Link Prediction Based on Time Series of Similarity Coefficients and Structural Function

Piotr Stąpor(✉), Ryszard Antkiewicz, and Mariusz Chmielewski

Institute of Computer and Information Systems, Military University of Technology,
ul. gen. Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland
piotr.stapor@wat.edu.pl

Abstract. A social network is a structure whose nodes represent people or other entities embedded in a social context while its edges symbolize interaction, collaboration or exertion of influence between these forementioned entities [3]. From a wide class of problems related to social networks, the ones related to link dynamics seems particularly interesting. A noteworthy link prediction technique, based on analyzing the history of the network (i.e. its previous states), was presented by Prudêncio and da Silva Soares in [5]. In this paper, we attempt to improve the quality of edges' formation prognosis in social networks by proposing a modified version of aforementioned method. For that purpose we shall compute values of certain similarity coefficients and use them as an input to a supervised classification mechanism (called *structural function*). We stipulate that this function changes over time, thus making it possible to derive time series for all of its parameters and obtain their next values using a forecasting model. We might then predict new links' occurrences using the forecasted values of similarity metrics and supervised classification method with the predicted parameters. This paper contains also the comparison of ROC charts for both legacy solution and the novel method.

Keywords: Social network · Dynamic graph · Link prediction · Structural function

1 Introduction

Currently, networks are a commonly used tool used to describe a wide range of real-world phenomena [6]. A great amount of attention has been devoted to social networks analysis. The problem of link prediction has been already described extensively in literature. Liben-Nowell and Kleinberg [3] provide useful information and insights for regarding that issue, with references to some classical prediction measures based on topological features of analyzed network [7]. Lu and Zhou summarized, in [4], popular algorithms used for linkage is inside complex networks. Another interesting paper worth mentioning is [7]. In

S. Kozielski et al. (Eds.): BDAS 2019, CCIS 1018, pp. 168–179, 2019.
https://doi.org/10.1007/978-3-030-19093-4_13

this publication, authors provide a comprehensive and systematic survey[1] of the link prediction problem in social networks. The topics discussed there cover both classical and latest link prediction techniques, their applications, and active research groups [7]. Many solutions described there either make use of network's various topological metrics or perform data mining in order to reveal new or apply already existing structural patterns. The paper, however, neglected the analysis of how does these topological metrics evolve over time.

Especially interesting link prediction technique was proposed by Prudêncio and da Silva Soares in [5]. Authors proposed there calculating similarity scores for each pair of disconnected nodes at different time-frames, thus building a separate time series for each such pair. Subsequently, a forecasting model is applied to the series in order to predict their next values, which are then going to be used as input to unsupervised and supervised link prediction methods [5].

In this paper, we present modifications to the original method proposed by Soares and Prudêncio. Predicted values of similarity metrics are treated as an input to a supervised classification method. In further parts of this article, we will refer to this classification mechanism by a term *structural function*, as its value decides for whether a link exists for any given pair of nodes. We stipulate that this so called structural function changes over time, thus making is possible to derive time series for all parameters of a given structural function, and obtain their next values using the forecasting model. We might then predict new links' occurrences using the forecasted values of similarity metrics and supervised classification methods with predicted parameters.

The paper is organized as follows. In the Sect. 2 we will present preliminaries—the link prediction problem along with basic definitions. In Sect. 3 we will present the new method of link prediction. Section 4 contains a short description of a conducted experiment and its results. Conclusions are contained in Sect. 5.

2 Prerequisites

Definition 1 (Dynamic graph). *Let $G = (V, E)$ represent a graph containing vertices from set V and edges from E. Additionally, let $\mathbb{T}$ denote a set of moments in time, such that $\mathbb{T} = \{1, 2, ..., T, T+1, T+2, ..., \mathfrak{T}\}$, where $T > 1$ stands for the actual time. Through the term "dynamic graph" we shall understand an indexed family of graphs with t as a running index:*

$$\mathcal{G} = (G_t)_{t \in \mathbb{T}}, \tag{1}$$

where $G_t = (V_t, E_t)$ such that $V_1 = V_2 = ... = V_{\mathfrak{T}}$.

The link prediction problem can be formulated (based on [7]) as follows: Consider a social network of structure $\mathcal{G} = (G_t)_{t \in \mathbb{T}}$. The link prediction aims at: (a) forecasting a creation or disappearance of links between nodes in the future

[1] The authors cite 131 papers in their publication.

time-frame $t^* : t^* > T$ or (b) finding missing or unobserved links in current state of the network.

Definition 2 (Cumulative Dynamic graph). *A dynamic graph $\mathcal{G} = (G_t)_{t \in \mathbb{T}}$, where $G_t = (V_t, E_t)$, is a "cumulative dynamic graph" if additionally the following requirement is fulfilled:*

$$\forall t_1, t_2 \in \mathbb{T}\colon t_1 \leqslant t_2 \Longrightarrow E_{t_1} \subseteq E_{t_2}. \tag{2}$$

In this paper, we will limit our scope to sole prediction of new edges in graph G_{T+1}, while assuming that already-existing links are not deleted. Hence, whenever $\mathcal{G}$ appears, it symbolizes a cumulative dynamic graph.

Definition 3 (Graph's coefficient). *A coefficient C in the context of $\mathcal{G}$ can be thought as a function $C_{\mathcal{G}}\colon V^2 \times \mathbb{T} \rightarrow \mathbb{R}$, that returns a certain value for a pair of vertices and time frame t, according to the structure of graph G_t.*

2.1 Similarity Coefficients

In order to be able to compare our method against the one proposed in [5], we have decided to focus our preliminary research around measures used therein. Let $\Gamma_{\mathcal{G}}(v,t)$ denote a set of neighbors of a given vertex v in graph G_t. The common neighbors (CN) measure, for a pair of two vertices (v and w) can be defined as follows:

$$CN_{\mathcal{G}}(v,w,t) = |\Gamma_{\mathcal{G}}(v,t) \cap \Gamma_{\mathcal{G}}(w,t)| \tag{3}$$

According to CN measure suggests that the higher the mutual neighbors count, for a given pair of nodes, the higher the possibility that a connection between that pair should exist, yet it remains hidden or will exist. By its definition, CN is closely tied with Jaccard's coefficient (JC), known also as Link Relevance measure, which in fact is a CN value divided by analyzed pair's all neighbors count.

$$JC_{\mathcal{G}}(v,w,t) = \frac{|\Gamma_{\mathcal{G}}(v,t) \cap \Gamma_{\mathcal{G}}(w,t)|}{|\Gamma_{\mathcal{G}}(v,t) \cup \Gamma_{\mathcal{G}}(w,t)|} \tag{4}$$

JC is used to measure connection strength and thus plays an important role in the process of hidden links discovery inside graph knowledge-bases [1,2].

The Preferential Attachment (PA) is another measure that we will make use of. It assigns higher link materialization possibility to pairs of vertices with greater adjacent nodes count product. Though simple, the results obtained from experiments assert its ability to predict link formation.

$$PA_{\mathcal{G}}(v,w,t) = |\Gamma_{\mathcal{G}}(v,t)| \times |\Gamma_{\mathcal{G}}(w,t)| \tag{5}$$

Lastly, [5] proposes Adamic-Adar (AA) measure.

$$AA_{\mathcal{G}}(v,w,t) = \sum_{z \in |\Gamma_{\mathcal{G}}(v,t) \cap \Gamma_{\mathcal{G}}(w,t)|} \frac{1}{\log |\Gamma_{\mathcal{G}}(z,t)|} \tag{6}$$

We will leave JC and AA and utilize CN and PA for now. The reason behind this decision is to reduce the time complexity during the proof of concept phase. Furthermore, we would also like to avoid probable interdependence between applied measures. Thus, we refrain from using the JC as it seems to be correlated to a certain degree with CN. Once the proposed method yields positive results, we shall include more coefficients in future tests.

2.2 Description of the Algorithm Proposed by Prudêncio and da Silva Soares

The algorithm of link prediction proposed in [5] has the following steps:

I. For each pair of non-connected (v, w) nodes, create a time series $\overline{C}_T(v, w)$ of similarity coefficients' vector

$$\overline{C}_T(v, w) = (\boldsymbol{s}_t^{v,w})_{t=1,..,T}\,, \tag{7}$$

$$\boldsymbol{s}_t^{v,w} = \left[\, C_{\mathcal{G}}^1(v, w, t)\; C_{\mathcal{G}}^2(v, w, t)\; ...\; C_{\mathcal{G}}^N(v, w, t) \,\right]^\top, \tag{8}$$

where $C_{\mathcal{G}}^i(v, w, t)$ is the value of i-th similarity coefficient for pair of nodes (v, w) at moment t; the following metrics can be used as a similarity coefficient: Common Neighbors (CN), Preferential Attachment (PA), Adamic-Adar (AA) and Jaccard's Coefficient (JC);

II. Using time series $\overline{C}_T(v, w)$ and one of the forecasting methods (Moving Average, Average, Random Walk, Linear Regression, Simple Exponential Smoothing or Linear Exponential Smoothing) compute the future $T + 1$ values:

$$\boldsymbol{s}_{T+1}^{*v,w} = \left[C_{\mathcal{G}}^{*1}(v, w, T+1)\; C_{\mathcal{G}}^{*2}(v, w, T+1)\; ...\; C_{\mathcal{G}}^{*N}(v, w, T+1) \right]^\top. \tag{9}$$

III. Basing on the value of $\boldsymbol{s}_{T+1}^{*v,w}$, use either unsupervised or supervised method to predict new links.

In the unsupervised methods, the pairs of disconnected nodes are ranked according to their scores defined by a chosen similarity coefficients. It is assumed, that the top ranked pairs have highest probability of being connected in the future. The link prediction is treated as a classification task in the supervised approach. As a classifier, *Support Vector Machine* (SVM) is used in [5]. Data from the family $\mathcal{G}' = (G_1, G_2, ..., G_T)$ play the role of a training set, while data from graph $G_T + 1$ are used as a test network.

3 The Novel Method Proposition

The method of link prediction, proposed in this paper, is a modification of the one presented in [5]. We assume that not only values of similarity coefficients are changing in time but also the relation (approximated by a structural function) between values of a given similarity coefficient and probability of new link appearance.

For every $t \in \mathbb{T}$, the task of finding whether a given edge exists can be expressed as a classification problem. A pair of vertices $(v, w) \in V_t$ shall be assigned either to class 1 if there exists an edge $(v, w) \in E_t$ or to class 0 otherwise—i.e. $(v, w) \notin E_t$. Due to the randomness of the edge occurrence and the randomness[2] of similarity coefficients' values, we can use a classifier in the form of a regression function:

$$E\left\{I_{\{(v,w)\in E_t\}}|\boldsymbol{s}_t^{v,w} = \boldsymbol{val}\right\}, \tag{10}$$

where $\boldsymbol{val} \in \mathbb{R}^N$ (N is the number of used similarity coefficients) and also:

$$I_{\{(v,w)\in E_t\}} = \begin{cases} 1, & (v,w) \in E_t \\ 0, & (v,w) \notin E_t \end{cases}. \tag{11}$$

Due to the character of (11) the classifier takes the form of

$$P\left\{I_{\{(v,w)\in E_t\}} = 1|\boldsymbol{s}_t^{v,w} = \boldsymbol{val}\right\}. \tag{12}$$

Definition 4 (Structural function). *A structural function for a given $\mathcal{G} = (G_t)_{t\in\mathbb{T}}$, N coefficients and time step t is a function of a signature*

$$f_{\text{str}}^{C,\mathcal{G}} : \mathbb{R}^N \times \mathbb{T} \to [0,1], \tag{13}$$

that maps coefficients' values $\boldsymbol{val} \in \mathbb{R}^N$ obtained from graph G_t to the conditional probability of a link existence:

$$f_{\text{str}}^{C,\mathcal{G}}(\boldsymbol{val}, t) = P\left\{I_{\{(v,w)\in E_t\}} = 1|\boldsymbol{s}_t^{v,w} = \boldsymbol{val}\right\} \tag{14}$$

It might be helpful to note that a realization of $f_{\text{str}}^{C,\mathcal{G}}$ for one coefficient $C_{\mathcal{G}}$ is a mapping:

$$val, t \mapsto \frac{\left|\left\{(v,w) \in V_t^2 | C_{\mathcal{G}}(v,w,t) = val \wedge (v,w) \in E_t\right\}\right|}{\left|\left\{(v,w) \in V_t^2 | C_{\mathcal{G}}(v,w,t) = val\right\}\right|}. \tag{15}$$

3.1 Forecasting Links with a Structural Function of the Last Known Moment

To predict edges for time $t^* = T + 1$, evaluate coefficients for every vertices' pair and each time-step $t \in \{1, 2, ...T\}$ in series. The employment of polynomial regression mechanism allows to obtain forecasted values $\boldsymbol{s}_{T+1}^{*v,w}$ for each pair of nodes. Now, we may assess the probability that a link exists between a pair of vertices (v, w) in $\mathcal{G}$, with respect to selected coefficients, by inserting forecasted values into the structural function obtained from the last known moment - T. The final decision whether a link exists involves choosing a certain threshold value $\alpha \in [0, 1]$. If the obtained probability value is higher or equal to the threshold value, we assume that the link materializes.

[2] Similarity coefficients' values are random variables as they are functions of a random graph structure.

3.2 Overview of Our Derived Method

Our version of algorithm has the following form:

I. For each pair of non-connected nodes create a time series of similarity coefficients' vector as in (7);

II. For each $t = \overline{1,T}$ approximate a structural function for a mapping between similarity coefficients' vector for any pair of vertices to the existence or absence of a link $\boldsymbol{s}_t^{v,w} \mapsto \{1,0\}$. In other words, estimate parameters of the structural function for each time period $t = 1, 2, ..., T$ and therefore obtain time series of these parameters;

III. Calculate $\boldsymbol{s}_{T+1}^{*v,w}$ using time series $\overline{C}_T(u,v)$ and polynomial regression (9);

IV. Calculate the values of the structural function's parameters for $T+1$ by applying a polynomial regression to time series of structural function parameters;

V. Finally, predict new links existence by passing $\boldsymbol{s}_{T+1}^{*v,w}$ as an argument to the structural function and comparing its result with a threshold value.

3.3 Forecasting Links with the Predicted Structural Function

The structural function depends on time interval via its second argument. By restricting it to some $\tau \in \mathbb{T}$, we may observe how coefficient C relates to the probability of edge existence in that snapshot. One way of observing, how this behavior changes through time is to look at $f_{\text{str}}^{C,\mathcal{G}}$ as a sequence of its own restrictions.

$$f_{\text{str}}^{C,\mathcal{G}}\Big|_{t=1}, f_{\text{str}}^{C,\mathcal{G}}\Big|_{t=2}, ..., f_{\text{str}}^{C,\mathcal{G}}\Big|_{t=\tau}, ..., f_{\text{str}}^{C,\mathcal{G}}\Big|_{t=T} \tag{16}$$

To simplify future formulas we shall apply here some syntactic sugar and treat $f_{\text{str}}^{C,\mathcal{G}}|_{t=1}$ as $\mathfrak{f}_1^C$, $f_{\text{str}}^{C,\mathcal{G}}|_{t=2}$ as $\mathfrak{f}_2^C$ and so on, keeping the obvious $\mathcal{G}$ context in mind.

$$\mathfrak{f}_\tau^C\,(val) \triangleq f_{\text{str}}^{C,\mathcal{G}}(val,t)\Big|_{t=\tau} \tag{17}$$

The changes occurring in dynamic graph's structure throughout its subsequent phases may cause $\mathfrak{f}_\tau^C$ to return quite different value than $\mathfrak{f}_{\tau+1}^C$ for the very same pair of nodes and coefficient C. If the analyzed net alters in a particular manner, the obtained values may reveal a certain trend. For example, during our research we have found out that a network showing collaborations between authors of scientific publications exhibits a characteristic of a logistic function. This corollary led us to the idea of prognosing the structural function values at time $t^* = T+1$, that is what $\mathfrak{f}_{T+1}^{*C}$ would have looked like at time t^*. Performing logistic regression (or any that fits the trend) for each function from $\mathfrak{f}_1^C, \mathfrak{f}_2^C, ..., \mathfrak{f}_T^C$ sequence will leave us with T corresponding vectors of logistic models coefficients: $\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_T$. To discovery of their behavior can be achieved by running polynomial regression for each position in obtained vectors.

If $\boldsymbol{B}_i = \begin{bmatrix} b_{i1} \; b_{i2} \; ... \; b_{in} \end{bmatrix}^\top$ then applying the polynomial regression for each of its position will allow us to predict a coefficients' vector for the time $t^* = T+1$, which we will denote it as $\boldsymbol{B}^{*}_{T+1}\left[b^{*}_{1}\; b^{*}_{2}\; ...\; b^{*}_{n}\right]^{\top}$.

$$\begin{aligned} b_{11}, b_{21}, b_{31}, ..., b_{T1} &\xrightarrow{\text{pred. by reg.}} b^{*}_{1} \\ b_{12}, b_{22}, b_{32}, ..., b_{T2} &\xrightarrow{\text{pred. by reg.}} b^{*}_{2} \\ &\vdots \\ b_{1n}, b_{2n}, b_{3n}, ..., b_{Tn} &\xrightarrow{\text{pred. by reg.}} b^{*}_{n} \end{aligned}$$

The predicted coefficients $\boldsymbol{B}^{*}_{T+1}$ can then be inserted into logistic function formula, hence unfolding the expected shape of structural function $\mathfrak{f}^{C}_{t^{*}}$. For vector $\boldsymbol{s}^{*v,w}_{T+1} = \left[s^{*}_{1}\; s^{*}_{2}\; ...\; s^{*}_{N}\right]^{\top}$:

$$\mathfrak{f}^{*C}_{T+1}\left(\boldsymbol{s}^{*v,w}_{T+1}\right) = \frac{\exp\left(\boldsymbol{B}^{*}_{T+1} \cdot \overline{\boldsymbol{s}}\right)}{1 + \exp\left(\boldsymbol{B}^{*}_{T+1} \cdot \overline{\boldsymbol{s}}\right)}, \tag{18}$$

where $\overline{\boldsymbol{s}} = \left[1\; s^{*}_{1}\; s^{*}_{2}\; ...\; s^{*}_{N}\right]^{\top}$.

For example, the application of logistic regression for one coefficient C will result in a series of vectors $\boldsymbol{B}_{i} = \left[b_{1}\; b_{2}\right]^{\top}$ and a prediction: $\boldsymbol{B}^{*}_{T+1} = \left[b^{*}_{1}\; b^{*}_{2}\right]^{\top}$. In this case, the probability value can be evaluated with the formula:

$$\mathfrak{f}^{*C}_{T+1}\left(C^{*}_{\mathcal{G}}(v, w, T+1)\right) = \frac{\exp\left(b^{*}_{1} + b^{*}_{2}C^{*}_{\mathcal{G}}(v, w, T+1)\right)}{1 + \exp\left(b^{*}_{1} + b^{*}_{2}C^{*}_{\mathcal{G}}(v, w, T+1)\right)}. \tag{19}$$

Again, as in Sect. 3.1, a link (v, w) will materialize when $\mathfrak{f}^{*C}_{T+1}\left(\boldsymbol{s}^{*v,w}_{T+1}\right) \geqslant \alpha$, where α is the chosen threshold.

3.4 Extending the Method for N Coefficients

The method can be accommodated to take any positive number of measures into account. This can be accomplished by inserting each measure's values into a set of even-length intervals. The number of divisions and their size may vary for each coefficient. Let $\mathbb{C} = \left\{C^{1}_{\mathcal{G}}, C^{2}_{\mathcal{G}}, ..., C^{N}_{\mathcal{G}}\right\}$ be a set of N coefficients' evaluation functions. Now, let $\mathfrak{D}\colon \mathbb{C} \times \mathbb{N}^{+} \rightarrow 2^{\mathbb{R}}$ denote a function that, for a given coefficient $C_{\mathcal{G}}$, divides space $[0, \max C_{\mathcal{G}}(v, w, t)]$ into a set of consecutive, equal-length, d intervals: $(C_{\mathcal{G}}, d) = \left\{\Delta^{C}_{1}, \Delta^{C}_{2}, ..., \Delta^{C}_{d}\right\}$. Through $\max C_{\mathcal{G}}(v, w, t)$ we marked the highest value achieved for a given $C_{\mathcal{G}}$ up to the predicted time-frame.

To every interval we will now assign its representative value (via $\mathfrak{R}\colon 2^{\mathbb{R}} \rightarrow \mathbb{R}$ function)—in our study we decided to use interval's average value.

Having got through the definitions we may now construct, for a given $\mathcal{G}$, an indexed family of tables: $\mathcal{T} = (tabl_{t})_{t \in \{1,2,...,T\}}$ (one table per each time frame) that will constitute a data to be consumed by logistical regression mechanism while searching for structural function coefficients. This calls for evaluating all of N coefficients for every pair of vertices in each time step.

The table contains information on how many pairs of vertices can be found in a given N-dimension space fragment and how many of them are actually linked. A given pair of nodes (v, w) belongs to the space fragment represented by $\left(\mathfrak{R}(\Delta^{C^1_{\mathcal{G}}}), \mathfrak{R}(\Delta^{C^2_{\mathcal{G}}}), ..., \mathfrak{R}(\Delta^{C^N_{\mathcal{G}}})\right)$ if $\forall i \in 1, 2, ..., N\colon C^i_{\mathcal{G}}(v, w, t) \in \Delta^{C^i_{\mathcal{G}}}$.

Table 1. Data of $tabl_t$ from time frame t, used for finding the coefficients of logistic structural function that utilizes n measures. Column designations: LP stands for *linked pairs*, AP – *all pairs*, C_i^{rep} – a representative value for a given C_i's interval.

LP	AP	C_1^{rep}	C_2^{rep}	...	C_N^{rep}
lp_1	ap_1	$\mathfrak{R}(\Delta_1^{C^1_{\mathcal{G}}})$	$\mathfrak{R}(\Delta_1^{C^2_{\mathcal{G}}})$	...	$\mathfrak{R}(\Delta_1^{C^N_{\mathcal{G}}})$
lp_2	ap_2	$\mathfrak{R}(\Delta_1^{C^1_{\mathcal{G}}})$	$\mathfrak{R}(\Delta_1^{C^2_{\mathcal{G}}})$	...	$\mathfrak{R}(\Delta_2^{C^N_{\mathcal{G}}})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
lp_j	ap_j	$\mathfrak{R}(\Delta_1^{C^1_{\mathcal{G}}})$	$\mathfrak{R}(\Delta_2^{C^2_{\mathcal{G}}})$	...	$\mathfrak{R}(\Delta_1^{C^N_{\mathcal{G}}})$
lp_{j+1}	ap_{j+1}	$\mathfrak{R}(\Delta_1^{C^1_{\mathcal{G}}})$	$\mathfrak{R}(\Delta_2^{C^2_{\mathcal{G}}})$	...	$\mathfrak{R}(\Delta_2^{C^N_{\mathcal{G}}})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$
lp_k	ap_k	$\mathfrak{R}(\Delta_2^{C^1_{\mathcal{G}}})$	$\mathfrak{R}(\Delta_1^{C^2_{\mathcal{G}}})$	...	$\mathfrak{R}(\Delta_1^{C^N_{\mathcal{G}}})$
lp_{k+1}	ap_{k+1}	$\mathfrak{R}(\Delta_2^{C^1_{\mathcal{G}}})$	$\mathfrak{R}(\Delta_1^{C^2_{\mathcal{G}}})$	...	$\mathfrak{R}(\Delta_2^{C^N_{\mathcal{G}}})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	...	$\vdots$

Let us now introduce a logit function [8]: $L(p_i) = \ln(p_i/(1 - p_i))$, where, $p_i = lp_i/ap_i$. This lets us apply the generalized least squares (GLS) technique from [8] to find structural functions' coefficients. The algorithm continues then as shown in Sect. 3.3.

3.5 A Detailed Pseudo-Code for the Proposed Method

Let $\mathbb{X}_t$ and $\mathbb{Y}_t$ be mutable integer maps (for time frame t)—i.e. mappings of type $\mathbb{N}_0 \rightarrow \mathbb{N}_0$, such that:

- initially every $n \in \mathbb{N}_0$ is associated with zero—$n \mapsto 0$,
- every IncrementMappingValueForKey($\mathbb{M}, n$) call, where $\mathbb{M}$ is a mapping, increments the value returned for n by 1 (E.g. After two such calls with 3 as the second argument $\mathbb{M}(3) = 2$.)

Initially, a value of coefficient C is computed ($C_{\mathcal{G}}(v, w, t)$) for every pair of nodes in the graph at every historical time step $1, 2, ..., T$. The results form a matrix $\boldsymbol{X}$, such that its every row contains a series of sequential values obtained at different moments of time. At line 6 we increase a number of pairs with similar result by one, while at line 8 only existing links with that value are accounted for. Line 9 is responsible for fitting a regression curve of a structural function at time t. At line 9 a structural function is predicted for time $T+1$. At line 13 we than obtain a forecast of coefficient C at $T+1$. Finally the link occurrence prediction can be assessed.

Algorithm 1. The algorithm for one coefficient C returning $\mathbb{N}_0$.

```
 1: procedure PREDICT($\mathcal{G}$, $C_{\mathcal{G}}$, $\alpha$)
 2:     for every $t = \overline{1, T}$ do
 3:         for every $(v, w) \in V_t^2$ of graph $G_t = (V_t, E_t)$ do
 4:             $c \leftarrow C_{\mathcal{G}}(v, w, t)$
 5:             $\boldsymbol{X}$[GETROWFORNODEPAIR$(v, w), t] \leftarrow c$
 6:             INCREMENTMAPPINGVALUEFORKEY$(\mathbb{X}_t, c)$
 7:             if $(v, w) \in G_t$ then
 8:                 INCREMENTMAPPINGVALUEFORKEY$(\mathbb{Y}_t, c)$
 9:         $\boldsymbol{B}_t \leftarrow$ LOGISTICREGRESSIONFIT$(\mathbb{Y}_t, \mathbb{X}_t)$
10:     $\boldsymbol{B}^* \leftarrow$ PREDICTNEXTUSINGPOLYNOMIALREGRESSION$(\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_T)$
11:     for every row $r$ in $\boldsymbol{X}$ do
12:         $\boldsymbol{x} \leftarrow$ GETROWFROMMATRIX$(\boldsymbol{X}, r)$
13:         $x^* \leftarrow$ PREDICTNEXTUSINGPOLYNOMIALREGRESSION$(\boldsymbol{x})$
14:         $p \leftarrow$ VALUEOFLOGISTICFUNWITHPARAMSAT$(\boldsymbol{B}^*, x^*)$
15:         if $p \geqslant \alpha$ then
16:             $\boldsymbol{P}[r] \leftarrow 1$
17:         else
18:             $\boldsymbol{P}[r] \leftarrow 0$
19:     return $\boldsymbol{P}$
```

The next algorithm also requires some commentary. The $\boldsymbol{D}$ vector contains numbers of divisions (intervals) for each coefficient found in a list $\mathbb{C}$. The function GETMAXIMUMINCOLUMN at line 12 returns a maximum value ever returned by a given coefficient. PREPAREDIVTABLES (line 13) creates a table for each coefficient containing intervals and their representative values. The last function call that may appear obscure to the reader is PREPAREDATATABLE() from line 15. Its purpose is to create a table of a form presented by Table 1 in Sect. 3.4.

4 The Experiment

In order to evaluate the novel method an experiment was conducted in which a prediction about future collaboration of authors in Arxiv[3] publications' database was to be attained. Like in the case of [5], the scope included all articles in *High Energy Physics – Lattice archive* (hep-lat[4]) published between 1993 and 2010 with an accuracy to a month. Each time-frame corresponded to one year. (In order to gain some reduction in algorithms' execution time, yearly data, that

[3] https://arxiv.org.

[4] https://arxiv.org/archive/hep-lat.

Algorithm 2. The algorithm for n coefficients in a list $\mathbb{C}$.

```
procedure $\textsc{Predict}(\mathcal{G}, \mathbb{C}, \boldsymbol{D}, \alpha)$
    $N \leftarrow \textsc{Length}(\mathbb{C})$
    for every $t = \overline{1, T}$ do
        for every $(v, w) \in V_t^2$ of graph $G_t = (V_t, E_t)$ do
            for $i \leftarrow 1$ to $N$ do
                $\boldsymbol{X}_t[\textsc{GetRowsForNodePair}(v, w), i] \leftarrow \mathbb{C}[i](v, w, t)$
            if $(v_1, v_2) \in G_t$ then
                $\boldsymbol{Y}_t[\textsc{GetRowForNodePair}(v, w)] \leftarrow 1$
            else
                $\boldsymbol{Y}_t[\textsc{GetRowForNodePair}(v, w)] \leftarrow 0$
    for $i \leftarrow 1$ to $N$ do
        $\boldsymbol{Max}[i] \leftarrow \textsc{GetMaximumInColumn}(i, \langle \boldsymbol{X}_1, \boldsymbol{X}_2, ..., \boldsymbol{X}_T \})$
    $divTables \leftarrow \textsc{PrepareDivTables}(\boldsymbol{Max}, \boldsymbol{D})$
    for every $t = \overline{1, T}$ do
        $dataTable_t \leftarrow \textsc{PrepareDataTable}(divTables, \boldsymbol{X}_t, \boldsymbol{Y}_t)$
        $\boldsymbol{B}_t \leftarrow \textsc{LogisticRegressionFit}(dataTable_t)$
    $\boldsymbol{B}^* \leftarrow \textsc{PredictNextUsingPolynomialRegression}(\boldsymbol{B}_1, \boldsymbol{B}_2, ..., \boldsymbol{B}_T)$
    for every row $r$ in $\boldsymbol{X}_1$ do
        for $i \leftarrow 1$ to $N$ do
            $\boldsymbol{x}_i \leftarrow \textsc{CreateVector}(\boldsymbol{X}_1[r, i], \boldsymbol{X}_2[r, i], ..., \boldsymbol{X}_T[r, i])$
            $x_i^* \leftarrow \textsc{PredictNextUsingPolynomialRegression}(\boldsymbol{x}_i)$
        $\boldsymbol{x}^* \leftarrow \textsc{CreateVector}(x_i^*, x_2^*, ..., x_N^*)$
        $p \leftarrow \textsc{ValueOfLogisticFunWithParamsAt}(\boldsymbol{B}^*, \boldsymbol{x}^*)$
        if $p \geqslant \alpha$ then
            $\boldsymbol{P}[r] \leftarrow 1$
        else
            $\boldsymbol{P}[r] \leftarrow 0$
    return $\boldsymbol{P}$
```

was actually taken into account and processed, had been limited only to records from four months: 3rd, 6th, 9th and 12th.) It should be noted that edges are added in a cumulative manner - i.e. once a pair of vertices is bond by an edge, it remains connected. Two approaches were confronted:

- using the last structural function to obtain prognosis (AP1) and
- utilizing the predicted structural function for same purpose (AP2).

The computation has been done using a dedicated software. In order to reduce prognosis complexity, the developed software divides the analyzed multi-graph into weak connected components with respect to its last structure in time series so that for any pair of vertices coming from different components neither PA nor CN may yield output other than zero. The comparison of the quality of both solutions was assessed with the help of ROC (*Receiver Operating Characteristic*) charts.

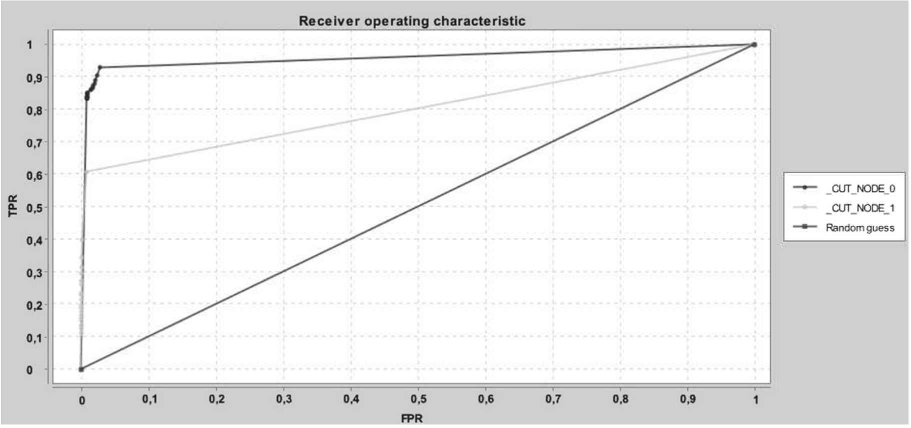

Fig. 1. ROC chart for single coefficient model—CN. The upper series curve (_CUT_NODE_0) illustrates the effectiveness of AP2 approach, while _CUT_NODE_1 of AP1. The diagonal line across represents expected the prognostic ability of a random guess.

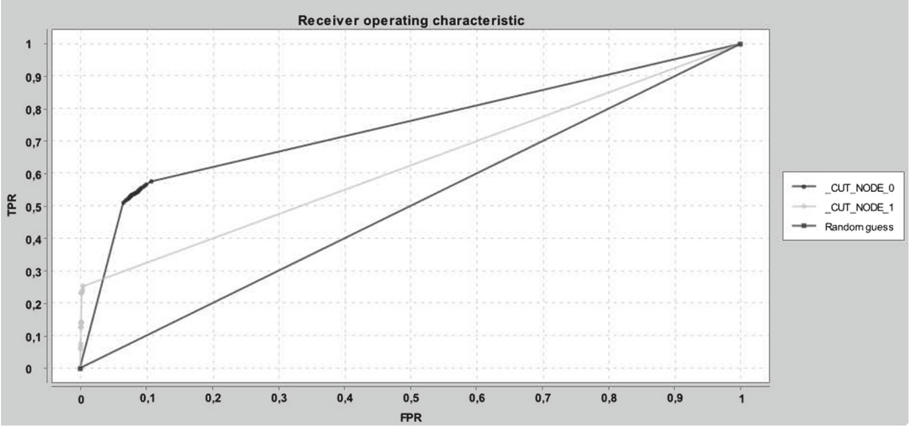

Fig. 2. ROC chart for single coefficient model—PA

A note on interpreting the ROC charts. The mechanism, that forecasts of weather the link should or should not exist, can be viewed as kind of a binary classifier, hence the usage of ROC charts as the assessment tool. The presented ROC charts shows how well the classifier performs for different levels of threshold α (please refer to Sect. 3.3 for α). The TPR (*True Positive Rate*) value assesses how well the mechanism performs for positives—i.e. how many observations categorized belong there truthfully. The higher the value, the better. On the other hand, the FPR (*False positive Rate*) measures how poorly the classification works for negatives—i.e. how many observations categorized as negatives, have been wrongly assigned. The lower the value, the better. In conclusion, the closer the ROC curve passes by the upper-left corner, the better the classifier works.

As it can be seen in Fig. 1, the usage of the forecasted structural function with CN improves the classification results. When it comes to PA (Fig. 2), the proposed modification causes FPR to increase slightly, but on the other hand it performs a better job when classifying positives

5 Conclusion and Future Work

As shown in the results section, performing prediction with the help of a forecasted structural function improves the classification rate of true positives (TPR). Although obtained false positive ratio (FPR) seems inferior to last-step structural function forecast, the gain from the improved TPR classification surpasses by far that loss, thus making the usage of forecasted structural function sensible and advisable. In near future we plan to experiment with other similarity measures (such as AA or JC) and running a series of experiments for multi-coefficient version of the algorithm. Further research would concentrate on: (a) experimentation with other kinds of social networks, (b) proposal of recommended set of uncorrelated coefficients, (c) taking into account that some pairs of nodes may become disconnected and (d) the introduction of cooperation intensity concept.

References

1. Barthélemy, M., Chow, E., Eliassi-rad, T.: Knowledge representation issues in semantic graphs for relationship detection. In: AAAI Spring Symposium on AI Technologies for Homeland Security, pp. 91–98. AAAI Press (2005)
2. Chmielewski, M., Stąpor, P.: Protégé based environment for DL knowledge base structural analysis. In: Jędrzejowicz, P., Nguyen, N.T., Hoang, K. (eds.) ICCCI 2011. LNCS (LNAI), vol. 6922, pp. 314–325. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23935-9_31
3. Liben-Nowell, D., Kleinberg, J.: The link-prediction problem for social networks. J. Am. Soc. Inf. Sci. Technol. **58**(7), 1019–1031 (2007)
4. Lu, L., Zhou, T.: Link prediction in complex networks: a survey. Phys. A: Stat. Mech. Appl. **390**(6), 1150–1170 (2010)
5. da Silva Soares, P.R., Prudêncio, R.B.: Time series based link prediction. In: Proceedings of the International Joint Conference on Neural Networks, June 2012
6. Rossetti, G., Guidotti, R., Miliou, I., Pedreschi, D., Giannotti, F.: A supervised approach for intra-/inter-community interaction prediction in dynamic social networks. Soc. Netw. Anal. Min. **6**(1), 86 (2016)
7. Wang, P., Xu, B., Wu, Y., Zhou, X.: Link prediction in social networks: the state-of-the-art. CoRR abs/1411.5118 (2014). http://arxiv.org/abs/1411.5118
8. Zeliaś, A., Pawełek, B., Wanat, S.: Prognozowanie ekonomiczne: teoria, przykłady, zadania. Wydawnictwo Naukowe PWN (2003)