More Accurate Inference of User Profiles in Online Social Networks.

Raïssa Yapan Dougnon[1], Philippe Fournier-Viger[1],
Jerry Chun-Wei Lin[2], and Roger Nkambou[3]

[1] Dept. of Computer Science, Université de Moncton, Moncton, Canada
[2] School of Computer Science and Technology, Harbin Institute of Technology
Shenzhen Graduate School, China
[3] Dept. of Computer Science, Université du quebec à Montréal
eyd2562@umoncton.ca, philippe.fournier-viger@umoncton.ca,
jerrylin@ieee.org, nkambou.roger@uqam.ca

Abstract. Algorithms for social network user profiling suffer from one or more of the following limitations: (1) assuming that the full social graph is available for training, (2) not exploiting the rich information that is available in social networks such as group memberships and likes, (3) treating numeric attributes as nominal attributes, and (4) not assessing the certainty of its predictions. In this paper, we address these challenges by proposing an improved algorithm named PGPI+ (Partial Graph Profile Inference+). PGPI+ accurately infers user profiles under the constraint of a partial social graph using rich information about users (e.g. group memberships, views and likes), handles nominal and numeric attributes, and assesses the certainty of predictions. An experimental evaluation with more than 30,000 user profiles from the Facebook and Pokec social networks shows that PGPI+ predicts user profiles with considerably more accuracy and by accessing a smaller part of the social graph than five state-of-the-art algorithms.

Keywords: social networks, inference, user profiles, partial graph

1 Introduction

Online social networks have become extremely popular. Various types of social networks are used such as friendship networks (e.g. Facebook), professional networks (e.g. ResearchGate) and interest-based networks (e.g. Flickr). An important problem for ad targeting on social networks is that users often disclose few information publicly [1, 10]. To address this issue, an important sub-field of social network mining is now interested in developing algorithms to infer detailed user profiles using publicly disclosed information. Various approaches have been used to solve this problem such as relational Naïve Bayes classifiers [12], label propagation [9, 11], majority voting [4], linear regression [10], Latent-Dirichlet Allocation [2] and community detection [13]. It was shown that these approaches can accurately predict hidden attributes of user profiles in many cases. However, all these approaches suffer from at least two of the following four limitations.

1. **Assuming a full social graph.** Many approaches assume that the full social graph is available for training (e.g. [9]). However, in real-life, it is generally unavailable or may be very costly to obtain or update [2, 6]. A few approaches do not assume a full social graph such as majority-voting [4]. However, they do not let the user control the trade-off between the number of nodes accessed and prediction accuracy, which may lead to low accuracy.
2. **Not using rich information.** Several algorithms do not consider the rich information that is available on social networks. For example, several algorithms consider links between users and user attributes but do not consider other information such as group memberships, "likes" and "views" that are available on some social networks [1, 4, 9, 11–13].
3. **Not handling numeric attributes.** Many approaches treat numeric attributes (e.g. age) as nominal attributes [1, 4], which may decrease inference accuracy. Others are designed to handle numeric attributes but requires the full social graph, which is often unpractical [5, 9, 10, 13].
4. **Not assessing certainty.** Few approaches assess the certainty of their predictions. But this information is essential to determine if a prediction is reliable and actions should be taken based on the prediction. For example, if there is a low certainty that a user profile attribute is correctly inferred, it may be better to not use this attribute for ad targeting, rather than showing an ad that is targeted to a different audience [3].

To address limitations 1 and 2, the PGPI algorithm (Partial Graph Profile Inference) was recently proposed [6]. PGPI lets the user select how many nodes of the social graph can be accessed to infer a user profile, and can use not only information about friendship links and profiles but also about group memberships, likes and views, when available. In this paper, we present an extended version of PGPI named PGPI+ to also address limitations 3 and 4. Contributions are threefold. First, we design a new procedure for predicting values of numeric attributes. Second, we introduce a mechanism to assess the certainty of predictions for both nominal and numeric attributes. Third, we introduce four optimizations that considerably improve the overall prediction accuracy of PGPI.

We report results from an extensive empirical evaluation against PGPI and four other state-of-the-art algorithms, for 30,000 user profiles from the Facebook and Pokec social networks. Results show that the proposed PGPI+ algorithm can provide a considerably higher accuracy for both numeric and nominal attributes while accessing a much smaller number of nodes from the social graph. Moreover, results show that the calculated certainty well assesses the reliability of predictions. Moreover, an interesting result is that profile attributes such as status (student/professor) and gender can be predicted with more than 95% accuracy using PGPI+.

The rest of this paper is organized as follows. Section 2, 3, 4, 5 and 6 respectively presents the related work, the problem definition, the proposed algorithm, the experimental evaluation and the conclusion.

2 Related Work

We review recent work on social network user profile inference. Davis Jr et al. [4] inferred locations of Twitter users by performing a majority vote over the locations of directly connected users. A major limitation of this approach is that a single attribute is considered. Jurgens [9] predicted locations of Twitter/Foursquare users using a *label propagation* approach. However, it is an iterative algorithm that requires the full social graph since it propagates known labels to unlabeled nodes through links between nodes. Li et al. [11] proposed an iterative algorithm to deduce LinkedIn user profiles based on relation types. This algorithm also requires a large training set to discover relation types.

Mislove [13] applied community detection on more than 60k user profiles with friendship links, and then inferred user profiles based on similarity of members from the same community. Lindamood et al. [12] applied a Naïve Bayes classifier on 167k Facebook profiles with friendship links, and concluded that if links or attributes are erased, accuracy of the approach can greatly decrease. This study highlights the challenges of performing accurate predictions using few data. Recently, Blenn et al. [1] utilized bird flocking, association rule mining and statistical analysis to infer user profiles in a dataset of 3 millions Hyves.nl users. However, all these work assume that a large training set is available for training and they only use profile information and social links to perform predictions.

Chaabane et al. [2] inferred Facebook user profiles using Latent Dirichlet Allocation (LDA) and majority voting. The approach extracts a probabilistic model from music interests and additional information provided from Wikipedia. But it requires a large training set, which was difficult and time-consuming to obtain [2]. Kosinski et al. [10] also utilized information about user preferences to infer Facebook user profiles. Kosinski et al. applied Singular Value Decomposition to a huge matrix of users/likes and then used regression to perform prediction. A limitation of this work is that it does not utilize information about links between users and requires a very large training dataset.

He et al.[7] proposed an approach consisting of building a Bayesian network based on the full social graph to then predict user attribute values. The approach considers similarity between user profiles and links between users to perform predictions, and was applied to data collected from LiveJournal. Recently, Dong et al. [5] used graphical-models to predict the age and gender of users. Their study was performed with 1 billion phone and SMS data and 7M user profiles. Chaudhari [3] also used graphical models to infer user profiles. The approach has shown high accuracy on datasets of more than 1M users from the Twitter and Pokec social networks. A limitation of these approaches however, it that they assume a large training set for training. Furthermore, they only consider user attributes and links but not additional information such as likes, views and group membership.

Some of the above approaches infer numeric attributes, either by treating them as nominal attributes [1, 2, 4], or by using specific inference procedures. However, these latter require a large training set [5, 9, 10, 13].

Besides, current approaches generally do not assess the certainty of predictions. But this information is essential to determine if a prediction is reliable, and actions should be taken based on this prediction. To our knowledge, only Chaudhari [3] provides this information. However, this approach is designed to use the full social graph.

3 Problem Definition

The problem of user profiling is commonly defined as follows [1, 3, 9, 11–13].

Definition 1 (social graph). A *social graph* $\mathcal{G}$ is a quadruplet $\mathcal{G} = \{N, L, V, A\}$. N is the set of nodes in $\mathcal{G}$. $L \subseteq N \times N$ is a binary relation representing the links (edges) between nodes. Let be m attributes to describe users of the social network such that $V = \{V_1, V_2, ...V_m\}$ contains for each attribute i, the set of possible attribute values V_i. Finally, $A = \{A_1, A_2, ...A_m\}$ contains for each attribute i a relation assigning an attribute value to nodes, that is $A_i \subseteq N \times V_i$.

Example 1. Let be a social graph with three nodes $N = \{Tom, Amy, Lea\}$ and friendship links $L = \{(Tom, Lea), (Lea, Tom), (Lea, Amy), (Amy, Lea)\}$. Consider two attributes *gender* and *status*, respectively called attribute 1 and 2 to describe users. The set of possible attribute values for gender and status are respectively $V_1 = \{male, female\}$ and $V_2 = \{professor, student\}$. The relations assigning attributes values to nodes are $A_1 = \{(Tom, male), (Amy, female), (Lea, female)\}$ and $A_2 = \{(Tom, student), (Amy, student), (Lea, professor)\}$.

Definition 2 (Problem of inferring user profiles in a social graph). The problem of inferring the user profile of a node $n \in N$ in a social graph $\mathcal{G}$ is to guess the attribute values of n using the other information provided in $\mathcal{G}$.

The problem definition can be extended to consider additional information from social networks such as Facebook (views, likes and group memberships).

Definition 3 (extended social graph). An *extended social graph* $\mathcal{E}$ is a tuple $\mathcal{E} = \{N, L, V, A, G, NG, P, PG, LP, VP\}$ where N, L, V, A are defined as previously. G is a set of groups that a user can be a member of. The relation $NG \subseteq N \times G$ indicates the membership of users to groups. P is a set of publications such as pictures, texts, videos that are posted in groups. PG is a relation $PG \subseteq P \times G$, which associates a publication to the group(s) where it was posted. LP is a relation $LP \subseteq N \times P$ indicating publication(s) liked by each user (e.g. "likes" on Facebook). VP is a relation $VP \subseteq N \times P$ indicating publication(s) viewed by each user (e.g. "views" on Facebook), such that $LP \subseteq VP$.

Example 2. Let be two groups $G = \{book\_club, music\_lovers\}$ such that $NG = \{(Tom, book\_club), (Lea, book\_club), (Amy, music\_lovers)\}$. Let be two publications $P = \{picture1, picture2\}$ published in the groups $PG = \{(picture1, book\_club), (picture2, music\_lovers)\}$. The publications viewed by users are $VP = \{(Tom, picture1), (Lea, picture1), (Amy, picture2)\}$ while the publications liked by users are $LP = \{(Tom, picture1), (Amy, picture2)\}$.

Definition 4 (Problem of inferring user profiles in an extended social graph). The problem of inferring the user profile of a node $n \in N$ in an extended social graph $\mathcal{E}$ is to guess the attribute values of n using the information in $\mathcal{E}$.

But the above definitions assume that the full social graph may be used to perform predictions. The problem of inferring user profiles using a limited amount of information is defined as follows [6].

Definition 5 (Problem of inferring user profiles using a partial (extended) social graph). Let $maxFacts \in \mathbf{N}^+$ be a parameter set by the user. The problem of inferring the user profile of a node $n \in N$ using a partial (extended) social graph $\mathcal{E}$ is to accurately predict the attribute values of n by accessing no more than $maxFacts$ facts from the social graph. A *fact* is a node, group or publication from N, G or P (excluding n).

The above definition can be extended for numeric attributes. For those attributes, instead of aiming at predicting an exact attribute value, the goal is to predict a value that is as close as possible to the real value. Moreover, in this paper, we also extend the problem to consider the certainty of predictions. In this setting, a prediction algorithm must assign a *certainty value* in the [0,1] interval to each predicted value, such that a high certainty value indicates that a prediction is likely to be correct.

4 The Proposed PGPI+ Algorithm

We next present the proposed PGPI+ algorithm. Subsection 4.1 briefly introduces PGPI. Then, subsections 4.2, 4.3 and 4.4 respectively present optimizations to improve its prediction accuracy and coverage, and how it is extended to handle numerical attributes and assess the certainty of predictions.

4.1 The PGPI algorithm

The PGPI algorithm [6] is a lazy algorithm designed to perform predictions under the constraint of a partial social graph, where at most $maxFacts$ facts from the social graph can be accessed to make a prediction. PGPI (Fig. 1) takes as parameter a node n_i, an attribute k to be predicted, the $maxFacts$ parameter, a parameter named $maxDistance$, and an (extended) social graph $\mathcal{E}$. PGPI outputs a predicted value v for attribute k of node n_i. To predict the value of an attribute k, PGPI relies on a map M. This map stores pairs of the form (v, f), where v is a possible value v for attribute k, and f is positive real number called the *weight* of v. PGPI automatically calculates the weights by applying two procedures named PGPI-G and PGPI-N. These latter respectively update weights by considering the (1) views, likes and group memberships of n_i, and (2) its friendship links. After applying these procedures, PGPI returns the value v associated to the highest weight in M as the prediction. In PGPI, half of the $maxFacts$ facts that can be used to make a prediction are used by PGPI-G

and the other half by PGPI-N. If globally the $maxFacts$ limit is reached, PGPI does not perform a prediction. PGPI-N or PGPI-G can be deactivated. If PGPI-N is deactivated, only views, likes and group memberships are considered to make a prediction. If PGPI-G is deactivated, only friendship links are considered. In the following, we respectively refer to these versions of PGPI as PGPI-N and PGPI-G (and as PGPI-N+/PGPI-G+ for PGPI+).

PGPI-N works as follows. To predict an attribute value of a node n_i, it explores the neighborhood of n_i restricted by the parameter $maxDistance$ using a breadth-first search. It first initializes a queue Q and pushes n_i in the queue. Then, while Q is not empty and the number of accessed facts is less than $maxFacts$, the first node n_j in Q is popped. Then, $F_{i,j} = W_{i,j}/dist(n_i, n_j)$ is calculated. $W_{i,j} = C_{i,j}/C_i$, where $C_{i,j}$ is the number of attribute values common to n_i and n_j, and C_i is the number of known attribute values for node n_i. $dist(x, y)$ is the number of edges in the shortest path between n_i and n_j. Then, $F_{i,j}$ is added to the weight of the attribute value of n_j for attribute k, in map M. Then, if $dist(x, y) \leq maxDistance$, each unvisited node n_h linked to n_j is pushed in Q and marked as visited. PGPI-G is similar to PGPI-N. It is also a lazy algorithm. But it uses a majority voting approach to update weights based on group and publication information (views and likes). Due to space limitation, we do not describe it. The reader may refer to [6] for more details.

Algorithm 1: The PGPI algorithm

input : n_i: a node, k: the attribute to be predicted, $maxFacts$: a user-defined threshold, $\mathcal{E}$: an extended social graph
output: the predicted attribute value v

$M = \{(v, 0) | v \in V_k\}$;
`// Apply PGPI-G`
`// ...`
`// Apply PGPI-N`
Initialize a queue Q and add n_i to Q;
while Q is not empty and $|accessedFacts| < maxFacts$ **do**
  $n_j = Q.pop()$;
  $F_{i,j} \leftarrow W_{i,j}/dist(n_i, n_j)$;
  Update (v, f) as $(v, f + F_{i,j})$ in M, where $(n_j, v) \in A_k$;
  if $dist(n_i, n_j) \leq maxDistance$ **then for each** node $n_h \neq n_i$ such that $(n_h, n_j) \in L$ *and* n_h is unvisited, push n_h in Q and mark n_h as visited ;
end
return a value v such that $(v, z) \in M \wedge \nexists (v', z') \in M | z' > z$;

4.2 Optimizations to improve accuracy and coverage

In PGPI+, we redefine the formula $F_{i,j}$ used by PGPI-N by adding three optimizations. The new formula is $F_{i,j} = W_{i,j} \times (T_{i,j} + 1)/newdist(n_i, n_j) \times R$.

The first optimization is to add the term $T_{i,j} + 1$, where $T_{i,j}$ is the number of common friends between n_i and n_j, divided by the number of friends of n_i. This term is added to consider that two persons having common friends (forming a triad) are more likely to have similar attribute values. The constant 1 is used so that if n_i and n_j have no common friends, $F_{i,j}$ is not zero.

The second optimization is based on the observation that the term $dist(n_i, n_j)$ makes $F_{i,j}$ decrease too rapidly. Thus, nodes that are not immediate neighbors but were still close in the social graph had a negligible influence on their respective profile inference. To address this issue, $dist(n_i, n_j)$ is replaced by $newdist(n_i, n_j) = 3-(0.2\times dist(n_i, n_j))$, where it is assumed that $maxDistance < 15$. It was empirically found that this formula provides higher accuracy.

The third optimization is based on the observation that PGPI-G had too much influence on predictions compared to PGPI-N. To address this issue, we multiply the weights calculated using the formula $F_{i,j}$ by a new constant R. This thus increases the influence of PGPI-N+ on the choice of predicted values. In our experiments, we have found that setting R to 10 provides the best accuracy.

Furthermore, a fourth optimization is integrated in the main procedure of PGPI+. It is based on the observation that PGPI does not make a prediction for up to 50% of users when $maxFacts$ is set to a small value [6]. The reason is that PGPI does not make a prediction when it reaches the $maxFacts$ limit. However, it may have collected enough information to make an accurate prediction. In PGPI+, a prediction is always performed. This optimization was shown to greatly increase the number of predictions.

4.3 Extension to handle numerical attributes

The PGPI algorithm is designed to handle nominal attributes. In PGPI+, we performed the following modifications to handle numeric attributes. First, we modified how the predicted value is chosen. Recall that the value predicted by PGPI for nominal attributes is the one having the highest weight in M (line 12). However, for numeric attribute, this approach provides poor accuracy because few users have exactly the same attribute value. For example, for the attribute "weight", few users may have the same weight, although they may have similar weights. To address this issue, PGPI+ calculates the predicted values for numeric attributes as the weighed sum of all values in M.

Second, we adapted the weighted sum so that it ignores outliers because if unusually large values are in M, the weighted sum provides inaccurate predictions. For example, if a young user has friendship links to a few 20 years old friends but also a link to his 90 years old grandmother, the prediction may be inaccurate. Our solution to this problem is to ignore values in M that have a weight more than one standard deviation away from the mean. In our experiment, it greatly improves prediction accuracy for numeric attributes.

Third, we change how $W_{i,j}$ is calculated. Recall that in PGPI, $W_{i,j} = C_{i,j}/C_i$, where $C_{i,j}$ is the number of attribute values common to n_i and n_j, and C_i is the number of known attribute values for node n_i. This definition does not work well for numeric attributes because numeric attributes rarely have the same value. To

consider that numeric values may not be equal but still be close, $C_{i,j}$ is redefined as follows in PGPI+. The value $C_{i,j}$ is the number of values common to n_i and n_j for nominal attributes, plus a value $CN_{i,j,k}$ for each numeric attribute k. The value $CN_{i,j,k}$ is calculated as $(v_i - v_j)/\alpha_k$ if $(v_i - v_j) < \alpha_k$, and is otherwise 0, where α_k is a user-defined constant. Because $CN_{i,j,k}$ is a value in [0,1], numeric attributes may not have more influence than nominal attributes on $W_{i,j}$.

4.4 Extension to evaluate the certainty of predictions

PGPI+ also extends PGPI with the capability of calculating a certainty value $CV(v)$ for each predicted value v. For numeric attributes, calculating the certainty value of a predicted value v requires to find a way to assess how "accurate" the weighted sum for calculating v is. Intuitively, we can expect it to be accurate if (1) the amount of information taken into account by the weighted sum is large, and (2) if values considered in the weighted sum are close to each other. These ideas are captured in our approach by using the relative standard error. Let $E_M = \{v_1, v_2, ...v_m\}$ be the set of values in the map M that were used to calculate the weighted sum. The amount of information used by the weighted sum is measured as the number of updates made by PGPI-N+/PGPI-G+ to the map M, denoted as *updates*. The relative standard error is defined as $RSE(v) = stdev(E_M)/(\sqrt{updates} \times avg(E_M))$. The RSE is a value in the [0,1] interval that assesses how close the average of the sample might be to the average of the population. Because we want a certainty value rather than an error value, we calculate the certainty value of v as $CV(v) = 1 - RSE(v)$. A drawback of the RSE is however that it is sensible to outliers. To address this issue, we ignore values that are more than one standard deviation away from the mean to when calculating $CV(v)$.

For nominal attributes, calculating the certainty value of a predicted value v is done differently. Our idea is to evaluate how likely the weight of value v in M is to be as large at it is, compared to other weights in M. To estimate this, we use a simulation-based approach where larger is defined in terms of standard deviations from the mean. Let $F_M = \{f_1, f_2, ...f_m\}$ be the weights of values in the map M, and $f_{M,v}$ be the weight of v in map M. We initialize a value $count = 0$ and perform 1,000 simulations. During the j-th simulation, we create a map B, and perform *updates* random updates to B. At the end of the j-th simulation, we increase the *count* variable by 1 if $(f_{B,v} - avg(F_B))\ /stdev(F_B) \geq (f_{M,v} - avg(F_M))\ /stdev(F_M)$. After the 1,000 simulations, the certainty value is calculated as $CV(v) = count/1000$, which gives a value in the [0,1] interval.

5 Experimental Evaluation

We compared the accuracy of the proposed PGPI+, PGPI-N+ and PGPI-G+ algorithms with the original PGPI, PGPI-N and PGPI-G algorithms and four additional state-of-the-art algorithms for predicting attribute values of nodes in a social network. The three first are Naïve Bayes classifiers [8]. Naïve Bayes

(NB) infer user profiles strictly based on correlation between attribute values. Relational Naïve Bayes (RNB) consider the probability of having friends with specific attribute values. Collective Naïve Bayes (CNB) combines NB and RNB. To be able to compare NB, RNB and CNB with the proposed algorithms, we have adapted them to work with a partial graph. This is done by training them with $maxFacts$ users chosen randomly instead of the full social graph. The last algorithm is label propagation (LP) [9]. Because LP requires the full social graph, its results are only used as a baseline. Each algorithm was tuned with optimal parameter values.

Datasets. Two datasets are used. The first one is 11,247 user profiles collected from Facebook in 2005 [14]. Each user is described according to seven attributes: a student/faculty status flag, gender, major, second major/minor (if applicable), dorm/house, year, and high school, where year is a numerical attribute. The second dataset is 20,000 user profiles from the Pokec social network obtained at `https://snap.stanford.edu/data/`. It contains 17 attributes, including three numeric attributes: age, weight and height. Because both datasets do not contain information about groups, and this information is needed by PGPI-G and PGPI, synthetic data about groups was generated using the generator proposed in [6], using the same parameters. This latter generator is designed to generate group having characteristics similar to real-life groups.

Accuracy for nominal attributes w.r.t number of facts. We first ran all algorithms while varying the $maxFacts$ parameter to assess the influence of the number of accessed facts on accuracy for nominal attributes. The *accuracy* for nominal attributes is defined as the number of correctly predicted values, divided by the number of prediction opportunities. Fig. 1 shows the overall results for the Facebook and Pokec datasets. Note that PGPI algorithms are not shown in these tables due to lack of space. It can be observed that PGPI+/PGPI-N+/PGPI-G+ provides the best results. For example, on Facebook, PGPI+ and PGPI-G+ provide the best results when 66 to 700 facts are accessed, and for less than 66 facts, PGPI-N+ provides the best results followed by PGPI+. No results are provided for PGPI-N+ for more than 306/6 facts on Facebook/Pokec because PGPI-N+ relies solely on links between nodes to perform predictions and the datasets do not contains enough links. It is also interesting to note that PGPI-N+ only uses real data (contrarily to PGPI+/PGPI-G+) and still performs better than all other algorithms. The algorithm providing the worst results is LP (not shown in the figure). LP provides an accuracy of 43.2%/47.31% on Facebook/Pokec. This is not good considering that LP uses the full social graph of more than 10,000 nodes. For the family of Naïve Bayes algorithms, NB has the best overall accuracy. It can be further observed that the accuracy of PGPI+ algorithm is up to 34% higher than PGPI, which shows that proposed optimizations have a major influence on accuracy.

Best results for each nominal attribute. We also analyzed accuracy for each nominal attribute separately. The best results in terms of accuracy for each attribute and algorithm for Facebook and Pokec are respectively shown in Table 1 and 2. The last row of each table indicates the number of accessed facts

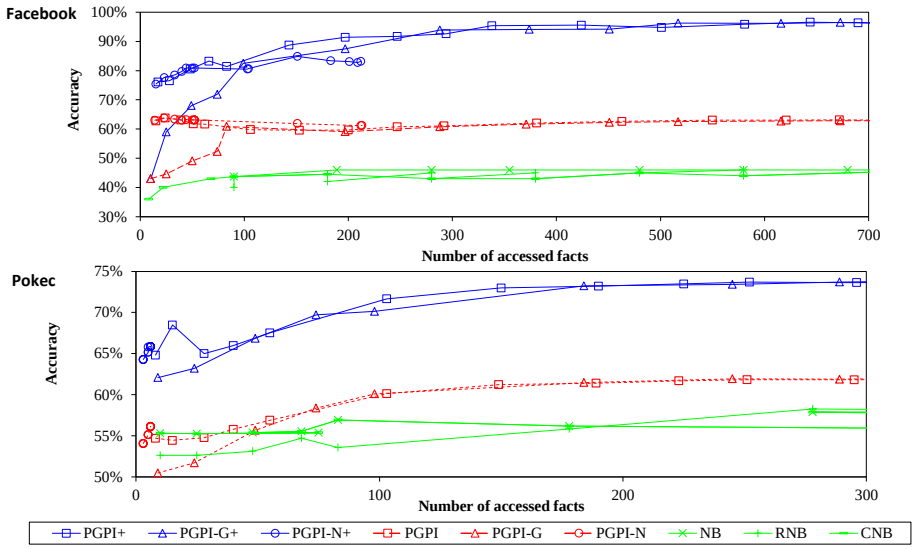

Fig. 1. Accuracy w.r.t. number of accessed facts for nominal attributes

to obtain these results. The best accuracy was in general achieved by PGPI+ algorithms for all attributes.

attribute	PGPI+	PGPI-N+	PGPI-G+	NB	RNB	CNB	LP
status	92.0%	92.6 %	**93.0%**	88.0%	80.2%	88.0%	83.0%
gender	**96.1%**	84.7 %	95.8%	51.1%	57.7%	47.3%	50.4%
major	**33.8%**	30.4%	32.8%	16.6%	15.0%	9.0%	16.7%
minor	76.0%	76.4%	**76.6%**	74.4%	74.2%	74.0%	56.3%
residence	**64.6%**	62.4%	64.4%	55.6 %	55.0%	54.8%	49.1%
school	7.4%	**16.8%**	7.0%	10.6%	9.8%	10.6%	10.1%
$\|facts\|$	482	226	431	189	580	189	10K

Table 1. Best accuracy results for nominal attributes on Facebook

Best results for each numeric attribute. We also compared the best accuracy of PGPI/PGPI+ algorithms for numeric attributes on Pokec/Facebook in terms of average error and standard deviation of predicted values from the real values. Results (Table 3) indicates that PGPI+ performs the best on overall. The other algorithms could not be compared for numeric attributes because they are designed for nominal attributes. We attempted to compare with the algorithm of Kosinski [10]. However, linear regression failed using a partial social graph for training.

attribute	PGPI+	PGPI-N+	PGPI-G+	NB	RNB	CNB	LP
Gender	95.60%	61.40%	**95.77%**	52.80%	53.80%	53.60%	49.20%
English	**76.35%**	63.79%	76.00%	69.74%	69.74%	69.74%	65.40%
French	**87.46%**	84.48%	87.42%	86.91%	85.60%	86.87%	67.15%
German	62.39%	54.31%	**62.85%**	47.83%	48.12%	47.83%	50.00%
Italian	94.87%	94.25%	94.85%	94.65%	95.38%	**95.41%**	85.75%
Spanish	**95.15%**	94.54%	95.14%	94.38%	95.08%	94.29%	80.52%
Smoker	65.21%	62.34%	**65.42%**	63.43%	63.43%	63.12%	60.19%
Drink	**71.65%**	63.36%	71.47%	70.41%	70.41%	70.41%	49.16%
Marital status	**76.57%**	70.86%	76.40%	76.11%	76.02%	76.07%	69.92%
Hip-hop	**86.51%**	82.20%	86.47%	86.01%	85.83%	85.93%	61.82%
Rap	69.33%	63.78%	**69.52%**	69.08%	69.35%	69.35%	45.78%
Rock	77.93%	73.80%	**78.09%**	76.33%	74.93%	74.93%	53.69%
Disco	58.40%	52.50%	**58.56%**	50.07%	53.18%	53.46%	47.28%
Metal	**86.19%**	83.52%	86.15%	84.75%	84.61%	84.61%	63.79%
Region	18.60%	10.20%	**18.71%**	6.20%	6.20%	6.20%	10.00%
$\|facts\|$	334	6	347	375	378	278	10k

Table 2. Best accuracy results for nominal attributes on Pokec

algorithm	year	age	weight	height
PGPI+	0.95 (0.85)	2.94 (4.55)	**9.83 (10.32)**	**7.70 (11.75)**
PGPI-N+	0.68 **(0.72)**	3.92 (4.56)	14.60 (12.56)	10.32 (12.55)
PGPI-G+	0.99 (0.89)	2.89 **(4.45)**	9.83 (10.37)	7.71 (11.76)
PGPI	0.46 (0.93)	2.55 (4.80)	11.67 (11.53)	8.75 (12.43)
PGPI-N	0.46 (0.87)	4.35 (5.11)	17.28 (15.61)	14.0 (36.52)
PGPI-G	**0.39** (0.93)	**2.20** (4.78)	10.75 (10.86)	8.35 (12.45)

Table 3. Average error and standard deviation for numerical attributes

Assessment of certainty values. We also assessed certainty values calculated by PGPI+. Fig. 2 shows the best accuracy obtained for numerical attributes "year" and "age" for PGPI-N+. Each line represents the accuracy of predictions having at least a given certainty value. It can be seen, that a high certainty value generally means a low average error, standard deviation and coverage (percentage of predictions made), as expected. Results of PGPI+/PGPI-G+ are similar and not shown due to lack of space. For nominal attributes, the accuracy of predictions made by PGPI-N+/PGPI+/PGPI-G+ having a certainty no less than 0 and no less than 0.7 are respectively 59%/59%/61% and 91%/91%/93% on Facebook, which also shows that the proposed certainty assessment is a good indicator of accuracy.

Best results using the full social graph. We also compared the accuracy of the algorithms using the full social graph. The best accuracy obtained for each algorithm on the Facebook and Pokec datasets is shown in Table 4. It can be observed that the proposed PGPI+ algorithms provide an accuracy that is

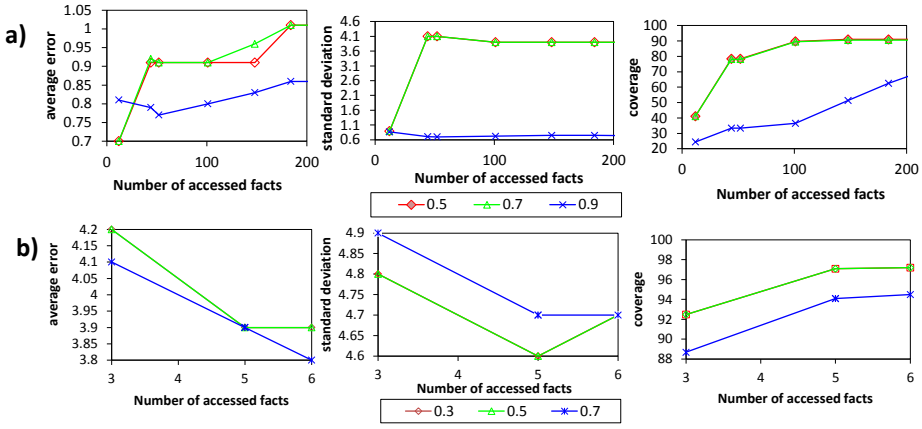

Fig. 2. Average error, standard deviation and coverage w.r.t. certainty for attributes (a) year and (b) age

considerably higher than the accuracy of the compared algorithms, even when using the full social graph.

algorithm	Facebook	Pokec	algorithm	Facebook	Pokec
PGPI+	**96.6**	73.8	PGPI-G	62.8	56.2
PGPI-N+	96.4	**73.9**	NB	48.67	57.48
PGPI-G+	84.9	65.9	RNB	50.11	56.37
PGPI	63.8	62.0	CNB	50.11	56.40
PGPI-N	63.8	62.1	LP	48.03	47.31

Table 4. Best accuracy for nominal attributes using the full social graph

6 Conclusion

We proposed an improved algorithm named PGPI+ for user profiling in online social networks under the constraint of a partial social graph and using rich information. PGPI+ extends the PGPI algorithm with new optimizations to improve its prediction accuracy and coverage, to handle numerical attributes, and assess the certainty of predictions. An experimental evaluation with more than 30,000 user profiles from the Facebook and Pokec social networks shows that PGPI+ predicts user profiles with considerably more accuracy and by accessing a smaller part of the social graph than five state-of-the-art algorithms. Moreover, an interesting result is that profile attributes such as status (student/professor) and gender can be predicted with more than 95% accuracy using PGPI+.

References

1. Blenn, N., Doerr, C., Shadravan, N., Van Mieghem, P.: How much do your friends know about you?: reconstructing private information from the friendship graph. In: Proc. of the Fifth Workshop on Social Network Systems, pp. 1–6. ACM (2012)
2. Chaabane, A., Acs, G., Kaafar, M.A.: You are what you like! information leakage through users interests. In: Proc. of the 19th Annual Network and Distributed System Security Symposium, The Internet Society (2012)
3. Chaudhari, G., Avadhanula, V., Sarawagi, S.: A few good predictions: selective node labeling in a social network. In: Proc. of the 7th ACM international conference on Web search and data mining, pp. 353–362. ACM (2014)
4. Davis Jr, C. A. et al.: Inferring the location of twitter messages based on user relationships. Transactions in GIS, 15(6), 735–751 (2011)
5. Dong, Y., Yang, Y., Tang, J., Yang, Y., Chawla, V. N.: Inferring user demographics and social strategies in mobile social networks. In: Proc. of the 20th ACM international conference on Knowledge discovery and data mining, pp. 15–24. ACM (2014)
6. Dougnon, Y. R., Fournier-Viger, P., Nkambou, R.: Inferring User Profiles in Social Networks using a Partial Social Graph. In: Proc. 28th Canadian Conference on Artificial Intelligence, Springer, LNAI 9091, pp. 84–99 (2015)
7. He, J., Chu, W. W., Liu, Z. V.: Inferring privacy information from social networks. In: Proc. of 2006 IEEE International Conference on Intelligence and Security Informatics. pp. 154–165. Springer, Heidelberg (2006)
8. Heatherly, R., Kantarcioglu, M., Thuraisingham, B.: Preventing private information inference attacks on social networks. IEEE Transactions on Knowledge and Data Engineering. 25(8), 1849–1862 (2013)
9. Jurgens, D.: Thats what friends are for: Inferring location in online social media platforms based on social relationships. In: Proc. of the 7th International AAAI Conference on Weblogs and Social Media, pp 273–282, AAAI Press (2013)
10. Kosinski, M., Stillwell, D., Graepel, T.: Private traits and attributes are predictable from digital records of human behavior. National Academy of Sciences, 110(15), 5802–5805 (2013)
11. Li, R., Wang, C., Chang, K. C. C. User profiling in an ego network: co-profiling attributes and relationships. In: Proc. of the 23rd international conference on World wide web, pp. 819–830. ACM (2014)
12. Lindamood, J., Heatherly, R., Kantarcioglu, M., Thuraisingham, B.: Inferring private information using social network data. In: Proc. of the 18th international conference on World wide web, pp. 1145–1146. ACM (2009)
13. Mislove, A., Viswanath, B., Gummadi, K. P., Druschel, P.: You are who you know: inferring user profiles in online social networks. In: Proc. of the 3rd ACM international conference on Web search and data mining, pp. 251–260. ACM (2010)
14. Traud, A. L., Mucha, P. J., Porter, M. A.: Social structure of Facebook networks. Physica A: Statistical Mechanics and its Applications. 391(16), 4165–4180 (2012)