Link Prediction and Topological Feature Importance in Social Networks

Stephan A. Curiskis **Thomas R. Osborn** **Paul J. Kennedy**

Centre for Quantum Computation and Intelligent Systems
Faculty of Engineering and Information Technology
University of Technology, Sydney,
15 Broadway, Ultimo, NSW 2007,
Email: stephan.a.curiskis@uts.edu.au

Abstract

The problem of link prediction describes how to account for the development of connection structure in a graph. There are many applications of link prediction, such as predicting missing links and future links in online social networks. Much of the literature has focused on limited characteristics of the graph topology or on node attributes, rather than a broad range of measures. There is a rich spectrum of topological features associated with a graph, such as neighbourhood similarity scores, node centrality measures, community structure and path-based distance measures. In this paper we formulate a supervised learning approach to link prediction using a feature set of graph measures chosen to capture a wide range of topological structure. This approach has the advantage that it can be applied to any graph where the connection structure is known. Random forest learning models are used for their high accuracy and measures of feature importance. The feature importance scores reveal the strength of contribution of the topological predictors for link prediction in a variety of synthetically generated network datasets, as well as three real world citation networks. We investigate both undirected and directed cases. Our results show that this approach can deliver very high model precision and recall performance in certain graphs, and good performance generally. Our models also consistently outperform a simpler comparison model we developed to resemble earlier work. In addition, our analysis of variable importance for each dataset reveals meaningful information regarding deep network properties.

Keywords: social networks, link prediction, supervised learning, centrality, community structure, graph topology

1 Introduction

Link prediction describes how the likelihood for a link existing between two nodes in a complex network can be estimated. Many approaches have been proposed, but most require non-topological, node specific information to achieve high accuracy. A key goal of link prediction is the development of an accurate model that can be applied universally to any social network dataset (Shibata et al. 2012).

There are many applications of link prediction. For instance, these methods can be applied for link recommendation to users in online social networks. Another application identified for link prediction models is the evaluation of evolving social network models (Lu & Zhou 2011). Link prediction algorithms estimate the influence of a set of features on the likelihood of link formation. A wide range of topological features can provide information regarding emergent properties of a social network through their predictive importance. This information may provide an empirical basis for the derivation of the rule set of an evolving social network mode. The research question of using a link prediction model to discover information about the deep structure of a social network is still open in the literature.

In this paper, we present a novel link prediction framework that can be applied universally to any network where the topology is known. We define a set of topologically derived features which capture a wide range of network properties, and apply a random forest supervised learning model. Our method is tested against a variety of synthetic networks and real world datasets, for both the undirected and directed cases. We also evaluate whether feature importance scoring can provide information about global and emergent properties of a network.

2 Related Work

There are three common frameworks to link prediction: the similarity based approach, methods based on maximum likelihood estimation, and probabilistic modelling approaches.

The simplest framework for link prediction is the similarity-based approach. In this method, each pair of nodes i and j is assigned a score s_{ij}, defined as the similarity between i and j. All unobserved edges are then ranked according to their scores, with higher ranks having a greater link likelihood. Much of the early literature on link prediction focussed on the use of singular similarity features, or small sets of features. For example, Adamic & Adar (2003) developed a similarity measure based on common neighbours to predict connections amongst web pages. Liben-Nowell & Kleinberg (2007) experimented with a wider range of similarity based measures, but still used each in isolation to rank node pairs with the highest scores. Many similarity indices have been proposed, see Lu & Zhou (2011) for more details. As these methods are relatively simple, each similarity index only considers a limited amount of information regarding the graph topology. As such, the accuracy of these indices is generally quite low.

A more recent framework is to use algorithms based on maximum likelihood estimation. This approach assumes some topological form of the network, e.g. an exponential random graph (Zaccarin & Rivellini 2010). An algorithm then estimates the model parameters against the dataset. Another method was proposed by Clauset et al. (2008) and starts by inferring the hierarchical organisation of a network. The hierarchical structure is assumed to extend across the network, and is applied to predict missing links. A modelling approach known as a stochastic block model has also been developed. These models partition nodes into groups, which strongly influence link probabilty. However, this method is known to ignore heterogeneity in node degree. Zhang et al. (2014) recently extended the stochastic block model by correcting for variable node degree, with improved results. Heterogeneity across other network properties may still be unaccounted for in this approach, limiting its performance in real world datasets. For these methods to work well in practice, the network structure should be first understood and matched to the closest topological form before a model is be built.

The third common framework uses probabilistic modelling. This methodology aims to abstract the underlying link structure of the network through training a probabilistic model, commonly a supervised learning model. For instance, Backstrom & Leskovec (2011) developed a supervised random walk approach. This method combines node attributes as a supervised learning problem to guide a random walk to nodes which are more likely to be connected. There are many types of probabilistic link prediction models which can be applied. Wang et al. (2011) tested a range of supervised learning models on a social network derived from mobile phone data. They found that a decision-tree model performed most effectively when trained on a mixture of both network and data specific node attribute measures.

Recent publications in this area have focussed on wider sets of features used in a supervised learning framework to predict link formation. More diverse topological features can capture different types of complex structure. Shibata et al. (2012) applied a support vector machine learning model to a variety of features on citation networks, including similarity scores, some centrality measures, community classification and node attributes. This approach achieved high performance, and the model weights for each feature were provided as importance measures. The non-topological node attribute features strongly influenced most of their models, and it was found that different models were required for different citation networks. Bliss et al. (2014) analysed link prediction on a large Twitter social network using a wide variety of topological and node attribute similarity features. An evolutionary algorithm was used to estimate the coefficients for a linear combination of features, with good results. Both of these publications, however, utilised non-topological attributes. This limits the network datasets to which their application can be applied to, and are difficult to compare across other networks. There is a great deal of interest in this area, so for a detailed review see Lu & Zhou (2011) and, more recently, Wang et al. (2014).

The method we present in this paper is motivated by the above publications using the supervised learning framework. However, we only consider topological features to ensure that our approach can be applied universally to any network where the topology is known. We contribute to the existing literature by extending the range of topological features used. We also show that a model with a wider range of topological features consistently outperforms a reduced feature set model. A random forest model is used as it can deliver high accuracy and analysis of feature importance. The results outlined in this work provide information regarding emergent properties of synthetically generated networks, as well as three real world citation networks.

The rest of this paper is organised as follows. Section 3 outlines the research methods. Section 4 delivers results regarding the model performance on each dataset, and the analysis of feature importance. We conclude in Section 5, and outline the future directions for this work.

3 Research Methods

In this section, we outline our supervised learning framework for link prediction using a purely topological feature set. We proceed by describing the datasets used and the data preparation and modelling approach. The features are then defined, followed by the model evaluation methodology.

We define a graph G as an ordered pair $G = (V, E)$ comprising a set of nodes V and edges E. The graph is endowed with nodes v_i and edges e_{ij}, where $i, j = 1, \ldots, n$. Edges are symmetric for undirected networks, i.e. $e_{ij} = e_{ji}$, but $e_{ij} \neq e_{ji}$ for directed networks. For directed networks, we will refer to the "to" node as y, and the "from" node as x.

3.1 Data Description

Our model is applied over three types of synthetically generated networks: an Erdős-Rényi random graph, a small-world network, and a scale-free network. While idealised and simplistic, these three graphs can provide topology often observed in real world networks (Newman 2003). These networks are generated using the *igraph* package with *R* statistical software (Csardi & Nepusz 2006). For each model type, we define a variable m which varies the number of edges in the network. The parameter m roughly gives the number of edges as $m \times V$, and we take $m \in \{1, 2, 3\}$. Higher values of parameter m tend to give unrealistic properties, such as much higher graph densities than the real world networks. The model performance also does not vary substantially with $m > 3$.

- *Scale-free* networks are generated by the preferential attachment mechanism, where new nodes are connected preferentially to existing nodes with higher degrees. We generate scale-free networks SF_m with 2,000 nodes. Parameter m is defined as the number of edges added per node.
- *Small-world* networks start with a regular lattice, then proceed to rewire edges randomly across the network. We generate small-world networks SW_m with starting lattice dimension equal to 1, nodes equal to 2,000 and rewiring probability of 0.05. Parameter m is defined as the lattice connection neighbourhood distance.
- *Erdős-Rényi* random graphs start with a fixed number of nodes, and edges are created randomly with uniform probability. We generate Erdős-Rényi random graphs ER_m with 2,000 nodes. The uniform connection probability is defined as $\frac{m}{1{,}000}$.

We have chosen three real world network data sets to apply the link prediction model to: Cora, Citeseer and WebKB (Sen et al. 2008). These data sets have been chosen as the full connection structure is provided, and they have been used in recent studies (De et al. 2013). All three data sets represent citation networks, are directed, and all contain the full connection structure and node attributes. Table 1 outlines the key properties of each graph. *Di* gives the diameter of the network, *APL* is the average path length, *Cls* gives the clustering coefficient of the network, and *Dns* describes the graph density.

Table 1: Key properties of graph datasets

Graph	V	E	Di	APL	Cls	Dns
Synthetic						
SF$_1$	2,000	1,999	17	7.8	0	0.001
SF$_2$	2,000	3,997	7	3.8	0.006	0.002
SF$_3$	2,000	5,994	6	3	0.01	0.003
SW$_1$	2,000	2,000	132	51.4	0	0.001
SW$_2$	2,000	4,000	20	10.1	0.362	0.002
SW$_3$	2,000	6,000	13	7	0.436	0.003
ER$_1$	2,000	1,913	33	11	0.001	0.001
ER$_2$	2,000	3,917	12	5.7	0.002	0.002
ER$_3$	2,000	5,951	8	4.5	0.003	0.003
Real World						
WebKB	878	1,388	8	3.1	0.036	0.004
Cora	2,709	5,278	19	6.3	0.093	0.001
Citeseer	3,328	4,552	28	9.3	0.13	0.001

In addition to the key graph measures, the networks are plotted with a Fruchterman-Reingold layout, which gives a visual indication of their structure. Figures 1(a) and 1(b) show the plots of the first two scale-free networks SF$_1$ and SF$_2$, respectively. SF$_3$ has been excluded as the structure becomes difficult to discern visually. The branch-like structure is clearly visible in SF$_1$. Similarly, Figures 1(c) and 1(d) depict the structure of the small-world networks SW_1 and SW_2, and Figures 1(e) and 1(f) plot ER_1 and ER_2. It is noted that these latter graphs are not fully connected, however the largest component makes up the majority of the network in both cases.

Figures 1(g), 1(h) and 1(i) show the structure of the real world datasets. Anecdotally, we can see some common patterns amongst the three real world graphs and our generated graphs. For example, the WebKB graph shown in Figure 1(g) appears to have a similar branching structure to the scale-free network in Figure 1(a). In terms of graph measures, WebKB possesses similar properties to SF$_2$ and SW$_2$, although the scale-free network has a lower clustering coefficient, and the small-world network has a smaller diameter.

While the link prediction method outlined in this paper can be applied to both directed and undirected networks, we note that there may be differences in performance in each case. We therefore consider both directed and undirected interpretations of the above networks in our application.

3.2 Data Preparation and Learning Method

Link prediction is known to be a very unbalanced classification problem (Wang et al. 2014). There are usually a large number of node pairs to predict over, and a small number of actual links. We address these issues through implementing a sampling process. We also adopt a random forest learning model. This modelling approach has been chosen because it can be effectively trained to distinguish between unbalanced classes (Breiman 2001). It is proven to be particularly robust to data outliers and also very accurate. Lastly, it can provide measures for the importance of each feature on the likelihood of link formation, which we evaluate for each data set.

Table 2: Link prediction features

Feature	Category
1. Jaccard coefficient	Similarity
2. Adamic-Adar index	
3. Dice similarity	
4. Degree	Centrality
5. Betweenness	
6. Closeness	
7. Eigenvector	
8. PageRank	
9. In same community	Community
10. Community density	
11. Community clustering	
12. Cross-community edge weight	
13. Node pair geodesic	Distance
14. Community pair geodesic	

In all data sets, we have prepared the data for modelling in the following way:

- We create the feature set for all node pairs in G, as defined in Section 3.3
- The data is split into 70%/30% training/testing sets using random sampling.
- In the training data set, we select all the link observations (True class), and sample the remaining set of node pairs (False class) such that there is a 1:100 ratio between the True and False link classes. Through experimentation, we found that oversampling the False cases in the training data produces a more accurate model. 1:100 provides a reasonably representative sample of False cases, while still providing a proportion of True cases high enough for the random forest to model accurately.
- The random forest model is trained using 50 trees, since we found through experimentation that the model accuracy did not improve with additional trees. We also set the number of features included in each tree to the floor of the square root of the total number of features.

3.3 Model Features

We present four categories of measures in this paper: similarity scores, node centrality, community structure and distance measures. While the real world datasets have available certain node attributes, our aim is to use only topological features in our model. This allows for our method to be applied in a standard way to a wide variety of networks in future. We also consider directed cases of each measure. Table 2 outlines the features used by their category. We present the directed version of the features as the generalisation to undirected graphs is trivial.

We refer to node pairs $x, y \in G$, where $x \neq y$ as xy, and classify x as the *from* node and y as the *to* node.

3.3.1 Similarity Measures

Similarity measures have been applied extensively to link prediction. A simple approach to link prediction is to rank all pairs of nodes by their score according to a specific similarity index, and take those node pairs with the highest score to be the most likely connected pairs (Lu & Zhou 2011). However, these

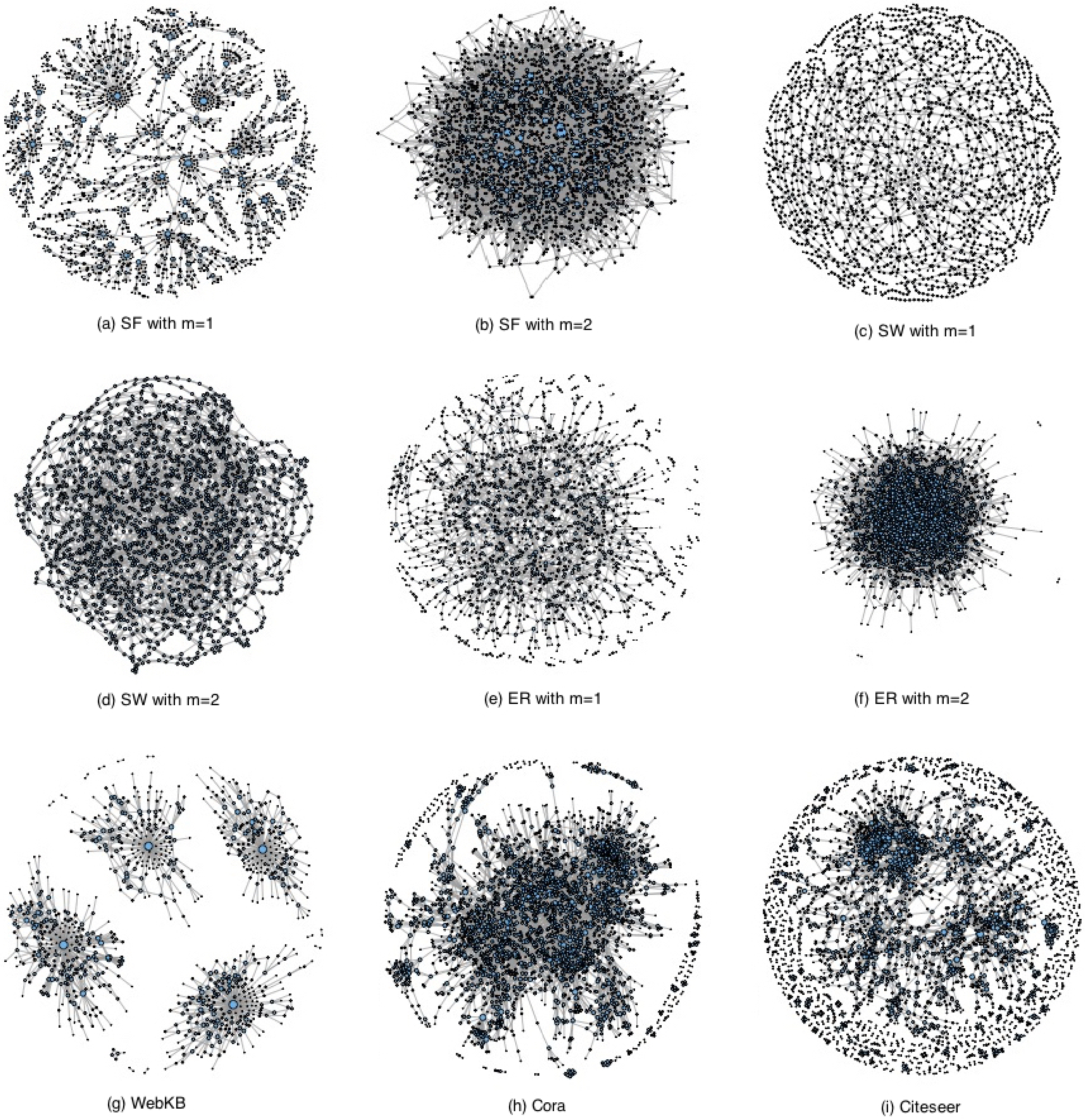


Figure 1: Plots of all network datasets used, in the undirected interpretation. Node size is scaled to represent the degree of the node. (a) represents the scale-free network SF_1, and SF_2 is shown in (b). The branch-like structure is clearly visible in (a), where each node is only connected to one existing node. (c) and (d) depict the plots of the small world networks, SW_1 and SW_2. (e) and (f) show the plots of the Erdős-Rényi networks with $m = 1$ and $m = 2$ respectively. It is clear that there is a large connected component in ER_1, with a number of smaller components around the edges. With $m = 2$ in (f), most of the network is connected. (g), (h) and (i) show plots of the real world networks WebKB, Cora and Citeseer, respectively. It is evident that WebKB consists of four connected components, each with a highly connected hub node, indicating scale-free structure. Cora and Citeseer, shown in (h) and (i), both show a large connected component with a number of smaller components, similar to (e).

measures can also be used as features in the supervised learning approach to link prediction. We consider the following three similarity measures to use as features in the supervised learning problem.

(1) Jaccard similarity coefficient (Sim_{xy}^{Jacc}):

The Jaccard similarity coefficient of two vertices is the number of common neighbours divided by the union of the neighbours of both vertices. For a node x, let $\Gamma(x)$ denote the set of neighbours of x. For directed networks, $\Gamma(x)$ defines the set of nodes with a link *from* node x. The Jaccard similarity coefficient is then defined as

$$Sim_{xy}^{Jacc} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}.$$

(2) Adamic-Adar index (Sim_{xy}^{AA}) (Adamic & Adar 2003):

This index extends the simple counting of common neighbours to include a term which gives less connected neighbours higher weight. Letting $k_{out,z}$ be the out-degree of node z, the Adamic-Adar Index is defined as

$$Sim_{xy}^{AA} = \sum_{z \in \Gamma(x) \cup \Gamma(y)} \frac{1}{\log k_{out,z}}.$$

(3) Dice similarity coefficient (Sim_{xy}^{Dice}):

This similarity coefficient, also known as the Sørensen index, is similar to the Jaccard index in form, but has been applied to ecological communities and is known to be robust to outliers (Sørensen 1948). The Dice similarity coefficient is given by

$$Sim_{xy}^{Dice} = 2\frac{|\Gamma(x) \cap \Gamma(y)|}{k_{out,x} + k_{out,y}}.$$

3.3.2 Node Centrality Measures

Centrality measures have been used for a long time within the field of social network analysis to describe the relative influence of a node over the network. Indeed, scale-free networks arose from the insight that individuals may connect preferentially to more highly connected individuals, i.e. those with high degree centrality (Barabási & Albert 1999). Shibata et al. (2007) showed that in citation networks, the betweenness centrality measure was positively correlated between pairs of nodes where a connection existed, and proved to be a significant predictor for future connections. Further to these results, we expect that in social networks individuals may connect preferentially to other individuals according to a variety of centrality measures. We therefore include a broad range of features derived from centrality measures into our supervised learning model and evaluate their predictive power and importance.

The question of how the centrality measures for each node pair are used as features requires consideration. A number of approaches have been applied previously. For example, Shibata et al. (2007) derive a feature based on the difference between the in-degrees of the two nodes, $Cn_{xy}^{PA_2} = k_{in,x} - k_{in,y}$. This measure will capture differences in the in-degree between the nodes, but may not distinguish between high and low in-degree nodes. Another feature derivation commonly used for preferential attachment is defined as the product of each node's degree, i.e. $Cn_{xy}^{PA_1} = k_x \times k_y$ (Lu & Zhou 2011, Barabási & Albert 1999). This method gives more weight to either node having high degree, however the formula does not distinguish between the *to* and *from* nodes. Additional consideration of this measure is therefore required for directed networks. However, this approach is more common in the literature (Wang et al. 2014) so we adopt the same convention. To accomodate the directed network case we also include as a separate feature the centrality measure for the *to* node, y. The undirected centrality product features are labelled as $Cn_{x\times y}^{Measure}$, and the additional directed centrality features are labelled $Cn_{y}^{Measure}$. The list of centrality features are outlined as follows.

(4) Degree centrality ($Cn_{x\times y}^{Dgre}, Cn_{y}^{Dgre}$):

The degree centrality is defined simply as the number of connections of each node in an undirected network, or the in-degree in a directed network. We construct a feature for the product of the in-degrees of both nodes, and another feature for the in-degree of the *to* node:

$$\begin{aligned} Cn_{x\times y}^{Dgre} &= k_{out,x} \times k_{in,y}, \\ Cn_{y}^{Dgre} &= k_{in,y} \end{aligned}$$

We expect that a higher degree product indicates high connection likelihood in the undirected case. In the directed case, we also expect that *from* nodes x with a higher out-degree and *to* nodes y with a higher in-degree are more likely to be connected. The in-degree of the *to* node y as a separate feature for directed networks.

(5) Betweenness centrality ($Cn_{x\times y}^{Btwn}, Cn_{y}^{Btwn}$):

Previous studies have revealed that betweenness centrality can be a useful predictor of links in citation networks (Shibata et al. 2012). This measure represents the extent to which a node lies on the shortest paths (geodesics) between other nodes, which can be a useful indicator of influence on network flow. Nodes with high betweenness centralities tend to bridge otherwise unconnected subsets of a network. Formally, for nodes i, s and t, let

$$n_{st}^{i} = \begin{cases} 1 & \text{if } i \text{ lies on the geodesic path from } s \text{ and } t \\ 0 & \text{otherwise.} \end{cases}$$

The betweenness centrality of a node i is then defined as:

$$Cn_{i}^{Btwn} = \sum_{st} \frac{n_{st}^{i}}{g_{st}},$$

where g_{st} gives the total number of geodesic paths from s to t. Our features are then defined as:

$$Cn_{x\times y}^{Btwn} = Cn_{x}^{Btwn} \times Cn_{y}^{Btwn},$$

with Cn_{y}^{Btwn} included in the directed case.

(6) Closeness centrality ($Cn_{x\times y}^{Clse}, Cn_{y}^{Clse}$):

Closeness centrality measures the inverse of the mean distance from a node to all other nodes (Newman 2010). Letting d_{ij} be the length of the geodesic path from node i to j, closeness centrality is defined as:

$$Cn_{i}^{Clse} = \frac{n}{\sum_j d_{ij}}.$$

We therefore construct our features as

$$\begin{aligned} Cn_{x\times y}^{Clse} &= Cn_{x}^{Clse} \times Cn_{y}^{Clse} \\ &= \frac{n}{\sum_i d_{xi}} \times \frac{n}{\sum_i d_{yi}}, \end{aligned}$$

with Cn_{y}^{Clse} included as a separate feature in the directed case.

(7) Eigenvector centrality ($Cn_{x\times y}^{Egnv}, Cn_{y}^{Egnv}$):

Eigenvector centrality extends from degree centrality with the acknowledgement that not all neighbours are equal. This measure awards each node with a score proportional to the sum of the scores of its neighbours. Derivations of eigenvector centrality have been applied effectively to link prediction (Symeonidis et al. 2013), so we expect that it will be a useful feature. The eigenvector centrality measure for node i, v_i, is defined as:

$$v_i = \kappa_1^{-1} \sum_j A_{ij} v_j,$$

where κ_1 is the leading eigenvalue of A, A_{ij} is the ij^{th} element of A, and v_j is the eigenvector centrality of node j. We construct our feature as:

$$\begin{aligned} Cn_{x\times y}^{Egnv} &= Cn_x^{Egnv} \times Cn_y^{Egnv} \\ &= \left(\kappa_1^{-1} \sum_i A_{xi} v_i\right) \times \left(\kappa_1^{-1} \sum_i A_{yi} v_i\right). \end{aligned}$$

As before, we also use Cn_y^{Egnv} as a feature in the directed case.

(8) PageRank ($Cn_{x\times y}^{PgRk}, Cn_y^{PgRk}$):

PageRank was originally designed as a measure of web page importance (Page & Brin 1998). It is similar to the eigenvector centrality in form, however the neighbour centrality score for each neighbour i is divided by that node's out-degree, k_i^{out}. This penalises the influence of neighbours with very high out-degree. We construct our PageRank features as follows:

$$\begin{aligned} Cn_{x\times y}^{PgRk} &= Cn_x^{PgRk} \times Cn_y^{PgRk} \\ &= \left(\alpha \sum_i A_{xi} \frac{v_i}{k_i^{out}} + \beta\right) \left(\alpha \sum_i A_{yi} \frac{v_i}{k_i^{out}} + \beta\right), \end{aligned}$$

where α and β are constants. We also include Cn_y^{PgRk} as before in the directed case.

3.3.3 Community Measures

Community detection describes the problem of partitioning a graph into densely connected subsets, commonly referred to as communities. A graph's community structure has been shown to be predictive of link formation, as nodes within the same community are more likely to be connected. For example, Shibata et al. (2012) include a feature for whether two nodes are in the same community in their supervised learning model for link prediction, with good performance.

The problem of community detection has received a great deal of interest, see Fortunato (2010) for a comprehensive review. However, many of the community detection methods have been developed for undirected graphs, and some have a high computational cost. We utilise the infomap community detection algorithm as it has computational time $O(V(V \times E))$ and can handle directed graphs (Rosvall et al. 2009). Once the graph is partitioned into communities, we create the community graph G_C by aggregating the vertices of G to their community partitions. Each node in G_C is therefore a community partition of G, and we use the symbols μ and ν to refer to the *from* and *to* nodes in G_C, respectively. We also let $\mu, \nu = 1, \ldots, k$, where k is the total number of community partitions in G. We assign edge weights according to the number of links between each community pair in G. The community features are defined as follows:

(9) In same community (Cm_{xy}^{Comm}):

We expect that a pair of nodes xy in the same community should have a higher likelihood of being connected given that the communities are partitioned according to connection density. This measure is defined for node pairs simply as:

$$Cm_{xy}^{Comm} = \begin{cases} 1 & \text{if } x \text{ is in the same community as } y \\ 0 & \text{otherwise.} \end{cases}$$

Including a feature for two links being in the same community should effectively reduce the link likelihood space dramatically. This will yield more accurate link prediction in networks where the community clustering is strong.

(10) Community density (Cm_{xy}^{Dens}):

The density of a graph is simply the number of edges divided by the number of possible edges. We apply this measure to each community μ, represented as induced subgraphs of G, $\mu \subset G$. Letting $E(G)$ define the set of edges in G, we define $|E(\mu)|$ as the number of edges xy, with $x, y \in \mu, G$. The density of μ is then defined as

$$Dens_\mu = \frac{|E(\mu)|}{\sum_{x,y \in \mu, G} 1}.$$

We then construct our feature vector for each node pair in G as

$$Cm_{xy}^{Dens} = \begin{cases} Dens_\mu & \text{if } x \text{ and } y \text{ share community } \mu \\ 0 & \text{otherwise.} \end{cases}$$

We expect that two nodes in the same community with higher density will have a greater likelihood of being connected than those in a different community with lower density, or in separate communities.

(11) Community clustering coefficient (Cm_{xy}^{Clst}):

The clustering coefficient, also known as transitivity, for community subgraph μ is defined as

$$Clst_\mu = \frac{\text{(number of closed paths of length two)}}{\text{(number of paths of length two)}}.$$

Similarly to the community density, we expect that nodes in the same community with higher clustering are more likely to be connected. The feature vector is constructed as

$$Cm_{xy}^{Clst} = \begin{cases} Clst_\mu & \text{if } x \text{ and } y \text{ share community } \mu \\ 0 & \text{otherwise.} \end{cases}$$

(12) Cross-community edge weight (Cm_{xy}^{Ewgt}):

The community graph G_C provides a more coarse network from which we can take attributes to use for link prediction of node pairs in G. One straightforward measure is the edge weight between each community pair. We construct a feature vector for the node pairs in G based on the edge weights between their respective communities, where the nodes are not in the same community. Let A_C denote the adjacency matrix for G_C, and let $\mu, \nu \in G_C$ denote the community classifications for nodes $x, y \in G$ respectively. The cross-community edge weight is then defined as:

$$Cm_{xy}^{Ewgt} = \begin{cases} \sum_{xy \in G} A_{C_{\mu\nu}} & \text{if } \mu \neq \nu \\ 0 & \text{otherwise.} \end{cases}$$

3.3.4 Distance Measures

Distance measures indicate the number of links between a pair of nodes. Our expectation is that nodes which are closer together are more likely to be connected. For the purposes of link prediction, we only consider distance greater than one for each node pair. This ensures any direct connections are not counted, as links in the graph have distance of one. We construct distance measures for both the individual node pairs, and the distance between their communities in G_C.

(13) Node pair geodesic (D_{xy}^{Node}):

For a pair of nodes $x, y \in G$, let $d_{xy,2}$ be the shortest path (geodesic) from x to y of length greater than or equal to 2. Our feature vector is then simply given by

$$D_{xy}^{Node} = d_{xy,2}$$

(14) Community pair geodesic (D_{xy}^{Comm}):

For a pair of nodes $x, y \in G$ lying in communities $\mu, \nu \in G_C$ respectively, our feature vector for each node pair is defined as the distance from μ to ν:

$$D_{xy}^{Comm} = d_{\mu\nu,2}$$

An issue that arises with the distance features, particularly on small networks, is that there may not be a path with length greater than one for node pairs which are connected. If the network is not fully connected, then there will also be node pairs that do not have a geodesic. These issues effectively introduce missing values into the observations. To account for this, we simply replace any missing distance values with the mean across the dataset which allows for these observations to be included in the model. More sophisticated approachs may be developed to account for missing distance values, but we leave this to future work.

3.4 Model Evaluation

Given the unbalanced nature of the link prediction problem, measuring the performance of a model requires consideration. A common approach is to use the model precision, recall, and the associated F_1 measure (Wang et al. 2014). With our model trained on the training data set, we apply the following evaluation measures on the testing data set only. We abbreviate true positives to TP, false positives to FP, and false negatives to FN.

$$Precision = \frac{TP}{TP + FP},$$

$$Recall = \frac{TP}{TP + FN},$$

$$F_1 = \frac{TP}{TP + FP + FN}.$$

We have chosen precision, recall and F_1 specifically because they are useful in unbalanced problems since they ignore true negatives. There are likely to be a very large number of records classified as true negatives due to the high number of node pairs that do not have links, which distort the performance results. In addition to these evaluation measures, we also provide the precision and recall charts.

Table 3: Link prediction features for comparison model

Feature	Category
1. Jaccard coefficient	Similarity
2. Adamic-Adar index	
3. Dice similarity	
4. Degree	Centrality
5. Betweenness	
6. In same community	Community
7. Community density	

Table 4: Model performance on all undirected network datasets

Network	Precision	Recall	$\mathbf{F}_1$	AUC
Synthetic				
$\mathbf{SF}_1$	1	1	1	1
$\mathbf{SF}_2$	0.552	0.729	0.628	0.9932
$\mathbf{SF}_3$	0.529	0.527	0.528	0.9729
$\mathbf{SW}_1$	1	0.998	0.999	1.000
$\mathbf{SW}_2$	0.548	0.851	0.666	0.9932
$\mathbf{SW}_3$	0.56	0.829	0.669	0.9858
$\mathbf{ER}_1$	0.82	0.964	0.886	0.9999
$\mathbf{ER}_2$	0.622	0.476	0.54	0.951
$\mathbf{ER}_3$	0.463	0.267	0.339	0.8206
Real World				
WebKB	0.709	0.791	0.748	0.9867
Cora	0.402	0.727	0.518	0.983
Citeseer	0.406	0.876	0.555	0.9969

In this paper, we expect that a wider range of topological features can produce a more accurate link prediction model, as well as revealing diverse information about deeper graph properties. To demonstrate the performance improvement, we create a comparison model based on the same topological feature set used by Shibata et al. (2012), with a random forest model as the learning algorithm rather than the support vector machine approach. The list of features for the comparison models is given in Table 3. It is noted that the approach by Shibata et al. (2012) includes non-topological features, so we are not making a direct comparison between the two different approaches. We also modify the comparison model features to be applicable to undirected networks. The directed and undirected cases are treated in the same way as outlined in Section 3.3. We compare the performance of models trained with the full feature set to the comparison set by producing precision recall charts for both.

The last aspect of model evaluation we consider is to determine the relative importance of the topological features. We provide an analysis of the random forest mean decrease accuracy importance measure, and discuss the results with respect to model accuracy.

4 Results

4.1 Model Performance: Undirected Graphs

Table 4 outlines the model performance on all data sets, interpreted as undirected networks. The models trained on the synthetic networks with parameter $m = 1$ give very high performance. The model trained on the scale-free network with $m = 1$ actually classifies every node pair correctly. However, as we increase the value of parameter m, the model perfor-

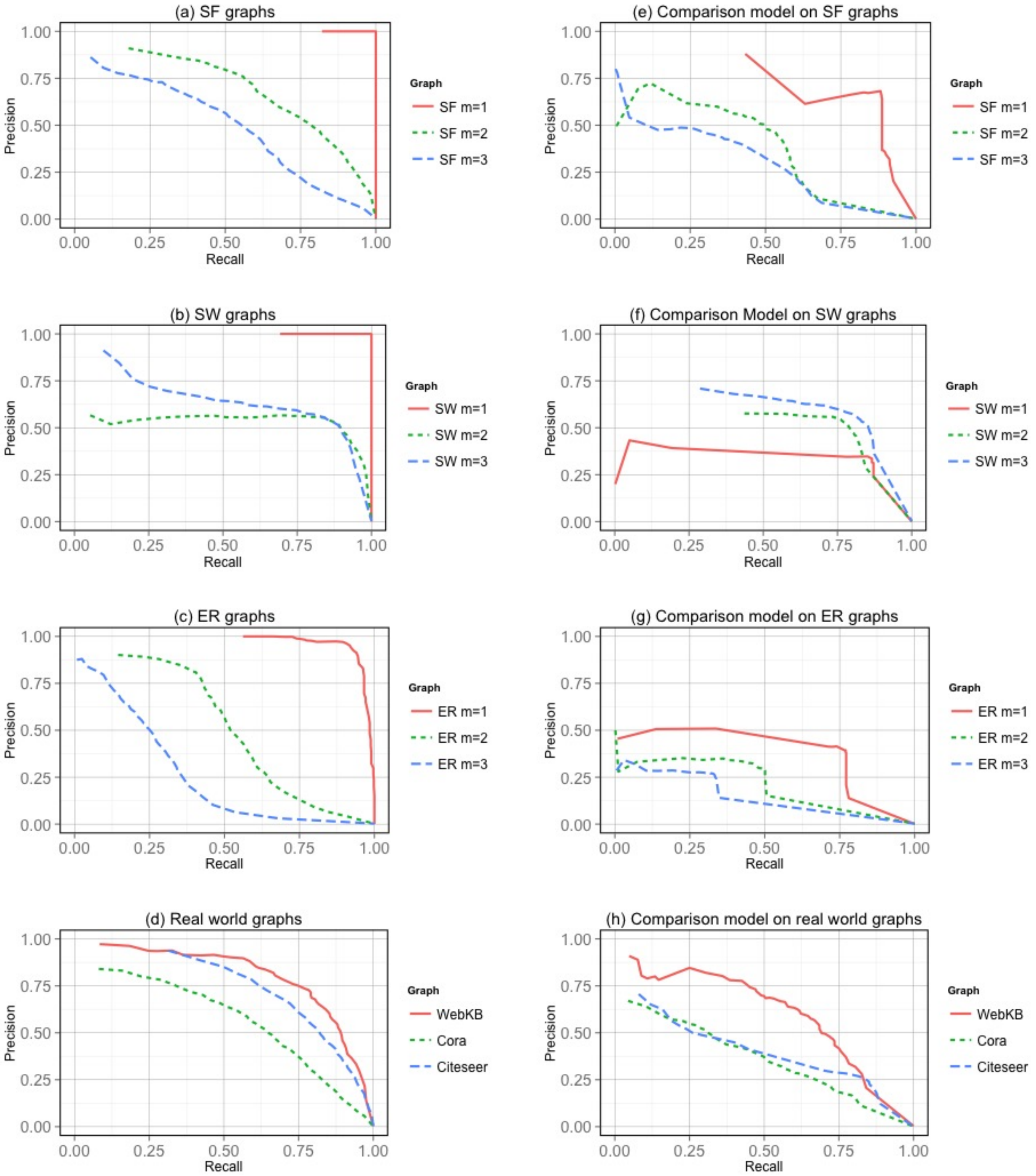


Figure 2: Plot of the precision and recall charts for the undirected scale-free networks (a), small-world networks (b), Erdős-Rényi random graphs (c), and the real world networks (d). Plots (e) to (h) depict the performance of the comparison models trained on the same network as opposite with a smaller range of topological features.

mance decreases in all cases. The scale-free network models lose precision with higher values of m, and recall drops substantially as well. The small-world network maintains good recall and F_1, although precision does drop with $m > 1$. The Erdős-Rényi random graph loses precision as m is increased, but the recall drops much more rapidly to 0.267 with $m = 3$.

Figure 2(a-c) depicts the precision and recall curves for the undirected synthetic networks, where recall is plotted on the x-axis and precision on the y-axis. It is clear from Figure 2(a) that the model performs well for the scale-free graphs with $m = 1, 2$ and 3, given that these curves remain in the top triangle of the plot, and their concavity is down. The small-world networks in Figure 2(b) retain good performance as well. It is interesting that the model on the network with $m = 3$ actually seems to outperform that with $m = 2$. We can see the former model achieves a slightly higher precision of 0.56, compared to 0.548. The Erdős-Rényi random graphs in Figure 2(c) show good performance for the case with $m = 1$, but a large difference to the case with $m = 2$, and again with $m = 3$. All models, except ER$_3$, deliver AUC higher than 0.97. These results indicate that our learning model can accurately describe the network generating processes of these three undirected synthetic networks, however this accuracy diminishes as nodes are added to the network with more edges.

On the real world datasets the model gives consistently high recall and AUC values. This indicates that a high proportion of the actual links are being classed correctly. Precision gives how many of the predicted links are actually links, and our model performs well on the WebKB dataset with precision of 0.709. However, the precision values for Cora and Citeseer are much lower at around 0.4. Figure 2(d) gives the precision against recall curves for the real world datasets. We can see that our model performs the best on WebKB, followed closely by Citeseer. The model classification gives a low precision score to the model trained on Citeseer, however the recall is much higher than Cora. The precision recall curve for Citeseer indicates that the model could deliver higher precision with a small drop in recall.

Table 5: Model performance on all directed network datasets

Network	Precision	Recall	F_1	AUC
Synthetic				
SF$_1$	0.201	0.926	0.33	0.9994
SF$_2$	0.237	0.522	0.326	0.9443
SF$_3$	0.206	0.194	0.2	0.7293
ER$_1$	0.29	0.45	0.353	0.9855
ER$_2$	0.345	0.154	0.213	0.9401
ER$_3$	0.35	0.075	0.123	0.8391
Real World				
WebKB	0.487	0.796	0.604	0.9988
Cora	0.297	0.798	0.433	0.9988
Citeseer	0.237	0.955	0.379	0.9997

Figure 2(e) to 2(h) show the precision and recall curves for the comparison model with a reduced topological feature set, trained on the same graphs as the full model. The key observation from these plots is that these models fail to achieve high performance in both recall and precision. Many of these curves are also not smooth or monotonic. This is likely due to the feature sampling of the random forest over the reduced feature set; the curve may change sharply when an important feature is excluded or included. We therefore conclude that by including a wider range of topological features in our supervised learning model, we achieve much higher performance than a smaller set of features.

4.2 Model Performance: Directed Graphs

When we apply the same methodology to the directed versions of these networks, including the centrality measures for the *to* nodes y, the results differ substantially. Table 5 gives the model performance on the directed interpretation of the networks. *igraph* does not currently support a directed version of the small-world network, so we have removed it from this analysis. It is clear that the approach is not as accurate on directed networks as their undirected interpretation. Precision is down significantly for all networks, however recall remains high for the real world network models and the synthetic graphs with $m = 1$. This result suggests that the model can still identify the node pairs that have links, but cannot accurately distinguish the direction of the link. We will explore this more through an analysis of the feature importance on each network.

Figure 3(a-c) show the precision and recall charts for the directed networks. We can clearly see that precision is not as high in the undirected case for the real world graphs, although recall is slightly improved. However, it is clear that the synthetic networks have not performed as well. To compare, Figure 3(d-f) show the precision and recall charts for the comparison model with the reduced feature set. It is clearly evident that the full model outperforms the simpler model. However, it is noted that the comparison model seems to perform slightly better on the real world networks than the synthetic. This may be due to a more balanced feature importance. We explore feature importance in Section 4.4.

4.3 Feature Importance: Undirected Graphs

As mentioned earlier, one of the advantages of using a random forest model is that features which are of lesser use in prediction tend to be effectively downweighted. One measure of the importance of the features is the *mean decrease accuracy*, which is a scaled average of the prediction accuracy of each feature. It effectively measures the decrease in model accuracy when values of each feature are randomly permuted (Breiman 2001). Figure 4 shows a matrix of the mean decrease accuracy for each feature and graph, for both the undirected and directed cases. To save space, we have only shown the synthetic networks with $m = 2$ as these networks are the most similar to the real world graphs.

The undirected models and graphs are given in Figure 4(a). In the model for Citeseer, community based features and distance measures hold the highest importance. The most important feature in this dataset is the community density, followed by the same community flag and the node geodesic. The importance of this feature set may be due to the large number of unconnected components in the Citeseer graph; nodes in these minor components are likely to be attributed to the same community. The model has therefore limited the possible connection space dramatically by assigning a higher importance to community based features. The analysis is very similar for the Cora dataset as well, however the node geodesic is flagged as the most important feature. In both graphs, the centrality measures are also important.

Of all the networks analysed, the model trained on WebKB delivered the best performance. In this dataset, we see a broader distribution of predictive importance across the feature classes. All the features have a mean decrease accuracy in the same order of magnitude. Community based features have the strongest importance, however centrality measures are very close in magnitude, as are the similarity features. These results are not surprising, given the existence of four key connected components, each with a strongly connected hub node.

The Erdős-Rényi random graph model produced a low score for the recall measure, and we see centrality measures strongly influencing the predictions. Degree centrality was the most important, followed by eigenvector and closeness centralities. It seems that the model has trained closely to the node influence features, given the lack of strong neighbourhood and community structure. This may also explain the low model performance, since node influence is not likely to show huge variation in a random graph.

The feature importance in the small-world network is assigned primarily to the community and node geodesic measures. Again, this makes sense given that local connections are far more common in this network type. Centrality measures are marked with low importance, given that highly influential nodes are rarer.

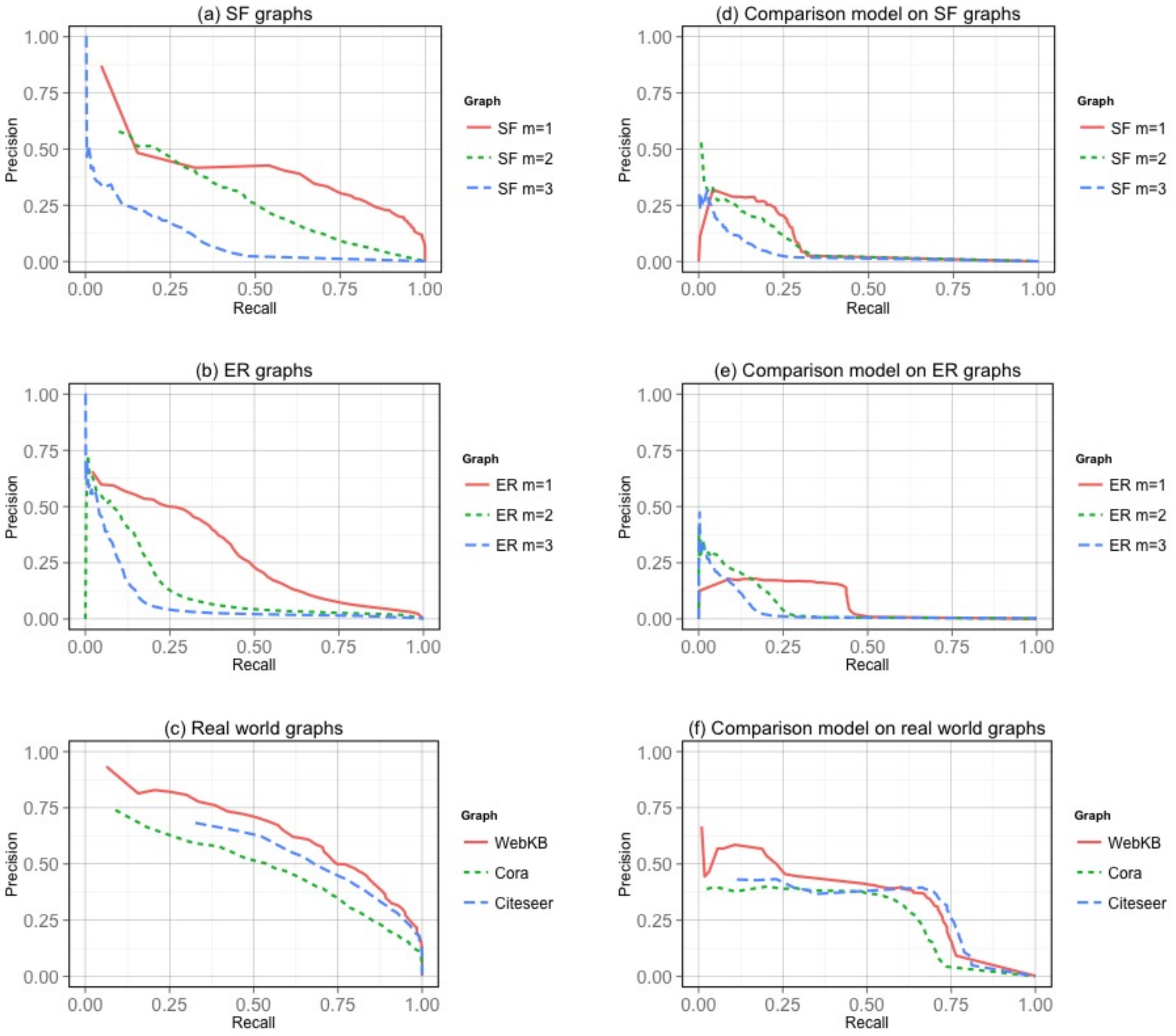


Figure 3: Plot of the precision and recall charts for the directed scale-free networks (a), Erdős-Rényi random graphs (b), and the real world networks (c). Plots (d) to (f) depict the performance of the comparison models trained on the same network as opposite with a smaller range of topological features.

In the scale-free network centrality measures are important as we expect given the preferential attachment growth mechanism. Closeness, PageRank and degree centralities all score highly. However, the most important feature is actually the node geodesic. Community features are also flagged with high importance. These results suggest that community structure and local, densely connected sets of nodes have self-organised in this artificial network.

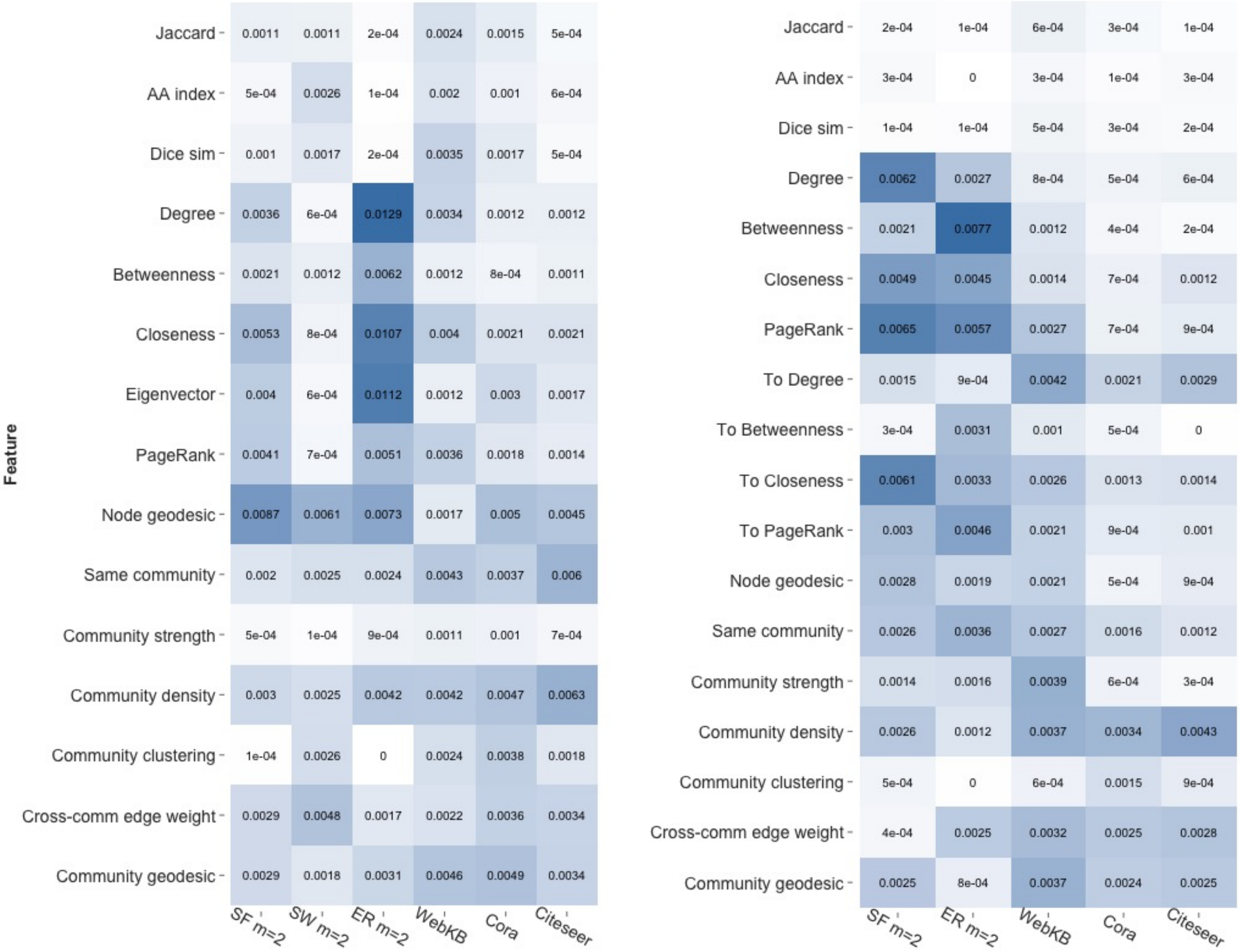
4.4 Feature Importance: Directed Graphs

For the directed case, Figure 4(b) depicts the feature importance per network. As discussed in Section 4.2, the drop in precision described in Section 3.1 may be due to the model assigning more importance to measures that don't necessarily discriminate between the two nodes. In other words, they will identify the node pairs most likely to be connected, but will not accurately determine the link direction.

For the Cora and Citeseer directed networks in Figure 4(b), the model has scored the community based measures highly, particularly the community density feature. This implies that many of the predicted links will likely be in high density communities. However, this measure does not carry any information regarding the direction of the link within a community. These models all deliver high recall, which shows that the community based features are predictive of a link existing between two nodes in either direction. The only other highly important feature is the *to* node's in-degree. It seems that with only one important centrality measure carrying link direction, these models deliver low precision, but manage to correctly classify a majority of the links.

Similarly to the undirected case, the WebKB model achieved the best performance. Again, there is a distribution of predictive importance across the feature categories. Community structure and node centrality measures are assigned high scores, for both the centrality products and the *to* node centralities. The higher precision for this model relative to the models for Cora and Citeseer can be explained by the fact that most of the centrality features are important. However, the node neighbourhood features are not important in the directed case, which likely explains the drop in precision relative to the undirected model.

The synthetic networks, on the other hand, show strong influence of node centrality product features.



(a) Undirected Graphs

(b) Directed Graphs

Figure 4: Plot of the feature importance for both the undirected and directed models. Feature importance is determined by the Random Forest mean decrease accuracy measure. Higher mean decrease accuracy is given darker colour.

Betweenness is highly important for the ER_2 model, and both degree features in the SF_2 model. Relative to the undirected feature importance for these models, similarity scores receive very low importance. Community based features are also less important in the directed version. The lower precision and recall for these models is likely due to the bias towards centrality features.

The results for the directed case clearly suggest that more work is required on the feature set. Particularly, more features are required that carry information regarding the direction of the link. As well as this, additional variations of measures based on in-links and out-links should be considered.

5 Conclusion

We have described a new approach to link prediction using a broad range of graph topological features and a random forest learning model. This approach has the distinct advantage of being applicable to any network where the connection structure is known. It also can discover global or emergent properties of the network through analysis of feature importance.

Our method was tested on three types of synthetically generated datasets, as well as three real world citation networks. In the undirected case, the model performs very well in terms of precision, recall, F_1 and AUC. It was shown that the model can achieve perfect classification of classes on a scale-free network with one edge added per node. However, our approach clearly performs more accurately on synthetic networks with less complex structure, i.e. where the number of edges added per new node is small. We also found that the model in the directed case delivers lower precision as it does not accurately distinguish the direction of the link. Modifying the feature set in the directed case to account for the link direction more strongly may address this issue. To demonstrate that including a broader range of topological features can give higher performance, we compared our approach to a model with fewer features. It was shown that a larger set of features consistently gives higher performance.

It was found that the importance of the input features vary significantly with the different networks. The model performs best when the features are more evenly important across the feature categories, as in the WebKB network. Generally, the model tends to perform very well when strong neighbourhood and community structure is present, in addition to high centrality importance. Finally, the analysis of feature importance provides a method for the discovery of complex, emergent properties within a social network. This was demonstrated through the varying importance of neighbourhood and community-based features in many of the datasets considered.

5.1 Further Work

In future, we aim to extend our approach and apply it to more network datasets, particularly large social networks with complex structure. To reduce the computational cost associated with several of the features used, we will consider reducing the number of features used to those with the highest predictive importance, while retaining as much accuracy as possible. Alternative supervised learning methods will be considered as well. The directed version of our approach also needs further development so that the model can more accurately distinguish the link direction.

References

Adamic, L. A. & Adar, E. (2003), 'Friends and neighbors on the web', *Social Networks* **25**, 211–230.

Backstrom, L. & Leskovec, J. (2011), Supervised random walks: predicting and recommending links in social networks, *in* 'WSDM Conference', Hong Kong, pp. 635–644.

Barabási, A. & Albert, R. (1999), 'Emergence of scaling in random networks', *Science* **286**(5439), 509–512.

Bliss, C., Frank, M., Danforth, C. & Dodds, P. (2014), 'An evolutionary algorithm approach to link prediction in dynamic social networks', *Journal of Computational Science* **5**, 750–764.

Breiman, L. (2001), 'Random forests', *Machine Learning* **45**(1), 5–32.

Clauset, A., Moore, C. & Newman, M. E. J. (2008), 'Hierarchical structure and the prediction of missing links in networks', *Nature* **453**, 98–101.

Csardi, G. & Nepusz, T. (2006), 'The igraph software package for complex network research', *InterJournal* **Complex Systems**, 1695.
URL: *http://igraph.org*

De, A., Ganguly, N. & Chakrabarti, S. (2013), Discriminative link prediction using local links, node features and community structure, *in* '2013 IEEE 13th International Conference on Data Mining', pp. 1009–1014.

Fortunato, S. (2010), 'Community detection in graphs', *Physics Reports* **486**, 75–174.

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.

Lu, L. & Zhou, T. (2011), 'Link prediction in complex networks: a survey', *Physica A* **390**, 1150–1170.

Newman, M. E. J. (2003), 'The structure and function of complex networks', *SIAM Review* **45**(2), 167–256.

Newman, M. E. J. (2010), *Networks: an introduction*, Oxford University Press.

Page, L. & Brin, S. (1998), 'The anatomy of a large-scale hypertextual web search engine', *Proceedings of the Seventh International World Wide Web Conference* **30**(1–7), 107–117.

Rosvall, M., Axelsson, D. & Bergstrom, C. T. (2009), 'The map equation', *The European Physical Journal Special Topics* **178**(1), 13–23.

Sen, P., Namata, G. M., Bilgic, M., Getoor, L., Gallagher, B. & Eliassi-Rad, T. (2008), 'Collective classification in network data', *AI Magazine* **29**(3), 93–106.

Shibata, N., Kajikawa, Y. & Matsushima, K. (2007), 'Topological analysis of citation networks to discover the future core articles', *Journal of the American Society for Information Science and Technology* **56**(6), 872–882.

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.

Sørensen, T. (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', *Biologiske Skrifter* **5**(4).

Symeonidis, P., Iakovidou, N., Mantas, N. & Manolopoulos, Y. (2013), 'From biological to social networks: Link prediction based on multi-way spectral clustering', *Data and Knowledge Engineering* **87**, 226–242.

Wang, D., Pedreschi, D., Song, C., Giannotti, F. & Barabási, A.-L. (2011), Human mobility, social ties, and link prediction, *in* 'Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', KDD '11, ACM, New York, NY, USA, pp. 1100–1108.

Wang, P., Xu, B., Wu, Y. & Zhou, X. (2014), 'Link prediction in social networks: the state-of-the-art', *Science China* **58**(1–38).

Zaccarin, S. & Rivellini, G. (2010), Modelling network data: An introduction to exponential random graph models, *in* F. Palumbo, C. N. Lauro & M. J. Greenacre, eds, 'Data Analysis and Classification', Studies in Classification, Data Analysis, and Knowledge Organization, Springer Berlin Heidelberg, pp. 297–305.

Zhang, X., Wang, X., Zhao, C., Yi, D. & Xie, Z. (2014), 'Degree-corrected stochastic block models and reliability in social networks', *Physica A* **393**, 553–559.