Accurate Online Social Network User Profiling

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. We present PGPI+ (Partial Graph Profile Inference+) an improved algorithm for user profiling in online social networks. PGPI+ infers user profiles under the constraint of a partial social graph using rich information about users (e.g. group memberships, views and likes) and handles nominal and numeric attributes. Experimental results with 20,000 user profiles from the Pokec social network 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

Various approaches have been proposed to infer detailed user profiles on social networks using publicly disclosed information such as relational Naïve Bayes classifiers [8], label propagation [6], majority voting [3], linear regression [7], Latent-Dirichlet Allocation [1] and community detection [9]. It was shown that these approaches can accurately predict hidden attributes of user profiles in many cases. However, all these approaches assume that the full or a large part of the social graph is available for training (e.g. [6]). However, in real-life, it is generally unavailable or may be very costly to obtain or update [4]. A few approaches do not assume a full social graph such as majority-voting [3]. 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. Furthermore, several algorithms do not consider the rich information that is available on social networks (e.g. group memberships, "likes" and "views") [3, 6, 8, 9]. Besides, many approaches treat numeric attributes (e.g. age) as nominal attributes [3], which may decrease inference accuracy. To address these limitations, we present the PGPI+ (Partial Graph Profile Inference) algorithm, which extends our previous work PGPI [4]. PGPI+ lets the user select how many nodes of the social graph can be accessed to infer a user profile, can use not only information about friendship links and profiles but also about group memberships, likes and views, when available. Moreover, it can predict values of numeric attributes.

2 Problem Definition

The problem of user profiling is commonly defined as follows [2, 6, 8, 9]. 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 $A_i \subseteq N \times V_i$. 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).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 $LP \subseteq VP$.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). Lastly, 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.

3 The Proposed PGPI+ Algorithm

The proposed PGPI+ algorithm extends PGPI [4] (Algorithm 1) to improve its prediction accuracy and coverage, and to handle numerical attributes. PGPI [4] is a lazy algorithm designed to perform predictions under the constraint of a partial social graph. PGPI 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 [4] 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 $v$
1 $M = \{(v, 0) | v \in V_k\}$;
2 // Apply PGPI-G and PGPI-N
3 Initialize a queue $Q$ and add $n_i$ to $Q$;
4 while $Q$ is not empty and $|accessedFacts| < maxFacts$ do
5     $n_j = Q.pop()$; $F_{i,j} \leftarrow W_{i,j}/dist(n_i, n_j)$;
6     Update $(v, f)$ as $(v, f + F_{i,j})$ in $M$, where $(n_j, v) \in A_k$;
7     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 ;
8 end
9 return a value $v$ such that $(v, z) \in M \wedge \nexists (v', z') \in M | z' > z$;
```

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+. 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.

Extension to handle numerical attributes In PGPI+, to handle numeric attributes, we first 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. 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 Experimental Evaluation

We compared the accuracy of PGPI+, PGPI-N+ and PGPI-G+ algorithms with PGPI, PGPI-N and PGPI-G, and three Naïve Bayes classifiers [5]: (1) Naïve Bayes (NB) infer user profiles based on correlation between attribute values, (2) Relational Naïve Bayes (RNB) consider the probability of having friends with specific attribute values, and (3) Collective Naïve Bayes (CNB) combines NB and RNB. Those three latter algorithms are adapted to work with a partial graph by training them with *maxFacts* users chosen randomly of the full social graph. We used a dataset of 20,000 user profiles from the Pokec social network obtained at `https://snap.stanford.edu/data/`. We used 10 attributes, including three numeric attributes. Synthetic data about groups was generated as in [4].

Fig. 1 shows the influence of the number of accessed facts on accuracy for each algorithm when the *maxFacts* parameter is varied, for nominal attributes. The *accuracy* is the number of correctly predicted values, divided by the number of prediction opportunities. PGPI algorithms are not shown in this figure due to lack of space. It can be observed that PGPI+/PGPI-N+/PGPI-G+ provides the best results. No results are provided for PGPI-N+ for more than 6 facts because the dataset do not contains enough links. It is interesting to note that PGPI-N+ only uses real data (contrarily to PGPI+/PGPI-G+) and still performs better than all other algorithms.

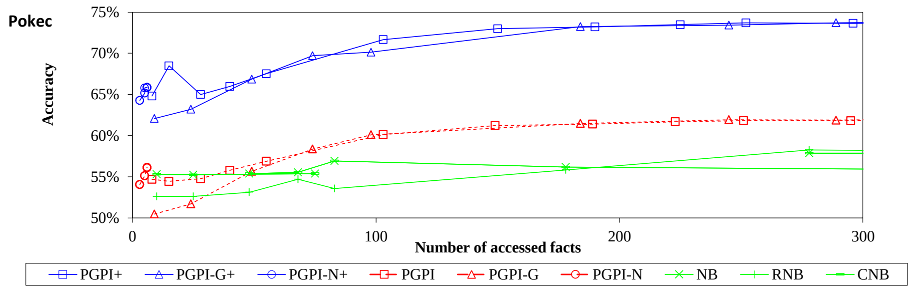

Fig. 1. Accuracy w.r.t. number of accessed facts for nominal attributes

Table 1 shows the best results in terms of accuracy for each attribute and algorithm. The last row of each table indicates the number of accessed facts to obtain these results. The best accuracy was in general achieved by PGPI+ algorithms for all attributes.

Table 2 compares the best accuracy of PGPI/PGPI+ algorithms for numeric attributes in terms of average error and standard deviation of predicted values from the real values. PGPI+ performs the best on overall. Other algorithms could not be compared because they are designed for nominal attributes.

Table 3 shows the best accuracy for each algorithm. The proposed PGPI+ algorithms provide an accuracy that is considerably higher than the accuracy of the compared algorithms.

attribute	PGPI+	PGPI-N+	PGPI-G+	NB	RNB	CNB
Gender	95.60%	61.40%	**95.77%**	52.80%	53.80%	53.60%
English	**76.35%**	63.79%	76.00%	69.74%	69.74%	69.74%
French	**87.46%**	84.48%	87.42%	86.91%	85.60%	86.87%
German	62.39%	54.31%	**62.85%**	47.83%	48.12%	47.83%
Italian	94.87%	94.25%	94.85%	94.65%	95.38%	**95.41%**
Spanish	**95.15%**	94.54%	95.14%	94.38%	95.08%	94.29%
Smoker	65.21%	62.34%	**65.42%**	63.43%	63.43%	63.12%
Drink	**71.65%**	63.36%	71.47%	70.41%	70.41%	70.41%
Marital status	**76.57%**	70.86%	76.40%	76.11%	76.02%	76.07%
Region	18.60%	10.20%	**18.71%**	6.20%	6.20%	6.20%
$\|facts\|$	334	6	347	375	378	278

Table 1. Best accuracy results for nominal attributes

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

Table 2. Average error and standard deviation for numerical attributes

algorithm	accuracy	algorithm	accuracy
PGPI+	73.8	PGPI-G	56.2
PGPI-N+	**73.9**	NB	57.48
PGPI-G+	65.9	RNB	56.37
PGPI	62.0	CNB	56.40
PGPI-N	62.1	LP	47.31

Table 3. Best accuracy for nominal attributes using the full social graph

5 Conclusion

We proposed an algorithm named PGPI+ for user profiling in online social networks under the constraint of a partial social graph and using rich information. PGPI+ improves PGPI's prediction accuracy/coverage and handle numerical attributes. Experimental results show that PGPI+ predicts user profiles with much 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 gender can be predicted with more than 95% accuracy using PGPI+.

References

1. 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)
2. 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)
3. Davis Jr, C. A. et al.: Inferring the location of twitter messages based on user relationships. Transactions in GIS, 15(6), 735–751 (2011)
4. 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)
5. 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)
6. 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)
7. 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)
8. 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)
9. 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)