A framework for empirically evaluating pretrained link prediction models

Olivares, E.S.; Boekhout, H.D.; Saxena, A.; Takes, F.W.; Cherifi, H.; Rocha, L.M.; ... ; Donduran, M.

A framework for empirically evaluating pretrained link prediction models

Emilio Sánchez Olivares, Hanjo D. Boekhout, Akrati Saxena, and Frank W. Takes

Leiden Institute of Advanced Computer Science,
Leiden University, the Netherlands
{h.d.boekhout, a.saxena, f.w.takes}@liacs.leidenuniv.nl

Abstract. This paper proposes a novel framework for empirically assessing the effect of network characteristics on the performance of pretrained link prediction models. In link prediction, the task is to predict missing or future links in a given network dataset. We focus on the pretrained setting, in which such a predictive model is trained on one dataset, and employed on another dataset. The framework allows one to overcome a number of nontrivial challenges in adequately testing the performance of such a pretrained model in a proper cross-validated setting. Experiments are performed on a corpus of 49 structurally diverse real-world complex network datasets from various domains with up to hundreds of thousands of nodes and edges. Overall results indicate that the extent to which a network is clustered is strongly related to whether this network is a suitable candidate to create a pretrained model on. Moreover, we systematically assessed the relationship between topological similarity and performance difference of pretrained models and a model trained on the same data. We find that similar network pairs in terms of clustering coefficient, and to a lesser extent degree assortativity and gini coefficient, yield minimal performance difference. The findings presented in this work pave the way for automated model selection based on topological similarity of the networks, as well as larger-scale deployment of pretrained link prediction models for transfer learning.

Keywords: Link Prediction, Transfer Learning, Pretrained Models

1 Introduction

In recent years, researchers have studied complex networks to understand and analyze the intricate relationships that underlie various real-world systems. Complex networks, characterized by their non-trivial topological structures, have applications in diverse fields such as the social sciences, biology, transportation, and information technology [6]. Understanding the dynamics of these different types of networks and predicting the formation of new or missing connections, also known as "link prediction", is a well-known and well-studied problem in the field [17]. Link prediction aims to uncover hidden or potential interactions in a network; for example, to predict who might connect to whom in a social network

or which proteins are likely to interact in a biological network. Furthermore, link prediction is also used in various other applications and tasks, including recommender systems, anomaly detection, privacy control, network routing, and understanding the underlying mechanisms that govern network evolution [11, 14, 9, 10].

In literature, different types of methods for link prediction have been proposed. Initial methods focused on node pair similarities, such as the Jaccard coefficient, Adamic-Adar index, and resource-allocation index [17, 1]. Node pair similarity relies on the notion that if a given pair of nodes has a similarity score higher than some threshold, then this pair is more likely to be connected [11]. Later, researchers proposed other types of methods, including (i) maximum likelihood-based methods that work on maximizing the likelihood of the observed structure so that any missing link can be calculated using the identified rules and parameter [22], (ii) probabilistic models based methods that focus on modeling the underlying network structure and then use the learned model to predict the missing links [24], (iii) machine learning-based methods that train a machine learning model based on node pair features for existing and non-existing links [2, 5], and (iv) network embedding-based methods that create a low dimensional representation of the network using word2vec models or matrix-factorization, and then train a machine learning model using these vector representation of nodes to predict missing links [8, 23, 16]. In literature, it has been shown that the third category, machine learning based methods, outperforms other types of methods and has lately become the focus of link prediction research [12]. An additional advantage is that the use of topological features of the node pairs ensures the interpretability and explainability of resulting models through the analysis of feature importance. However, one limitation of these methods is that a link prediction model must be trained for each new network dataset.

To solve this problem, people have used transfer learning, i.e., a machine learning technique where a model developed for a particular task is reused or adapted as the starting point for a model on a second task [20]. Instead of training a new model from scratch, transfer learning leverages the knowledge gained from solving one problem and applies it to a different but related problem. By using a pretrained model as a starting point, one can save time and resources compared to training a new model from the ground up. In this work, we investigate the feasibility of transfer learning for link prediction in real-world complex networks.

In the remainder of this work, we analyze the characteristics and topology of 49 networks to understand how they affect the ability to train and predict links across networks. Specifically, we first propose a framework to perform cross-validation across multiple datasets, to efficiently test and compare the transfer learning performance of pretrained models for link prediction. Working towards automated pretrained model selection, we subsequently investigate what kind of topological network properties are important for selecting a well-performing pretrained model. Finally, we analyze what topological network similarities between training and testing networks, yield good transfer learning performance. In do-

ing so, we aim to understand to what extent transfer learning can be applied to predict unseen links in real-world networks by employing pretrained models.

The structure of the remainder of this paper is as follows. In Section 2, we discuss the approach followed to train our link prediction model, as well as the framework to test transfer learning. Then, Section 3 describes the data, evaluation criteria used, and the experimental setup developed, as well as the experimental results. Finally, we draw conclusions and propose future directions of research in Section 4.

2 Methodology

In this section, we first discuss the network features used to train predictive models for link prediction. Then, we give an overview of machine learning algorithms used to predict missing links and explain how we split the datasets for training and testing.

2.1 Features

Working towards a machine learning model that takes node pairs as input, and outputs whether this node pair is likely to be connected in the future, features that describe these node pairs are required.

In this work, we employ features commonly used in link prediction models, focusing on the work presented by Bors [3], to design a good link prediction model and test transfer learning. The chosen features balance simplicity, speed, and performance. We note that this study aims not to design the best link prediction model with the most comprehensive set of features, but instead aims to assess the feasibility of transfer learning in link prediction.

The selected set of features used throughout our experiments are as follows: (i) total neighbors, i.e., the union of all neighbors of the source and target nodes; (ii) common neighbors, i.e., the number of nodes connected to both the source and target nodes; (iii) Jaccard Coefficient [17], i.e., the ratio between the common and total neighbors; (iv) Adamic-Adar [1], which used to compute the closeness of nodes based on their shared neighbors; (v) preferential attachment [17, 19], i.e., the multiplication of the number of neighbors of the source and target nodes; (vi) degree of the source node (vii) degree of target nodes, (viii) ratio of degrees of source and target node, (ix) triangle count for the source node, and (x) triangle count for the target nodes, i.e., denoting the number of triangles they are involved in.

2.2 Training and testing set generation

To generate a training dataset from the network, the node pairs with existing links are considered as positive cases, and node pairs with distance two are used as negative cases. We consider only node pairs at a distance of at most two, as links are more likely to be formed between already close nodes. This is reflected

in the chosen features, many of which are not applicable to nodes at a distance larger than two. Moreover, it assists in reducing the class imbalance.

2.3 Stacked classifier

In this work, we use a supervised machine learning model for link prediction classification. Based on the findings by Ghasemian et al. [7], we set up a stacked classifier using *Scikit-Learn* [21], in which we use the most commonly used classifier models in the literature [23, 7, 18, 15]: a random forest classifier, logistic regression, naive-bayes, and quadratic discriminant analysis models. Together, these models serve as base estimators. We then use Logistic Regression as meta-model to combine ("stack") these predictions to make a final prediction.

2.4 Cross-validation framework

Here we discuss the proposed framework for adequate cross-validation training. We split the datasets into training and testing subsets to evaluate the performance of the prediction model. The datasets are split using k-fold, and in this work, we set $k = 4$, resulting in a random 75–25% split, to avoid bias in the sample. Additionally, due to the sparsity of networks, we down-sample the majority class of the training dataset to account for class imbalance. We perform cross-validation training as shown in Figure 1 to validate a model's results on different data portions. Essentially, we train models for each fold per dataset and then validate them on each split of the datasets. Note that in Figure 1, we only train and validate on the same split when testing on the same network. This is because, within the same split, the data is disjoint, so we do not have the same observations in the train and validation sets. Otherwise, we would encounter the same data in both the training and validation sets, which is not ideal for a machine learning model since it adds bias to the model by predicting previously seen data. Thus, we removed those cases from our testing set.

3 Experiments

This section covers the experimental setup and results. First, in Section 3.1, we discuss the datasets and metrics used. Then, in Section 3.2, we determine the overall feasibility of using transfer learning for link prediction by studying the AUC (loss) matrix and distributions. Next, in Section 3.3, we discuss what the most important topological features are that affect the performance of a pretrained model. Finally, in Section 3.4, we determine which structural network similarities yield good transfer learning performance.

3.1 Experimental Setup

Datasets In order to test the capabilities of transfer learning, we analyze 49 network datasets from the KONECT Project [13]. The networks were chosen

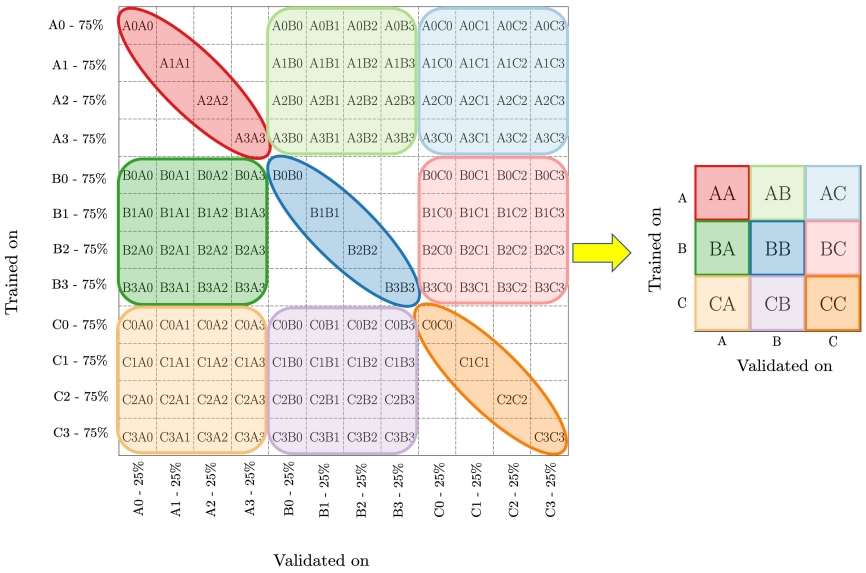


Fig. 1: Cross-validation training assignments. For each split, we generate a training set with 75% of the data and a validation set with the remaining 25%. (Letters indicate the dataset, numbers indicate the split.)

such that they cover a variety of topological properties, sizes, and categories. The datasets are presented in Table 1, along with their respective number of nodes and edges. All networks are interpreted as undirected and unweighted. Additionally, we only consider the largest connected component of each network.

Metrics Similarly to previous works [7, 4], we measure the performance of the classifier using the Area Under the Receiver Operating Characteristic Curve (AUC). Additionally, we define AUC loss, which we use to measure the loss in accuracy resulting from applying transfer learning between two different networks. We calculate loss $\mathcal{L}_{i,j}$ as follows:

$$\mathcal{L}_{i,j} = AUC_{i,i} - AUC_{i,j}$$

Here, $\mathcal{L}_{i,j}$ is the loss of training a model with network i and validating on network j and $AUC_{i,j}$ is the performance score of training a model with network i and validating on network j.

For all pairs of network datasets considered, the resulting AUC score of using a model trained on network i to predict missing links on network j, can be presented in a matrix as shown in Figure 1. In other words, for each pair of networks, we train on each fold of network i and test each trained model on each fold of network j (provided $i \neq j$), and then aggregate over all combinations of folds by averaging the AUC scores.

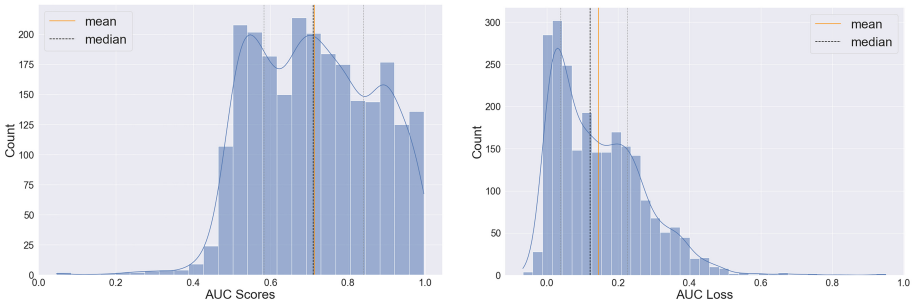
Table 1: Datasets sourced from [13], along with the number of nodes and edges.

dolphins (62/159)	residence (217/2,672)	copperfield (112/425)
cora (23,166/91,500)	karate (34/78)	proteins (1,706/6,207)
dblp (12,590/49,759)	adolescent (2,539/12,969)	reactome (6,327/147,547)
hepph (34,546/421,578)	blogs (1,224/19,025)	yeast (1,870/2,277)
hepth (27,770/352,807)	foldoc (13,356/125,207)	asoif (796/32,629)
astroph (18,771/198,050)	airtraffic (1,226/2,615)	sistercities (14,274/20,573)
astrophysics (16,046/121,251)	newyork (264,346/730,100)	lesmis (77/254)
erdos (6,927/11,850)	openflights (3,425/67,633)	pgp (10,680/24,316)
networkscience (1,461/2,742)	contiguous (49/107)	wikipedia (7,118/103,675)
digg (30,398/87,627)	euroroad (1,174/1,417)	hamsters (2,426/16,631)
dnc (2,029/39,264)	chess (7,301/65,053)	twitter (23,370/33,101)
facebook (46,952/876,993)	football (115/613)	filmtrust (874/1,853)
slashdot (51,083/140,778)	congress (219/764)	florida_dry (128/2,137)
uc_irvine (1,899/59,835)	bible (1,773/16,401)	florida_wet (128/2,106)
caida (26,475/53,381)	eat (23,132/511,764)	littlerocklake (183/2,494)
gnutella25 (22,687/54,705)	wordnet (146,005/656,999)	chesapeake (39/170)
routeviews (6,474/13,895)		

3.2 Feasibility of transfer learning in link prediction

To test whether transfer learning is feasible for link prediction, we assess if it is possible to pretrain a model on one network and test it on another with minimal AUC loss. Therefore, we applied the cross-validation training procedure detailed in Section 2.4 to all 49 datasets. The distribution of the resulting AUC scores and AUC loss are shown in Figure 2. Since we use the same model and the same features for all networks for link prediction, the AUC scores are, as expected, not particularly high, with an average AUC score of 0.71. More importantly, Figure 2b shows that AUC loss is very low for many combinations of training and testing networks. However, in many use cases, the average AUC loss of 0.14 can still be considered too significant. This signals the important conclusion that one can not simply choose a random network to pretrain and apply it to any new network. Instead, an appropriate network should be selected to minimize the AUC loss.

To investigate which (types of) networks make transfer learning in link prediction more feasible, Figure 3 shows the matrix of AUC loss for all pairs of training and testing networks for all 49 networks under consideration. Rows in the matrix depict the training performance of a single network, while columns represent the ease of prediction for a single network. In Figure 3, we can see that networks from citation and co-authorship categories (i.e., `cora`, `dblp`, `hpph`, `hepth`, `astroph` and `astrophysics`), as well as miscellaneous (`asoif`, `sistercities`, `lesmis`), show favourable training performance with minimal AUC loss, suggesting they are good baseline for pretrained models. Similarly, computer networks (`caida`, `gnutella25`, and `routeviews`), infrastructure networks (`airtraffic`, `newyork`, `euroroad`) and metabolic networks (`proteins`, `yeast`) display good



(a) AUC scores distribution.

(b) AUC losses distribution.

Fig. 2: Distributions of the AUC scores and AUC loss across all networks. Alongside the mean and median the quartiles are depicted with light grey lines. The blue line approximates the trend.

validation performance, meaning they are usually easy to predict regardless of the choice of pretrained model. On the other hand, the bible network is the worst performing training network, with several very large AUC losses. Furthermore, some human contact (`residence`, `karate`), human social (`adolescent`), lexical networks (`bible`, `eat`, `wordnet`) and trophic networks, i.e., relating to biological interactions of species (commonly food chains), (`florida_dry`, `florida_wet`, `littlerocklake`, `chesapeake`) show substandard validation performance, with few to no pretrained models providing low AUC loss. As such, there are some networks that often do well for training (pretrained) models and those for whom many pretrained models work well, but there are also some for which no pretrained model appears to perform well. Thus, although transfer learning for link prediction is feasible for most networks given the right choice of pretrained model, it is not effective in all cases, i.e., there is no one-size-fits-all kind of solution.

3.3 Topological feature importance for pretraining models

Next, we set to understand how a network's topological features might affect a model's learning performance. For this, we train decision tree and random forest algorithms by using the topological features of the networks to fit their average AUC score as pretrained models aggregated over all testing networks. The top splits, i.e., the top discriminating decisions, of the resulting trees are visualized in Figure 4. By studying the top discriminating decisions of these trees, we can understand which are the most important topological features for good transfer learning performance and how these features affect the performance.

The decision tree depicted in Figure 4a suggests that a high normalized number of triangles (#triangles/edges) results in, on average, higher AUC scores. Furthermore, for pretrained models from networks with a high maximum degree and few triangles (per link), we observe that the transitivity is a great indicator

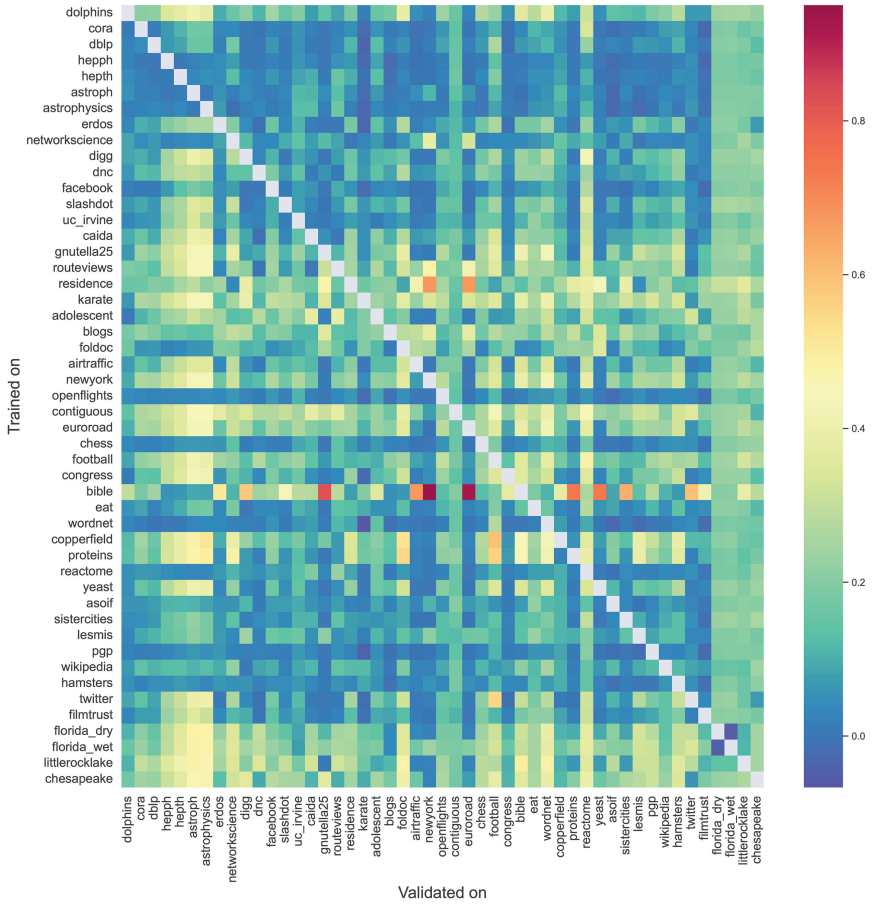
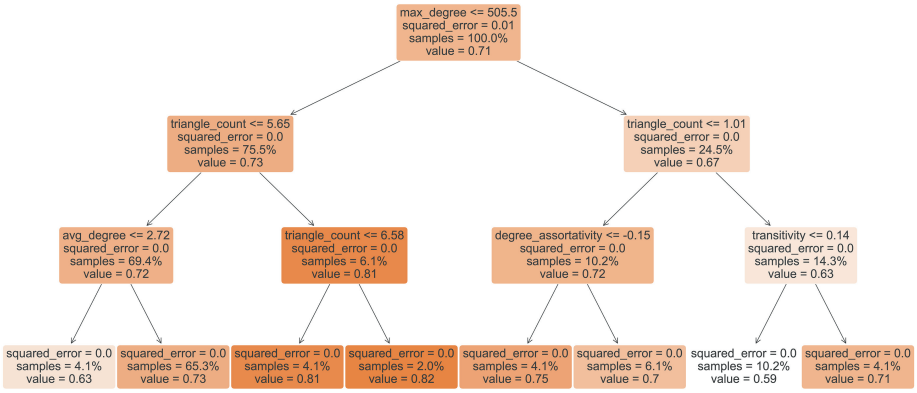


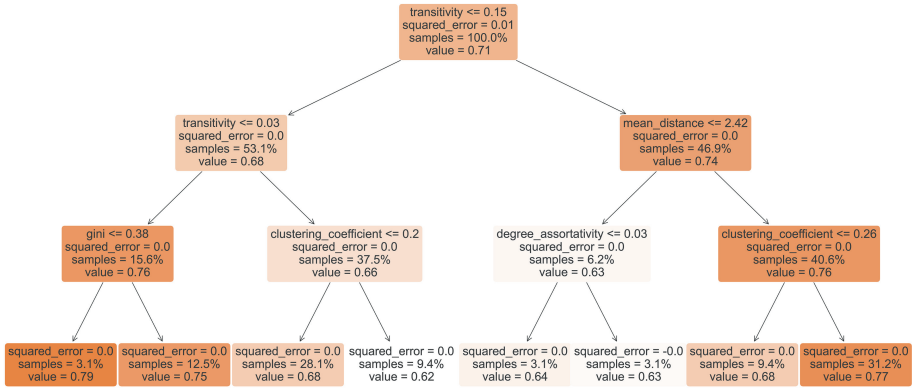
Fig. 3: AUC loss matrix for all 49 datasets.

of either high or low resultant AUC scores, whereas, for lower maximum degree models and few triangles (per link), the average degree can be a good indicator.

The decision tree obtained from the random forest algorithm in Figure 4b suggests that very low transitivity or relatively high transitivity with high mean distance and clustering coefficients tend to result in higher AUC scores. On the contrary, low-to-middle transitivity with higher clustering coefficients and high transitivity with small mean distances or small clustering coefficients result in low AUC scores. As such, only high transitivity or clustering coefficient are not universally good or bad for the transfer learning link prediction performance of a pretrained model. However, note that clustering coefficient and transitivity are usually correlated, so a network with a high clustering coefficient will likely not have low transitivity. Interestingly, our previous observation from Figure 4b indicates that high AUC scores are obtained when these topological features are



(a) Decision tree.



(b) Random forest.

Fig. 4: Tree-based topological feature importance. Darker coloured nodes/leafs indicate higher average AUC scores achieved by the pretrained models of included networks, whereas lighter coloured leafs indicate lower average AUC scores. The average AUC scores of the node/leaf are indicated by 'value'. For each parent node the left child includes the pretrained models from networks that adhere to the condition specified by the parent node and the right child includes those that do not. For example, in figure (a) the left child of the root node includes all pretrained models from networks with a maximum degree ≤ 505.5 while the right child includes all those with a maximum degree > 505.5.

indeed correlated for a network, and low AUC scores are obtained when they are not.

In short, we find that some of the most important topological features influencing the performance of pretrained models are the number of triangles (per link), the transitivity, and the clustering coefficient. Notably, the level of correlation between these features can be especially indicative of the resulting high or low AUC scores of a pretrained model.

3.4 Influence of network dissimilarity on transfer learning

Finally, we examine what structural network similarities between a training and a testing network lead to good transfer learning performance. We do so by comparing the AUC loss (as defined in Section 3.1) to how dissimilar the topological properties are between two networks. This allows us to understand if there is a relationship between their similarity and the performance of the model. Figure 5 illustrates the relation of loss in performance when predicting missing links in one network using a model trained on another, compared to the topological dissimilarities of both networks. It is clear that there is a trend in almost all topological features (except for maximum degree), and if two networks are more different, there is more loss in the performance. Specifically, we observe that if two networks are more similar in terms of clustering coefficient, it leads to the lowest AUC loss, while the most dissimilar networks for this feature have the second highest AUC loss. Furthermore, when it concerns degree assortativity, gini coefficient, and transitivity, we note that highly similar networks also show reasonably low AUC losses. Overall, our results indicate that considering the similarity in terms of clustering coefficient, and to a lesser extent in terms of degree assortativity, gini coefficient, and transitivity is especially important in choosing a pretrained model for link prediction.

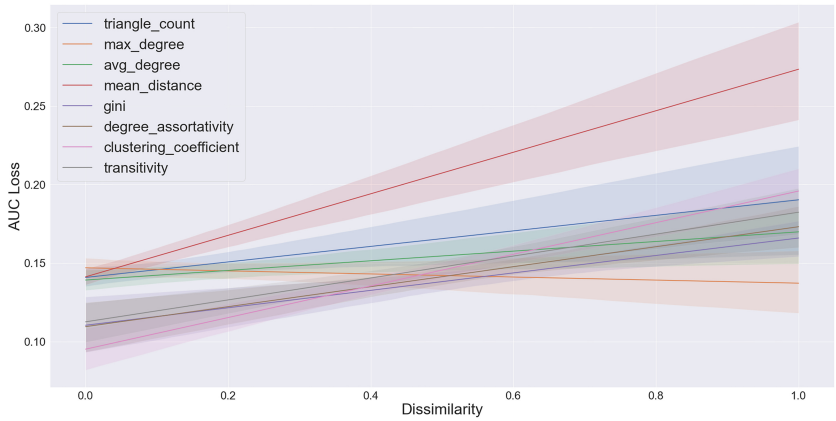


Fig. 5: AUC loss vs. network dissimilarity.

4 Conclusion

In this work, we studied the feasibility of using pretrained link prediction models in complex networks. Moreover, we studied the network characteristics that impact model training, and how these can be used for selecting a well-performing pretrained model. We conducted experimental analysis on a large corpus of structurally diverse networks, including co-authorship, citation friendship, human interaction, biological, and transportation networks. Through our experiments, we observed that transfer learning for link prediction is a feasible way to move forward, and some network categories perform better as sources for training and others to predict missing links on. Furthermore, we found that network features based on local connectivity, such as clustering coefficient, number of triangles, or transitivity, are important indicators when picking a network for training a predictive model. Specifically, we found that when two networks show very dissimilar topologies in terms of clustering coefficient, but also in terms of degree assortativity, gini coefficient, and transitivity, it is likely that the performance of transfer learning is hindered.

This work demonstrates the feasibility of using pretrained models in link prediction. Future work could focus on designing better transfer learning methods to achieve higher accuracy using topological properties of an unseen network and the network used for pre-training. Additionally, this work opens an avenue to use transfer learning for complex network problems, such as node classification, role identification, and influence maximization.

References

1. Adamic, L.A., Adar, E.: Friends and neighbors on the web. Social networks **25**(3), 211–230 (2003)
2. Al Hasan, M., Chaoji, V., Salem, S., Zaki, M.: Link prediction using supervised learning. In: SDM06: workshop on link analysis, counter-terrorism and security, vol. 30, pp. 798–805 (2006)
3. Bors, P.P.: Topology-aware network feature selection in link prediction (2022)
4. de Bruin, G.J., Veenman, C.J., van den Herik, H.J., Takes, F.W.: Supervised temporal link prediction in large-scale real-world networks. Social Network Analysis and Mining **11**(1), 1–16 (2021)
5. van Engelen, J.E., Boekhout, H.D., Takes, F.W.: Explainable and efficient link prediction in real-world network data. In: Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings 15, pp. 295–307. Springer (2016)
6. Estrada, E.: The structure of complex networks: theory and applications. Oxford University Press, USA (2012)
7. Ghasemian, A., Hosseinmardi, H., Galstyan, A., Airoldi, E.M., Clauset, A.: Stacking models for nearly optimal link prediction in complex networks. Proceedings of the National Academy of Sciences **117**(38), 23,393–23,400 (2020)
8. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 855–864 (2016)

9. Guimerà, R., Sales-Pardo, M.: Missing and spurious interactions and the reconstruction of complex networks. Proceedings of the National Academy of Sciences **106**(52), 22,073–22,078 (2009)
10. Huang, Z., Zeng, D.D.: A link prediction approach to anomalous email detection. In: 2006 IEEE international conference on systems, man and cybernetics, vol. 2, pp. 1131–1136. IEEE (2006)
11. Kumar, A., Singh, S.S., Singh, K., Biswas, B.: Link prediction techniques, applications, and performance: A survey. Physica A: Statistical Mechanics and its Applications **553**, 124,289 (2020)
12. Kumari, A., Behera, R.K., Sahoo, K.S., Nayyar, A., Kumar Luhach, A., Prakash Sahoo, S.: Supervised link prediction using structured-based feature extraction in social network. Concurrency and Computation: practice and Experience **34**(13), e5839 (2022)
13. Kunegis, J., Staab, S., Dünker, D.: KONECT – The Koblenz Network Collection. In: Proc. Int. Sch. and Conf. on Netw. Sci. (2012)
14. Li, J., Zhang, L., Meng, F., Li, F.: Recommendation algorithm based on link prediction and domain knowledge in retail transactions. Procedia Computer Science **31**, 875–881 (2014)
15. Li, Y., Liu, X., Wang, C.: Research on link prediction under the structural features of attention stream network. In: 2021 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), pp. 148–154. IEEE (2021)
16. Li, Y., Wang, Y., Zhang, T., Zhang, J., Chang, Y.: Learning network embedding with community structural information. In: IJCAI, pp. 2937–2943 (2019)
17. Liben-Nowell, D., Kleinberg, J.: The link prediction problem for social networks. In: Proceedings of the 12th International Conference on Information and knowledge management, pp. 556–559 (2003)
18. Liu, Z., Zhang, Q.M., Lü, L., Zhou, T.: Link prediction in complex networks: A local naïve bayes model. Europhysics Letters **96**(4), 48,007 (2011)
19. Newman, M.E.: Clustering and preferential attachment in growing networks. Physical review E **64**(2), 025,102 (2001)
20. Niu, S., Liu, Y., Wang, J., Song, H.: A decade survey of transfer learning (2010–2020). IEEE Transactions on Artificial Intelligence **1**(2), 151–166 (2020)
21. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)
22. Redner, S.: Teasing out the missing links. Nature **453**(7191), 47–48 (2008)
23. Saxena, A., Fletcher, G., Pechenizkiy, M.: Nodesim: node similarity based network embedding for diverse link prediction. EPJ Data Science **11**(1), 24 (2022)
24. Wang, C., Satuluri, V., Parthasarathy, S.: Local probabilistic models for link prediction. In: Seventh IEEE international conference on data mining (ICDM 2007), pp. 322–331. IEEE (2007)