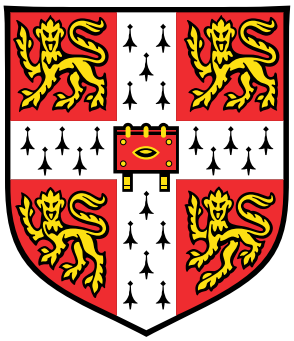
Link prediction in drug-target interactions network using similarity indices



Yiding Lu (Homerton College)
Supervisor: Dr Anna Korhonen, Dr Pietro Lio

Department of Computer Science

University of Cambridge

A dissertation submitted to the University of Cambridge in partial fulfilment of the requirements for the degree of

Master of Philosophy in Advanced Computer Science

June 2015

Declaration

I, Yiding Lu of Homerton College , being a candidate for the M.Phil in Advanced Computer Science, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

Total word count: 12753

Abstract

Network analysis has been a popular topic amongst researchers for a long time due to its wide range of applications. One particular area of interest within network analysis is link prediction, to discover potential edges between nodes that are not explicitly shown in the network. In this paper, we look at a specific application of link prediction, which is *in silico* prediction of drug-target interactions (DTIs). *In silico* DTI prediction refers to a computerised search for potential interactions between chemicals and proteins that have not been discovered by scientists. This has become an important issue in the field of biomedical science due to the explosion of information on biological and chemical knowledge in recent years. The large amount of information available makes it impossible to manually analyse all potential drug candidates. Therefore, the ability to discover potential DTIs with high precision and accuracy has become a key component in genomic drug discovery[1].

Currently, most DTI prediction methods are based on machine learning, such as the bipartite local model which uses genomic sequence similarity in proteins and chemical similarity to generate classification rules[1]. However, this cannot function if we do not know the characteristics of the proteins and chemicals. An alternative method using restricted Boltzmann machine (RBM)[2] requires the system to know if the modes of interaction between chemicals and proteins are direct or indirect for it to function satisfactorily. This creates a problem whereby we require much information and annotation of the nodes and edges in the DTI network for current approaches to deliver satisfactory results.

In this paper, we have proposed and developed a new method for DTI prediction using similarity indices. The distinct advantage of similarity indices is that they are entirely based on network topology, requiring no information about the characteristics of the nodes or edges. We applied four different similarity indices, namely Common Neighbours, Jaccard index, Preferential Attachment and Katz index, to the MATADOR database of DTIs[3]. Comparing our results against the RBM approach, we found that the similarity indices approach has higher precision than the RBM approach for recall between 0 to 0.35 when the modes of interaction in the DTIs are not distinguished between direct interaction and indirect interaction. This shows that similarity indices provide a better approach for DTI prediction when minimum information about the nodes and edges are available.

Acknowledgements

I would like to express my deepest appreciations to my supervisors **Dr Anna Korhonen** and **Dr Pietro Lio** for their guidance and support throughout the project. They provided a clear direction for the project and without them, this thesis would not be possible.

I would also like to thank **Dr Guo Yufan** for her constant support. She also provided many interesting and useful ideas for me to explore which helped in the progress of this project.

In addition, I would like to express my gratitude to **Dr Michael Zeng** and **Dr Wang Yuhao**, authors of the paper *Predicting drug-target interactions using restricted Boltzmann machines* for sharing the results of their experiments with me and allowing me to make a fair comparison of our method of link prediction against their method.

Table of contents

1 Introduction **1**

2 MATADOR Database **4**

3 Related Work **5**

4 Method & Implementation **8**

4.1 Bipartite Graphs 8

4.2 Similarity Indices 10

4.2.1 Common Neighbours (CN) 11

4.2.2 Jaccard Index 11

4.2.3 Preferential Attachment (PA) 12

4.2.4 Katz Index 13

4.3 Implementation 13

4.3.1 10-Fold Cross Validation 14

4.3.2 Precision-Recall Curve 14

5 Results & Discussions **16**

5.1 Pilot Experiment 16

5.2 Common Neighbours (CN) 17

5.3 Jaccard Index 19

5.4 Preferential Attachment (PA) 21

5.5 Katz Index 22

5.6 Weak-Tie Theory 24

5.7 Comparison Between Methods 26

6 Conclusion **29**

7 Future Work **31**

7.1 Additional Datasets 31

7.2 Strength Of Weak-Ties 31

7.3 Literature-Based Discovery 32

References **34**

1. Introduction

Many social, biological and information systems can be represented in the form of networks, where nodes represent individuals, chemicals, proteins, web users, etc., and edges represent the interactions or relationships between nodes[4]. For example, in a social network we can represent human beings as nodes and various social relationships as edges. Likewise in an information network, we represent webpages as nodes and hyperlinks between these webpages as edges. These networks were originally described as random graphs and studied under graph theory in mathematics[5]. However, scientists have recently begin to question if the networks are fundamentally random. The intuition is that these complex networks should exhibit some forms of organising principles and these information should be contained partially in the network topology. This has led to the birth of network analysis, whereby researchers try to develop tools and measurements to discover the latent organising principles of these complex networks.

One important issue that is relevant to network analysis is link prediction, i.e. to estimate the likelihood of the existence of a link between two nodes, based on the given links in the network and the characteristics of the nodes[6]. Link prediction has a wide range of applications including predicting potential friendships between people in a social network, predicting participating actors in an event and predicting semantic relationships between webpages. For example, the user recommendation system developed by Amazon.com recommends new books to users based on a link prediction algorithm[7]. Similarly, Facebook uses link prediction to recommend new friends to users[8]. In this paper, we are interested in a particular application of link prediction — *in silico* prediction of drug-target interactions (DTI).

In recent years, there has been an explosion of information in the field of biomedical science. The abundance of information has paved the way for the building of various large molecular databases, such as the Kyoto Encyclopedia of Genes and Genomes (KEGG), a comprehensive small molecule database, and Therapeutic Target Database (TTD), a linked list of small molecule drugs and drug targets[9]. These databases are useful in providing a library of targets to test out the reactions of new drugs to ensure that they function as expected without harmful side effects[10]. However, this plethora of knowledge also led to a conundrum — it is too time-consuming and costly to research every single possibility of an interaction between each drug and protein. Therefore, *in silico* prediction of drug-target interactions

(DTI) was developed as a solution for this problem.

In silico prediction of DTI refers to an automated search for potential interactions between chemicals and proteins. It is now an integral and important part for the discovery of drugs for known diseases. Being able to predict new DTI with high levels of accuracy and precision is a fundamental requirement and, at the same time, the holy grail for the *in silico* method[11]. There has been huge interest in this field of research and various approaches have been proposed for DTI prediction.

Most current methods for DTI prediction are based on using machine learning. For example, Yamanishi et al. (2008) developed a supervised learning method which integrated both genomic space and chemical space by mapping them together[1] while Wang and Zeng (2013) developed a method based on restricted Boltzmann machine (RBM)[2]. Although machine learning methods tend to perform well, they require information about the characteristics of the nodes, i.e the proteins and the chemicals, to function. For example, the bipartite local model proposed by Bleakley and Yamanishi (2009) uses chemical structure similarity and protein sequence similarity to form its classification rules[11]. Likewise, in the RBM model, the algorithm produces less satisfactory results when the mode of interaction between chemicals and proteins are unknown[2]. This presents a problem as often, in the real world it is impractical or impossible to obtain all the information about the chemicals, proteins and how they interact. Therefore, we propose a new method for DTI prediction based on similarity indices, which requires no prior knowledge about the nodes and edges in the network.

Similarity indices have often been used for link prediction in various complex networks. They are based on the essential attributes of nodes: two nodes are considered similar if they have many common features[12]. Since many attributes of the nodes are generally hidden or unknown, we focus on the structural similarity between nodes which is based solely on the network structure[4]. Similarity indices have the distinct advantage that unlike machine learning methods in link prediction, they do not require prior knowledge about the nodes and edges. The algorithm for similarity indices is also simple to implement. A score s_{xy} is assigned to each pair of nodes x and y based on the similarity between x and y. All resultant non-observed links are then ranked according to their score and hence, the links connecting more similar nodes have a higher likelihood of existence.

While similarity indices have been used for link prediction in social networks and have shown good performance[13], to the best of our knowledge, it has not been used to predict links DTI networks. In this work, we have applied four different types of similarity indices to the Manually Annotated Target and Drugs Online Resources (MATADOR) database of DTIs[3]: Common Neighbours, Jaccard index, Preferential Attachment and Katz index. Given the bipartite nature of the DTI network, we make modifications to several of the similarity indices.

The results of the experiment are compared against the RBM method. Our results show that when the modes of interaction between chemicals and proteins are not determined, Common Neighbours, Jaccard index and Katz index have higher precision than the RBM method when recall is below 0.35. This is important as this allows researchers to experimentally confirm these potential DTI with higher accuracy, thus saving time and money. Therefore, we have developed a new method of using similarity indices that is superior to the current methods when the input dataset does not contain information about the nodes and edges.

2. MATADOR Database

The Manually Annotated Target and Drug Online Resource (MATADOR) database is a free online database of DTIs[3]. It includes all possible modes of interaction between chemicals and proteins, unlike other resources such as DrugBank which only include the main mode of interaction. These modes of interaction between drugs and targets can be direct, indirect, and in some cases, a combination of the two.

To enable the extraction of information about the DTIs, the creators of MATADOR have developed a one-stop data warehouse *SuperTarget*, which integrates DTIs found in several other databases such as DrugBank, BindingDB and SuperCyp. *SuperTarget* also uses text-mining techniques to retrieve a list of articles in the PubMed database which are relevant to DTI discovery. Manual revision was then performed on the 7000 top-ranking articles. The first release of the *SuperTarget* database contained about 7300 DTIs. Out of the 7300 DTIs, 4900 DTIs were then manually annotated to incorporate additional binding information and indirect interactions, creating the MATADOR database.

The MATADOR database, as of April 2015, contains a total of 15843 entries of DTIs. Each entry contains 13 fields. The important ones for the formation of a DTI network are:

1. Chemical ID: The PubChem compound identifier.
2. Protein ID: The protein identifier, either corresponding to genes from STRING 7 database or from Medical Subject Headings (MeSH)
3. Protein Score: A score for the confidence in the interaction. 950 ("95%") is the (arbitrary) highest value, used for individually derived annotations.
4. MeSH Score: Interactions derived from MeSH terms receive lower scores, depending on the diversity of the protein sequences belonging to the MeSH term.
5. MATADOR Score: The maximum of the two previous scores.

The Chemical ID and Protein ID in each entry of the MATADOR database are required for forming the DTI network. For example, if the Chemical ID is 1 and Protein ID is 50, we connect node 1 in the set of chemicals to node 50 in the set of proteins. The DTI network created is a bipartite graph, which will be further explained in Section 4.1. The MATADOR score allows us to create a weighted version of the DTI network.

3. Related Work

The current methods of link prediction in DTI network are largely based on machine learning. One such method uses bipartite local model, proposed by Bleakley and Yamanishi (2009)[11]. In the bipartite local model, the chemicals and proteins used to form the DTI network are first pre-processed. A similarity score is then calculated between the chemicals based on the chemical structure similarity. Likewise, a normalised Smith-Waterman score is calculated between the proteins which is based on the genomic sequence similarity[14]. This allows the construction of similarity matrix between chemicals and that between proteins

Now, let the set of chemicals be $V_c = \{c_1, c_2, \cdots, c_n\}$ and the set of proteins be $V_p = \{p_1, p_2, \cdots, p_m\}$; the link prediction between a chemical c_i and protein p_j for the bipartite local model works in the following way:

1. exclude p_j from the network, then label the other proteins in the network with $+1$ if they are linked to c_i or -1 if they are not linked to c_i
2. look for a classification rule based on the genomic sequence similarity to distinguish $+1$ labels and -1 labels
3. take this classification rule and predict if there is an edge between c_i and p_j
4. now exclude c_i from the network and label all the chemicals $+1$ or -1 in the network based on whether they are linked to p_j in a similar fashion
5. look for a classification rule based on chemical similarity to distinguish between the labels
6. take this new classification rule and predict if there is an edge between c_i and p_j

This method provides two independent predictions of the same edge between c_i and p_j. A heuristic can then be applied to aggregate the two predictions x and y. The simple heuristic proposed by Bleakley and Yamanishi (2009) was $s = max(x, y)$. The bipartite model was compared against a baseline link prediction model which simply looked at the similarity matrix and found the most similar chemical or protein to the one that is investigated. It was shown that the bipartite local model outperforms the baseline method when evaluated on four different datasets.

Another method of link prediction in DTI network was proposed by Wang and Zeng (2013) [2] and uses restricted Boltzmann machine (RBM) for prediction. A RBM is a two-layer

network consisting of one layer of "visible" units, or observed states; and one layer of "hidden" units, or feature detectors[15]. In the RBM model for DTI prediction, the "visible" unit is encoded as the type of interaction between chemicals and proteins. 2 variables x_{direct} and $x_{indirect}$ are used for the "visible" units. For a direct interaction between the chemical and protein, the variables are set to $x_{direct} = 1$ and $x_{indirect} = 0$. If there is an indirect interaction between the chemical and the protein, the variables are set to $x_{direct} = 0$ and $x_{indirect} = 1$. If there is no link between the chemical and protein, both variables are set to 0. For each protein in the multidimensional DTI network, a corresponding RBM is constructed according to the types of interactions found between the protein and every chemical in the network. At the same time, all the RBMs formed share the same "hidden" parameters. The RBMs are trained by maximising the log-likelihood of the visible parameters. The corrections of the multidimensional network are then used to make predictions. When the chemicals send messages to the "hidden" units, they are updated accordingly. Once the states of the "hidden" units are obtained, they send messages back to the "visible" units and update their corresponding states. This process can be iterated many times to predict all possible interactions between the chemicals and the proteins.

Wang and Zeng (2013) applied the RBM technique on the MATADOR dataset to predict direct and indirect DTI. In particular, they performed three tests:

1. Integrating both direct and indirect DTIs with distinction, the input "visible" unit is a multidimensional vector indicating the mode of interaction

2. Mixing both direct and indirect DTIs without distinction, input "visible" unit is a one-dimensional binary vector indicating whether DTIs are observed

3. Using only a single interaction type

The precision-recall (PR) curve for each test is then calculated as shown in Figure 3.1.

We can see that the RBM approach for DTI prediction produces remarkable results when using both direct and indirect DTIs with distinction. When using only direct or indirect DTIs, the algorithm still produces quite good results. However, when there is no distinction between direct and indirect DTIs, the results are less satisfactory. This particularly concerns indirect DTIs, for which the precision is only about 0.4. This is the shortcoming of the RBM approach; without knowing the attributes of the edges in the network, the RBM method cannot perform satisfactorily.

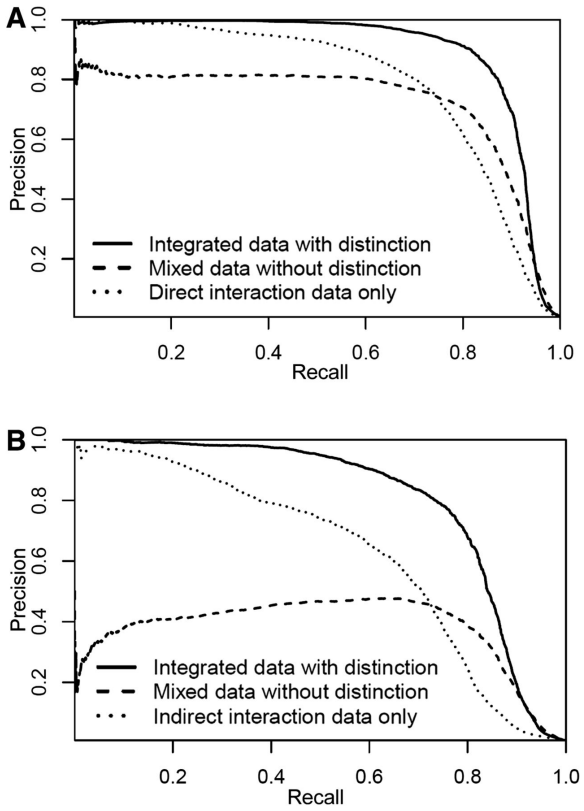


Fig. 3.1 PR curves for **(A)** predicting direct DTIs **(B)** predicting indirect DTIs[2]

Likewise, the bipartite local model requires knowledge about the nodes in the network for calculating the chemical similarity and protein similarity. Without knowing the attributes such as genomic sequence, the bipartite local model cannot be used. Information about the nodes, i.e. proteins and chemicals, often require manual annotation and therefore many DTIs still lack annotation. As mentioned in Chapter 2, only 4900 DTIs out of 7300 DTIs present in the *SuperTarget* database are captured in the MATADOR database. Thus, we developed a new method of using similarity indices for link prediction in DTI network, which will be further explained in Chapter 4.

We compare our method against the baseline method of the RBM model by Wang and Zeng (2013). The authors used MATADOR database for their experiment as well, which allows for direct comparison with our method. Specifically, we aim to tackle the problem of not knowing whether an interaction between chemicals and proteins is direct or indirect, and to improve on their precision levels.

4. Method & Implementation

4.1 Bipartite Graphs

In graph theory, a graph can be represented by $G = (V,E)$ where V is the set of nodes and E is the set of edges. We can picture a graph by using dots as nodes and lines as edges. Figure 4.1 shows a sample graph. In this graph, we have $V = \{1,2,3,4,5\}$ and $E = \{\{1,3\},\{2,4\},\{2,5\},\{3,4\}\}$

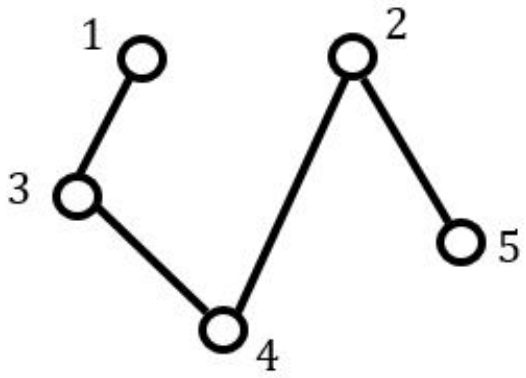


Fig. 4.1 Example of a graph

A common method to represent a graph is an adjacency matrix. An adjacency matrix is a $n \times n$ matrix where n represents the number of nodes in the network and the entries a_{ij} are given by[16]:

$$a_{ij} = \begin{cases} 1 & \text{if there is an edge between } i \text{ and } j \\ 0 & \text{otherwise} \end{cases}$$

The network edges can also be weighted. In this case, instead of $a_{ij} = 1$ when there is an edge between nodes i and j, we use the weight of the edge instead. We can create the adjacency matrix of the network shown in Figure 4.1:

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

However, as mentioned earlier, the DTI network formed from the MATADOR database is a bipartite graph. This means that the set of nodes V can be partitioned into two disjoint sets U and W whereby each edge in the network links one node from U to another node in W and there is no edges between nodes within U or W. In our case, the two sets of nodes are chemicals and proteins. Since there are no interactions within the MATADOR database which depict chemical-chemical interactions or protein-protein interactions, we form a bipartite graph. A simple bipartite graph is shown in Figure 4.2.

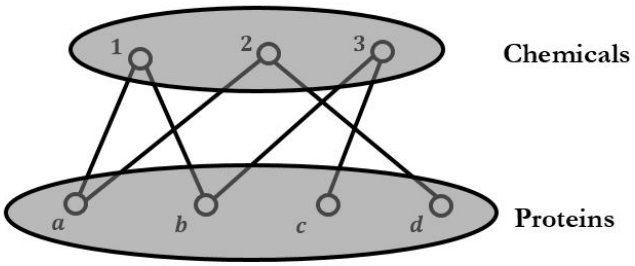


Fig. 4.2 A bipartite graph with 3 chemicals and 4 proteins

We can clearly see that there are two disjoint sets in the bipartite graph in Figure 4.2. In the case of bipartite graph, we can represent the network in the form of a biadjacency matrix. The creation process is similar to the adjacency matrix except that the matrix is $n \times m$, where n represents the number of nodes in set U and m represents the number of nodes in set W. Therefore, the simple bipartite graph shown in Figure 4.2 can be represented by a 4×3 matrix:

$$\boldsymbol{B} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$

The adjacency matrix of a bipartite graph can be created using the biadjacency matrix $\boldsymbol{B}$[16]:

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{0} & \boldsymbol{B} \\ \boldsymbol{B}^{\top} & \boldsymbol{0} \end{pmatrix}$$

The fact that DTI networks are bipartite graphs provides extra challenges to the link prediction problem. Most similarity indices were originally proposed for social networks which are not bipartite in nature and therefore, they do not work for a bipartite graph. For example, in Common Neighbours, if we try to predict a link between chemical c_i and protein p_j, we will

find that the neighbouring nodes of c_i are all proteins and likewise, the neighbouring nodes of p_j are all chemicals. Therefore, the simple Common Neighbours index will give us zero value for all c_i and p_j.

Fortunately, bipartite graphs occur in other networks as well. One example is a user recommendation network where the two disjunct sets are users and recommended items[17]. Therefore, modified similarity indices have been developed to suit bipartite graphs and these modified indices will be further explained in Section 4.2.

4.2 Similarity Indices

Similarity-based algorithms can be considered as the simplest framework for link prediction in networks. As mentioned earlier, they can be calculated using structural similarity which is solely based on the network structure. This ensures that similarity indices can easily be implemented on networks even if no prior knowledge of the nodes is provided. Similarity indices can be classified into local similarity indices and global similarity indices. Local similarity indices are node-dependent, which means that the only information required are the degrees of the node and its nearest neighbourhood. Global similarity indices, on the other hand, are path-dependent and global knowledge of the network topology is required[18].

Similarity indices have proved promising at link prediction in many different networks. Liben-Nowell and Kleinberg (2007) used a variety of similarity indices to predict the evolution of social networks over time[13]. It was shown that many similarity indices, such as Common Neighbours and Jaccard index, vastly outperform a random link predictor, indicating that much information about networks is contained within the network topology alone. Zhou, Lü and Zhang (2009) also applied several similarity indices to 6 different networks and obtained a high level of area under precision-recall curve (AUPR) for the majority of the networks[19]. These results demonstrate the suitability of using similarity indices as a method for link prediction, therefore we believe that it can be similarly applied to DTI networks to identify potential chemical-protein interactions.

For our project, we employed 3 local similarity indices: Common Neighbours, Jaccard Index and Preferential Attachment; and 1 global similarity index: Katz Index and applied them to the MATADOR database for link prediction, with the aim of providing a comprehensive overview of the level of performance of similarity indices for DTI prediction.

4.2.1 Common Neighbours (CN)

For a node x, let $\Gamma(x)$ denote the set of neighbours of x. With this in mind, if two nodes x and y share many common neighbours, a link is likely to exist between these two nodes. A simple measure of this is given by:

$$S_{xy}^{CN} = |\Gamma(x) \cap \Gamma(y)|$$

This essentially counts the number of nodes which have both x and y as their neighbouring nodes. As mentioned previously, in a bipartite network, if we look at chemical x, its neighbours will always be proteins. At the same time, there are no links between proteins and proteins. This means that $|\Gamma(x) \cap \Gamma(y)|$ will always be zero. Hence, we need to modify our definition of CN for a bipartite graph. Now, we define $\widehat{\Gamma}(x) = \bigcup_{c \in \Gamma(x)} \Gamma(c)$ as the set of neighbours of node x's neighbours[20], we can then redefine CN as:

$$S_{xy}^{CN'} = |\Gamma(x) \cap \widehat{\Gamma}(y)|$$

In the original CN index, we basically count the total number of unique paths of length 2 (x to a common neighbour, common neighbour to y). In the modified CN index, we have increased the path length to 3. Thus, we can represent CN in the following way::

$$S_{xy} = \sum_{\substack{z_1 \in \Gamma(x) \cap \Gamma(z_2) \\ z_2 \in \Gamma(z_1) \cap \Gamma(y)}} w(x, z_1) + w(z_1, z_2) + w(z_2, y)$$

For an unweighted network, we simply have all 3 values of w being 1. For a weighted network, we use the weight of each link in the path instead.

4.2.2 Jaccard Index

The Jaccard index is a commonly used similarity metric in information retrieval. For a randomly selected feature f that either node x or node y has, the Jaccard index measures the probability that both nodes x and y have feature f[13]. In our case, the features are the neighbours of the node; therefore we define Jaccard index as:

$$S_{xy}^{Jaccard} = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

Similar to CN, we have to modify the Jaccard index for a bipartite graph[20]:

$$S_{xy}^{Jaccard'} = \frac{|\Gamma(x) \cap \widehat{\Gamma}(y)|}{|\Gamma(x) \cup \widehat{\Gamma}(y)|}$$

The Jaccard index[21] is basically a normalised version of the CN. The aim of the Jaccard index is to ensure that a high score between two nodes x and y is due to the similarity between the nodes instead of their influence. For instance, in a social network, a highly influential individual is naturally well-connected to other individuals in the network. Therefore, it is likely that two highly influential individuals will share many common neighbours even though they are not close friends. As a further example, two celebrities are likely to share a number of mutual fans even if they may not know each other. In this case, they will obtain a high CN score based on their influence in the network. The Jaccard index solves this problem by placing more emphasis on the links of non-influential nodes and less emphasis on the links of highly influential nodes to ensure that the common neighbours that they share are due to their similarity instead of their influence.

For the weighted Jaccard index, we simply take the weighted CN and divide it by the total number of neighbours between nodes x and the neighbouring nodes of y.

4.2.3 Preferential Attachment (PA)

Let k_x be the degree of node x, then the PA between node x and y is defined as[4]:

$$S_{xy}^{PA} = k_x \times k_y$$

PA is based on the phenomenon that nodes with many links tend to generate even more links in the future. This phenomenon can be found in many scenarios. For example, film actors who are well-connected in Hollywood are more likely to acquire new roles in movies which then increase their fame[22]. Likewise, in scientific journals, the most cited articles induce researchers to read them and hence increase their citation numbers. This is known as the Matthew effect[23] where the "rich gets richer".

As PA does not require information about the neighbourhood and is only dependent on the degree of the nodes x and y, it has the lowest computational complexity of all the similarity indices. It also does not require any modification of the bipartite graph. For the weighted PA, instead of using the degree of the node x, we use the sum of the weights between node x and

its neighbours, therefore we have the following index:

$$S_{xy}^{PA} = \sum_{z_1 \in \Gamma(x)} w(x, z_1) \times \sum_{z_2 \in \Gamma(y)} w(z_2, y)$$

4.2.4 Katz Index

The Katz index[24] is a path-dependent global similarity index, which directly sums over the collection of paths in the network and is exponentially damped to give the shorter paths more weight[4]. Let $\boldsymbol{A}$ be the adjacency matrix whereby $a_{xy} = 1$ if x is connected to y, else $a_{xy} = 0$; The Katz index can be defined as:

$$s_{xy}^{Katz} = \beta \boldsymbol{A}_{xy} + \beta^2 (\boldsymbol{A}^2)_{xy} + \beta^3 (\boldsymbol{A}^3)_{xy} + \ldots$$

The whole similarity matrix can be written as:

$$S^{Katz} = (I - \beta \boldsymbol{A})^{-1} - I$$

The damping factor β controls the path weights. A small β value means that longer paths contribute less to the Katz index score and vice versa. This means that small β values generate results which are similar to CN. To ensure that the Katz index converge, the value of β must be less than the reciprocal of the largest eignevalue of $\boldsymbol{A}$. Since the Katz index requires the calculation of an inverse of the matrix, we use the adjacency matrix $\boldsymbol{A}$ created by the biadjacency matrix $\boldsymbol{B}$ using the method explained in Section 4.1. This works for both weighted and unweighted biadjacency matrix $\boldsymbol{B}$.

4.3 Implementation

The programming language and development environment which we used for our experiment and analysis of results is MATLAB R2014b. MATLAB was chosen as in our experiments, we are representing DTI networks in the form of matrices. MATLAB has its own built-in libraries for manipulating and calculating matrices, which is helpful in our experiments. For example, in calculating the Katz index, the function `eye` in MATLAB generates an identity matrix and the function `inv` calculates the inverse of the matrix.

Moreover, MATLAB supports the development of applications with graphical user interfaces (GUI) via graph-plotting tools. The function `plot` can plot a graph from two input vectors x and y. This allows us to plot the precision-recall (PR) curve of the results obtained after ap-

plying the similarity indices and to observe the trends in the curve and compare the different indices in terms of performance.

The entries of the MATADOR database are contained within an Excel file. This is first read into MATLAB using the command `xlsread`. Following that, we can form the biadjacency matrix using the various fields of the MATADOR entries as explained in Chapter 2. For our experiment, we created 2 biadjacency matrices, an unweighted matrix $\boldsymbol{B_u}$ and a weighted matrix $\boldsymbol{B_w}$. For $\boldsymbol{B_u}$, we have $b_{xy} = 1$ if there is a interaction between protein x and chemical y, else we have $b_{xy} = 0$. For $\boldsymbol{B_w}$, we take the MATADOR score of the link between x and y instead and normalise it against the maximum MATADOR score of 950. In doing so, we obtain a 2901×801 matrix $\boldsymbol{B}$ with 15843 non-zero entries.

4.3.1 10-Fold Cross Validation

Cross validation is a common technique for assessing whether the results of the similarity indices will generalise to any independent dataset. One of the most popular cross validation methods is the k-fold cross validation. This means that the input dataset is partitioned into k sub-datasets of approximately equal size. The experiment is then performed k times where each time, one out of the k sub-datasets is used as validation data and the other $k-1$ sub-datasets are used as training data. The k results can then be averaged to obtain a single result.

For our experiments, we used 10-fold cross validation, which is known to give the lowest bias and variance in the sub-datasets[25]. Hence, the entries within the MATADOR database are randomly divided into 10 non-overlapping subsets of approximately equal size in terms of the number of DTIs. Following that, we create a biadjacency matrix for each of the 10 subsets. For each similarity index, we apply the algorithm to the sum of 9 biadjacency matrices and use the remaining biadjacency matrix as the validation data to check if the links that we predicted actually exist. This process is then repeated 10 times using a different biadjacency matrix as the validation data. The precision and recall calculated from the 10 iterations are averaged to obtain a single result.

4.3.2 Precision-Recall Curve

Precision is defined as:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

and recall as:

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

In our case of link prediction, true positive (TP) refers to the links predicted using the training data that are found in the validation data. False positive (FP) refers to the links predicted using the training data that are not found in the validation data. False negative (FN) refers to the links that are not predicted using the training data but are found in the validation data.

In our experiments, after applying a similarity index to the training data, we ranked the links predicted according to their scores. Then, we took the top n links predicted and calculated the precision and recall based on the n links. This was repeated for $n = 1$ to 10000. With this, we obtained 10000 points at different precision and recall values. This is then averaged for the 10 iterations of the different training data for each similarity index. We could then plot the PR curve for each similarity index and compare their performance.

5. Results & Discussions

5.1 Pilot Experiment

Before applying the similarity indices to the training data, we first conducted a pilot experiment on one of the validation datasets. Assuming two nodes x and y share a link in the validation dataset, we would like to find out the path length required to move between x and y using the links in the training dataset. For example, as shown in Figure 3.1, to reach from node x to node y (a link present in the validation dataset), we need to move from x to a, a to b and finally b to y. This shows that a path length of 3 is required using the links in the training dataset to reach the link in the validation dataset.

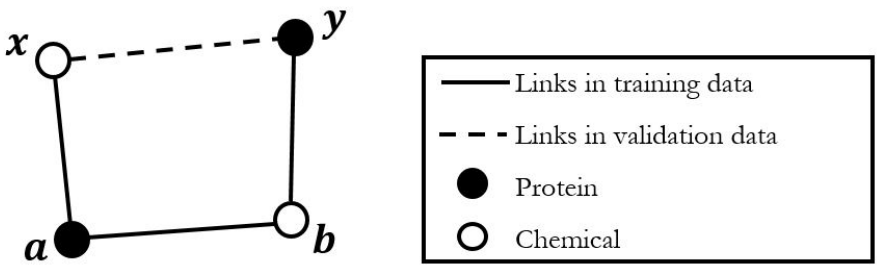


Fig. 5.1 Example of a link in validation data that requires 3 links in training data to reach

The aim of this pilot experiment was to determine the average path length required in the training data to reach the links in the validation data. Since the input dataset is partitioned into training dataset and validation dataset, we know that the links within the validation dataset will not be found in the training dataset. Therefore, to reach a link found in the validation dataset, we need a minimum of 3 links, i.e. a path length of 3, in the training dataset as shown in Figure 5.1 due to the bipartite nature of the graph. These links in the validation dataset which require the minimum of path length 3 will be assigned a higher score using our similarity indices, as they have common neighbours between them in the training dataset. For example, in Figure 5.1 the link between the pair of nodes x and y will be assigned a higher score as they have common neighbours a and b between them. Thus, a lower average path length justifies our method of using similarity indices for DTI link prediction.

Out of the 1614 links within the validation dataset which we used for the pilot experiment, we found that 1477 links required only a path length of 3 to reach using the training dataset. This shows that a large majority of the links in the validation dataset have common neighbours

in the training dataset, therefore they will be assigned higher score using similarity indices. However, we also found 119 links with a path length of 13, which is the maximum path length that we set for our search. This means that our search algorithm was unable to find a path between these 119 pairs of nodes using the links in the training dataset. The reason for this could be due to the formation of disconnected sub-networks in the training dataset, as shown in Figure 5.2.

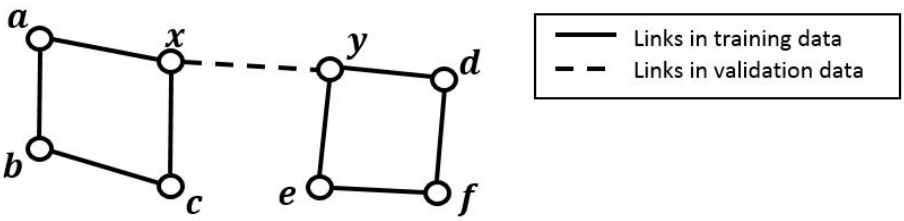


Fig. 5.2 Removing link between x and y creates 2 disconnected sub-networks

In Figure 5.2, by removing a link between node x and y from the input dataset, we have created two disconnected sub-networks in the training dataset. Therefore, it is impossible to reach from x to y using the links in the training set. This shows one of the drawbacks of using similarity indices for link prediction: it is difficult to achieve 100% recall. This is due to the fact that similarity indices are entirely based on network topology and not on the attributes of the individual nodes. Therefore, it cannot predict a link between disconnected networks.

5.2 Common Neighbours (CN)

We first look at the results of using CN, which is the simplest form of node-dependent similarity index. The experiment was performed for both weighted biadjacency matrix $\boldsymbol{B_w}$ and unweighted biadjacency matrix $\boldsymbol{B_u}$ created from the MATADOR database. 10-fold cross validation was performed on the input dataset and the results of the 10 iterations are averaged to obtain the mean precision and recall. Figure 5.3 shows the PR curve of applying CN to both $\boldsymbol{B_u}$ and $\boldsymbol{B_w}$.

We can see that using CN, we obtain high precision when the recall is low. However, when recall reaches around 0.3, precision deteriorates drastically. This is most likely because there are many potential links between nodes in $\boldsymbol{B_w}$ and $\boldsymbol{B_u}$ with common neighbours. Hence, although CN manages to predict the correct links within the network, it also predicts many non-existent links. Moreover, we can see that the PR curve tapers off at recall rate of 0.6. As shown earlier, CN is unable to predict the links in the validation data which require a path

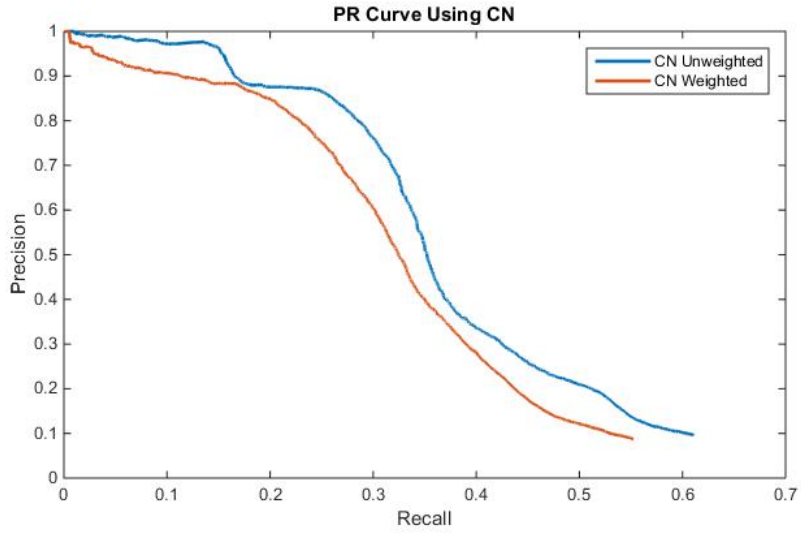


Fig. 5.3 PR Curve using CN

length of more than 3. Therefore, we cannot achieve a recall rate of 1.0. In our pilot test, we have shown that out of the 1614 links in validation dataset 1, only 1477 links can be reached in 3 steps. Therefore, the maximum recall that we can obtain is 0.915. Since we are looking at only the top 10000 links predicted, the recall obtained is lower than the maximum value.

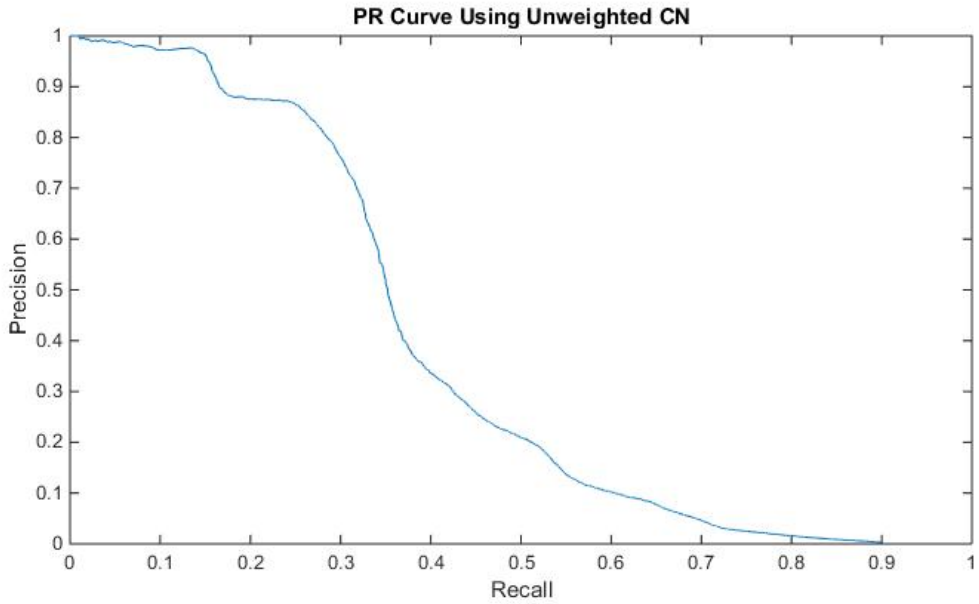


Fig. 5.4 Unweighted CN using top 500000 links

In Figure 5.4, we show the unweighted CN using top 500000 links instead of top 10000 links as our threshold. In unweighted CN, each fold of the experiment predicts around 580000 links. Hence, we used most of the links predicted by the unweighted CN to calculate the PR curve. We can see that the recall rate has reached 0.9, which is close to the maximum recall of 0.915. However, we notice that at 0.6 recall, the precision rate has already fallen to 0.1, meaning that for every true positive link predicted, the algorithm predicts 9 false positive links. We must keep in mind that link prediction in the DTI network is meant to guide researchers to find potential DTIs. It is not feasible to carry out research on all the links generated when most are incorrectly predicted. Therefore, considering the top 10000 links for the PR curve calculation seems suitable.

One thing visible from Figure 5.3 is that unweighted CN actually has a better performance than weighted CN. This is surprising as we would presume that by assigning a weight on the edges, we can predict the links more accurately. However, it seems that the weights on the links actually deteriorate the performance of CN. This is likely due to the weak-tie theory in network analysis, which will be further explained in Section 5.6.

Another observation from Figure 5.3 is that the PR curve for the unweighted CN is smoother than that for the weighted CN. Most noticeably, there is a sharp drop in performance for unweighted CN at recall of 0.15. This is probably because in the DTI network, there might be several pairs of nodes with the same number of common neighbours. For example there might be 10 paths with length of 3 between chemical c_1 and protein p_1 and between chemical c_2 and protein p_2. Since the paths are unweighted, $S_{1,1}$ and $S_{2,2}$ will have equal scores. However, it is possible that in the validation data, there is only a link between c_1 and p_1 but not between c_2 and p_2. Therefore, the equal scores given to pairs of nodes with the same number of common neighbours causes sharp a performance drop when some of these links with the same score are not found in the validation dataset. Weighted CN, on the other hand, uses the weight of the paths to calculate a score and therefore, it is unlikely that multiple pairs of nodes have equivalent scores. This leads to a much smoother PR curve.

5.3 Jaccard Index

Next, we apply Jaccard index to the 10 folds of input datasets for both weighted adjacency matrix $\boldsymbol{B_w}$ and unweighted adjacency matrix $\boldsymbol{B_u}$. Similar to CN, we average the 10 sets of results and calculate the average precision and recall to plot the PR curve for Jaccard index,

shown in Figure 5.5.

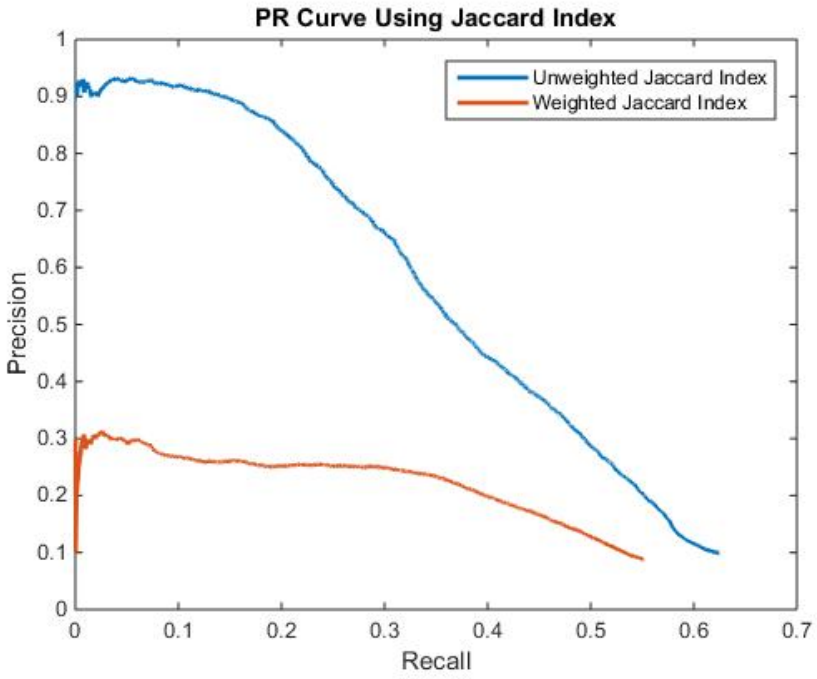


Fig. 5.5 PR curve using Jaccard index

We notice from Figure 5.5 that unweighted Jaccard index outperforms weighted Jaccard index by a large margin. This again can be partly explained by the weak-tie theory occurring in the DTI network. Another reason is due to how weighted Jaccard index is calculated. As mentioned in Section 4.2.2, weighted Jaccard index takes S_{xy}^{CN} for the nodes x and y and normalises it with the total number of neighbours of x and y. By dividing weighted CN by the total number of neighbours, we are not taking into account the weight of the link between the node and its neighbours. In Section 4.3, we explained that the maximum weight of the links in the weighted network is 1, as the links are normalised by the maximum MATADOR score. Therefore, pairs of nodes with large number of neighbours are affected by a larger margin than pairs of nodes with small number of neighbours.

To understand more about the effects of the normalising factor in weighted Jaccard index, we can compare the results of weighted Jaccard index to weighted CN. By looking at the dataset, we found that in weighted CN, the predicted link with the highest score is between chemical number 39 and protein number 241 with a score of 1213. However, using weighted Jaccard index the score is only 1.3521, which means that the total number of neighbours between chemical 39 and protein 241 is 897. On the other hand, the highest score using weighted

Jaccard is 5.2859, between chemical 300 and protein 2213. However in CN, the score is only 79.29. This means that there are only 15 neighours between chemical 300 and protein 2213. Therefore, it seems that in weighted Jaccard index, the normalising factor is causing overcompensation in the score calculated causing it to have a much worse performance than the unweighted Jaccard index and the weighted CN index.

Another thing that we notice with Jaccard index is that unlike CN, the PR curve for unweighted Jaccard index is smooth, without a sudden drop in performance. This is also due to the normalising factor. Since the total number of neighbours between pairs of nodes varies, even when two pairs of nodes have the same number of common neighbours, their Jaccard index will be different. Therefore, it will not create the situation where there are many predicted links with the same score, causing sudden decrease in performance when many links predicted at that score are not found in the validation set.

5.4 Preferential Attachment (PA)

Next, we apply PA to the biadjacency matrices $\boldsymbol{B_u}$ and $\boldsymbol{B_w}$. The PR curve of the results are shown below in Figure 5.5

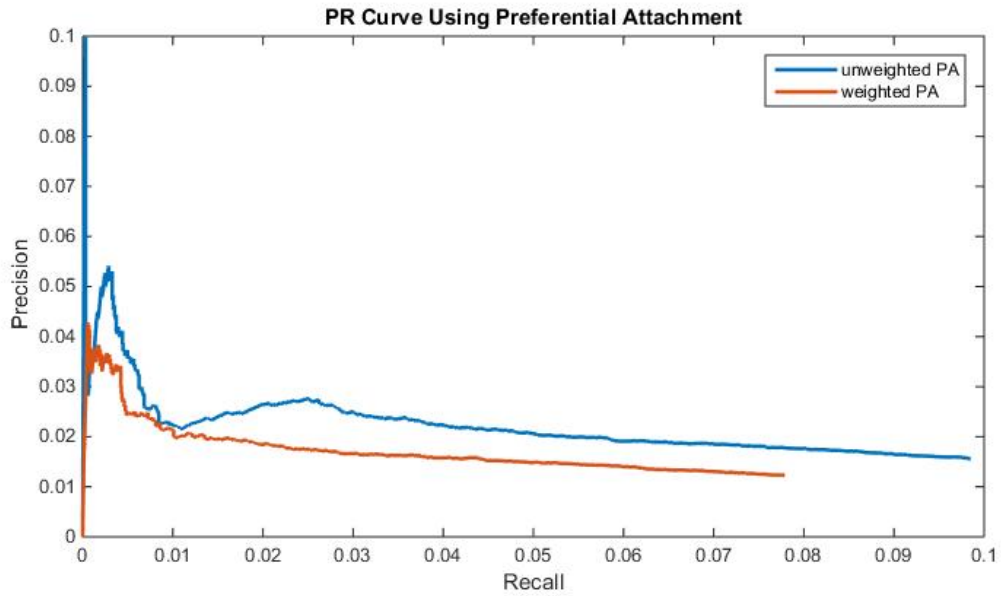


Fig. 5.6 PR curve using PA

Again, the unweighted PA outperforms the weighted PA, likely due to the effect of weak-ties. As we can see, since there is no normalising factor in PA, the performance of the weighted and unweighted PA do not differ significantly. At the same time, the performance of PA is very poor. Using the top 10000 predicted links to calculate the PR curve, we only obtain recall of 0.1. Since there are roughly 1600 links in each validation dataset, this means that only 160 links out of the 10000 predicted actually exist in the validation set. This is also seen from the low precision level of around 0.02 for larger recall levels. This means that for every true positive link that is predicted, the PA method also predicts 49 false positive links.

The poor performance of PA is probably due to the PA being originally based on the influence level of individuals in social networks. As previously mentioned in Section 4.2.3, PA is based on the Matthew effect. For example, a new member of a social network is much more likely to know about an influential member of the network than about a non-influential member. Therefore, the influence of a person in the network further increases his influence level. However, this is not the case in a DTI network. If a protein interacts with many different chemicals, it does not make the protein more likely to interact with the next chemical. Therefore, the underlying assumption of PA does not work in a DTI network causing it to have poor performance.

Also, the PR curve shows that there are some fluctuations in precision at very low recall (<0.01) while at higher recall, the fluctuations in precision disappears. If we look at PA as a random link predictor, then for the first few links predicted, each true positive link affects the precision by a large margin. For example, if there is 1 true positive link predicted in the first 10 links, the precision will be affected by 0.1. Therefore, large fluctuations appear at very low recall. However, when the number of links predicted increases, each correct link only affects the precision by a small margin, thus the fluctuations disappear and the precision remains at a constant level.

5.5 Katz Index

Finally, we apply Katz index to the 10 fold of input datasets. For Katz index, to find out the effect of the damping factor β on link prediction, we used 3 different β values for our calculations. To ensure that the Katz index converges, we require the β value to be less than the reciprocal of the largest eigenvalue of the adjacency matrix $\boldsymbol{A}$[4]. We calculated that the maximum value which β can take is about 0.022 for the 10 folds of input. Hence, we chose the 3 values of beta for the experiment to be 0.02, 0.01 and 0.005 respectively. Figure

5.7 shows the PR curves for the weighted and unweighted Katz index for different values of β.

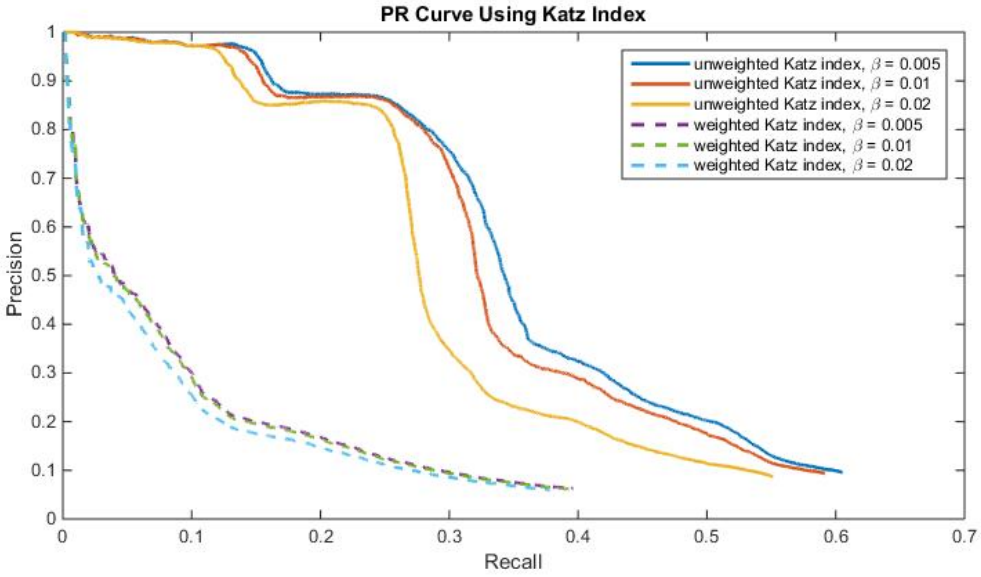


Fig. 5.7 PR curve using Katz index at different β values

The solid lines in Figure 5.7 represent unweighted Katz index at different β values while the dotted lines represent weighted Katz index at different β values. We observe that the unweighted Katz index vastly outperforms the weighted Katz index. Since Katz index is a path-dependent similarity index which looks at the entire network topology, weak-tie effects are more pronounced as compared to local similarity indices such as CN. This might account for the large difference in performance between weighted and unweighted Katz index.

The PR curve for the unweighted Katz index has a similar shape to CN. This is expected as Katz index is basically an extension of CN, obtained by considering paths with length longer than 3 in the network. However, it is surprising that weighted Katz index does not look similar to the weighted CN. The weighted Katz index does not perform as well as the weighted CN. This seems to suggest that taking into account paths that are longer than 3 negatively affect the precision of link prediction in a weighted network.

We also looked at the impact of different β values on the PR curve. In Figure 5.7, we can clearly see that for the unweighted Katz index, a smaller β value leads to a better performance than a larger β value. This concurs with the results of our pilot experiment, which shows that 91.5% of the links in validation dataset 1 can be reached in 3 steps using the links in the training dataset 1. Therefore, by placing more emphasis on shorter paths, we generate better

results. For weighted Katz index, β value of 0.01 and 0.005 seems to give similar results which are better than the results when using a β value of 0.02. This again shows that lower β value favouring shorter paths generates better results. Another observation that can be made from Figure 5.7 is the difference in results using β value of 0.02 and 0.01 is much larger than the difference in results using β value of 0.01 and 0.005. This seems to indicate that β value affects the performance of Katz index in the DTI link prediction exponentially as its value increases. However, more experiments using a larger range of β values need to be performed to fully confirm this.

5.6 Weak-Tie Theory

The weak-tie theory, first proposed by Granovetter in 1973 [26], states that in a social network, a person obtains more information from his acquaintances than from his close friends. This is due to clusters that form within social networks, illustrated in Figure 5.8. The weak-tie theory assumes that a person in the social network, such as P_1, and his close friends form a high-density sub-network, or a cluster, in the social network. On the other hand, P_1 and his acquaintances form a low density network, such as from P_1 to P_2. At the same time, P_2 has his own close friends and therefore, his own cluster as well. Thus, the weak-tie between P_1 and P_2 is not a trivial relationship but a crucial bridge between different clusters that allows information to travel to distant parts of the social system[27]. This means that in a weighted network, the edges with lower weight are often as important as the edges with higher weight. Therefore, the lower weights that are assigned to these weak ties are actually undervaluing the importance of weak ties in the network.

Although the weak-tie theory is based on social networks, it seems to apply to biological and other types of networks as well. In the study of brain networks, a basic conundrum that occurs is that the networks exhibit strong modular and hierarchical structure while being highly efficient at the same time[29]. High modularity and hierarchical structure require the modules to be isolated and independent from each other. However, the efficiency of information transfer within the brain also demonstrates that the network is a small world, which mean that large local clustering and many short paths exist within the brain network, going against the strong modularity observation. Gallos, Makse and Sigman (2012) found that by incorporating weaker ties into the brain network, it is possible to retain a highly modular structure while at the same time, converts the network into a small world, solving the conundrum that exists. Csermely (2004) also demonstrated that weak ties are just as

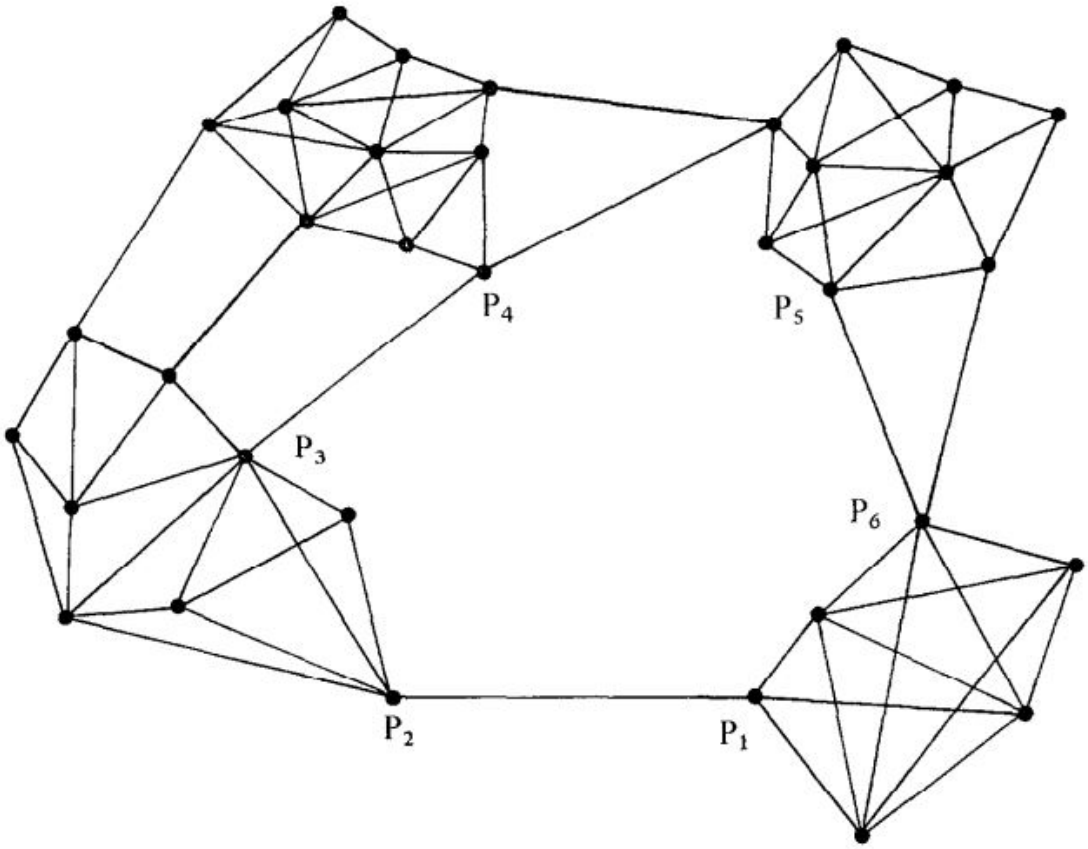


Fig. 5.8 A social network with 4 clusters[28]

important as strong ties in the metabolism network for *E. Coli*[30]. Therefore, the weak-tie theory is not limited to only social network but can be prevalent in other forms of network as well.

To investigate further the role of the weak-tie theory in our experiments, we looked at how the MATADOR database assigns scores to the DTIs. In the extraction of DTI to form the MATADOR database, synonyms of drugs, proteins and Medical Subject Headings (MeSH) for groups of proteins are used as keywords to search for potential DTIs in the Medline database of papers related to biomedical science[3]. Some of the DTIs extracted do not show chemicals interacting with specific proteins, but rather with protein families or protein complexes. The chemical might interact with an unknown specific member of the protein family or in the case of protein complexes, interact with more than one of the sub-units. This will often result in lower scores for the DTI due to uncertainty, depending on the size and nature of the protein families. A more heterogeneous family will results in lower confidence level and hence, lower MATADOR score. Looking at our results, it seems to indicate that this

method of assigning MATADOR score is undervaluing some of the links in the database. It is possible that some links with lower confidence level are actually correct, therefore assigning them lower scores causes the weighted link prediction method to perform worse.

5.7 Comparison Between Methods

After analysing the performance of the individual similarity indices, we compare these methods against each other with the RBM approach as our benchmark. As the RBM approach separates the links predicted into direct and indirect links, we cannot perform a direct comparison between our methods. Fortunately, Dr Wang Yuhao and Dr Michael Zeng, the developers of the RBM method, were willing to share their dataset with us. By combining the precision and recall of their results for predicting direct links and indirect links, we were able to plot a combined PR curve for their experiment.

To obtain a fair comparison with their experiment, we focus on cases where no distinction between the links in the DTI network is made and therefore, we use only the unweighted version of the similarity indices. Figure 5.8 shows the PR curve for the different methods. Note that for the Katz index, we chose β value to be 0.005 as it has the best performance out of the 3 different β values.

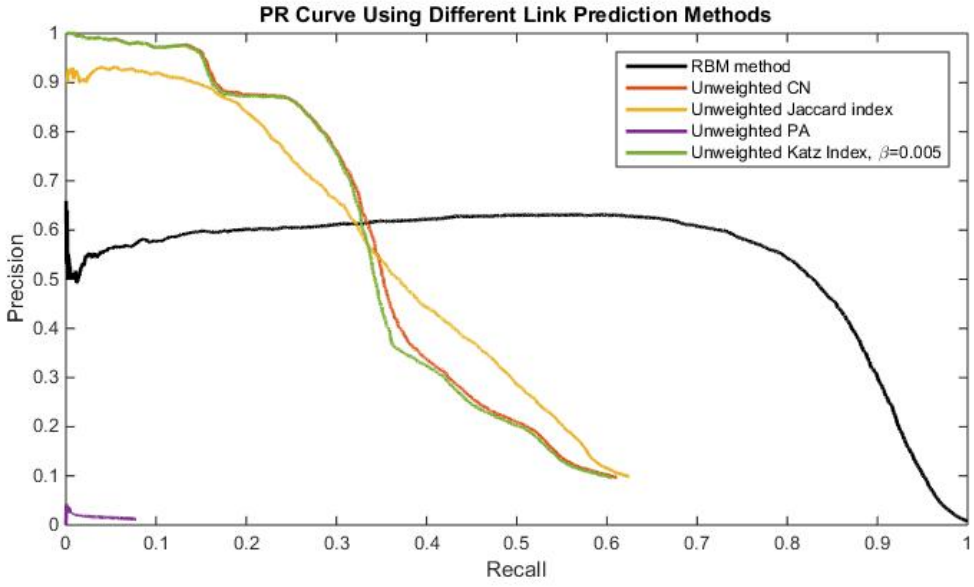


Fig. 5.9 A comparison of the PR curves using different link prediction methods

As seen from Figure 5.9, the three similarity indices: CN, Jaccard index and Katz index, outperform the RBM approach at low recall. However, PA shows the lowest performance out of the 5 different methods. As mentioned in Section 5.5, the assumption of PA does not hold in a DTI network. Hence, we show that PA is not a suitable link prediction method to be used for a DTI network. The other 3 similarity indices have a higher precision than the RBM method for recall between 0 to 0.35. Interestingly, the 3 different similarity indices intersect the RBM method at almost the same point.

Link prediction of DTIs is meant to guide researchers to finding potential interactions between chemicals and proteins. The confirmation of these potential DTIs using laboratory experiments cost time and money, therefore it is unlikely that institutes and pharmaceutical companies are able to test out all the potential DTIs identified from the link prediction methods. Hence, it is arguably more important for a link prediction method to have higher precision at low recall than at high recall. As all 3 similarity indices perform better than the RBM method for recall between 0 to 0.35 and the typical number of links in a validation dataset is around 1600, this means that for the first 560 correct links predicted, the similarity indices outperform the RBM method. Therefore, we have demonstrated our new method of using similarity indices is more suitable for pharmaceutical researchers to use than the RBM method when we do not have information about the modes of interaction between chemicals and proteins.

Now, let us look at the trends of PR curve of the various similarity indices. First, we notice that Katz index and CN have very similar PR curves. This is because we are using a very small value of β, which gives more weight to shorter paths and less weight to longer ones. Thus, the Katz index functions almost like CN, producing a similar PR curve. We assume that the Katz index should have a larger maximum recall than CN, as it is able to predict links in the validation dataset that require more than 3 steps of the links in the training dataset to reach. Indeed, when we look at the total number of links predicted by each link prediction method, CN predicts around 580000 links while Katz index predicts about 1.85 million links. As this is more than 3 times the number predicted by CN, Katz index will have a larger maximum recall. However, looking at the PR curve for both methods, the maximum recall the two indices is about 0.6. This shows that the top 10000 links predicted are not actually affected by the longer paths incorporated in Katz index. If we look at the PR curves closely, CN has the best performance out of the 3 similarity indices at low recall. This is unexpected as CN is the simplest index, and intuition tells us that the more complex indices should have better performance as they have factored in more features of the network. However, this

is not the case with the DTI network. In fact, this has also been observed by various other researchers. Zhou, Lü and Zhang (2009) used 9 different similarity indices for link prediction on 6 different networks and found that CN performed the best for all the datasets[19]. This shows that the CN actually is a very strong link prediction method despite its simplicity.

In Figure 5.9, we compare the Jaccard index to the CN and Katz indices. We see that the Jaccard index has a smoother curve than the other two. In CN and Katz index, we can see that there is a sharp drop in precision at about 0.15 recall and also between 0.3 to 0.35 recall. As mentioned in Section 5.2, this is mainly due to the large number of links having equal score because of having the same number of common neighbours. In Jaccard index, we see that the normalising factor produces a much smoother curve. At the same time, this also causes the precision of Jaccard index to decrease at a slower rate. From Figure 5.9, we can clearly see that when recall is below 0.35, Jaccard index does not perform as well as CN and Katz index. However, when recall is above 0.35, the Jaccard index outperforms CN and Katz index. Yet, as explained above, it is more important to have higher precision at low recall level than at high recall level. Therefore, Katz index and CN are better link prediction methods than Jaccard index. Moreover, if we are interested in having good precision at high recall, it is better to simply use the RBM method which significantly outperforms any of the similarity indices at high recall.

In conclusion, comparing the different link prediction methods shows that the similarity indices do have their own distinct advantages over the RBM method when the modes of interaction are not labelled as direct or indirect. This is particularly true at low recall (<0.35) when the similarity indices have precision levels higher than the RBM method. In *in silico* DTI prediction, our aim is to identify potential links for scientists to carry out experiments and to verify the existence of these interactions. Since valuable resources such as time, manpower and money are required to validate potential interactions, it is arguably more important for the precision to be high at lower recall than higher recall. Therefore, we believe that our method of using similarity indices is more suitable for our task of DTI link prediction than the RBM approach when the input dataset is not annotated.

6. Conclusion

In this project, we have developed a new method for DTI prediction based on similarity indices. We found that previous methods for DTI prediction require the algorithm to know about the characteristics of the nodes and edges to function satisfactorily. However, this is not practical in many real life scenarios. For example, we have seen that the MATADOR database requires manual annotation to identify the modes of interaction between chemicals and proteins and as a result, only 4900 out of the top 7000 links from the *SuperTarget* database are captured in the MATADOR database. Our method of using similarity indices does not rely on any information about the nodes and edges but is based entirely on network topology, i.e. how the various nodes in the network are connected to each other. This ensures that our algorithm performs up to standard even if the input dataset does not contain information about the nodes and edges.

We applied four different similarity indices to the MATADOR database and compared the results to the state-of-the-art method, the RBM approach. In particular, we aimed to improve on the results of the RBM approach when minimum information about the network is available. Therefore, we used the unweighted version of the similarity indices and compared it against the RBM method when the modes of interaction between chemicals and proteins are not distinguished.

We showed that CN, Jaccard index and Katz index has higher precision than RBM for recall less than 0.35. However, the RBM method has better performance when recall is larger than 0.35. It is important to remember that the goal of *in silico* DTI prediction is to identify the DTIs which have the highest probability of being correct. This allows the scientists to experimentally confirm their existence. Since scientists will start working from the highest ranking predictions made by the algorithms, we would like the precision to be as high as possible at low recall values, so as to reduce cost of research and to ensure faster discoveries of new DTIs. Therefore, the 3 similarity indices are more suitable for DTI link prediction than the RBM approach. Moreover, new DTIs confirmed by scientists provide more information about the DTI network, which in turn allows us to predict potential DTIs at a higher precision. This creates a positive feedback loop which is beneficial for the drug discovery process. On the other hand, PA performs significantly worse than other methods for all recall values. This is due to PA being based on the Matthew's effect, which is a wrong assumption in a DTI network.

Our experiments also led to some interesting results. Firstly, we showed that the unweighted version of the similarity indices actually outperforms the weighted version. This could be attributed to the weak-tie theory, which states that in a social network, people often obtain their information from acquaintances rather than from close friends. The weak-tie theory is also not limited to the area of social networks but manifests in biological networks as well, such as in brain networks and in the metabolism network for *E. Coli*. We believe that this phenomenon in a DTI network can be attributed to the weak-tie theory as well. Another point of interest is that CN, our simplest similarity index, actually has better performance than the other indices. This is also observed by other researchers, which seems to indicate that taking too many factors into consideration might hurt the accuracy of link prediction due to overcompensation.

At the same time, similarity indices are not perfect. One main weakness is that it cannot retrieve all the potential links in the network. This is because a pair of potentially connected nodes might belong to different disconnected subnetworks, thus there are no paths connecting them in the current network. Based on network topology alone, we cannot predict the existence of such links. Another weakness of similarity indices is that we cannot predict the links of a new node that appears in the network. Assuming that a new chemical x is manufactured, it will naturally have no links within the DTI network. Therefore, all the similarity indices will generate a score of zero for the predicted links between x and the proteins.

In conclusion, our new approach of using similarity indices is a better method for DTI link prediction than traditional machine learning methods when information about the nodes and edges are not available. However, when these information are available, it is better to use traditional machine learning method such as the RBM method. Thus, our new approach is not a competing method but rather a complementary method. These different methods are all aimed to provide better screening for potential DTI discoveries which will improve the completeness of the DTI network, leading to better predictions. We hope for this positive feedback loop in the process of DTI prediction in the future, enabling us to find the cure for all our diseases.

7. Future Work

In the future, this project could be expanded in various ways. We discuss a number of ideas in the sections below.

7.1 Additional Datasets

In our experiment, we focused on the MATADOR database. It would be interesting to apply similarity indices on a different database of DTIs, such as the TTD or DrugBank. Although based on the 10-fold cross validation technique, the similarity indices should generate similar performance for independent datasets, it would still be worth looking at the PR curves generated and compare them against the results on the MATADOR database to see if there are any differences in characteristics of the different DTI networks formed.

Another interesting idea would be to look at the time evolution of the DTI network. In our experiments, we basically took a snapshot of the DTI network frozen in time. But we can look at the change in DTI networks over time, such as new interactions between chemicals and proteins being discovered by researchers. This idea was explored by Liben-Nowell and Kleinberg (2007) on social networks where they used a different time intervals for training and test datasets[13]. Although their method looks at a continuous time period of network formation and tries to predict links that will form in the next time period, we could modify and simply the method for DTI networks. We can define two timestamps t_0 and t_1 where $t_1 > t_0$ and generate the DTI network for all the interactions that are documented between chemicals and proteins at time t_0 and use this as the training data. We can then apply link prediction to training data and predict interactions that might occur at time t_1. Finally, we can use the interactions that were documented at time t_1 as validation data to see if the potential interactions predicted actually exist. This might show us different behaviours that are not discovered when we look at a specific timestamp.

7.2 Strength Of Weak-Ties

Another interesting investigation would be the strength of the weak ties within the DTI network. From our experiments, we have already demonstrated that the weak ties play a significant role in the network and assigning these weak ties a lower score than for strong ties leads to less accurate predictions. However, we do not know just how significant the

weak-ties are. A simple way of looking at the strength of weak ties is to modify the weighted CN which we used. In weighted CN, we have :

$$S_{xy} = \sum_{\substack{z_1 \in \Gamma(x) \cap \Gamma(z_2) \\ z_2 \in \Gamma(z_1) \cap \Gamma(y)}} w(x,z_1) + w(z_1,z_2) + w(z_2,y)$$

where $w(x,y)$ represents the weight of the link between nodes x and y. We can add in a free parameter α to control the relative contribution of the weak ties to the similarity indices[31]:

$$S_{xy} = \sum_{\substack{z_1 \in \Gamma(x) \cap \Gamma(z_2) \\ z_2 \in \Gamma(z_1) \cap \Gamma(y)}} w(x,z_1)^{\alpha} + w(z_1,z_2)^{\alpha} + w(z_2,y)^{\alpha}$$

When we have $\alpha = 1$, we simply have weighted CN. When $\alpha = 0$, we get unweighted CN. Therefore, we can use a range of different α values for our link prediction and look at the performance using different α values. The smaller α is, the stronger the role of weak-ties in the network. In fact, Lü and Zhou (2010) found that in certain networks, the optimal value of α is negative, showing that the weak ties in the network are actually more important than the stronger ties. Thus, it is possible that assigning a much higher weight to weak ties in the DTI network leads to better performance.

7.3 Literature-Based Discovery

Finally, we could look at other applications of similarity indices beyond the area of *in silico* DTI prediction. Another area in biomedical science that could benefit from similarity indices is literature-based discovery. Due to the large amount of publications in the field of biomedical science in recent years, it has become impossible for researchers to read through all the articles that are relevant to his or her field of research. This has led to many potential knowledge and discoveries being left undiscovered for many years. One way of solving this problem is literature-based discovery. This was first proposed by Swanson (1986) when he found links between seemingly disparate ideas of dietary fish oil and Raynaud's syndrome which have never been previously mentioned in the same paper[32]. Dietary fish oil was known to reduce blood lipids, platelet aggregability, blood viscosity, and vascular reactivity. On the other hand, Reynaud's syndrome is a circulatory disorder that is associated with and exacerbated by high platelet aggregability, high blood viscosity, and vasoconstriction. Therefore, Swanson hypothesised that dietary fish oil can alleviate Raynaud's disease – a

hypothesis that was later proven to be true[33]. This shows the potential of literature-based discovery and has been coined the ABC method, whereby if A is linked to B and B is linked to C, then it is possible that A is linked to C.

One problem that occurs in literature-based discovery is the lack of objective literature-based interestingness measures[34], i.e. a measure of the strength of links between the different concepts in biomedical science. This problem is similar to the problem of *in silico* DTI prediction and we believe that similarity indices could potentially lead to good results in literature-based discovery as well. In fact, the ABC method proposed by Swanson is in essence the same as CN. It is possible that other indices such as Katz index will perform better in the concept network of biomedical terms. This could lead to novel scientific discoveries and is worth investigating in the near future.

References

[1] Y. Yamanishi, M. Araki, A. Gutteridge, W. Honda, and M. Kanehisa, "Prediction of drug–target interaction networks from the integration of chemical and genomic spaces," *Bioinformatics*, vol. 24, no. 13, pp. i232–i240, 2008.

[2] Y. Wang and J. Zeng, "Predicting drug-target interactions using restricted boltzmann machines," *Bioinformatics*, vol. 29, no. 13, pp. i126–i134, 2013.

[3] S. Günther, M. Kuhn, M. Dunkel, M. Campillos, C. Senger, E. Petsalaki, J. Ahmed, E. G. Urdiales, A. Gewiess, L. J. Jensen, *et al.*, "Supertarget and matador: resources for exploring drug-target relationships," *Nucleic acids research*, vol. 36, no. suppl 1, pp. D919–D922, 2008.

[4] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 6, pp. 1150–1170, 2011.

[5] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[6] L. Getoor and C. P. Diehl, "Link mining: a survey," *ACM SIGKDD Explorations Newsletter*, vol. 7, no. 2, pp. 3–12, 2005.

[7] R. R. Sarukkai, "Link prediction and path analysis using markov chains," *Computer Networks*, vol. 33, no. 1, pp. 377–386, 2000.

[8] M. Al Hasan and M. J. Zaki, "A survey of link prediction in social networks," in *Social network data analytics*, pp. 243–275, Springer, 2011.

[9] D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey, "Drugbank: a comprehensive resource for in silico drug discovery and exploration," *Nucleic acids research*, vol. 34, no. suppl 1, pp. D668–D672, 2006.

[10] A. Koutsoukas, B. Simms, J. Kirchmair, P. J. Bond, A. V. Whitmore, S. Zimmer, M. P. Young, J. L. Jenkins, M. Glick, R. C. Glen, *et al.*, "From in silico target prediction to multi-target drug design: current databases, methods and applications," *Journal of proteomics*, vol. 74, no. 12, pp. 2554–2574, 2011.

[11] K. Bleakley and Y. Yamanishi, "Supervised prediction of drug–target interactions using bipartite local models," *Bioinformatics*, vol. 25, no. 18, pp. 2397–2403, 2009.

[12] D. Lin, "An information-theoretic definition of similarity.," in *ICML*, vol. 98, pp. 296–304, 1998.

[13] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American society for information science and technology*, vol. 58, no. 7, pp. 1019–1031, 2007.

[14] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of molecular biology*, vol. 147, no. 1, pp. 195–197, 1981.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] N. Biggs, *Algebraic graph theory*. Cambridge university press, 1993.

[17] N. Benchettara, R. Kanawati, and C. Rouveirol, "Supervised machine learning applied to link prediction in bipartite social networks," in *Advances in Social Networks Analysis and Mining (ASONAM), 2010 International Conference on*, pp. 326–330, IEEE, 2010.

[18] W. Liu and L. Lü, "Link prediction based on local random walk," *EPL (Europhysics Letters)*, vol. 89, no. 5, p. 58007, 2010.

[19] T. Zhou, L. Lü, and Y.-C. Zhang, "Predicting missing links via local information," *The European Physical Journal B-Condensed Matter and Complex Systems*, vol. 71, no. 4, pp. 623–630, 2009.

[20] Z. Huang, X. Li, and H. Chen, "Link prediction approach to collaborative filtering," in *Proceedings of the 5th ACM/IEEE-CS joint conference on Digital libraries*, pp. 141–142, ACM, 2005.

[21] P. Jaccard, *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz, 1901.

[22] B. A.-L. BARABÁSI and E. Bonabeau, "Scale-free," *Scientific American*, 2003.

[23] R. K. Merton, "The matthew effect in science," *Science*, vol. 159, no. 3810, pp. 56–63, 1968.

[24] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 18, no. 1, pp. 39–43, 1953.

[25] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, pp. 1137–1145, 1995.

[26] M. S. Granovetter, "The strength of weak ties," *American journal of sociology*, pp. 1360–1380, 1973.

[27] M. Granovetter, "The strength of weak ties: A network theory revisited," *Sociological theory*, vol. 1, no. 1, pp. 201–233, 1983.

[28] N. Friedkin, "A test of structural features of granovetter's strength of weak ties theory," *Social Networks*, vol. 2, no. 4, pp. 411–422, 1980.

[29] L. K. Gallos, H. A. Makse, and M. Sigman, "A small world of weak ties provides optimal global integration of self-similar modules in functional brain networks," *Proceedings of the National Academy of Sciences*, vol. 109, no. 8, pp. 2825–2830, 2012.

[30] P. Csermely, "Strong links are important, but weak links stabilize them," *Trends in biochemical sciences*, vol. 29, no. 7, pp. 331–334, 2004.

[31] L. Lü and T. Zhou, "Link prediction in weighted networks: The role of weak ties," *EPL (Europhysics Letters)*, vol. 89, no. 1, p. 18001, 2010.

[32] D. R. Swanson, "Fish oil, raynaud's syndrome, and undiscovered public knowledge," *Perspectives in biology and medicine*, vol. 30, no. 1, pp. 7–18, 1986.

[33] R. A. DiGiacomo, J. M. Kremer, and D. M. Shah, "Fish-oil dietary supplementation in patients with raynaud's phenomenon: a double-blind, controlled, prospective study," *The American journal of medicine*, vol. 86, no. 2, pp. 158–164, 1989.

[34] N. R. Smalheiser, "Literature-based discovery: Beyond the abcs," *Journal of the American Society for Information Science and Technology*, vol. 63, no. 2, pp. 218–224, 2012.