Explainable and Efficient Link Prediction in Real-World Network Data

Jesper E. van Engelen[1], Hanjo D. Boekhout[1], and Frank W. Takes[1,2]

[1] Leiden University (LIACS), The Netherlands
[2] University of Amsterdam (AISSR), The Netherlands
{jvengele,hdboekho,ftakes}@liacs.nl

Abstract. Data that involves some sort of relationship or interaction can be represented, modelled and analyzed using the notion of a network. To understand the dynamics of networks, the *link prediction* problem is concerned with predicting the evolution of the topology of a network over time. Previous work in this direction has largely focussed on finding an extensive set of features capable of predicting the formation of a link, often within some domain-specific context. This sometimes results in a "black box" type of approach in which it is unclear how the (often computationally expensive) features contribute to the accuracy of the final predictor. This paper counters these problems by categorising the large set of proposed link prediction features based on their topological scope, and showing that the contribution of particular categories of features can actually be explained by simple structural properties of the network. An approach called the *Efficient Feature Set* is presented that uses a limited but explainable set of computationally efficient features that within each scope captures the essential network properties. Its performance is experimentally verified using a large number of diverse real-world network datasets. The result is a generic approach suitable for consistently predicting links with high accuracy.

1 Introduction

Many real-world phenomena, structures and interactions can be described by networks. Examples include links on the web, social interactions through online media, co-authorship of scientific papers, connectedness of physical devices and the spread of diseases. The field of *network science* [15] (or social network analysis [24]) is concerned with mining patterns and structures in these networks. Dynamic networks uncover a new property of networks to study: the process of their *evolution*. Is it possible to predict which links will form between pages on the web, to estimate the probability of two researchers co-authoring a paper in the future, to predict whether a friendship between users in a social network will at some point be formed; or to say which links are missing on Wikipedia? In short: can we predict how a network will evolve in the future? In this paper, we specifically consider the *link prediction* problem: based on the topology of a network at a certain time, we want to predict exactly which links will form in the

future. We wish to do so for *real-world networks* representing actual interactions, relations or communication within a real system or environment [25]. Although the underlying data of these networks is diverse, it is well-known that they have a common structure: there is a power law degree distribution (they are scale-free), there is a substantially higher than random number of closed triangles (measured by the clustering coefficient) and the average distance between two nodes is very low, typically between four and eight. This is altogether frequently referred to as the small-world property of real-world networks [13]. To ensure the generalisability of a link prediction technique across real-world networks, we consider generic methods that can be defined for any type of real-world network and are thus only based on the *structure* (*topology*) of the network.

In previous work, a number of topological measures have been proposed for predicting whether links will form between two nodes in a network. We refer to these measures as *features*: properties of the network that can be used to predict whether a link will form. Link prediction then becomes a *supervised learning task*: develop a classifier, trained on features derived from the current network, able to predict which links will form in the future. The goal is not to compare different supervised learning algorithms. Instead, we want to provide a suitable alternative for the "black box" type of approach resulting from simply reusing the large number of (often computationally expensive) features proposed in previous work [2,7,16,17,18,19,21,23]. Therefore, we propose a categorization of existing and new link prediction features into sets of features operating at distinct topological scopes of the network, followed by an experimental assessment of their performance. As a result, we are able to choose a much smaller subset of useful features: the *Efficient Feature Set*. It provides a number of advantages in terms of explainability, efficiency, consistency and general applicability.

The remainder of this paper is structured as follows. We explain how we approach the link prediction problem in Section 2, followed by outlining previous work in Section 3. We elaborate on link prediction features in Section 4, after which we introduce the proposed *Efficient Feature Set* in Section 5. In Section 6 we conduct experiments on real-world dynamic network data to test our method. Conclusions and suggestions for future work are given in Section 7.

2 Problem statement

In an undirected, unweighted network or graph $G = (V, E)$, we have a set of nodes V and a set of links E, where $\{u, v\} \in E$ denotes whether a link exists between nodes $u, v \in V$. In the case of a directed network, the nodes of a link are ordered pairs and $(u, v) \in E$ denotes that there is a directed link from node u to v, but not necessarily that there is a link from v to u. For brevity, we henceforth refer to the number of nodes $|V|$ as n and the number of edges $|E|$ as m. For undirected networks, m denotes the number of undirected edges. In weighted networks, each link is assigned a weight. We denote this weight as w_{uv} for a link $(u, v) \in E$. In undirected networks, one weight is assigned to each undirected link such that $w_{uv} = w_{vu}$ for all links $\{u, v\} \in E$.

Link prediction problem: given a network $G_\tau = (V_\tau, E_\tau)$, denoting the network at time τ, predict the newly formed links in the evolved network $G_{\tau'} = (V_{\tau'}, E_{\tau'})$ at time $\tau' > \tau$, i.e., predict the contents of $E_{\tau'} \backslash E_\tau$.
Depending on the considered network, directed or undirected links are predicted. Note that we always consider predicting whether a link will form, and not what weight it has when it is formed. To do link prediction as defined above, we must have at least two observations of the same network at different points in time. The link prediction problem can then be seen as a supervised binary classification task, where, given a set of features $\boldsymbol{x}$, the value of y has to be predicted. Vector $\boldsymbol{x}$ is the value of particular features we deem suitable for predicting whether a link (u, v) will be formed at time τ' (with $\tau' > \tau$) and y is a binary value indicating whether a link (u, v) has formed ($y = 1$) at τ' or not ($y = 0$). We can then use metrics common in data mining to assess the quality of some classifier and thus the accuracy of the performed link prediction task. Ultimately, we aim for a classifier which satisfies the following properties:

1. *Explainable* in its performance based on simple topological network features.
2. *Efficient* in terms of computational complexity (and thus running time).
3. *Accurate* in providing correct link predictions.
4. *Consistent* in its accuracy relative to larger feature sets.
5. *General*, yielding reliable results across different real-world networks.

The five requirements above will be used to validate the performance of different link prediction feature sets in Section 6.

3 Related work

The roots of link prediction lie in the context of information retrieval [3], focussing on the prediction or retrieval of missing data elements rather than future network links. The term "link prediction" [18] was given to describe the problem of predicting the formation of links and thus the evolution of a network. Early algorithms focused on single measures, such as the work of Sarukkai using Markov chains [23]. Later, it was tackled as a supervised learning task, for example using linear regression, including the work by Popescul et al. [21] and O'Madadhain et al. [19]. The seminal paper of Liben-Nowell and Kleinman [16] introduced a large set of features to combine using supervised learning. In fact, many features used in link prediction stem from other subfields of data mining, including the popular Katz measure [12] and Adamic/Adar measure [1], both discussed later.

To exploit the large number of features nowadays available, several frameworks and algorithms have been devised, using existing classifiers from popular data mining tools [9] combined with domain-independent feature sets. Lichtenwalter et al. propose a framework in [17] and show that in supervised contexts, existing methods are less accurate than their proposed PropFlow measure. Several frameworks specifically focus on easy-to-compute features, see for example [2] or the excellent extensive overview of further related work provided in [7].

This paper attempts to distinguish itself from previous work by aiming to satisfy each (rather than a subset) of the five goals explained in Section 2.

4 Link prediction methods and approaches

The link prediction methods proposed in literature (e.g. [7,16]) and throughout this section are formulated as topological *features* of the network. We distinguish between *node features*, *neighbourhood features*, and *path features*. Each feature assigns a score $S(u, v)$ to a candidate pair (u, v), which can be used by a classifier to determine the probability of a link forming between source u and target v.

For convenience in the following definitions, for directed networks, we define $\Gamma(v) = \{u \in V : (u, v) \in E \vee (v, u) \in E\}$ as the neighbourhood of v. We can then also define the out-neighbourhood and the in-neighbourhood of v as $\Gamma_{\text{out}}(v) = \{w \in V : (v, w) \in E\}$ and $\Gamma_{\text{in}}(v) = \{u \in V : (u, v) \in E\}$, respectively. For undirected networks, we define $\Gamma(v) = \{v \in V : \{u, v\} \in E\}$. The *degree* $d(v)$ of a node is simply the size of its neighborhood: $d(v) = |\Gamma(v)|$. The in-degree and out-degree for a directed network are then respectively $d_{\text{in}}(v) = |\Gamma_{\text{in}}(v)|$ and $d_{\text{out}}(v) = |\Gamma_{\text{out}}(v)|$. Below, to improve readability, where applicable, we choose not to formally define the trivial extension of each measure to the equivalent in- and out-measures for directed networks.

4.1 Node features

Node features are derived from the properties of a node and its links, only considering the node currently under evaluation.

- *Degree*: The node degree feature simply uses $d(u)$ and $d(v)$, i.e., the degree of source node u and target node v.
- *Volume*: The node volume measures the total weight of all incoming or outgoing links (or both) of both the source and the target node. For source u it is defined as $\sum_{t \in \Gamma(u)} w_{ut}$. The target node volume is defined analogously.

4.2 Neighbourhood features

Neighbourhood features also consider patterns and relations of the direct neighbours of the source and target node.

- *Total neighbours*: This measure counts the total number of distinct neighbours that exist for the candidate pair and is defined as $|\Gamma(u) \cup \Gamma(v)|$.
- *Common neighbours*: Here we compute the number of neighbours two nodes have in common. Formally, we write $|\Gamma(u) \cap \Gamma(v)|$.
- *Transitive common neighbourhood*: This is a variation on the common neighbours measure intended for use in directed networks. It determines the number of neighbours to which u has a link and that have a link to v: $|\Gamma_{\text{out}}(u) \cap \Gamma_{\text{in}}(v)|$.
- *Jaccard coefficient*: This coefficient proposed in [10] considers the number of common neighbours two nodes have, relative to the total number of distinct neighbours they have. Formally, we write $|\Gamma(u) \cap \Gamma(v)| / |\Gamma(u) \cup \Gamma(v)|$.

- *Transitive Jaccard coefficient*: To capture link direction in directed networks, we propose to combine the concept of the transitive common neighbourhood with the Jaccard coefficient: $|\Gamma_{\text{out}}(u) \cap \Gamma_{\text{in}}(v)|/|\Gamma_{\text{out}}(u) \cup \Gamma_{\text{in}}(v)|$.
- *Adamic/Adar*: This measure, introduced by Adamic and Adar in [1], considers properties shared by two nodes, favouring properties that not many other nodes have. In a network, such a property can be the set of out-neighbours of u and v. Below, the number of in-neighbours of w signals the same feature, so we sum these ratios for u and v, obtaining: $\sum_{w \in (\Gamma_{\text{out}}(u) \cap \Gamma_{\text{out}}(v))} \frac{1}{\log(\Gamma_{\text{in}}(w))}$.
- *Preferential attachment*: This concept is used in the well-known Barabási-Albert graph generation model [4] to model the creation of a network. It is based on the observation that nodes that already have a high degree are more likely to attract new links than nodes with a lower degree. We use this property in link prediction by computing the product of degrees of the source and target node. So for the out-degree, it is defined as $|\Gamma_{\text{out}}(u)| * |\Gamma_{\text{out}}(v)|$.
- *Opposite direction link*: We consider a feature introduced in [7] for directed networks, where the probability of a link (u, v) is assumed to be dependent on the existence of (v, u), captured in a binary feature value of 0 or 1.

4.3 Path features

The entire topology of the network is considered in path features: not only direct neighbours, but also nodes further away are considered in the evaluation.

- *Shortest path length*: This measure indicates the length of a shortest path from the source node to the target node, i.e., the minimal number of edges that have to be traversed to reach node v starting in node u. This measure is commonly referred to as the *distance*, denoted $d(u, v)$.
- *Number of shortest paths*: This metric proposed in [17] ranks candidate links based on how many shortest paths of length $d(u, v)$ exist from u to v. Parameter ℓ_{MAX} defines the maximum distance $d(u, v)$ that is considered. We denote this measure by $\text{paths}_{u,v}^{(\ell_{\text{MAX}})}$.
- *Restricted Katz measure*: Introduced by Katz [12], it awards importance to the number of paths between two nodes as a predictor of the likelihood of a link, but exponentially decreases the importance as the path length grows, determined by the parameter $\beta < 1$; typical values of β are around 0.05 [12]. In accordance with [17] we again use ℓ_{MAX}, giving: $\sum_{\ell=0}^{\ell_{\text{MAX}}} \beta^{\ell} |\text{paths}_{u,v}^{(\ell)}|$.
- *PropFlow*: The last measure we utilise is PropFlow, introduced by Lichtenwalter et al. in [17], relating the probability of a link forming between nodes u and v to the probability that a random walk starting in node u ends up in node v, considering all walks from u of at most length ℓ_{MAX}. In weighted networks, it assigns the probabilities for following each link proportional to their weight. Starting at u, the score update rule for each neighbouring node t in an iteration is $S(u, t) = S(u, t) + S(u, v) * (w_{ut})/(\sum_{x \in \Gamma(u)} w_{ux})$.

5 Efficient Feature Set

In this section, we explain our approach to selecting features for link prediction, taking into account the first two quality goals posed in Section 2. As stated before, one of the problems with gathering a vast number of features and simply using all of them, is that it does not lead to an *explainable* method, as the classifier considers so many features that it becomes impossible to distinguish between well and poor performing features, and that it is hard if not impossible to understand exactly how different features use the topological structure of the network to predict future links. We solve the abovementioned problem by grouping our features based on their *topological scope*, explicitly distinguishing between:

- *Individual properties* (evaluating properties of the source and target node).
- *Local properties* (considering similarities and differences between the neighbourhoods of the source and target node).
- *Global properties* (considering paths between the source and target node).

These notions correspond to the different levels at which small world networks are typically studied: the micro level, meso level and macro level [24]. More importantly, they coincide with the grouping of features used in Section 4, namely node features, neighbourhood features and path features. We present the categorized features in Table 1. The "Compl." column contains information on the computational complexity of computing the feature for a single candidate pair, expressed as a function of the number of nodes n and links m.

The second goal that we achieve with EFS deals with *efficiency*. We hypothesise that, when constructing a set of features for training a classifier, choosing features whose category is already sufficiently represented in the feature set will

Table 1: List of candidate pair features, categorized by topological scope.

Node features			
Feature	Variant	Compl.	EFS
Degree (source)	-	$O(1)$	✓
Degree (source)	d_{in}	$O(1)$	
Degree (source)	d_{out}	$O(1)$	
Degree (target)	-	$O(1)$	✓
Degree (target)	d_{in}	$O(1)$	
Degree (target)	d_{out}	$O(1)$	
Volume (source)	-	$O(m/n)$	
Volume (source)	d_{in}	$O(m/n)$	✓
Volume (source)	d_{out}	$O(m/n)$	✓
Volume (target)	-	$O(m/n)$	
Volume (target)	d_{in}	$O(m/n)$	✓
Volume (target)	d_{out}	$O(m/n)$	✓
Path features			
Feature	Param.	Compl.	EFS
Shortest path length	-	$O(m+n)$	✓
Num. shortest paths	$\ell_{\text{MAX}} = 3$	$O(m+n)$	
Restricted Katz	$\ell_{\text{MAX}} = 3$, $\beta = 0.05$	$O(m+n)$	
PropFlow	$\ell_{\text{MAX}} = 3$	$O(m+n)$	✓

Neighbourhood features			
Feature	Var.	Compl.	EFS
Total neighbours	-	$O(m/n)$	
Total neighbours	Γ_{in}	$O(m/n)$	
Total neighbours	Γ_{out}	$O(m/n)$	
Common neighbours	-	$O(m/n)$	✓
Common neighbours	Γ_{in}	$O(m/n)$	
Common neighbours	Γ_{out}	$O(m/n)$	
Trans. comm. neigh.	-	$O(m/n)$	
Jaccard Coeff.	-	$O(m/n)$	✓
Jaccard Coeff.	Γ_{in}	$O(m/n)$	
Jaccard Coeff.	Γ_{out}	$O(m/n)$	
Trans. Jacc. Coeff.	-	$O(m/n)$	✓
Adamic/Adar	-	$O(m/n)$	
Pref. attachment	-	$O(1)$	
Pref. attachment	Γ_{in}	$O(1)$	
Pref. attachment	Γ_{out}	$O(1)$	
Opp. direction link	-	$O(1)$	✓

only marginally increase prediction accuracy, whereas computation time may increase substantially. Based on this, we construct a subset of features in which we sufficiently represent each feature group. We propose the *Efficient Feature Set* (EFS), which is constructed by choosing a small subset of features that captures the widest variety of topological properties possible. To do so, we first split the full feature set along the first *variety* dimension: topological scope (as explained above). From each category, we then select the features that exhibit a high degree of diversity within the second dimension: the *balance* between undirectedness vs. directedness, weighted vs. unweighted and absolute vs. relative counts.

From the node features, we therefore include in EFS the degree and the in- and out-volume of both the source and the target node. For the neighbourhood features, we wish to capture both the absolute and the relative size of the joined neighbourhood. So, we include the transitive Jaccard coefficient to further support directed networks and the opposite direction link feature as a direct indicator of reciprocation. From the path features, we capture general propagation properties of the network by using the undirected shortest path length, and finally we capture properties of weighted and directed graphs by using PropFlow. In Table 1, the column "EFS" summarizes the selected features. In terms of efficiency, EFS attains an immediate speed-up of two or more in each category. We performed experiments to empirically verify the contribution and performance of the three proposed feature categories. For space and readability reasons, these results are presented in Section 6.3, followed by the results of using only EFS in Section 6.4.

6 Experiments and results

In this section, we discuss the considered network datasets in Section 6.1, after which we outline preprocessing steps followed by the encountered class imbalance problem. Next we explain the experimental setup in Section 6.2 before evaluating the results in Section 6.3 and Section 6.4.

6.1 Network datasets

An overview of the considered networks is given in Table 2, listing the network name, source, type and network properties such as the number of nodes, the

Table 2: Characteristics of network datasets used for testing.

dataset	Type	Nodes	Links	CC	Type	Dist	3Γ
digg [6]	news communication	30,398	86,404	0.01	+ D	4.68	45%
fb-links [26]	social friendship	63,731	817,035	0.22	- U	4.31	88%
fb-wall [26]	social communication	46,952	274,086	0.11	+ D	5.71	61%
infectious [11]	disease spread	410	2,765	0.46	+ U	3.57	83%
liacs	scientific collaboration	1,036	4,650	0.84	+ U	3.86	100%
lkml-reply [14]	email communication	27,927	242,976	0.30	+ D	5.19	99%
slashdot [8]	web communication	51,083	131,175	0.02	+ D	4.59	75%
topology [27]	network topology	34,761	107,720	0.29	+ U	3.78	97%
ucsocial [20]	social communication	1,899	20,296	0.11	+ D	3.07	99%
wikipedia [22]	information network	100,312	746, 114	0.21	- D	3.83	89%

number of links and the clustering coefficient (in the "CC" column). The "Type" column indicates the directedness (`D` for directed, `U` for undirected) and weighting of the network (`-` for unweighted, `+` for weighted). Finally, the average shortest path length is listed in the "Dist" column. The diverse real-world network datasets cover a range of different networks (see the sources listed in Table 2 for details). We generated the `liacs` weighted scientific collaboration network from raw data on co-authorship of researchers involved with the computer science institute LIACS in the period 2005–2014, a link denoting the joined publication count.

For each network, we choose τ (see Section 2) such that 95% of the links were formed before time τ. This is because we are predicting individual links as opposed to macroscopic evolution, and thus require a relatively developed state of the network for training.

The number of candidate pairs for which we could give a prediction is very large: with n nodes and m directed links at time τ, potentially $(n \cdot (n-1)) - m$ links could be formed at time τ'. The number of links that are actually formed between τ and τ' is in practice only a tiny fraction. Calculating features for all possible node pairs would be infeasible in terms of computation time. To address this, we look at column "3Γ" of Table 2, showing the percentage of nodes formed between nodes at distance 3 or less. In our networks, on average 84% of all newly formed links were at undirected distance 3 of one another, so we limit our predictions to candidate pairs at that distance. This results in the omittance of a substantial portion of the node pairs where no links are formed while retaining a large majority of all positive instances. As the considered networks are all small-world networks in which the average pairwise distance is very low, we further reduce the class imbalance by randomly removing negative instances as extensively discussed and suggested in [7].

From preliminary experiments we found that shifting the ratio between the number of negative and positive class instances did not significantly influence classification performance. From this range of acceptable balances, we have chosen a class balance of nine negative instances for each positive instance.

6.2 Experimental setup

To determine the accuracy, we want to capture the relation between true positive and false positive rates, so we use the well-known ROC-curves, plotting these two rates against each other. The area under the ROC-curve (AUROC) can be used to assess the quality of the predictor. As explained before, our work focuses on the comparison between different feature sets to obtain a robust classifier, and not on the comparison of supervised learning algorithms. Therefore we use random forests, which are repeatedly and consistently identified as well-performing general purpose supervised learning algorithms, see for example the discussion in [5]. We use the implementation of [9] with an ensemble of 50 random decision trees. Along similar lines, we abstract away from specific hardware and software, comparing the efficiency of different feature sets based on the computational complexity as listed in Table 1.

6.3 Results — Topological scopes

The ROC-curves of the predictions of the three feature sets discussed in Section 5 are depicted in Figure 1, and the corresponding AUROC values are listed in Table 3. The variance in the performance of the single feature class predictors underlines the degree to which the performance of these feature sets alone is significant, justifying the construction of EFS in Section 5. For instance, in Figure 1 we see that for the `wikipedia` dataset, the set of node features outperforms all other individual feature sets, whereas it is outperformed by all other feature sets in the `liacs` dataset. Trivially, we find that the full feature set is the best predictor in terms of accuracy and consistency, performing well across all datasets. We note that in one case (`ucsocial`), the set of node features outperformed the set of all features by 0.007, likely due to the randomness in the random forest classifier (an optimization step beyond the interest and scope of this paper). Below, we discuss the main results for each of the feature sets, relating their performance at the particular topological scope to the considered network datasets.

Node features. The node features appear to perform better than might be expected from such local metrics. An interesting observation is the exceptionally good performance on the networks modelling online conversations (`lkml-reply`, `ucsocial`, `slashdot` and `digg`). This might be because users who are active in replying to messages and receiving replies to their messages are likely to remain active in the future. As expected, the performance of node features appears negatively correlated with the mean distance in the graph: as this distance grows, the node features classifier performance decreases.

Neighbourhood features. In directed networks, we observe that a high rate of reciprocity gives good performance, which may be because in many real-world networks, non-reciprocated links tend to be reciprocated at some point in the future, captured by the *opposite direction link* feature. Indeed, in the `fb-wall`, `ucsocial` and `lkml-reply` networks, each having a reciprocity rate of around 65%, we observe that the neighbourhood features generate an AUROC of 99.2% of the AUROC of the full feature set against an overall average 97.2% for the neighbourhood features in directed networks. The `digg` network, having a reciprocity rate of just 2% yields the second to worst performance. The low performance of neighborhood features on the `infectious` network can be explained by its degree distribution which as opposed to all other networks, does not follow a power law, but is instead distributed around the average (see Figure 2).

Table 3: AUROC for each network and each set of features.

Features	*digg*	*fb-links*	*fb-wall*	*infectious*	*liacs*	*lkml*	*slashdot*	*topology*	*ucsocial*	*wikipedia*
Full	0.830	0.933	0.887	0.967	0.997	0.975	0.928	0.967	0.913	0.970
Node	**0.827**	0.700	0.710	**0.955**	0.969	0.971	**0.922**	0.949	0.911	**0.941**
Neighbourhood	0.761	**0.911**	**0.866**	0.794	**0.986**	**0.974**	0.920	**0.961**	**0.920**	0.926
Path	0.632	0.897	0.819	0.579	0.979	0.925	0.777	0.940	0.673	0.827
EFS	0.825	0.930	0.876	0.958	0.995	0.973	0.921	0.965	0.910	0.967
EFS Performance	**99.4%**	**99.6%**	**98.8%**	**99.1%**	**99.8%**	**99.8%**	**99.2%**	**99.8%**	**99.7%**	**99.7%**

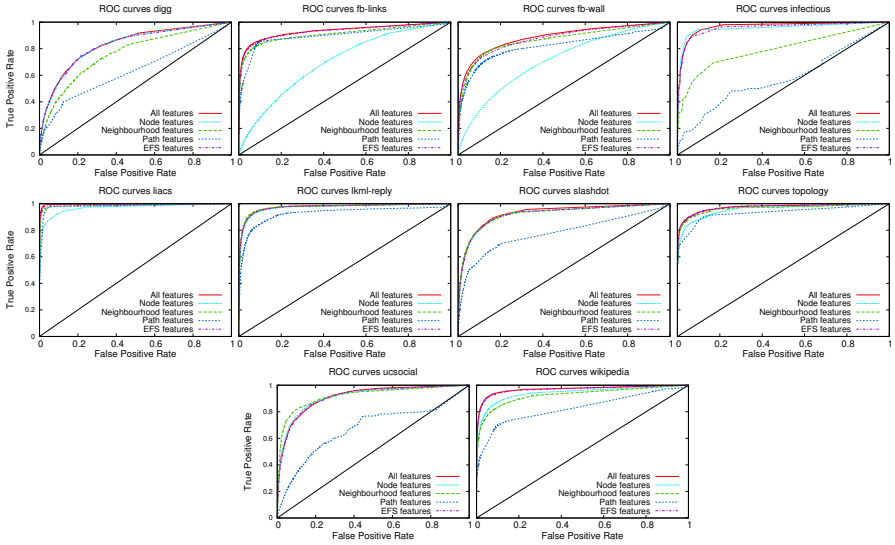

Fig. 1: ROC curves for the network datasets in Table 2, for each feature set.

Path features. By themselves, the path features are likely too global to sufficiently capture local network patterns. Nevertheless, they do add value to the full feature set as well as EFS. Most notably, the performance of path features increases as the mean path length grows, highlighting the importance of this feature set in less dense networks.

6.4 Results — EFS

We evaluate our Efficient Feature Set using the five criteria for a link prediction technique outlined in Section 2. In Section 5, we already elaborated on the

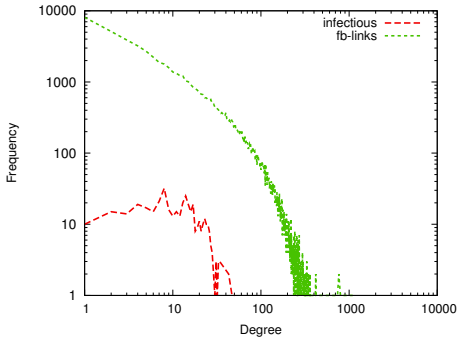

Fig. 2: Degree distribution of datasets `infectious` and `fb-links`.

explainability and efficiency of EFS. Here we continue by substantiating the claim that EFS also meets the accuracy, consistency and generality criteria.

Accuracy. The results in Table 3 show that EFS is able to accurately predict links, achieving results very similar to the accuracy achieved using the full feature set. The bottom row of Table 3 shows how on average EFS achieves an AUROC of a remarkable 99.5% of the AUROC value of the full feature set.

Consistency. Not only did the EFS-based classifier yield very high AUROC values relative to the full feature set, it did so with a high degree of consistency: the lowest EFS AUROC measured was 98.8% and the highest AUROC measured was 99.8% of the full feature set AUROC value.

Generality. We have tested EFS on a broad range of diverse networks, as can be seen in Table 2. The networks vary in number of nodes and links, clustering coefficient, diameter, directedness, weightedness and many other properties. The Efficient Feature Set performs well regardless of these differences in network properties. Even the absence of a power law in the degree distribution, such as in the `infectious` network, does not influence the prediction ability of the Efficient Feature Set. It is indeed a generic way of predicting links in real-world networks.

7 Conclusion

The proposed Efficient Feature Set (EFS) is a relatively small set of structural network features, categorized based on the topological scope of the network at which they each uniquely capture dynamics. Together, the feature categories can be used in a supervised learning framework to predict future links in real-world networks in an efficient and explainable way. Experiments show that the approach reaches over 99% of the accuracy of a much larger and more complex set of features. EFS is explainable as the contribution of its feature categories can be linked to the topological properties of the considered networks. Furthermore, the method shows consistent performance with respect to larger feature sets, independent of the network size, hinting towards high scalability. EFS works well independent of network density and type, demonstrating that it is a generic approach to predict links in real-world network data that is evolving over time.

In future work, we want to investigate the order in which links appear and how suitable EFS is for predicting the timestamp of a link. Furthermore, we will look at whether the same approach could be applied to the removal of links, allowing not only the expansion but also the contraction of networks to be predicted.

References

1. L. A. Adamic and E. Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.
2. M. Al Hasan, V. Chaoji, S. Salem, and M. Zaki. Link prediction using supervised learning. In *Workshop on Link Analysis, Counter-terrorism and Security*, 2006.
3. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. 2nd edition, Addison-Wesley, 2011.

4. A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
5. R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *Proceedings ICDM*, pages 96–103, 2008.
6. M. D. Choudhury, H. Sundaram, A. John, and D. D. Seligmann. Social synchrony: Predicting mimicry of user actions. In *Proceedings ICCSE*, pages 151–158, 2009.
7. M. Fire, L. Tenenboim-Chekina, R. Puzis, O. Lesser, L. Rokach, and Y. Elovici. Computationally efficient link prediction in a variety of social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(1):10, 2013.
8. V. Gómez, A. Kaltenbrunner, and V. López. Statistical analysis of social network discussion threads in Slashdot. In *Proceedings WWW*, pages 645–654, 2008.
9. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software. *SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
10. Z. Huang, X. Li, and H. Chen. Link prediction approach to collaborative filtering. In *Proceedings DLT*, pages 141–142, 2005.
11. L. Isella, J. Stehlé, A. Barrat, C. Cattuto, J.-F. Pinton, and W. V. den Broeck. What's in a crowd? Analysis of face-to-face behavioral networks. *Journal of Theoretical Biology*, 271(1):166–180, 2011.
12. L. Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.
13. J. Kleinberg. The small-world phenomenon: An algorithmic perspective. In *Proceedings STOC*, pages 163–170, 2000.
14. KONECT. Linux mailing list replies network, http://konect.uni-koblenz.de, 2015.
15. T. G. Lewis. *Network science: Theory and applications*. John Wiley & Sons, 2011.
16. D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *Proceedings CIKM*, pages 556–559, 2003.
17. R. N. Lichtenwalter, J. T. Lussier, and N. V. Chawla. New perspectives and methods in link prediction. In *Proceedings KDD*, pages 243–252, 2010.
18. L. Lü and T. Zhou. Link prediction in complex networks: A survey. *Physica A: Statistical Mechanics and its Applications*, 390(6):1150–1170, 2011.
19. J. O'Madadhain, J. Hutchins, and P. Smyth. Prediction and ranking algorithms for event-based network data. *SIGKDD Explorations Newsletter*, 7(2):23–30, 2005.
20. T. Opsahl and P. Panzarasa. Clustering in weighted networks. *Social Networks*, 31(2):155–163, 2009.
21. A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI Workshop on Learning Statistical Models from Relational Data*, 2003.
22. J. Preusse, J. Kunegis, M. Thimm, T. Gottron, and S. Staab. Structural dynamics of knowledge networks. In *Proceedings ICWSM*, 2013.
23. R. R. Sarukkai. Link prediction and path analysis using Markov chains. *Computer Networks*, 33(1):377–386, 2000.
24. J. Scott. *Social Network Analysis*. Sage, 2012.
25. F. W. Takes. *Algorithms for Analyzing and Mining Real-World Graphs*. PhD Thesis, Leiden University, 2014.
26. B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi. On the evolution of user interaction in Facebook. In *Proceedings WOSN*, pages 37–42, 2009.
27. B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level topology. *SIGCOMM Computer Communication Review*, 35(1):53–61, 2005.