Supervised-Learning Link Recommendation in the DBLP co-authoring network

Gabriel P. Gimenes, Hugo Gualdron, Thiago R. Raddo, Jose F. Rodrigues Jr.
Instituto de Ciencias Matematicas e de Computacao
Universidade de Sao Paulo
Avenida Trabalhador Sao-carlense, 400 - Centro, Sao Carlos, SP 13560-970, Brazil
Email: ggimenes@icmc.usp.br, gualdron@icmc.usp.br, raddo@usp.br, junio@icmc.usp.br

***Abstract*—Currently, link recommendation has gained more attention as networked data becomes abundant in several scenarios. However, existing methods for this task have failed in considering solely the structure of dynamic networks for improved performance and accuracy. Hence, in this work, we present a methodology based on the use of multiple topological metrics in order to achieve prospective link recommendations considering time constraints. The combination of such metrics is used as input to binary classification algorithms that state whether two pairs of authors will/should define a link. We experimented with five algorithms, what allowed us to reach high rates of accuracy and to evaluate the different classification paradigms. Our results also demonstrated that time parameters and the activity profile of the authors can significantly influence the recommendation. In the context of DBLP, this research is strategic as it may assist on identifying potential partners, research groups with similar themes, research competition (absence of obvious links), and related work.**

I. Introduction

In the last decade, advances in the World Wide Web have led to improved mechanisms for users to interact and to share experiences, both for the general public and for corporations (industry and academy). Most of these social interactions are dynamics, receiving or loosing vertices and edges [1]. The dynamism of networks is itself a source of valuable, though not obvious, information; understanding such dynamism involves several variables that pose a complex problem [2]. This problematic has been dealt by several subfields, as graph theory, complex networks, and social network analysis (SNA); similar areas that differ by some subtleties. For the rest of this paper, we pick SNA as our area of concentration.

SNA refers to techniques and paradigms among which *link recommendation* (also known as link prediction) is of special interest [3]. Link recommendation refers to algorithmically foreseeing/identifying new associations between the existent vertices of a network; it is based on the assumption that the past and the present behavior of the net can indicate what may happen in the future. Such mechanism helps, for example, in problems like forecasting the behavior of a terrorist network [4]; in biology, it is used to identify associations that, otherwise, would demand intense experimentation to be discovered [2]; and, also, it is used in several kinds of social networking to expand the interaction among individuals.

One of the main paradigms of link recommendation is machine supervised learning, which is based on three different approaches [5]: the topological structure of the network, the semantic similarity among the properties of the vertices, and the description of the network behavior by means of probabilistic models [6]. Specifically, in this work, we use the topological structure of the network in order to recommend links by considering eight edge-oriented metrics based on neighborhood, path distance, and clustering properties [7]. We use these metrics in combination with a vast set of machine learning algorithms, presenting a comparative study that evaluates their relative efficiency.

We run experiments over the Digital Bibliography & Library Project (DBLP), a public database of Computer Science publications that defines a co-authorship graph. Link recommendation, in this sense, refers to identifying potential co-authoring (research collaboration) given previous and current co-authoring patterns. In the context of DBLP, the link recommendation that we propose is useful in identifying potential partners, research groups with similar themes, research competition (absence of obvious links), and related work. A recommended link does not necessarily mean that the correspondent authors should work together; rather, it is an indication that they should pay attention one to each other. For this task, we use machine learning classification algorithms; in our dynamic problem setting, the pairs of vertices are classified as positive or negative according to the edges that are created, or not, between their respective vertices. We considered techniques [8] J48, Naïve Bayes, Multilayer Perceptron, Bagging, and Random Forest, all of them available in the Weka framework, developed by the University of Waikato [9].

Specifically, our contribution is the use of supervised machine learning classification in the task of link recommendation in temporal graphs, proposing a systematic approach for computation and evaluation considering the time of publications and the profiles (number of publications) of the authors.

Following, we present works related to our proposal in Section II and the formalization of our methodology in Section III. In Section IV, we describe a vast set of experiments whose results are discussed in Section V. Section VI concludes the paper.

II. Related works

Liben-Nowell and Kleinberg [10] present one of the most important works on link prediction/recommendation; the authors formalize the link prediction problem as the question of whether it is possible to infer which new interactions are likely to occur given a snapshot of a social network. They

978-1-4799-2736-4/14/$31.00 ©2014 IEEE

use an unsupervised learning approach and, by calculating similarity measures, they create a ranking by descending order of similarity. The ranking is then used to recommend the interactions that are likely to occur, in such a way that the higher the rank, the more likely the interaction is to appear in the future. The authors also acknowledge that the results of using such ranking are not satisfactory and propose that other approaches should be explored. In light of this matter, we take a different direction by considering supervised learning methods.

Recently, Aiello *et al.* [11] describe how friends that have similar profiles (homophily) tend to get interconnected. In their study, the authors consider the groups to which the users belong, and the annotations (tags) of the users, among other features. With these features, the authors calculate the similarity between users, proposing a similarity threshold to state whether two users are to define a connection, or not. Regardless of its significative results, this study extrapolates the topological information of the network; it relies on information that, often, is not available or is not well-defined. This same limitation is faced by Brandao *et al.* [12] and Lim *et al.* [13].

Clauset *et al.* [14] present the link prediction problem based on a hierarchical analysis approach. Their method not only provides interesting results for link recommendation, but also explains many characteristics of the network. Despite its results, their work demands that the graph representation be hierarchically partitioned, a requirement that adds up complexity and processing demands; the same characteristic is observed in Guo *et al.* [15]. In this work, we use a simpler, yet efficient, method to accomplish link recommendation with similar potential.

Zhou *et al.* [16] firstly evaluated how metrics that are exclusively topological can be used for link recommendation. In their work, the authors compare the performance of local and global metrics. They conclude that local metrics, as used in our work, is the better choice. However, different from our approach, they consider the sole metrics instead of their combination for improved performance. In [17], Papadimitriou *et al.* use global graph processing in order to recommend friends in social networks; although they achieved good results, the technique is computationally expensive.

Menon *et al.* [18] analyses the effectiveness of matrix factorization techniques for the structural link prediction problem. They discuss a novel mechanism to allow their model to overcome the imbalance characteristic using the idea of optimizing for a ranking loss. Their results show good performance related to the imbalance overcome. Finally the authors suggest that the model can be used in conjunction with other approaches to further improve the technique.

Sa and Prudencio [19] addresses link prediction by means of classification algorithms and edge-oriented metrics; although their work evaluates the role of edge weighting in the task of foreseeing new links, they do not consider the multiple parameters that influence the problem. The same approach is used by Herman *et al.* [20]. With a different orientation, we obtain better recommendation results that are discussed in light of empirical experiments considering a wider spectrum of configurations.

In this work, we present a methodology that extends former proposals by defining a topology-exclusive link recommendation, with low complexity and processing cost, considering the combination of multiple metrics in a comparative context. We perform experiments that reveal how the different parameters of the link recommendation problem affect diverse classification algorithms. Our result is a method that reaches superior rates ($\approx 90\%$) of recommendation accuracy and the same time that it indicates what are the most effective classification algorithms for link prediction and recommendation.

III. Methodology

In terms of a co-authorship graph, we have $G = (V, E)$, where V is the set of vertices (authors), E is the set of edges, so that each edge $e = (u, v) \in E$ represents the co-authoring between authors u and v. Also, since we are worried about the dynamic behavior of the network, each edge e has a time label t that states when the edge was created; from the DBLP dataset, we are considering the snapshot $1974 \leq t \leq 2007$. In this work, given a snapshot of a network at time t', we are interested in recommending the edges that most likely should/could exist in time t'', $t' < t''$; but that, for some reason, are still latent.

For link recommendation, we shall use the past and the present behavior to recommend prospective new edges. Therefore, it is necessary to break the set E into two disjoint subsets according to the time labels, defining past and present intervals of time. The first interval – the past, is delimited by two moments t_1 and t_2, $t_1 < t_2$, and is used as the training interval, which we refer to as the induced subgraph $G[t_1, t_2] = (V, E_{past})$. The second interval – the present, is delimited by other two moments, t_3 and t_4, $t_3 < t_4$, which we refer to as $G[t_3, t_4] = (V, E_{pres})$.

Metric	Definition
Number of common neighbors (CN)	$CN(x,y) = \lvert\Gamma(x) \cap \Gamma(y)\rvert$
Jaccard's coefficient (JC)	$JC(x,y) = \frac{\lvert\Gamma(x)\cap\Gamma(y)\rvert}{\lvert\Gamma(x)\cup\Gamma(y)\rvert}$
Preferential attachment (PA)	$PA(x,y) = \lvert\Gamma(x)\rvert * \lvert\Gamma(y)\rvert$
Adamic-Adar coefficient (AA)	$AA(x,y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{\log\lvert\Gamma(z)\rvert}$
Path distance (PD)	Shortest path between x and y
Resource allocation index (RA)	$RA(x,y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1}{\lvert\Gamma(z)\rvert}$
Local path (LP)	$LP(x,y) = \left\lvert paths^{(2)}_{x,y}\right\rvert + e * \left\lvert paths^{(3)}_{x,y}\right\rvert$
Local clustering coefficient (CC)	$ANCC(x,y) = cc(x) + cc(y)$

TABLE I. Metrics used in this work.

Given a co-authorship graph G, link recommendation becomes a two-class problem to be treated with classification techniques – positive instances refer to pairs of vertices (potential links) that could be connected in the future, and negative ones refer to the other case. In order to be classified, the pairs must be represented as vectors of numbers; in this case, each dimension of the vectors is a metric. We use edge-oriented metrics calculated straightly from the topological information of the network. The advantage of such metrics is their domain-independence because they can be calculated from any kind of network. In Table I, we present the metrics that we use – for these metrics, we consider the following definitions: let $\Gamma(x)$ be the set of neighbors of vertex x; $|\Gamma(x)|$ be the degree of x; and $e(x, y)$ be the non-directed edge between x and y.

The classifiers we employ – J48, Naïve Bayes, Multilayer

Perceptron, Bagging, and Random Forest, see Table II, learn from the past of the network, which is represented as pre-classified vectors corresponding to the pairs of vertices; these pairs are classified according to what is observed in the present of the network. In our experiments, the classifiers use 10% of the present information to pre-classify the vectors (pairs), using this data to lean and recommend the remaining 90% of the present data. Therefore, the accuracy corresponds to how precise the recommendations match the known 90% of the present. We use the classical 10-fold cross-validation, that is, we perform the same classification 10 times, each one using only 10% of what is already known about the data. The final performance is given by the average of the results.

Classifier	Details	Parameters
J48	Decision tree algorithm	-C 0.25 -M 0.2
Naïve Bayes (NB)	Probabilistic	
Multilayer Perceptron (MLP)	Neural network	-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Bagging	Meta-classifier	-P 100 -S 1 -I 10 -W ADTree -B 10 -E -3
Random Forest (RF)	Combination of decision trees	-I 10 -K 3 -S 1

TABLE II. CLASSIFIERS USED IN THIS WORK.

Figure 1 presents the general flow of the link prediction task, (1) shows the extraction of the topological features from the DLBP network, (2) represents the training of the classifiers that we use, (3) is the classification process itself in which the new links are recommended, number (4) presents the evaluation step where measures like AUC are used to quantify the efficiency of the classifiers, finally (5) consists of the analysis of the results obtained and can be considered the end of the task, from the results knowledge can be obtained in such a way that it is possible to learn from the behavior of the network and use it to predict and recommend new links.

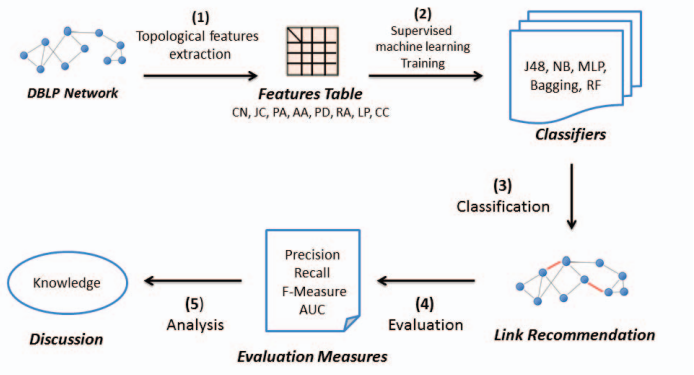

Fig. 1. Link recommendation task overview.

In dynamic graphs, there are vertices that remain active throughout the life span of the network, and there are vertices that simply pop out and become inactive right after. Therefore, another aspect is which vertices to consider for recommendation. To this end, we use the number of coauthorings (the degree) of the authors as criterion; we consider only the authors that have *at least* k *coauthorings* in both the past and in the present intervals. The set of vertices that satisfies the criterion for a given k is denoted *core of authors*. In our experiments, we discuss parameter k empirically by evaluating different values for it.

IV. EXPERIMENTS

We have run the link recommendation experiment considering three different time settings; using classifiers J48, Naïve Bayes, Multilayer Perceptron, Bagging, and Random Forest; and considering $k \in \{1, 2, 4, 6, 8\}$.

Time settings

For the DBLP snapshot, whose years range from 1974 to 2007, we consider the following intervals for the past and the present of the network – see Section III.

- First time setting; past: G[1995, 2005], present: G[2006, 2007] – long past/short present;
- Second time setting; past: G[1990, 1999], present: G[2000, 2004] – long past/long present;
- Third time setting; past: G[1995, 1999], present: G[2000, 2004] – short past/short present.

Each time setting had a different *core of authors* depending on parameter k. In the following, we analyze the resulting core for each setting considering $k \in \{1, 2, 4, 6, 8\}$. For the first time setting, G[1995, 2005] – G[2006, 2007], the value of $k = 1$ induced a core reduction of 98%; from 512,929 authors to only 7,583 – see Table III. The reduction eliminated authors that do not have at least one edge either in the past, and/or in the present, as defined in Section III. For $k = 2$, the reduction was still significative, 77% less authors; the same holds for $k = 4$, with a reduction of 51%. For higher values, $k = 6$ and $k = 8$, the reduction was less intense – around 10%. The same behavior is observed for the second and third time settings. The observations indicate that the majority of the authors are eventual researchers with one or two publications, and also that there are very few researchers with a constant and high (> 4) number of publications. These results suggest that the value of k must be between 2 and 4.

	Vertices	Edges
$G[1995, 2005]$	512929	1622662
$G[2006, 2007]$	512929	224318
k		
1	7583	25781
2	1714	9458
4	826	5937
6	760	5681
8	756	5654
$G[1990, 1999]$	266877	676431
$G[2000, 2004]$	266877	156777
k		
1	1056	6627
3	569	4940
5	530	4758
7	529	4751
$G[1995, 1999]$	175671	401803
$G[2000, 2004]$	175671	141902
k		
1	869	4495
3	387	2748
5	365	2657
7	365	2657

TABLE III. DBLP TIME SETTINGS AND CARDINALITY OF THE *core of authors* FOR $k \in \{1, 2, 4, 6, 8\}$.

Processing

We used and extended the Stanford Network Analysis Project (SNAP) library to calculate the metrics needed in our methodology. For each time setting and value of k, in a total of 15 configurations, we calculated the topological metrics considering

only the vertices that satisfy to the *core of authors* definition. Algorithm 1 shows the pseudo code used to calculate the metrics in our experiment.

Algorithm 1: *Algorithm to calculate the metrics of a co-authoring network.*

```
Input: G: graph adjacency list
Input: t1: start of the first interval
Input: t2: start of the second interval
Input: k: Core of authors parameter
Input: N: number of pairs to consider
CalculateMetrics(Graph G, int t1, int t2, int k, int N)
begin
    Graph G_it1 = InduceIntervalSubgraph(G, t1,
    t2 − 1);
    Graph G_it2 = InduceIntervalSubgraph(G, t2,
    G.getLastYear());
    for for each v ∈ (G_it1 ∩ G_it2) do
        if ((G_it1.getDegree(v) < k) OR
        (G_it2.getDegree(v) < k)) then
            G_it1.delete(v);
        end
        ClassifyEdgesOf(v, G_it1, G_it2); //Edges
        appearing in G_it2 are positive
    end
    for i = 1 to N do
        e =
        G_it1.graphSearchRandomPositiveEdge();
        WriteFile(CalculateMetricsFor(e));
        e =
        G_it1.graphSearchRandomNegativeEdge();
        WriteFile(CalculateMetricsFor(e));
    end
end
```

Our algorithm receives 4 parameters, besides the graph that represents the network. The first two define the past and the present of a time setting, the third one refers to parameter k, and the last one sets the number of edges to consider in the experimentation; since the number of possible edges (pairs) is very high ($|V|(|V|-1)/2$), we randomly choose the pairs using graph search; that is, by traversing existing edges, or by jumping to random vertices.

Analysis of the metrics

Following, we empirically analyze the metrics calculated with algorithm 1 for the first time setting. In Figure 2, we present the distribution of the values of each metric considering 400 positive examples, and 400 negative examples. One can see that metrics Number of common neighbors (a), Jaccard's coefficient (b), Preferential attachment (c), Adamic-Adar coefficient (d), Resource allocation index (f), and Local path (g) are very sparse in the sense that the majority of the vertices produced value 0. This is because all of them are strongly related to the common neighbors of the vertices that define a potential edge (link), what is sensible to the density of the graph. Despite that, we noticed that excluding any of these metrics would lead to a significant drop in the performance. This is because the right side of the distributions – the values different from 0 – is composed of values that spam to a wide diversification, conferring to the classifiers more discriminant power. Besides that, the potential of the classifiers is further improved by the information provided by metrics Path distance (b) and Local clustering coefficient (h) that, as we can see in the figure, have a normal-like distribution.

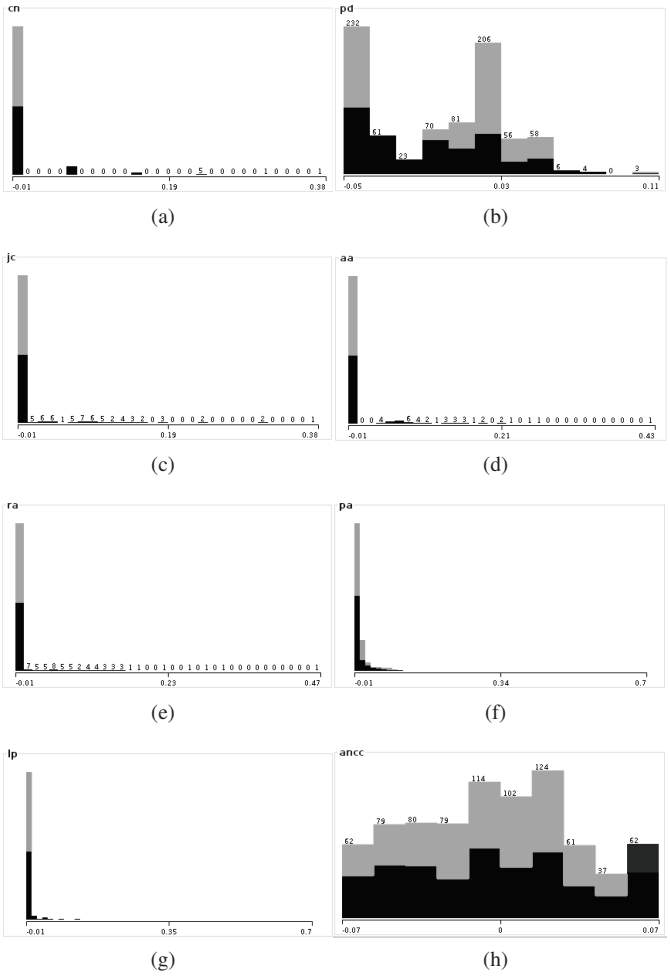

Fig. 2. Distribution of the values of the metrics used in this work; dark gray for positive, and light gray for negative examples. Number of common neighbors (a), Path distance (b), Jaccard's coefficient (c), Adamic-Adar coefficient (d), Resource allocation index (e), Preferential attachment (f), Local path (g), and Local clustering coefficient (h).

V. Results and discussion

The experimental setting allowed us to observe the influence of the parameters (time setting, value of k, and classifier) of the methodology. We present the results in Tables IV, V, and VI; and in the corresponding Figures 3, 4, and 5. Each table/figure corresponds to one time setting, including data for each classifier and k value, and presents evaluation measurements Precision, Recall, F-Measure, and Area Under Curve (AUC) corresponding to the Receiver Operating Characteristic (ROC). The numbers refer to the average results achieved with a 10-fold cross-validation. In the evaluation, the higher the values, the more trustworthy are the recommendations. Even the recommendations that did not match a future link are of interest; they can be interpreted as potential interactions that have not occurred; or interpreted as concurrent authors, in the case of rivalry.

In the three tables, the Random Forest (RF) classifier presented the highest scores in the three time settings, for all the values of k, and considering the 5 classifiers. Classifier RF

uses random combinations of the metrics to produce multiple decision trees that will be merged by voting; that is, the classification given by the bigger number of trees is the final classification. It generates combinations that disregard specific metrics one at a time, a course of action that allowed RF to be less sensible to the peculiarities of individual metrics. Empirically, we verified that the classification was very effective ($\approx 90\%$ accuracy), especially if compared to the other classifiers that necessarily depend on all the metrics.

In contrast to classifier RF, classifiers Multilayer Perceptron and Naïve Bayes demonstrated to be inadequate for the link recommendation task. In Figures 3, 4, and 5, it is evident that their recommendation potential is irregular and pronouncedly inferior. We suspect that the number of metrics and their strong non-linear separability posed hard challenges to these classifiers that, differently from the other three, are not decision-tree based. On the other hand, the fact that there are only two classes possibly led to the improved performance of J48, Bagging, and Random Forest.

The results also demonstrated that for $k \in \{2, 6\}$ the link recommendation had a higher performance. This observation corroborates our initial guess, according to which we should have better results with profiles of around 4 publications per period (past and present), discarding left-most and right-most outliers. These results are observed for the three time settings, but they are more evident for the third time setting – Table IV and Figure 3. It is an indication that the link recommendation practice has a bigger potential for short past and short present configurations, situations in which the memory of the system is more recent and can better explain the near future.

k	Classifier	PRECISION	RECALL	F-MEASURE	AUC
1	J48	0.723	0.706	0.7	0.764
	NB	0.741	0.585	0.505	0.626
	MLP	0.562	0.555	0.541	0.593
	Bagging	0.809	0.8	0.798	0.887
	RF	**0.877**	**0.868**	**0.867**	**0.939**
2	J48	0.787	0.759	0.753	0.817
	NB	0.777	0.598	0.52	0.648
	MLP	0.628	0.618	0.61	0.639
	Bagging	0.84	0.83	0.829	0.913
	RF	**0.914**	**0.903**	**0.902**	**0.977**
4	J48	0.852	0.845	0.844	0.87
	NB	0.773	0.585	0.499	0.704
	MLP	0.715	0.714	0.713	0.735
	Bagging	0.846	0.841	0.841	0.925
	RF	**0.917**	**0.913**	**0.912**	**0.974**
6	J48	0.827	0.771	0.761	0.79
	NB	0.778	0.601	0.526	0.727
	MLP	0.695	0.679	0.672	0.74
	Bagging	0.844	0.83	0.828	0.913
	RF	**0.897**	**0.888**	**0.887**	**0.972**
8	J48	0.861	0.839	0.836	0.867
	NB	0.786	0.626	0.566	0.741
	MLP	0.725	0.719	0.717	0.785
	Bagging	0.883	0.866	0.865	0.94
	RF	**0.914**	**0.908**	**0.907**	**0.971**

TABLE IV. RESULTS FOR TIME SETTING $G[1995, 2005], G[2006, 2007]$.

The results presented in Tables IV, V, and VI indicate around 90% of efficiency and accuracy. This rate is comparable to the works presented in Section II, being superior for DBLP and for similar datasets.

VI. CONCLUSIONS

We have touched the problem of link recommendation in the context of research collaboration over the DBLP dataset.

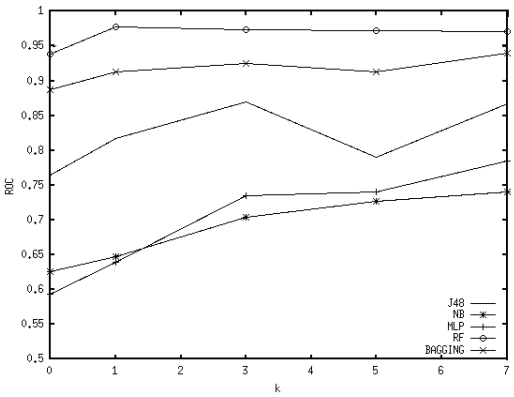

Fig. 3. AUC visualization of the data generated for the first time setting – $G[1995, 2005], G[2006, 2007]$.

k	Classifier	PRECISION	RECALL	F-MEASURE	AUC
1	J48	0.774	0.751	0.746	0.804
	NB	0.765	0.619	0.558	0.675
	MLP	0.609	0.605	0.602	0.642
	Bagging	0.815	0.803	0.801	0.886
	RF	**0.871**	**0.86**	**0.859**	**0.937**
2	J48	0.801	0.784	0.781	0.837
	NB	0.801	0.784	0.781	0.837
	MLP	0.562	0.561	0.559	0.613
	Bagging	0.837	0.828	0.826	0.895
	RF	**0.896**	**0.888**	**0.887**	**0.959**
4	J48	0.865	0.85	0.848	0.893
	NB	0.779	0.603	0.528	0.711
	MLP	0.703	0.698	0.696	0.761
	Bagging	0.866	0.859	0.858	0.931
	RF	**0.928**	**0.921**	**0.921**	**0.982**
6	J48	0.768	0.746	0.741	0.806
	NB	0.744	0.618	0.561	0.74
	MLP	0.739	0.725	0.721	0.764
	Bagging	0.856	0.845	0.844	0.924
	RF	**0.906**	**0.899**	**0.898**	**0.969**
8	J48	0.823	0.813	0.811	0.856
	NB	0.782	0.613	0.544	0.772
	MLP	0.75	0.75	0.75	0.823
	Bagging	0.86	0.854	0.853	0.937
	RF	**0.928**	**0.923**	**0.922**	**0.977**

TABLE V. RESULTS FOR TIME SETTING $G[1990, 2000], G[2001, 2004]$.

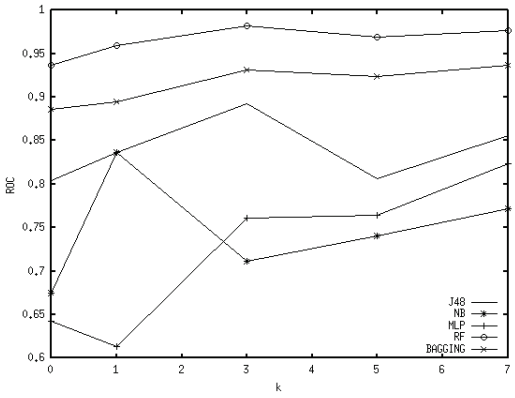

Fig. 4. AUC visualization of the data generated for the second time setting – $G[1990, 2000], G[2001, 2004]$.

Our technique is based on the combination of eight topological metrics – Number of common neighbors, Jaccard's coefficient, Preferential attachment, Adamic-Adar coefficient, Path distance, Resource allocation index, Local path, and Local clustering coefficient, that are used by machine learning classifiers in the task of latent link identification. We experi-

k	Classifier	PRECISION	RECALL	F-MEASURE	AUC
1	J48	0.84	0.813	0.809	0.834
	NB	0.791	0.64	0.586	0.752
	MLP	0.749	0.741	0.739	0.786
	Bagging	0.851	0.84	0.839	0.919
	RF	**0.891**	**0.888**	**0.887**	**0.957**
2	J48	0.857	0.838	0.835	0.855
	NB	0.785	0.655	0.611	0.762
	MLP	0.718	0.715	0.714	0.777
	Bagging	0.854	0.841	0.84	0.916
	RF	**0.925**	**0.915**	**0.915**	**0.971**
4	J48	0.883	0.878	0.877	0.91
	NB	0.781	0.626	0.567	0.79
	MLP	0.77	0.768	0.767	0.851
	Bagging	0.872	0.865	0.864	0.922
	RF	**0.93**	**0.924**	**0.923**	**0.974**
6	J48	0.843	0.824	0.821	0.877
	NB	0.764	0.64	0.592	0.767
	MLP	0.799	0.783	0.779	0.804
	Bagging	0.848	0.84	0.839	0.924
	RF	**0.917**	**0.908**	**0.907**	**0.968**
8	J48	0.823	0.815	0.814	0.86
	NB	0.753	0.62	0.563	0.761
	MLP	0.696	0.695	0.695	0.789
	Bagging	0.853	0.849	0.848	0.914
	RF	**0.921**	**0.914**	**0.913**	**0.958**

TABLE VI. RESULTS FOR TIME SETTING $G[1995, 1999], G[2000, 2004]$.

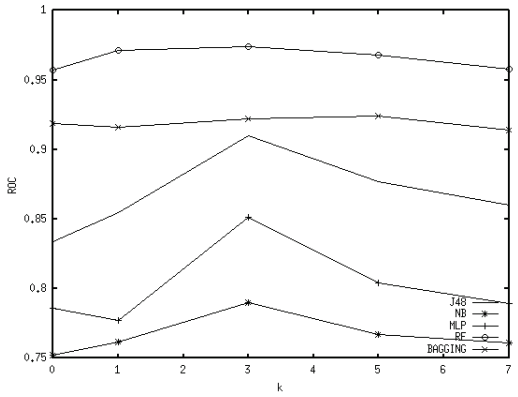

Fig. 5. AUC visualization of the data generated for the third time setting – $G[1995, 1999], G[2000, 2004]$.

mented with five classification algorithms – J48, Naïve Bayes, Multilayer Perceptron, Bagging, and Random Forest, whose binary output, positive or negative, states whether a given pair of nodes will (should) define a new link. In the context of DBLP, the recommended links answer for potential partners, related research groups, and, even, research competition.

Our results achieved recommendation accuracy rates similar or superior ($\approx 90\%$) than those of related works at smaller complexity and processing cost. We also demonstrated that time parameters can alter the results of the recommendation – in our experiments, DBLP was sensible to shorter periods of time (past and present); evidencing the short memory and the strong dynamism of the academic community. Another important aspect was the need to filter out the authors on a neighborhood basis; that is, in DBLP, one cannot work on link recommendation considering the entire set of authors, which come and go very often. In this sense, we used the concept of *core of authors*; a critical subset of authors empirically calculated. Finally, we extensively evaluated the set of classification algorithms considering Precision, Recall, F-Measure, and AUC-ROC. We found that decision trees work better than neural networks and Naïve Bayes classification, and, also, that Bagging and Random Forest can further improve the results.

ACKNOWLEDGEMENTS

This work received support from Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq-560104/2010-3), Fundacao de Amparo a Pesquisa do Estado de Sao Paulo (FAPESP-2011/13724-1, 2013/03906-0, 2013/10026-7) and Coordenacao de Aperfeicoamento de Pessoal de Nivel Superior (Capes).

REFERENCES

[1] E. M. Jin, M. Girvan, and M. E. J. Newman, "The structure of growing social networks," *Phys. Rev. E*, no. 64, p. 046132, 2001.

[2] R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla, "New perspectives and methods in link prediction," in *SIGKDD*. ACM, 2010, pp. 243–252.

[3] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *SIGKDD*. ACM, 2011, pp. 1100–1108.

[4] M. A. Hasan, V. Chaoji, S. Salem, and M. Zaki, "Link prediction using supervised learning," in *Siam DM workshop on Link Analysis, Counterterrorism and Security*, 2006, pp. 1–10.

[5] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A*, vol. 390, no. 6, pp. 1150–1170, 2011.

[6] P. S. Yu, J. Han, and C. Faloutsos, *Link Mining: Models, Algorithms, and Applications*, 1st ed. Springer, 2010.

[7] T. Murata and S. Moriyasu, "Link prediction of social networks based on weighted proximity measures," in *Conference on Web Intelligence*. IEEE Computer Society, 2007, pp. 85–88.

[8] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining - Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 2011.

[9] J. Breslin and S. Decker, "The future of social networks on the internet: The need for semantics," *IEEE Internet Computing*, vol. 11, no. 6, pp. 86–90, 2007.

[10] D. Liben-Nowell and J. Kleinberg, "The link prediction problem for social networks," in *CIKM*. ACM, 2003, pp. 556–559.

[11] L. Aiello, A. Barrat, R. Schifanella, C. Cattuto, B. Markines, and F. Menczer, "Friendship prediction and homophily in social media," *ACM Trans. Web*, vol. 6, no. 2, pp. 9:1–9:33, 2012.

[12] M. Brandao, M. Moro, G. Lopes, and J. Oliveira, "Using link semantics to recommend collaborations in academic social networks," in *World Wide Web*, 2013, pp. 833–840.

[13] E.-P. Lim, D. Correa, D. Lo, M. Finegold, and F. Zhu, "Reviving dormant ties in an online social network experiment," in *Conference on Weblogs and Social Media*. AAAI Press, 2013, pp. 361–369.

[14] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," *Nature*, vol. 453, pp. 98–101, 2008.

[15] F. Guo, Z. Yang, and T. Zhou, "Predicting link directions via a recursive subgraph-based ranking," *Physica A*, vol. 392, no. 16, pp. 3402 – 3408, 2013.

[16] Y.-C. Z. T. Zhou, L. Lu, "Predicting missing links via local information," *The Physics Approach To Risk: Agent-Based Models and Networks*, vol. 71, pp. 623–630, 2009.

[17] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos, "Fast and accurate link prediction in social networking systems," *Journal of Systems and Software*, vol. 85, no. 9, pp. 2119 – 2132, 2012.

[18] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 437–452.

[19] H. de Sa and R. Prudencio, "Supervised link prediction in weighted networks," in *Joint Conference on Neural Networks*, 2011, pp. 2281–2288.

[20] T. Herman, M. Monsalve, S. Pemmaraju, P. Polgreen, A. Segre, D. Sharma, and G. Thomas, "Inferring realistic intra-hospital contact networks using link prediction and computer logins," in *Social Computing*, 2012, pp. 572–578.