Overcoming the Link Prediction Limitation in Sparse Networks using Community Detection

Mohammad Pouya Salvati
Faculty of Electrical and Computer Engineering, Urmia University, Iran
pouya.salvati@yahoo.com
Jamshid Bagherzadeh Mohasefi
Faculty of Electrical and Computer Engineering, Urmia University, Iran
j.bagherzadeh@urmia.ac.ir
Sadegh Sulaimany*
Faculty of Computer Engineering, University of Kurdistan, Iran
S.Sulaimany@uok.ac.ir

Received: 01/Jul/2020 Revised: 15/May/2021 Accepted: 05/Jun/2021

Abstract

Link prediction seeks to detect missing links and the ones that may be established in the future given the network structure or node features. Numerous methods have been presented for improving the basic unsupervised neighbourhood-based methods of link prediction. A major issue confronted by all these methods, is that many of the available networks are sparse. This results in high volume of computation, longer processing times, more memory requirements, and more poor results. This research has presented a new, distinct method for link prediction based on community detection in large-scale sparse networks. Here, the communities over the network are first identified, and the link prediction operations are then performed within each obtained community using neighbourhood-based methods. Next, a new method for link prediction has been carried out between the clusters with a specified manner for maximal utilization of the network capacity. Utilized community detection algorithms are Best partition, Link community, Info map and Girvan-Newman, and the datasets used in experiments are Email, HEP, REL, Wikivote, Word and PPI. For evaluation of the proposed method, three measures have been used: precision, computation time and AUC. The results obtained over different datasets demonstrate that extra calculations have been prevented, and precision has been increased. In this method, runtime has also been reduced considerably. Moreover, in many cases Best partition community detection method has good results compared to other community detection algorithms.

Keywords: link Prediction; Sparse Network; Clustering; Time efficient.

1- Introduction

As networks grow, link prediction greatly helps our trade and communication in many large-scale online commercial and social networks. Besides attempting to find missing links, link prediction also seeks to predict new links that may establish in the future. It is precious in a complex network to predict this category of links. On the other hand, high costs are required in laboratories to detect new or missing relations or links for some networks, such as protein-protein interaction relations. Clearly, prediction of correct links in such networks can play a pivotal role in treatment of many diseases such as AIDS and cancer. However, these networks are almost imperfect, low-density, and sparse. Also, practical experimentation to correct them, especially for biological networks, causes high costs to incur.

Link prediction can predict and subsequently improve the structure of the networks[1]. Many prediction methods have been presented, attempting to improve prediction results. Many of the available networks are sparse, which causes high extra calculation. This means that number of zero entries that needs to be scored in the associate adjacency matrix are far more than the existing ones, in computation and loss of time and resources. To the best of our knowledge, this issue has been mentioned implicitly or explicitly in some researches, but the appropriate solution has not been found [2][3].

This paper seeks to present a new, more accurate approach for link prediction in sparse networks. Regarding the main pitfalls of sparse networks for link prediction, we reduce the time consuming computations in addition to improve the precision as well. Eliminating the extra computations will be possible by removing the unnecessary predictions that do not have significant effect on the main results. We will achieve this aim by clustering the nodes and localizing the computations on the compacted parts of the network.

* Corresponding Author

After that, we consider some effective strategies to implement between clusters link predictions. The proposed method can be used for both, predicting new links or finding missing links correctly, especially in sparse networks, somehow as the networks are sparse, the result becomes better. We may use the terms clustering or community detection interchangeably throughout this paper.

The rest of the paper is organized as follows. In section 2, the related works are illustrated. After that in section 3, the proposed method and the evaluation are explained. In section 4, results and discussion are reported, and finally, in section 5, future work and conclusion will be discussed.

2- Related Works

We review the related researches about link prediction using community detection and link prediction for sparse networks, in this section, after a short overview of the primary related concepts. Link prediction methods have mainly two major categories: unsupervised and supervised. There are several unsupervised methods where the score $score(x,y)$ is considered for each pair of nonexistent links. Clearly, the higher the score, the greater the probability of establishment of a link is. The methods are divided into two broad categories: neighborhood-based and path-based methods [4]–[7]. It is worth mentioning that we use the neighborhood-based methods, we will refer to them as basic methods, including CN, JC, AA, RA, and PA. The full name and ranking formula for the methods are shown in Table 1. It is popular for new ideas to be tested with basic methods.

Table 1. Different scoring functions for neighbourhood-based unsupervised link prediction. $\boldsymbol{\Gamma(x)}$ is the set of neighbours of node x, and $|\boldsymbol{\Gamma(x)}|$ is the number of neighbors of node *x*

Neighborhood-based	**Common Neighbors (CN)**	$score(x,y) = \lvert\Gamma(x) \cap \Gamma(y)\rvert$
	Jaccard Coefficient (JC)	$score(x,y) = \frac{\lvert\Gamma(x) \cap \Gamma(y)\rvert}{\lvert\Gamma(x) \cup \Gamma(y)\rvert}$
	Adamic-Adar (AA)	$score(x,y) = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log(\lvert\Gamma(z)\rvert)}$
	Resource Allocation (RA)	$score(x,y) = \sum_{c \in x \cap y} \frac{1}{\lvert\Gamma(c)\rvert}$
	Preferential Attachment (PA)	$score(x,y) = \lvert\Gamma(x)\rvert \times \lvert\Gamma(y)\rvert$

2-1- A Review of link Prediction using Community Detection

The community structure can be observed in many of the available networks. The notion of community or cluster depends largely on the type of the network or the information it contains[8]. In a metabolic network or an inter-protein interaction (protein-protein) network, for instance, a community can be a series of adjacent proteins that perform a biological operation inside a cell [9]. In a commercial network, a cluster can be a series of customers with similar purchasing backgrounds or similar tastes [10]. On the web, a cluster can be a series of pages about a certain issue [11].

In [12], the number of links between every two nodes $\boldsymbol{u}$ and $\boldsymbol{v}$ is calculated, and is normalized using the number of possible links between them. This value is referred to as the probability that there is a link between the two nodes $\boldsymbol{u}$ and $\boldsymbol{v}$. The drawback of this method is that prediction is made among all the nodes, and the network undergoing prediction is not necessarily sparse. Another cluster related link prediction type involves the stochastic block model [13], [14]. In this type of model, all the nodes are summed for categorization. The probability that two nodes are connected is obtained based on their membership in the relevant clusters. The most significant disadvantage of these methods is that they are impractical for large-scale networks due to the high time complexity of obtaining the optimal clustering.

In a similar study conducted in 2012 [15], community information has been used differently for prediction as a characteristic of the nodes. The major drawback of these methods involves the high time complexity of obtaining a clustering in large-scale networks and computation for both nodes. A method referred to as the spectral algorithm has been presented in [3]. The approach is similar to a semi-local method, which uses neither local information nor general network paths, making it highly time-consuming and infeasible in large networks.

The authors of [2] have proposed a distributed method based on clustering for link prediction, which depends on Google's MapReduce technology. Although, it has been mentioned in the paper's abstract that clustering is performed basically on dispersed vertices, so that they are grouped in an integrated fashion, the paper does not claim that it applies to sparse graphs.

2-2- A review of link Prediction in Sparse Graphs

Although numerous works on link prediction have been presented that have attempted to improve the precision of the results, the sparsity has been slightly considered in the works. However, many real-world networks are sparse, which causes poor prediction results and loss of time. Hence, the question in some works since 2013 is what is the best way of avoiding this issue [3]. In a supervised solution, the method used in [16] has utilized incidence rather than adjacency matrix factorization, demonstrating that the incidence matrix factorization (IMF) method performs better than adjacency matrix factorization (AMF) in a sparse matrix as well.

It has been mentioned in [17] that the available link prediction algorithms have focused on triangular structures. The method exhibits low efficiency over sparse tree networks. A method based on network degree heterogeneity has been presented in that paper. As authors stated, however, they have examined only tree structures, whereas many complex networks in the real world are sparse, and do not necessarily contain tree structures. In [18], it has been assumed that social network users' habits and characteristics correspond to their social communication on the networks. That is, links are predicted through the notion of aligned social networks.
Besides, in [19], the focus is mainly on the structure of the network, and the paper models the problem based on intrinsic characteristics of the network. A drawback of these models is the cost of training the model to handle the big data. Another interesting research used for sparse networks is [20], where the relationship between clustering and the precision of the methods has been investigated based on network structure.
Even though, all the unsupervised link prediction methods mentioned above attempt to improve results, reducing the extra computation in sparse networks by splitting it into separate communities and so improving the results in this way has not been considered yet. In this article, we try to overcome the poor result of link prediction in the sparse networks by dividing networks into multiple communities and concentrating on the inter and intra community computations. Subsequently, we will shorten the execution time of link prediction in the sparse networks also.

3- Proposed Method

The algorithm presented in the proposed method involves three major phases, and the validity of each phase affects the final results. The first step is data preparation and pre-processing, which is explained in the next paragraph. Other steps are clustering the network into some partitions and performing the link prediction inter and intra communities, and integrating the results according to some specific policies. The steps are mentioned below.

3-1- Data

Since some datasets are directional, we need to convert them to un-directional graphs because of the nature of the basic link prediction algorithms that do not consider the direction of nodes [21]. First, the dataset is mapped to a matrix, then the matrix needs to be symmetric, and the elements on the main diameter needed to be zero. In this research, five datasets have been used for experimentation. Email[1] (the email communications at the Rovira i Virgili University), the collaboration network on high-energy physics[2], the collaboration network of co-authors on physics-related topics on the arXiv website[3], the communication network of associated words[4], and the communication network of human protein[5]. Table 2 describes the properties of each data set, respectively.
The quality and precision of link prediction in this research depend to a large extent on correct cluster detection. The utilized clustering methods are as follows. Fast unfolding [22] is a link-based community detection algorithm. Link-community [23] which finds communities such that it may contain nodes overlapping others. Another method used in this research involves the InfoMap community detection algorithm [24], [25]. The Girvan-Newman algorithm utilizes the edge betweenness feature [26].

Table 2: Examined networks and their basic properties

Network	Nodes	Edges	Mean clustering coefficient	Density
Email	1133	5452	0.22	0.0085
Collaboration network on high-energy physics	9877	25998	0.47	0.0005
Collaboration network on general physics communication	5242	14496	0.52	0.001
Network of associated words	23219	305500	0.099	0.001
Wikipedia's network of manager selection	7115	100762	0.14	0.003
Human protein communications	30047	41327	0.101	0.00009

3-2- Cluster-Based Sparse Link Prediction (CBSLP)

For easy referencing to the algorithm, the abbreviation CBSLP, which stands for Cluster-Based Sparse Link Prediction, has been used hereafter. The data are first mapped into a graph after pre-processing, and the community detection algorithms mentioned in the previous section are then applied to them (line 9 of Figure 2). Prediction is made within each community; thereafter a matrix is defined for the inter-community step, in the relevant entries of which, all the edges between pair of communities are located. All the edges are traversed for finding inter-community edges, and each edge is inserted in the relevant entry of the matrix. Thus, graphs of inter-

[1] http://konect.uni-koblenz.de/networks/arenas-email

[2] http://snap.stanford.edu/data/ca-HepTh.html
[3] http://snap.stanford.edu/data/ca-GrQc.html
[4] http://vlado.fmf.uni-lj.si/pub/networks/data/dic/eat/Eat.htm
[5] http://www.hprd.org/

community edges are finally obtained. Next, each of the communities is subject to link prediction, each of the four basic neighborhood algorithms is examined (Table 1), and new links are predicted. Of course, probable repetitive edges resulting from the prediction in both steps are eliminated (Figure 2).

3-2-1- Intra-Cluster link Prediction

Using community detection, we divide the whole graph into several separated subgraphs that can be investigated independently for link prediction with more confidence of the closely connected links for better prediction results. Performance of CBSLP is as well as a divide and conquer method. First of all, seeking for communities and after that searching for the relation between those communities is performed. As seen in Figure 1(a), the obtained communities are represented as C_i. C_1 and C_2 are two of these clusters. Edges and vertices located in a single community are separated, and prediction is made within each of the communities, as clear from Figure 1(b). For edges indicted by dashed lines, link prediction is very likely made with the basic methods.

3-2-2- Inter-Cluster link Prediction

After dividing the main graph into communities and predicting the intra links in each community, it is necessary to investigate the probable links between each pair of communities. Because there are certainly several edges between communities that have not been considered in the calculations.

Here, we generate a graph between every two separate communities for the interconnected edges, and predicts links within each connected pair of the communities. The number of communities depends on the community detection algorithms. Some algorithms, like Best partition, automatically determine the appropriate number of communities, while some other clustering algorithms need a predefined number to break down the network into that number of communities. We utilize the elbow method to automatically determine the number of communities.

In order to perform the inter-community link prediction, first, we collect the common links between every two communities. Then we consider and add the links between the nodes located in each community, that participate in inter-community relations for the increment of the accuracy of the computations. For example, in Figure 1(d), we form an inter-community network including the {(a,b), (b,c), (c,d), (e,i), (i,h)} U {(a,h), (d,e)} edges.

Thus, the inter-community edges are taken into account, the total capacity of the network is used for prediction, and extra calculation is avoided at the same time as well. Traversing all the common edges between communities for finding inter-community relations that participate in the intersection communities' results in isolating new communities between pair of connected communities. Figure 1(c) shows the approach for two different communities. Implementation of the proposed method using inter-communities link prediction is also shown in Figure 2.

3-3- Evaluation

Three factors can be used for measuring the success of link prediction in large sparse networks: precision, AUC (Area Under Curve), and runtime. To calculate precision, 10-fold cross validation is performed. For each fold, 10% of the existing links are removed randomly to predict by the algorithm again. This is done ten times, and each time, a different 10% of the links are selected to be removed. This ensures that each link is withheld exactly once, so all links are present in the training data and the test data an equal number of times. Another evaluation metric for link prediction in unsupervised methods is AUC, also. It can be interpreted as the probability that a randomly chosen missing link is given a higher similarity score than a randomly chosen pair of unconnected links. If among n independent comparisons, there are n′ times the missing link having a higher score and n″ times they have the same score, the AUC value is calculated as the following[13]:

$$\text{AUC} = \frac{n' + 0.5n''}{n} \qquad (1)$$

The link prediction detailed above is taken, with an accurate chronometer measuring the time from the beginning to the end of the implementation, and average time, *i.e.*, mean runtime in each of the ten iterations, is calculated. This measure can be used for the assessment of the algorithm speed.

4- Results and Discussion

In this section, we will investigate the results of using the proposed method from different viewpoints including: decreasing the number of checked edges, comparing the best performance link prediction functions, and runtime comparison of CBSLP with basic methods.

4-1- Number of Edges under Examination

An interesting difference between the proposed method and the basic algorithms such as AA, PA, JC, and CN, lies in the numbers of edges and nodes under examination. This causes computations to be carried out in shorter times, regardless of the processing hardware that has been utilized, leading to good results over sparse networks. A summary of the comparison is provided in Table 3. It is

worth paying attention in CBSLP that we attempted to remove or ignore the lowest importance links. This table demonstrates the number of initial zero entries in the similarity matrix that should be calculated by basic methods and the proposed method. For example, for the

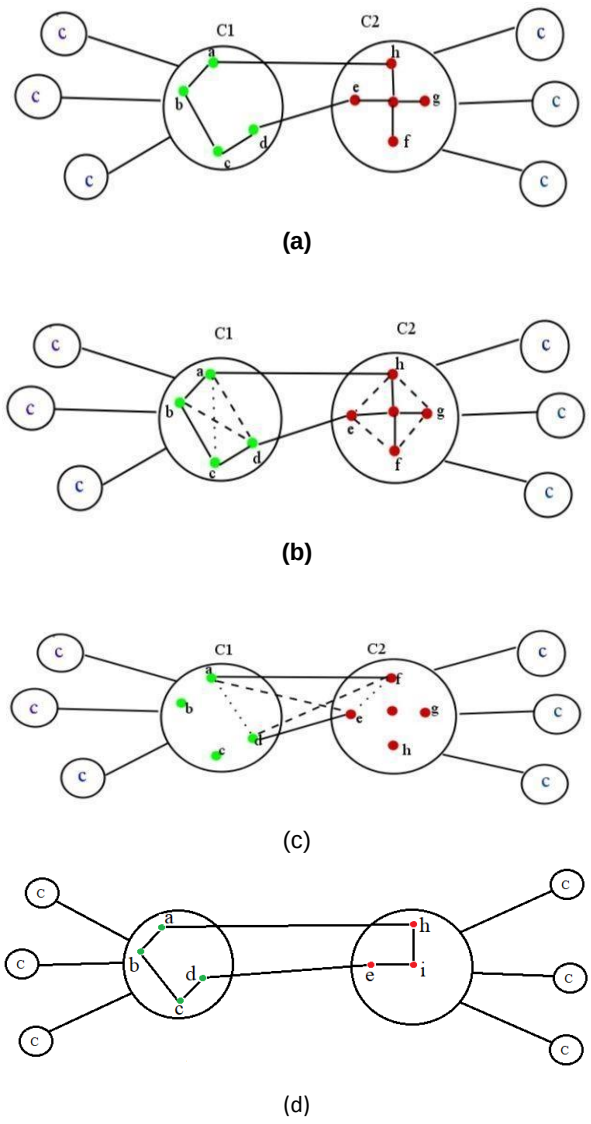

Fig 1: Detection of clusters and prediction of links within and between communities (a) Sample for obtained communities, such as C_1 and C_2 after running community detection algorithms (b) Performing intra-community link prediction (c) Considering inter-community links, (d) Inter-community graph formation based on the related links, for complementary link prediction.

Email dataset, the former methods have about 641844 calculations, while the latter method makes this value lower approximately one-fourth about 163350 in the worst case. Indeed, there are some inter community edged that should be taken into account, but they are few and can be ignored.

CBSLP(Cluster-Based Sparse Link Prediction)

[1]**Input: array of dataset as data$_a$**
[2]**Output: list of prediction item conducting CBSLP algorithm**

First phase starting (intra cluster)
[3] $array_t \leftarrow data_a$
[4] For item in $array_t$:
[5] **make $array_t$ symmetric and remove self loop and put in $array_{cp}$**
[6]**Return $array_{cp}$**
[7]**For** item **in $array_{cp}$**:
[8] **C**reate graph G_t from $array_{cp}$
[9]**R**un community detection algorithm ϵ
{best partition, link community, infomap,Girvan and newman} on G_t
[10]**For** $node_1$ **in** G_t:
[11] $node_1 \in \{C_1, C_2, C_3, \ldots, C_n\}$ # C_i Means community
[12]**Return** Clusters
[13]**Create** subgraph G_t
[14]**For** C_i **in** range (C_1, C_2, C_3,…., C_n)
[15] **IF** (Len C_i>2):
[16] $G_s \leftarrow G_i$
[17] Tupple(x,y,**score**)$_{fg}$ ←basic approach ∈ {AA,PA,CN,RA,Jc}(G_s)
[18] $G_s \leftarrow \emptyset$
[19]Return $Tupple_{fg}$ #can use output of first phase
First phase end
Second phase starting(inter cluster)
[20]array[][]$_a \leftarrow \emptyset$ #Make two-dimensional array
[21]**For** $edge_{u,v}$ **in** G_t:
[22] **If** (C_u !=C_v):
[23] $array[u][v]_a \leftarrow (u, v)$
[24]**For** edge$_{u,v}$ **in** G_t:
#Find edges between each two clusters and then make subcluster
[25] $array[u][v]_a \leftarrow neighbours(u, v)$
#Find all neighbor to make prediction more accurate
[26]$G_s \leftarrow \emptyset$
[27]**For** C_i **in** $array_a$: #**edges were put in** *array*[][]$_a$ **from cluster C**
[28] **If** (**len** c_i> **2**):
[29] $G_s \leftarrow C_i$
[30] $item$ ←basic approach ∈ {AA,PA,CN,RA,Jc}(G_s)
[31] for(item in range(0,len(Tupple$_{fg}$):
[32] #merge of result
[33] **If** (item **not in** Tupple$_{fg}$):
[34] **append** item to Tupple$_{fg}$
[35] **else:**
[36] Compare their score remove the less one from Tupple$_{fg}$
[37] Sort Tupple$_{fg}$ again by score
[38] $G_s \leftarrow \emptyset$
[39] Return Tupple$_{fg}$ #including cumulative results
Second phase end (inter cluster)

Fig 2: Pseudo-code of the proposed method for link prediction in sparse networks

4-2- Comparison with Similar Competing Methods

For evaluation of CBSLP, its performance is compared with primary methods. In Table 4, a summary of the results obtained by the proposed method is provided, along with

comparing to those of different community detection methods mentioned above. It should be mentioned that the column containing the cumulative results involves the overall results obtained from both intra-community and inter- community phases. The proposed method has no claim on dense graphs such as HEP or Rel, because it may not be appropriate for such a graph structure in a particular application, and may also be led to the elimination of valuable predictions from the graph. In Table 4 BP, LC, info are the abbreviations of best partition, link community and Infomap respectively where all of them are community detection methods that were mentioned before. The bold numbers show the best result in each column of Table4. As a result, CBSLP achieved better results in sparse networks such as Email, Word, Wiki-Vote, PPI. It is worth to mention that (–) in each column means that the pertaining method could not terminate the calculations within a reasonable time (72 hours). Another evaluation metric is AUC. Results in table 5 also confirm the precision metric findings.

Table 3: Comparison between the number of calculations in CBLSP and basic methods. Dividing the investigating graph into clusters reduces the total numbers of calculations considerably. Columns three to seven are the top most populated clusters for each dataset in order to take into account the upper bound calculation numbers in CBLSP method in comparison.

Network	Total number of clusters	Node-cluster 1	Node-cluster 2	Node-cluster 3	Node-cluster 4	Upper bound of the sum of entries examined in the CBSLP	Size of the matrix examined in the basic methods
Email	12	165	165	134	126	12*(165*165)/2=163350	641844
HEP	209	756	620	418	410	209*(756*756)/2=59725512	48778564
REL	210	308	267	258	251	210*(308*308)/2=9960720	13739282
Word	9	4211	3354	3216	3096	9*(4211*4211)/2=79796344	269560980
Wiki-vote	6	1704	1610	1593	1384	6*(1704*1704)/2=1451808	25311612
PPI	48	1497	1152	925	777	48*(1497*1497)/2=53784216	450000000

Table 4: Precision results obtained from the basic methods, and the proposed method using different clustering algorithms. Cells with dash sign are the calculations has not committed in a rational time, 72 hours of computation with our hardware. Precision of the CBSLP for the inter-community relations is not remarkable compared to intra-community results.

Network	Basic methods	CBSLP BP method	CBSLP LC method	CBSLP INFO method	CBSLP Girvan-Newman method	CBSLP with cumulative results	Precision of the proposed method for inter-communities
Email	0.141AA	**0.146AA**	0.139AA	0.141AA	0.141AA	0.141AA	0.033(AA)
HEP	**0.37CN**	0.35CN	0.37CN	0.35CN	0.35CN	0.34CN	0.033(CN)
REL	**0.5RA**	0.49RA	0.48RA	0.42RA	0.42RA	0.49RA	0.04(RA)
Word	-	-	**0.11AA**	-	-	0.1RA	0.021(AA)
Wiki-vote	0.09RA	**0.11AA**	0.09RA	0.11AA	-	0.11AA	0.036(AA)
PPI	0.06AA	**0.062AA**	0.057	0.043	-	0.062AA	0.014(AA)

Table 5: AUC results obtained from the basic methods, and the proposed method using different clustering algorithms. Cells with dash sign are the calculations has not committed in a rational time, 72 hours of computation with our hardware.

Network	Basic methods	CBSLP BP method	CBSLP LC method	CBSLP INFO method	CBSLP Girvan-Newman method	CBSLP with cumulative results	AUC of the proposed method for inter-communities
Email	0.87AA	**0.89AA**	0.83AA	0.821AA	0.823AA	0.821AA	0.95(AA)
HEP	**0.597CN**	0.591CN	0.592CN	0.63CN	0.621CN	0.61CN	0.80(CN)
REL	**0.63RA**	0.624RA	0.611RA	0.655RA	0.63RA	0.62RA	0.79(RA)
Word	-	-	**0.89RA**	-	-	-	-
Wiki-vote	0.88RA	**0.91RA**	0.90RA	0.90RA	-	0.91RA	0.91(RA)
PPI	0.91AA	**0.92AA**	0.91AA	0.91AA	-	0.89AA	0.89(AA)

4-3- Runtime Analysis and Comparison

In the above four sections, it was discussed that the basic methods have not been successful in link prediction over the Word network, and could not solve it within a reasonable time (72 hours). It is also noticeable that the basic methods CN could probably not be implemented over several similar large networks within a logical time, while the CBSLP in this research successfully computed a sample within a proper time. Therefore, this method has improved time as well, as shown in Table 3. The specification of the system used in this research is shown in Table 6.

Table 6: Specification of the system used in the research

Processor	**Intel Core i5 3250M**
Main Memory	8 Gigabytes
Hard disk memory	500 Gigabytes
Operating system	Linux Ubuntu

First, it is necessary to know what percentage of the links would be predicted correctly if and only if the links and nodes between communities were investigated and evaluated with the methods introduced in the inter-community step. The answer could be found in the Precision of inter-community results column in Tables 4 and 5. The runtimes of methods was calculated for each dataset, and the results can be observed in Table 7.

Table 7: Runtimes of the proposed method, and the basic methods

Network	**Runtime of the clustering method (Mean in one iteration)**	**Runtime of the basic (Mean in one iteration)**
Email	1-2 minutes	5-10 minutes
HEP	4-5 minutes	25-30 minutes
REL	3-4 minutes	15-20 minutes
Word	15 minutes	6 hours
Wiki-vote	2 hours	-
PPI	30 minutes	5 hours

Clearly, about 0.031 of the links predicted to occur between communities over a network like Email, which means that about 20% of the links occur between communities rather than within them. Unfortunately, however, not much change occurs when the inter- and intra- community links are predicted and evaluated at the same time, as clear from the proposed method with cumulative results' column in Table 4. This is because two lists with different scores are merged, which causes the scores to drift on the list with higher precision, and the results not to change and the final result to worsen even. If the results are cumulated correctly, the method will definitely succeed in denser graphs as well.

5- Conclusion and Future Works

The proposed method, CBSLP, involves a framework for large sparse graphs, since it prevents extra computation, improves runtime, and saves memory. Besides, it can be regarded as a new link prediction method for sparse networks due to its strategy details. However, CBSLP is an initial version of the framework, which should evolve greatly. In the proposed method, clustering was used as a tool not only for improvement of the prediction results but also for elimination of extra calculation. In addition, there is a lot that needs to be done for its evolution. For the precision of the proposed method to increase, attempts can be made to make link prediction also using path-based methods. An appropriate method among path-based algorithms that is recommended in sparse graphs is the SRW[1] method, which improves the results probably. One can attempt to experiment newer and better community detection algorithms for higher precision, such as [27] or [28]. Moreover, a mechanism has been sought to utilize weighted graph version of the network for improvement of the results using inter-cluster relations and their outcomes. It is possible even applying rank aggregation to link prediction lists with different scores for achieving better results. Methods such as that in [15] or [29] can be used to employ cluster information in order to improve the proposed method in terms of precision.

References

[1] D. Caiyan, L. Chen, and B. Li, "Link prediction in complex network based on modularity," Soft Comput., 2016, doi: 10.1007/s00500-016-2030-4.

[2] H. Yuan, Y. Ma, F. Zhang, M. Liu, and W. Shen, "A distributed link prediction algorithm based on clustering in dynamic social networks," in 2015 IEEE International Conference on Systems, Man, and Cybernetics, 2015, pp. 1341–1345.

[3] P. Symeonidis and N. Mantas, "Spectral clustering for link prediction in social networks with positive and negative links," Soc. Netw. Anal. Min., vol. 3, no. 4, pp. 1433–1447, 2013.

[4] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," J. Am. Soc. Inf. Sci. Technol., vol. 58, no. 7, pp. 1019–1031, 2007.

[5] G. SALTON and M. J. MCGILL, "Introduction to Modern Information Retrieval (pp. paginas 400)." 1986.

[6] M. E. J. Newman, "Clustering and preferential attachment in growing networks," Phys. Rev. E, vol. 64, no. 2, p. 25102, 2001.

[1] Superposed Random Walk

[7] P. Wang, B. W. Xu, Y. R. Wu, and X. Y. Zhou, "Link prediction in social networks: the state-of-the-art," Sci. China Inf. Sci., vol. 58, no. 1, pp. 1–38, 2014, doi: 10.1007/s11432-014-5237-y.

[8] M. K. Khouzani and S. Sulaimany, "Identification of the effects of the existing network properties on the performance of current community detection methods," J. King Saud Univ. - Comput. Inf. Sci., Apr. 2020, doi: 10.1016/j.jksuci.2020.04.007.

[9] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral, "A network-based method for target selection in metabolic networks," Bioinformatics, vol. 23, no. 13, pp. 1616–1622, 2007.

[10] N. Benchettara, R. Kanawati, and C. Rouveirol, "A supervised machine learning link prediction approach for academic collaboration recommendation," in Proceedings of the fourth ACM conference on Recommender systems, 2010, pp. 253–256.

[11] G. W. Flake, S. Lawrence, C. L. Giles, and F. M. Coetzee, "Self-organization and identification of web communities," Computer (Long. Beach. Calif)., vol. 35, no. 3, pp. 66–70, 2002.

[12] A. Clauset, C. Moore, and M. E. J. Newman, "Hierarchical structure and the prediction of missing links in networks," Nature, vol. 453, no. 7191, pp. 98–101, 2008.

[13] Z. Liu, Q.-M. Zhang, L. Lü, and T. Zhou, "Link prediction in complex networks," EPL (Europhysics Lett., vol. 96, no. 4, p. 48007, 2011.

[14] E. M. Airoldi, D. M. Blei, S. E. Fienberg, E. P. Xing, and T. Jaakkola, "Mixed membership stochastic block models for relational data with application to protein-protein interactions," in Proceedings of the international biometrics society annual meeting, 2006, vol. 15.

[15] J. H. S Soundarajan, "Using community information to improve the precision of link prediction methods," WWW (Companion Vol., vol. 2012, pp. 607–608, 2012.

[16] S. Yokoi, H. Kajino, and H. Kashima, "Link prediction in sparse networks by incidence matrix factorization," J. Inf. Process., vol. 25, pp. 477–485, 2017.

[17] K. Shang, T. Li, M. Small, D. Burton, and Y. Wang, "Link prediction for tree-like networks," Chaos An Interdiscip. J. Nonlinear Sci., vol. 29, no. 6, p. 61103, 2019.

[18] J. Zhang, J. Chen, S. Zhi, Y. Chang, S. Y. Philip, and J. Han, "Link prediction across aligned networks with sparse and low rank matrix estimation," in 2017 IEEE 33rd International Conference on Data Engineering (ICDE), 2017, pp. 971–982.

[19] C. H. Nguyen and H. Mamitsuka, "Latent feature kernels for link prediction on sparse graphs," IEEE Trans. neural networks Learn. Syst., vol. 23, no. 11, pp. 1793–1804, 2012.

[20] X. Feng, J. C. Zhao, and K. Xu, "Link prediction in complex networks: a clustering perspective," Eur. Phys. J. B, vol. 85, no. 1, p. 3, 2012.

[21] H. Liu, Z. Hu, H. Haddadi, and H. Tian, "Hidden link prediction based on node centrality and weak ties," EPL (Europhysics Lett., vol. 101, no. 1, p. 18004, Jan. 2013, doi: 10.1209/0295-5075/101/18004.

[22] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," J. Stat. Mech. theory Exp., vol. 2008, no. 10, p. P10008, 2008.

[23] Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multiscale complexity in networks," Nature, vol. 466, no. 7307, pp. 761–764, 2010.

[24] A. Lancichinetti and S. Fortunato, "Community detection algorithms: a comparative analysis," Phys. Rev. E, vol. 80, no. 5, p. 56117, 2009.

[25] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," Proc. Natl. Acad. Sci., vol. 105, no. 4, pp. 1118–1123, 2008.

[26] K. Esders, "Link Prediction in Large-scale Complex Networks," Bachelor's Thesis Karlsruhe Inst. Technol., no. February, 2015, [Online]. Available: https://hackernoon.com/link-prediction-in-large-scale-networks-f836fcb05c88.

[27] M. Hajiabadi, H. Zare, and H. Bobarshad, "IEDC: An integrated approach for overlapping and non-overlapping community detection," Knowledge-Based Syst., vol. 123, pp. 188–199, 2017.

[28] H. Zare, M. Hajiabadi, and M. Jalili, "Detection of community structures in networks with nodal features based on generative probabilistic approach," IEEE Trans. Knowl. Data Eng., 2019.

[29] H. Zare, M. A. N. Pour, and P. Moradi, "Enhanced recommender system using predictive network approach," Phys. A Stat. Mech. its Appl., vol. 520, pp. 322–337, 2019.

Mohammad Pooya Salavati received his MSc. degree in Computer Engineering from Urmia University, West Azarbaijan, Iran in 2020. His research interests include Link prediction, Community detection and Big data analysis.

Jamshid Bagherzadeh Mohasefi received bachelor of computer engineering from Sharif University of Technology in Iran at 1996 and master of computer engineering from Tarbiat Modares University in Iran at 1999. He got his PhD in computer engineering from Indian Institute of Technology Delhi (IITD) in India at 2006. He joined Urmia University as a faculty member in 2006. He has worked since there in two major fields including information security and artificial intelligence. He has published more than 80 referred journal and conference papers. He is currently associate professor at Urmia University, Iran. His current research interests include machine learning, data/text mining, and network security.

Sadegh Sulaimany is assistant professor at the department of computer engineering in University of Kurdistan, Iran. He received his PhD in Bioinformatics from Tehran University in 2017. His Master degree is in Computer Science from Amirkabir University of Technology, Tehran, and his Bachelor degree is in Computer Engineering from Isfahan University of Technology, Isfahan, Iran. His research interest lies in but not limited to the biological and social network analysis, especially link prediction algorithms for different application areas.