Journal of Intelligent & Fuzzy Systems 34 (2018) 2667–2678
DOI:10.3233/JIFS-17770
IOS Press

A new study of using temporality and weights to improve similarity measures for link prediction of social networks

Farshad Aghabozorgi and Mohammad Reza Khayyambashi*
Faculty of Computer Engineering, University of Isfahan, Isfahan, Iran

Abstract. Link prediction is the problem of inferring future interactions among existing network members based on available knowledge. Computing similarity between a node pair is a known solution for link prediction. This article proposes some new similarity measures. Some of them use nodes' recency of activities, some weights of edges and some fusion of both in their calculation. A new definition of recency is provided here. A supervised learning method that applies a range of network properties and nodes similarity measures as its features set is developed here for experiments. The results of the experiments indicate that using proposed similarity measures would improve the performance of the link prediction.

Keywords: Link prediction, supervised learning, recency, similarity measures, social networks

1. Introduction

Immense real-world social networks exhibit a range of interesting properties [2, 15]. One of the noteworthy approaches in this line of research is to design models that predict the global structure of network [2, 15, 19].

Social networks are highly dynamic; they grow and change quickly through new interactions of the nodes in the network. Therefore, studying these changes at individual edge creation level is intriguing. Identifying the mechanism through which the new individual edges are added to the network is a fundamental research challenge. This challenge is called link prediction problem [9].

The study of networks to predict the upcoming links has beneficial applications in researches and organizational context in a variety of fields. Link prediction can help inferring the missing links from a network [10]. This issue can improve the security of the network by allowing one to assume the unobserved interactions among particular nodes [13]. Another application of link prediction is to predict the friendship or participations of actors in social networks [15].

To solve the link prediction problem, it needs to determine the probability of formation or dissolution of links. Usually, such probability is measured by similarities. There are many generic similarity measures, which use information of nodes and social network topology. Liben-Nowell and Kleinberg have discussed several metrics based on the graph structural features [15], after their work, many topology-based metrics were proposed [23, 31, 36]. These similarity metrics are commonly applicable in online social networks because of their tolerable computation time complexity [34]. Most of the available studies in this field lack of considering time factor in similarity measures computations.

A few studies consider temporality of similarity measures with introductory definitions of recency

*Corresponding author. Mohammad Reza Khayyambashi, Faculty of Computer Engineering, University of Isfahan, Postal Code 81746-73441, Isfahan, Iran. Tel.: +98 913 3676728; E-mail: m.r.khayyambashi@comp.ui.ac.ir.

[27, 35]. The former recency definitions do not discriminate new added nodes and the most recent active nodes. These former definitions do not consider recency of communications received by nodes too.

This article will discuss the impact of applying recency of nodes interactions on link prediction. A new definition of recency of nodes is greatly emphasized in this article, which considers both send and receive communications and includes timestamps of previous interactions. This recency was applied to propose some new similarity measures based on well-known classic similarity measures. The experimental study shows that applying recency of nodes significantly improve prediction capability of models.

Additionally, many improved versions of classic similarity measures are proposed in this article. These improvements focus on considering weights of links in the network and fusing recency and weights in similarity measures computations. Various aspects of applying these similarity measures and their impacts of prediction task are examined in an experimental manner. A supervised learning experiment framework is applied which considers different classifiers. The experiments indicate that applying recency of nodes alongside weights of edges would promote similarity measures too.

2. The literature review

The issue of link prediction in social network is a general concern consisting of several topics.

2.1. Link prediction problem

According to Liben-Nowell and Kleinberg a classical definition of link prediction is: "Given a snapshot of a social network at time t, we seek to accurately predict the edges that will be added to the network during the interval from time t to a given future time t'" [15]. This issue can be considered as a problem of supervised learning models, the objective of which is to predict existence of edge between a pair of nodes [11, 16]. Link prediction problem is considered as a supervised learning solution subject of the study on the contributions to various properties of network structure.

There exist various studies on this problem. The link prediction learning method can be divided into two broad categories: a) link prediction based on unsupervised learning methods [1, 15, 26], and b) link prediction based on supervised methods [11, 20, 31]. More studies are conducted on the category (a) and have become restricted due to heavy data load, have their weaknesses. The category (b) provides capabilities to improve link prediction results with more accuracy [16]. In this article, category (b) will be investigated and as a result a new method is proposed for link prediction of evolving online social network.

Most of the real world social networks data are immense and are presented in graph formats. Majority of the link prediction researchers applied classic datasets like co-authorship for their experiments [11, 15, 16]. But real world social networks graphs are sparse and encounter class imbalance problem. In [16], the authors apply some classic methods to overcome the existing class imbalance. An adjustment of a number of samples in dataset is made in this article to have more realistic results.

Different classifiers' results are investigated on link prediction problem by [11]. Although in [16] studied some factors regarding classification process improvement. Different similarity measures could be applied as the classifiers feature sets. The prediction results of models trained through these feature sets are applied in evaluating similarity measures. Some innovations are proposed in this article to improve classification by applying some innovative parameters as classification feature sets.

2.2. Nodes similarity

There exist different approaches to calculate similarity of nodes in several articles. The similarity of nodes is a common parameter to be applied as predictor. Cutting edge methods in similarity calculation are combined and compared in [15] and [19]. Similarity measures are commonly applied method in unsupervised learning studies. The basic idea is sorting the nodes based on similarity measure and selecting more similar nodes as probable ones to make links in future.

Many possible improvements in similarity measures are applicable. Some studies applied weights of edges for this aim. Authors in [23] proposed weighted Common Neighbor and weighted Adamic Adar and applied them in link prediction. De Sá and Prudêncio [31] applied three weighted similarity measures as their classification features Authors in [18] studied the role of weak ties in link prediction and proposed parameter free versions of weighted similarity measures. In this article some new similarity measures are proposed which merge weight of edges and recency of nodes interactions.

3. Methodology

A supervised learning approach is applied to evaluate the applicability of this article's proposed measures. The properties for supervised learning method, social network data used in this experiments, preprocessing job that prepare data for supervised learning and final classification job are presented in this section.

3.1. Baseline similarity measures

These baseline similarity measures contribute to the classification features of this experiments.

3.2. Node neighborhood based similarity measures

The nodes proximities in graph is the basic approach to assign similarity between them. The idea is to assign a connection weight $score(x, y)$ to each pair of the nodes x and y based on their proximity. For a node x, let $\Gamma(x)$ represent the set of neighbors of x in social network. There are some commonly applied similarity measures that are selected as the baseline measures in this experiment:

3.2.1. Common neighbors

Newman [24] computed $score(x, y)$ in the context of collaboration network, and verified that there is a correlation between the number of neighbors that x and y have in common at time t and the probability of their collaboration in future, presented as follows:

$$CN(x, y) = |\Gamma(x) \cap \Gamma(y)| \tag{1}$$

3.2.2. Jaccard's coefficient

This measure is commonly applied in information retrieval. By considering common neighbors as feature f that either x or y has, Jaccard coefficient can be computed by the probability of both x and y having common neighbors [15], Equation (2).

$$JC(x, y) = |\Gamma(x) \cap \Gamma(y)|/|\Gamma(x) \cup \Gamma(y)| \tag{2}$$

3.2.3. Adamic/Adar

This measure is a refined version of the common neighbors that take rare neighbors more serious [1]. The Adamic/Adar score can be presented through the following equation:

$$AA(x, y) = \sum\nolimits_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log |\Gamma(z)|} \tag{3}$$

3.2.4. Resource allocation

By considering a pair of nodes, x and y, which are not directly connected, node x can send some resource to node y, where their common neighbors act as transmitters. In the simplest case, it is assumed that each one of the transmitters has a unit of resource, and will distribute it to all of its neighbors on average. The similarity between x and y can be defined as the amount of resource y received from x, which is presented through the following equation [36]:

$$RA(x, y) = \sum\nolimits_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k(z)} \tag{4}$$

Where $k(z)$ is the degree of node z.

The findings in experiments of [36] indicate that the three measures of common neighbors (CN), Adamic/Adar (AA) and resource allocation (RA) have the best overall performance in comparison with other neighborhood based similarity measures. In this article these three similarity measures are applied in order to generate their extended versions.

3.3. Kernel based similarity measure

Authors of [32] and [12] studied the effectiveness of the *Neumann kernels* as a link analysis measure. They revealed that the *Neumann kernels* could be used as an efficient and powerful mechanism for determining "relatedness" between vertices. The relatedness between vertices is applied as a similarity measure in this article.

The *Neumann kernel* is defined in terms of an adjacency matrix X that its (i, j)-element is the frequency of i-th node relation with j-th node. From X, destination correlation matrix $K = X^T X$ and source correlation matrix $M = XX^T$ are first constructed.

Definition. Let X be an adjacency matrix, and let $K = X^T X$ and $M = XX^T$. The *Neumann kernel* matrices with *diffusion factor* $\gamma(\geq 0)$, denoted by $\widehat{K_\gamma}$ and $\widehat{M_\gamma}$, are defined as the solution to the following system of equations.

$$\widehat{K_\gamma} = \gamma X^T \widehat{M_\gamma} X + K, \ \widehat{M_\gamma} = \gamma X^T \widehat{K_\gamma} X + M \tag{5}$$

The similarity between nodes i and j is given by (i, j)-element of $\widehat{K_\gamma}$ and $\widehat{M_\gamma}$ combined implies an alternative representation based on the Neumann series presented through Equation (6).

$$\widehat{K_\gamma} = K\sum_{n=0}^{\infty}\gamma^n K^n,\ \widehat{M_\gamma} = M\sum_{n=0}^{\infty}\gamma^n M^n \tag{6}$$

Hence, when $\gamma < \rho(K)^{-1}(= \rho(M)^{-1})$, the solution exists and is given by $\widehat{K_\gamma} = K(I - \gamma K)^{-1}$ and $\widehat{M_\gamma} = M(I - \gamma M)^{-1}$.

3.4. Newly proposed similarity measures

Some new similarity measures, which apply recency of nodes; weights of edges between nodes, and fusing recency with weights of edges are proposed.

3.5. Recency based similarity measures

The active users of a social network are determined by time factor. By applying the recency concept, how recently a user acted would be determined. Recency is a metric which was preliminary used for the first time by Potgieter et al. [27]. They applied a very simple definition of recency to investigate temporality in link prediction. To them recency is "One plus the number of time steps elapsed since the node last communicated". In [35] authors used another definition for recency as "The time elapsed since a node made its last new link". The second definition does not discriminate between new added nodes and most recent active nodes in the network.

In this article another definition of recency is applied; which consider node recent activities alongside the most recent actions. The definition of recency concept by [22] where the following two equations are effected from is applied in this study:

$$recency_{sender}(u) = \sqrt{\frac{a}{i+1}} \tag{7}$$

$$recency_{reciever}(u) = \sqrt{\frac{a}{j+1}} \tag{8}$$

Where i is the difference between timestamp of current communication and last communication made by u, and j is the difference between timestamp of current communication and the last communication received by u. The value of $recency_{sender}(u)$ is the recency of the node with respect to the last communication made by, whereas $recency_{reciever}(u)$ is the recency value of the node u with respect to the last communication received by.

The values of i and j can be represented in seconds, minutes, hours and days depending on the application. In this article i and j are represented as hours, hence α take the value of 24.

The similarity measures are calculated based on the nodes' recency to provide better predictions. According to this idea, the temporal versions of common neighbor, Adamic/Adar and resource allocation are presented as TCN, TAA and TRA, respectively:

$$TCN(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} 1 + recency_{sender}(x) + recency_{reciever}(y) \tag{9}$$

$$TAA(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1 + recency_{sender}(x) + recency_{reciever}(y)}{\log|\Gamma(z)|} \tag{10}$$

$$TRA(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{1 + recency_{sender}(x) + recency_{reciever}(y)}{k(z)} \tag{11}$$

3.6. Weighted Similarity measures

The effectiveness of applying both the graph similarity measures and the weights of existing links in a social network is studied in [23], where the results indicate that their method outperforms previous approaches. By applying this weights of edges and with respect to common neighbor, Adamic/Adar and resource allocation similarity measures the weighted versions are applied in this study. These similarity measures are presented by WCN, WAA and WRA and can be obtained through:

$$WCN(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z) + w(z,\ y)}{2} \tag{12}$$

$$WAA(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z) + w(z,\ y)/2}{\log|1 + S(z)|} \tag{13}$$

$$WRA(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z) + w(z,\ y)/2}{S(z)} \tag{14}$$

Where, $w(x,\ y)$ is the weight of link between nodes x and y, and $S(x) = \sum_{z\in\Gamma(x)} w(x,\ z)$.

3.7. Fusing weights of edges and the nodes' recencies for similarity measures

As described, the recency of nodes and weights of the edges could be applied to propose new similarity measures. Consequently, the weighted temporal

versions of common neighbors, Adamic/Adar and resource allocation similarity measures are presented as:

$$WTCN(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z)+w(z,\ y)}{2} + recency_{sender}(x) + recency_{reciever}(y) \quad (15)$$

$$WTAA(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{(w(x,\ z)+w(z,\ y))/2}{\log|1+\mathrm{S(z)}|} + \frac{recency_{sender}(x)}{\log|1+\mathrm{S(z)}|} + \frac{recency_{reciever}(y)}{\log|1+\mathrm{S(z)}|} \quad (16)$$

$$WTRA(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{(w(x,\ z)+w(z,\ y))/2}{S(z)} + \frac{recency_{sender}(x)}{S(z)} + \frac{recency_{reciever}(y)}{S(z)} \quad (17)$$

3.8. Empirical data

Almost all the previous works have used the co-authorship dataset. This dataset does not contain the creation time of nodes; therefore, another dataset with complete longitudinal information is chosen here as the empirical data. The social network adopted in this experiment is a network created from an online community [25]. This network dataset includes 1899 students at the University of California, Irvine. The observation period is from April to October 2004. This dataset covers every online message the students sent to each other. A total number of 59835 messages between students could be seen in 20296 directed edges among them. Each edge has a weight attribute which presents the number of messages between two nodes. In addition there exist timestamps of nodes creation and interactions, which is applicable in time awareness of the proposed method.

In spite of the excellent contributions on the researchers' part, they have overlooked the overfitting of prediction model. Social network data are in the format of large graphs that merely contain presence of links. The dataset constructed by these graphs would have plenty of positive instances that can skew prediction results. To solve this problem some negative instances should be added to the dataset. In [11] and [31] authors added few negative instances to the dataset in a random manner to solve the problem. Here a weighted sampling is applied to confront this problem.

3.9. Development setting

All the experiments in this study are developed through R language and environment [29]. The igraph package [4] is applied in order to handle graph related functions. And the Caret package [8] is applied for supervised learning functions of the experiments. The pROC package [30] is applied for evaluations. The experiments are run on a windows 32-bit platform.

3.10. Evaluation metrics

Usually a classifier labels instances as either positive or negative. These prediction results can be represented in a structure known as a confusion matrix. The confusion matrix has four categories of true positive (*TP*), false positive (*FP*), true negative (*TN*), and false negative (*FN*). Confusion matrix can be used to construct different evaluation metrics for classifiers.

Accuracy is the ratio between the number of correct predictions and the total number of predictions. This evaluation metric is defined as:

$$ACC = (TP + TN)/Total\ Population \quad (18)$$

The Kappa statistic is another evaluation metric which is applied in this article [33]. This metric compares an observed accuracy with a random accuracy. The kappa statistic is used not only to evaluate a single classifier, but also to evaluate classifiers amongst themselves. This evaluation metric can be presented through:

$$Kappa = \frac{Accuracy - Random\ Accuracy}{1 - random\ Accuracy} \quad (19)$$

The random accuracy of the classification task is defined as the sum of the products of reference likelihood (actual true/false) and result likelihood (predicted true/false) which can be written as:

$$Random\ Accuracy = \frac{(TN+FP)*(TN+FN)+(FN+TP)*(FP+TP)}{Total*Total} \quad (20)$$

Sensitivity (a.k.a *True Positive Rate* and *Recall*) is the ratio of the number of *TP* to the total number of *positive* instances in a given dataset [28].

$$Sensitivity = \frac{TP}{TP + FN} \tag{21}$$

Specificity (a.k.a *True Negative Rate*) is the ratio of the number of *TN* to the total number of negative instances in a given dataset.

$$Specificity = \frac{TN}{TN + FP} \tag{22}$$

Precision (a.k.a *Positive Predictive Value*) is the ratio of the number of *TP* to the total number of instances predicted as positive.

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

Precision is interpreted as the ratio of the number of missing links predicted correctly [19].

Another evaluation metric constructed from confusion matrix is the *Receiver Operating Characteristic* (*ROC*) curve [6]. A *ROC* curve is a two-dimensional representation of the classifier performance which can be used to evaluate 2-class models. A *ROC* curve represent tradeoffs between *TP* and *FP*. Actually *ROC* curve plots the sensitivity (*TPR*) by one minus specificity (*TNR*). The *Area Under ROC* curve (*AUC*) could be used as a performance measure of classifiers [3, 6]. Any algorithm that obtains the highest *AUC* could be labeled as a better classifier.

4. Results and discussion

Any supervised learning algorithm needs a feature set in order to construct its model. All of the candidate similarity measures for feature sets in this experiments are tabulated in Table 1. By considering different combinations of these similarity measures; feature sets of this assessment would be made. To see their impact on link prediction it is best to add these features to feature set incrementally.

4.1. Preprocessing

Social networks data are in the form of sparse graphs. These graphs only represent existence of edges between the nodes, but there is a need for both positive and negative classes in supervised link prediction. This would lead to overfitting of the results; therefore, creation of negative classes is a major part of preprocessing. In spite of the excellent contributions on the researchers' part, they have overlooked the overfitting of prediction model. In [11] and [31] authors added few negative instances to the dataset in a random manner. The dataset constructed by these graphs would have plenty of positive instances that can skew prediction results.

Table 1
The list of all similarity measures used in the experiments

Similarity Measure	Denotation
Common Neighbor	CN
Weighted Common Neighbor	WCN
Temporal Common Neighbor*	TCN
Weighted Temporal Common Neighbor*	WTCN
Adamic/Adar	AA
Weighted Adamic/Adar	WAA
Temporal Adamic/Adar*	TAA
Weighted Temporal Adamic/Adar*	WTAA
Resource Allocation	RA
Weighted Resource Allocation*	WRA
Temporal Resource Allocation*	TRA
Weighted Temporal Resource Allocation*	WTRA
Jaccard's Coefficient	JC
Neumann Kernel	NK

*Proposed by this article.

A weighted sampling procedure is applied here, to confront this problem with the idea of adding some non-connected pairs of nodes as negative instances. The sampling size is calculated through Equations (23 and 24).

$$Negative\ instances = \frac{CE}{CE + NE} \times NE \tag{24}$$

$$Positive\ instances = \frac{NE}{CE + NE} \times CE \tag{25}$$

Where, CE is the number of connected edges presented in graph, and NE is the number of non-connected edges.

4.2. Classification

There exist many classification algorithms for supervised learning. Their performances are different for any given dataset. In this study two six different learning algorithms are applied: Linear Discriminant Analysis (LDA), Stochastic Gradient Boosting (GBM) [7], Boosted Classification Trees (ADA) [5] and C5.0 Decision Trees [14], Logistic Regression (GLM) and Random Forest (RF). In the first step LDA and GBM classification models are applied, and then the other four classifiers are applied in order to compare and study more.

Here 3-fold cross validation is applied for the classification algorithms. The performance of the classifiers are assessed through evaluation metrics *Accuracy, Kappa*, *AUC* and *Precision*.

Table 2
Prediction results of using baseline similarity measures

	LDA	GBM
NK	Acc: 0.6346630	Acc: 0.7497522
	Kappa: 0.2693	Kappa: 0.4995045
	AUC: 0.68	AUC: 0.8267
	Precision: 0.6441	Precision: 0.7017
CN	Acc: **0.6850842**	Acc: **0.7578048**
	Kappa: **0.3701685**	Kappa: **0.5156095**
	AUC: **0.753**	AUC: **0.8272**
	Precision: 0.669	Precision: 0.7178
AA	Acc: 0.6824827	Acc: 0.7578048
	Kappa: 0.3649653	Kappa: 0.5156095
	AUC: 0.7478	AUC: 0.8263
	Precision: 0.6718	Precision: 0.7165
JS	Acc: 0.6800050	Acc: 0.7711843
	Kappa: 0.3600099	Kappa: 0.5423687
	AUC: 0.7369	AUC: 0.8254
	Precision: 0.6712	Precision: 0.7089
RA	Acc: 0.6599356	Acc: 0.7583003
	Kappa: 0.3198712	Kappa: 0.5166006
	AUC: 0.7221	AUC: 0.8266
	Precision: 0.6636	Precision: 0.7164

The prediction results of the two selected classifiers are assessed in this article. In each runs a selection of nodes characteristics is applied as the classifiers' feature set. The classifiers are trained through feature sets in order to have different prediction models for prediction performances comparison.

The similarity measures in Table 1 are applied by classifiers singly and their prediction results are tabulated in Table 2. Unlike [36], where it is claimed that Resource Allocation outperforms others, the Common Neighbors similarity measure is ranked the first prediction results. It shows that in practical applications, one should choose right similarity measures according to the characteristics of social networks. There is no an absolutely dominating similarity measure for different datasets.

The prediction results, where this newly proposed similarity measures are applied, are tabulated in Table 3. These results indicate that applying weighted and temporal versions of similarity measures alone have minor impact on prediction performance. Assessments run on more aspects of applying these measures are tabulated in Tables 3–5.

Having the nodes' recency of activities would yield a new feature set consisting of an edge's sender and receiver's nodes temporal information as a new similarity measure. The prediction result of applying nodes recencies as similarity measure would allow to examine the impact of nodes recencies of activities on their upcoming activities. The experiments classifiers are trained with feature set and the prediction results are tabulated in Table 4.

Table 3
Prediction results of using weighted and temporal similarity measures

	LDA	GBM
WCN	Acc: 0.6849604	Acc: 0.756442
	Kappa: 0.369920	Kappa: 0.512884
	AUC: 0.7528	AUC: 0.8267
	Precision: 0.661	Precision: 0.7196
TCN	Acc: 0.6926412	Acc: 0.7570614
	Kappa: 0.385282	Kappa: 0.5141229
	AUC: 0.7585	AUC: 0.8272
	Precision: 0.6719	Precision: 0.7175
WTCN	Acc: 0.6848365	Acc: 0.7553271
	Kappa: 0.369672	Kappa: 0.5106541
	AUC: 0.753	AUC: 0. 8259
	Precision: 0.6689	Precision: 0.7171
WAA	Acc: 0.6819871	Acc: 0.7585481
	Kappa: 0.363974	Kappa: 0.5170961
	AUC: 0.7481	AUC: 0.8275
	Precision: 0.6716	Precision: 0.7159
TAA	Acc: 0.6865709	Acc: 0.7576809
	Kappa: 0.373141	Kappa: 0.5153617
	AUC: 0.7532	AUC: 0.8257
	Precision: 0.6695	Precision: 0.7169
WTAA	Acc: 0.6818632	Acc: 0.7573092
	Kappa: 0.363726	Kappa: 0.5146184
	AUC: 0.7479	AUC: 0.8278
	Precision: 0.6718	Precision: 0.7166
WRA	Acc: 0.6660059	Acc: 0.7581764
	Kappa: 0.332011	Kappa: 0.5163528
	AUC: 0.729	AUC: 0.8264
	Precision: 0.6623	Precision: 0.7174
TRA	Acc: 0.6652626	Acc: 0.757557
	Kappa: 0.330525	Kappa: 0.515114
	AUC: 0.7289	AUC: 0.8266
	Precision: 0.6626	Precision: 0.7165
WTRA	Acc: 0.663776	Acc: 0.7573092
	Kappa: 0.327552	Kappa: 0.5146184
	AUC: 0.7269	AUC: 0.8259
	Precision: 0.663	Precision: 0.716

Table 4
Prediction results of applying recencies of an edge's nodes as the similarity measure

	LDA	GBM
Recencies	Acc: 0.7124628	Acc: 0.8067393
	Kappa: 0.424925	Kappa: 0.613478
	AUC: 0.7854	AUC: 0.8856
	Precision: 0.706	Precision: 0.825

Table 5
Prediction results of applying combinations of similarity measures as classification feature sets

	LDA	GBM
ALL Sim	Acc: 0.7618930	Acc: 0.8527007
	Kappa: 0.523785	Kappa: 0.705401
	AUC: 0.8392	AUC: 0.9263
	Precision: 0.7436	Precision: 0.8616
ALL Sim no Recencies	Acc: 0.7171705	Acc: 0.8496036
	Kappa: 0.434340	Kappa: 0.699207
	AUC: 0.787	AUC: 0.9187
	Precision: 0.6883	Precision: 0.8548

To better represent the impact of this feature set on the prediction results an *AUC* comparison is applied, Fig. 1. The area under the ROC curve related to the classifier learnt through recencies is more than the other ROC curves. Moreover, This ROC curve is further to the 45-degree diagonal which means that the related classifier is more accurate. The results not only indicate that the recencies of nodes are usable as similarity measure but also this similarity measure outperforms other well-known similarity measures.

The next part of experiments consist of applying combinations of similarity measure as feature sets where two different combinations of similarity measures are applied in classification. The first combination consists of similarity measures which do not contain temporal information on the nodes' activities. The second one contains temporal versions of similarity measures in addition to the former measures. These results, tabulated in Table 5; indicate the strong impact of applying temporal similarity measures in the prediction task.

The ROC plots of three different feature sets applied in the experiments; the first feature set uses temporal resource allocation as similarity measure, second feature set uses a combination of all similarity measures except temporal ones and the third feature set use all similarity measures presented in this article are illustrated in Fig. 2. As shown in the diagram areas under the *ROC* curves (*AUCs*) related to combinations feature sets are remarkably better. These curves are closer to left and top borders which indicate more accurate classifiers. The best performance is related to the feature set which contains temporal information of the nodes. This comparison leads

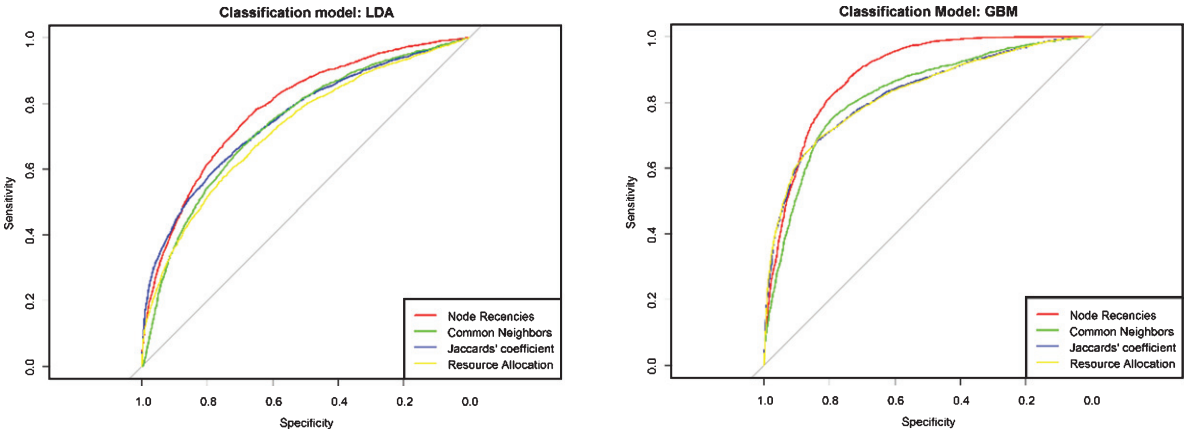

Fig. 1. The ROC plots of different classification models learnt through Node Recencies as similarity measures and other well-known similarity measures (Common Neighbors, Jaccards' coefficient and Resource Allocation) as feature sets. Area under these ROC curves (a.k.a AUCs) is a reliable evalution metric that indicate the classifier learnt through node recencies is more accurate and better predictor.

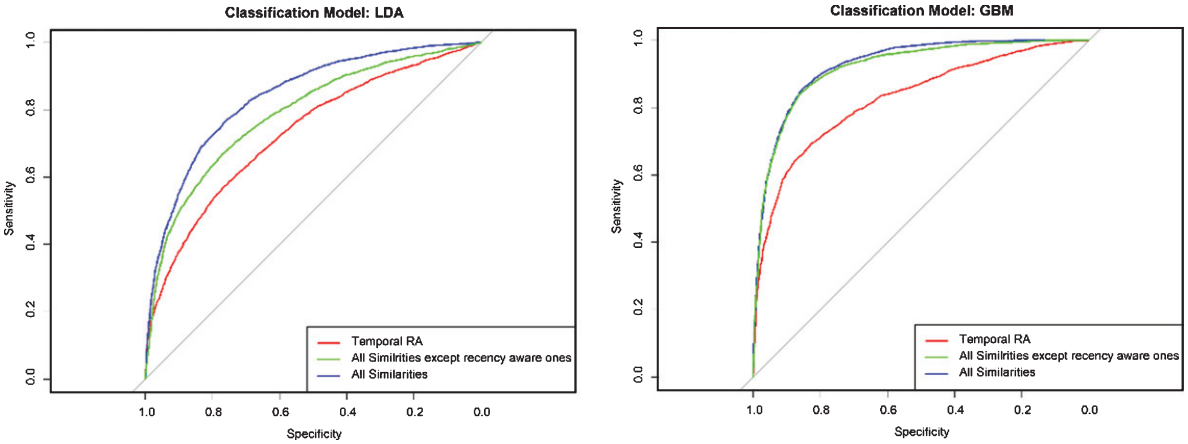

Fig. 2. The *ROC* curves of different classifiers learnt through Temporal Resource Allocation, All similarity measures except Temporal measures and All similarity measures as feature sets. The *ROC* curve related to classifier learnt through combination of similarity measures contain node recencies has has bigger *AUC* which indicate better prediction capability.

to two results: applying a combination of similarity measures improves the prediction results and applying temporal similarity measures where containing the recency of nodes would significantly improve prediction performance.

Different classifiers perform differently on a particular data set, therefore, the performance comparison for different classifiers on the empirical data is made in the experiment setting. The four well-known Boosted Classification Trees (ADA) and C5.0 Decision Trees, Logistic Regression (GLM) and Random Forest (RF) classification models would be applied in addition to the former classification models. The prediction results of applying these four classifiers are presented in Table 6.

The results are similar to the prior classification models applied in the first experiment. Applying the proposed similarity measures alone would slightly

Table 6
The prediction results of applying four other classification models in order to compare more. The boosted classification trees (ADA), C5.0 decision trees (C5.0), logistic regression (GLM) and random forest (RF) are applied

Feature set	ADA Results	C5.0 Results	GLM Results	RF Results
JS	Acc: 0.7662	Acc: 0.7762	Acc: 0.6991	Acc: 0.7807
	AUC: 0.8182	AUC: 0.8287	AUC: 0.7484	AUC: 0.8517
	Precision: 0.7076	Precision: 0.7123	Precision: 0.6752	Precision: 0.7106
NK	Acc: 0.7511	Acc: 0.75	Acc: 0.6391	Acc: 0.7566
	AUC: 0.8295	AUC: 0.8298	AUC: 0.685	AUC: 0.8348
	Precision: 0.702	Precision: 0.7018	Precision: 0.6471	Precision: 0.7051
CN	Acc: 0.756	Acc: 0.7578	Acc: 0.7409	Acc: 0.7751
	AUC: 0.8189	AUC: 0.8178	AUC: 0.8025	AUC: 0.8524
	Precision: 0.715	Precision: 0.7258	Precision: 0.6938	Precision: 0.7666
WCN	Acc: 0.7562	Acc: 0.7606	Acc: 0.7242	Acc: 0.7772
	AUC: 0.8183	AUC: 0.8266	AUC: 0.7871	AUC: 0.853
	Precision: 0.7163	Precision: 0.7377	Precision: 0.6878	Precision: 0.7717
TCN	Acc: 0.7532	Acc: 0.7604	Acc: 0.7412	Acc: 0.7767
	AUC: 0.8204	AUC: 0.8267	AUC: 0.8025	AUC: 0.8537
	Precision: 0.715	Precision: 0.7185	Precision: 0.694	Precision: 0.7715
WTCN	Acc: 0.7534	Acc: 0.7629	Acc: 0.7404	Acc: 0.7782
	AUC: 0.8187	AUC: 0.8276	AUC: 0.8025	AUC: 0.854
	Precision: 0.715	Precision: 0.7405	Precision: 0.6933	Precision: 0.7735
AA	Acc: 0.7549	Acc: 0.7566	Acc: 0.742	Acc: 0.7793
	AUC: 0.8192	AUC: 0.8207	AUC: 0.8011	AUC: 0.8565
	Precision: 0.7078	Precision: 0.7268	Precision: 0.6965	Precision: 0.7735
WAA	Acc: 0.7561	Acc: 0.7609	Acc: 0.7426	Acc: 0.7791
	AUC: 0.8205	AUC: 0.8265	AUC: 0.8011	AUC: 0.8563
	Precision: 0.7105	Precision: 0.7242	Precision: 0.6968	Precision: 0.7744
TAA	Acc: 0.7568	Acc: 0.7618	Acc: 0.7421	Acc: 0.7802
	AUC: 0.82	AUC: 0.8252	AUC: 0.801	AUC: 0.8592
	Precision: 0.7137	Precision: 0.7277	Precision: 0.6965	Precision: 0.7751
WTAA	Acc: 0.7553	Acc: 0.7606	Acc: 0.7426	Acc: 0.7796
	AUC: 0.8196	AUC: 0.8259	AUC: 0.8012	AUC: 0.8574
	Precision: 0.7128	Precision: 0.7242	Precision: 0.6969	Precision: 0.7763
RA	Acc: 0.7532	Acc: 0.7501	Acc: 0.7281	Acc: 0.7776
	AUC: 0.8202	AUC: 0.8245	AUC: 0.7901	AUC: 0.8572
	Precision: 0.7151	Precision: 0.7397	Precision: 0.6921	Precision: 0.7705
WRA	Acc: 0.7561	Acc: 0.7576	Acc: 0.7286	Acc: 0.7775
	AUC: 0.8223	AUC: 0.8308	AUC: 0.7907	AUC: 0.8569
	Precision: 0.7143	Precision: 0.7182	Precision: 0.6909	Precision: 0.7722
TRA	Acc: 0.7561	Acc: 0.7636	Acc: 0.7285	Acc: 0.7797
	AUC: 0.8206	AUC: 0.8289	AUC: 0.7901	AUC: 0.8572
	Precision: 0.7145	Precision: 0.7306	Precision: 0.6923	Precision: 0.774
WTRA	Acc: 0.7564	Acc: 0.7615	Acc: 0.7294	Acc: 0.7801
	AUC: 0.8251	AUC: 0.8262	AUC: 0.7908	AUC: 0.8574
	Precision: 0.7143	Precision: 0.7306	Precision: 0.6919	Precision: 0.7763
Recencies	Acc: 0.8012	Acc: 0.8273	Acc: 0.7158	Acc: 0.8325
	AUC: 0.8807	AUC: 0.9038	AUC: 0.784	AUC: 0.91
	Precision: 0.806	Precision: 0.863	Precision: 0.7029	Precision: 0.864
All Sim no Recencies	Acc: 0.8474	Acc: 0.8531	Acc: 0.8013	Acc: 0.8606
	AUC: 0.9233	AUC: 0.9182	AUC: 0.884	AUC: 0.929
	Precision: 0.8495	Precision: 0.855	Precision: 0.7445	Precision: 0.8657

improve prediction capability. The results indicate that applying temporal similarity measures combined with other measures would significantly increase prediction capability.

Another experiment is conducted in order to compare all of the classification models of this study. A complete combination of similarity measures presented in this article is used as the feature set. The prediction results are tabulated in Table 7. The *AUC* comparison of classifiers is shown in Fig. 3.

The random classifier illustrated by diagonal line has an accuracy of 0.5 by classifying all test data to be equal to positive or negative. *ROC* curves of different classifiers with a same feature set are plotted in Fig. 3. The classifier closer to the left and top borders and bigger area under *ROC* curve (*AUC*) has better prediction capability. With respect to the *ROC* curves of classifiers and their related *AUC* it is deduced that the Random Forest classifier performs better than other classifiers in supervised link prediction with an observable wider *AUC*.

4.3. Role of weak ties

In experiments it was observed that sometimes the weighted similarity measures perform similar or even worse that the simple versions. This observation reminds the weak ties theory, which state that sometimes the weak ties have more impact on link creation [17, 18, 21]. Therefore the role of weak ties in link prediction is investigated.

Table 7
Prediction results of applying different classification models on the empirical data, trained with all of the similarity measures

	LDA	GBM	ADA	C5.0	GLM	RF
Acc	0.761	0.852	0.846	0.866	0.805	0.868
Kappa	0.523	0.705	0.693	0.732	0.601	0.737
AUC	0.839	0.926	0.925	0.935	0.888	0.937
Precision	0.743	0.761	0.852	0.875	0.771	0.881

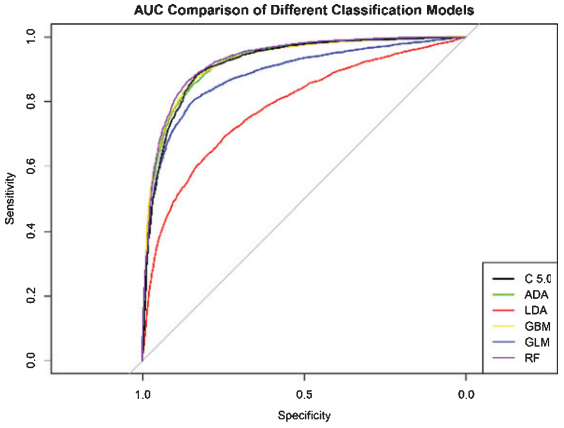

Fig. 3. *ROC* plots of different classifiers trained with all similarity measres a) LDA (red) b) GBM (yellow) c) ADA (green) d) C5.0 (black) e) GLM (blue) and f) RF (purple).

In order to investigate the role of weak ties in link prediction, the parameter free indices for WCN, WAA and WRA are applied [18]. These indices are presented as:

$$WCN^*(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z)^\alpha + w(z,\ y)^\alpha}{2} \quad (26)$$

$$WAA^*(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z)^\alpha + w(z,\ y)^\alpha/2}{\log|1+\mathrm{S(z)}|} \quad (27)$$

$$WRA^*(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z)^\alpha + w(z,\ y)^\alpha/2}{S(z)} \quad (28)$$

The α is a free parameter that range from –1 to 1.

The weights of edges are considered in three measures of WTCN, WTAA and WTRA. Therefore, the parameter free indices for these measures are proposed as:

$$WTCN^*(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{w(x,\ z)^\alpha + w(z,\ y)^\alpha}{2} + recency_{sender}(x) + recency_{reciever}(y) \quad (29)$$

$$WTAA^*(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{(w(x,\ z)^\alpha + w(z,\ y)^\alpha)/2}{\log|1+\mathrm{S(z)}|} + \frac{recency_{sender}(x)}{\log|1+\mathrm{S(z)}|} + \frac{recency_{reciever}(y)}{\log|1+\mathrm{S(z)}|} \quad (30)$$

$$WTRA^*(x,\ y) = \sum_{z\in\Gamma(x)\cap\Gamma(y)} \frac{(w(x,\ z)^\alpha + w(z,\ y)^\alpha)/2}{S(z)} + \frac{recency_{sender}(x)}{S(z)} + \frac{recency_{reciever}(y)}{S(z)} \quad (31)$$

When $\alpha = 1$, the similarity measures are equivalent to weighted indices. When $\alpha = 0$, the similarity

Table 8
Optimal values of parameter α subject to the highest precision

	WCN*	WAA*	WRA*	WTCN	WTAA	WTRA
α	–0.3	–0.1	–0.2	–0.3	–0.1	–0.5

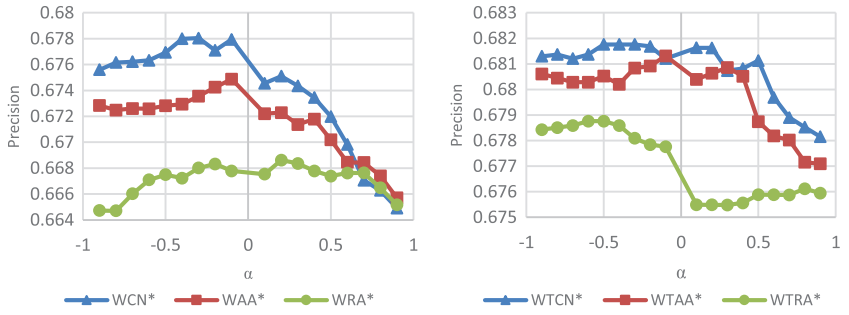

Fig. 4. Precision as a function of parameter α. The optimal values of α are negative and smaller that 1.

measures degenerate to the unweighted cases. The results of applying the parameter free indices in link prediction are given in Table 8 and Fig. 4. The optimal values of α are smaller than 1, which indicate that weak ties play a more important role than weights in the link prediction. These optimal values are negative, which indicate that the weak links play a more important role than the strong links.

5. Conclusion

Link prediction in social networks is an important issue with vast area of applications. The focus of this article is on studying nodes similarity measures and their analysis due to supervised learning algorithms for link prediction. One of the main contributions of this article is introducing innovative approaches in order to improve the similarity estimation of social networks' nodes. For this purpose the recency of the nodes' activities are applied as a source of knowledge. The new versions of many classic similarity measures where recency of nodes is applied are proposed in this article.

A supervised learning experiment setting is applied to assess the innovative aspects of this article. The supervised learning is chosen because of its capability to deliberate different combinations of similarity measures as feature sets. The classification models are one of the most reliable in prediction studies. Moreover, these models are capable of coping with the dynamic of social networks.

In this article categories of similarity measures are suggested which should be considered in supervised learning feature set. These categories consist of a) similarity measures of nodes based on their proximities, b) weighted version of well-known similarity measures, c) kernel based similarity measures, d) temporal versions of similarity measures where the recency of nodes is considered and e) similarity measures benefited from both weights of edges and temporal features.

The first groups of the experiments consist of comparing prediction results of applying different similarity measures in a separate manner as classification feature sets. The findings of the experiments indicate that using recency of nodes improve prediction results for some similarity measure slightly. Main improvement made in the prediction results is observed merely when recency of sender and receiver activities are applied as similarity measure of nodes.

The second groups of experiments assess the impact of applying temporal measures in feature sets. For this purpose two different combinations of similarity measures are applied as feature sets. A combination consisting of non-temporal similarity measures and a combination consisting of all measures are compared. The results indicate that applying temporal similarity measures improve the link prediction in a significant manner.

In order to more inspection prior experiments finding different classification models were applied on the data. The findings show that different classification models similarly affected by the proposed similarity measures.

The impact of applying different classification models on the prediction results is assessed too. The results have indicate that the predictions results would be improved through better classifiers.

Since it is observed that sometimes the similarity measures contain weights of edges perform similar or even worse than other measures; the parameter free versions of these measures are assessed. The results show that the weak ties play a great role in link prediction.

The general conclusion here is: regarding link prediction where supervised learning is applied, adopting temporal similarity measures in the classification would improve prediction capability.

Moreover, applying recency of nodes alongside weights of edges would promote similarity measures too.

In the future it would be interesting to investigate other ways to improve similarity measures. Moreover, the temporality could be applied in other related problems such as recommender systems.

Funding Acknowledgement

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] L.A. Adamic and E. Adar, Friends and neighbors on the web, *Soc Networks* **25** (2003), 211–230.

[2] A-L Barabâsi, H. Jeong, Z. Néda, E. Ravasz, A. Schubert and T. Vicsek, Evolution of the social network of scientific collaborations, *Phys A Stat Mech its Appl* **311** (2002), 590–614.

[3] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, *Pattern Recognit* **30** (1997), 1145–1159.

[4] G. Csardi and T. Nepusz, The igraph software package for complex network research, *Inter Journal Complex Sy* (2010), 1695.

[5] M. Culp, K. Johnson and G. Michailidis, ada: The R Package Ada for Stochastic Boosting, 2016.

[6] T. Fawcett, An introduction to ROC analysis, *Pattern Recognit Lett* **27** (2006), 861–874.

[7] J.H. Friedman, Stochastic gradient boosting, *Comput Stat Data Anal* **38** (2002), 367–378.

[8] M.K.C. Jed Wing, S. Weston, A. Williams, C. Keefer, A. Engelhardt, T. Cooper, Z. Mayer, B. Kenkel, the R Core Team, M. Benesty, R. Lescarbeau, A. Ziem and L. Scrucca, caret: Classification and Regression Training, 2015.

[9] L. Getoor, C. Park and C.P. Diehl, Link mining: A survey, *ACM SIGKDD Explor Newsl* **7** (2005), 3–12.

[10] D.S. Goldberg and F.P. Roth, Assessing experimentally derived interactions in a small world, *Proc Natl Acad Sci* **100** (2003), 4372–4376.

[11] M. Al Hasan, V. Chaoji, S. Salem and M. Zaki, Link prediction using supervised learning, *SDM'06 Work Link Anal Counter-terrorism Secur*, 2006.

[12] T. Ito, M. Shimbo, T. Kudo and Y. Matsumoto, Application of kernels to link analysis, In: *Proc Elev ACM SIGKDD Int Conf Knowl Discov data Min*, 2005, pp. 586–592.

[13] V.E. Krebs, Mapping networks of terrorist cells, *Connections* **24** (2002), 43–52.

[14] M. Kuhn, S. Weston, N. Coulter, code for C5.0 by R. Quinlan MCC, C50: C5.0 Decision Trees and Rule-Based Models, 2015.

[15] D. Liben-Nowell and J. Kleinberg, The link-prediction problem for social networks, *J Am Soc Inf Sci Technol* **58** (2007), 1019–1031.

[16] R.N. Lichtenwalter, J.T. Lussier and N.V. Chawla, New perspectives and methods in link prediction, In: *Proc 16th ACM SIGKDD Int Conf Knowl Discov data Min*, pp. 243–252.

[17] H. Liu, Z. Hu, H. Haddadi and H. Tian, Hidden link prediction based on node centrality and weak ties, *EPL (Europhysics Lett* **101** (2013), 18004.

[18] L. Lü and T. Zhou, Link prediction in weighted networks: The role of weak ties, *EPL (Europhysics Lett* **89** (2010), 18001.

[19] L. Lü and T. Zhou, Link prediction in complex networks: A survey, *Phys A Stat Mech its Appl* **390** (2011), 1150–1170. doi: 10.1016/j.physa.2010.11.027

[20] Z. Lu, B. Savas, W. Tang and I.S. Dhillon, Supervised link prediction using multiple sources, In: *2010 IEEE Int Conf data Min*, 2010, pp. 923–928.

[21] C. Ma, T. Zhou and H.-F. Zhang, Playing the role of weak clique property in link prediction: A friend recommendation model, *Sci Rep* **6** (2016).

[22] S. Mohan and M. Subramanian, A New Method of Identifying Individuals' Roles in Mobile Telecom Subscriber Data for Improved Group Recommendations, In: *Multidiscip Soc Networks Res Springer*, 2014, pp. 213–227.

[23] T. Murata and S. Moriyasu, Link prediction of social networks based on weighted proximity measures, In: *Web Intell IEEE/WIC/ACM Int Conf*, 2007, pp. 85–88.

[24] M.E.J. Newman, Clustering and preferential attachment in growing networks, *Phys Rev E* **64** (2001), 25102.

[25] T. Opsahl and P. Panzarasa, Clustering in weighted networks, *Soc Networks* **31** (2009), 155–163.

[26] A. Papadimitriou, P. Symeonidis and Y. Manolopoulos, Fast and accurate link prediction in social networking systems, *J Syst Softw* **85** (2012), 2119–2132. doi: 10.1016/j.jss.2012.04.019

[27] A. Potgieter, K. April, R.J.E. Cooke and I.O. Osunmakinde, Temporality in link prediction: Understanding social complexity, *Sprouts Work Pap Inf Syst* **7** (2007).

[28] D.M. Powers, Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation, 2011.

[29] R Core Team, R: A Language and Environment for Statistical Computing, 2015.

[30] X. Robin, N. Turck, A. Hainard, N. Tiberti, F. Lisacek, J.-C. Sanchez and L.M. Mi, pROC: An open-source package for R and S+ to analyze and compare ROC curves, *BMC Bioinformatics* **12** (2011), 77.

[31] H.R. De Sá and R.B.C. Prudêncio, Supervised link prediction in weighted networks, In: *Neural Networks (IJCNN), 2011 Int Jt Conf*, 2011, pp. 2281–2288.

[32] M. Shimbo, T. Ito, D. Mochihashi and Y. Matsumoto, On the properties of von Neumann kernels for link analysis, *Mach Learn* **75** (2008), 37–67. doi: 10.1007/s10994-008-5090-6

[33] A.J. Viera, J.M. Garrett and others, Understanding interobserver agreement: The kappa statistic, *Fam Med* **37** (2005), 360–363.

[34] P. Wang, B. Xu, Y. Wu and X. Zhou, Link prediction in social networks: The state-of-the-art, *Sci China Inf Sci* **58** (2015), 1–38.

[35] Y. Yang, N.V. Chawla, Y. Sun and J. Han, Predicting Links in Multi-relational and Heterogeneous Networks, In: *ICDM*, 2012, pp. 755–764.

[36] T. Zhou, L. Lü and Y.-C. Zhang, Predicting missing links via local information, *Eur Phys J B* **71** (2009), 623–630.