User Identity Linkage for Different Behavioral Patterns across Domains

Genki Kusano, Masafumi Oyamada
NEC Corporation
g-kusano@nec.com, oyamada@nec.com

Abstract

As customers use and benefit from multiple services, a large amount of customer data are accumulating daily. Connecting a customer's identity on a service with her identity on a different service, known as *user identity linkage* (UIL), enables a comprehensive understanding of users in a variety of real-world applications. The difficulties of UIL tasks in marketing applications are mainly the lack of user demographics and diverse user behavioral patterns, which differs from UIL tasks in social networking services that previous UIL methods have mainly been used to tackle. In this paper, we propose a novel method for UIL for different behavioral patterns to determine whether two given behavioral histories come from the same user without using any user demographics. Our proposed method links users by using natural language processing to efficiently characterize user intrinsic features and bridging the gap between two different behavioral patterns of the same user. We conducted experiments to evaluate our proposed method for three real-world open source datasets and observed that it successfully linked users compared to conventional UIL methods.

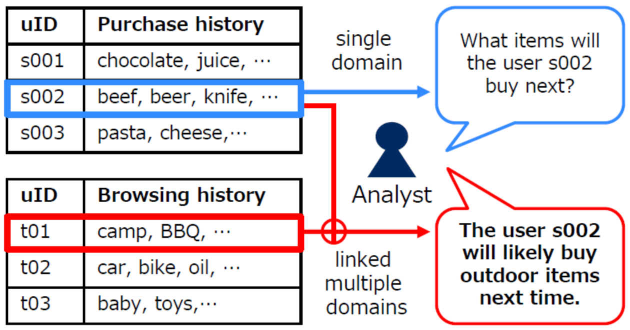


Figure 1: Linking purchase history of target user to her browsing history is useful, for example, to recommend items of category related to sites she visited, measure the effectiveness of advertisements on each web site, and improve web sites to be more attractive to visitors by analyzing their purchase patterns. For these applications, UIL $s002 = t01$ is essential.

Introduction

Background

Understanding customer behavioral patterns from diversified perspectives is key to various marketing tasks. Since customers come across and use many online- and offline-services every day, their activity logs (e.g., browsing history on an electronic commerce (EC) store, purchase history on a retail store, and posts on a social networking service (SNS)) are just the tip of the iceberg, and examining data of a single domain does not always produce enough information. To understand customers in such a situation, tracking customer behaviors among several services is necessary, and this has motivated researchers to carry out *user identity linkage* (UIL), which connects a customer's identity on a service with her identity on a different service. Figure 1 shows a use case of UIL.

Although current application-level integration mechanisms for UIL, such as HTTP cookies and social login systems, successfully connect users among different services, they are facing several difficulties: (1) regulations, such as the EU General Data Protection Regulation (GDPR), that prohibit the use of cookies without user consent [1], and (2) preparing a common authorization system among every possible service that users might use is almost impossible as the number of services that users use is increasing.

To overcome these difficulties, studies on "predictive" UIL methods, which are used to predict the linkage of two users in different disconnected domains, are becoming a popular topic in data mining (Zafarani and Liu 2009; Liu et al. 2014; Shu et al. 2016; Hadgu and Gundam 2020). Predictive UIL methods model a user as a feature vector computed from identity-implying information such as demographics (e.g., gender, age, and addresses) or behaviors (e.g., trajectory data, tagging logs on SNS). To determine whether a user u on domain s is identical to a user u' on do-

[1]While such regulations are concerned that a personal data (e.g., her e-mail or SNS ID) is connected to her activities (e.g., her browsing histories) without user consent, it is desired to utilize user data after ethical concerns are removed. For example, the Personal Information Management System (PIMS) was recently proposed, where users can share their data to utilize them while being assured of their privacy (e.g., anonymization). Therefore, UIL methods are useful to utilize anonymized user data.

main t, these methods compute the features of both users and similarities of these features on the basis of particular metrics. If the similarity satisfies a particular condition, such as surpassing a pre-determined threshold, the methods output a pair of (u, u') as a linked user.

Research Challenges

While previous UIL methods have been successful in terms of identification accuracy, they have limited applicabilities in real-world enterprise marketing due to the following challenges.

Lack of User Demographics In a data management platform (DMP) system, a commonly used marketing platform that gathers customer-related data from various data sources, a user can be anonymized and treated as a hash value (Elmeleegy et al. 2013). Thus, UIL methods using user demographics cannot be applied to datasets in which users are anonymized.

Behavioral Pattern Differences across Domains Recent methods (Iofciu et al. 2011; Goga et al. 2013; Kong, Zhang, and Yu 2013; Riederer et al. 2016; Feng et al. 2019) that do not require demographics information successfully determine the identities of users based on the similarities of user behavioral patterns by focusing on common objects target users react to on both domains [2]. However, in reality, different domains do not always have common objects. Furthermore, even if common objects appear on both domains, user behaviors can be completely different [3]. In such cases, it is difficult for conventional UIL methods to link users whose behaviors in underlying domains are not similar.

Contributions

To address the above challenges, we propose a method for UIL for different behavioral patterns across domains. Our method determines whether two given behavioral histories come from the same user without using any demographics even if underlying domains of user behaviors are not similar.

Our proposed method consists of four main modules: vectorization, mapping, selection, and matching. The vectorization module extracts user intrinsic features as a numerical vector from her behavioral history. As a behavioral history is a time-series of actions, by regarding an action as a word, we efficiently characterize it using natural language processing. In UIL setting, since a target user is registered in two domains, s and t, two vectors are obtained from the same user. The mapping module then transforms a vector of one domain s, into the corresponding vector of the other domain t. One way to obtain such a mapping is to train a regression model by learning the correspondence between two vectors of the same user. Note that a good combination of vectorization and mapping models should depend on a target dataset; hence, our method enables arbitrary combinations. The selection module automatically finds the best combination suitable for a target dataset in a cross-validation manner. Finally, the matching module links users based on the selected combination.

Our proposed method, in fact, contains two conventional UIL methods (Iofciu et al. 2011; Goga et al. 2013) as realizations of certain combinations of vectorization and mapping models. From this point of view, our method generalizes these methods.

Our contributions can be summarized as follows.

- We propose a novel method for UIL for different behavioral patterns across domains. Our method requires only behavioral histories and does not require any user demographics.
- We utilize methods based on natural language processing to efficiently extract user intrinsic features from her behavioral history.
- We bridge the gap between two domains by utilizing the mapping module of our method, which uses the correspondence between different behavioral patterns of the same user.
- We utilize the selection module of our method to automatically find the best combination of vectorization and mapping models for a target dataset.
- We evaluated our method on three real-world open source datasets, in which each behavioral history is of buying food, viewing advertisements, and reviewing items. The results indicate that our method achieved high matching accuracy in identifying across different domains compared to conventional UIL methods.

Preliminaries

Notations

Let U and I be sets of users and items, respectively. A *behavioral history* is defined as a sequence of items in time order. For example, if a user $u \in U$ purchases one banana, two apples, and one salmon, the sequence $b_u = (i_{banana}, i_{apple}, i_{apple}, i_{salmon})$ is naturally seen as the behavioral history, where i indicates the item ID. Since many datasets for items have categorical information, such as bananas and apples being in the same fruit category and salmon being in the fish category, we also use categories of items.

In UIL setting, users act in different domains. Throughout this paper, we fix (U^s, I^s, C^s, B^s) and (U^t, I^t, C^t, B^t) as tuples of sets of users, items, categories, and behavioral histories in domains s and t, respectively. The set of users who are registered in both domains s and t is denoted as $U^{link} := U^s \cap U^t$ and we assume that some linked users are known [4]. In a UIL task, we focus on users who are not

[2]As a typical application example of those methods, suppose a Facebook (domain s) user posted text about a location during her trip and posted photos tagged with the location on Instagram (domain t). In this case, a common object is the location.

[3]In a task of measuring the effectiveness of advertising in which two datasets correspond to clicking (domain s) and viewing (domain t), user behaviors of clicking and viewing differ across domains, though ads appear on both domains. On the other hand, the user's behaviors of the example of Facebook and Instagram are similar since her intrinsic behavior is the same on both domains.

[4]This is a reasonable assumption as many DMPs provide a means to obtain a limited number of already linked users in a pay-

Symbol	Description
$\mathbf{R}$	Set of real numbers
s, t	Domains
U	Set of users
$u \in U$	User
I	Set of items
$i \in I$	Item
B	Set of behavioral histories
$b_u = (i_1, \ldots, i_m) \in B$	Behavioral history of a user u
C	Set of item categories
$c \in C$	Item category
$v : B \to \mathbf{R}^d$	Vectorization model
$\mathbf{v}_u = v(b_u)$	Vector representation of a user u
$f : \mathbf{R}^{d_s} \to \mathbf{R}^{d_t}$	Mapping model

Table 1: List of main symbols.

linked yet. To be precise, we link a user's behavioral history $b_u^s \in B^s$ in domain s with her behavioral history in domain t by searching B^t.

We set $U^{train} \subset U^{link}$ and $U^{test} \subset U^{link} \setminus U^{train}$ as the sets of users who are already known to be linked and users who are not known to be linked yet but in fact linked, respectively [5]. As the symbols indicate, we use U^{train} and U^{test} for supervised learning.

Table 1 summarizes the main symbols used in this paper.

Review on Vectorization for a Document

A behavioral history $b_u = (i_1, \ldots, i_m) \in B$ can be considered a document by considering an item as a word. In what follows, we discuss constructing a vectorization model $v : B \to \mathbf{R}^d$ transforming a behavioral history into a numerical vector. In this section, we briefly review three vectorization models for a document $v_{bow}, v_{tfidf}, v_{bm25}$.

Let B be a set of documents [6], W be the set of all words in B, $n = |B|$ be the number of documents in B, and $d = |W|$ be the number of words in B.

A simple document vectorization method is using the bag-of-words (***BoW***). The vectorization model by BoW is denoted as $v_{bow} : B \to \mathbf{R}^d$. The BoW vector $v_{bow}(b) \in \mathbf{R}^d$ of a document $b \in B$ is defined as $v_{bow}(b)[w] = k$ if b contains word w a total of k times.

Another well-known method is term frequency-inverse document frequency (***TFIDF***, (Jones 2004)). The TFIDF model $v_{tfidf} : B \to \mathbf{R}^d$ is defined as

$$v_{tfidf}(b)[w] := TF[w, b] \times IDF[w]$$

where $TF[w, b] := |b \cap \{w\}|/|b|$ is the term frequency [7] of w in b and $IDF[w] := \log((n + 1)/(D[w] + 1)) + 1$ $(D[w] := |\{b \in B \mid w \in b\}|)$ is the inverse document frequency of w in B.

Although TFIDF measures the word importance, a word in a document with many words tends to have a small weight even if the word is in fact meaningful. To balance the document length in TFIDF, the study (Jones, Walker, and Robertson 2000) proposed *BM25*, another vectorization method, by introducing the average document length $avgDL := n^{-1} \sum_{b' \in B} |b'|$. To be precise, the formulation of the IDF in BM25 is slightly different than that of TFIDF, and it is defined as $IDF'[w] = \log\{(n - D[w] + c)/(D[w] + 1 - c) + 1\}$ where c is a parameter. Then, the BM25 model $v_{bm25} : B \to \mathbf{R}^d$ is defined as

$$v_{bm25}(b)[w] := \frac{TF[w, b](k_1 + 1) \times IDF'[w]}{TF[w, b] + k_1(1 - k_2 + k_2 \frac{|b|}{avgDL})},$$

where k_1 and k_2 are parameters. We set $c = 0.5, k_1 = 2, k_2 = 0.75$, as the original paper (Jones, Walker, and Robertson 2000) proposed.

Proposed Method

Strategy

To solve the UIL problem, we set the intermediate goal of this study to construct a similarity function

$$\mathrm{sim} : B^s \times B^t \to \mathbf{R},\ (b_u^s, b_{u'}^t) \mapsto \mathrm{sim}(b_u^s, b_{u'}^t) \qquad (1)$$

of behavioral histories that returns a high value if $u = u'$. Once such a similarity function is obtained, for a behavioral history b_u^s, we can find the corresponding behavioral history b_u^t by searching for those that have a high similarity to b_u^s.

We briefly explain obtaining the similarity function (1). First, we prepare vectorization models $v^s : B^s \to \mathbf{R}^{d_s}$ and $v^t : B^t \to \mathbf{R}^{d_t}$, where d_s and d_t are positive integers, respectively. Second, we prepare a mapping model $f : \mathbf{R}^{d_s} \to \mathbf{R}^{d_t}$ to make the inferred vector $f(v^s(b_u^s))$ similar to $v^t(b_u^t)$ for linked users. Based on the vectorization model v and mapping model f, we finally define the similarity function by

$$\mathrm{sim}_{v,f}(b_u^s, b_{u'}^t) := \cos(f(v^s(b_u^s)), v^t(b_{u'}^t))$$

where $\cos(\mathbf{v}, \mathbf{v}') := \langle \mathbf{v}, \mathbf{v}' \rangle / \|\mathbf{v}\| \|\mathbf{v}'\|$ $(\mathbf{v}, \mathbf{v}' \in \mathbf{R}^{d_t})$ is the cosine similarity function. The details of v and f are given below.

Vectorization Module

We search for the best vectorization model for a behavioral history among a variety of models; hence, we introduce 30 vectorization models in this subsection. Without loss of generality, we omit the domain notation, fix $B^{train} := \{b_u \mid u \in U^{train}\}$ as a set of behavioral histories of all training users, and regard it as a set of documents.

Item Dimension A vectorization model v_* $(* \in \{bow, tfidf, bm25\})$ trained from B^{train} transforms a behavioral history into a numerical vector, the dimension of which is aligned with items in I. Here, $\mathbf{X}_*$ denote the user-item matrix, e.g., $\mathbf{X}_{bm25}[u, i] = v_{bm25}(b_u)[i]$ for a user $u \in U$ and item $i \in I$.

per-ask questionnaire manner.

[5]The notation $A \setminus B$ for sets A and B indicates the difference set between A and B.

[6]Since we regard a behavioral history as a document, a set of document is also denoted as B in this subsection.

[7]The notation $|A|$ for a set A indicates the cardinality of A.

Category Dimension Since a user-item matrix is generally sparse, we use categories to reduce the sparsity. Although the symbol C is the set of categories, we also regard C as a map transforming an item $i \in I$ to its category $c \in C$ by abuse of notation. Here, $C(b_u) := (c_1, \ldots, c_m)$, where c_k is the corresponding category of an item $i_k \in b_u$, and $C(B) := \{C(b_u) \mid b_u \in B\}$ denote the sequence of categories of b_u and a set of sequences of categories in B, respectively. Now that a vectorization model v_* ($* \in \{bow, tfidf, bm25\}$) can be also trained from $C(B^{train})$, we define the vectorization model $C \circ v_*$ by $(C \circ v_*)(b_u) := v_*(C(b_u))$, the dimension of which is aligned with categories in C [8]. Here, $\mathbf{C}_*$ denote the user-category matrix, e.g., $\mathbf{C}_{bm25}[u, c] = (C \circ v_{bm25})(b_u)[c]$ for a user $u \in U$ and a category $c \in C$.

Concatenation In the above constructions, a concatenation of the three models can be also considered. We define the concatenation model v_{concat} by $v_{concat}(b_u) := [v_{bow}(b_u), v_{tfidf}(b_u), v_{bm25}(b_u)]$. Note that the dimension of v_{concat} is 3 times larger than that of v_* ($* \in \{bow, tfidf, bm25\}$). In the same manner as the category dimension construction, the concatenation model for categories $C \circ v_{concat}$ is well-defined.

Item-Category Matrix Unlike $C \circ v_*$, we can also compress a user-item matrix $\mathbf{X}_*$ by multiplying $\mathbf{X}_*$ by an item-category matrix $\mathbf{I}$, which is defined as $\mathbf{I}[i, c] = 1$ if an item $i \in I$ belongs to a category $c \in C$; otherwise, 0. The corresponding vectorization model is denoted as $v_* \cdot \mathbf{I}$, i.e., $(v_* \cdot \mathbf{I})(b_u) := v_*(b_u) \cdot \mathbf{I}$ where $\cdot$ is the dot product. For the concatenation construction, we set $(v_{concat} \cdot \mathbf{I})(b_u)$ as $[v_{bow}(b_u) \cdot \mathbf{I}, v_{tfidf}(b_u) \cdot \mathbf{I}, v_{bm25}(b_u) \cdot \mathbf{I}]$. Note that $\mathbf{X}_{bow} \cdot \mathbf{I} = \mathbf{C}_{bow}$, but, in general, $\mathbf{X}_* \cdot \mathbf{I} \neq \mathbf{C}_*$ for $* \in \{tfidf, bm25, concat\}$.

Non-Negative Matrix Factorization We consider non-negative matrix factorization (*NMF*, (Lee and Seung 2000)) as another method for compressing a user-item matrix $\mathbf{X}_*$. For a user u and item i, it decomposes $\mathbf{X}_*[u, i]$ as the inner product $\langle \mathbf{P}_*[u], \mathbf{Q}_*[i] \rangle$ of the user vector $\mathbf{P}_*[u] \in \mathbf{R}^k_{\geq 0}$ and item vector $\mathbf{Q}_*[i] \in \mathbf{R}^k_{\geq 0}$, where a positive integer k is the pre-determined parameter for NMF. We treat $\mathbf{P}_*[u]$ as the feature vector of user u and write the corresponding vectorization model as $\mathrm{NMF}_k \circ v_*$, i.e, $(\mathrm{NMF}_k \circ v_*)(b_u) := \mathbf{P}_*[u]$. In the same manner, the NMF model for categories $\mathrm{NMF}_k \circ C \circ v_*$ is well-defined. With respect to parameter k, we try $k = 10, 50, 100$ for $\mathrm{NMF}_k \circ v_*$ and $k = 10$ for $\mathrm{NMF}_k \circ C \circ v_*$.

Doc2Vec Recent studies on deep learning have demonstrated its effectiveness in a variety of field, including document vectorizations. Doc2Vec (Paragraph Vector, (Le and Mikolov 2014)), a deep learning based document vectorization method, learns the distributed representation of documents from B^{train} and returns a pre-defined dimensional vector $v_{doc2vec}(b)$ of a document b. We set the window size and dimension of Doc2Vec as 5 and 300, respectively [9]. In addition, the Doc2Vec model $C \circ v_{doc2vec}$ for a sequence of categories, which is defined as $(C \circ v_{doc2vec})(b_u) := v_{doc2vec}(C(b_u))$, is also considered.

In summary, we listed a total of 30 vectorization models: $v_*, C \circ v_*$ ($* \in \{bow, tfidf, bm25, concat, doc2vec\}$), $v_* \cdot \mathbf{I}$ ($* \in \{bow, tfidf, bm25, concat\}$), $\mathrm{NMF}_k \circ v_*$ ($* \in \{bow, tfidf, bm25, concat\}, k = 10, 50, 100$), and $\mathrm{NMF}_{10} \circ C \circ v_*$ ($* \in \{\text{bow, tfidf, bm25, concat}\}$). Here, $\mathcal{V}$ denotes the set of such vectorization models.

Mapping Module

Let v^s be a vectorization model trained from $B^{s,train} := \{b^s_u \mid u \in U^{train}\}$ and $\mathbf{v}^s_u := v^s(b^s_u) \in \mathbf{R}^{d_s}$ be a vector representations of $b^s_u \in B^s$. In the same manner, $B^{t,train}, v^t, d_t, \mathbf{v}^t_u$ are defined. We introduce mapping models $f : \mathbf{R}^{d_s} \to \mathbf{R}^{d_t}$ between two vectors such that $f(\mathbf{v}^s_u) \approx \mathbf{v}^t_u$ for linked users $u \in U^{link}$ in this subsection.

We treat a mapping model $f : \mathbf{v}^s_u \mapsto \mathbf{v}^t_u$ as a regression model trained from $\{(\mathbf{v}^s_u, \mathbf{v}^t_u) \mid u \in U^{train}\}$ because each pair $(\mathbf{v}^s_u, \mathbf{v}^t_u)$ comes from the same user u. We consider regression models of linear ridge f_{linear}, kernel ridge f_{kernel} with the radial basis function (RBF) kernel, and ℓ-layer neural networks $f_{\ell nn}$. To be precise, we set parameters in the ridge regression models by grid search with cross-validation. With respect to neural networks, we consider a two-layer neural network f_{2nn} with one 100-dimensional hidden layer and three-layer neural network f_{3nn} with two 100-dimensional hidden layers, use Adam (Kingma and Ba 2015) as the optimization algorithm and ReLU (Nair and Hinton 2010) as the activation function, and set the batch size as 100 and the epoch size as 100.

The identity map f_{id} is also considered if the two vectors $\mathbf{v}^s_u$ and $\mathbf{v}^t_u$ have the same dimension. For v_* ($* \in \{bow, tfidf, bm25, concat\}$), $d_s \neq d_t$ in general. To arrange for the dimensions to be the same, we reset the coordinates of each vectorization model to all items in $I^s \cup I^t$ and denote the reset vector of $\mathbf{v}^s_u$ as $\tilde{\mathbf{v}}^s_u \in \mathbf{R}^d$ where $d = |I^s \cup I^t|$, which is defined as $\tilde{\mathbf{v}}^s_u[i] = \mathbf{v}^s_u[i]$ ($i \in I^s$); 0 ($i \in I^t \setminus I^s$). In the same manner, the identity map for vectors, the dimensions of which are aligned with categories, is well-defined by resetting the dimensions as $C^s \cup C^t$.

In summary, we introduced a total of 5 mapping models: $f_{linear}, f_{kernel}, f_{2nn}, f_{3nn}, f_{id}$. Here, $\mathcal{F}$ denotes the set of such mapping models.

Selection Module

In this subsection, we discuss searching for the best combination of models $(v, f) \in \mathcal{V} \times \mathcal{F}$ to link users by the similarity function $\mathrm{sim}_{v,f}$ with higher accuracy.

[8]The notation $\circ$ denotes the composition of mappings. For two maps $f : X \to Y, g : Y \to Z$, the *composition* $g \circ f : X \to Z$ is defined by $(g \circ f)(x) := g(f(x))$ ($x \in X$).

[9]We applied the Gensim module (https://radimrehurek.com/gensim/models/doc2vec.html) in experiments and other parameters were set to default values. Since document sets of behavioral histories are not real documents, any released pre-trained models cannot be transferred and we train all Doc2Vec model from scratch.

Algorithm 1 Selection Module

```
Input: A set of training users $U^{train}$, a positive integer $K \geq 1$
Output: A vectorization model $\hat{v}$, a mapping model $\hat{f}$
    Initialization : Sets of models of vectorizations $\mathcal{V}$ and mappings $\mathcal{F}$ with randomly initialized parameters.
 1: Divide $U^{train}$ into $K$ (mostly) equally sized parts $U_1, \ldots, U_K$.
 2: for $k = 1, \ldots, K$ do
 3:     Set $U_{cv}^{te} := U_k$ and $U_{cv}^{tr} := U^{train} \setminus U_k$.
 4:     for $v$ in $\mathcal{V}$ do
 5:         Train $v^s$ from $\{b_u^s \mid u \in U_{cv}^{tr}\}$.
 6:         Train $v^t$ from $\{b_u^t \mid u \in U_{cv}^{tr}\}$.
 7:         for $f$ in $\mathcal{F}$ do
 8:             Train $f$ from $\{(v^s(b_u^s), v^t(b_u^t)) \mid u \in U_{cv}^{tr}\}$.
 9:             Calculate $s_k(v, f) = \mathrm{MRR}_{v,f}(U_{cv}^{te}, U^{train})$.
10:         end for
11:     end for
12: end for
13: Compute the mean $\overline{s}(v, f) := K^{-1} \sum_{k=1}^{K} s_k(v, f)$.
14: Select the best combination as
        $$\hat{v}, \hat{f} = \operatorname*{argmax}_{v \in \mathcal{V}, f \in \mathcal{F}} \overline{s}(v, f).$$
```

To evaluate the performance of the similarity function in a UIL task, we use the mean reciprocal rank (MRR). For UIL, the MRR depends on the similarity function $\mathrm{sim}_{v,f}$, a set of test users U^{test}, and set of candidate users $U^{candi} (\supset U^{test})$ from whom we search for linked users of test users. Once $\mathrm{sim}_{v,f}$ is obtained, for each $u \in U^{test}$, it sorts candidate users as $u_1, \ldots, u_N \in U^{candi}$ by $\mathrm{sim}_{v,f}(b_u^s, b_{u_1}^t) \geq \mathrm{sim}_{v,f}(b_u^s, b_{u_2}^t) \geq \cdots$. If $b_u^t = b_{u_{r_u}}^t$, the rank of u is given by r_u. Then, the MRR is defined as

$$\mathrm{MRR}_{v,f}(U^{test}, U^{candi}) := \frac{1}{|U^{test}|} \sum_{u \in U^{test}} \frac{1}{r_u}.$$

The notation $\mathrm{MRR}_{v,f}(U^{test}, U^{candi})$ is used to emphasize the dependency of $v, f, U^{test}, U^{candi}$.

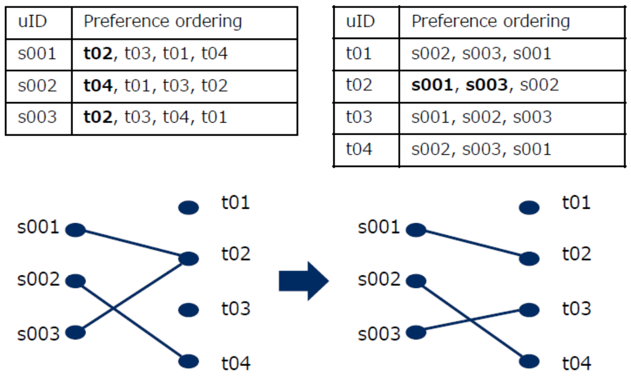
By using the MRR, we search for the best combination of models $(v, f) \in \mathcal{V} \times \mathcal{F}$ in the K-fold cross-validation [10]. First, the set of training users U^{train} is divided into K (mostly) equally sized parts $U_1, \ldots, U_K$. Let $U_{cv}^{tr} := U^{train} \setminus U_k$ and $U_{cv}^{te} := U_k$. Second, we train models v and f from the behavioral histories of U_{cv}^{tr} and calculate the MRR $s_k(v, f) := \mathrm{MRR}_{v,f}(U_{cv}^{te}, U^{train})$. We repeat this procedure from $k = 1$ to K and calculate the mean $\overline{s}(v, f) := K^{-1} \sum_{k=1}^{K} s_k(v, f)$. Finally, we select the models achieving the highest mean MRR from $\mathcal{V} \times \mathcal{F}$.

The pseudo-code for searching for the best model combination is given in Algorithm 1.

Matching Module

In Algorithm 1, we use the MRR to measure the performance of the similarity function on UIL. The top-k accuracy

[10]In our experiments, we set $K = 3$.



uID	Preference ordering
s001	**t02**, t03, t01, t04
s002	**t04**, t01, t03, t02
s003	**t02**, t03, t04, t01

uID	Preference ordering
t01	s002, s003, s001
t02	**s001**, **s003**, s002
t03	s001, s002, s003
t04	s002, s003, s001

Figure 2: Example of using the Gale-Shapley algorithm. Matching by highest similarity score links $t02$ with $s001$ and $s003$ from preferences of $s001$ and $s003$ (left graph). Then, the Gale-Shapley algorithm links $t02$ with $s001$ because $t02$ prefers $s001$ to $s003$ and the unmatched user $s003$ with $t03$ from her next preference (right graph).

is also widely used in UIL and it is defined by

$$\mathrm{Acc}_k := \frac{1}{|U^{test}|} \sum_{u \in U^{test}} \mathrm{id}(r_u \leq k)$$

where id is the indicator function[11] and r_u is the rank of u.

In particular, the top-1 accuracy measures the matching accuracy by the highest similarity score, i.e.,

$$\mathrm{id}(r_u = 1) = \mathrm{id}\left(u = \operatorname*{argmax}_{u' \in U^{candi}} \mathrm{sim}_{v,f}(b_u^s, b_{u'}^t)\right).$$

However, this matching has the problem of the one-to-one mapping of users. In other words, two users in domain s is predicted to link to the same user in domain t, which is a contradiction in terms in UIL.

To avoid the above contradiction, we apply the Gale-Shapley algorithm (Gale and Shapley 1962), which outputs a set of one-to-one pairs of two vertices in a bipartite graph, to our proposed method (Figure 2). Note that the authors (Kong, Zhang, and Yu 2013) also applied this algorithm to link users satisfying the one-to-one mapping constraint.

To apply the Gale-Shapley algorithm to the UIL problem, we first compute $\mathrm{sim}(b_u^s, b_{u'}^t)$ for all pairs of users in $u \in U^{test}$ and $u' \in U^{candi}$. We then obtain the sequences $(b_{u_1}^t, b_{u_2}^t, \ldots)$ and $(b_{u'_1}^s, b_{u'_2}^s, \ldots)$ of behavioral histories sorted by the similarity function, respectively. The Gale-Shapley algorithm takes lists of orderings of preferences $\{(b_{u_1}^t, b_{u_2}^t, \ldots) \mid u \in U^{test}\}$ and $\{(b_{u'_1}^t, b_{u'_2}^s, \ldots) \mid u' \in U^{candi}\}$ as input, and outputs a set of one-to-one linked pairs $\{(b_u^s, b_{\hat{u}}^t) \mid u \in U^{test}\}$.

The pseudo-code for linking users by using the Gale-Shapley algorithm is given in Algorithm 2.

We define matching accuracy as

$$\mathrm{Acc}_{\mathrm{GS}} := \frac{1}{|U^{test}|} \sum_{u \in U^{test}} \mathrm{id}(u = \hat{u})$$

[11]For an event E, the indicator function $\mathrm{id}(E)$ returns 1 if E is true; otherwise, 0.

Algorithm 2 One-to-One Matching Algorithm

```
Input: A set of test and candidate users $U^{test}$ and $U^{candi}$, similarity function $\mathrm{sim}_{v,f}$
Output: A set of one-to-one matching pairs $\{(b^s_u, b^t_{\hat{u}}) \mid u \in U^{test}\}$
1: Sort the preference $(b^t_{u_1}, b^t_{u_2}, \cdots)$ for each $u \in U^{test}$ such that $\mathrm{sim}(b^s_u, b^t_{u_1}) \geq \mathrm{sim}(b^s_u, b^t_{u_2}) \geq \cdots$.
2: Sort the preference $(b^s_{u'_1}, b^s_{u'_2}, \cdots)$ for each $u' \in U^{candi}$ such that $\mathrm{sim}(b^s_{u'_1}, b^t_{u'}) \geq \mathrm{sim}(b^s_{u'_2}, b^t_{u'}) \geq \cdots$.
3: Run the Gale-Shapley algorithm on these preference sequences.
4: Obtain $\{(b^s_u, b^t_{\hat{u}}) \mid u \in U^{test}\}$.
```

where $\hat{u}$ is the predicted user for u from Algorithm 2. Since Algorithm 2 solves the one-to-one matching problem, $\mathrm{Acc}_{\mathrm{GS}}$ should be higher than Acc_1.

Experiments

Dataset

For our experiments to evaluate our proposed method, we used the Instacart, Click-Through Rate (CTR), and Amazon datasets and divided each dataset into two domains s and t to match the setting in UIL.

- The Instacart dataset[12] contains users' food-purchase histories in one year on an EC site, called Instacart. In this dataset, the exact timestamp for a purchase is hidden, and time T is replaced with the number of days that have passed from the day a user made a purchase for the first time. Since all users purchase items at $T = 0$ but a smaller number still purchase items after $T > 300$, we used $T = 100$ to balance the number of items many users purchase before and after T. Then, we set domains s and t as a set of purchase histories before and after $T = 100$, respectively.
- The CTR dataset[13] contains users' click logs for online advertisements. In this dataset, we regard device_ip, site_id, and site_category as column names of users, items, and categories, respectively. Since the CTR dataset has logs on whether a site was clicked, we set domains s and t as a set of clicking and viewing histories, respectively.
- The Amazon dataset[14] (Ni, Li, and McAuley 2019) contains users' reviewing histories on Amazon. We regarded a reviewed item as an item attached to a user behavioral history and used "Books" and "Kindle Store" data of "5-core in Small subsets for experimentation" as domain s and t, respectively.

According to (Khan, Ibrahim, and Ghani 2017), who conducted a survey study of cross-domain recommendation, the difference in domains are classified into several levels. Using the notions introduced in that paper, the Instacart, CTR, and Amazon datasets correspond to reproducing situations of domain difference in the levels of time, system, and item, respectively. Although we treat two user datasets divided from one ground-truth dataset on the same platform as two domains for evaluation, these datasets are naturally seen as ones from different platforms; for example, online viewing ads and offline interacting histories (CTR) and purchase histories on different real stores (Amazon). In real-world applications, the settings of the CTR and Amazon datasets appear as tasks of measuring the effectiveness of advertising and cross-domain recommendation, respectively.

Number of Users We fixed the number of training and test users as $5,000$ and 500, respectively, for all three datasets. To guarantee the quality of users, we imposed a restriction that a training user should have more than 50 items in both domains and prepared the training data U^{train} with a set of $5,000$ of such users randomly selected from linked users. Similar to the training data, we prepared the test data U^{test} with a set of users who had more than 50 items in domain s [15]. The maximum number of candidate users $U^t \setminus U^{train}$ is shown in the Candidates column in Table 2.

Observations To analyze datasets, we calculate the similarities of two domains by the Jaccard similarity, which measures the similarity between two sets A and B as $\mathrm{Jac}(A, B) := |A \cap B|/|A \cup B|$. To be precise, we calculate these similarities on the training dataset, that is, the Jaccard similarities for items and categories were calculated as $\mathrm{Jac}(I^{s,train}, I^{t,train})$ and $\mathrm{Jac}(C^{s,train}, C^{t,train})$ where I^{train} and C^{train} are sets of all items and categories of training users, respectively, and the values are listed in the Jac(D) rows in Table 2. As $\mathrm{Jac}(b^s_u, b^t_u)$ can also be calculated for each training user $u \in U^{train}$, the average and standard deviation of the Jaccard similarities for training users are listed in the Jac(U-inter) rows in Table 2. If Jac(U-inter) is high, it is considered to easily link users because users show similar behaviors across domains. We also calculated $J_z := \{\mathrm{Jac}(b^z_u, b^z_{u'}) \mid u \neq u' \in U^z\}$ for each domain $z \in \{s, t\}$ and listed the mean and standard deviation of J_z in the Jac(U-z) rows in Table 2. If Jac(U-t) is high, it is considered difficult to identify the correct user in domain t because there are many similar users in domain t who are linked to the target user in domain s.

We define the linkability by Jac(U-inter) and identifiability by 1 minus the mean of Jac(U-s) and Jac(U-t). Figure 3 illustrates the relations among datasets from the viewpoint of the linkability and the identifiability.

From Table 2 and Figure 3, we characterize each dataset as follows.

- The Instacart dataset had both higher linkability and identifiability in terms of items. The score of the user-wise inter-domain similarity Jac(U-inter) of items was 0.25 ± 0.1, indicates most users bought the same items before and after T.

[12]https://www.kaggle.com/c/instacart-market-basket-analysis/data

[13]https://www.kaggle.com/c/avazu-ctr-prediction/data

[14]https://nijianmo.github.io/amazon/index.html

[15]In this situation, the behaviors of any test users in the domain t are unknown; thus, any assumptions cannot be imposed on test users on domain t.

	Instacart	CTR	Amazon
Item			
s	28,356	1,907	83,423
t	30,338	2,749	209,762
Jac(D)	0.67	0.69	0
Jac(U-inter)	0.25±0.1	0.48±0.17	0
Jac(U-s)	0.02±0.02	0.17±0.174	<0.01
Jac(U-t)	0.02±0.02	0.15±0.12	<0.01
Category			
s	134	1,295	15
t	134	1,823	88
Jac(D)	1	0.71	0.17
Jac(U-inter)	0.5±0.1	0.49±0.1	0.54±0.17
Jac(U-s)	0.25±0.08	0.19±0.14	0.70±0.17
Jac(U-t)	0.28±0.09	0.16±0.12	0.64±0.17
Candidates	130,974	6,729,486	1,856,344

Table 2: Summaries of datasets. Entries in the s and t rows indicate the number of items and categories in the training dataset.

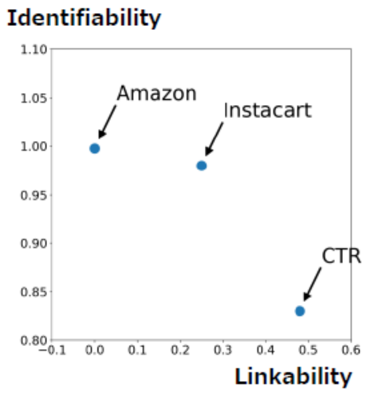


Figure 3: The linkability and identifiability in terms of items for each dataset.

- The CTR dataset had lower identifiability in terms of items. Even though $b_u^s \supset b_u^t$ for all users u from the construction, Jac(U-inter) was almost half, which indicates most users did not click half the advertisements.
- The Amazon dataset had lower linkability in terms of items because items in two domains were completely disjointed.
- All datasets had higher linkability and lower identifiability in terms of categories.

We also conducted an additional experiment about the number of behaviors of test users. If $|b_u^s| \approx |b_u^t|$, the dataset is considered to have a bias in terms of the number of behaviors [16]. We calculated the ratio $|b_u^s|/|b_u^t|$ for $u \in U^{test}$.

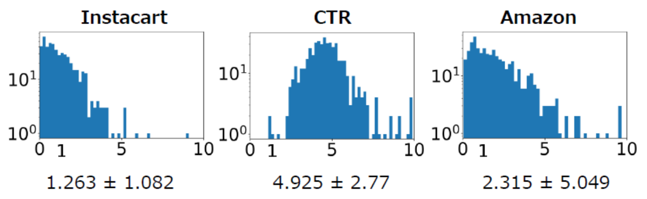


Figure 4: The histogram of ratio $|b_u^s|/|b_u^t|$ (truncated over 10). Each value $m \pm s$ below the histogram means the mean and standard deviation of the histogram. Regarding the Instacart dataset, although the mean is indeed close to 1 because we set $T = 100$ to balance the number of items many users purchase before and after T, the standard deviation is large.

Remember that $|b_u^s| \geq 50$ to guarantee the quality of test users and there is no restriction on $|b_u^t|$. From Figure 4, it is observed that any datasets do not have a strong peak; hence, we conclude that our datasets have no bias in terms of the number of behaviors.

Results

Main result Table 3 indicates the selected combination of models from using our proposed method, number of candidate users U^{candi}, MRR, matching accuracy from using Algorithm 2, top-k accuracies ($k = 1, 5, 10, 50$) on the Instacart, CTR, and Amazon datasets.

We first discuss the case of $10,000$ candidate users to evaluate each result in a fair situation in terms of the number of users [17]. The matching accuracies (Acc$_{\text{GS}}$) for the Instacart, CTR, and Amazon datasets were $92.2, 54.0, 86.8\%$, respectively, which suggests that our proposed method successfully linked users with high accuracy. Regarding the CTR dataset, which corresponds to a real-world task of measuring the effectiveness of advertising, our method could link users with a matching accuracy of 54%, which is surprising because Cookie Sync, a UIL method using HTTP cookie, has been reported to link users with a matching accuracy of $62 - 73\%$ (Englehardt and Narayanan 2016; Papadopoulos, Kourtellis, and Markatos 2019). Remember that the 3rd party cookie is being restricted and our proposed method does not use cookies. The selection module also successfully created a similarity function suitable for each target dataset since selected models differed on each dataset. It is remarkable that the matching accuracy on the Amazon dataset was 86.8%, while item sets of this dataset had no intersection, as shown in Table 2. The gap between non-overlapping domains is considered closed by the mapping module induced by the regression model f_{linear}. Comparing Acc$_{\text{GS}}$ with Acc$_1$, we confirmed that the one-to-one mapping by using Algorithm 2 improved matching accuracy.

[16]In such a situation, a user who purchased m items in domain s and a user who purchased around m items in domain t are likely to be linked regardless of the contents of items. This analysis is inspired by the work (Goga et al. 2015), which found out that users who filled out their profiles in domain s were likely to fill out them in domain t and easily to be linked.

[17]Then, the chance rate of matching accuracy is 0.01%, which is the worst case. Note also that $\text{MRR}_{v,f}(U^{test}, U^{candi,1}) \geq \text{MRR}_{v,f}(U^{test}, U^{candi,2})$ if $U^{test} \subset U^{candi,1} \subset U^{candi,2}$.

Dataset	Selected Models	Candidate users	MRR	$\mathrm{Acc}_{\mathrm{GS}}$	Acc_1	Acc_5	Acc_{10}	Acc_{50}
Instacart	(v_{concat}, f_{id})	10,000	93.4	92.2	92.0	95.0	95.8	98.2
		130,974	87.3	84.2	84.2	90.8	92.0	94.6
CTR	(v_{bm25}, f_{id})	10,000	56.1	54.0	50.4	61.6	69.0	79.2
		6,729,486	32.3	29.2	29.2	36.4	38.2	43.6
Amazon	(v_{bm25}, f_{linear})	10,000	76.2	86.8	65.0	91.0	95.6	97.6
		1,856,344	12.8	4.0	2.8	20.6	36.2	72.8

Table 3: Results of our experiments. The MRR and accuracies Acc were listed by percentage value (%).

For the maximum number of candidate users $U^{candi} = U^t \setminus U^{train}$, the matching accuracy decreased. However, the decrease rates were moderate ($92.2 \rightarrow 84.2\%$ for the Instacart dataset and $54.0 \rightarrow 29.2\%$ for the CTR dataset) although the number of candidate users in the Instacart and CTR datasets became over 13 and 672 times larger, respectively. Unfortunately the matching accuracy on the Amazon dataset decreased drastically ($86.8 \rightarrow 4.0\%$), but the top-50 accuracy remained high ($97.6 \rightarrow 72.8\%$).

Ablation Study In this subsection, we focus on models that were not selected with the selection module.

As seen in Table 3, we observed that different combinations of (v, f) achieved the highest MRR depending on datasets, which indicates that the selection module of our method always selects the best combinations of models (v, f) depending on the dataset. In fact, realizations of (v_{bm25}, f_{id}) and (v_{tfidf}, f_{id}) correspond to UIL methods in (Iofciu et al. 2011) and (Goga et al. 2013), respectively; hence, our proposed method which generalizes those methods always shows better performances as far as the selection module prevents overfitting.

Figure 5 illustrates the average MRRs $\overline{s}(v, f)$ in the cross validation (Algorithm 1) of selected combinations of vectorization and mapping models. Each axis represents (whether a behavioral history is seen as a sequence of items or categories)-(which vectorization is used for a behavioral history)-(how to reduce the dimension of the vector)@(which mapping model is used), e.g., item-concat-none@Identity means (v_{concat}, f_{id}) and category-tfidf-nmf_10@MLPRegressor_2 means $(\mathrm{NMF}_{10} \circ C \circ v_{tfidf}, f_{2nn})$. In each bar, the top 5 values are sorted in descending order from the highest MRR, middle 7 values were models achieving the highest MRR where one model was fixed as category dimension, NMF, Doc2Vec, linear ridge, kernel ridge, neural network, and identity map, and the bottom 2 values were the combinations of (Iofciu et al. 2011) and (Goga et al. 2013).

Among the three datasets, item dimensional vectorizations, that is, item-$*$-none ($* \in$ {bow, tfidf, bm25 concat}) patterns, obtained higher MRR than the other combinations of models (Nos. 1 - 5 in Table 5). The methods where one model was fixed as category dimension, NMF, Doc2Vec, kernel ridge, and neural network linked users at a certain degree of accuracy, but the values were lower than those of item dimension. The linear regression also did not outperform the identity map on the Instacart and CTR datasets, but significant differences were not observed. The identity map did not work on the Amazon dataset because item sets of this dataset had no intersections. With respect to regression models, linear ridge showed better performances in this order $f_{linear} > f_{kernel} > f_{2nn} (> f_{3nn})$. It was interesting to observe that all selected combinations of models were uncomplicated, whereas we initially expected that deep learning based models (e.g., a modification of $v_{doc2vec}$ and f_{nn}) linked users with high accuracy, as is the case that some domain specific state-of-the-art UIL methods use deep learning (Zhou et al. 2018; Feng et al. 2019). The selection module also helped prevent these prejudices by searching for the best models on each dataset.

Method	Instacart	CTR	Amazon
Our method	**99.49**	**62.57**	**54.49**
(Iofciu et al. 2011)	99.42	**62.57**	0.42
(Goga et al. 2013)	98.04	37.14	0.42

Table 4: Details of average MRRs (%).

Table 4 indicates the results from comparing our proposed method with conventional methods of (Iofciu et al. 2011) and (Goga et al. 2013). Our method showed the highest MRRs than the other methods. Regarding the Amazon dataset, the score of our method was 54.49 while the methods of (Iofciu et al. 2011; Goga et al. 2013) using overlapped items in both domains could not link users.

Related Works

User Identity Linkage The UIL problem has been studied in a variety of fields, and the methods differ depending on the type of data associated with users. There are basically three types of data: user-profile, user-network, and user-content, as classified in a survey study (Shu et al. 2016). Typically, user-profile information corresponds to username, biography, education, and gender; user-network information corresponds to the social graphs of users, such as the follower/following relationships in Twitter; and user-content information corresponds to sentences, images, and activity logs. UIL methods have been proposed, for example, for when the type of data is given in the form of only user-profile information (Zafarani and Liu 2009, 2013; Mu et al. 2016), only user-network information (Man et al. 2016; Zhou et al. 2018; Zhong et al. 2018; Zhou et al. 2018), and combinations of user- profile, network, and content information (Liu et al. 2014; Hadgu and Gundam 2020).

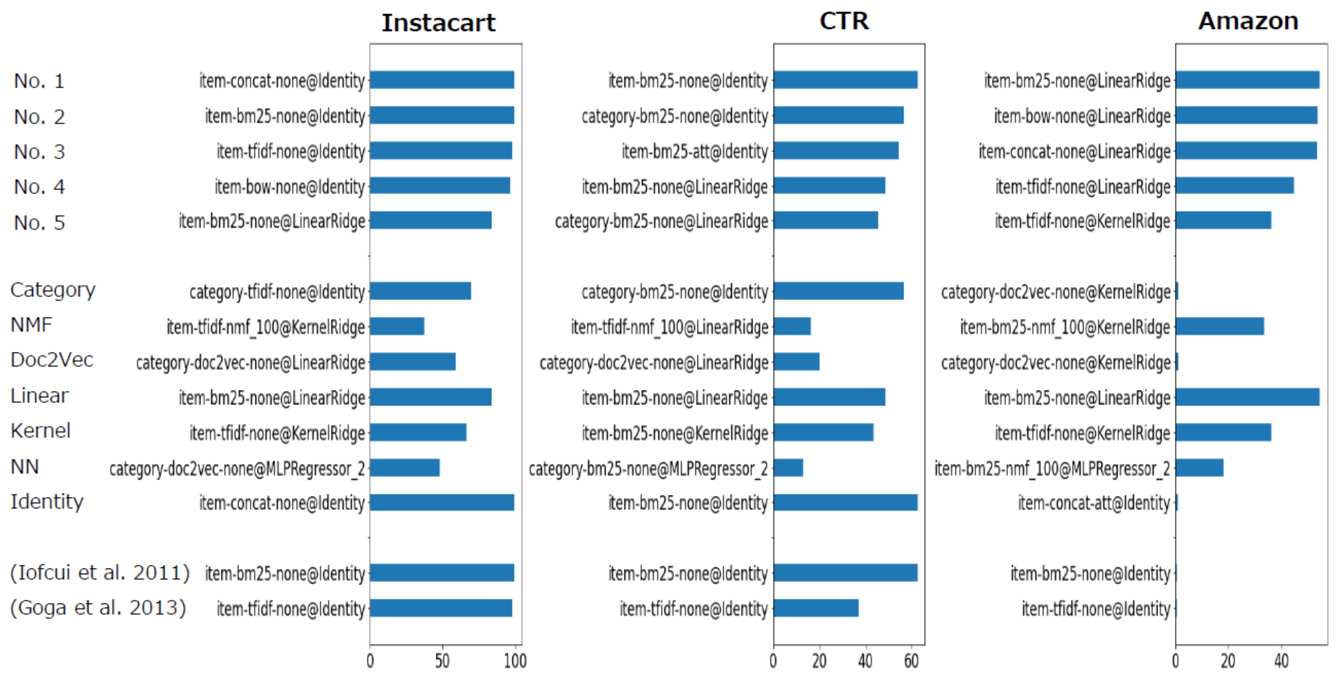


Figure 5: Average MRRs (%) in Algorithm 1

De-anonymization While most studies on UIL have developed methods for receiving the benefits of linking users, studies on de-anonymization (Frankowski et al. 2006; Narayanan and Shmatikov 2008; Gambs, Killijian, and del Prado Cortez 2014; Perez, Musolesi, and Stringhini 2018) warn us that our accounts are identified even if our information is anonymized to protect against linking users through malicious attacks. Although the aims are different in terms of benefits and warnings, algorithms for de-anonymization are similar to those for UIL.

User Identity Linkage for Behaviors Data in our problem setting, which is given as a user behavioral history, are categorized as user-content information. Many methods for user-content information focus on user trajectory data (Goga et al. 2013; Kong, Zhang, and Yu 2013; Riederer et al. 2016; Feng et al. 2019), that is, a user action of visiting a place. Unlike our problem setting, most methods require domain-specific attributes, e.g., the geographic coordinate system (latitude and longitude), grids, and zip-codes; hence, they cannot be applied to our problem.

Domain-independent UIL methods for behavioral histories have been propose in (Iofciu et al. 2011) and the TFIDF version of (Goga et al. 2013). As the authors (Iofciu et al. 2011) explained, the tag "tools" in one social network often appear while the tag rarely appeared in the other social network, the authors proposed to train the BM25 on each social network independently and aggregated tags to arrange for the dimensions to be aligned. While they succeeded in linking users on datasets by focusing on the difference in tag frequency, they implicitly assumed that multiple tags should appear in both social networks.

Conclusion

We proposed a novel method for user identity linkage with which behavioral patterns of the same user can be different on each domain and there is no user demographics. Our proposed method consists of a vectorization module to extract user intrinsic features and a mapping module to bridge the gap between two different behavioral patterns of the same user. Among a variety of such models, our method automatically selects the best combination suitable for a target dataset in a cross-validation manner. In our experiments, test users in the real-world datasets of the Instacart, CTR, and Amazon were correctly linked with $92.2, 54.0, 86.8\%$ matching accuracy from $10,000$ candidate users, respectively. The higher matching accuracy on the Amazon dataset, where two domains had no overlapping, was remarkable, while conventional UIL methods failed to link users.

Acknowledgments

We are especially grateful to Dr. Yuyang Dong and Hao Jia for useful comments and discussions.

References

Elmeleegy, H.; Li, Y.; Qi, Y.; Wilmot, P.; Wu, M.; Kolay, S.; Dasdan, A.; and Chen, S. 2013. Overview of Turn Data Management Platform for Digital Advertising. *Proc. VLDB Endow.* 6(11): 1138–1149.

Englehardt, S.; and Narayanan, A. 2016. Online Tracking: A 1-million-site Measurement and Analysis. In *CCS*, 1388–1401. ACM.

Feng, J.; Zhang, M.; Wang, H.; Yang, Z.; Zhang, C.; Li, Y.; and Jin, D. 2019. DPLink: User Identity Linkage via

Deep Neural Network From Heterogeneous Mobility Data. In *WWW*, 459–469. ACM.

Frankowski, D.; Cosley, D.; Sen, S.; Terveen, L. G.; and Riedl, J. 2006. You are what you say: privacy risks of public mentions. In *SIGIR*, 565–572. ACM.

Gale, D.; and Shapley, L. S. 1962. College Admissions and the Stability of Marriage. *The American Mathematical Monthly* 69(1): 9–15.

Gambs, S.; Killijian, M.; and del Prado Cortez, M. N. 2014. De-anonymization attack on geolocated data. *J. Comput. Syst. Sci.* 80(8): 1597–1614.

Goga, O.; Lei, H.; Parthasarathi, S. H. K.; Friedland, G.; Sommer, R.; and Teixeira, R. 2013. Exploiting innocuous activity for correlating users across sites. In *WWW*, 447–458. International World Wide Web Conferences Steering Committee / ACM.

Goga, O.; Loiseau, P.; Sommer, R.; Teixeira, R.; and Gummadi, K. P. 2015. On the Reliability of Profile Matching Across Large Online Social Networks. In *KDD*, 1799–1808. ACM.

Hadgu, A. T.; and Gundam, J. K. R. 2020. Learn2Link: Linking the Social and Academic Profiles of Researchers. In *ICWSM*, 240–249. AAAI Press.

Iofciu, T.; Fankhauser, P.; Abel, F.; and Bischoff, K. 2011. Identifying Users Across Social Tagging Systems. In *ICWSM*. The AAAI Press.

Jones, K. S. 2004. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 60(5): 493–502.

Jones, K. S.; Walker, S.; and Robertson, S. E. 2000. A probabilistic model of information retrieval: development and comparative experiments. *Inf. Process. Manag.* 36(6): 779–840.

Khan, M. M.; Ibrahim, R.; and Ghani, I. 2017. Cross Domain Recommender Systems: A Systematic Literature Review. *ACM Comput. Surv.* 50(3): 36:1–36:34.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR*.

Kong, X.; Zhang, J.; and Yu, P. S. 2013. Inferring anchor links across multiple heterogeneous social networks. In *CIKM*, 179–188. ACM.

Le, Q. V.; and Mikolov, T. 2014. Distributed Representations of Sentences and Documents. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, 1188–1196. JMLR.org.

Lee, D. D.; and Seung, H. S. 2000. Algorithms for Non-negative Matrix Factorization. In *NIPS*, 556–562. MIT Press.

Liu, S.; Wang, S.; Zhu, F.; Zhang, J.; and Krishnan, R. 2014. HYDRA: large-scale social identity linkage via heterogeneous behavior modeling. In *SIGMOD Conference*, 51–62. ACM.

Man, T.; Shen, H.; Liu, S.; Jin, X.; and Cheng, X. 2016. Predict Anchor Links across Social Networks via an Embedding Approach. In *IJCAI*, 1823–1829. IJCAI/AAAI Press.

Mu, X.; Zhu, F.; Lim, E.; Xiao, J.; Wang, J.; and Zhou, Z. 2016. User Identity Linkage by Latent User Space Modelling. In *KDD*, 1775–1784. ACM.

Nair, V.; and Hinton, G. E. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML*, 807–814. Omnipress.

Narayanan, A.; and Shmatikov, V. 2008. Robust De-anonymization of Large Sparse Datasets. In *IEEE Symposium on Security and Privacy*, 111–125. IEEE Computer Society.

Ni, J.; Li, J.; and McAuley, J. J. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *EMNLP/IJCNLP (1)*, 188–197. Association for Computational Linguistics.

Papadopoulos, P.; Kourtellis, N.; and Markatos, E. P. 2019. Cookie Synchronization: Everything You Always Wanted to Know But Were Afraid to Ask. In *WWW*, 1432–1442. ACM.

Perez, B.; Musolesi, M.; and Stringhini, G. 2018. You Are Your Metadata: Identification and Obfuscation of Social Media Users Using Metadata Information. In *ICWSM*, 241–250. AAAI Press.

Riederer, C. J.; Kim, Y.; Chaintreau, A.; Korula, N.; and Lattanzi, S. 2016. Linking Users Across Domains with Location Data: Theory and Validation. In *WWW*, 707–719. ACM.

Shu, K.; Wang, S.; Tang, J.; Zafarani, R.; and Liu, H. 2016. User Identity Linkage across Online Social Networks: A Review. *SIGKDD Explorations* 18(2): 5–17.

Zafarani, R.; and Liu, H. 2009. Connecting Corresponding Identities across Communities. In *ICWSM*. The AAAI Press.

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

Zhong, Z.; Cao, Y.; Guo, M.; and Nie, Z. 2018. CoLink: An Unsupervised Framework for User Identity Linkage. In *AAAI*, 5714–5721. AAAI Press.

Zhou, F.; Liu, L.; Zhang, K.; Trajcevski, G.; Wu, J.; and Zhong, T. 2018. DeepLink: A Deep Learning Approach for User Identity Linkage. In *INFOCOM*, 1313–1321. IEEE.