Chapter 6

Community-based Link Prediction

In this chapter, an application is designed by leveraging the community information of networks. The influence of locality of nodes in different communities is explored for better utilizing the community information.

6.1 Introduction

Community detection has significant role in understanding functional properties of complex networks. Nowadays, community information has been utilized immensely in different applications (see section 2.5 for the review). The problem of identifying missing links or future links or spurious links in the network is referred as *link prediction* [205]. Community-guided link prediction schemes are becoming popular. Various properties such as hierarchical organization of communities [43], community similarity feature [44], similarity of communities [134, 174] and herd phenomenon in different communities [105] are used to develop efficient link prediction algorithms.

In this chapter, other aspects of communities such as locality of nodes in communities and importance of the neighbors are explored for link prediction. The information about communities is utilized to influence the likelihood scores of missing links based on the localities of nodes associated with existing links or connections. Various edge centrality measures are studied for computing importance of neighbors of a node. A Community-based Link Prediction (CLP) algorithm is developed. Three variants of CLP have been proposed based on the edge centrality measure used in the algorithm.

6.2 Proposed Approach

Simple undirected and unweighted graph $G(V,E)$ is considered, where V is the set of nodes and E is the set of links. Let $n = |V|$and $m = |E|$ are the number of nodes and links respectively in the graph G. With n number of nodes, $n \times (n-1)/2$ number of links are possible, if multiple links and self-loops are not allowed. The set of all possible links in the graph is denoted by U. Thus, $U - E$ will be the set of all non-existing or non-observed links [1]. The problem is to find the missing links or the links that are more likely appear in near future from the set of non-existing links.

In general, link prediction problem is expressed in terms of likelihood score of links. Let $S(e_{ij}|G)$ denotes the likelihood score for presence of a link between two nodes i and j subject to the graph G, the link prediction problem is deduced as to determine the likelihood score $S(e_{ij}|G)$ for all $e_{ij} \in U - E$ non-existing links. Often, likelihood scores of all links i.e. U are computed to verify the accuracy of the predicted links. Mostly, link prediction problems consider only the connectivity pattern of graph or node attributes to determine the likelihood scores $S(e_{ij}|G)$ of non-existing links. The notion of incorporating community structure in addition to the connectivity pattern of the graph is

[1] The terms, non-existing link and non-observed link are used interchangeably throughout this chapter.

proposed. Nodes of different communities are identified with their assigned community labels. Same labels are assigned to all nodes that belong to same community. Let C is a community structure, where a community label is assigned to all nodes in G. Likelihood score of having a link in-between nodes i and j subject to given community structure C is denoted by $S(e_{ij}|G,C)$. Thus, link prediction with additional element (i.e. the community structure) is defined as follows. For a given graph G and community structure C, the new link prediction problem is to determine the likelihood score $S(e_{ij}|G,C)$ for all $e_{ij} \in U - E$ i.e. all non-existing links.

The concepts of edge centrality and community structure are considered to design the proposed algorithm. The important or significant links are determined by considering the edge centrality measures, while incorporation of community structure accounts the influence of communities in link prediction. There are mainly three edge centrality measures, which are developed in last decades. Girvan and Newman developed edge betweenness centrality [61]. It considers shortest paths to compute edge centrality. Another measure called k-path edge centrality, where paths of specific length are considered [5, 40]. A spanning tree based edge centrality measure called spanning edge centrality is shaped by several researcher in recent years [80, 128, 154, 191]. Formal definitions of these edge centrality measures are given below.

Definition 6.1 (Edge Betweenness Centrality)**.** Given a graph $G(V,E)$, edge betweenness centrality of link e_{ij} in between nodes i and node j is defined as follows:

$$EC_B(e_{ij}) = \sum_{s \neq t \in V} \frac{\sigma_{st}(e_{ij})}{\sigma_{st}} \tag{6.1}$$

where, $s \in V$ and $t \in V$ are any two pair of nodes such that $s \neq t$, σ_{st} is the number of shortest path between nodes s and t, and $\sigma_{st}(e_{ij})$ is the number of shortest paths between nodes s and t that includes the link e_{ij}.

Definition 6.2 (K-path Edge Centrality)**.** Given a graph $G(V,E)$, k-path edge centrality of link e_{ij} in between nodes i and node j is defined as follows:

$$EC_P(e_{ij}) = \sum_{s \in V} \frac{\sigma_s^k(e_{ij})}{\sigma_s^k} \tag{6.2}$$

where, s are all the possible source nodes for the paths of at least length k (i.e. k-path), σ_s^k is the number of k-paths starting from node s, and $\sigma_s^k(e_{ij})$ is the number of k-paths that include the link e_{ij}.

Definition 6.3 (Spanning Edge Centrality)**.** Given a graph $G(V,E)$, spanning edge centrality of link e_{ij} in between nodes i and node j is defined as follows:

$$EC_S(e_{ij}) = \sum_{s \in V} \frac{T_s(e_{ij})}{T_s} \tag{6.3}$$

where, s are all the possible source nodes of spanning tree, T_s is the number of spanning trees originating from node s, and $\sigma_s^k(e_{ij})$ is the number of spanning trees that include the link e_{ij}.

Besides the edge centrality measure, the community structure of the network is also considered to define community influence on likelihood scores of future links.

Definition 6.4 (Community Structure)**.** Given a graph $G(V,E)$ and a set of k community labels L, a simple disjoint community structure is as a community label assignment to all the nodes such that the groups of nodes having same community labels are connected densely, while the groups of the nodes having different community labels are connected sparsely. Suppose, C is a community structure, every $C_i \in C$ is the community label of node i such that $C_i \in L$.

Weights are assigned to all links E based on their locality in the community structure. If the nodes that are associated with a link have same community label (i.e. belong to

same community) positive weight is assigned, otherwise negative weight is assigned. The assignment of positive and negative weights to links accounts positive and negative influence of existing links respectively on future links during likelihood score computation. Community-based edge weight (CEW) of any link $e_{ij} \in E$ is computed as follows:

$$CEW(e_{ij}) = \begin{cases} +\frac{\delta_i}{n}, & \text{if } C_i = C_j \\ -\frac{\delta_i}{n}, & \text{otherwise} \end{cases} \tag{6.4}$$

where, δ_i is the size of the community that includes node i, n is the number of nodes in G, C_i is the community label of node i and C_j is the community label of node j. Incorporating CEW, the importance of node with respect to its neighbor using Edge Centrality (EC)is defined as follows. Let $EC(e_{ij})$ be the edge centrality of the link e_{ij}. Thus, overall importance of node j to node i is computed as follows:

$$I(e_{ij}) = CEW(e_{ij}) \times EC(e_{ij}) \tag{6.5}$$

To determine likelihood score of link $e_{ij} \notin E$ between two node i and j, the importance of common neighbors to both nodes is considered. Let Γi and Γj are the set of neighbors of node i and j respectively. The importance of any common node $x \in \Gamma i \cap \Gamma j$ to node i is computed as follows:

$$I(e_{ix}) = CEW(e_{ix}) \times EC(e_{ix}) \tag{6.6}$$

Similarly, importance of any common node $x \in \Gamma i \cap \Gamma j$ node to node j is computed as

$$I(e_{jx}) = CEW(e_{jx}) \times EC(e_{jx}) \tag{6.7}$$

Algorithm 6.1: Community-based Link Prediction (CLP)

Input: adjacency or weight matrix W, community structure C
Output: Scores of non-existing links S
$EC \leftarrow$ Compute edge centrality for all $e_{ij} \in E$
$CEW \leftarrow$ Compute community-based edge weight for all $e_{ij} \in E$
for all $e_{ij} \in U - E$ **do**
    $\Gamma i \leftarrow$ neighbors of node i
    $\Gamma j \leftarrow$ neighbors of node j
    $S(e_{ij}) \leftarrow \sum_{\forall x \in \Gamma i \cap \Gamma j} CEW(e_{ix}) \times EC(e_{ix}) + CEW(e_{jx}) \times EC(e_{jx})$
end for
return S

Total likelihood score of link between two node i and j is computed as follows:

$$S(e_{ij}) = \sum_{\forall x \in \Gamma i \cap \Gamma j} I(e_{ix}) + I(e_{jx}) \tag{6.8}$$

$$S(e_{ij}) = \sum_{\forall x \in \Gamma i \cap \Gamma j} CEW(e_{ix}) \times EC(e_{ix}) + CEW(e_{jx}) \times EC(e_{jx}) \tag{6.9}$$

Incorporating the concepts explained above, the proposed Community-based Link Prediction (CLP) algorithm is designed. A pseudo code of CLP algorithm is presented in Algorithm 6.1. The CLP algorithm has three simple steps: 1) computation of edge centrality (EC) of all existing links i.e. all $e_{ij} \in E$, 2) computation of community-based edge weight (CEW) of all existing links and 3) computation of likelihood score (S) of all non-existing links i.e. $e_{ij} \in U - E$. Three variants of CLP algorithm is proposed based on the edge centrality measures used in the algorithm. These three variants are referred as CLP-EB, CLP-EP and CLP-ES respectively for the CLP algorithm with edge betweenness centrality, k-path edge centrality and spanning edge centrality.

The time complexity of the CLP algorithm is dependent on three main steps of the algorithm as mentioned above. The computation costs of both CEW and S are same for

all three variants CLP-EB, CLP-EP and CLP-ES. However, computation costs of EC for CLP-EB, CLP-EP and CLP-ES are different as they use different edge centrality measures. Computation cost of obtaining CEW is $O(n^2)$. Computation cost of obtaining likelihood scores S of all links is $O(n^2)$. Brande's algorithm is considered for computing edge betweenness centrality [19]. Brande's algorithm costs $O(n^2)$. Thus, cost of CLP-EB algorithm is $O(n^2)$. Edge Random Walk k-Path Centrality (ERW-Kpath) algorithm is considered for computing k-path edge centrality [40]. The ERW-Kpath algorithm costs $O(km)$, where k is the maximum length of paths. In worst case scenario, number of links become $m = \frac{n(n-1)}{2}$. Thus, cost of CLP-EP becomes $O(n^2)$.

Hayashi et al. [80] approximation algorithm is considered for measuring spanning edge centrality. They showed the approximation cost $\left\lceil \frac{(\log((2\times m)/\delta))}{(2\times\varepsilon^2)} \right\rceil \times \sum_{u\in V-r} \pi_G(u)\kappa_G(u,r)$, where δ is the failure probability, ε is the expected error, $\pi_G(u)$ denotes the probability of staying at node $u \in V$ in the stationary distribution of a random walk on G, $\kappa_G(u,r)$ denotes the commute time between two nodes, and r is the root node chosen for the spanning tree. Their method generates $q = \left\lceil \frac{\log((2\times m)/\delta}{2\times\varepsilon^2} \right\rceil$ numbers of uniform random spanning trees. They suggested to consider $\delta = 1/n$ and $\varepsilon = 0.05$. For all possibles edges with n nodes the value of $m = \frac{n(n-1)}{2}$. Putting these values $q = \left\lceil \frac{\log(n-1)}{0.005} \right\rceil = 200\log(n-1)$ or simply $O(\log n)$. To compute edge spanning centrality of an edge, all of the q spanning trees are checked if the edge is included in the spanning tree or not. Thus, worst case approximation cost of spanning tree centrality with Hayashi et al. method becomes $O(n^3 \log n)$. Therefore, cost of CLP-ES algorithm is $O(n^3 \log n)$.

6.3 Evaluation Strategy

The accuracy of the proposed algorithm is evaluated as follows. Those links are already present in the graph i.e. E, those are termed as observed links. The observed links E is

randomly divided into two subsets: the training set E^T and the probe set E^P. The set E^T is considered as known information, while the set E^P is kept only for validation purpose; $E^T \cup E^P = E$ and $E^T \cap E^P = \phi$. K-fold cross-validation is considered, where observed links E is randomly divided into K subsets. One of the K subsets are considered as probe set, while remaining $K-1$ subsets constitute training set. This implies K different pairs of probe sets and training sets. The algorithm is executed for each of the K pairs and analyze results in terms various performance metrics.

6.3.1 Performance Metrics

A link prediction algorithm outputs list of non-observed links (i.e. $U - E^T$) along with their scores determined by the algorithm. Various performance metrics are designed to evaluate the scores returned by any link prediction algorithm. Three standard metrics are used to quantify the accuracy of link prediction algorithms: area under the receiver operating characteristic curve (AUC) [75] and Precision [58, 82] and ranking score (RS) [28].

AUC: The scores of non-observed links are used to compute AUC value. It is measured in probabilistic terminology. The AUC value gives the probability that a randomly selected missing link (i.e. a link in E^P) has a higher score than a randomly selected nonexistent link (i.e. a link in $U - E$). The non-observed links resulted by a link prediction algorithm is used to compute AUC value as follows. Each time a missing link and a nonexistent link are chosen from $U - E^T$ to compare their scores. y independent comparisons are performed, and if there are y' times the missing links having higher scores and y'' times both have same scores, the AUC value is

$$AUC = \frac{y' + 0.5 \times y''}{y}. \tag{6.10}$$

If all the scores are generated from an independent and identical distribution, the AUC value should be about 0.5. Therefore, the degree to which the value exceeds 0.5 indicates how much better the algorithm performs than pure chance.

Precision: In information retrieval context, precision is defined as the ratio of relevant items retrieved to the number of items retrieved. In the case of link prediction, precision is defined as follows. All non-observed links are ranked based on their scores determined by the algorithm. Top-β links from the non-observed are considered. If β_r links are right among the top-β links (i.e. β_r links of top-β links are in the probe set E^P), then the precision is defined as $\frac{\beta_r}{\beta}$. Higher precision indicates higher prediction accuracy.

RS: Ranking score (RS) is defined based on the ranks obtained by the links in prob set E^P. Let $H = U - E^T$ be the set of non-observed links. Let e_i be a link in the probe set E^P that is ranked r_i among all non-observed links after sorting in descending order of similarity scores. The ranking score (RS_i) of link e_i is computed as:

$$RS_i = r_i/|H| \tag{6.11}$$

The ranking score (RS) of the link prediction result is defined as:

$$RS = \frac{1}{|E^P| \sum\limits_{e_i \in E^P} RS_i} \tag{6.12}$$

$$= \frac{1}{|E^P| \sum\limits_{e_i \in E^P} \frac{r_i}{|H|}} \tag{6.13}$$

Table 6.1: Comparison of CLP-EB, CLP-EP, and CLP-ES in terms of accuracy quantified by AUC

Algorithms	Measures	Networks			
		Dolphin	Football	Karate	Strike
CLP-EB	Best	0.8850	0.8100	0.8450	**0.9400**
	Mean	0.7760	0.7610	0.6925	**0.8095**
	SDev	0.0716	0.0402	0.0868	0.1528
CLP-EP	Best	0.8900	0.8300	0.8300	**0.9400**
	Mean	0.8135	0.7785	**0.7540**	0.7525
	SDev	0.0454	0.0506	0.0581	0.1187
CLP-ES	Best	**0.9050**	**0.9000**	**0.8800**	0.9300
	Mean	**0.8140**	**0.7805**	0.7225	0.6975
	SDev	0.0639	0.0569	0.0787	0.1647

6.3.2 Experimental Setup

The proposed Community-based Link Prediction (CLP) does not have any specific parameters for all three variants: CLP-EB, CLP-ES and CLP-EP. However, the algorithm of spanning tree centrality and k-path centrality have some parameters, so for CLP-ES and CLP-EP certain parameters need to set. Spanning tree centrality algorithm [80] has three parameters: δ, ε and q. The values suggested by Hayashi et al. [80] are used: $\delta = \frac{1}{n}$, $\varepsilon = 0.05$ and the value of q is computed with the formula $\left\lceil \frac{(\log((2\times m)/\delta))}{(2\times\varepsilon^2)} \right\rceil$, where n is the number of nodes and m is the number links in the network. The algorithm of edge k-path centrality measure [40] also has three parameters: k, ρ and β. These parameter values are considered as follows: $k = 5$, $\rho = m - 1$ and $\beta = \frac{1}{m}$, where m is the number of links in the network. Edge betweenness centrality algorithm [19] does not have any parameter so CLP-EB is free from parameter setting.

Table 6.2: Comparison of CLP-EB, CLP-EP, and CLP-ES in terms of accuracy quantified by Precison

Algorithms	Measures	Networks			
		Dolphin	Football	Karate	Strike
CLP-EB	Best	0.0769	0.2097	0.2500	**0.2000**
	Mean	0.0293	0.1773	0.1059	0.0200
	SDev	0.0286	0.0206	0.0837	0.0632
CLP-EP	Best	**0.2069**	0.2101	0.2727	**0.2000**
	Mean	0.1002	**0.1791**	**0.1268**	**0.0343**
	SDev	0.0542	0.0147	0.0645	0.0735
CLP-ES	Best	0.1429	**0.2177**	**0.3000**	0.0000
	Mean	**0.1069**	0.1653	0.1071	0.0000
	SDev	0.0288	0.0367	0.0983	0.0000

6.4 Result Analysis

The accuracy of results predicted by CLP-EB, CLP-EP and CLP-ES is evaluated in terms of AUC and Precision. Ranking score (RS) is used for evaluating the quality of predicted results. 10-fold cross-validation is considered for measuring both accuracy and quality of predicted results. In 10-fold cross-validation, the set of existing links E of the graph is randomly partitioned into 10 subsets. Among these 10 subsets, one subset (say E^P) is considered as probe set for validation, while remaining 9 subsets (i.e. $E - E^P = E^T$) constitute training set that is by the link prediction algorithms. The cross-validation process is then repeated 10 times and both accuracy and quality measures are computed for each execution. Three values: best, mean and standard deviation are presented for AUC, Precision and RS. In addition to this, total time required by each of the algorithms to complete 10-fold cross-validation is also analyzed.

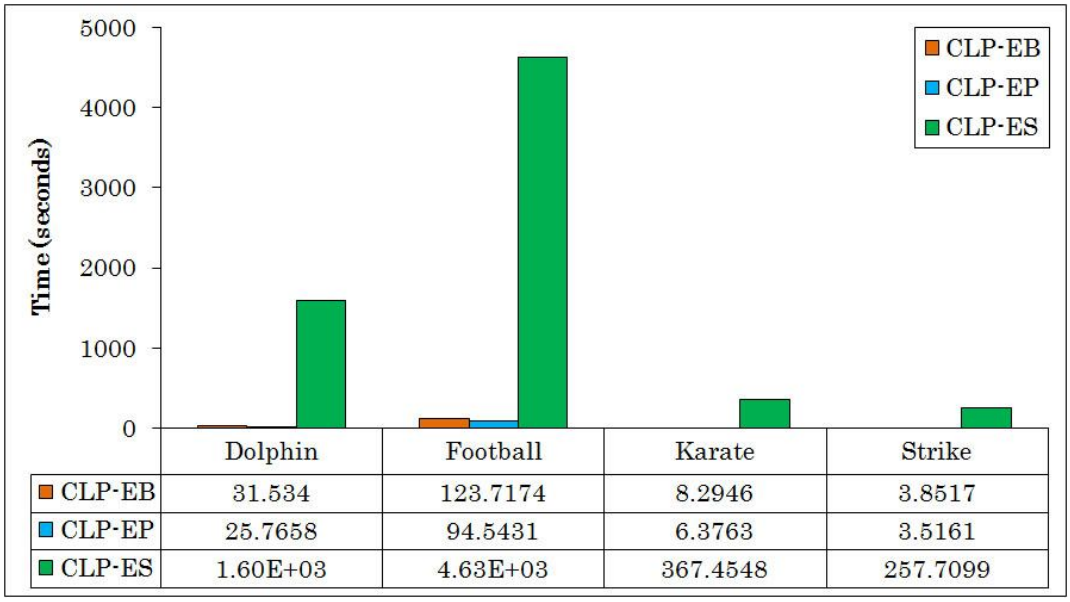
Table 6.1 presents best, mean and standard deviation of AUC values for 10-fold cross-validation on four networks: Dolphin, Football, Karate and Strike. Bold entries indicate

Table 6.3: Comparison of CLP-EB, CLP-EP, and CLP-ES in terms of Ranking Score (RS)

Algorithms	Measures	Networks			
		Dolphin	Football	Karate	Strike
CLP-EB	Best	**0.2914**	0.2751	**0.4608**	**0.7960**
	Mean	**0.2272**	**0.2386**	**0.3176**	0.2225
	SDev	0.0437	0.0157	0.0878	0.2217
CLP-EP	Best	0.2376	0.2428	0.3597	0.5058
	Mean	0.1827	0.2280	0.2523	0.2617
	SDev	0.0401	0.0122	0.0709	0.1506
CLP-ES	Best	0.2824	**0.2814**	0.3628	0.6402
	Mean	0.1919	0.2145	0.2647	**0.3090**
	SDev	0.0566	0.0334	0.0773	0.1864

Table 6.4: W-values obtained with Wilcoxon rank test for AUC, Precision and RS values on four networks

Networks	Method pair		W values obtained for		
	Method 1	Method 2	AUC	Precision	RS
Dolphin	CLP-EB	CLP-EP	10.5	0	10
	CLP-EB	CLP-ES	17	2	13
	CLP-EP	CLP-ES	27	22	23
Football	CLP-EB	CLP-EP	22	25.5	17
	CLP-EB	CLP-ES	15	18	13
	CLP-EP	CLP-ES	23	9.5	15
Karate	CLP-EB	CLP-EP	13	16	16
	CLP-EB	CLP-ES	15.5	26	11
	CLP-EP	CLP-ES	19	20.5	25
Strike	CLP-EB	CLP-EP	18	0	24
	CLP-EB	CLP-ES	14	0	15
	CLP-EP	CLP-ES	24	0	25

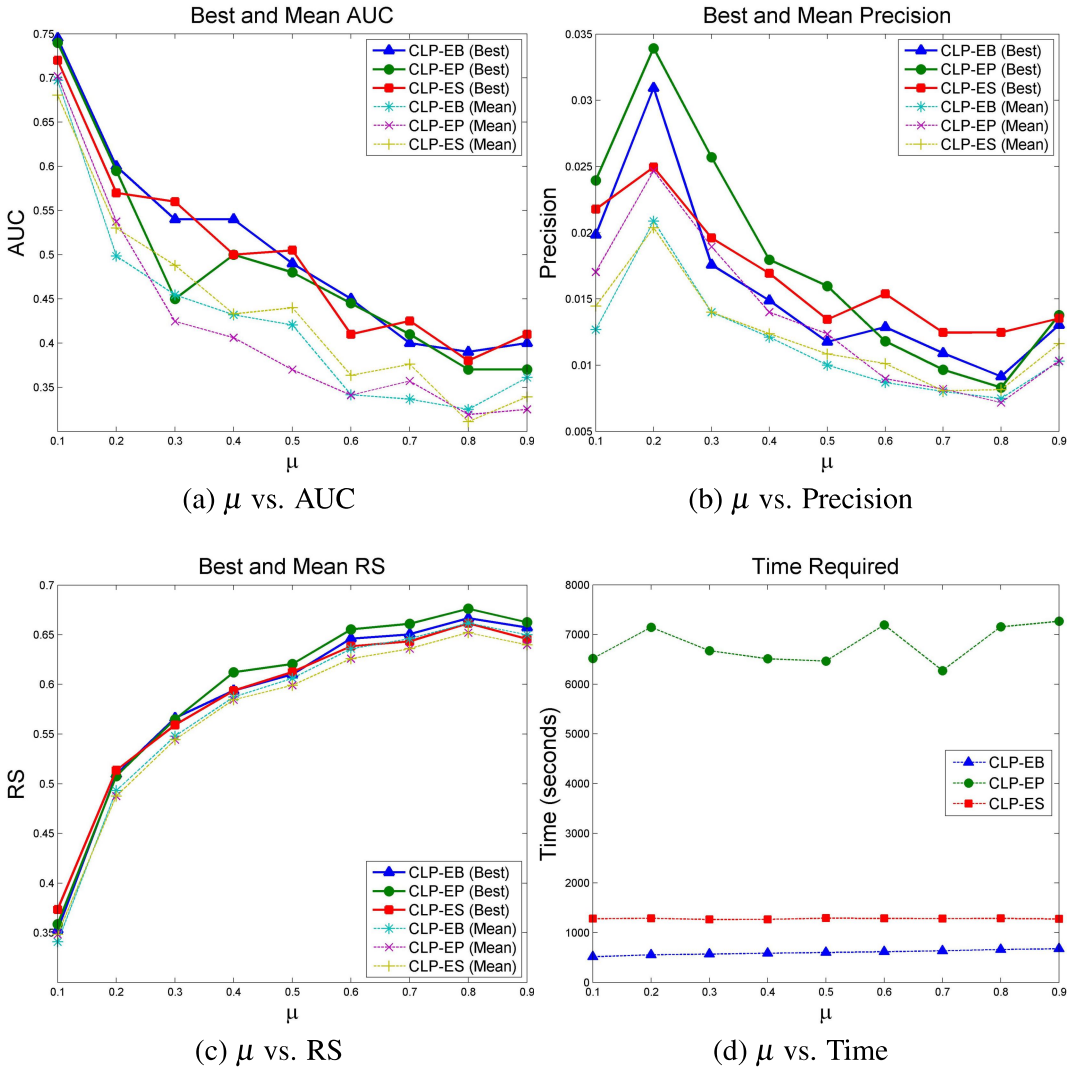


	Dolphin	Football	Karate	Strike
CLP-EB	31.534	123.7174	8.2946	3.8517
CLP-EP	25.7658	94.5431	6.3763	3.5161
CLP-ES	1.60E+03	4.63E+03	367.4548	257.7099

Figure 6.1: Time required to complete 10-fold cross-validation

that results are statistically significant as the best performing algorithm(s). Clearly, CLP-ES performs best in terms of AUC values. CLP-ES shows best AUC values on Dolphin, Football and Karate networks, and it shows highest mean AUC values on Dolphin and Football networks. CLP-EB performs best on Strike network both in terms of mean and best AUC values. Although on Karate network, CLP-ES shows best AUC value but CLP-EP show best mean AUC value. CLP-EP performs better than CLP-EB on both Dolphin and Football networks.

Table 6.2 presents best, mean and standard deviation of Precision for 10-fold cross-validation on four networks: Dolphin, Football, Karate and Strike. Both CLP-EP and CLP-ES performs almost same in terms of Precision. CLP-EP shows best Precision values on Dolphin and Strike networks, while CLP-ES shows best values on Football and Karate networks. Again, CLP-EP shows highest mean values on Football and Karate networks, while CLP-ES shows highest mean Precision values on Dolphin network. CLP-ES shows worst results on strike network. As a matter of fact, execution of CLP-ES on Strike network was not possible due to a loop present the network. Some additional links had



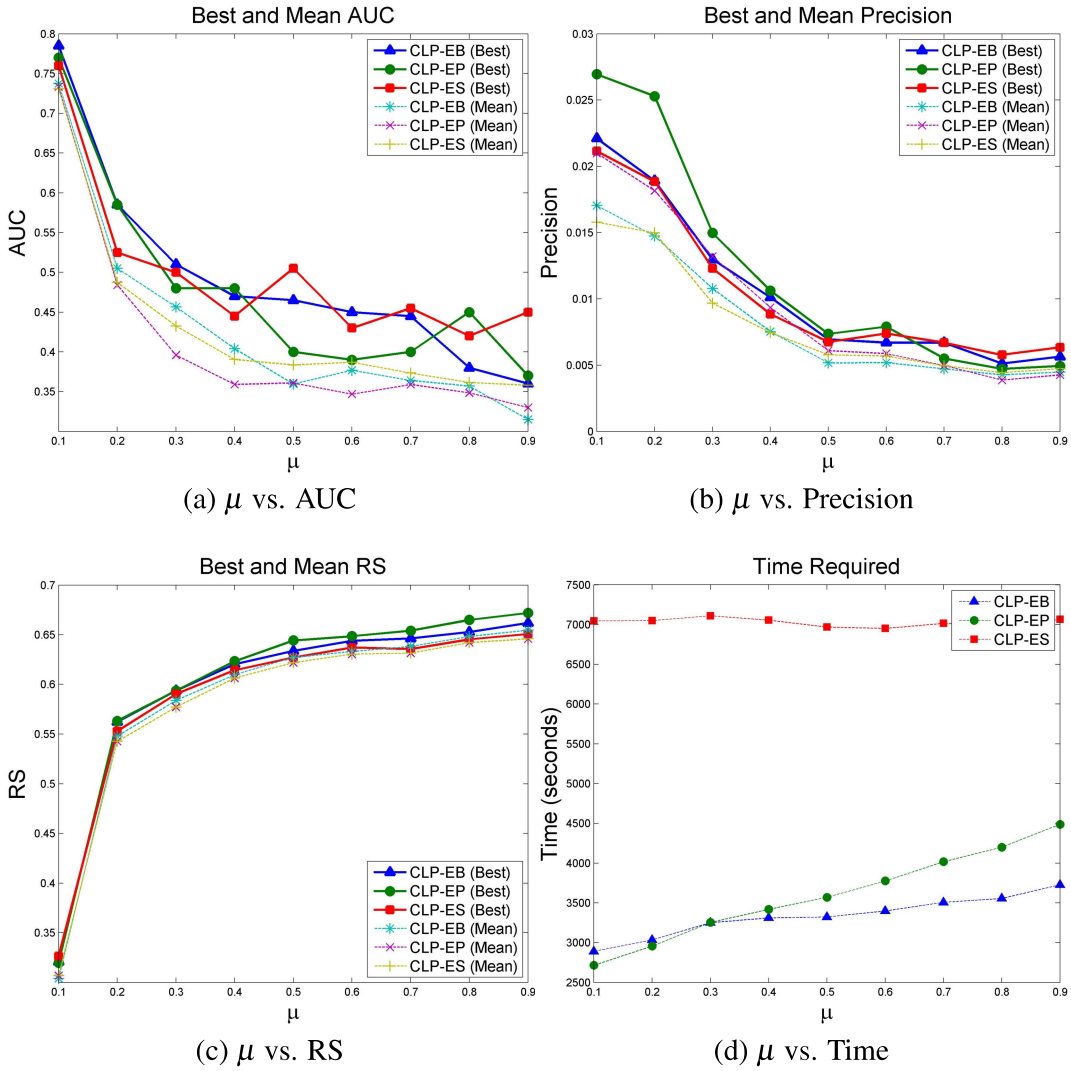
(a) μ vs. AUC (b) μ vs. Precision

(c) μ vs. RS (d) μ vs. Time

Figure 6.2: AUC, Precision, RS and Time on LFR1 networks with 500 nodes.

to include in the network for creating alternative paths to come out of that loop. The performance of CLP-EB is poor on all four networks in terms Precision values.

Table 6.3 presents best, mean and standard deviation of RS values for 10-fold cross-validation on four networks: Dolphin, Football, Karate and Strike. Although, performance of CLP-EB showed poor accuracy, quality of predicted links better as indicated by the RS values. CLP-EB shows best performance in terms of RS values on all networks. CLP-ES results better RS values than CLP-EP all four networks. In fact, the RS values



(a) μ vs. AUC (b) μ vs. Precision

(c) μ vs. RS (d) μ vs. Time

Figure 6.3: AUC, Precision, RS and Time on LFR2 networks with 1000 nodes.

are slightly lagging behind the RS values of CLP-EB. Moreover, CLP-ES shows highest RS value on football network and highest mean RS value on Strike network.

Results on Table 6.1, Table 6.2 and Table 6.3 indicate that CLP-EB, CLP-EP and CLP-ES show different results in terms AUC, Precision and RS. For instance, CLP-ES performs better in terms of AUC but not in terms of Precision and RS. Similarly, CLP-EB and CLP-EP show better results only in terms of RS and Precision respectively. Wilcoxon signed rank test is considered to show whether AUC, Precision, and RS values of three

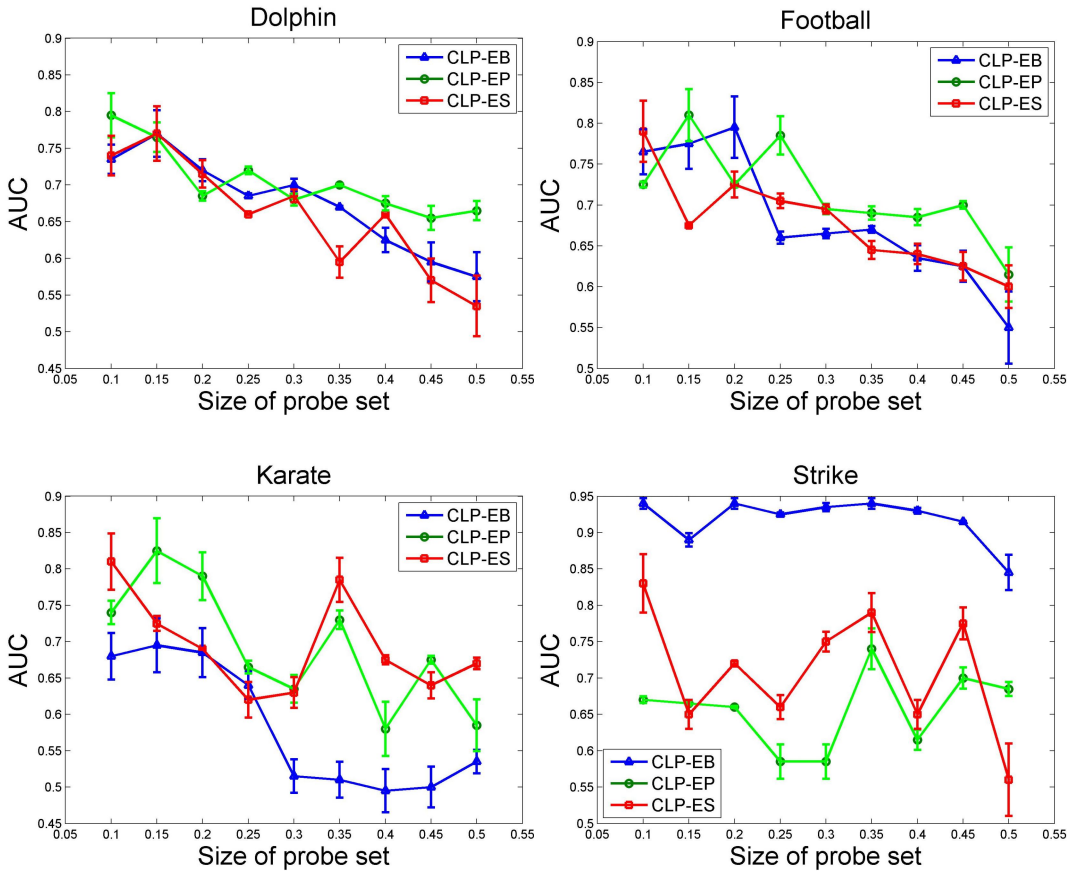


Figure 6.4: AUC values of predicted missing links for different sizes of probe set.

algorithms are statistically different or not. Wilcoxon signed-rank test is a non-parametric statistical hypothesis test and it is used when two related samples or matched samples are compared. Wilcoxon signed-rank test does not make assumptions about the distribution of the data. The W values of AUC, Precision and RS are computed by considering each pair of algorithms on all of the four networks. Table 6.4 presents the W values obtained against AUC, Precision and RS values. Confidence level $\alpha = 0.05$ and the number of samples $n = 10$ are considered. From the table of criteria for W value, it is already know that $W_{(\alpha=0.05,n=10)} = 8$. Clearly, all the W values against AUC are greater than $W_{(0.05,10)}$ on all networks, which indicates that there are significant difference among the AUC values of all algorithms. Similarly, W values against RS also indicates that there are

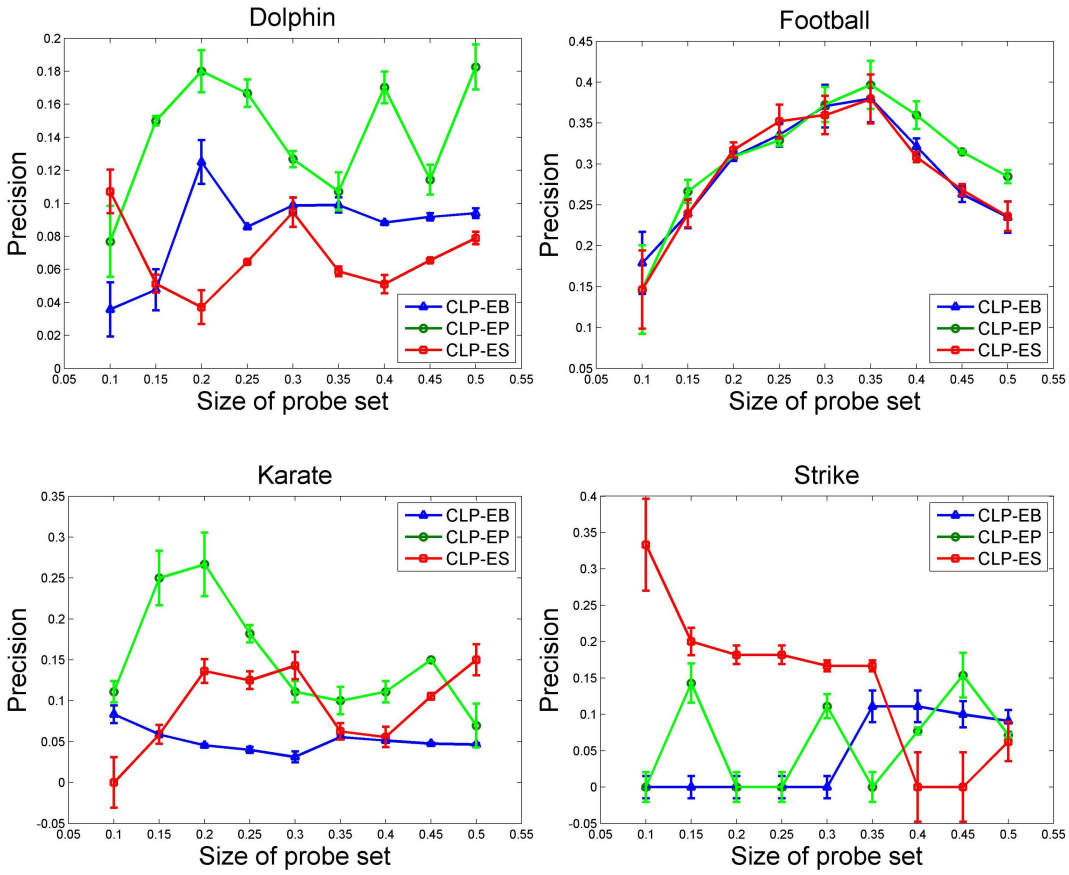


Figure 6.5: Precision values of predicted missing links for different sizes of probe set.

significant difference among the RS values of all algorithms on all of the four networks. However, W values against Precision only on Football and Karate networks indicate significant difference. Results indicate that Precision values of CLP-EB on Dolphin network are not different from the Precision values of CLP-ES and CLP-EP, while there has significant difference between Precision values of CLP-EP and CLP-ES. Ironically, W values against Precision on Strike network indicates that there is no significant difference among Precision values of algorithms.

Figure 6.1 presents time required by the algorithms for completing 10-fold cross-validation on Dolphin, Football, Karate and Strike networks. Results indicate CLP-EP is the best performer in terms of execution time. CLP-EP requires least time to complete 10-fold

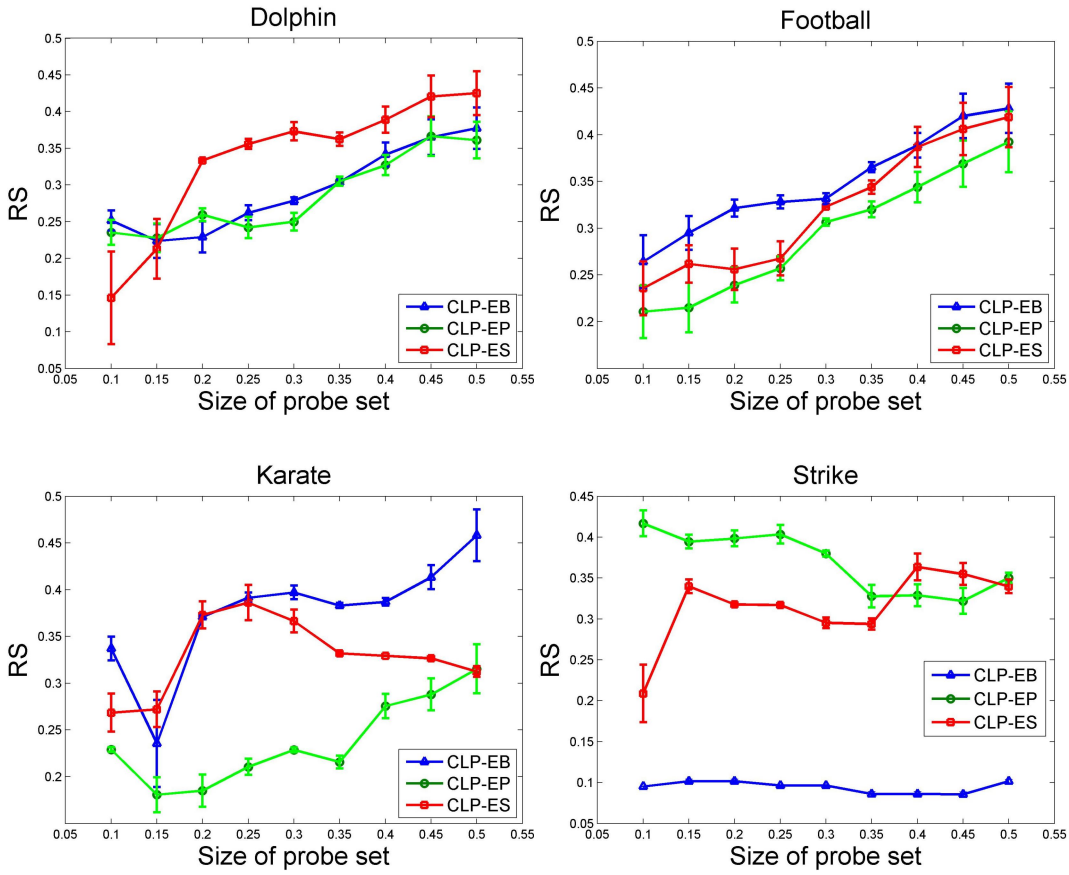


Figure 6.6: RS values of predicted missing links for different sizes of probe set.

cross-validation on all of the four networks. CLP-EB requires comparatively more time than CLP-EP, but the execution time quite low. CLP-ES is worst performer on all of the four networks. The execution times of CLP-ES on Dolphin and Football are 1000 times more than that of CLP-EB and CLP-EP.

Now, the performances of CLP-EB, CLP-EP and CLP-ES are compared on synthetic networks, namely LFR graphs with variation of parameter μ. Followed the same 10-fold cross-validation as earlier cases for all the graphs obtained for each μ values. Smaller μ values result in smaller number of neighbors of any node to be in different communities. Larger μ values ensure availability of more link information for processing. Predicted links will also have higher scores for larger μ values. Therefore, if an algorithm shows

better quality results i.e. high RS values for larger μ that will be more significant. However, accuracy may degrade for large μ values as most likely future links will have much higher scores than other non-observed links. Thus, if algorithm show higher accuracy i.e. AUC and Precision values for larger μ values that will be more significant. Since in earlier cases it has been noticed that CLP-ES is a time consuming process so the q parameter value used in edge spanning tree computation is set as 100 for reducing execution time of CLP-ES.

Figure 6.2 presents best and mean AUC, Precision and RS values obtained for different μ values on LFR1 networks with 500 nodes. Best AUC values of all three algorithms are almost same. However, best AUC values of both CLP-ES and CLP-EB are comparatively better than CLP-EP. Mean AUC values of CLP-ES are better than both CLP-EB and CLP-EP. Mean AUC values of CLP-EP are worst. Best Precision values of CLP-EP are better for smaller μ values, while CLP-ES shows better results for larger μ values. Although, mean Precision values of CLP-EP are highest for smaller μ values, all three algorithms show similar mean Precision for larger μ values. Quality of predicted links indicated by RS values seems to be same for all three algorithms. Quality of predicted links improves with increment of μ values. Time required by CLP-EP is highest, while CLP-EB requires least time. Despite CLP-ES is executed with limited q value, it still takes higher time than CLP-EB. Interestingly, the time required by all the algorithms remain almost constant.

Figure 6.3 presents best and mean AUC, Precision and RS values obtained for different μ values on LFR2 networks with 1000 nodes. Performance of all three algorithms in terms of best AUC values seems to be similar. For smaller μ values both CLP-EB and CLP-EP shows better results than CLP-ES. Although for smaller μ values, CLP-EB show better mean AUC values, for larger μ, CLP-ES shows highest mean AUC values. Mean AUC values of CLP-EP are worst for most μ values. Both best and mean Precision values of CLP-EP are mostly better than both CLP-EB and CLP-ES. Like LFR1 networks, same

trend can be noticed on RS values of all of the three algorithms. Time required by CLP-ES is much higher than both CLP-EB and CLP-EP. Although for some smaller μ values, CLP-EB takes more time than CLP-EP, as μ values increases CLP-EB takes less time.

Furthermore, sensitivity of CLP-EB, CLP-EP and CLP-ES to probe set E^P is also analyzed. Since training set $E^T = E - E^P$ is used for predicting links, the original network has been changed. Therefore, it is required to check how the performance of algorithms are effected by the modified network. The size of probe set E^P is varied from 10% of E to 50% of E. Results of probe set size variation are presented in Figure 6.4, Figure 6.5 and Figure 6.6. As indicated by the values of AUC, Precision and RS, clearly CLP variants are insensitive to the probe set size. Although, in some cases, like Dolphin and Football networks, AUC values degrade with increment in probe set size, while RS values improves with increment in probe set size. Nevertheless, mostly algorithms are insensitive to probe set size as indicated by AUC, Precision and RS. The insensitivity of algorithms to probe set size is more clear with Precision values as shown in Figure 6.5. Although RS increases with probe set size, in the case of link prediction correctness is more important. Therefore, smaller probe sets are recommended since in some cases AUC degrades with size of probe set, 10% of the total edge will be ideal probe set.

6.5 Conclusion

In this chapter, a community-based link prediction (CLP) scheme is proposed. Three variants of CLP are proposed: CLP-EB, CLP-EP, and CLP-ES. Comparative analysis is performed on various real-world networks as well as on synthetic networks. The analysis revealed following facts.

- Incorporation of community structure in link prediction accounts positive influence for intra-community future links and negative influence for inter-community links.
- Results indicated by three metrics AUC, Precision and RS show that performance of all three variants are almost same.
- CLP-ES is too much time, while both CLP-EB and CLP-EP require almost similar time for link prediction.
- The time complexity of both CLP-EB and CLP-EP is shown to be $O(n^2)$, while $O(n^3 \log n)$ for CLP-ES.
- Overall, CLP-EP is most efficient one among three variants in terms of time requirement on real-world networks, while CLP-EB on synthetic networks.