Joint Link-Attribute User Identity Resolution in Online Social Networks

Sergey Bartunov
ISPRAS
sbartunov@gmail.com

Anton Korshunov
ISPRAS
korshunov@ispras.ru

Seung-Taek Park
DMC R&D Center,
Samsung Electronics Co., Ltd.
seungtaek.park@samsung.com

Wonho Ryu
DMC R&D Center,
Samsung Electronics Co., Ltd.
wonho.ryu@samsung.com

Hyungdong Lee
DMC R&D Center,
Samsung Electronics Co., Ltd.
h.dong.lee@samsung.com

ABSTRACT

In the modern Web, it is common for an active person to have several profiles in different online social networks. As new general-purpose and niche social network services arise every year, the problem of social data integration will likely remain actual in the nearest future. Discovering multiple profiles of a single person across different social networks allows to merge all user's contacts from different social services or compose more complete social graph that is helpful in many social-powered applications. In this paper we propose a new approach for user profile matching based on Conditional Random Fields that extensively combines usage of profile attributes and social linkage. It is extremely suitable for cases when profile data is poor, incomplete or hidden due to privacy settings. Evaluation on Twitter and Facebook sample datasets showed that our solution significantly outperforms common attribute-based approach and is able to find matches that are not discoverable by using only profile information. We also demonstrate the importance of social links for identity resolution task and show that certain profiles can be matched based only on social relationships between online social networks users.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data mining

General Terms

Algorithms, Experimentation

Keywords

Identity resolution, social network, Conditional Random Fields, profile integration, web mining, de-anonymization

The 6th SNA-KDD Workshop '12 (SNA-KDD'12) August 12, 2012, Beijing, China.

1. INTRODUCTION

Motivation: One of the major hindrances to exploitation of social network data is the fragmentation of its population into numerous proprietary networks with different purposes. Despite attempts to introduce universal solutions that allow to cross-platform information exchange and usage (like Google's OpenSocial initiative[1]), there are still many media applications and services that tend to build its own social network rather than building upon the rich data available about existing social relationships. Impressive statistics on online social networks (OSN) users overlapping are gathered in [20, 14, 4, 12]. The most recent of them [14] indicates that, for instance, 91% of Twitter users are also users of Facebook, while only 20% of Facebook users have an account at Twitter. The difference in percentages is caused mainly by the difference in total number of users.

From the marketing perspective, today 84% of online customers belong to at least one OSN [20]. As one may expect, many of them split their social activities into several different networks. Therefore, a modern web marketer must have a tool to sift myriads of network accounts from different OSNs in search of valuable customer with all his virtual identities. One of frequently emerging use-cases about such a tool deals with ***targeted marketing via promotional messages***. Once a target user is detected, the marketer should try not to bother him with multiple messages with same content.

Merged profiles of a single user would help to build a more comprehensive view of all available data. The result is more ***complete social graph*** that might be valuable for scientists and entrepreneurs in following areas: collaborative filtering [11], information retrieval [7], sentiment analysis [21], and many other fields.

Another important application area is ***automatic contacts merging*** that takes place mostly in mobile devices. Often advanced users provide full access to several OSN and email accounts in order to somehow integrate the data streams from friends. Several solutions are known (*Gist*[2],

[1]http://code.google.com/apis/opensocial/
[2]http://gist.com/

8hands[3], *AddressBookSync*[4], *Google Sync*[5], etc), but their functionality relies mostly on too simple attribute-based heuristics that fail in most cases when profile fields are not identical or even just written in different languages.

Challenges: A major research challenge, thus, is to match user's information across different OSNs. Modern publications refer to this problem as *identity resolution* which is an instance of more general problem called *entity resolution*, or *entity matching*, or *record linkage*, etc. All these areas have been intensively studied during the past decades with an eye to enhance matching of different objects, from price lists to citation networks and criminal databases. For our purposes, these approaches should be applied to profiles of OSN users. This imposes additional demands to the matching algorithm: it should be able to process raw data from sources with different schemas and make decision based not only on profile data (possibly poor and incomplete) but also on numerous links.

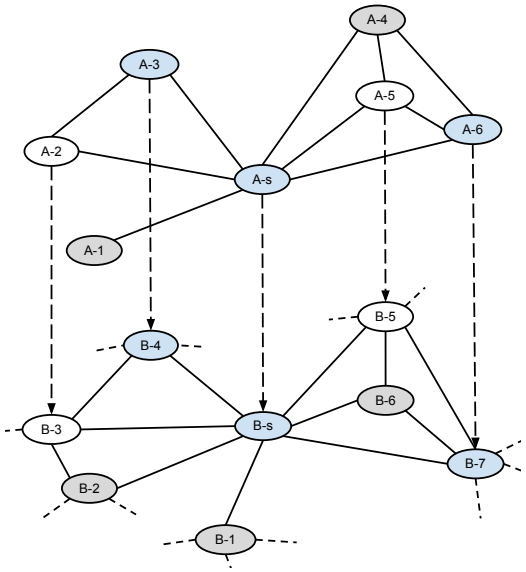
Problem: We define the problem of *user identity resolution (UIR)* as discovering as many as possible correctly matched profile pairs $(v, u), v \in A, u \in B$. $< A, B >$ are social graphs being compared. By social graph we mean a network that represents certain part of online social service where nodes are user profiles with their attributes (such as full name, date of birth, etc) and edges are social links between them. Such links can be directed or undirected depending on relationships they represent. For profile $v \in A$ we will denote the matched profile as $\mu(v)$ and call it *projection* of $v \in A$ to graph B. Another synonym of projection is *match*. For brevity, we will use these two terms for denoting a pair of matched profiles unless any other context is given. If there's no correct match for profile $v \in A$, then we assign it *neutral* projection: $\mu(v) = \mathbf{N}$. An example of such projection configuration is shown in Figure 1.

This research work has made a number of significant **contributions**, as summarized below:

- We introduce novel *Joint Link-Attribute (JLA)* approach to discovering multiple profiles of a single user across different OSNs that combines information from profile attributes with structures of the networks.
- We formulate *user identity resolution* problem in terms of inference in Conditional Random Fields model constructed on a social graph. We also improve the results by filtering out unwanted projections based on a numerous network features.
- We validate our proposed ideas and evaluate JLA algorithm through a comprehensive experimental study[6], using a real dataset collected from Facebook and Twitter. The experimental results show that JLA method outperforms common approaches and produces high-quality results.
- We share the anonymized dataset consisting of 16 ego-network samples from Facebook and Twitter with manually mapped right projections. The dataset could be utilized for evaluating different UIR approaches.
- We demonstrate the importance of social links for identity resolution task and show that certain profiles can be matched based only on social relationships between OSN users.
- We show that the proposed algorithm could also be successfully utilized as a *de-anonymization* tool that aims at matching user profiles when no attribute data is available for one of the networks.

[3]http://8hands.en.softonic.com/
[4]http://danauclair.com/addressbooksync/
[5]http://www.google.com/mobile/sync/
[6]Demo: http://modis.ispras.ru/uir/

Figure 1: An example of UIR results. Graph A is projected to graph B by dashed lines. Blue node pairs are anchored, gray nodes are not projected, white node pairs are projections discovered by the algorithm



The rest of the paper is organized as follows. Section 2 provides a brief introduction in what have already been done in the field of user identity resolution in OSNs. Section 3 provides the algorithm's description and implementation details. Section 4 describes experimental setup and results with discussion. We conclude in Section 5 with possible directions for future work.

1.1 Global and Local Perspectives

The described problem could be approached from both *local* and *global* perspectives. The latter means arbitrary graph merging, in particular large samples or even complete social graphs. Despite the attractiveness of such large scale analysis, this approach has several significant drawbacks that could bother an OSN researcher. At first, it may be hard to obtain large samples from online social service. Social services such as last.fm and Twitter may provide legal access to public user data via API, but limit a number of requests. Facebook which is currently the most popular OSN restricts web-page scraping without explicit permission by terms of service[7]. Access to Facebook API also requires direct permission of each particular user[8]. Automatic crawling an OSN may also affect users' privacy, hovewer, legal issues of user identity resolution are out of scope of the paper.

The second issue is about preparing large test collections. While it's easy to manually discover and mark up a number of profile matches and somehow measure *precision* of UIR al-

[7]`https://www.facebook.com/legal/terms`
[8]`https://developers.facebook.com/docs/authentication/`

gorithm, reliable estimation of *recall* (that is, how many correct profile matches were found) requires virtually all those matches to be presented in the reference set. Since it could not be done without effort of profile owners, vast organizational work is required to compose tests for global-oriented algorithms. Thus, most related studies on user identity resolution (section 2.1) are lacking direct testing on completely marked-up data.

In this paper, we only consider *local* identity resolution, that is, discovering matching profile pairs across the contacts of the given *seed* user. In other words, our approach requires A and B to be *ego-networks* of the seed user. In a sense, this follows several practical use-cases focused on seed's ego-networks such as contact merging in mobile devices. We assume that seed user's permission implies access not only to seed data, but also to profiles of her friends and connections between them (first neighbourhood). This assumption holds for Facebook API (with appropriate permission set), so it should be right in other environments too. Such a restriction helps to comply with privacy of users and makes quality testing realistic.

2. RELATED WORK

We consider three directions relevant to our research. They differ in data sources and underlying models which results in different application areas. *User identity resolution* deals with user data obtained from different sources and aims at agglutinating profiles that belong to the same real person. *Entity resolution* does a similar job but for virtually any kind of data with the same structure, that is, its main application is to find *duplicated* records/profiles/etc in homogeneous data sources. Finally, *de-anonymization* takes anonymized user data (e.g., social graphs) as an input and tries to *re-identify* users in an anonymous graph. Moreover, this technique is not required to produce exact mappings. That is, for a given user, a *set of possible mappings* is often considered an acceptable result.

2.1 User identity resolution

To date, the largest effort in this field seems to be the master's thesis of Veldman [22]. She introduces a lot of heuristics that exploit both raw account data and existing linkage among profiles. Similar studies are presented in [13, 6, 16, 23]. Motoyama et al [13] attempt to match Facebook profiles against MySpace ones. In the study of Gae-won et al [6], the same is done for Twitter and EntityCube accounts. Raad et al [16] generate random social network profiles and then apply much sophisticated heuristics to them; their goal is not to miss any possibly useful piece of information within the network. In the work of Vozecky et al [23], user profiles from Facebook and StudiVZ are represented as n-dimensional vectors. Then, the vectors are compared by means of exact matching, partial matching and fuzzy matching. The authors also investigate the importance of different profile files for matching.

Also, interesting are Foaf-o-matic[9] and OKKAM[10] projects that aim at social profiles integration by means of formal FOAF (Friend-of-a-friend) semantics. Their features are described in [1] and [2].

Despite the progress made by the authors of aforementioned studies, they all utilize too simple profile comparison model based mainly on pairwise comparison using string similarity of attributes. The most common approach is to independently choose for each profile $v \in A$ the most similar one $u \in B$ by applying fuzzy comparison methods (mostly string matching techniques) to attribute pairs, computing total similarity score, and cutting off with certain threshold.

The key of improving existing attribute-based UIR methods is involving additional data sources, in particular social linkage data. The most widely used yet straightforward way for incorporating such information is assuming some profiles mapped successfully (i.e. by high attribute similarity) and then taking into account distance/similarity functions in *partially mapped* contact lists, just like it is done in [22]. Clearly, this heuristic could lead to a bias although being useful in some cases. Our JLA-model only compares original contact lists and is thus free from this drawback.

2.2 Entity resolution and Record de-duplication

Kopcke et al [8] share comparison results of different *entity resolution frameworks*. Among other things, they describe basic requirements, design patterns, and accuracy evaluation standards for such frameworks. One of the most recent efforts is OYSTER[11] project which is highly-configurable framework that allows for entity/identity resolution, management, and capture. Stanford Entity Resolution Framework[12] is also highly relevant to our needs; the authors have put lots of efforts to various theoretical aspects of entity resolution.

The work of Singla et al [18] is an example of applying graphical models to entity resolution task. The authors used Markov logic to find matches between database entries and real-world entities. They construct a Markov Logic Network, where nodes are atomic statements with 0 weight (potential) if they are known to be unsatisfiable and 1 if they are known to be true. Connections between these nodes represent their logical relationship. Such network could be processed in order to estimate most probable configuration of statement weights.

The most similar to the present work is a method proposed by Singla et al in the earlier study [19]. The authors successfully used Conditional Random Fields for record deduplication in the citation network for scientific papers. The main idea is to formulate the global problem of identity resolution as a set of interconnected local problems in terms of probabilistic logic. The authors build CRF graph where *record nodes* represent questions such as "are these two records the same?" while *attribute nodes* store similarity values of object attributes. And the result of inference consists of "yes/no" answers for each belonging question. This approach has following disadvantages that hinder its use for identity resolution of OSN users:

- The granularity of CRF graph makes this approach diffucult to scale for big data.
- Only string similarity metrics are used for object comparison whereas graph similarity metrics are not employed.

The proposed JLA model is strictly intended for social graphs and thus uses more natural representation of the graphical model built on top on of the graphs being com-

[9] http://www.foaf-o-matic.org/
[10] http://www.okkam.org/
[11] http://sourceforge.net/projects/oysterer/
[12] http://infolab.stanford.edu/serf/

pared. It also uses both string and network similarity which is important in a social network processing tasks.

2.3 De-anonymization

Since our algorithm claims to identify users even with anonymized setting when only social graph information is available, it is reasonable to compare our approach to the recent results in the de-anonymization field [24].

The authors extensively use network information in order to get all profile pairs that belong to the same persons from two given social graphs. Their approach shares some features with proposed JLA model. It includes finding *seed* users, using them as *anchor nodes* (already mapped), recursively propagating information through one of the graphs, and extending the set of anchor nodes. Conceptually, this approach is similar to the proposed JLA model in the idea of using social links for propagating information and measuring network distances in order to find best matching profile pairs. With that, there are a set of significant differences that are mainly caused by the fact that we only consider local perspective:

- Graphs are processed by greedy recursive algorithm while we perform inference by global structural optimization which provides better results although is much more computationally complex.
- Initial mappings are found by searching both graphs for k-cliques with matching properties while our algorithm accepts arbitrary set of anchor nodes if any.
- Algorithm tries to match unmapped nodes from *different* graphs by comparing *mapped* neighbours of each node. In JLA-model, we compare *unmapped* neighbours of nodes from *single* graph.
- Finally, we test our method with manually mapped data while the authors of [24] compare their results to profile pairs with exact name matches. Therefore, we measure the quality with more precision and reliability.

3. JOINT LINK-ATTRIBUTE MODEL

JLA model is based on the next basic considerations:

1. Choosing projections for adjacent nodes in graph A are interdependent problems, or in other words choosing projection for one node depends on such decisions for all adjacent nodes.
2. If two nodes in graph A are connected, then they should have lowest possible value of network distance.

Let's follow the explanatory example shown in Figure 2. To find a projection for node v, we consider the next criteria:

- How similar node v is to its possible projection based on similarity of important profile fields?
- How many contacts a possible projection shares with projections of neighbours of node v? If we already know projections for nodes a, b, and c (*anchor nodes*), then intuitively a projection for v should share a lot of contacts with $\mu(a)$, $\mu(b)$, and $\mu(c)$. In other words, $\mu(v)$ should minimize network distance between itself and projections of nodes surrounding v. This captures the fact that social networks tend to have high clustering coefficient.

The same holds for node u and any other unprojected node. Even if there are no anchor nodes around, sometimes it is still possible to find a projection, because the information *propagates* from anchor nodes to all others through the network.

Thus, JLA model involves both attribute information and structure of graphs A and B. One of them is utilized for network distance computation while the second one is used as dependency graph in which a link between two nodes requires minimization of network distance for their projections.

3.1 Probabilistic model

For solving a problem of finding optimal configuration of profile projections we use probabilistic framework called *Conditional Random Fields (CRF)* [9].

In this paper we consider so called associative pairwise CRFs built on top of graph $A = (V, E)$. This impose a restriction on graph A by requiring it to be undirected. Observed variables are modeled by nodes (profiles) of graph A: $\mathbf{X} = \{\mathbf{x}_v = v \mid v \in V\}$. Hidden variables are correct projections of the nodes: $\mathbf{Y} = \{\mathbf{y}_v = \mu(v) \mid v \in V\}$. They take values from a node set of the graph B: $\mathbf{y}_v \in B$. Each hidden variable $\mathbf{y}_v$ is connected to $\mathbf{x}_v$ by factor Φ. Edge modeled by factor Ψ between hidden variables $\mathbf{y}_v$ and $\mathbf{y}_u$ exists if and only if $(v, u) \in E$.

The posterior probability of projections configuration is:

$$p(\mathbf{Y}|\mathbf{X}) \propto \exp(-E(\mathbf{Y}|\mathbf{X})), \tag{1}$$

$$E(\mathbf{Y}|\mathbf{X}) = \sum_{v \in V} \Phi(\mathbf{y}_v|\mathbf{x}_v) + \sum_{(v,u) \in E} \Psi(\mathbf{y}_v, \mathbf{y}_u),$$

where E is *energy functional* consisting of *unary energy* function Φ and *binary energy* function Ψ. More detailed explanation of energy functions is given in section 3.2.

The joint nature of the model is expressed by associating profile distance (opposite of similarity) with unary energy Φ and network distance with binary energy Ψ.

Thus, the model is highly adaptable to the available input data (Figure 3). If there is no graph data, then optimization of the energy functional (1) degrades to greedy profile matching model as it is implemented in most UIR systems (see section 2.1) and $\Psi \equiv 0$. On the other hand, in anonymized setting with no profile attribute information, $\Phi \equiv 0$ and only network information is used.

It should be noticed that information required for our model is not symmetric for graphs $< A, B >$. Thus, only seed user profile, profiles of its friends, and inner connections between them are taken from graph A, whereas complete

Figure 2: Information propagation from anchor nodes (filled blue). Projections for nodes inside each dashed area are estimated independently

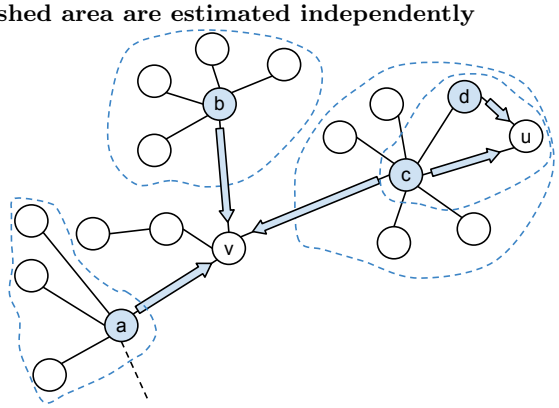
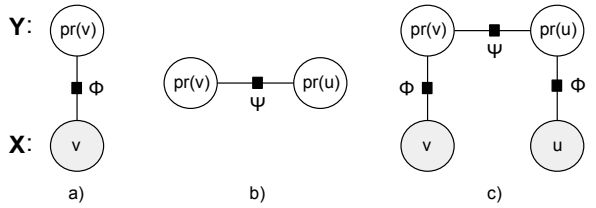


Figure 3: Factor structure of the different models: a) attribute-based UIR b) anonymized JLA c) JLA



friend lists are needed for network distance computation on graph B. UIR process could be initiated by an online social service in order to enrich user contacts by connecting them with profiles from another social network. In such case, friend lists of seed contacts should be available and so this "internal" network may act as graph B while "external" ego-network could be obtained by seed user permission and intended for graph A. Apparently, this is the minimum information needed for incorporating network features into the UIR procedure.

The optimal, or *most a posteriori probable* (MAP) configuration of hidden variables $\mathbf{Y}$ with given values of observed variables $\mathbf{X}$ minimizes the energy functional:

$$\mathbf{Y}^* = \underset{Y}{\operatorname{argmin}} E(\mathbf{Y}|\mathbf{X}) \tag{2}$$

Our representation of CRF is very similar to pairwise Markov Random Field, so we adapted the inference method based on quadratic programming relaxations [17] originally proposed for Markov Random Fields.

3.2 Energies Computation

3.2.1 Energy Functions

Unary energy function Φ is for scaled attribute-based distance between profile $v \in A$ and its projection $\text{pr}(v) \in B$:

$$\Phi(\mathbf{y}_v|\mathbf{x}_v) = \alpha(v) \cdot \textsf{profile-distance}(v, \mu(v))$$

We assume that $\textsf{profile-distance}(v, \mu(v))$ ranges from 0 to 1 while minimum value means that v and it's projection belong to the same person according to profile fields.

Since the topology of CRF is not predefined and is being constructed from graph A, then the sum of binary energies between node $v \in A$ and adjacent nodes could lie in $[0, d(v)]$ depending on the degree $d(v)$ of node v, while $\textsf{profile-distance}(p, \mu(v)) \leq 1$. In order to balance between profile and network similarities, we introduce balancing factor $\alpha(v)$. We achieved best results with $\alpha(v) = log(d(v))$.

Binary energy function Ψ also ranges from 0 to 1 and represents network distance between projections of nodes v and u:

$$\Psi(\mathbf{y}_v, \mathbf{y}_u) = \begin{cases} \infty & \text{if } \mathbf{y}_v = \mathbf{y}_u \\ \textsf{network-distance}(\mu(v), \mu(u)) & \text{otherwise} \end{cases}$$

Here $\textsf{network-distance}(\mu(v), \mu(u))$ could be any normalized distance function such as Dice coefficient. We don't allow adjacent nodes to have the same projection because it would likely output minimum energy but lead to output with many same projections.

3.2.2 Unary Energy

Table 1: Schema mapping for attribute comparison of Facebook-Twitter profile pair

Facebook	Twitter	Comparison function
Name	Name	VMN
	Screen name	Screen Name measure
Website	URL	URL measure

For attribute-based profile comparison, the model described by Vozecky et al [23] is used. Each user profile is represented as a vector of attributes $P_v = (f_1, f_2, ..., f_n)$ where f_i is the i-th profile field.

After the vectors are built, the algorithm utilizes matching functions to calculate a similarity score between corresponding vector fields. A similarity vector V is obtained as a result of this operation, such that its k-th element is:

$$V_k^{(P_v, P_u)} = sim_k(f_i^v, f_j^u), \tag{3}$$

where $1 \leq i \leq |P_v|, 1 \leq j \leq |P_u|, P_v \in A, P_u \in B$.

sim_k is a particular field comparison function that takes data in field f_i^v from P_v and f_j^u from P_u and returns a value between 0 and 1. In general, $V_k = 1$ if f_i^v and f_j^u are identical; $V_k = 0$ if there is no similarity between those fields. The function sim_k differs from field to field since each field may have a different format and semantics.

3.2.3 Twitter and Facebook Profiles Comparison

For comparing Facebook and Twitter profiles, we employed the concept of *schema mappings* [10] in order to align their attribute vectors. This concept is used to automate the process of mapping structures from different sources. Taking Facebook and Twitter profiles as data sources, we result in the following mapping (Table 1).

Each pair of attributes is associated with a specific similarity function. *VMN* is a fuzzy matching technique reported to be very accurate for name matching [23]. *Screen Name measure* simply compares Facebook user name with Twitter full user name and Twitter screen user name (ID). It returns 1 if they are equal and 0 otherwise. *URL measure* is introduced for handling different URLs available from profiles; it attempts to find matches in different combinations of URLs and user names.

As a preliminary step, we detect the language of profile fields with help of *language-detection* library[13]. Later on, when two profile fields are compared, we first check if the profiles are in the same language. If they are not, then we transliterate both strings into Roman transcription (*romanization*) by means of *JUnidecode* library[14].

3.2.4 Learning profile distance function

To account for all available features (sometimes missing if so is corresponding attribute) for profile distance/similarity evaluation, we constructed the $\textsf{profile-distance}(v, u)$ function in machine learning fashion. We have trained a classifier which is able to differentiate between correct and incorrect projections. It takes a feature vector (3) as an input and outputs the probability that projection $u = \mu(v)$ is not correct for profile v. Since Φ is bounded in $[0, 1]$ and so is

[13] http://code.google.com/p/language-detection/
[14] http://sourceforge.net/projects/junidecode/

Table 2: Performance of profile distance classifiers

classifier	recall	precision	F_1
Naive Bayes	**0.862**	0.308	0.453
C4.5	0.569	0.86	0.685
C4.5 with MultiBoosting	0.669	**0.879**	**0.76**

corresponding distance function, we assume that:

$$\text{profile-distance}(v, \mu(v)) = \text{P}(\textit{wrong projection}|V^{(P_v,P_u)})$$

where $V^{(P_v,P_u)}$ is similarity vector (3).

Evaluation results for different classifiers with our dataset (section 4.1) using 3-fold cross-validation are shown in Table 2. As one can see, C4.5 [15] decision trees with multi-boosting [25] in average performed better than other classifiers and thus were chosen for further experiments with Laplace smoothing enabled. Certain classifiers such as SVM could not be applied to the problem directly due to missing attributes in the dataset and thus were excluded from the comparison. This experiment also shows that none of classifiers could perfectly "explain" the choice of correct projections using only attribute information.

3.2.5 *Binary Energy*

In our experiments we used several variations of *Dice coefficient* which is a normalized number of vertices directly connected to both nodes. It was chosen mostly due to its computational simplicity and markovness. We calculate the network distance as an opposite of its value:

$$\text{network-distance}(v, u) = 1 - \frac{2 \cdot w(L_v \cap L_u)}{w(L_v) + w(L_u)}, v, u \in B,$$

where L_v and L_u are sets of first neighbours of v and u respectively. $w(L)$ is a weight of node set L.

In case when standard Dice coefficient is used, weighting function is defined as $w(L) = |L|$. We have implemented several alternative weighting functions and empirically estimated the best one: $w(L) = \sum_{v \in L} 1/\log(d(v))$ where $d(v)$ is the degree of node v. We use this weighting function in all further research and experiments because it allows to account the fact that sharing a friend with small number of friends brings greater increase of similarity than sharing a popular friend with many contacts.

3.3 Anchor nodes

Anchor nodes (also referred to as *seed* nodes) are nodes in graph A for which right projections in graph B are known a priori. They could be found by searching for user profiles that are linked to another OSN by their owners. Another way is to find complete or near-complete user name matches between profiles. We employed this heuristic in our experiments by using profile-distance function with thresholds.

For each anchor node v we fix the projection $\mu(v)$ to it's matched profile anchor(v) by assigning $\Phi(\mathbf{y}_v|\mathbf{x}_v) = \infty$ if $\mathbf{y}_v \neq \text{anchor}(v)$.

Anchor nodes are also necessary in anonymized setting. Moreover, they help to reduce the computational time required for inference. If a subgraph of graph A is connected to the rest of nodes only through anchor nodes, then inference is performed *independently* for this component (Figure 2).

3.4 Results Pruning

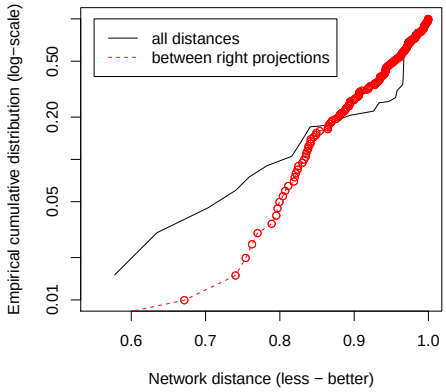


Figure 4: Cumulative distribution of network distance

Our goal was to achieve the highest possible recall while keeping precision close to the maximum (see section 4.3 for details on accuracy evaluation). From our point of view, that is what most of users expect from identity resolution application: they are not willing to spend any time checking the results. Therefore, every answer found by the algorithm should be precise for certain. Here is where real challenge lies: maximize the fraction of discovered projections and try to provide no false positives at all. In this subsection we describe another important contribution which helps to comply with this requirement by removing suspicious projections from the results.

Most identification errors are caused by weak connectivity in the model graph. Indeed, projection that minimizes network distance with the only neighbour node would not be necessary a right match. On the other hand, if the number of surrounding nodes is relatively large, choice of projection more likely tend to be correct.

In the ideal case, we could design the binary energy functions in a way to prevent such situations by assigning a special value to $\Psi(\mathbf{N}, \mathbf{y}_u)$ and $\Psi(\mathbf{y}_v, \mathbf{N})$ with a threshold semantics. This would force the algorithm to choose neutral projections in cases when it minimizes the full energy. However, there is no simple way to implement this using only information about two projections (and possibly the nodes themselves). Figure 4 demonstrates the network distances distribution in our main dataset. As we may see, there is no reasonable value for neutral projections in this distribution since lesser values are better.

Making the model more complicated by extending it from pairwise interactions to more complex structure would affect the computational complexity of inference. Thus, we propose two different solutions for the problem.

3.4.1 *Mutual projections*

Firstly, we project graph A to graph B and also perform reverse projection from B to A (CRF is consturcted from graph B and network distances are computed with graph A). Then, we merge the results in order to keep only mutual projections discovered by both direct and reverse passes.

This technique shows good accuracy comparing to attribute-based baselines. However, it is quite straightforward, takes two times longer, and doesn't take into account the cause of each mistake. To overcome these drawbacks, we elaborated

Table 3: Performance of pruning classifiers

classifier	recall	precision	F_1
Naive Bayes	0.762	0.256	0.383
Support Vector Machine	0.662	0.935	0.775
C4.5	0.715	**0.939**	0.812
C4.5 with MultiBoosting	**0.844**	0.902	**0.872**

supervised machine learning approach.

3.4.2 Pruning classifier

We have trained a classifier that is able to find incorrect projections using properties of surrounding nodes (including anchor nodes) and profile distance. In other words, the classifier determines whether the *context* of the particular node was *qualitatively* enough to infer good projection.

For each match we compute the following set of features:

1. Profile distance of the projection
2. Average network distance to the surrounding projections
3. Fraction of anchor nodes in the first neighbourhood
4. Consistency of surrounding anchor projections:

$$\frac{1}{n} \cdot \sum_{v} \frac{1}{n-1} \sum_{u \neq v} \mathsf{network\text{-}distance}(\mu(v), \mu(u))$$

In case when no anchor nodes are given, we exclude the third feature and assume that all surrounding nodes are anchors when calculating the fourth.

Performance evaluation results for trained classifiers are shown in Table 3. All possible projection configurations for Twitter → Facebook projection built from main dataset were used as training set. C4.5 with multi-boosting performed better than others and thus was chosen for the experiments.

4. EXPERIMENTS

In this section we describe our experiments with Twitter and Facebook sample graphs. All evaluations of supervised machine learning algorithms were performed using 3-fold cross-validation in order to provide confident results.

4.1 Dataset

In order to collect our *main dataset*, we ran snowball sampling of 16 seed users in both Twitter and Facebook. Each sample includes publicly available profile information and contact lists for the seed node and its first order *mutual* contacts. The motivation for such setup is provided in Section 1.1.

It is also important to decide what we consider *connection*. In Facebook, all connections are mutual and have "friendship" semantics. The situation in Twitter is different, connections are directed and represent rather "subscription" relationship. Therefore, we consider only *mutual following* in Twitter in order to simulate "friendship" relationship. Clearly, that could only *understate* the accuracy of results, therefore such modeling seems to be representative enough. Nevertheless, JLA model does not require all connections to be directed. Hence, one may use Facebook (or any other undirected graph) as a model graph and compute energies on Twitter (or any other graph with or without directed edges) by constructing its own network distance function taking into account both follower and followee lists.

Table 4: Dataset statistics

	Twitter	**Facebook**
Main dataset		
# of seeds	16	
# of profiles	398	977
# of connections	1 728	10 256
total # of matches	141	
total # of anchor nodes	71	
Reidentification dataset		
# of seeds	17	
# of profiles	1 499	7 425
# of connections	15 943	172 219
total # of matches / anchor nodes	161	

All samples have been marked up with right projections (ground truth) by the owners of seed profiles. Structure of the samples is shown in Figure 1.

Anonymized ground truth is available from our server[15].

We also prepared *reidentification dataset* in a similar way for other 17 seed profile pairs. This dataset is intended only for experiments in re-identification experiments on our algorithms in section 4.5.

Overall dataset statistics are provided in Table 4.

4.2 Baseline

There are two baseline algorithms in our study:

1. Compute profile distance as a weighted sum of the features described in section 3.2.3. The weights are estimated by linear regression assuming that weighted sum should be 0 for right projections.
2. Compute profile distance using classification probability (defined in section 3.2.4). We used this approach for anchor nodes extraction when running JLA algorithm.

They both rely on all-to-all pairwise comparison between Facebook and Twitter profiles. Given a set of profile pairs with measured attribute-based distance, best projections are estimated upon the condition that each profile from graph A can be matched with at most one profile from graph B and vice-versa (*assignment* problem). We have also estimated reasonable thresholds for all baselines in order to maximize recall while keeping precision high as it would be required in a real system.

4.3 Accuracy Evaluation

We use following versions of conventional *precision* and *recall* metrics:

$$\text{recall} = \text{tp}/(\text{tp} + \text{fn}), \text{precision} = \text{tp}/(\text{tp} + \text{fp})$$

Here we assume correct profile pair match as true-positive (tp), incorrect as false-positive (fp), and non-discovered as false-negative (fn). For testing we use lists of correct projections obtained from the owners of seed profiles. Table 5 contains summarized accuracy evaluation results.

Both baselines showed high precision, but could not discover many matches using only attribute information which led to relatively low recall. These results prove the necessivity of involving link data into UIR and emphasize the

[15]http://modis.ispras.ru/uir/

Table 5: Accuracy evaluation results

algorithm	R	P	F_1
agnostic to direction of projection			
Baseline 1 (weighted sum)	0.45	0.94	0.61
Baseline 2 (probability distance)	0.51	**1.0**	0.69
JLA, intersection, anonymized	0.6	**1.0**	0.76
JLA, intersection	0.66	0.99	0.79
Twitter → Facebook			
JLA, anonymized ($\Phi \equiv 0$)	0.62	**1.0**	0.77
JLA	0.79	**1.0**	0.89
Facebook → Twitter			
JLA, anonymized ($\Phi \equiv 0$)	0.61	**1.0**	0.76
JLA	**0.8**	**1.0**	**0.89**

limitations of attribute-based UIR systems in environments with poor profile data, such as Twitter.

All experiments with JLA were performed in normal and anonymized setting. Anonymized setting means that no profile attribute information is available.

JLA with mutual projections pruning (section 3.4.1) performed worse comparing to classifier pruning (section 3.4.2), because the former technique appeared too aggressive and lowered the recall. As a result, mutual projections pruning was not able to significantly outperform the baselines. From now we will consider only JLA with classifier pruning.

JLA with different projection directions (Twitter → Facebook and Facebook → Twitter) showed similar results. When computing network distances with Twitter graph, we assumed *contact* list as a list of mutual followers and got the best results. JLA could discover about 80% of correct matches with 28% advantage comparing to the baseline. Since it is a supervised machine learning algorithm, we performed all experiments with 3-fold cross-validation.

4.4 Impact of Anchor Nodes

Another experiment has been done in anonymized setting, but with access to anchor nodes' projections. It was intriguing to investigate the dependency of JLA accuracy on the amount of given anchor nodes.

Figure 5 illustrates how the average performance (mostly recall) changes with the fraction of anchor nodes at the input. The margin between anonymized JLA and second baseline is 10% in recall (since baseline 2 is used to get the anchor projections). That is, JLA algorithm could find about 10% more matches with knowledge about half of anchor nodes. It demonstrates that although being intended for UIR, JLA model could be useful as a de-anonymization tool whose accuracy depends on the amount of provided anchor nodes.

In normal setting with profile information available (Figure 6), the drop of recall is not so clear because some matches are still found based on low unary energy. Due to good connectivity of Facebook graphs in average (see Table 4), the information could propagate from projections with low unary energy and facilitate discovering other matches. This explains the nearly horizontal recall graph for Facebook → Twitter projection in the [0;0.5] interval of ANF.

4.5 Re-identification

As we have discussed already, it is hard to obtain reliable dataset for UIR without involving seed profile owners. Thus, we decided to test our algorithm automatically on

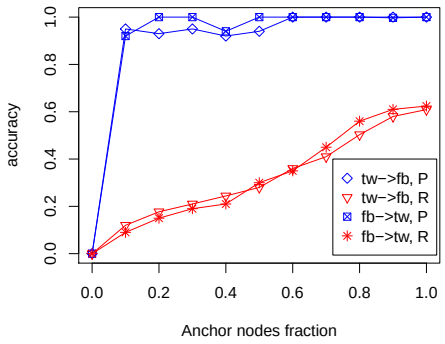


Figure 5: Impact of anchor nodes fraction. Profile information is missing

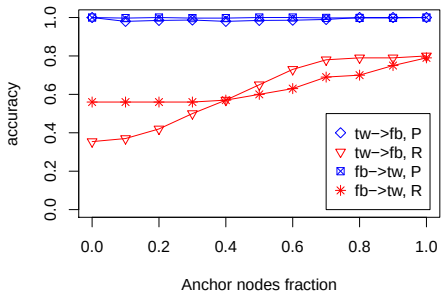


Figure 6: Impact of anchor nodes fraction. Profile information is available

our *reidentification dataset* (section 4.1). At first, we found some matches using profile information only. After that, we investigated the performance of JLA according to only those easy-found anchor nodes by removing some of them and measuring success of the algorithm. We ignored projections for all other nodes. To make the experiment honest, we removed all profile information for the anchor nodes and their possible projections. Doing so, we ensured that they could be matched only by network distances to surrounding nodes. For other nodes profile information was available. We reused our main dataset for training all supervised learning algorithms.

Results are shown in Figure 7. As one can see, with no prior information about anchor nodes' projections, the precision is not that high, but it quickly grows with anchor nodes fraction (ANF). Starting from 80% ANF, all of them could be identified correctly. This experiment shows once again that complete knowledge about anchor nodes isn't required

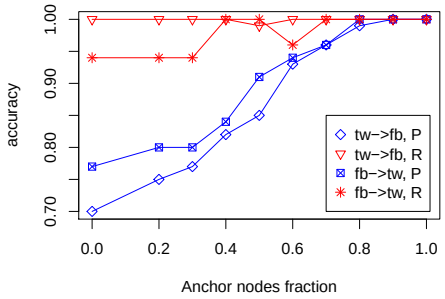


Figure 7: Impact of anchor nodes on re-identification. Profile information is available

for their successful identification because enough information is contained in the social links.

One of the important results is that we confirmed the intuitive idea: the greater is the node degree, the more likely it would be identified correctly. The converse is also true - the greater are degrees of the anchor nodes, the better results will be achieved.

5. CONCLUSION & FUTURE WORK

In this paper, we have presented new Joint Link-Attribute model for user identity resolution and evaluated it with real data from Twitter and Facebook. We showed its efficiency as UIR or de-anonymization tool. We proved the significance of social links for the problem and empirically measured the impact of several kinds of available information. We demonstrated that building the model on the more connected graph (Facebook in our case) is preferred in most cases (especially when some information is hidden or missing). Finally, we demonstrated the interesting and natural application of graphical models in social network analysis.

Despite the algorithm success in a local perspective, a major challenge would be scaling it up to large social graphs. We believe that it is possible to make the inference feasible by decomposing a big problem into a number of small ones with help of anchor nodes and by reusing the sparseness of the data.

Although we experimented only with first neighbourhood samples, it is an open question how robust our algorithm is to different sampling techniques and link data corruption. Dealing with these problems is another interesting direction of our future work.

6. ACKNOWLEDGEMENTS

We are grateful to our colleagues and friends for suggestions and reviewing and also to all people who helped us to collect and mark up the test data.

7. REFERENCES

[1] S. Bortoli, H. Stoermer, P. Bouquet (2007). *Foaf-O-Matic - Solving the Identity Problem in the FOAF Network*. In: Proceedings of the Fourth Italian Semantic Web Workshop (SWAP2007), Bari, Italy, Dec.18-20, 2007.

[2] P. Bouquet, S. Bortoli (2010). *Entity-centric Social Profile Integration*. In: Proceedings of the International Workshop on Linking of User Profiles and Applications in the Social Semantic Web (LUPAS 2010) 52-57.

[3] R. Burkard, M. Dell'Amico, S. Martello (2009). *Assignment Problems*. SIAM.

[4] *Connecting the Social Graph: Member Overlap at OpenSocial and Facebook*. `http://blog.compete.com/2007/11/12/connecting-the-social-graph-member-overlap-\at-opensocial-and-facebook/`

[5] F. Fouss, A. Pirotte, J.-M. Renders, M. Saerens. *Random-walk computation of similarities between nodes of a graph, with application to collaborative recommendation*. IEEE Transactions on Knowledge and Data Engineering, vol. 19, No. 3, March 2007.

[6] Gae-won Y., Seung-won H., Zaiqing N., Ji-Rong W. *SocialSearch: Enhancing Entity Search with Social Network Matching*. EDBT 2011.

[7] G. Kazai, N. Milic-Frayling. *Trust, authority and popularity in social information retrieval*. In CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge mining. 2008

[8] H. Kopcke, E. Rahm. *Frameworks for entity matching: A comparison*. Data & Knowledge Engineering, Vol. 69, No. 2. (2010), pp. 197-210.

[9] J. D. Lafferty, A. McCallum, P. McCallum, C. N. Fernando. *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data*. Proceedings of the Eighteenth International Conference on Machine Learning, 2001.

[10] M. Lenzerini. *Data Integration: a Theoretical Perspective*. In PODS '02: Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems, pages 233-246. 2002.

[11] Soo Ling Lim. *Social Networks and Collaborative Filtering for Large-Scale Requirements Elicitation*. School of Computer Science and Engineering. 2010

[12] R. MacManus. *OpenSocial and Facebook Stats from Rapleaf*. `http://www.readwriteweb.com/archives/opensocial_and_facebook_statistics.php`

[13] Motoyama, M., Varghese, G. *I Seek You - Searching and Matching Individuals In Social Networks*. WIDM '09: Proceeding of the eleventh international workshop on Web information and data management.

[14] S. Perez. *Who Uses Social Networks and What Are They Like? (Part 1)*. `http://www.readwriteweb.com/archives/who_uses_social_networks_and_what_are_they_like_part_1.php`

[15] R. Quinlan (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.

[16] Raad, E., Chbeir, R., Dipanda, A. *User Profile Matching in Social Networks*. 13th International Conference on Network-Based Information Systems (NBiS), 2010.

[17] P. Ravikumar, J. D. Lafferty. *Quadratic Programming Relaxations for Metric Labeling and Markov Random Field MAP Estimation*. Proceedings of the 23rd international Conference on Machine Learning. ICML '06, vol. 148, pp. 737-744.

[18] P. Singla, P. Domingos. *Entity Resolution with Markov Logic*. In Proc. of the Sixth International Conference on Data Mining (ICDM'06).

[19] P. Singla, P. Domingos. *Multi-relational Record Linkage*. KDD Workshop on Multi-Relational Data Mining (pp. 31-48), 2004.

[20] *Social Networking: MySpace, YourSpace and TheirSpace, Connecting With Your Customers in Online Social Networks*. American Marketing Association, 2008(?).

[21] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou, P. Li. *User-Level Sentiment Analysis Incorporating Social Networks*. SIGKDD 2011.

[22] Veldman, I. (2009) *Matching Profiles from Social Network Sites*. Master's thesis, University of Twente.

[23] Vosecky, J., Dan Hong, Shen, V.Y. *User identification across multiple social networks*. In Proc. of First International Conference on Networked Digital Technologies, 2009.

[24] Arvind Narayanan, Vitaly Shmatikov. *De-anonymizing Social Networks*. IEEE Symposium on Security & Privacy, 2009.

[25] G. I. Webb (2000). *MultiBoosting: A Technique for Combining Boosting and Wagging*. Machine Learning, 40(2): 159-196.