Cold Start Link Prediction

Vincent Leroy
INSA Rennes, UEB
Rennes, France
vincent.leroy@irisa.fr

B. Barla Cambazoglu
Yahoo! Research
Barcelona, Spain
barla@yahoo-inc.com

Francesco Bonchi
Yahoo! Research
Barcelona, Spain
bonchi@yahoo-inc.com

ABSTRACT

In the traditional link prediction problem, a snapshot of a social network is used as a starting point to predict, by means of graph-theoretic measures, the links that are likely to appear in the future. In this paper, we introduce *cold start link prediction* as the problem of predicting the structure of a social network when the network itself is totally missing while some other information regarding the nodes is available. We propose a two-phase method based on the *bootstrap probabilistic graph*. The first phase generates an implicit social network under the form of a probabilistic graph. The second phase applies probabilistic graph-based measures to produce the final prediction. We assess our method empirically over a large data collection obtained from Flickr, using interest groups as the initial information. The experiments confirm the effectiveness of our approach.

Categories and Subject Descriptors H.2.8 [**Database Management**]: Database Applications - *Data mining*
General Terms Algorithms, Experimentation
Keywords Link prediction, probabilistic graph, social networks

1. INTRODUCTION

Link prediction, introduced by Liben-Nowell and Kleinberg [15], refers to a basic computational problem underlying social network evolution in time. Given a snapshot of a social network at time t and a future time t', the problem is to predict the new friendship links that are likely to appear in the network within the time interval $[t, t']$. As Liben-Nowell and Kleinberg state, the link prediction problem is about to what extent the evolution of a social network can be modeled using features intrinsic to the network itself. Indeed, in their framework, they consider only the features that are based on the link structure of the network.

In this paper, we tackle a similar, but fundamentally different problem, which we dub the *cold start link prediction* problem: similar because we also aim at predicting links, but different because we aim at doing this without any, even partial, knowledge of an existing link structure. Herein, we assume that either the social network explicitly exists, but is kept secret by its owner, or it does not exist at all. In both cases, we are interested in predicting possible links among the users of a service by exploiting other types of information available, e.g., interest groups, tags, sales data.

Consider, for instance, a company C selling music online and a general-purpose social networking service S. Suppose C and S have made the following agreement: (1) S offers a functionality so that users can make reviews of their favorite songs and these reviews are made available to their contacts, and (2) when a user clicks on a song title, the user is redirected to the corresponding page in C's website, but (3) S keeps the structure of its social network as a trade secret (this may be a mandatory constraint due to privacy regulations). Thanks to this agreement, users of S might influence each other to buy songs, and the easiest way for them to purchase songs would be through the website of C. In this scenario, C only owns information about the buying history of its customers, but has no explicit knowledge of the social network implicitly underlying its set of customers.

The question tackled herein is the following: *can C nevertheless infer the social network (to an acceptable level of accuracy), just using the information contained in the sales history?*

This would be useful for many reasons. First, it would enable adoption of *viral marketing* strategies [8, 14]. Second, it would facilitate social-network-based services, such as providing personalized advertisement. Third, if C decides, in the future, to adopt its own explicit social network, this may be used to recommend possible friends to users, thus speeding up the initial growth of the network.

In this context, we propose a two-phase method based on the *bootstrap probabilistic graph* for cold start link prediction. In the first phase, based on some limited information (potentially, weakly correlated with the link structure of the network), the method predicts the existence of links. The output of this phase is a probabilistic graph, i.e., a graph where each edge is labeled with a probability representing the confidence of the prediction, or in other terms, the uncertainty of the existence of a link. The second phase takes as input the probabilistic graph and refines it by adopting graph-theoretic measures as done in the classical link prediction setting. The difference is that, in our case, the input graph is probabilistic and hence the traditional measures must be adapted to deal with this case [18].

We apply our method to a large data collection obtained from Flickr, a popular online community for image and video

KDD'10, July 25–28, 2010, Washington, DC, USA.
Copyright 2010 ACM 978-1-4503-0055-1/10/07 ...$10.00.

sharing. We keep the existing social network (made of directed arcs[1]) as the ground truth for the link structure that we aim to predict. As the available auxiliary information, we use users' memberships in interest groups.

Three observations are note-worthy. First, the cold start link prediction problem is intrinsically a very difficult binary prediction problem due to the skewness of the target variable. In fact, assuming directed arcs as in our context, given n nodes, we have a universe of n^2-n possible links, of which only a very small fraction exists in the ground truth. In our data, the positive class (existing links) constitutes approximately 0.07% of all possible links.

The second observation is that we apply our method starting with "little" information, which provides a very small coverage of existing links. Indeed, in Flickr, interest group membership[2] is a very weak predictor of links, as a group gathers people interested in photos regarding a specific subject or technique (e.g., *"Nikon Selfportrait"*, *"HDR Panoramas"*, *"Cat and Dog: not Cat or Dog"*), and they are not groups of friends or small communities (as it is mostly the case in Facebook groups). More precisely, in our data, considering only the users who belong to at least one group, we have approximately 28M links, of which only 1.9% (approximately 550K links) are among two users that share at least one common group. Despite the difficulty of our prediction task, our two-phase method based on the bootstrap probabilistic graph can achieve good prediction performance.

Finally, it is very important to note that, even though the first phase features used herein are geared to group membership information, the overall framework we propose is general and applicable to any input information. In some cases, the available auxiliary information could be much more predictive than the one used in our experiments or more information might be available. If this is the case, all available information should be used in order to bootstrap the probabilistic graph as accurately as possible.

Our contributions can be summarized as follows:

- We introduce cold start link prediction as the problem of predicting the link structure of a social network assuming unavailability of an initial network and using other available information.
- Our work has privacy and security implications as it sheds light on to what extent a social network can be reconstructed and how resilient an anonymization solution is to link prediction attacks.
- We propose a two-phase method based on the bootstrap probabilistic graph as a feasible solution to the cold start link prediction problem.
- We apply our method to predict the link structure of the Flickr social network by using only the interest group membership information. As discussed previously, group membership is a weak predictor for friendship and hence suits well to our purpose of starting the social network with little available information.
- We assess the predictive power of various features based on group membership (phase 1), such as the time a user joins a group and the size of the group. The features we use herein can be considered to be somewhat general and are applicable to other problem instances as long as the available information can be mapped to a group structure. For instance, a thread on a discussion board can be mapped to a group, and the same features can be used for predicting friendship between the users of a bulletin board service.
- We adapt various graph-theoretic measures to deal with probabilistic graphs (phase 2).

The rest of the paper is organized as follows. In the next section, we discuss related work. In Section 3, we present the formal definition of the problem and the proposed two-phase method. In Section 4, we describe the data we use for assessing our method, which is then developed in Section 5 (phase 1) and Section 6 (phase 2). Finally, in Section 7, we discuss future research lines and conclude the paper.

2. RELATED WORK

Liben-Nowell and Kleinberg [15] introduce the link prediction problem and show that simple graph-theoretic measures, such as the number of common neighbors, are sufficient to efficiently detect links that are likely to appear in a social network. Through the use of more elaborated measures that consider the ensemble of all paths between two nodes (e.g., the Katz measure), they further improve the prediction quality. The graph features presented in Section 6 are inspired by Liben-Nowell and Kleinberg's work but they are adapted to probabilistic graphs.

Taskar *et al.* [21] apply link prediction to a social network of universities. They rely on machine learning techniques and use personal information of users (music, books, etc.) to increase the accuracy of predictions. Following a similar approach, O'Madadhain *et al.* [17] focus on predicting events between entities and use the geographic location as a feature. Clauset *et al.* [6] apply link prediction to biology and physics using hierarchical models in order to detect links that have not been observed during experimentation. All these approaches rely on the availability of an initial link structure for prediction while the method we present addresses the cold start problem, i.e., the case where no initial link structure is available.

Several probabilistic models such as Markov logic [7], relational Markov networks [21], Markov random fields [5], and probabilistic relational models [12] have been used to efficiently capture the relation in data. Unfortunately, these approaches have not been proved to scale as they have been tested only on small data sets.

Van der Aalst *et al.* [22] extract a social network from logs of interactions between workers in a company. Similar works include mining email communications [3] and proximity interactions [9]. In each case, the authors start with a very dense graph and the idea is to identify the social network in this graph. The difficulty of the task is due to the huge amount of data. In our problem, we have the opposite situation: the information used to generate the bootstrap probabilistic graph, which enables link prediction, is very sparse. Hence, the information needs to be spread, not pruned.

Since we deal with reconstructing information that might be considered sensitive (the links of a social network), our work has privacy implications. In fact, our method can be used by an attacker to threaten link privacy in a social network, thus it can be used to test the resilience of anonymization solutions. Several papers [2, 13, 16] study the problem of social network anonymization and the impact of the available knowledge on the inference of hidden information that should remain secret. Zheleva *et al.* [26] consider a so-

[1] We use the terms *arc* and *link* interchangeably.

[2] `http://www.flickr.com/groups`

cial network in which some users hide their information and others make it public. The applications presented include predicting the country of Flickr users through their group membership information. Their results indicate that group membership is a weak predictor.

3. PROBLEM AND METHOD

3.1 Problem

We are given a set $\mathcal{U}$ of users and a multiset $\mathcal{G}$ of groups of users. We denote the set of groups to which a user u belongs to, $m(u) = \{g \in \mathcal{G} | u \in g, g \subseteq \mathcal{U}\}$, as her *membership set*. Our task is to reconstruct the links of a social graph $\mathcal{N} = (\mathcal{U}, \mathcal{A})$, where the nodes are the users and the arcs $\mathcal{A} \subseteq \mathcal{U} \times \mathcal{U}$ represent a (one-way) relation between two users. Reconstructing the social network $\mathcal{N}$ means to predict which of the links in $\mathcal{U} \times \mathcal{U}$ actually exist in $\mathcal{A}$, or in other terms, to build a function $f : \mathcal{U} \times \mathcal{U} \to \{0, 1\}$.

3.2 Method

We propose a two-phase method based on the bootstrap probabilistic graph for cold start link prediction. During the first phase, we predict the existence of links based only on the group membership information. The output of the first phase is the bootstrap probabilistic graph, i.e., a directed probabilistic graph $BPG = (\mathcal{U}, E, p_1)$, where $E \subseteq \mathcal{U} \times \mathcal{U}$, and every link $(u, v) \in E$ is labeled with a probability $p_1(u, v) > 0$ representing the confidence (or uncertainty) about the link's existence, i.e., $p_1 : \mathcal{U} \times \mathcal{U} \to [0, 1]$.

In particular, after the first phase, we have $p_1(u, v) = 0$ and $p_1(v, u) = 0$ for every user pair (u, v), where $m(u) \cap m(v) = \emptyset$. This is because if two users have no groups in common, a prediction cannot be made about the existence of a link between them. Moreover, we have $p_1(u, v) > 0$ for every user pair (u, v) such that $m(u) \cap m(v) \neq \emptyset$ (this will also hold for the reverse arc (v, u)). Links with null probabilities do not exist in BPG.

The second phase takes as input the bootstrap probabilistic graph BPG, and it refines the probability distribution p_1 into a new probability distribution p_2, by means of graph-based features. Therefore, the output of the second phase is a probabilistic graph $PG = (\mathcal{U}, E, p_2)$. After the second phase, some links that previously had $p_1(u, v) = 0$ can now possibly have a non-null score, $p_2(u, v) > 0$, thus extending the overall recall of the method.

3.3 Result presentation

In most real-world social networks, the links in a social graph $\mathcal{A}$ form only a small fraction of the total number of possible links, i.e., $|\mathcal{A}| \ll |\mathcal{U}|^2$. This means that *accuracy* is not a very meaningful measure in this context, given that, by predicting always 0 (the link does not exist), it is possible to achieve an accuracy of 99.93% in our data. Also, comparing different predictors by means of *precision* and *recall* is not very appropriate, given the very low maximum recall achievable (only 0.037 in our data). Therefore, in order to compare the performance of different predictive functions by eliminating the skewness between possible and existing links, we adopt the *ROC curve* metric [19] as the main way of presenting our results. For the best predictor of each feature group, we also provide *recall/fallout* ratios. Recall is the ratio between the number of *true positives* (correctly predicted existing links) and *positives* (existing links) while fallout is the ratio between the number of *false positives* (links erroneously predicted to exist) and *negatives* (not existing links). Given a predictor function f, we may interpret recall and fallout as the following probabilities, respectively: $p(f(u, v) = 1 | (u, v) \in \mathcal{A})$ and $p(f(u, v) = 0 | (u, v) \notin \mathcal{A})$. Now, suppose to have recall/fallout = 8 for a predictor f. This means that for two users $(u, v) \in \mathcal{A}$ for whom a link exists, it is 8 times more likely to have $f(u, v) = 1$ than for two users who are not connected.

Table 1: Number and type of links

	Possible links	Contact	Friend	Family
Total	39, 328, 640, 910	28, 249, 755	13, 529, 634	2, 145, 691
Phase 1	54, 945, 936	553, 977	282, 168	42, 426
Phase 2	1, 165, 664, 850	1, 072, 595	499, 811	76, 453

4. DATASET

Flickr is a highly popular online social network, whose primary objective is to facilitate sharing of images among people. In Flickr, a user can place other users in three privilege classes: *contact*, *friend*, and *family*. Depending on the class, the user can restrict access to its properties (e.g., images, videos). In this work, unless otherwise stated, we work on the contact class. In Flickr, links are directed. In our data, we found that approximately only 1/3 of the links are unidirectional. Most of the features we use in Section 5 are symmetric. This means that we predict the same likelihood for links' existence in both directions.

4.1 Dataset preparation

We sample a subset of the entire Flickr social network by applying the snowball sampling strategy, starting from a single, highly connected seed user and following the contact links between users in an iterative manner. The adopted sampling strategy increases the chance of selecting more active users, who have higher connectivity in the network (i.e., more links). In our case, this is desirable as users with few or no friends are relatively less interesting for our prediction task. For each user in the sample set, we store all links and groups associated with the user as well as some other information in a MySQL database for latter processing. From the sampled set of users, we remove the ones who are not members of any group. This is because the proposed techniques are applicable to users who have at least one group membership.

4.2 Dataset properties

After the above-mentioned pruning, we are left with 198,315 users. The type and number of existing links are reported in Table 1. The same table (second line) also reports how many of these existing links are among two users that have at least one common group: this is the maximum number of links predictable in the first phase, or in other terms, links for which we will have $p_1(u, v) > 0$. In the second phase, we use measures based on paths formed by the links between users. This means that we cannot predict the existence of a link between two users, each belonging to a different connected component of the bootstrap probabilistic graph. The maximum number of links predictable in the second phase is also reported in Table 1 (the third line). The number of groups we have is 69,793. Various properties of our dataset are displayed in Table 2.

Table 2: Dataset properties

Property	Min.	Avg.	Max.
# of users in a group	1	8.8	3,497
# of groups of a user	1	3.1	172
# of contact links of a user	1	142.5	11,956
# of friend links of a user	1	68.2	11,121
# of family links of a user	1	10.8	4,250

Table 3: Features evaluated in the bootstrap phase

Type	Name	Dir.	Formula
# of groups	`sum`	$\geq$	$\|m(u)\|+\|m(v)\|$
	`prod`	$\geq$	$\|m(u)\|\times\|m(v)\|$
Common groups	`overlap`	$\geq$	$\|c(u,v)\|$
	`frac_1`	$\geq$	$\|c(u,v)\|/\|m(u)\|$
	`frac_2`	$\geq$	$\|c(u,v)\|/\|m(v)\|$
	`jaccard`	$\geq$	$\|c(u,v)\|/\|m(u)\cup m(v)\|$
	`cos`	$\geq$	$\overrightarrow{m(u)}\cdot\overrightarrow{m(v)}/(\|\|\overrightarrow{m(u)}\|\|\times\|\|\overrightarrow{m(v)}\|\|)$
Group size	`min_s`	$\leq$	$\min_{g\in c(u,v)}\|g\|$
	`avg_s`	$\leq$	$\mathrm{avg}_{g\in c(u,v)}\|g\|$
	`sum_rec_s`	$\geq$	$\sum_{g\in c(u,v)}(1/\|g\|)$
	`ad_ad_s`	$\geq$	$\sum_{g\in c(u,v)}(1/\log(\|g\|))$
Inter-arrival time	`min_t`	$\leq$	$\min_{g\in c(u,v)} t(u,v,g)$
	`avg_t`	$\leq$	$\mathrm{avg}_{g\in c(u,v)} t(u,v,g)$
	`sum_rec_t`	$\geq$	$\sum_{g\in c(u,v)}(1/t(u,v,g))$
	`ad_ad_t`	$\geq$	$\sum_{g\in c(u,v)}(1/log(t(u,v,g)))$

According to Fig. 1 (left), the frequency of group sizes follows a highly skewed distribution, i.e., there are few, very large but many, very small groups. 35.3% of groups have only one member. Groups of size less than 3 constitute about half of the total number of groups. Frequency distribution for group membership is even more skewed. 47.0% of users are members in only one group. The number of users who are members in at most 10 groups constitutes 95.6% of the total number of users. A highly skewed distribution is also observed in frequency of users' link counts (Fig. 1 (right)).

5. BOOTSTRAP PHASE

5.1 Basic features

In the first phase, we bootstrap the probabilistic graph using the group membership information. In particular, we explore four types of features: *number of groups*, *number of common groups*, *size of common groups*, and *difference in joining time*. Since, in this phase, the probabilities are assigned to only the links between users who share at least one group, in the rest of this section, we report ROC curves computed only on this subset of links. For the recall/fallout curves, we instead use the whole dataset, as this gives a better idea about the discriminative power of a feature. Throughout this section, the reader may refer to Table 3 for definitions of the features. We denote by $c(u,v)$ the set of groups that are common to both users u and v, i.e., $c(u,v)=m(u)\cap m(v)$. The absolute value of the difference in time that u and v joined group g is denoted by $t(u,v,g)$.

5.1.1 Number of groups

The number of groups of a user might be a good indicator of the user's level of engagement and activity in the social network. As the user is more active, he may tend to have more links. Fig. 2 shows how the number of links of a user changes as the number of group memberships increases. We observe a very linear behavior, which may indicate a correlation between the number of groups and the number of links. We also observe that the number of out-links increases at a slightly faster rate than in-links as users join more groups.

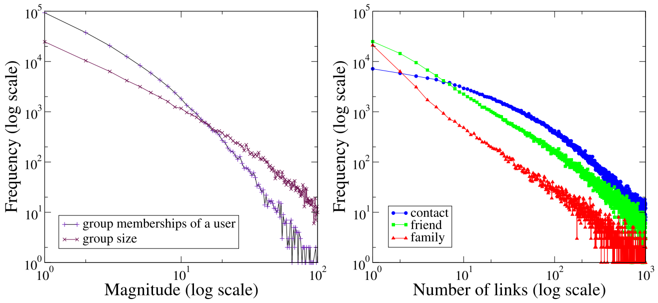

Figure 1: (Left) Frequency distributions for group membership and group size. (Right) Frequency distribution for the number of links that users have.

We evaluate two features based only on the number of groups. Given two users u and v, we define `sum` and `prod`, respectively, as the sum and product of $|m(u)|$ and $|m(v)|$ values. Obviously, as the feature values increase, the likelihood of having a link increases. In Fig. 3, we report the recall/fallout ratio for different thresholds of `prod`. The plot gives an indication of how predictive the feature is. As an example, for two linked users u and v, it is 200 times more likely to have $|m(u)|\times|m(v)|\geq 1000$ than two users without a link.

Fig. 4 shows the ROC curves for the two features: according to the plot, the performances of `sum` and `prod` are very close, but `prod` performs slightly better.

5.1.2 Common groups

Intuitively, being a member of the same groups should be a strong indicator of the existence of a possible link. An active group member may influence his existing friends to join the group as new members. This means that members of the same group are more likely to have existing friendship links among themselves. From another perspective, groups may be a suitable medium to meet other users and form friendships, thus groups may lead to creation of new links.

Fig. 5 verifies this hypothesis by measuring the fraction of links among users having membership in the same groups. Specifically, for each value x of common groups, we compute

$$\frac{|\{u,v\in\mathcal{U}\mid x=|c(u,v)|\wedge(u,v)\in\mathcal{A}\}|}{|\{u,v\in\mathcal{U}\mid x=|c(u,v)|\}|}.$$

As expected, the fraction of links increases as the users have more groups in common.

Fig. 6 reports the recall/fallout curve for different numbers of common groups. We can observe that for two linked users, the probability of having more than 10 common groups is approximately 600 times larger than for two users that have no link. This number grows to 1800 for the probability of having no less than 24 common groups.

We evaluated five different features based on common groups: `overlap`, which is the number of common groups; `frac_1` and `frac_2`, which are the overlap normalized by

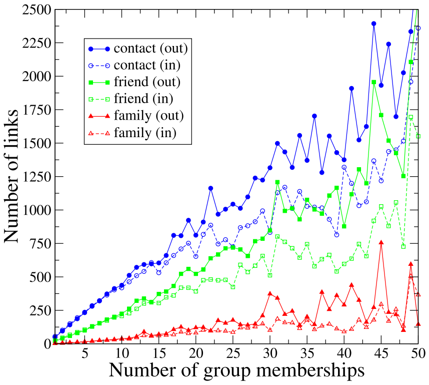

Figure 2: Indegrees and outdegrees of users as their number of group memberships increases.

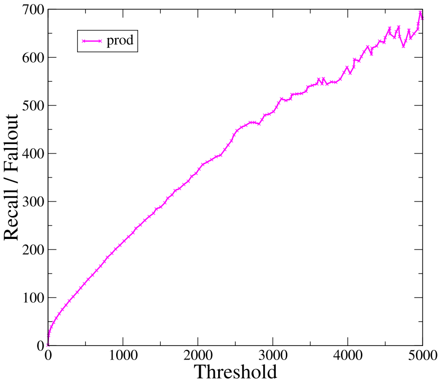

Figure 3: The recall/fallout curve for the `prod` feature.

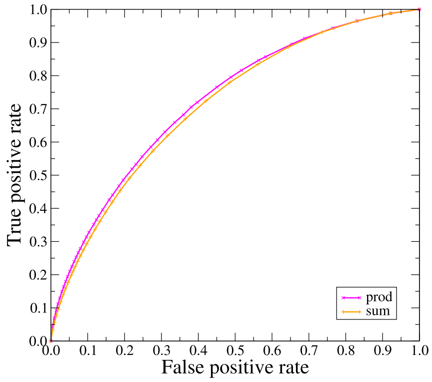

Figure 4: ROC curves for features related to the number of group memberships of users.

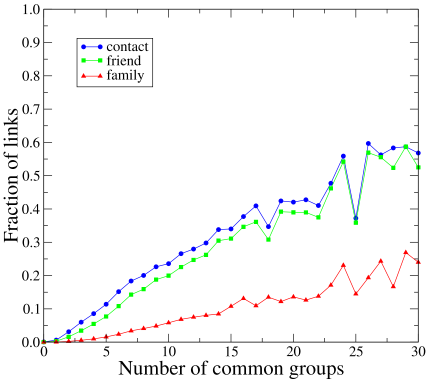

Figure 5: Fraction of links between user pairs as their number of common groups increases.

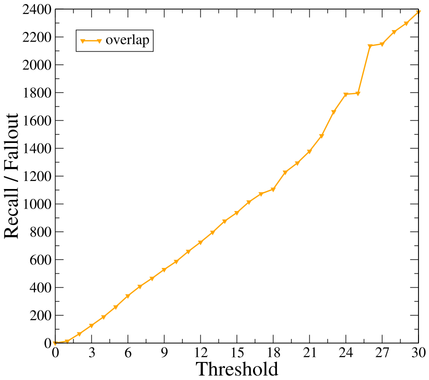

Figure 6: The recall/fallout curve for the `overlap` feature.

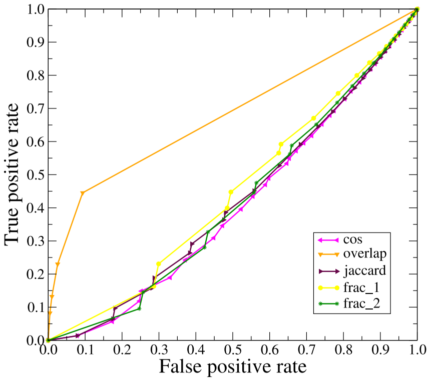

Figure 7: ROC curves for features related to similarity of group memberships.

the number of groups of the first and second users, respectively; `jaccard`, which is the Jaccard coefficient; and `cos`, indicating the *cosine similarity* commonly used in information retrieval. ROC curves of these five features are shown in Fig. 7. Interestingly, we observe that all features except for `overlap` perform worse than random prediction, which would correspond to the diagonal of the plot. This can be easily explained with the very high number of pairs of users being members of only one group. When two of such users are part of the same group, they receive the maximum value of `frac_1`, `frac_2`, `jaccard`, and `cos` although these two users are very likely to be not linked. We observed this effect also for variations of these features, such as the weighted cosine similarity feature (using tf-idf weighting), and also for some other features with normalization.

5.1.3 Group size

It can be claimed that two users are more likely to be friends if they are members of a small group than a large group as 1) group founders are more likely to prefer their friends over other users in invitations they send and 2) large groups are more likely to be general-purpose groups, attracting different users with equal likelihood. We verify this claim in Fig. 8, which shows the density of links with increasing group size. The link density is computed as

$$\frac{\sum_{g\in\mathcal{G}} |\{u,v\in\mathcal{U} \mid (u,v)\in\mathcal{A} \wedge x=|g| \wedge g\in c(u,v)\}|}{|\{g\in\mathcal{G} \mid x=|g|\}| \times x \times (x+1)},$$

where the average link count for groups of size x is normalized with the maximum possible link count ($x\times(x+1)$).

For contact and friend links, we observe a significant drop in density values at very small group sizes, followed by a linear drop as the group size increases. The density of family links is not affected much by the increase in the group size, potentially, due to the very low number of such links.

We try four different features based on the size of common groups: `min_s` and `avg_s` denote the minimum and the average size of the common groups, respectively; the summation of the reciprocal of size is denoted by `sum_rec_s`; and *Adamic/Adar-size* is denoted by `ad_ad_s`. The last feature is inspired by the measure defined by Adamic and Adar in [1] for deciding when two personal home pages are strongly related, and then borrowed and adapted by Liben-Nowell and Kleinberg [15] to deal with common neighbors in the context of link prediction. Here, we re-adapt this measure to deal with the size of common groups and compute

$$\texttt{ad\_ad\_s}(u,v) = \sum_{g\in c(u,v)} \frac{1}{\log(|g|)}.$$

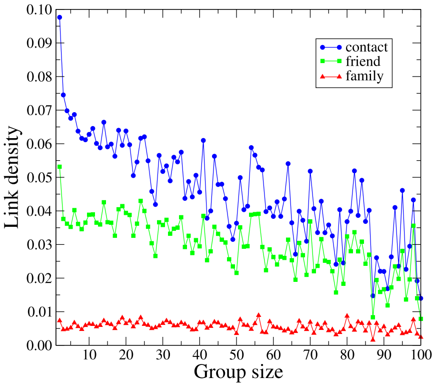

Figure 8: Link density within a group as group size increases.

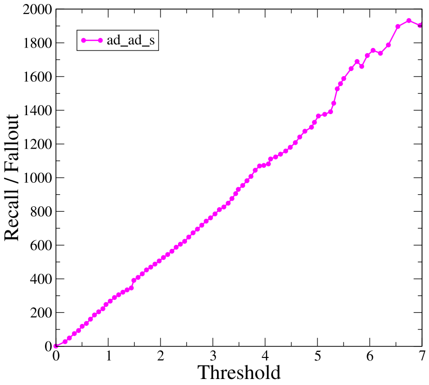

Figure 9: The recall/fallout curve for the `ad_ad_s` feature.

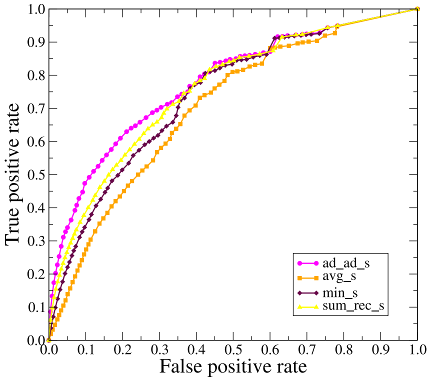

Figure 10: ROC curves for features related to group sizes.

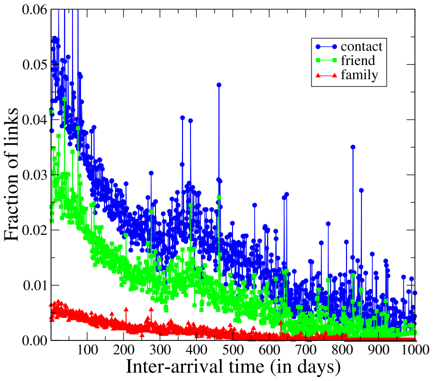

Figure 11: Fraction of links with varying inter-arrival time.

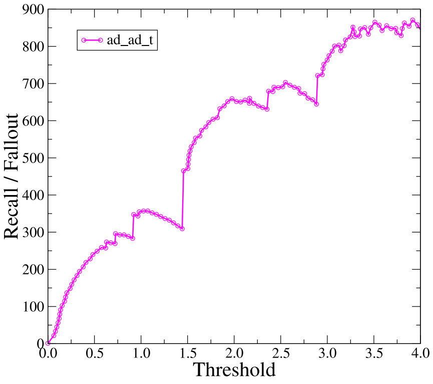

Figure 12: The recall/fallout curve for the `ad_ad_t` feature.

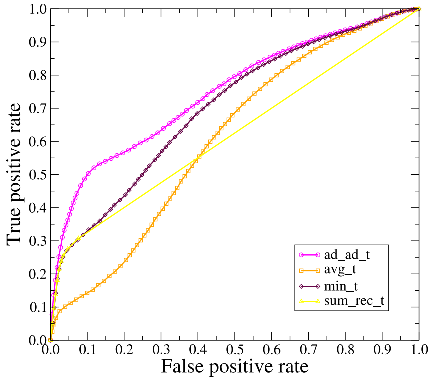

Figure 13: ROC curves for features related to the difference in dates of membership.

The recall/fallout curve for `ad_ad_s` is displayed in Fig. 9. We observe that the probability of having `ad_ad_s`$(u, v) \geq 1$ for an existing link (u, v) is approximately 200 times larger than for a non-existing link (u, v), and the ratio keeps growing almost linearly.

Fig. 10 shows that features that are based on group size perform quite well, with `ad_ad_s` outperforming the others.

5.1.4 Difference in joining time

In the last set of features, we investigate the temporal coherency between linked users joining the same group. We may expect that friends are likely to inform each other from existence of a group just before (or just after) joining it. Hence, we may expect that linked users are likely to join the same group with small time gaps (inter-arrival time).

Fig. 11 shows the fraction of links with increasing inter-arrival time. For each possible inter-arrival time value x (discretized into days), we compute

$$\frac{|\{u,v \in \mathcal{U}, g \in \mathcal{G} \mid (u, v) \in \mathcal{A} \wedge x = t(u, v, g) \wedge g \in c(u, v)\}|}{|\{u,v \in \mathcal{U}, g \in \mathcal{G} \mid x = t(u, v, g) \wedge g \in c(u, v)\}|}.$$

According to the figure, as expected, linked users are more likely to join the same group with a small inter-arrival time. It is interesting to note that an increase is observed in the likelihood of having a link, around a year inter-arrival time. This may be explained by the existence of "seasonal" groups, i.e., groups that refer to events held once per year and that attract new members in that period (e.g., *"Glastonbury Festival"* or *"Christmas Worldwide"*).

As the features, we try the same feature set we used for the group size, simply by replacing group size with inter-arrival time: `min_t`, `avg_t`, `sum_rec_t`, and `ad_ad_t`. As in the case of group size features, *Adamic/Adar-time* (`ad_ad_t`) performs the best among all features of this class (Figs. 12 and 13). The irregular shape of the ROC curve as well as of the recall/fallout curve are due to the seasonal behavior discussed before.

5.2 Combining basic features

Next, we try to combine a number of features from previous sections to create a single, hybrid feature with higher predictive power than the basic features. For this purpose, we evaluate various possible combinations of our best performing features, trying to find a good trade-off between predictive power and simplicity (which also translates to generality). The best performing combination turns out to be `ad_ad_s` × `ad_ad_t` × log `prod`, referred to as `combined`.

An important observation is that the features `ad_ad_t`, `ad_ad_s`, and `prod` perform relatively well for high, medium, and low confidence intervals, respectively. This is the rea-

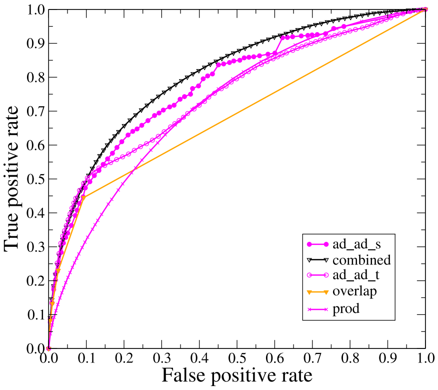

Figure 14: ROC curves for the best feature from each category and the best combined feature.

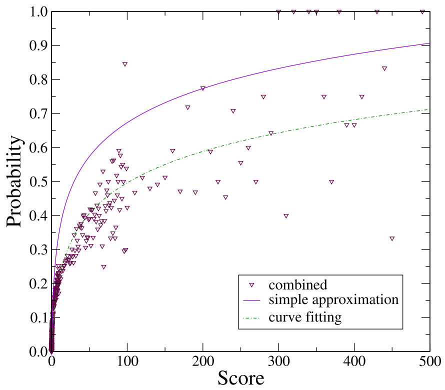

Figure 15: Mapping scores to probabilities, for generating the bootstrap probabilistic graph.

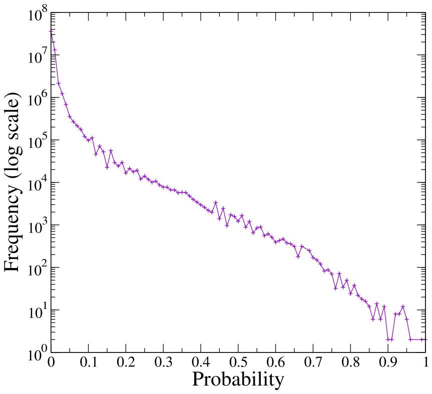

Figure 16: Distribution of probabilities in the bootstrap probabilistic graph.

son for the `combined` feature, which unifies them, to perform the best across all intervals. No features from the class using the number of common groups is directly used in the combined feature as this is subsumed in the two Adamic/Adar features, which compute a sum over all common groups.

Fig. 14 compares `combined` against the best-performing feature from each of the four categories. We use `combined` to bootstrap the probabilistic graph, as shown next.

5.3 Bootstrap probabilistic graph

So far, we have proposed various measures and evaluated their predictive power. We have then combined them under a simple but yet effective feature. We now finalize the first phase of our method by producing the bootstrap probabilistic graph. To this end, we need to convert the scores provided by the `combined` feature into probabilities. This is a mandatory step to be able to combine the values of the edges in a meaningful way.

Converting scores to probabilities is not straightforward since the relation between them is often not linear. This problem has been studied for different kinds of classifiers [11, 23, 24, 25], but with score distributions different from the one we observe in our case.

In Fig. 15, we observe a logarithmic shape in the distribution of probabilities with respect to scores. Using a curve fitting algorithm, we could map the function to the data, but depending on the feature used, this mapping could be completely different. In our case, to remain general, we just assume the knowledge that it follows a logarithmic distribution. We design a very simple function that maps the highest score output by the combined feature to a probability of 1 and assigns the remaining probabilities as

$$\mathit{probability} = \frac{\log(\mathit{score}+1)}{\log(\mathit{max\_score}+1)}.$$

Fig. 15 shows that our simple approximation is very rough. However, it is good enough as it will be shown in the next section. The probability distribution in the bootstrap probabilistic graph is reported in Fig. 16. The graph consists of 1,238 connected components, of which 42 have more than 1,000 nodes, 10 have more than 5,000 nodes, and the largest connected component has more than 20,000 nodes.

6. PROBABILISTIC GRAPH MEASURES

6.1 Graph-theoretic features

In the first phase of our method, we have predicted, for some pairs of users, the probability to have a link. In the second phase, we refine and extend this prediction by considering transitivity of contact relationship. As shown in [15], users who have many common contacts are more likely to be friends. Using graph-based features, we can spread the link prediction to pairs of users who have no common groups but share contacts. Therefore, we compute graph-based measures on the bootstrap probabilistic graph for all pairs of users who are in the same connected component. If two users are not connected by a path in the probabilistic graph, then they will have a null probability also after phase 2. In the following, we adapt to the probabilistic case three graph-based measures that are reported to perform well in [15]: `common_neighbors`, `katz`, and `rooted_pagerank`.

6.1.1 *Probabilistic common neighbors*

Having a high number of common contacts may be an indication of the existence of a link. We adapt this idea to our probabilistic graph in a straightforward way. For a given user pair (u, v), `common_neighbors` simply computes the sum of the probability that each node is connected to both u and v, i.e.,

$$\texttt{common\_neighbors}(u, v) = \sum_{y \in \mathcal{U}} p_1(u, y) \times p_1(v, y).$$

A consequence of this definition is that all pairs of users who are more than two hops away in the graph are assigned a zero score. Recall that the probability p_1 computed in the first phase is symmetric, i.e., $p_1(u, v) = p_1(v, u)$.

6.1.2 *Probabilistic Katz*

The Katz measure computes a score between two users based on the number of paths existing between them, exponentially damped by length to count short paths more heavily. In other words, a path of length ℓ is weighted by β^{ℓ}, where $0 \leq \beta \leq 1$. We adapt this measure to deal with probabilistic graphs by further weighting each path by its existence probability, which is the product of the probabilities of the links that compose it.

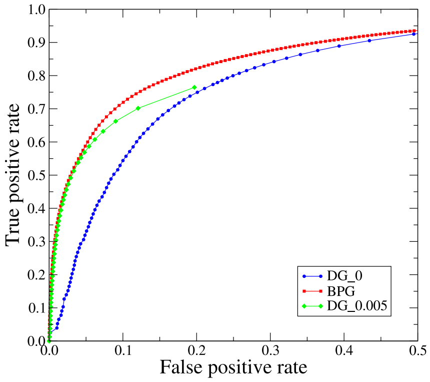

Figure 17: ROC curves for the `common_neighbors`.

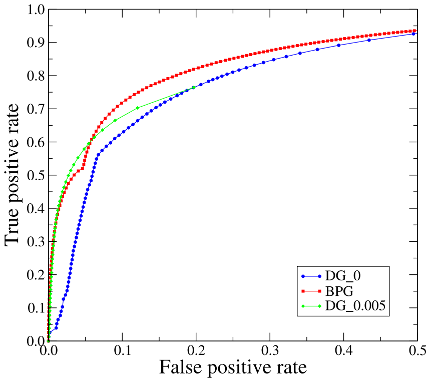

Figure 18: ROC curves for the `katz` feature.

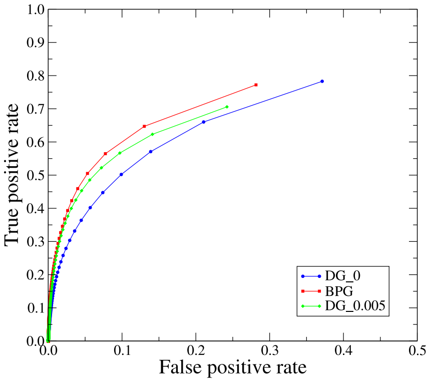

Figure 19: ROC curves for the `rooted_pagerank` feature.

Let $path_{u,v}^{\langle\ell\rangle}$ be the set of paths of length ℓ between u and v in $\mathcal{U}$ and $pathProb(h)$ be the existence probability of a path h. Then, `katz` is computed as

$$\texttt{katz}(u,v) = \sum_{\ell=1}^{\infty} (\beta^{\ell} \times \sum_{h \in path_{u,v}^{\langle\ell\rangle}} pathProb(h)).$$

6.1.3 Probabilistic rooted PageRank

The `rooted_pagerank` feature computes a score between nodes u and v by running rooted PageRank, starting from u. We use an algorithm [10] based on random walks to get an estimation of PageRank scores. Inspired by [18], we adapt it to our probabilistic graph by sampling existing links at each step of a walk using the probabilities in the graph. The walk continues using an existing edge chosen at random as in the classic unweighted version of PageRank. We set a stopping probability α and run W walks for each node. `rooted_pagerank`(u, v) is not null if at least one walk starting at u reaches v. Thus, a user u potentially has a positive score with all other users in his connected component. However, if the number of walks and link probabilities are both very low, this may not be the case. If u and v are not in the same component, then the link (u, v) receives a null score.

6.2 Experimental results

We evaluate the three features by comparing our method based on the bootstrap probabilistic graph (BPG) with two alternative methods. These methods work on deterministic graphs (DG) obtained from BPG, by selecting only the links with a probability higher than a given threshold. Note that since the probability function we designed is monotonic, this is equivalent to a score threshold. It is also worth noting that using a high threshold would compromise seriously the recall of the method. Indeed, a larger threshold implies that fewer links will be selected, which in turn implies a smaller density for the resulting graph and thus a large number of small components. Since graph-based features produce scores only for pairs of users belonging to the same connected component, using a large threshold would give very small recall. Therefore, for the two alternative methods, we use 0 and 0.005 as the thresholds. The methods are accordingly named as DG_0 and DG_0.005. While DG_0 can achieve the same recall as BPG, DG_0.005 can only predict a smaller number of links due to the lower density as discussed above.

Fig. 17 displays the ROC curves for `common_neighbors`. Despite its simplicity, `common_neighbors` performs quite well with BPG as seen from the sharp rise in the true positive rate for predictions with high confidence (early data points). The ROC curve of DG_0 remains always under the curve of BPG. DG_0.005 produces results closer to BPG, but at the price of a lower recall. In the figure, some portion of the curves are not displayed for better visibility of the rest (the last data point is $(0.607, 0.953)$ for DG_0 and BPG curves).

For `katz`, following [15], we set β to 0.005. For scalability reasons, we also set an upper bound on the path length. Since BPG is quite dense, the number of paths becomes important for large values. As β is small and thus long paths have very little weight, the impact on precision is negligible. In our experiments, pairs that are more than two hops away in BPG receive a zero score. Fig. 18 shows the performance of `katz`. BPG still outperforms DG_0 and DG_0.005, but the gap is small, relative to `common_neighbors`.

For experiments on `rooted_pagerank`, we set α to 0.15 and W to 1000. According to the ROC curves shown in Fig. 19, `rooted_pagerank` performs poorly, relative to `common_neighbors` and `katz`, in terms of both coverage of predictions and their quality. Increasing W does not have a significant effect on the result quality, but it increases the coverage of predictions. We believe that `rooted_pagerank` can be efficiently applied on a given user to sort the other users by contact probability. However, the scores obtained through this measure are not comparable across different users. This is due to the fact that PageRank shares a fixed, total score among all users. A user with many potential neighbors is assigned a score lower than a more isolated one, resulting in poor quality predictions at a full graph scale.

Fig. 20 brings together the ROC curves for the `combined` feature and the three phase 2 features (assuming the BPG scenario). The plot demonstrates the gain achieved by phase 2 over the results of phase 1. The performance of `katz` is seen to be very close to `common_neighbors`.

As a representative, Fig. 21 shows the precision–recall plot for the `katz` feature. BPG achieves pretty high precision values relative to DG_0 and DG_0.005 under the equal recall constraint. Despite the difficulty of the problem, precision and recall values achieved by BPG indicate the validity of the proposed method (e.g., at a recall of 1%, we observe a precision around 31%).

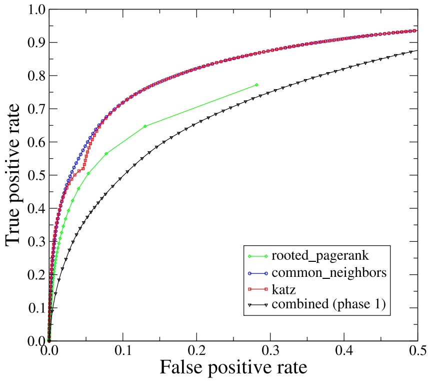

Figure 20: ROC curves for the three phase 2 features and the combined feature from phase 1.

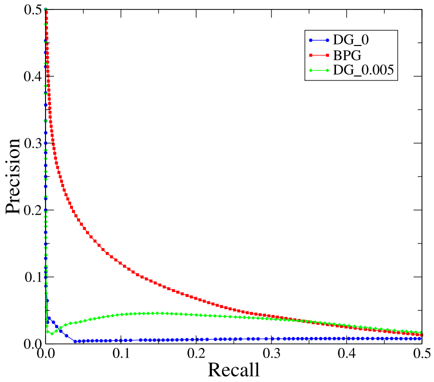

Figure 21: Precision–recall for the katz feature.

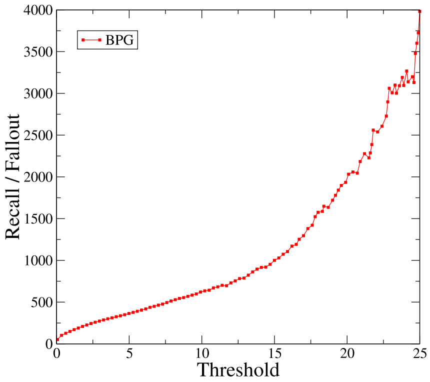

Figure 22: Recall/fallout for the common_neighbors feature.

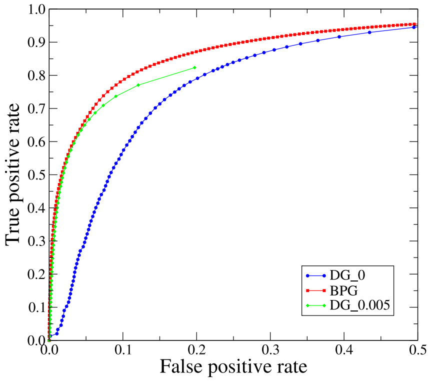

Figure 23: ROC curves for the common_neighbors (friends).

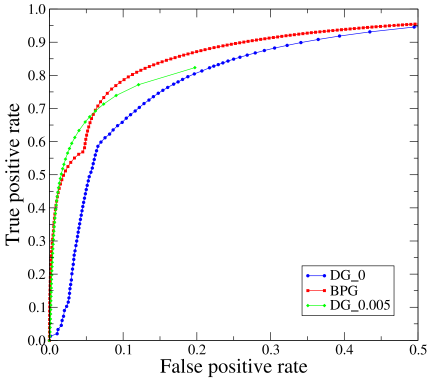

Figure 24: ROC curves for the katz feature (friends).

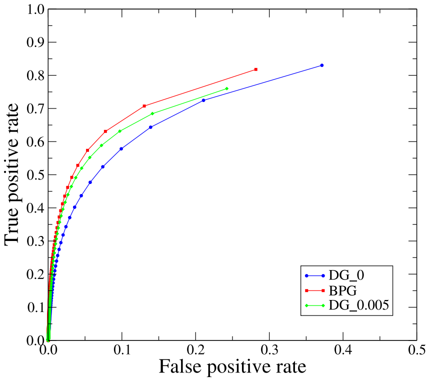

Figure 25: ROC curves for the rooted_pagerank feature (friends).

As another representative, Fig. 22 shows the recall/fallout curve for the common_neighbors feature. BPG is able to leverage the low probability edges to increase recall without losing precision while non-probabilistic approaches either prune this noise and lose recall or use these edges but suffer from low precision.

We have also conducted experiments using friend links instead of contacts. We do not report results for family links due to space limitations. Figs. 23, 24, and 25 show performance of common_neighbors, katz, and rooted_pagerank, respectively, in predicting friend links. For all three techniques, higher prediction quality is achieved relative to contact links.

In this work, we have preferred not to combine the predictive power of the three features of phase 2 (e.g., by means of machine learning techniques). The rationale behind this choice is two-fold. First, as we have explained before, rooted_pagerank is not suitable for prediction at a full graph scale. Second, common_neighbors can be seen as a special case of katz in which the maximum path length is 2 and β is set to 1. Therefore, we expect only little improvement in prediction accuracies by further combinations.

7. CONCLUSIONS AND FUTURE WORK

We presented the cold start link prediction problem and a two-phase method that enables link prediction in the absence of a social network. The first phase of the proposed method generates a bootstrap probabilistic graph using any available feature while the second phase applies various link prediction algorithms to this probabilistic graph. We tested our approach over a data set obtained from Flickr, by using group memberships as the only available information.

For the sake of generality, we applied our method to interest groups, a very simple and common kind of information in social networks. Thus, the features we present can be applied to other networks. Obviously, as more information is available, higher prediction accuracies can be achieved.

In the context of Flickr, we might use information that is more specific to photography in order to improve prediction performance. For instance, in [20], Singla and Weber study the impact of the social network on camera brands of Flickr users. We could leverage such information to create more accurate predictors. Similarly, as observed in [4], information diffusion often follows the social network. This is known as the social cascade phenomena and can be observed in Flickr when users favorite others' pictures and post comments about them. If this information is available, it can be used as a bootstrap feature to generate a probabilistic social graph that matches the observed social cascade.

As pointed in Section 2, we believe that our method can be applied as an attack against link privacy in social networks. Determining to which extent our approach can be combined with existing attacks to improve the predictive power of publicly available attributes is worth future research.

In this work, we have adapted the graph-theoretic algorithms in [15] to probabilistic graphs. Potamias et al. [18] introduces different measures of distances in probabilistic graphs and present algorithms to compute k-nearest neighbor queries. We believe that probabilistic graphs are a powerful tool and designing algorithms to extract their characteristics can create new approaches also to other research problems.

8. REFERENCES

[1] L. Adamic and E. Adar. Friends and neighbors on the web. *Soc. Networks*, 25(3), 2003.

[2] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proc. of the Sixteenth Int. Conf. on World Wide Web (WWW'07)*, pages 181–190, 2007.

[3] J. Baumes, M. Goldberg, M. Hayvanovych, M. Magdon-Ismail, W. Wallace, and M. Zaki. Finding hidden group structure in a stream of communications. In *Proc. of the IEEE Int. Conf. on Intelligence and Security Informatics (ISI'06)*, pages 201–212, 2006.

[4] M. Cha, A. Mislove, and K.P. Gummadi. A measurement-driven analysis of information propagation in the Flickr social network. In *Proc. of the Eighteenth Int. Conf. on World Wide Web (WWW'09)*, pages 721–730, 2009.

[5] R. Chellappa and A. Jain. Markov random fields: theory and application. *Boston Academic Press*, 1993.

[6] A. Clauset, C. Moore, and M. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453:98–101, 2008.

[7] P. Domingos and M. Richardson. Markov Logic: A unifying framework for statistical relational learning. In *Proc. of the ICML'04 Workshop on Statistical Relational Learning and its Connections to Other Fields*, pages 49–54, 2004.

[8] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proc. of the Seventh ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'01)*, pages 57–66, 2001.

[9] N. Eagle and A. Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[10] D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Journal of Internet Mathematics*, 2(3):333–358, 2005.

[11] Jing Gao and Pang-Ning Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Proc. of the Sixth IEEE Int. Conf. on Data Mining (ICDM'06)*, pages 212–221, 2006.

[12] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of link structure. *Machine Learning*, 3:679–707, 2003.

[13] Michael Hay, Gerome Miklau, David Jensen, Donald F. Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114, 2008.

[14] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proc. of the Ninth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'03)*, pages 137–146, 2003.

[15] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7):1019–1031, 2007.

[16] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *Proc. of the Thirtieth IEEE Symp. on Security and Privacy*, pages 173–187, 2009.

[17] J. O'Madadhain, J. Hutchins, and P. Smyth. Prediction and ranking algorithms for event-based network data. *ACM SIGKDD Exploration Newsletter*, 7(2):23–30, 2005.

[18] Michalis Potamias, Francesco Bonchi, Aristides Gionis, and George Kollios. k-nearest neighbors in uncertain graphs. In *Proc. of the VLDB, the Thirty-Sixth Int. Conf. on Very Large Databases (PVLDB 2010)*, volume 3, 2010.

[19] Foster J. Provost, Tom Fawcett, and Ron Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth Int. Conf. on Machine Learning (ICML'98)*, pages 445–453, 1998.

[20] A. Singla and I. Weber. Camera brand congruence in the Flickr social graph. In *Proc. of the Second ACM Int. Conf. on Web Search and Data Mining (WSDM'09)*, pages 252–261, 2009.

[21] B. Taskar, M.F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Neural Information Processing Systems*, volume 15, 2003.

[22] W.M.P. Van Der Aalst, H.A. Reijers, and M. Song. Discovering social networks from event logs. *Computer Supported Cooperative Work*, 14(6):549–593, 2005.

[23] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proc. of the Eighteenth Int. Conf. on Machine Learning (ICML'01)*, pages 609–616, 2001.

[24] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proc. of the Eighth ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'02)*, pages 694–699, 2002.

[25] Jian Zhang and Yiming Yang. Probabilistic score estimation with piecewise logistic regression. In *Proc. of the Twenty-first Int. Conf. on Machine Learning (ICML'04)*, page 115, 2004.

[26] E. Zheleva and L. Getoor. To join or not to join: the illusion of privacy in social networks with mixed public and private user profiles. In *Proc. of the Eighteenth Int. Conf. on World Wide Web (WWW'09)*, pages 531–540, 2009.