From competition to collaboration: Ensembling similarity-based heuristics for supervised link prediction in biological graphs

Md Kamrul Islam, Sabeur Aridhi, Malika Smail-Tabbone

From competition to collaboration: Ensembling similarity-based heuristics for supervised link prediction in biological graphs

Md Kamrul Islam[0000−0003−3072−3982], Sabeur Aridhi[0000−0002−3657−3762], and Malika Smail-Tabbone[0000−0002−8119−2117]

Universite de Lorraine, CNRS, INRIA, LORIA, 54000 Nancy, France
{kamrul.islam,sabeur.aridhi,malika.smail-tabbone}@loria.fr

Abstract. Link prediction is a fundamental problem in the field of graph mining. The aim of link prediction is to infer/discover unobserved links in graphs. Link prediction in biological graphs is highly challenging. There exist many similarity-based methods in the literature for link prediction. These methods compete for victory in graphs from various domains. Unfortunately, they are efficient only in some specific graphs, and no one wins in all graphs. In this paper, we study some well-known similarity-based methods and consider them as independent features to define a feature set. The feature set is then used to train traditional supervised learning methods for link prediction in biological graphs. We evaluate the methods on ten biological graphs from different organisms. Experimental results show that the similarity-based methods collaboratively improve prediction performance, and are even comparable to high-performing embedding-based methods in some biological graphs. We compute the importance score of similarity-based features in order to explain the leading features in a graph.

Keywords: Biological graphs · Link prediction · Similarity-based heuristics · Supervised learning.

1 Introduction

Many complex biological systems can be well-represented with graphs where a node represents a biological entity (e.g. protein, gene, etc.) and a link represents the interaction between two entities. Most real-world biological graphs are incomplete in nature. For example, 99.7% of the molecular interactions in human cells are still not known[1]. The links in biological graphs must be validated by field and/or laboratory experiments, which are expensive and time consuming. Researchers have developed link prediction methods to compute the plausibility of a link between two unconnected nodes in a graph to avoid the blind checking of all possible interactions. Formally, link prediction is the task of predicting the likelihood of a link between two nodes based on available topological/attribute information of a graph[2]. Link prediction methods help us toward a deep understanding of the structure, evolution, and functions of biological graphs [3].

Similarity-based methods are the simplest and unsupervised methods of link prediction in biological graphs, which define the proximity of a link by the similarity between its end nodes. The great advantage of these methods is their interpretibility which is essential for any biological system [4]. However, each of the similarity-based methods performs well only in some particular graphs and no one wins in all graphs. These methods necessitate manually formulating various heuristics based on prior beliefs or extensive knowledge of various biological graphs. The lack of universal applicability of similarity-based methods motivates researchers to study machine learning methods to automatically learn the heuristics from a graph. To learn the appropriate heuristics automatically from a graph, researchers have developed embedding-based methods which represent nodes, edges, graphs in low dimensional vector space [5]. The embedding-based method has become a popular link prediction tool in graphs over the last decade. These methods show impressive link prediction performance in most of the graphs. The downside of embedding-based methods is that they seriously suffer from the well-known 'black-box' problem. As the link decisions in biological graphs are critical, a link prediction method should be sufficiently interpretable to achieve trust among stakeholders [6]. The requirement for link prediction methods to be interpretable may limit the use of embedding-based methods in real-world biological systems. Researchers are still working on opening the 'black-box' of embedding-based methods [9, 10].

Another group of link prediction methods is developed based on traditional supervised learning-based methods. These methods extract features from a graph and train a traditional classifier for the link prediction task [11–17]. These methods are nearly as performant as embedding-based methods and as interpretable as similarity-based methods in many biological graphs. These methods describe the link prediction problem as a link classification problem with two classes: existence and absence of a link. In this paper, we intend to investigate whether the existing similarity-based heuristics collaboratively improve the link prediction performance in biological graphs. We study similarity-based heuristics for feature extraction and utilize the features in supervised learning-based classifiers for link prediction in biological graphs. We find that this is not the first attempt to study supervised learning methods to link prediction problem in graphs. But there are important differences between past works [12, 18, 19] and this study. The existing methods mostly focus on node attributes for extracting features which are application dependent. However, node attributes are not available in many real-world biological graphs. In contrast, our supervised learning-based method is developed based on only the topological features (similarity-based heuristics). Kumari et al [17] studied a few local (four) and global (three) similarity heuristics for supervised link predictions, which is the closest work in the literature to our study. However, for large graphs, global methods are not the best option as they are computationally expensive [20]. In this study, we enrich the feature set by including fourteen local similarity-based heuristics. In addition, we extract few other topological features of nodes and derive link-based features based on end node features. We study these features in supervised machine

learning methods for link prediction in biological graphs. We see that supervised learning methods show comparable prediction results in many of the biological graphs. We also demonstrate the feature importance in different datasets for different supervised learning-based methods.

1.1 Similarity-based link prediction

Link prediction is the task of discovering or inferring a set of non-existing links in a graph based on the current snapshot of the graph. Similarity-based is the simplest category of link prediction methods, which is formulated based on the assumption that two nodes interact if they are similar in a graph [20]. Generally, these methods compute similarity scores of non-existent links, sort the links in decreasing order of their scores and top-L links are predicted as potential existent links. Defining the similarity is a crucial and non-trivial task which differs from graphs to graphs [20]. Consequently, numerous similarity-based methods exist in the literature. These methods are broadly categorized into three categories: local, global and quasi-local methods. Local methods are developed based on local topological or neighbourhood information, whereas global methods use the global topological information of graphs to define similarity functions [20]. Quasi-local methods consider the neighbourhood up-to a predefined hop for defining the similarity function. The high computational time of global methods motivates us to study only local and quasi-local methods. We study fourteen well-known local similarity-based methods for link prediction in graphs, thirteen of which are summarized in Table. 1 local and one quasi-local. We summarize the similarity-based methods and the rest one (Preferential Attachment (PA)) in Table. 2 with basic principles and the definition of similarity functions.

2 Methodology

In a broader sense, we consider the similarity-based heuristics as individual features to generate the feature set for a supervised learning-based classifier.
We describe each of the steps in Sections 2.1-2.3.

2.1 Feature extraction

The most crucial task of a supervised learning-based classifier is to define an appropriate feature set [12]. Given a graph and a train set of links, we extract structural features for the train links. When extracting the features of a link, the link is temporarily removed from the graph and re-connected after feature extraction to ensure that the extracted features are not biased by the existence of the train link. We are motivated to use only topological features for defining our feature set as they exist in all kinds of graphs. Our feature set contains twenty topological features which are broadly categorized into two categories: similarity-based and derived link features (Fig. 1).

Table 1: Summary of similarity-based methods. Each method is considered as an individual link feature. $S(x, y)$ is the similarity function between two end nodes x and y. Γx and Γx denote the neighbour sets of nodes x and y respectively. A is the adjacency matrix and λ is a free parameter.

Method	**Principle**	**Similarity-function**
Common Neighbours (CN) [21]	Two nodes are more likely to be linked if they have more neighbours in common.	$CN(x,y) = \lvert\Gamma x \cap \Gamma y\rvert$
Adamic-Adar (AA) [22]	A variant of CN in which each common neighbour is penalized logarithmically by its degree.	$AA(x,y) = \sum_{z\in\Gamma x\cap\Gamma y} \frac{1}{log\lvert\Gamma z\rvert}$
Resource Allocation (RA) [23]	Based on the resource allocation mechanism, the high degree common neighbours will be penalized even more.	$RA(x,y) = \sum_{z\in\Gamma x\cap\Gamma y} \frac{1}{\lvert\Gamma z\rvert}$
Jaccard Index(JA) [24]	The score is punished for each non-common neighbour in the normalization of CN.	$JA(x,y) = \frac{\lvert\Gamma x\cap\Gamma y\rvert}{\lvert\Gamma x\cup\Gamma y\rvert}$
Salton Index(SA) [25]	The cosine similarity between adjacency vectors for a pair of nodes is used to compute the link probability.	$SA(x,y) = \frac{\lvert\Gamma x\cap\Gamma y\rvert}{\sqrt{\lvert\Gamma x\rvert\times\lvert\Gamma y\rvert}}$
Sørensen Index(SO) [26]	The overall fraction of common neighbours from a local perspective is what the link prediction is described as.	$SO(x,y) = \frac{2\times\lvert\Gamma x\cap\Gamma y\rvert}{\lvert\Gamma x\rvert+\lvert\Gamma y\rvert}$
Hub Promoted Index (HPI) [27]	Link establishment between high-degree nodes and hubs is encouraged.	$HPI(x,y) = \frac{\lvert\Gamma x\cap\Gamma y\rvert}{max(\lvert\Gamma x\rvert,\lvert\Gamma y\rvert)}$
Hub Depressed Index (HDI) [27]	Link establishment between low-degree nodes and hubs is encouraged.	$HDI(x,y) = \frac{\lvert\Gamma x\cap\Gamma y\rvert}{min(\lvert\Gamma x\rvert,\lvert\Gamma y\rvert)}$
Local Leicht-Holme- Newman (LLHN) [28]	The real and expected number of shared neighbours are used to define the similarity between two nodes.	$LLHN(x,y) = \frac{\lvert\Gamma x\cap\Gamma y\rvert}{\lvert\Gamma x\rvert\times\lvert\Gamma y\rvert}$
Cannistrai–Alanis–Ravai (CAR) [29]	In measuring the similarity score between two end nodes, level-2 linkages are combined with shared neighbourhood information.	$CAR(x,y) = \sum_{z\in\Gamma x\cap\Gamma y} 1 + \frac{\lvert\Gamma x\cap\Gamma y\cap\Gamma z\rvert}{2}$
Clustering Coefficient-based Link Prediction (CCLP) [30]	The influence of each shared neighbour is quantified by using the node's local clustering coefficient.	$CCLP(x,y) = \sum_{z\in\Gamma x\cap\Gamma y} CC_z$
Node and Link Clustering(NLC) [31]	The contribution of each common neighbor is quantified by using the node's node and link clustering coefficients.	$NLC(x,y) = \sum_{z\in\Gamma x\cap\Gamma y} \left(\frac{CN(x,z)}{\lvert\Gamma z\rvert-1}\times CC_z + \frac{CN(x,z)}{\lvert\Gamma z\rvert-1}\times CC_z\right)$
Local Path Index(LPI) [32]	Similarity is calculated using the second and third order paths between the end nodes.	$LPI(x,y) = [A^2 + \lambda A^3]_{x,y}$

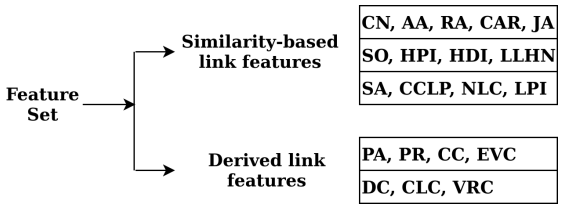


Fig. 1: Feature set for supervised learning

Table 2: Summary of derived link features: The derived link feature function $S(x, y)$ is defined based on end nodes features.

Feature	Principle	Link feature function
Preferential Attachment (PA) [33]	Based on the rich-get-richer principle, in which the link likelihood between two high-degree nodes is greater than that between two low-degree nodes.	$PA(x, y) = \|\Gamma x\| \times \|\Gamma y\|$
Pager Rank (PR) [34]	PageRank computes a ranking of the nodes based on the structure of the links.	$PR(x, y) = PR(x) + PR(y)$
Clustering Coefficient (CC) [35]	The clustering coefficient of a node is the fraction of possible triangles through that node that exist	$CC(x, y) = CC(x) + CC(y)$
Degree Centrality (DC) [36]	The degree centrality for a node is the fraction of nodes it is connected to.	$DC(x, y) = DC(x) + DC(y)$
Eigen vector centrality (EVC) [37]	Eigenvector centrality computes the centrality for a node based on the centrality of its neighbors.	$EVC(x, y) = EVC(x) + EVC(y)$
Closeness Centrality(CLC) [38]	Closeness centrality of a node is the reciprocal of the average shortest path length to other reachable nodes.	$CLC(x, y) = CLC(x) + CLC(y)$
Vote Rank Centrality(VRC) [39]	Ranking of the nodes based on a voting scheme where a node casts votes to its neighbours. A node with the highest votes has the best (lowest) ranking.	$VRC(x, y) = \frac{1}{VRC(x)} + \frac{1}{VRC(y)}$

Similarity-based link features : We define the link-based features as the features which are related to the common topological information of end-nodes of a link. We use thirteen existing similarity-based heuristics as link-based features, which are summarized in Table 1. For instance, the number of common neighbours of end nodes of a link is used as the common neighbour(CN) feature.

Derived link features : Few link-based features are derived from the individual features of the link's end nodes. We summarized six derived features in Table 2. These features are related to the topological information of individual nodes only. For example, the degree of end nodes is multiplied in Preferential Attachment (PA) to define the similarity score. Note that the link features in Table 2 except PA are not directly defined in the literature. We derive the link features based on the end node feature. To compute the link feature, features of end nodes are simply added except PA. As the voterank centrality computes low ranks for high-influencing nodes in a graph, the reciprocals of the voterank scores of end nodes are summed to define the voterank centrality feature.

2.2 Feature scaling

In general, the magnitude scale for different features in different graphs varies [7, 8]. Supervised learning-based methods are easily affected by the non-uniform scaling as there is a high chance that features with higher magnitude play a more decisive role during the training of a classifier. But, it is not desirable for the classifier to be biased towards one particular feature. Hence, we normalize each feature in the range of 0-1.

2.3 Classifier training and link prediction

For the link prediction task, we train a traditional supervised machine learning classifier to classify a link into either existent or non-existent classes. There exist many classifiers in the literature which perform better than others in some particular datasets. In this paper, we study three traditional classifiers: Support Vector Machine(SVM) with RBF kernel, Decision Tree, and Logistic Regression. We extract the features of the test links and classify them into existent or non-existent classes using a trained classifier to evaluate the link prediction performance.

3 Experiments

3.1 The baselines

To evaluate the prediction performance of supervised learning methods, we consider two categories of link prediction methods: similarity-based and embedding-based methods.

For the similarity-based category, we consider all the heuristics in Table 1 in Table 2. For the embedding-based methods, we choose two popular methods: Node2Vec [40] and SEAL [41]. We shortly describe Node2Vec and SEAL methods. For more details, we refer to the original papers. **Node2Vec** [40] is a classical skip-gram model-based graph embedding method which learns node embeddings by optimizing a neighbourhood preserving objective function. It makes an interpolation between BFS(Breadth First Search) and DFS(Depth First Search) to define a 2^{nd} order random walk. A fixed size neighbourhood is sampled using the 2^{nd} order random walk and fed into the well-known skip-gram model [42] to learn the node embedding. The link embedding is then computed as the Hadamard product of the end node embeddings. A logistic regression-based classifier is then trained for the link prediction task. SEAL, the second embedding-based approach, is based on neural networks (NN). **Learning from Sub-graphs, Embeddings and Attributes (SEAL)** utilizes the latent and explicit features of end nodes and structural information of the graph to learn the link embedding. SEAL starts with extracting a h-hop neighbouring sub-graph and node labeling by a double radius node labeling (DRNL) algorithm. In the second step, the labelled sub-graph is then used to generate the structural encoding. The link embedding is the concatenation of structural encoding, pre-computed latent encoding and explicit feature encoding. In the final step, a neural network(NN) is trained for link prediction task.

3.2 Experimental datasets

In this study, we focus on only biological graphs. For evaluating performance, we collect six biological graphs from the Network Repository [1]. Table 3 summarizes the topological statistics and descriptions of the graph datasets.

[1] https://networkrepository.com/bio.php

Table 3: The graph datasets: number of nodes($|\mathbf{V}|$), links($|\mathbf{E}|$), average node degree (NDeg), average clustering coefficient (CC), and description.

Graph	Organism	$\|\mathbf{V}\|$	$\|\mathbf{E}\|$	NDeg	CC	Description
CE-GT [43]	Worm	924	3239	7.01	0.605	Nodes: Genes in C. elegans Links: Gene functional associations in C.elegans
CE-HT [43]	Worm	2617	2985	2.28	0.008	Nodes: Proteins in C. elegans Links: High-throughput protein-protein interactions
Celegans [43]	Worm	453	2040	9.01	0.647	Nodes: Substrates in Caenorhabditis elegans Links: Metabolic reactions between substrates in C.elegans
CE-LC [44]	Worm	1387	1648	2.37	0.076	Nodes: Proteins in C.elegans worm Links: Small/medium-scale protein-protein interactions (compiled from protein-protein interaction data bases)
Diseasome [45]	Human	516	1188	4.61	0.636	Nodes: Known genetic disorders in H.sapiens Links: Connections between pair of disorders when they share minimum one gene.
DM-HT [45]	Fly	2989	4660	3.12	0.009	Nodes: Proteins in D.melan-ogaster fly Links: High-throughput protein-protein interactions
DM-LC [44]	Fly	658	1129	3.43	0.105	Nodes: Proteins in D.melanogaster fly Links: Small/medium-scale protein-protein interactions (compiled from protein-protein interaction data bases)
HS-HT [44]	Human	2570	13691	10.65	0.169	Nodes: Proteins in human Links: Protein-protein interactions in human protein network
SC-LC [44]	Yeast	2004	20452	20.41	0.168	Nodes: Proteins in S.cerevisiae yeast Links: Small/medium-scale protein-protein interactions in yeast network
Yeast [44]	Yeast	2114	2277	2.15	0.059	Nodes: Proteins in S.cerevisiae yeast Links: Protein-protein interactions in yeast network

The link prediction performance is evaluated using a random sampling validation protocol [41, 7, 8]. For a graph dataset, train and test sets are prepared by splitting the existent links. The train set consists of 90% existent and an equal number of non-existent links. The test set contains the remaining 10% existent and equal number of non-existent links. To prepare five train and five test sets for each graph, we repeat the link splitting operation five times independently. The datasets are available in a GitLab repository [2].

3.3 Evaluation metrics

The link prediction problem is considered as a binary classification problem [46]. A traditional classifier, in general, learns a threshold to classify links as existent or non-existent. However, for similarity-based link classification methods, we find no standard approach for computing the threshold. The threshold is calculated in an optimistic manner. We first normalize the link scores to a range of 0-1 and then use the normalized scores to compute a ROC curve. The curve gives the true

[2] https://gitlab.inria.fr/kislam/supervised-lp

positiverate (TPR) and false positive rate (FPR) for different score threshold settings.The threshold point with the highest [TPR+(1-FPR)] is computed as the *threshold* as we want to maximize TPR as well as minimize FPR. We classify links based on this threshold. A link with a $score >= threshold$ is classified as existent and non-existent otherwise. Based on the true and predicted classes of links, we define four metrics: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP is the number of existent links predicted to be existent, TN is the number of non-existent links predicted to be non-existent, FP is the number of non-existent links predicted to be existent, and FN is the number of existent links predicted to be non-existent links. We compute the following three well-known metrics using these four metrics.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

3.4 Results and discussion

In this section, we describe the prediction performance of supervised learning-based methods on six biological graphs. We also illustrate the importance of the features in graphs.

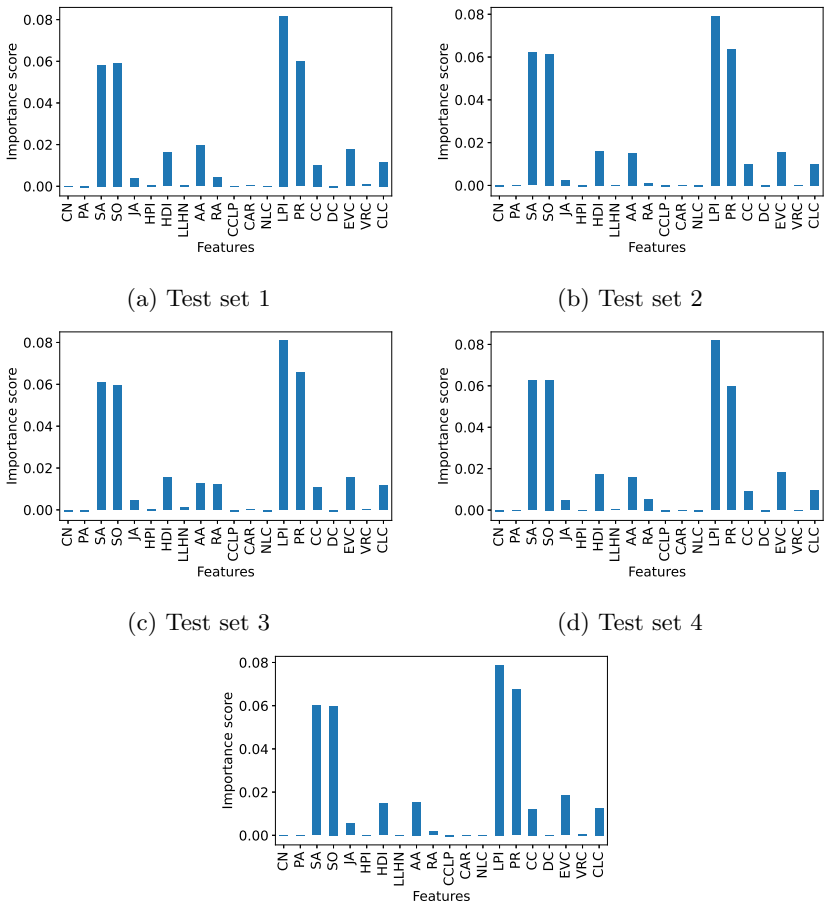
Prediction performance The prediction performance is computed for all methods over all the five sets for each graph, and the average scores are recorded. We do not include the standard deviation results as the values are very low in all the experiments. The precision, recall and F1 scores are tabulated Table 4, where the best two similarity-based methods are denoted with Sim^1 and Sim^2. We compute the precision scores of similarity-based methods in a optimistic way. The precision scores of similarity-based methods (best and second best) are very high and highest among all the methods in all the graphs, as shown in the table. This demonstrates the ability of similarity-based methods to predict high-quality links. However, the recall scores are low, implying that these methods identify the majority of existing test links as non-existent. As a result, the F1 score for similarity-based methods is very low. We also see that, as expected, the two best-performing similarity-based methods differ for different datasets. Among the supervised learning methods (SVM, DT, LR), DT shows the worst prediction results, but it is still much better than similarity-based methods. The other two classifiers have similar performance scores. The performance of the other two classifiers in terms of prediction scores is impressive. Yet in many graphs, supervised learning-based classifiers show superior prediction performance than embedding-based methods. Relating the performance to graph properties, we see that traditional classifiers outperform embedding-based methods in dense

Table 4: Performance metrics: The dataset-wise best and second best precision, recall and F1 scores are indicated in bold and underline. The best and second best similarity-based methods are denoted with Sim^1 andSim^2 respectively. For Sim^1 and Sim^2 methods, the methods are specified and the performance scores are given in ().

Datasets	Metric	Sim^1	Sim^2	N2V	SEAL	SVM	DT	LR
CE-GT	Precision	NLC **(0.960)**	JA (0.896)	0.707	0.842	0.842	0.828	0.901
	Recall	NLC (0.039)	JA (0.042)	0.707	**0.931**	0.827	0.776	0.900
	F1	NLC (0.075)	JA (0.078)	0.707	0.885	0.834	0.801	**0.901**
CE-HT	Precision	RA **(0.996)**	AA (0.996)	0.596	0.705	0.753	0.745	0.752
	Recall	RA (0.001)	AA (0.001)	0.593	**0.791**	0.529	0.519	0.510
	F1	RA (0.002)	AA (0.002)	0.594	**0.745**	0.622	0.612	0.608
Celegans	Precision	RA **(0.938)**	CCLP (0.932)	0.778	0.806	0.899	0.850	0.907
	Recall	RA (0.042)	CCLP (0.041)	0.777	0.888	0.899	0.830	**0.906**
	F1	RA (0.08)	CCLP (0.079)	0.778	0.845	0.899	0.840	**0.906**
CE-LC	Precision	AA **(0.969)**	RA **(0.969)**	0.658	0.763	0.715	0.763	0.789
	Recall	AA(0.009)	RA(0.009)	0.647	**0.794**	0.620	0.584	0.673
	F1	AA(0.028)	RA(0.028)	0.652	**0.778**	0.664	0.662	0.726
Diseasome	Precision	NLC **(0.991)**	AA (0.988)	0.757	0.914	0.926	0.800	0.927
	Recall	NLC (0.035)	AA (0.040)	0.756	0.896	0.919	0.692	**0.920**
	F1	NLC (0.067)	AA (0.078)	0.756	0.905	0.922	0.742	**0.924**
DM-HT	Precision	CCLP **(0.999)**	NLC (0.998)	0.712	0.720	0.780	0.796	0.770
	Recall	CCLP (0.001)	NLC (0.001)	**0.704**	0.703	0.657	0.661	0.644
	F1	CCLP (0.002)	NLC (0.002)	0.708	0.712	0.714	**0.722**	0.701
DM-LC	Precision	PA **(0.979)**	CCLP(0.944)	0.696	0.790	0.829	0.828	0.812
	Recall	PA(0.02)	CCLP(0.007)	0.688	**0.835**	0.771	0.770	0.777
	F1	PA(0.039)	CCLP(0.014)	0.692	0.812	**0.799**	0.798	0.794
HS-HT	Precision	NLC **(0.954)**	CCLP (0.949)	0.797	0.854	0.861	0.847	0.861
	Recall	NLC (0.031)	CCLP (0.031)	0.794	0.815	0.840	0.791	**0.848**
	F1	NLC (0.060)	CCLP (0.061)	0.796	0.834	0.850	0.818	**0.854**
SC-LC	Precision	NLC **(0.893)**	AA (0.873)	0.772	0.784	0.868	0.850	0.853
	Recall	NLC (0.035)	AA (0.036)	0.770	0.815	**0.849**	0.810	0.844
	F1	NLC (0.067)	AA (0.068)	0.771	0.799	**0.859**	0.829	0.849
Yeast	Precision	CCLP **(0.971)**	RA (0.967)	0.699	0.705	0.753	0.746	0.755
	Recall	CCLP (0.006)	RA (0.008)	0.699	**0.726**	0.567	0.551	0.598
	F1	CCLP (0.012)	RA (0.015)	0.699	**0.716**	0.647	0.634	0.668

graphs. This is intuitive as the majority of the studied similarity-based heuristics are based on common neighbours (see Table. 1). The performance scores of traditional classifiers are worse in the sparse graphs (CE-HT, CE-LC, Yeast), where embedding-based methods show better performance scores.

Feature importance In this section, we investigate the influence of each feature in a classifier for the link prediction task. To compute the feature importance coefficient, we use the Permutation importance module from the sklearn



(a) Test set 1

(b) Test set 2

(c) Test set 3

(d) Test set 4

(e) Test set 5

Fig. 2: Feature importance in HS-HT graph by Logistic regression classifier

python-based machine learning tool [3]. When a feature is unavailable, the coefficient is calculated by looking at how much the score (accuracy) drops [47]. The higher the coefficient, the higher the importance of the feature. In Fig. 2, we demonstrate the feature importance in the HS-HT biological graph in the logistic regression (LR) classifier to investigate how the importance of features differs in different sets of the same biological graph. In the LR classifier for the HS-HT biological graph, four features dominate. The dominance of multiple heuristics or features in a graph shows that heuristics that work collaboratively perform bet-

[3] https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html

ter than heuristics that work alone. We can also find that the feature importance coefficients in all five sets in the HS-HT graph are substantially identical.

We further investigate the importance score of features in three classifiers (SVM, DT, LR) for three different datasets. We evaluate the importance score of features for only one set for each graph. We see that different classifiers give different importance coefficients to different features in different datasets. In DM-HT dataset, all the classifiers compute high coefficient for LPI feature and they have close prediction performance (in Table 4). In the Celegans dataset, the HPI feature dominates in SVM and LR classifiers whereas LPI dominates in the DT classifier. In the Celegans dataset, SVM and LR outperform DT in terms of prediction (in Table 4), demonstrating that LR and SVM compute feature importance scores more correctly. Surprisingly, we see that DT has a tendency to give more importance to the LPI feature in these three datasets.

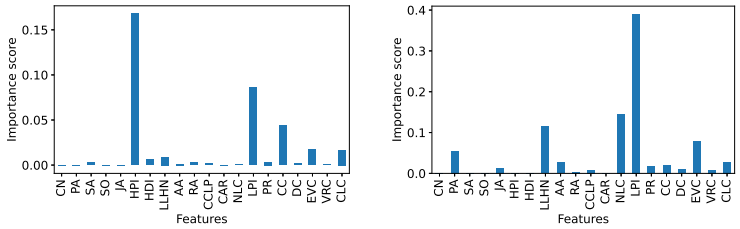
4 Conclusion

Do similarity-based heuristics compete or collaborate for link prediction task in graphs? In this article, we study this question. We study fourteen similarity-based heuristics in six biological graph from three different organisms. As expected, we observe they perform well only in some particular biological graphs and no one wins in all graphs. Rather than using them as standalone link prediction methods, we consider them as features for supervised learning methods. In addition, we derive six link features based on the node's topological information. Based on the twenty features, we train three traditional supervised learning methods: SVM, DT and LR-based classifiers. We see that the similarity-based heuristics collaboratively improve link prediction performance remarkably, even outperforming embedding-based methods in some graphs.

We propose three future dimensions of this study. Firstly, studying collaboration of similarity-based heuristics in large scale biological as well as social graphs could be a potential future work as the graphs in the current study are small/medium in size. Secondly, exploring some other heuristics might improve prediction performance in sparse graphs. The final future research could be studying other classifiers like Random Forest, AdaBoost, K-Neighbors for the link prediction task in graphs.

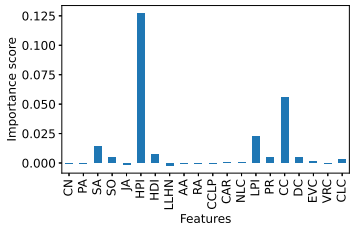
References

1. Stumpf, M. P., Thorne, T., De Silva, E., Stewart, R., An, H. J., Lappe, M., Wiuf, C. (2008). Estimating the size of the human interactome. Proceedings of the National Academy of Sciences, 105(19), 6959-6964.
2. Xu, Z., Pu, C., Yang, J.(2016). Link prediction based on path entropy. Physica A: Statistical Mechanics and its Applications, 456, 294-301.
3. Shen, Z., Wang, W. X., Fan, Y., Di, Z., Lai, Y. C.(2014). Reconstructing propagation networks with natural diversity and identifying hidden sources. Nature communications, 5(1), 1-10.

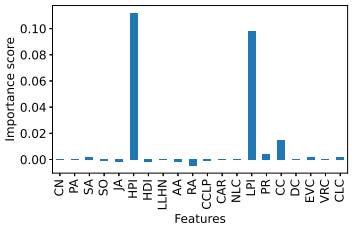


(a) SVM in Celegans

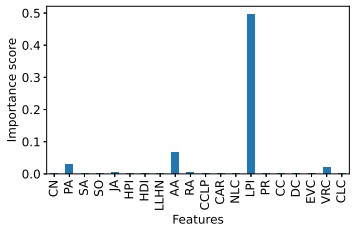
(b) DT in Celegans



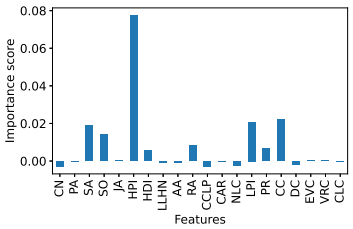
(c) LR in Celegans



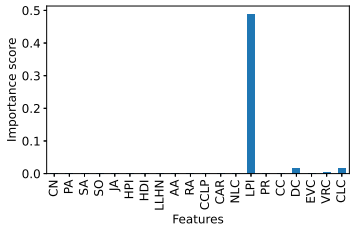
(d) SVM in Diseasome



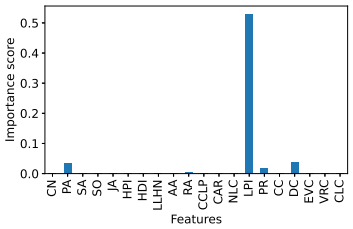
(e) DT in Diseasome



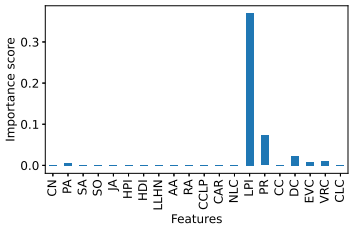
(f) LR in Diseasome



(g) SVM in DM-HT



(h) DT in DM-HT



(i) LR in DM-HT

Fig. 3: Feature importance in different datasets by different supervised methods: (a)-(c) in Celegans, (d)-(f) in Diseasome, (g)-(i) in DM-HT

4. Zhou, T., Lee, Y. L., Wang, G.(2021). Experimental analyses on 2-hop-based and 3-hop-based link prediction algorithms. Physica A: Statistical Mechanics and its Applications, 564, 125532.
5. Cui, P., Wang, X., Pei, J., Zhu, W. (2018). A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering, 31(5), 833-852.
6. Gerke, S., Minssen, T., Cohen, G. (2020). Ethical and legal challenges of artificial intelligence-driven healthcare. In Artificial intelligence in healthcare (pp. 295-336). Academic Press.
7. Islam, M. K., Aridhi, S., Smail-Tabbone, M. (2021). Appraisal study of similarity-based and embedding-based link prediction methods on graphs. Proceedings of the 10th International Conference on Data Mining & Knowledge Management Process, 81-92.
8. Islam, M. K., Aridhi, S., Smaïl-Tabbone, M. (2021). An experimental evaluation of similarity-based and embedding-based link prediction methods on graphs. International Journal of Data Mining & Knowledge Management Process, 11, 1-18.
9. Faber, L., Moghaddam, A. K., Wattenhofer, R. (2020). Contrastive Graph Neural Network Explanation. In Proceedings of the 37th Graph Representation Learning and Beyond Workshop at ICML 2020 (p. 28). International Conference on Machine Learning.
10. Yuan, H., Yu, H., Wang, J., Li, K., Ji, S. (2021). On explainability of graph neural networks via subgraph explorations. Proceedings of the 38th International Conference on Machine Learning.
11. Cukierski, W., Hamner, B., Yang, B. (2011, July). Graph-based features for supervised link prediction. In The 2011 International Joint Conference on Neural Networks (pp. 1237-1244). IEEE.
12. Al Hasan, M., Chaoji, V., Salem, S., Zaki, M. (2006, April). Link prediction using supervised learning. In SDM06: workshop on link analysis, counter-terrorism and security (Vol. 30, pp. 798-805).
13. Berton, L., Valverde-Rebaza, J., de Andrade Lopes, A. (2015, July). Link prediction in graph construction for supervised and semi-supervised learning. In 2015 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.
14. Benchettara, N., Kanawati, R., Rouveirol, C. (2010, September). A supervised machine learning link prediction approach for academic collaboration recommendation. In Proceedings of the fourth ACM conference on Recommender systems (pp. 253-256).
15. Ahmed, C., ElKorany, A., Bahgat, R. (2016). A supervised learning approach to link prediction in Twitter. Social Network Analysis and Mining, 6(1), 24.
16. Shibata, N., Kajikawa, Y., Sakata, I. (2012). Link prediction in citation networks. Journal of the American society for information science and technology, 63(1), 78-85.
17. Kumari, A., Behera, R. K., Sahoo, K. S., Nayyar, A., Kumar Luhach, A., Prakash Sahoo, S. (2020). Supervised link prediction using structured-based feature extraction in social network. Concurrency and Computation: Practice and Experience, e5839.
18. Liben-Nowell, D., Kleinberg, J. (2007). The link-prediction problem for social networks. Journal of the American society for information science and technology, 58(7), 1019-1031.
19. De Sá, H. R., Prudêncio, R. B. (2011, July). Supervised link prediction in weighted networks. In The 2011 international joint conference on neural networks (pp. 2281-2288). IEEE.
20. Martínez, V., Berzal, F., Cubero, J. C. (2016). A survey of link prediction in complex networks. ACM computing surveys (CSUR), 49(4), 1-33.

21. Lorrain, F., White, H. C. (1971). Structural equivalence of individuals in social networks. The Journal of mathematical sociology, 1(1), 49-80.
22. Adamic, L. A., Adar, E. (2003). Friends and neighbors on the web. Social networks, 25(3), 211-230.
23. Zhou, T., Lü, L., Zhang, Y. C. (2009). Predicting missing links via local information. The European Physical Journal B, 71(4), 623-630.
24. Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat, 37, 547-579.
25. Salton, G., McGill, M. J. (1983). Introduction to modern information retrieval. Mcgraw-hill
26. Sorensen, T. A. (1948). A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. Biol. Skar., 5, 1-34.
27. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N., Barabási, A. L. (2002). Hierarchical organization of modularity in metabolic networks. science, 297(5586), 1551-1555.
28. Leicht, E. A., Holme, P., Newman, M. E. (2006). Vertex similarity in networks. Physical Review E, 73(2), 026120.
29. Cannistraci, C. V., Alanis-Lobato, G., Ravasi, T. (2013). From link-prediction in brain connectomes and protein interactomes to the local-community-paradigm in complex networks. Scientific reports, 3(1), 1-14.
30. Wu, Z., Lin, Y., Wang, J., Gregory, S. (2016). Link prediction with node clustering coefficient. Physica A: Statistical Mechanics and its Applications, 452, 1-8.
31. Wu, Z., Lin, Y., Wan, H., Jamil, W. (2016). Predicting top-L missing links with node and link clustering information in large-scale networks. Journal of Statistical Mechanics: Theory and Experiment, 2016(8), 083202.
32. Lü, L., Jin, C. H., Zhou, T. (2009). Similarity index based on local paths for link prediction of complex networks. Physical Review E, 80(4), 046122.
33. Barabási, A. L., Albert, R. (1999). Emergence of scaling in random networks. Science, 286(5439), 509-512.
34. Langville, A. N., Meyer, C. D. (2005). A survey of eigenvector methods for web information retrieval. SIAM review, 47(1), 135-161.
35. Onnela, J. P., Saramäki, J., Kertész, J., Kaski, K. (2005). Intensity and coherence of motifs in weighted complex networks. Physical Review E, 71(6), 065103.
36. Dubitzky, W., Wolkenhauer, O., Cho, K. H., Yokota, H. (Eds.). (2013). Encyclopedia of systems biology (Vol. 402). New York: Springer.
37. Bonacich, P. (1987). Power and centrality: A family of measures. American journal of sociology, 92(5), 1170-1182.
38. Freeman, L. (1979). Centrality in networks: I. conceptual clarifications. social networks. Social Network.
39. Zhang, J. X., Chen, D. B., Dong, Q., Zhao, Z. D. (2016). Identifying a set of influential spreaders in complex networks. Scientific reports, 6, 27823.
40. Grover, A., Leskovec, J. (2016, August). node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 855-864).
41. Zhang, M., Chen, Y. (2018). Link prediction based on graph neural networks. Advances in Neural Information Processing Systems, 31, 5165-5175.
42. Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). Efficient estimation of word representations in vector space. In International Conference on Learning Representations.

43. Duch, J., Arenas, A. Community identification using extremal optimization, 2005. Phys. Rev. E, 72, 027104.
44. Cho, A., Shin, J., Hwang, S., Kim, C., Shim, H., Kim, H., ... Lee, I. (2014). WormNet v3: a network-assisted hypothesis-generating server for Caenorhabditis elegans. Nucleic acids research, 42(W1), W76-W82.
45. Goh, K. I., Cusick, M. E., Valle, D., Childs, B., Vidal, M., Barabási, A. L. (2007). The human disease network. Proceedings of the National Academy of Sciences, 104(21), 8685-8690.
46. Kumar, A., Singh, S. S., Singh, K., Biswas, B. (2020). Link prediction techniques, applications, and performance: A survey. Physica A: Statistical Mechanics and its Applications, 553, 124289.
47. Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.