Representing a Content-based link Prediction Algorithm in Scientific Social Networks

Hosna Solaimannezhad*
Faculty of Electrical and Computer Engineering,University of Tehran, Tehran, Iran
hosna.sn91@yahoo.com
Omid Fatemi
Faculty of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
omid@fatemi.net

Received: 23/Apr/2017 Revised: 05/Sep/2017 Accepted: 08/Oct/2017

Abstract

Predicting collaboration between two authors, using their research interests, is one of the important issues that could improve the group researches. One type of social networks is the co-authorship network that is one of the most widely used data sets for studying. As a part of recent improvements of research, far much attention is devoted to the computational analysis of these social networks. The dynamics of these networks makes them challenging to study. Link prediction is one of the main problems in social networks analysis. If we represent a social network with a graph, link prediction means predicting edges that will be created between nodes in the future. The output of link prediction algorithms is using in the various areas such as recommender systems. Also, collaboration prediction between two authors using their research interests is one of the issues that improve group researches. There are few studies on link prediction that use content published by nodes for predicting collaboration between them. In this study, a new link prediction algorithm is developed based on the people interests. By extracting fields that authors have worked on them via analyzing papers published by them, this algorithm predicts their communication in future. The results of tests on SID dataset as co-author dataset show that developed algorithm outperforms all the structure-based link prediction algorithms. Finally, the reasons of algorithm's efficiency are analyzed and presented.

Keywords: Link prediction; Social networks; Content-based; Interest.

1. Introduction

As a part of recent study progress, a great deal of attention has been devoted to computational analysis of social networks. Indeed, a social network is a social structure composed of a set of social actors and set of communications between these actors. A social network can be imagined as a graph in which the nodes show the entities that are in a social context and the edges refer to the communication, interaction, collaboration or the effect between these entities. Any unit that could connect to other units could be considered as a node in the social network. One of the social networks is co-author network. In these networks, the nodes represent the papers' authors and the edges show the collaboration of these authors in writing papers. Like other social networks, these social networks are dynamic and they are changing over the time via adding the nodes and edges. The dynamics of these networks has turned them into a challenging topic for study. The network becomes even more complex as network nodes and edges grow, and they can be analyzed using the network analysis. In the social network analysis, analyzing the communication between this network's actors and analyzing the content published by these actors are the main concerns. Indeed, a social network analyst seeks to discover how their entities are created and connected to the social network. The social network analysts believe that the success or failure of a community depends on the structural patterns in the social network graph [1]. SNA fields are divided into descriptive and predictive, and they can focus on the links or social networks' entities. Link prediction is a prediction issue focusing on the links. Link prediction is a sub-branch of social network analysis being used in other fields as recommender systems, molecular biology and criminal researches. This is the only sub-branch of social network analysis focusing on the links instead of focusing on the entities. This makes the link prediction attractive and makes it distinguished from other data mining domains. Link prediction is a sub-set of link mining [2]. Link mining is a subset of data mining. Our field of study is to predict the network of collaborator writers. In these networks, the authors represent the network nodes and their collaboration is shown as link in the network graph. These networks show the collaboration between the papers' authors.

There are few studies on link prediction using the content published by nodes to predict the link. For this reason we have developed an algorithm based on the content published by nodes. To define the problem, suppose a snapshot of a social network, can we infer which new interactions among its members are likely to occur in the near future? We consider this question as the link prediction problem. This issue focuses on the links

* Corresponding Author

between the entities in network. Indeed, the collaboration prediction between the entities in the network is called link prediction. The goal of link prediction is to estimate the probability of creating links between the nodes in social networks [3], [2]. This estimation could be performed via analyzing attributes of entities and network structure. The meaning of network structure is the structure of other links in the network. In fact, the structural characteristics of the social network are the same as the topological properties of the network graph. In link prediction, the target is predicting links at time t_2, while we have links at time t_1 [4]. Any link prediction algorithm gives a score to a non-existing link and this score shows the probability of existence of the link at time t_2 and it calculates the similarity between the nodes that are in the end of the link. All the nonexistent links will be sorted in a descending order according to their scores, and the links at the top of the list are most likely to exist in the future [5], [2]. Considering a social network G (V, E) at time t1, where V is the set of nodes and E is the set of edges. The interaction between nodes u ,v is shown as edge e, as e ϵ E. Link prediction is defined as: The link prediction algorithm only by accessing the graph of the time interval t_1, should predict existence of the edges that exist in time interval t_2, but they haven't existed in time interval t_1. As $t_2>t_1$, set t1 is called training set and set t_2 is called test set.

In this study, different methods of link prediction have been analyzed and investigated and a content-based method has been presented for link prediction in co-authorship network.

2. Literature Review

From a specific view, link prediction methods are divided into four categories [6]: Node-based methods, topology-based methods, social theory-based methods and learning-based methods.

2.1 Node-based Methods

The calculation of similarity between a node pair is a solution for link prediction. This solution is based on a simple idea: nodes that are more similar to each other are more likely to have a link between each other. Indeed, people tend to create relationship with people who are similar in educations, religions, interests and locations. In this method, the similarity between non-connected pair of nodes in a social network is computed. This method is based on the criterion to analyze the proximity of nodes. Each pair (x,y) has a score and higher score means x , y are more likely to communicate with each other in the future,while lower score means that the nodes are more likely to have no link in the future. Thus, a list of scores in descending order will be achieved and the links at the top are most likely to exist in future. By this list, we can predict the links that will be created in future [6].

In a social network, a node has some attributes such as a profile in online social networks, mail name in e-mail networks and a series of published articles in scientific social networks. The information is used directly to calculate the similarity of two nodes. Since in most cases the values of node's properties are textual, typically, text-based and string-based similarity metrics are used. Papers [7], [8] have discussed about these criteria in details.

The authors [9] have defined a tree model to study the keywords of the user profile. They have used the distance between the keywords to estimate the similarity between the node pairs. They also have shown that by increasing the number of friends and keywords, the average similarity between the user and his friends decreases.

The authors [10] have found that most user profiles in current social networks are missed. To overcome this limitation, they have proposed a method, using stronger profiles, to infer some of the lost values, before calculating similarity.

The authors [11] have used overlapping user interests to measure the similarity. User interests are inferred from the actions they take, such as editing an article in Wikipedia. All actions that a user does can be shown as a vector, then, the similarity between two users will be obtained via the cosines similarity of their vectors.

Generally, node-based metrics use the attributes and actions reflecting the user interests to calculate the similarity between node pairs. These methods are useful if we can access to the user profile information and his performance, or we can infer them.

2.2 Topology-based Metrics

Even in networks where no information is available of nodes and edges, we can calculate the similarity between nodes by many other criteria, because the majority of criteria are based on graph topology and they don't need to know the attributes of nodes and edges. The graph structural attributes are defined in details in [12]. These methods are called similarity-based criteria. These methods use simple algorithms that, at the worst state, have time complexity of O (n^3). According to the attributes of these criteria, we can divide them into three groups of neighbor –based criteria, path-based criteria and random walk-based criteria [6].

In social networks, people tend to create new relationships with people that are closer to them. It is clear that the neighbors are the closest people to a social network user. For this reason, many neighbor-based criteria are developed by researchers for predicting links that will exist in future. For example, an algorithm called common neighbors is developed by the authors of paper [13]. In this algorithm, to estimate the similarity between the nodes, their common neighbors are computed. Other neighbor- based algorithms are generalizations of this algorithm. For example, the authors of paper [14] have introduced Adamic/Adar criterion to compute the similarity between websites. Paper [4] shows that Adamic/Adar (AA) is one of the best link prediction methods. After the extraction of the attributes of web-

pages, they will give the higher weight to the common attributes of two websites that is rare. It means that the attribute that is common between just two websites takes the highest weight. This criterion can be generalized to common neighbors of two nodes. In this way, the common neighbors between the two nodes, which are rare and shared only between the two nodes, take higher weight. Indeed, the common neighbors between the node pair that has lower degree take higher weight. Whenever the network studied is a co-author network, if an author has many co-workers, the probability of being shared between the nodes will be greater, but this probability will be very low for the nodes with lower degree.

This criterion is presented as [14]:

$$AA(x,y) = \sum_{z\in\Gamma(x),\Gamma(y)} \frac{1}{\log|\Gamma(z)|} \quad (1)$$

Paper [4] shows that Adamic/Adar (AA) is one of the best link prediction methods.

In addition to node-based criteria and neighbor –based criteria, we can use the path between two nodes to estimate similarity between node pair. For example, authors of paper [15] have established Katz based on the influence of all paths. This criterion [15] counts all paths between all pairs of nodes. In this criterion, we can give more weight to shorter path, because it is obvious that: the longer the length of the path, the less impact will have in linking the nodes. The formula of this criterion is represented as [15]:

$$Katz(x,y) = \sum_{l=1}^{\infty} \beta^l . |path_{x,y}^l| = \beta A_1 + \beta^2 A_2 + \beta^3 A_3 + \cdots \quad (2)$$

In this formula, $Path_{x,y}^l$ is the set of all paths from x to y that have length 1 and $\beta > 0$. If β is very small, this criterion will act similar to common neighbors, because long paths are not included in the calculation.

There are other criteria estimating the similarity between nodes via paths between them. Papers [16], [17], [18] have developed path-based criteria for link prediction.

The social relations between the social network nodes can be modeled using random walk. There are some methods computing the similarity between nodes in social networks based on random walk. These methods by defining a special destination for a random walk, from a special node, use the probability of going to the neighbors for prediction. For example, hit time (HT) is expressed by [19]. In this algorithm, the similarity between x,y nodes is estimated using calculation of the required walk number for a random walker, from the node x to node y. The smaller this number means the two nodes are more similar to each other. This method is formulated as [19]:

$$HT(x,y) = 1 + \sum_{w\in\Gamma(x)} P_{x,w}\, HT(w,y) \quad (3)$$

In this equation, Pi,j is the probability of going from node i to node j. Matrix P is defined as: $P=D_A^{-1}A$. In this formula, D_A is the diagonal matrix A in which $(D_A)_{i,i} = \sum_j A_{i,j}$. Clearly, the smaller this value is, the more similar the two nodes. So, in order to obtain the similarity between nodes, we multiply the value in negative.

The other random walk-based methods are expressed in papers [20], [21], [22], [23].

2.3 Social Theory-based Criteria

Social theory-based criteria can improve the efficiency of link prediction using additional information about social relationships. These methods are particularly suitable for large-scale social networks. In recent years, many researchers have applied old social theories, such as community, triadic closure, strong and weak ties, homeomorphisms, and structural balance, for analyzing and exploring social networks.

In paper [24], by considering user interest and user behavior, topology information is combined with community information. In this study, tweeter dataset is used for link prediction. They have shown that this method could improve the link prediction efficiency in directional, big scale networks.

In a paper [25], a link prediction model has been developed based on weak ties. It also uses three characteristics of the centrality of common friends, such as centrality, proximity, and betweenness. Each common neighbor, depends on their centrality, plays a different role in probability of communication between nodes. The weak tie is also considered for improving prediction accuracy. This model can be defined as follows [25]:

$$LCW(x,y) = \sum_{z} \big(w(z).f(z)\big)^{\beta} \quad (4)$$

F(z) is the switch function, and if z is common neighbor of x, y nodes, its value will be 1, otherwise its value will be zero. W(z) defines the centrality value of a node. β Parameter can moderate the quota of each common neighbor in probability of connecting two nodes. It is obvious that when this parameter is greater than 1, larger centrality values will be much more effective than smaller centrality values. When this parameter is less than 0, it restrains and prevents impacts of larger centrality values more than lower centrality values.

There are other social-theory based methods for link prediction. The authors of papers [26], [27], [28] have proposed some social theory-based criteria for link prediction.

2.4 Learning-based Methods

In recent years, many methods are presented based on learning. These methods use the external information and the attributes provided by algorithms considered in the previous sections for link prediction. These methods also create a training mechanism for prediction and consider special patterns that are special for each graph, in prediction. These algorithms have better efficiency compared to the previous algorithms, but due to the time-consuming training phase, they have high time complexity and sometimes they couldn't be applied on large networks.

For example, in paper [29], Support vector machine (SVM) is used for link prediction. The performance of this algorithm is in this way that each sample of paired nodes taken will be mapped to a point in the space. These samples have positive or negative label. So, two classes with empty gap are in the space. Now, the samples entering, based on their closeness to the classes, will be mapped to a class. This algorithm plots some hyper planes and attempts to extract a hyper plane showing the distinction between the classes better.

Generally, these methods are performed using a link prediction training mechanism.

3. Problem Solving Method

In this section, we explain the proposed method to extract people interest and method of interest vector formation for each author.

Generally, this section consists of six parts. First section is network preparation, as the algorithm can be applied to it. Second section is language processing of the network's content. Third section is extraction of papers' keywords. Fourth stage is extraction of the papers' topics and forming subject vector for papers. In the fifth stage, authors' study fields are extracted. Finally, in the sixth stage, based on the fields of authors, link prediction is performed. In the following, each of these sections is explained separately.

3.1 The Preparation Method of Network

The network that is used for this study is co-authorship network of SID site. This network consists of raw data in XML format. For each year, there is a XML file. At first, these files are integrated with each other, then to use this network, it is required to extract the graph of training time interval and testing time interval. The graph of training set includes papers' authors and their links during 2000-2005. The testing graph includes papers' authors and their links during 2006-2012. In this stage, different types of mapping are performed. These mappings include:

The mapping of papers to corresponding XML file: since this network is stored in XML format, using this mapping, the location of the article can be obtained in the corresponding XML file. By this mapping, we can easily get other information about the articles, such as keywords, relevant organization, the link to PDF format of the paper and gathering time.

The mapping of paper to the authors: Using this mapping, we can achieve the authors of each paper based on their ID.

The mapping of authors to the papers: Using this mapping, we can achieve the papers written by each author.

The mapping of papers to the papers' summary: Using this mapping, we can achieve the abstract of each paper.

Mapping papers to the keywords of papers: Using this mapping, we can achieve the keywords of each paper's abstract.

The mapping of papers to the root words of papers' keywords: Using this mapping, we can achieve root of the keywords in papers' abstract.

For these mappings, we use some Hash Map functions. By defining key and value for these functions, mappings are performed.

3.2 Language Processing of the Network Content

In this section, the abstract of each paper is processed by language processing to be used in the next steps for extracting articles from articles. In this stage, Persian processing tool, developed by [30] in telecommunication research center, is used. This system can do all the necessary actions for different layers of Persian language processing, from its initial layer which is lexical layer up to the upmost layer which is syntax. This Toolkit performs a combination of these processes: normalization, tokenization, Spell checker, morphological analysis, Persian Dependency Parser, Semantic Role Labeling. This toolkit, by receiving the Persian raw data, performs semantic and morphological analyses. These analyses include normalization, Tokenization, stemming and Lemmatizing. Thereupon, by adding this information to raw text, by Dependency Parser ParsiPardaz, Dependency Parse Tree is generated for Persian sentences.

The processes we've been using with this toolkit include the following:

Text normalization: The characters of words that are in the abstract are normalized in this stage. For example, converting Arabic ی to Persian ی or converting the characters "إ"، "أ"، "آ" to a unified form "ا". As the same way, we convert the words "سِر" و "سُر"، "سَر" to a uniform word "سر". Before any language processing, the upper level of this conversion should be performed. It means that the similar characters should be unified. This challenge that exists in Persian language is called Unicode Ambiguity. To solve this problem, Lemmatizing is used to perform normalization task.

Tokenization: In this stage, text sections are defined. For this stage, Persian language processing toolkit, developed by [30] in telecommunication research center, is used. This toolkit performs text tokenization based on syntactic dependency rules and some semantic and syntactic features. In this method, all compound words are connected by hemi-space. For example, instead of" آمده است", آمده‌است" is used. Since in the next steps, the extraction of subjects, the border between words is specified by a Space character, thus, tokenization is used which specifies the boundary of words by putting together compound words with hemi-space and putting the Space character, the border of words is defined.

These two tools are in the first level of Persian Language processing toolkit ParsiPardaz, lexical layer.

3.3 Keyword Extraction

In this stage, Stop words, the words that are common and not useful, will be eliminated from the abstract. At

first, all punctuation marks are eliminated from the text. Then, stop words will be deleted from the abstract's words. To extract stop words, the different steps have been taken. These steps include:

Eliminating common Persian words: There is a list of common Persian words that using this list consisting 500 words, a part of common words has been eliminated from the abstract [31].

Eliminating some respect words such as Engineer, Doctor, etc.

Eliminating some verbs gaining a list of common words among all papers using tf/idf method: This step, depending on the textual content of each network, can provide a different list of words to the user.

Extracting root of keywords: in the next steps, keywords of abstract are used in two ways: Once by considering keywords without the root of words and another time by considering their root. we can use root of the words instead of the words, as keywords. In this way, some words like subject and subjects are considered to be the same and it will be effective in calculating similarity between vectors of people's study field. To extract the root of words, stemming, the developed toolkit in telecommunication research center is used [30]. Stemmer tool is located in the second level of this toolkit, Morphology Layer. This tool applies the word structure to extract the root of words and acts independent from its content.

This step is one of the important sections in our algorithm.

3.4 Extracting Articles' Topic

In this section, we label all existing articles in dataset with the extracted topics. Indeed, for each paper, its interference with the topics is calculated. The set of documents is denoted by D and the set of words in the domain is denoted by V. In this step, this set includes the extracted keywords in the previous step, because the input words of the LDA algorithm are more useful and show the concept of the document more effectively, the algorithm will perform better. A denotes the set of authors and H denotes the set of extracted topics in this section. Each author participates in writing some of papers that are member of the D set. Each author that is a member of A set collaborates in writing the set of papers and this set is denoted by D_a and a denotes the code of author of this set of papers. To extract the subject of papers the developed LDA algorithm in paper [32] is used. LDA is used due to its superiority compared to the similar methods of topic extraction. This topic model is used on discrete sets such as textual sets. LDA is applied on many topics such as Collaborative Filtering, Text Classification, Word Sense Disambiguation [33] and Community Recommendation [34]. LDA is a Generative Model for the text and other Discrete Data Collections. In the context of textual modeling, this model claims that each document is produced in a combination of subjects. So, this algorithm returns interference level between documents and subjects as output by taking the document text and the number of requested subjects.

This algorithm defines a matrix $|V| * K$ for each paper. The elements of this matrix are consistent with the initial values of relevance between the words and topics. These values are considered similar for all subjects. The optimal value of this parameter will be investigated in the next chapter. This algorithm defines a matrix for each word with dimensions $|D| * K$. The elements of this matrix are consistent with the initial values of relevance between documents and topics. These values are considered similar for all topics. The optimal value for this parameter will be investigated in the next chapter.

LDA considers each document as polynomial distribution on topics. For each word existing in the document, it gives the topic distribution on the words, it shows a matrix in which rows are words and columns are topics. K is one of the inputs of algorithm. It shows the number of topics that we expect the algorithm to extract. If we define k topics, a matrix $|V| * K$ is defined showing the distribution of topics on the existing words in the domain of words. This matrix is denoted by φ. The probability of existence of the word w on topic k_{th} is denoted by $\varphi_{w,k}$. This probability values are achieved by applying LDA algorithm to the initial matrix, several times. The optimal value of the number of these iterations will be investigated in the next chapter. Then, using this distribution, the distribution of document on the topics is computed using the used words in this document and distribution of topics on these words. Indeed, a matrix with dimensions $|D| * K$ is defined. The rows of this matrix are the papers and the columns are topics. Each element of this matrix indicates the probability of belonging an article to a special topic. This matrix is denoted by θ. The probability that document d_i is related to k_{th} topic is denoted by $\theta_{i,k}$. For each paper the sum of these values is 1. The values are achieved by applying LDA algorithm on the initial matrix, several times. The optimal value of these iterations will be investigated in the next chapter. In this study, we apply LDA to the extracted keywords of the existing papers in the network to have a set of users' interests and to increase the accuracy of link prediction algorithm. Generally, the fewer the number of defined topics is, in each topic there are more concepts, and the topics will therefore be more general and the more the number of topics is defined, the less the concepts within these topics will be and the topics will be more specific. The effect of defined topics will be evaluated in the next chapter.

Generally, for each social network user and for each paper, we define a vector denoting the topic distribution of the paper. Then, topic distributions are used to achieve the study fields of authors and then we use the similarity estimation of authors' study field for link prediction. This topic distribution is shown as a matrix that its elements show the thematic interference of papers with the defined topics. Finally, as an output of this step of the algorithm, we have the subject vector of the same number of articles

existing in the data set and the elements of these vectors show the interference of document with extracted topics. These vectors are extracted from matrix θ achieved by LDA algorithm. Indeed, each row of this matrix represents the corresponding vector of a paper.

For example, the topic vector is defined for d_i paper as:

$$S_i = [S_{i,1}, S_{i,2}, \dots, S_{i,|H|}]$$

These values show the similarity between document d_i and different topics. These values are between zero and one and define the relevance of document d_i to different study fields. Also, sum of the values of these elements is one.

3.5 Extracting Authors' Study Fields

In this step, using the vectors extracted in the previous step, for all authors, we make an interest vector. The elements of these vectors show the interest of the given author in the given study field. The number of elements in these vectors is equal with the number of elements in subject vectors of the papers. Indeed, using the topics of each paper, the study field of each author could be defined. Accordingly, the study field of each author is derived from the average of the subjects of all articles written by him. To extract the study field of authors, various methods, such as maximizing between the subjects of articles written by author, have been evaluated and this method has been the best.

It should be noted that the number of extracted topics for all articles and authors is the same and the topics are similar. For example, if 100 topics are defined for each paper, all authors have these 100 topics. The relevance of each author to each topic is computed using the probability number assigned to that subject for articles written by that author. Each person is the author of some papers. To achieve the activity of authors in different study fields, we compute the mean of the corresponding elements in the subject vector of the articles published by this author and we put them in a vector called Interest Vector. The interest vector is defined for the author a_i as:

$$I_i = [I_{i,1}, I_{i,2}, \dots, I_{i,|H|}]$$

These values show the belonging of a_i author to different topics.

The elements of vector I_i are computed as follows: For example, the first element of a_i authors' interest vector, if this author participates in writing the papers d1, d2, d3, d4, it is computed as:

$$I_{i,1} = \text{AVG}\,(S_{1,1}, S_{2,1}, S_{3,1}, S_{4,1})$$

Finally, this step's output is the amount of the authors' activity on extracted topics, as interest vectors for the authors. In the last section of this chapter, using these vectors, the level of similarity between the authors is estimated to predict the collaboration between them.

3.6 Link Prediction Based on the Similarity of Authors' Study Fields

In this method, the probability of creating link between two authors who have not previously collaborated is consistent with the similarity between study field vectors of these two authors. To calculate the similarity, Cosine Similarity formula is used. Other methods as Euclidean similarity have been evaluated and finally, Cosine Similarity has generated the best results. Cosine Similarity is a similarity criterion between two vectors which calculates the cosine of the angle between two vectors. Zero's cosine is equal to one, thus if two vectors are match in each other, their similarity is equal to one. It is clear that this value shows the highest similarity between two vectors. Indeed, if the interest vectors of two authors are consistent, the similarity of these two authors is considered as 1. For any other angle, this value is less than 1. If two vectors with angle 90 degree are in the space, their cosine similarity is zero. It is obvious that this value shows the lowest similarity between the vectors. Since cosine similarity is computed in positive space, the similarity between vectors will be between zero and one. In information retrieval and Text Mining, this criterion is used to estimate the similarity between document vector and query vector or the similarity between the vectors of two documents [35]. Also, in data mining, this method is used to estimate the coherence of clusters [36]. The reason to use cosine similarity is that this criterion is effective on evaluation, especially for sparse vectors, because only non-zero values are considered. Since the interest vectors of authors are sparse, we use this criterion. Cosine similarity of two vectors is computed as [35]:

$$\text{CosineSim} = \frac{\sum_{i=1}^{K} A_i * B_i}{\sqrt{\sum_{i=1}^{K} (A_i)^2} * \sqrt{\sum_{i=1}^{K} (B_i)^2}} \quad (5)$$

In this formula, A, B are study field vectors of two authors. K is the number of topics.

Based on the following formula, we compute the similarity between two authors with each other:

$$\text{Score}(x, y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{\text{cosine}(x, y)}{\log(|\Gamma(z)|)} \quad (6)$$

According to this formula, if there is much similarity between the topics that the two authors are interested in and the degree of the common node between two authors is low, they will be more similar to each other and it is highly probable that they will establish a link in future. The reason is that the lower the degree of the common author, the better it is and it shows that the mentioned node has higher similarity to these two authors, rather than the case in which common author has many common authorships. This criterion is the combination of content similarity between the nodes with Adamic- Adar criterion. Indeed, in Adamic- Adar criterion, the content similarity of the authors is not involved in link prediction.

This method is called Structure Topics Prediction.

4. Experiments and Results

We used a useful dataset in order to applying our algorithm to this. The students of DBRG Lab, a Lab in Tehran University, had produced a dataset that is extracted from co-authorships between authors of existing papers in Academic Jihad Scientific Information Database. The developers of this dataset have named it as SID. The generated graph of this dataset is not weighted and not directed. This dataset is related to articles from 1379 to 1390. In order to link prediction we divided the data into two periods of training and testing. These two periods are as follows: collaborations between the authors from 1379 to 1374 - collaborations between the authors from 1385 to 1390. The output of a link prediction algorithm is a sorted list of scores allocated to links not existing in the training time graph. One of the methods being used to evaluate the results of link prediction algorithms is the area under Receiver Operating Characteristic (ROC) curve [37]. This method is called AUC (Area under Receiver Operating Characteristic). The horizontal axle of ROC chart shows the false positive links and in the vertical axle, true positive links are considered. The false positive links are called FP and the number of true positive links is called TP. In this method, an algorithm has a better output that gives higher score to the links created in the testing time interval than the links not created in this time interval. The number of links not existing in training set is denoted by n. Also, n_1 is the set of the links created in the test set and n_2 is the set of links not created in the test set. We achieve all pairs $n_2 * n_1$ and represent the number of them with k. If in m number of k pairs, the score given to the existing link is higher than the score given to the non-existing link and in b numbers, it is opposite. AUC criterion is achieved as [38]:

$$\text{AUC} = \frac{\left(m + \frac{b}{2}\right)}{k} \tag{7}$$

Another method for evaluating link prediction algorithms is Area under Precision-Recall Curve (AUPR). This criterion uses the area under Precision-Recall chart. The precision and recall are defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{8}$$

$$\text{Recall} = \frac{TP}{TP + TN} \tag{9}$$

In fact, precision is equal to the percentage of predicted links that are correctly predicted and recall is is equal to the percentage of the links generated in the test interval that are predicted. Indeed, in recall points, the precision of the algorithm is measured and the chart is plotted.

In the following, we compare the proposed algorithm with other algorithms in the domain of the link prediction using the criteria mentioned.

Based on the results achieved from previous sections, we compare the results of proposed algorithm with other structural link prediction algorithms. In the following Table, for Katz algorithm, parameter β is 0.005 and the maximum distance from the source node is considered as 5. For RootedPageRank, the probability of return to the previous node is considered as 0.5. For StructureTopicsPrediction algorithm, α is 0.5 and β is 0.001, the number of topics is 150 and the number of iterations of LDA algorithm is 200.

Table 1. the results of proposed algorithms with other algorithms

Algorithm	AUPR	AUC
StructureTopicsPrediction	0.033441885	0.735185072
Common neighbors	0.028207159	0.630440334
Adamicadar	0.034373253	0.712968505
Jaccard Coefficient	0.009238340	0.467782634
Distance	0.010419157	0.500000000
PreferentialAttachment	0.017296527	0.627418673
Katz (5-0.005)	0.029558192	0.648661437
RootedPageRank (0.5)	0.014508203	0.612446540

As shown, Adamic-Adar and Katz algorithms have good results. In the comparison of the proposed methods with other methods, we can achieve the following results:

The combination of content and structure can improve prediction outcomes. The results of proposed algorithm indicate this matter.

Our method can improve Adamic- Adar, as the best algorithm between the other algorithm, results about 3% and can improve Katz,s results about 10%. The combination of content and structure can improve the prediction results. The results of Structure Topics Prediction algorithm are good examples.

The best AUC criterion belongs to Structure Topics Prediction. According to the studies [37], the importance of AUC criterion is higher than the importance of AUPR and the algorithm with the better AUC has good results. Although AUPR criterion shows the area under precision-recall chart, but AUPR chart is effective on the comparison between the performances of algorithms. Because it is possible that an evaluation method has the best value of AUPR, but in some cases it has lower quality than the other algorithms and vice versa. For this purpose, the following diagrams are used to evaluate the results of the algorithms more precisely. AUC, P-R charts are plotted to compare the proposed algorithm with two algorithms having the best results.

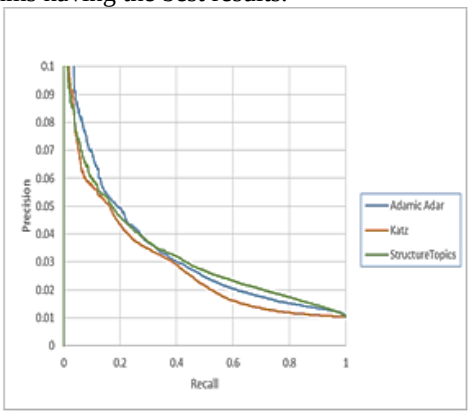

Fig. 1. ROC Chart

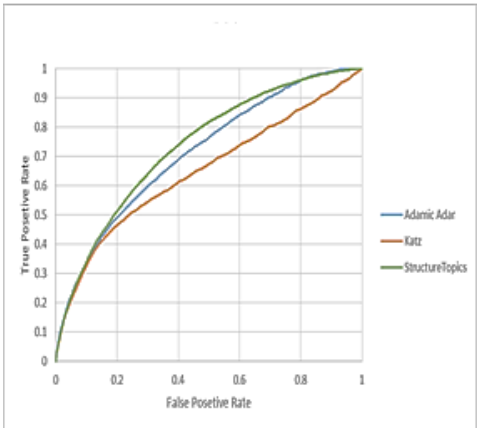

Fig. 2. P-R Chart

According to the results of these evaluations and the optimal adjustment of the input parameters of LDA algorithm, this method is compared with the existing methods in link prediction. The results of this comparison show that the combination of content and structure can increase accuracy of link prediction algorithms.

5. Conclusion and Further Studies

Based on the purpose of this study, the achievements of this study are: Different methods of link prediction have been studied. A link prediction algorithm based on the content and interests of people has been presented. The comparison between the performance of existing algorithms and presented algorithm in this study has been evaluated through several evaluation methods and finally, the outputs are analyzed.

In future studies, to improve the quality of algorithms, we can work effectively on extracting the subject of the content published by the network nodes and recover the network content effectively. For example, we can extract the keywords of texts exactly and to improve the content-based algorithm results, we can improve the extraction of keywords. Also, we can use the data fusion algorithms to combine the results of different algorithms and achieve better results. Also, the data fusion algorithms can be used in allocating study fields to the authors. Generally, to improve this section of algorithm, the extraction of authors' study fields from the topic of their study, we can make efforts.

It is possible that a link predicted by a method is created in a period after test set. Indeed, the link is predicted true but it is not created at appropriate time. The current methods for evaluating the accuracy of algorithms did not consider this matter. Considering this case, to some extent, could provide a more accurate assessment of the accuracy of the algorithms.

References

[1] F. Borko (Ed.), Handbook of Social Network Technologies and Applications, 2010 .
[2] L. Getoor, Christopher P. Diehl“ , Link Mining: A Survey ”, SIGKDD Explorations ,Vol.7 ,No. 2 ,pp. 3-12, 2005 .
[3] B.Taskar, MF Wong, P Abbeel, D Koller“ , Link prediction in relational data ”, Learning Statistical Patterns in Relational Data Using Probabilistic Relational Models , Vol.7 ,2005 .
[4] D. Liben-Nowell, J. Kleinberg“ ,The Link Prediction Problem for Social Networks ”,Journal of the American Society for Information Science and Technology , Vo.No.58,7 ,pp. 1019-1031, 2007 .
[5] X. Feng, Z.J., Xu K“ ,Link prediction in complex networks: a clustering perspective ”,European Physical Journal , vol.85 ,pp. 1-9, 2012 .
[6] P. Wang, B. Xu, Y. Wu, X. Zhou“ ,Link Prediction in Social Networks: the State-of-the-Art ”,Science China Information Sciences ,Vol.57 ,pp. 1-38, 2014 .
[7] V. StröEle, G. ZimbrãO, J. Souza“ ,Group and link analysis of multi-relational scientific Social Networks ”,Journal of Systems and Software ,Vol.86 ,pp. 1819-1830, 2013 .
[8] G. Rossetti, M. Berlingerio, F. Giannotti“ ,Scalable link prediction on multidimensional networks ”, in11th IEEE International Conference on Data Mining Workshops , Vancouver, Canada, 2011 .
[9] J. Mori, Y. Kajikawa, H. Kashima, “Machine learning approach for finding business partners and buildings reciprocal relationships ”,Expert Systems with Applications ,Vol.39 ,pp. 10402-10407, 2012 .
[10] W. Sen, J. Sun, J. Tang“ ,Patent partner recommendation in enterprise social networks ”,inthe 6th ACM International Conference on Web Search and Data Mining (WSDM'13) , Rome, Italy, 2013 .
[11] L. Aiello, A. Barrat, R. Schifanella,“ ,Friendship prediction and homophily in social media ”,inACM Transactions on the Web , 2012 .
[12] H. Chen , D.Miller, C.Giles“ ,The predictive value of young and old links in a social network ”, inthe ACM SIGMOD Workshop on Databases and Social Networks , New York, USA, 2013 .
[13] D. Davis, R. Lichtenwalter, N. Chawla“ ,Supervised methods for multi-relational link prediction ”,Social Networks Analysis and Mining ,Vol.3 ,pp. 127-141, 2013 .
[14] L. Adamic, and E.Adar “ ,Friend and Neighbors on the Web ”,Social Networks ,Vol.25 ,pp. 211-230, 2003 .
[15] P. Soares, R. Prudêncio“ ,Proximity measures for link prediction based on temporal events ”,Expert Systems with Applications ,Vol.40 ,pp. 6652-6660, 2013 .
[16] E. Richard, N. Baskiotis, T. Evgeniou,“ ,Link discovery using graph feature tracking ”,inthe 24th Annual Conference on Neural Information Processing Systems 2010 ,Vancouver, Canada, 2010 .
[17] S. Oyama, K. Hayashi, H. Kashima“ ,Cross-temporal link prediction ”,inthe 11th IEEE International Conference on Data Mining (ICDM'11 ,(Vancouver, Canada, 2011 .
[18] P. Ricardo ; Ricardo Bastos Cavalcante Prudêncio, “ ,Time series based link prediction ”, in International Joint

Conference on Neural Networks (IJCNN'12 ,(Brisbane, Australia, 2012 .

[19] E. Gilbert, K.Karahalios“ ,Predicting tie strength with social media ”, inthe SIGCHI Conference on Human Factors in Computing Systems ,Boston, USA, 2009 .

[20] J. O'Madadhain, J. Hutchins, P. Smyth“ ,Prediction and ranking algorithms for event-based network data ”,ACM SIGKDD Explorations ,Vol.7 ,pp. 23-30, 2005 .

[21] S. Brin , L. Page“ ,The Anatomy of a Large-Scale Hypertextual Web Search Engine ”,Computer Networks and ISDN Systems ,Vol.30 ,pp. 107-117, 1998 .

[22] R. Lichtenwalter, J. Lussier , N. Chawla“ ,New perspectives and methods in link prediction ”,in16th ACM SIGKDD International Conference of Knowledge Discovery and Data Mining ,Washington, DC, USA, 2010 .

[23] D. Dunlavy, T. Kolda, E. Acar“ ,Temporal link prediction using matrix and tensor factorizations ”,ACM Transactions on Knowledge Discovery from Data , Vol.5 ,pp. 1-27, 2011 .

[24] T. Kuo, R. Yan, Y. Huang, “Unsupervised link prediction using aggregative statistics on heterogeneous social networks”, inthe 19th ACM SIGKDD international conference on Knowledge discovery and data mining , Chicago, USA, 2013 .

[25] D. Yin, L. Hong, B. Davison“ ,Structural link analysis and prediction in microblogs ”,inthe 20th ACM International Conference on Information and Knowledge Management (CIKM'11 ,(Glasgow, UK, 2011 .

[26] R. Xiang, J. Neville, M. Rogati“ ,Modeling relationship strength in online social networks ”,inthe 19th International Conference on World Wide Web (WWW'10) , Raleigh, USA, 2010 .

[27] “ Z. Yin, M. Gupta, T. Weninger, J. Han” LINKREC: a unified framework For link Recommendation with user attributes and Graph structure ”, inthe 19th International Conference on World Wide Web (WWW'10) ,Raleigh, USA, 2010 .

[28] M. Sachan, R. Ichise“ ,Using semantic information to improve link prediction results in network datasets ”, International Journal of Computer Theory and Engineering, Vol.3 ,pp. 71-76, 2011 .

[29] C. Cortes, V. Vapnik“ ,Support-vector networks ”,Machine learning ,Vol.3 ,pp. 273-297, 1995 .

[30] Z. Sarabi,H.Mahyar,M.Farhoodi“ ,ParsiPardaz: Persian Language Processing Toolkit ”,2013 .

[31] “ http://www.ranks.nl/stopwords/persian ”,[intra-linear].

[32] Xu. HieuPhan,L.Nguyen,S. Horiguchi .“ ,Learning to Classify Short and Sparse Text&Web with Hidden Topics from Large-scale Data Collections”, inThe 17th International World Wide Web Conference, Beijing, China, 2008 .

[33] J. Boyd-Graber, D. Blei, X Zhu“ ,.A Topic Model for Word Sense Disambiguation ”, inthe 2007Joint Conf. on Empirical Methods in Natural Language Processing and Comp. Natural Language Learning ,2007 .

[34] W. Chen, J. Chu., J. Luan, H. Bai, Y. Wang, Y.Chang ,. “Collaborative Filtering for Orkut Communities: Discovery of User Latent Behavior ”, inInternational World Wide Web Conference ,2009 .

[35] S. Amit“ ,Modern Information Retrieval: A Brief Overview ”,Bulletin of the IEEE Computer Society Technical Committee on Data Engineering,, NO. 244 ,pp. 35-43, 2003 .

[36] P. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, 2005 .

[37] D. Hand“ , Measuring classifier performance: a coherent alternative to the area under the ROC curve ”,Machine learning ,pp. 103-123, 2009 .

[38] J. Davis, M. Goadrich“ ,The relationship between Precision-Recall and ROC curves ”, inthe 23rd international conference on Machine Learning , 2006 .

Hosna Solaimannezhad received the B.Sc. degree in Information Technology engineering from University of Sepahan , Isfehan, Iran, in 2010. She received the M.Sc. degree in Information Technology engineering from Tehran University, Tehran, Iran, in 2015. She is currently Ph.D student in Department of Electrical and Computer Engineering, Tehran University. Her area research interests include Data Mining, Information Retrieval, Big Data, Distributed systems and Social Networks. Her email address is:

Omid Fatemi received the B.Sc. degree in Electrical engineering from Tehran University, Tehran, Iran, in 1989. He received the M.Sc. degree in Electrical engineering from Tehran University, Tehran, Iran, in 1991. He received the Ph.D degree in Electrical and Computer engineering from University of Ottawa, Ottawa, Ontario, Canada, in 1999. Now, he works as assistant professor in Department of Electrical and Computer Engineering, Tehran University. His area research interests include Big Data, IT Governance, E-Learning and cloud Computing.