PDGM: Percolation-based Directed Graph Matching in Social Networks

Lijing Wang*‡, Jin-Hee Cho†, Ing-Ray Chen‡, Jiangzhuo Chen*
*Biocomplexity Institute
Virginia Tech, Blacksburg, VA
{lijingw9, chenj}@vbi.vt.edu
†US Army Research Laboratory
Adelphi, MD
jinhee.cho@us.army.mil
‡Department of Computer Science
Virginia Tech, Falls Church, VA
irchen@vt.edu

***Abstract*—Linking multiple accounts owned by the same user across different online social networks (OSNs) is an important issue in social networks, known as *identity reconciliation*. Graph matching is one of popular techniques to solve this problem by identifying a map that matches a set of vertices across different OSNs. Among them, percolation-based graph matching (PGM) has been explored to identify entities belonging to a same user across two different networks based on a set of initial pre-matched seed nodes and graph structural information. However, existing PGM algorithms have been applied in only undirected networks while many OSNs are represented by directional relationships (e.g., followers or followees in Twitter or Facebook). For PGM to be applicable in real world OSNs represented by directed networks with a small set of overlapping vertices, we propose a percolation-based directed graph matching algorithm, namely PDGM, by considering the following two key features: (1) similarity of two nodes based on directional relationships (i.e., outgoing edges vs. incoming edges); and (2) celebrity penalty such as penalty given for nodes with a high in-degree. Through the extensive simulation experiments, our results show that the proposed PDGM outperforms the baseline PGM counterpart that does not consider either directional relationships or celebrity penalty.**

I. Introduction

Online social networks (OSNs), such as Facebook, Twitter, Instagram, or LinkedIn, have become critical tools for people to communicate and maintain their social relationships. As OSNs become more and more popular than ever, people often have multiple accounts across different OSNs simultaneously. Identifying a single entity associated with multiple OSNs has been well motivated in various domains.

Social scientists have studied people's social behaviors based on information obtained from various OSN applications. However, they realized that analyzing a single OSN dataset would not provide a comprehensive view to understand human behaviors [9]. Further, identifying the same user across multiple networks has been needed for personalized advertisement, link recommendation, friend suggestion, or community analysis using information from multiple networks.

Graph matching has been studied as one of the popular methods to identify the same user across different networks. For example, a social network can be represented as a graph where users are nodes and a social tie (e.g., follower, followee, or friends) between two nodes is an edge. The graph matching technique reconciles the same user across different networks only based on graph structural features, such as the users' social behavior patterns (e.g., a common set of friends). Graph matching methods have been applied in various domains including image processing [2], matching gene sequences in gene/protein networks [17], alignment of protein interaction networks [7] and matching an image's segment graph [3].

Percolation theory has been applied in graph matching algorithms [5]. Percolation theory studies the presence of large clusters in random environments, such as lattices with missing nodes or links, or random graphs. In particular, a node is part of a cluster only if it has at least r neighbors that belong to the cluster. This is called *bootstrap percolation* [5]. The bootstrap percolation-based graph matching (PGM) method assumes that when a set of "pre-matched" initial seed node pairs are given a priori, additional matching pairs can be identified incrementally. This process is called the *percolation process*, starting from the initial seed pairs to the other node pairs identified based on the existing matched pairs. Thus, the set size of the initial seed nodes is a critical parameter to determine the size of percolation, representing the ratio of matched pairs across multiple graphs [17].

PGM has been studied by many researchers with interesting findings. Successful pair matching across two different networks is found in large-scale anonymized social networks [11], random graphs and scale-free graphs (e.g., preferential attachment graphs) [9]. In addition, a sharp phase transition is identified in the size of the final map based on the size of an initial seed set [17], and a dramatic reduction in the required size of the seed set is achieved only with a small increase in matching errors [6]. However, all graphs considered in the above [6, 9, 11, 17] are undirected networks. In addition, they use synthetic graph datasets, which may not ensure the performance of the proposed algorithms in real networks that often have less structural similarity between two different

networks. Besides, no algorithmic complexity is analyzed in the existing approaches [6, 9, 11, 17].

In this work, we propose a variant of PGM that is applicable to directed networks, and validate it through extensive experiments using real network datasets (i.e., Foursquare-Twitter datasets [18]). We call our proposed percolation-based directed graph matching algorithm "PDGM."

This paper has the following unique contributions:

- PDGM is a variant of PGM algorithms (two variants based on the concept of similarity) for directed social networks based on directional relationships;
- PDGM mitigates the celebrity effect by removing the effect of high in-degree nodes. This technique significantly enhances matching precision and reduces percolation delay per matched pair identified in the percolation process;
- PDGM is validated using real social network datasets with a small set of overlapping of nodes compared with synthetic datasets. Through our simulation experiments, we show that PDGM outperforms the baseline counterpart in terms of matching precision and percolation delay.

The rest of this paper is organized as follows. Section II discusses existing approaches in terms of the types of features considered for matching conditions across multiple networks. Section III describes the network model. Section IV provides the details of PDGM. Section V describes performance metrics and experimental setup, and discusses the overall trends of the observed experimental results. Section VI concludes the paper and suggests future research directions.

II. Related Work

We discuss the existing approaches to re-identify users across multiple networks based on the features, including ad-hoc identification, non-identification, graph structure, and hybrid.

A. Ad-hoc Identification Features

Ad-hoc identification features (AIF) include the username, gender, email address, home location, and/or unique tags to match user profiles or de-anonymize users across different social networks [1, 4, 10, 15]. In particular, Buccafurri et al. [1] use the similarities of usernames to detect anchor links considering a scaling coefficient to mitigate the effect of public figures. This work is similar to ours in that our PDGM also considers the celebrity effect by removing nodes with high in-degrees to distinguish an individual user's unique social interaction patterns. The main drawback of using AIF is the lack of datasets available due to the privacy settings of users.

B. Non-Identification Features

Non-identification features (NIF) consider users' mobility or similarity of behavioral patterns. They include temporal and/or spatial distributions of user accounts in which the geo-location is available by some OSNs, called location-based social networks (e.g., Twitter, Foursquares, Facebook), to trace user mobility patterns [8, 12, 13].

Kong et al. [8] show that only a few geo-locations of a user are sufficient to match the same user accounts across multiple networks. Riederer et al. [12] leverage the location-based information to model user mobility records. Rossi and Musolesi [13] utilize both spatial and temporal trajectories emerging from users' check-in time and the frequency of visits to specific locations, respectively. They prove that the same user is identified across multiple networks only based on a very small number of data points [12, 13]. However, location-based information is rarely available in practice.

C. Graph Structural Features

Graph structural features (GSF), including users' social behavior patterns in a network (e.g., the number of common friends), are used in graph matching methods. Given a set of initial pre-matched seed nodes, additional node pairs are identified as matching nodes [9, 11, 14, 16, 17], leveraging the percolation process in PGM.

Narayanan and Shmatikov [11] de-anonymize large scale social networks based on the number of common neighbors. Korula and Lattanzi [9] propose an efficient parallel algorithm based on local information in random or scale-free synthetic networks, given a subset of nodes as initially pre-matched nodes. However, their work does not analyze a phase transition in terms of the number of initial seed nodes. Kazemi et al. [6] propose a PGM algorithm to reduce the size of the seed set required for the starting of phase transition.

The work cited above [6, 9, 11] use synthetic datasets which may not reflect the performance in real networks whose overlapping nodes across multiple networks are significantly smaller than the synthetic networks.

Our work is similar to the PGM algorithm [17] that studies the phase transition over varying the initial seeding nodes in terms of matching performance. However, unlike the PGM algorithm [17], which applies to undirected networks only, our PDGM can be applied to directed networks by considering directional relationships and celebrity penalty.

D. Hybrid Features

Some existing approaches combine the AIF, NIF and GSF features together. Kong et al. [8] extract heterogeneous features from multiple networks for anchor link prediction, including user's social, spatial, temporal and text information. Srivatsa and Hicks [14] use both location-based and graph structural information.

III. Network Model

We consider a social network represented by a graph $G(V, E)$ where V is a set of nodes and E is a set of edges. Two observable graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ are obtained from $G(V, E)$, where $V_1, V_2 \subseteq V$ and $E_1, E_2 \subseteq E$. $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ partially overlap where the fraction of intersected nodes is denoted by $\alpha_V = \frac{V_1 \cap V_2}{V_1 \cup V_2}$ and the fraction of intersected edges is represented by $\alpha_E = \frac{E_1 \cap E_2}{E_1 \cup E_2}$. A known set of the anchor links, undirected edges linking two

nodes representing the same user in G_1 and G_2, is $L \subset V_1 \times V_2$ with the size $\ell = |L|$.

Network structural information of G_1 and G_2 is known a priori such that adjacency matrices A_1 and A_2 and weighted adjacency matrices W_1 and W_2 are available. We consider directed networks whose adjacency matrices are not necessarily symmetric. A_1 and A_2 are the adjacency matrices where each element a_{ij} is set to 1 for an edge from i to j; 0 for no edge between them. Edge weight in W_1 or W_2 is defined based on the degree but with the penalty for high in-degree nodes. We assume that a set of initial seed nodes, A_0, is given with the size $a_0 = |A_0|$ and will be randomly selected from the known set of anchor links, L.

TABLE I: Notations and their meaning.

Notation	Meaning
$G(V, E)$	Network with a set of nodes V and a set of edges E
n	Total number of nodes in G
m	Total number of edges in G
G_1, G_2	Two networks where partial vertices and edges overlap
A	Adjacency matrix of a network
W	Weighted adjacency matrix of a network
L	A set of known anchor links between G_1 and G_2 where $\ell = \|L\|$
A_0	A set of initial seed nodes where $a_0 = \|A_0\|$
A^*	A final mapping set where $a^* = \|A^*\|$
T	A final time step
$c^{out}_{i'j'}$, $c^{in}_{i'j'}$	Similarity credit based on either outgoing edges or incoming edges for a pair (i', j'), respectively
$c^{pgm}_{i'j'}$	Similarity credit of the PGM scheme for a pair (i', j')
S_{ij}	Similarity score of a pair (i, j)
$M(t)$	The set of matched pairs at time step t
$U(t)$	The set of matched pairs that have been used until time step t
$N(i)$	The set of neighbors of node i
K_{ij}	Number of marks of similarity witnesses for a pair (i, j)
r	Threshold of K_{ij}
d^{in}_i, d^{out}_i	In-degree and out-degree of node i
d_{tr}	Degree threshold for removing a node with high in-degree
D_{ij}	A set of candidate matched pairs (i, j)'s where $i \in V_1$ and $j \in V_2$
τ	Average time elapsed to identify a matched pair
τ_0	Time unit to compute a similarity credit
$\langle k \rangle$	Average node degree
s	Probability that an edge in G also appears in its observable graph
t_c	A critical time step to start percolating
a_c	A critical value of a set size of initial seed nodes to percolate

IV. Percolation-based Directed Graph Matching

This section discusses the details of the proposed PDGM in terms of the percolation conditions, similarity functions, and percolation process.

A. Percolation Conditions

The bootstrap percolation [5] is analyzed in the *Erdős-Rényi* (ER) network by proving the phase transitions in terms of the size of the final mapping a^*, given the size of initial seeding nodes, a_0. The phase transition with a_0 is also proven in PGM [17]. We adopt the percolation conditions following the phase transitions in [17] defining a critical value for a_0 to trigger the percolation process that matches additional pairs between two graphs, G_1 and G_2. Given a random graph $G(n, p)$ and s, r, where s is the probability that an edge in G also appears in its observable graph G_1 or G_2, r is the minimum number of matched neighbors for a pair to be considered as matched, the critical value, a_c, is given by:

$$a_c = (1 - \frac{1}{r})t_c, \tag{1}$$

where a_c is the critical size of the initial pre-matched seed set. The critical time, t_c, is given by:

$$t_c = (\frac{(r-1)!}{nq^r})^{\frac{1}{r-1}}, \tag{2}$$

where $q = ps^2$ and $r \leq 2$. This implies that when $a_0 < a_c$, the algorithm will stop before reaching t_c with the final size at most $2 \times a_0$; for $a_0 \geq a_c$, the algorithm will percolate most of both networks with the size of final mapping $a^* = n - o(n)$, which is the lower bound of the final mapping size.

For the algorithm to trigger the percolation that can lead to a high matching accuracy, the following two conditions should be met:

- `C1`: The percolation process should continue at least until t_c and successfully beyond t_c. If no new matched pair is identified due to insufficient evidence, then the percolation stops.
- `C2`: The percolation process should be performed with as few incorrectly matched pairs as possible. The incorrectly matched pairs may potentially trigger percolation to identify further incorrect pairs in the future, leading to cascading errors.

B. Similarity Functions

The similarity functions defined in PDGM follow [9] but reflect the directional relationships to be applied in directed networks. Neighbors of a node are defined as nodes connected to itself with either an incoming edge or an outgoing edge. That is, if there exists an edge from i to j or j to i, j is a neighbor of i. Let $N_1(i)$ or $N_2(j)$ denote a set of neighbors of i in G_1 or j in G_2, respectively. Accordingly, we define the similarity witness pairs and candidate matched pairs below.

Definition 1. Similarity witness pairs: A pair of nodes (i', j') where $i' \in V_1$, $j' \in V_2$ is said to be the similarity witness pair for (i, j) where $i \in V_1$, $j \in V_2$ if $i' \in N_1(i)$, $j' \in N_2(j)$ and i' has been identified as matched with j' (i.e., i' and j' represent the same user).

Definition 2. Candidate matched pairs: If a pair (i, j) has the similarity witness pair, (i', j'), then (i, j) is said to be a candidate matched pair for (i', j'). $D_{i'j'}$ represents the set of all candidate pairs of (i', j'), (i, j)'s.

Definition 3. Similarity credit ($c_{i'j'}$): This refers to the credit for the similarity obtainable from a similarity witness (i', j') for (i, j), which is computed by the similarity between i' and j' based on their degrees. The similarity score S_{ij} for (i, j) is the sum of $c_{i'j'}$'s obtained by all similarity witnesses pairs, (i', j')'s.

PDGM iteratively runs where each iteration is performed at a time step for $t \in \{0, 1, \ldots, T\}$. It stops when no matched pair is found. The stopping condition is $K_{ij} < r$ where K_{ij} represents the number of similarity witness pairs for (i, j). Similarity score, S_{ij}, is computed based on the sum of $c_{i'j'}$ at each time step until (i, j) is selected to be matched or the algorithm stops the percolation process:

$$S_{ij}(t) = \sum_{i' \in N_1(i) \wedge j' \in N_2(j) \wedge (i', j') \in M(t-1)} c_{i'j'}, \quad (3)$$

where $i' \in N_1(i) \wedge j' \in N_2(j) \wedge (i', j') \in M(t-1)$ represents similarity witnesses for (i, j) at time step t, $c_{i'j'}$ represents the credits that (i', j') contributes to the similarity between i and j. Considering the directional relationships in directed networks, we devise two types of similarity credit functions based on weighted adjacency matrices W_1 and W_2, corresponding to G_1 and G_2, respectively.

The celebrity effect in social networks has been studied in graph matching algorithms in that a celebrity or a person with high centrality (or influence) has little impact in identifying his/her neighbors' social behavior pattern in social networks [1]. In order to mitigate the effect of celebrity in measuring W_1 and W_2, the PDGM penalizes the relationship with a neighbor node that has a high in-degree.

Definition 4. Celebrity penalty: Node i's importance to its neighbors is defined based on how much i contributes to identifying its neighbors' unique social patterns in terms of node i's in-degree. The higher node i's in-degree, the lower its importance.

We devise a celebrity penalty factor called *threshold-based penalty*. This celebrity penalty factor penalizes node i's importance by a threshold, d_{tr}, where node i's in-degree, d_i^{in}, is only counted when $d_i^{in} \leq d_{tr}$ with a weight for W by:

$$w_{ij} = \begin{cases} 1 & \text{if } a_{ij} > 0 \wedge d_j^{in} \leq d_{tr} \\ 0 & \text{if } a_{ij} = 0 \end{cases} \quad (4)$$

where w_{ij} corresponds to the weight of the edge from i to j, a_{ij} is an element of the adjacency matrix A. Based on the celebrity penalty factor above, we devise the following similarity credit functions:

- PDGM with outgoing edges from (i, j) (PDGM-OUT):

$$c_{i'j'}^{out} = \begin{cases} 2 & \text{if } w_{ii'} w_{jj'} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

- PDGM with incoming edges to (i, j) (PDGM-IN):

$$c_{i'j'}^{in} = \begin{cases} 2 & \text{if } w_{i'i} w_{j'j} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Algorithm 1 provides the description of PDGM. Note that $simFunc((i', j'), (i, j))$ implements Eqs. (5) and (6) for the two design variants.

Algorithm 1 PDGM

Input:
$G_1(V_1, E_1, W_1)$, $G_2(V_2, E_2, W_2)$
$A_0 \subset L$ where $a_0 = |A_0|$
Output: A^* where $a^* = a_0 + T$
$t \leftarrow 0$: $M(0) = A_0$, $U(0) = \emptyset$, $S_{\forall(ij)}(0) = 0$
repeat
  $t \leftarrow t + 1$
  while $M(t-1) \setminus U(t-1) \neq \emptyset$ **do**
    Select a random pair $(i', j') \in M(t-1) \setminus U(t-1)$
    for all $(i, j) \in D_{i'j'}$ **do**
      $c_{i'j'} \leftarrow simFunc((i', j'), (i, j))$
      $S_{ij}(t-1) \leftarrow S_{ij}(t-1) + c_{i'j'}$
      $K_{ij}(t-1) \leftarrow K_{ij}(t-1) + 1$
    end for
    $S_{ij}(t) \leftarrow S_{ij}(t-1)$
    $K_{ij}(t) \leftarrow K_{ij}(t-1)$
    $U(t-1) \leftarrow U(t-1) + (i', j')$
  end while
  Select a random pair (i, j) at t with $\max[S_{ij}(t)'s] \wedge K_{ij}(t) \geq r$
  $M(t) \leftarrow M(t-1) + (i, j)$
  $U(t) \leftarrow M(t-1)$
until no candidate pair (i, j) is found with $K_{ij} \geq r$

C. Percolation Process

PDGM needs input including two graphs, $G_1(V_1, E_1, W_1)$, $G_2(V_2, E_2, W_2)$, the set of initial seed nodes, A_0. $M(t)$ is defined as the set of matched pairs at time step t such that $M(0) = A_0$. $U(t)$ is the set of used matched pairs at time step t where $U(0) = \emptyset$ and $U(t) \subseteq M(t)$. $S_{ij}(t)$ is the similarity score at time step t for the candidate matched pair (i, j) where $S_{ij}(0) = 0$. At each time step t, PDGM runs the following procedures: (1) select one unused mapped pair (i', j') from $M(t-1) \setminus U(t-1)$; (2) compute credits $c_{i'j'}$ (e.g., $c_{i'j'}^{out}$ or $c_{i'j'}^{in}$) for all its neighboring pairs $(i, j) \in D_{i'j'}$; (3) compute $S_{ij}(t)$ by adding credits to $S_{ij}(t-1)$; and (4) add (i', j') into $U(t)$. This process continues until every unused mapped pair has been tried. Then the algorithm chooses a random pair with a maximal similarity score that has at least r common similarity witnesses. This is repeated in each time step t until the stopping condition is reached.

The stopping condition is as follows: (1) keep track of K_{ij} for candidate matched pair (i, j) counting common similarity witnesses; and (2) when no candidate pair is found with $K_{ij} \geq r$, the algorithm stops.

Fig. 1 demonstrates the example percolation process of PDGM using Eq. (6) for calculating $c_{i'j'}$. At initial time, A_0 includes two seed pairs (red pairs in Fig. 1). After the first iteration, (i_1, j_1) is chosen to be a mapped pair with $S_{i_1 j_1}(1) = 4, K_{i_1 j_1}(1) = 2$; in the next iteration, another candidate pair is chosen; this mapping is continued until no candidate pair is found for $K_{ij} \geq r = 2$ for any pair (i, j)'s.

V. Numerical Results and Analysis

In this section, we discuss performance metrics, environmental conditions used for simulation experiments, and comparative performance results.

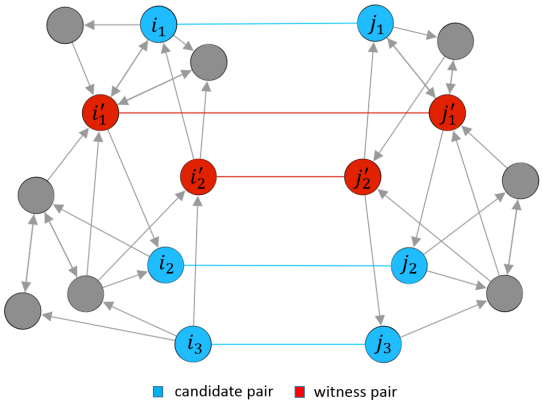


Fig. 1: Example percolation process of PDGM.

A. Metrics

We use the following three performance metrics:

- **Mapping rate** ($\mathcal{M}_r$): This refers to the final mapping rate based on the matched pairs identified by a given graph matching algorithm. Note that it does not necessarily reflect all correctly matched pairs. This implies that even if $\mathcal{M}_r$ is high, its precision can be low. $\mathcal{M}_r$ is computed by:

$$\mathcal{M}_r = \frac{a^*}{\min[n_1, n_2]} \tag{7}$$

  where a^* is the final mapping size and n_1, n_2 are the numbers of nodes in G_1, G_2, respectively.
- **Precision** ($\mathcal{P}_m$): This is the ratio of the number of correctly matched pairs over the number of identified anchor links a^* in the final map A^* based on the ground truth information, computed by:

$$\mathcal{P}_m = \frac{a_{TP}}{a^*} \tag{8}$$

  where a_{TP} denotes the number of correctly matched pairs (i.e., true positives) in the final map A^* where $a_{TP} \leq a^*$.
- **Percolation delay** (τ): This is the average delay to percolate a pair, implying the delay to identify a matched pair. We compute τ by:

$$\tau = \frac{\mathbb{T}}{a^*} = \langle k_1 \rangle \langle k_2 \rangle \tau_0 \tag{9}$$

  where $\langle k_1 \rangle$ is the mean degree of G_1, $\langle k_2 \rangle$ is the mean degree of G_2, and τ_0 is the time unit for a candidate pair to obtain the credit from a similarity witness pair. Each similarity witness has about $\langle k_1 \rangle \langle k_2 \rangle$ pairs, which leads to $\langle k_1 \rangle \langle k_2 \rangle \tau_0$, the time for each witness pair to add credits from all its neighbor pairs where the total running time is $\mathbb{T} = \langle k_1 \rangle \langle k_2 \rangle a^* \tau_0$.

B. Experimental Setup

We use real social network datasets, Foursquare-Twitter datasets [18], to conduct the comparative performance analysis. Table II describes the properties of the Foursquare-Twitter datasets. In Table II, we observe that the fraction of intersected vertices and edges is small (i.e., 43%, 8%) while existing approaches use synthetic datasets [6, 9, 11] with significantly more intersected vertices and edges (e.g., 100%, 49%). In our experiments, we set $a_0 = 1000$ and $d_{tr} = 1500$ by default unless they are varied to examine their impact on performance. Every result in this paper is an average of multiple replicates.

TABLE II: Foursquare-Twitter datasets [18]

Property	Foursquare	Twitter
# user	5,313	5,120
# following links	76,972	164,920
# anchor links	3,282	
Mean degrees $\langle k_1 \rangle$, $\langle k_2 \rangle$	28.98	64.42
Mean clustering coefficient	0.19	0.20
Fraction of intersected vertices α_V	0.43	
Fraction of intersected edges α_E	0.08	

C. Comparative Performance Analysis

In this section, we compare the performance of two variants of the proposed PDGM, called PDGM-OUT and PDGM-IN, with a baseline algorithm, called PGM [17], using a real network dataset [18]. PGM is originally designed only for undirected graphs. For fair comparison, it is implemented with the following similarity credit function to make it applicable to directed graphs:

$$c^{pgm}_{i'j'} = a_{ii'}a_{jj'} + a_{i'i}a_{j'j} \tag{10}$$

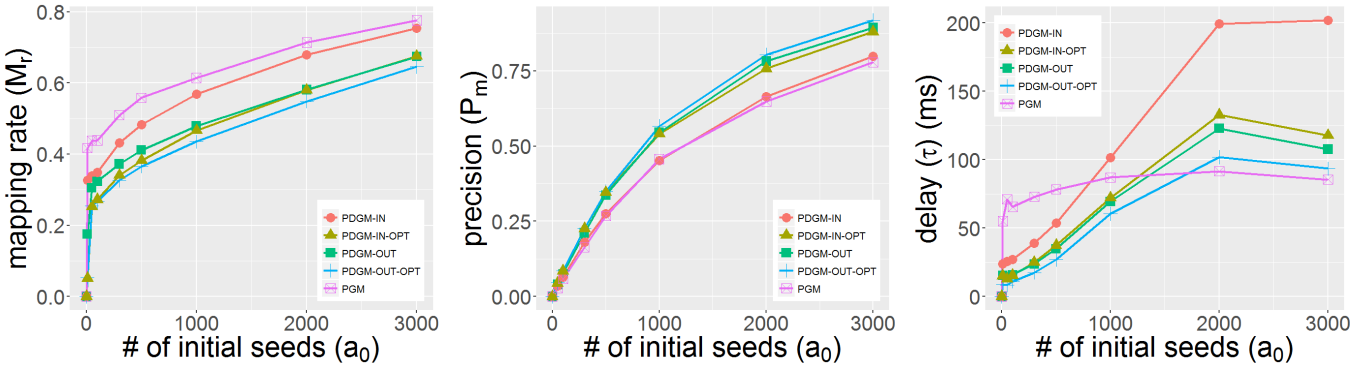
In PGM, no celebrity penalty is considered.

This section also shows the performance of PDGM-OUT and PDGM-IN when the celebrity penalty, d_{tr}, is optimized for maximum precision under a given set of initial seed nodes, denoted as PDGM-OUT-OPT and PDGM-IN-OPT. We use PDGM-OUT and PDGM-IN to label the cases without celebrity penalty, $d_{tr} = \infty$. The optimized result gives us an upper bound of PDGM precision.

Fig. 2 shows the effect of varying the size of initial seed nodes, a_0, on mapping rate, precision, and percolation delay. As expected, the overall performance in mapping rate and precision increases as a_0 increases. Since higher a_0 increases the chance to find more similarity witnesses for a pair of nodes during the percolation process. It leads the algorithm to percolate more time steps, higher mapping rate. However, higher a_0 also increases percolation delay to find a matched pair. The reason is, when more similarity witnesses for pairs of nodes are identified, the increasing rate of running time on computing similarity scores is larger than that of the number of matched pairs. According to Eq. (9), this will lead to the increase in percolation delay τ.

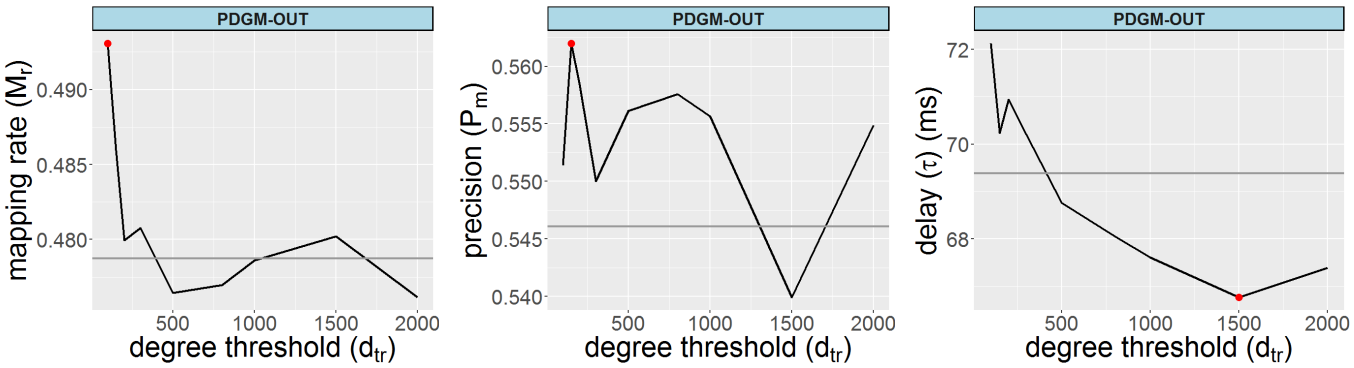
Fig. 2a shows that PGM outperforms PDGM in terms of the mapping rate. However, a higher mapping rate does not necessarily reflect all correctly matched pairs, because fast but incorrectly identified matched pairs lead to cascading failures by identifying more incorrectly matched pairs.

Very interestingly, we observe the precision performance is the opposite. In Fig. 2b, PDGM-OUT-OPT performs the



(a) Mapping rate ($\mathcal{M}_r$) (b) Precision ($\mathcal{P}_m$) (c) Percolation delay (τ)

Fig. 2: Performance comparison with respect to varying the size of initial seeds, a_0.



(a) Mapping rate ($\mathcal{M}_r$) under varying d_{tr} (b) Precision ($\mathcal{P}_m$) under varying d_{tr} (c) Percolation delay (τ) under varying d_{tr}

Fig. 3: Performance of PDGM with respect to varying the celebrity penalty factor, d_{tr}.

best among all. In addition, PDGM-OUT and PDGM-IN-OPT perform fairly well compared to other schemes. This proves that higher mapping rate does not necessarily result in higher precision.

In Fig. 2c, we show the effect of varying a_0 on the percolation delay, τ, under various PDGM schemes and the baseline PGM scheme. Notice that PDGM-OUT-OPT performs fairly well for varying values of a_0 selected in the experiment. In particular, when a_0 is sufficiently low, PDGMs outperforms PGM since a penalty factor is introduced in PDGMs, which helps to filter out nodes with a high in-degree so that a smaller number of nodes remain to be considered for a matching decision. However, the shorter percolation delay does not necessarily generate high precision, as observed in Figs. 2b.

D. Effect of a Celebrity Penalty Threshold in PDGM

Now we demonstrate how the celebrity penalty factor affects the performance of PDGM. To avoid clutter, we only present the effect of d_{tr} on PDGM-OUT that has the best performance in precision and percolation delay as shown in Fig. 2.

Fig. 3 shows the effect of d_{tr} on mapping rate, precision, and percolation delay of PDGM-OUT when $a_0 = 1000$. We label the optimal performance with a red dot. In addition, we label PDGM-OUT without the penalty factor ($d_{tr} = \infty$) by the gray line. In mapping rate and precision as shown in Figs. 3a and 3b, lower d_{tr} is preferred. We can always find an optimal d_{tr} for mapping rate and precision, respectively. The reason is that if d_{tr} is too high, PDGM-OUT will not be able to filter out less important neighbors, making less distinct similarities between pairs. On the other hand, as shown in Fig. 3c, percolation delay becomes low because the minimum r can be easily collected due to more qualified neighbors to be counted as matched pairs.

VI. Conclusion

In this paper, we studied how to link multiple accounts by the same user across different social networks based on the percolation-based graph matching (PGM) method. The proposed PDGM is the first PGM algorithm that is applicable in directed networks and validated based on a real network dataset. Based on the proposed PDGM algorithm, we identified matched pairs across two real networks, given a set of initial pre-matched seeding pair nodes. Through our extensive simulation experiments, PDGM-OUT, as one of PDGM variants, outperformed all other schemes because the outgoing edges play as a key feature representing a

user's unique social interaction patterns. We identified the optimal setting of celebrity penalty factor d_{tr} under which PDGM-OUT is optimized in matching precision or percolation delay by penalizing the importance of a relationship when a neighbor is a high in-degree node. PDGM-OUT-OPT using an optimal celebrity penalty factor maximized precision of matching pairs, penalizing the importance of a relationship when a neighbor is a high in-degree node, e.g., celebrity.

For our future research directions, we plan to conduct the following items: (1) improving PDGM by considering temporal, spatial information (e.g., dynamic connectivity over time or location information); (2) combining a user's static identification features with dynamic graph structural features to further improve the correct matched ratio; and (3) investigating the impact of the features of initial seeding nodes (e.g., selecting high centrality nodes).

ACKNOWLEDGMENT

This work was supported in part by the U. S. Army Research Office under contract number W911NF-12-1-0445, DTRA CNIMS Contract HDTRA1-11-D-0016-0001, NIH Grant 1R01GM109718, NSF BIG DATA Grant IIS-1633028, NSF DIBBS Grant ACI-1443054.

REFERENCES

[1] F. Buccafurri, G. Lax, A. Nocera, and D. Ursino, "Discovering missing me edges across social networks," *Information Sciences*, vol. 319, no. C, pp. 18–37, Oct. 2015.

[2] D. Conte, P. Foggia, C. Sansone, and M. Vento, "Graph matching applications in pattern recognition and image processing," in *IEEE International Conference on Image Processing*, Sept. 2003.

[3] A. Egozi, Y. Keller, and H. Guterman, "A probabilistic approach to spectral graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 18–27, Jan. 2013.

[4] O. Goga, P. Loiseau, R. Sommer, R. Teixeira, and K. P. Gummadi, "On the reliability of profile matching across large online social networks," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, 2015, pp. 1799–1808.

[5] S. Janson, T. Łuczak, T. Turova, and T. Vallier, "Bootstrap percolation on the random graph $g_{n,p}$," *The Annals of Applied Probability*, vol. 22, no. 5, pp. 1989–2047, Oct. 2012.

[6] E. Kazemi, S. H. Hassani, and M. Grossglauser, "Growing a graph matching from a handful of seeds," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, pp. 1010–1021, June 2015.

[7] G. W. Klau, "A new graph-based method for pairwise global network alignment," *BMC Bioinformatics*, vol. 10, no. 1, pp. 1–9, 2009.

[8] X. Kong, J. Zhang, and P. S. Yu, "Inferring anchor links across multiple heterogeneous social networks," in *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, San Francisco, CA, 2013, pp. 179–188.

[9] N. Korula and S. Lattanzi, "An efficient reconciliation algorithm for social networks," *Proceedings of the VLDB Endowment*, vol. 7, no. 5, pp. 377–388, Jan. 2014.

[10] A. Malhotra, L. Totti, W. Meira Jr., P. Kumaraguru, and V. Almeida, "Studying user footprints in different online social networks," in *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining*, Washington D.C., USA, 2012, pp. 1065–1070.

[11] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, Washington D.C., USA, 2008, pp. 111–125.

[12] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *Proceedings of the 25th International Conference on World Wide Web*, Quebec, Canada, 2016, pp. 707–719.

[13] L. Rossi and M. Musolesi, "It's the way you check-in: Identifying users in location-based social networks," in *Proceedings of the Second ACM Conference on Online Social Networks*, Dublin, Ireland, 2014, pp. 215–226.

[14] M. Srivatsa and M. Hicks, "Deanonymizing mobility traces: Using social network as a side-channel," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, Raleigh, NC, 2012, pp. 628–637.

[15] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, Oct. 2002.

[16] G. Wondracek, T. Holz, E. Kirda, and C. Kruegel, "A practical attack to de-anonymize social network users," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, 2010, pp. 223–238.

[17] L. Yartseva and M. Grossglauser, "On the performance of percolation graph matching," in *Proceedings of the First ACM Conference on Online Social Networks*, Boston, MA, 2013, pp. 119–130.

[18] J. Zhang, X. Kong, and P. S. Yu, "Transferring heterogeneous links across location-based social networks," in *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2014, pp. 303–312.