, "Wisdom in sum of parts: Multi-platform activity prediction in social collaborative sites," in *WebSci*, 2018.

[6] S. Liu, S. Wang, F. Zhu, J. Zhang, and R. Krishnan, "HYDRA: large-scale social identity linkage via heterogeneous behavior modeling," in *SIGMOD*, 2014, pp. 51–62.

[7] 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, pp. 1485–1494.

[8] X. Mu, F. Zhu, E.-P. Lim, J. Xiao, J. Wang, and Z.-H. Zhou, "User identity linkage by latent user space modelling," in *KDD*, 2016, pp. 1775–1784.

[9] R. Zafarani and H. Liu, "Connecting users across social media sites: a behavioral-modeling approach," in *KDD*, 2013, pp. 41–49.

[10] A. L. Yuille and A. Rangarajan, "The concave-convex procedure," *Neural Computation*, vol. 15, no. 4, pp. 915–936, 2003.

[11] C. Kang, S. Liao, Y. He, J. Wang, W. Niu, S. Xiang, and C. Pan, "Cross-modal similarity learning: A low rank bilinear formulation," in *CIKM*, 2015, pp. 1251–1260.

[12] K. Shu, S. Wang, J. Tang, R. Zafarani, and H. Liu, "User identity linkage across online social networks: A review," *SIGKDD Explorations*, vol. 18, no. 2, pp. 5–17, 2016.

[13] G. Salton and M. McGill, *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, 1984.

[14] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *NIPS*, 2014, pp. 2177–2185.

[15] M. Gao, E.-P. Lim, D. Lo, F. Zhu, P. K. Prasetyo, and A. Zhou, "CNL: collective network linkage across heterogeneous social platforms," in *ICDM*, 2015, pp. 757–762.

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

[17] C. J. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *WWW*, 2016, pp. 707–719.

[18] L. Liu, W. K. Cheung, X. Li, and L. Liao, "Aligning users across social networks using network embedding," in *IJCAI*, 2016, pp. 1774–1780.

[19] Z.-H. Zhou, "Learnware: On the future of machine learning," *Frontiers of Computer Science*, vol. 10, no. 4, pp. 355–384, 2016.

[20] B.-J. Hou, L. Zhang, and Z.-H. Zhou, "Learning with feature evolvable streams," in *NIPS*, 2017, pp. 1416–1426.

[21] C. Hou and Z.-H. Zhou, "One-pass learning with incremental and decremental features," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[22] J. Zhang and P. S. Yu, "Integrated anchor and social link predictions across social networks," in *IJCAI*, 2015, pp. 2125–2132.

[23] C. Xu, D. Tao, and C. Xu, "A survey on multi-view learning," *CoRR*, vol. abs/1304.5634, 2013.

[24] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," *Michigan State Universiy*, vol. 2, no. 2, 2006.

[25] M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.

[26] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet, "Simplemkl," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2491–2521, 2008.

[27] B. K. Sriperumbudur and G. R. G. Lanckriet, "On the convergence of the concave-convex procedure," in *NIPS*, 2009, pp. 1759–1767.

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

[29] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," *Pacific Journal of optimization*, vol. 6, no. 615-640, p. 15, 2010.

[30] Y. Zhang and J. G. Schneider, "Maximum margin output coding," in *ICML*, 2012.

[31] W. Xie, X. Mu, R. K.-W. Lee, F. Zhu, and E.-P. Lim, "Unsupervised user identity linkage via factoid embedding," in *ICDM*, 2018.

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