Research & Methods
ISSN 1234-9224, e-ISSN 2544-0799
Vol. 29 (1, 2020): 5–28
The Ohio State University Libraries
Columbus, Ohio, USA
Institute of Philosophy and Sociology
Polish Academy of Sciences, Warsaw, Poland
www.askresearchandmethods.org
https://doi.org/10.18061/ask.v29i1.0002

Proximity-based Methods for Link Prediction in Graphs with R package 'linkprediction'[1]

Michał Bojanowski[2]
Kozminski University

Bartosz Chroł
ICM, University of Warsaw

Link prediction is a problem of predicting future edges of an undirected graph based on a single snapshot of data of that graph. Vertex proximity measures are indicies giving numerical scores for every pair of vertices in a graph that can be used for predicting future edges. This short note describes an R package 'linkprediction' implementing 20 different vertex similarity and proximity measures from the literature. The article provides the definitions of implemented measures, describes the main user-facing functions, and illustrates the use of the methods with a problem of predicting future co-authorship relations between researchers of the University of Warsaw.

1. INTRODUCTION

Graphs are a popular way of representing structures of interactions between elements of a studied system. For example, in social sciences the vertices are used to represent people, organizations or other social entities, while edges represent ties such as friendship, collaboration or flow of capital. As such, network data and methods for its analysis appear in many disciplines (Wasserman and Faust 1994; Brandes et al. 2013; Barabási 2016; Jackson 2010; Goyal 2012; M. Newman 2010).

In many settings, networks are dynamic (for example: Van de Bunt, Van Duijn, and Snijders 1999; Ferligoj et al. 2015, and many others). The question arises whether and how we can use network data about the past or present to formulate predictions about the topology of the network in the future. Modeling the evolution of graphs is a complex problem with an ongoing research effort in formulating statistical models. One example of such models is the Temporal Exponential-family Random Graph Model (TERGM, Hanneke, Fu, and Xing 2010; Pavel N. Krivitsky and Handcock 2014) in which time is assumed to be discrete and the dynamics of the graph is represented by two conditional probability distributions – for edge formation and edge dissolution – each specifed as an Exponential-family Random Graph Model. Another example is the Stochastic Actor-Oriented Model (SAOM, Snijders 1996) in which network change is modeled as a continuous time process of actors (vertices) making multinomial choices about forming or dissolving the network edges they are incident on. Estimation of the mentioned models requires that we observe the network at least at two points in time. In addition, because the estimation relies on Markov Chain Monte Carlo methods (Pavel N. Krivitsky and Handcock 2018; Ripley et al. 2018), it becomes hardware-challenging to fit these models to network data beyond a couple thousand vertices. In both cases fitted models can be used to simulate future realizations of the graph.

A related and somewhat simpler problem is trying to predict future edges based on a single snapshot of the network. This problem has been framed as a problem of *link prediction* (Liben-Nowell and Kleinberg 2007). Approaches to solve it include various *node similarity* or *proximity-based* indices. These indices allow for computing a score for every pair of vertices in the given network, which in turn can be used for predicting if an edge is likely in the future. The indicies usually implement heuristics and qualitative ideas about how the network evolves. In contrast to the statistical modeling frameworks mentioned above, the node proximity measures are, on the one hand, rather simplistic and provide very limited insights into "how" or "why" the network evolves. On the other hand, they are usually computationally much less demanding, which makes their application to larger datasets more feasible. Proximity indices for link prediction have been used for example to: identify proteins likely to interact (Clauset, Moore, and Newman 2008), predict future co-authorship links (Liben-Nowell and Kleinberg 2007), prototype recommendation engines on social networking websites (Backstrom and Leskovec 2011), or forecast dynamics of terrorist networks (Desmarais and Cranmer 2013). To provide a better context, let us consider the last two publications above.

Backstrom and Leskovec (2011) analyzed Facebook data for all users living in Iceland (approx. 170 000 users). An often-implemented feature of social networking websites is recommending "friends" or contacts. A user is presented with a list of people with whom he is not connected but whom he might want to connect to. From the perspective of the social network data, this is indeed a link prediction problem – identifying pairs of users who are not connected at present but are likely to be connected in the future given their location in the network.

Understanding global international security systems and terrorism is one of challenges of contemporary political science. It is of particular interest to predict transnational terrorist attacks – not an easy task given the complex and evolving relationships of hostility and cooperation between different terrorist groups in different countries. Desmarais and Cranmer (2013) used the ITERATE dataset (Mickolus 2008) and its data on acts of terror where the perpetrator and victim come from different countries and, as such, constitute an edge in a larger graph of terrorist events. Using some of the methods presented in this article in their ERG models enabled them to forecast future terrorist events rather successfully.

In this short note we present an R package 'linkprediction' (Michal Bojanowski and Chrol 2018) that provides implementations of 20 node proximity indices collected through an extensive review of the literature. To our knowledge, these methods are not available to the R community apart from the function `similarity()` in the 'igraph' package (Csardi and Nepusz 2006) which implements three of these indices. Python library "NetworkX" (Hagberg, Schult, and Swart 2008) also implements some, but not all, of these indices.

Other features of the presented 'linkprediction' package are:

- It supports objects of class "igraph" (package 'igraph,' Csardi and Nepusz 2006) or "network" (package 'network,' Butts 2008, 2015) – probably the two most popular classes for network data in R.
- Where possible, functions use sparse matrices for efficient computation.
- For every index the results can be returned in three forms: a matrix, an edgelist data frame, or an "igraph" object with the scores assigned as edge attributes. This facilitates further analysis with functions from other packages. We provide more details and illustrations in Section 4.
- An example dataset with a subgraph of a co-authorship network from the University of Warsaw (1486 vertices, 7505 edges) is provided to facilitate examples and possibly test new measures/approaches to link prediction.

The remainder of the article is organized as follows. In Section 2 we provide a formal definition of a node proximity index. Section 3 provides a detailed list of implemented methods. Section 4 showcases the `proxfun()` function – the main interface to all the methods – its arguments and types of values that it can return. Section 5 provides an illustrative example of predicting co-authorship links among the researchers of the University of Warsaw. We also provide a simple empirical comparison of the measures. The article concludes with the discussion in Section 6.

2. GENERIC NODE PROXIMITY INDEX

To introduce some notation, let *graph* G consist of a set of *vertices* V and a set of *edges* $E \subseteq V \times V$ between these vertices. We will interchangeably use the terms "nodes" and "links" for "vertices" and "edges" respectively. A pair of vertices is called *a dyad*. If an edge exists between two vertices, they are adjacent to each other. An *adjacency matrix A* is a matrix representation of a graph. It is a square matrix with generic element a_{xy} equal to 1 if vertices x and y are adjacent and 0 otherwise.

A node proximity index is a function S giving a real number score to every dyad in graph G:

$$S(x, y): V \times V \mapsto \Re$$

It is convenient to arrange the values of $S(x, y)$ into a matrix $[s_{xy}]$. We will use the shorthand s_{xy} in the definitions of various measures in Section 3.

The terms "node similarity index" or "proximity-based index" seem to be used rather interchangeably in the literature. For the sake of clarity, we will use the first of the two terms throughout this article. The mentioned terms may also be a source of confusion due to two related concepts in network analysis: homophily and graph distance.

First, homophily is one of the mechanisms often found to be important in explaining the structure of networks (McPherson, Smith-Lovin, and Cook 2001). It implies that network edges tend to be more likely between vertices *similar* to each other in terms of the specified vertex attribute, such as gender, age, taste in music etc. (see also Bojanowski and Corten 2014). In contrast to the this homophily-related understanding, "node similarity" indices covered in this article do not use any information about possible vertex attributes and the term "similarity" has a rather informal meaning.

Second, graph distance is an important concept in graph theory. It is defined as the length of the shortest path connecting two vertices in the graph. Two vertices are said to be proximate if the graph distance between them is relatively short. Most of the indices we cover below use that concept more or less directly.

3. OVERVIEW OF IMPLEMENTED METHODS

The methods implemented in the package and described in this section have been gathered from many different sources. All measures give a proximity score between two vertices. Some measures are symmetrical by definition, some had to be modified to achieve symmetry. The scores of different measures have different scales, but for link prediction the rankings of dyads according to scores are of primary importance.

Following Lü and Zhou (2011), we group proximity measures into three categories: local, quasi-local, and global. Local methods focus only on the properties of the neighborhoods of the given pair of vertices. Global methods take into account information about the network as a whole. Quasi-local methods lie somewhere in between the local and global methods. They need more information than local methods, but still do not need information about the whole graph.

In the presented ‘linkprediction’ package measures can be computed with the function `proxfun()`, which we will describe in more detail in Section 4. The descriptions of the measures in the following sections contain short strings in parentheses next to the measure name. These are names or acronyms that can be supplied to the `method` argument of `proxfun()` function to select the appropriate measure; for example, the call `proxfun(g, method=”aa”)` will compute Adamic-Adar proximity scores for all dyads in supplied graph `g`.

3.1 Additional notation

We will use the following additional notation in measure definitions. A *subgraph* G' of a graph G is a graph with vertices $V' \subseteq V$ and edges $E' = E \cap V' \times V'$. A *path* connecting vertices x and y is a series of adjacent edges starting from x and finishing with y. A graph is *connected* if every pair of vertices is connected by a path, otherwise it is *disconnected*. The largest connected subgraph of a graph is called a *giant component*. A *neighborhood* of the vertex x is a set of vertices adjacent to x. The remaining notation is presented in Table 3.1. All vectors are assumed to be column vectors.

Table 3.1 Summary of notation.

Symbol	Description
$\lvert Q \rvert$	Cardinality of some set Q
$A = [a_{xy}]$	Adjacency matrix of a graph
n	Number of vertices in the graph
x, y, z	Generic vertices
$\Gamma(x)$	Set of all neighboring vertices of vertex x
$k_x = \sum_j a_{xj} = \lvert \Gamma(x) \rvert$	Degree of vertex x
$paths_{xy}^{<l>}$	Set of all paths of length l from x to y
$D = \begin{bmatrix} k_1 & 0 & \dots & 0 \\ 0 & k_2 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & k_n \end{bmatrix}$	Degree matrix
$L = D - A$	Laplacian matrix
$S = [s_{xy}]$	Proximity matrix
L^+	Moore-Penrose pseudo-inverse of matrix L

3.2 Local methods

Most of the local measures are variations of the “common neighbors” measure (M. E. J. Newman 2001).

Common neighbors (`cn`)
(M. E. J. Newman 2001) The measure implements an intuition that two scientists are more likely to collaborate if they have collaborated with the same group of people in the past. M. E. J. Newman (2001) used this method in the study of collaboration networks, showing positive relation between the number of common neighbors and probability of collaborating in the future.

$$s_{xy} = |\Gamma(x) \cap \Gamma(y)|,$$

Salton Index (`cos`)
It measures the cosine of the angle between columns of the adjacency matrix, corresponding to given vertices. This measure is commonly used in information retrieval.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\sqrt{k_x \times k_y}}.$$

Jaccard Index (`jaccard`)
(Jaccard 1912) Jaccard Index measures how many neighbors of given nodes are shared. It reaches its maximum if $\Gamma(x) = \Gamma(y)$, which means all neighbors are shared.

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

Sørensen Index (`sor`)
(Sørensen 1948) This method is similar to Jaccard Index, as it measures the relative size of an intersection of neighbors’ sets.

$$s_{xy} = \frac{2|\Gamma(x) \cap \Gamma(y)|}{k_x + k_y}.$$

Hub Promoted Index (`hpi`)
(Ravasz et al. 2002) This measure assigns higher scores to links adjacent to hubs (high-degree nodes), as the denominator depends on the lower degree only.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\min\{k_x, k_y\}}.$$

Hub Depressed Index (`hdi`)
(Ravasz et al. 2002) This measure, in contrast to Hub Promoted Index, assigns lower scores to links adjacent to hubs, since it penalizes big neighborhoods.

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{\max\{k_x, k_y\}}.$$

Leicht-Holme-Newman Index (`lhn_local`)

(Leicht, Holme, and Newman 2006) A variant of Common Neighbors, similar to Salton Index

$$s_{xy} = \frac{|\Gamma(x) \cap \Gamma(y)|}{k_x \times k_y}.$$

Preferential Attachment (`pa`)

(Barabási and Albert 1999) Preferential Attachment was developed as a model for the growth of a network in the sense of arrival of new nodes. This is often not really the case as the network might evolve (new links are added and some existing ones removed) without changes to the node set. However, if we follow the intuition behind the preferential attachment model, namely that nodes with high degree are more attractive to connect to, we may expect that links are more likely to be incident on nodes with high degree. Hence:

$$s_{xy} = k_x \times k_y.$$

Adamic-Adar Index (`aa`)

(Adamic and Adar 2001) This measure extends the idea of counting common neighbors by introducing weights inversely proportional to their degrees. A common neighbor that is unique to only a few nodes is more important (has more weight) than a high-degree node. Note that if a node *z* is a common neighbor of nodes *x* and *y*, then its degree is at least 2.

$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{\log k_z}.$$

Resource Allocation Index (`ra`)

(Zhou, Lü, and Zhang 2009) This measure is motivated by a resource transmission process in which common neighbors of nodes *x* and *y* play the role of transmitters spreading a unit of a resource. With an additional assumption that each transmitter spreads its resource equally across links, the measure captures how many resources *y* received from *x* (or vice versa) (c.f. Section 5 of Zhou, Lü, and Zhang 2009).

$$s_{xy} = \sum_{z \in \Gamma(x) \cap \Gamma(y)} \frac{1}{k_z}.$$

3.3 Global methods

Here we describe global methods.

Katz Index (`katz`)
(Katz 1953) Katz Index counts all the paths between the given pair of nodes, with shorter paths having larger weights.

$$s_{xy} = \sum_{l=1}^{\infty} \beta^l \, |paths_{xy}^{<l>}|,$$

where β is a free parameter. The sum converges when β is lower than the reciprocal of the largest eigenvalue of adjacency matrix. If this condition is satisfied, Katz Index can be expressed in a matrix form

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

where A is the adjacency matrix and I is the identity matrix.

Leicht-Holme-Newman Index, global version (`lhi_global`)
(Leicht, Holme, and Newman 2006) This is a variant of Katz Index, based on the concept that two nodes are proximate if their neighbors are proximate themselves. It counts all paths between two nodes, but weights them by the expected number of such paths in a random graph with the same degree distribution. This measure is proportional to the following matrix expression:

$$S = D^{-1}\left(I - \frac{\phi A}{\lambda_1}\right)^{-1} D^{-1},$$

where λ_1 is the largest eigenvalue of adjacency matrix A, and φ is a free parameter.

Average Commute Time (`act`)
(Klein and Randić 1993) ACT similarity index is given by

$$s_{xy} = \frac{1}{n(x,y)} = \frac{1}{m(x,y) + m(y,x)},$$

where $m(x, y)$ is the average number of steps required by a random walker starting from x to reach y. To achieve symmetry, we take the sum of two directional commute times. Thus, two nodes are similar if they are closer to each other and have shorter commute time. Average Commute Time could be computed by solving a collection of linear equations stemming from a Markov Chain analysis, but it is more straightforward to compute it in terms of the pseudo-inverse of the Laplacian matrix, L^+. Namely:

$$n(x,y) = 2M(l_{xx}^{+} + l_{yy}^{+} - 2l_{xy}^{+}),$$

where $l_{xy}^{+} = [L^{+}]_{xy}$ and M is the number of edges. Thanks to the special form of the Laplacian matrix, its pseudoinverse L^{+} could be computed using the formula of Fouss et al. (2007):

$$L^{+} = \left(L - \frac{ee^{T}}{n}\right)^{-1} + \frac{ee^{T}}{n},$$

where e is a column vector made of 1s.

Normalized Average Commute Time (`act_n`)

(Klein and Randić 1993) is a variant of ACT above, which takes into account node degrees, as for high-degree node (hub) y, $m(x, y)$ is usually small regardless of x.

$$s_{xy} = \frac{1}{(m(x,y)\pi_{y} + m(y,x)\pi_{x})},$$

where π is a stationary distribution of a Markov chain describing a random walker on the graph. It can be shown that on a connected graph

$$\pi(x) = \frac{k_{x}}{\sum_{y} k_{y}}.$$

Cosine based on (`cos_l`)

(Fouss et al. 2007) It measures the cosine of the angle between node vectors in a space spanned by columns of L^{+}.

$$s_{xy} = \frac{l_{xy}^{+}}{\sqrt{l_{xx}^{+} l_{yy}^{+}}}$$

Random Walk with Restart (RWR)

This is an adaptation of the PageRank algorithm (Brin and Page 1998). Consider a random walker starting from node x and, periodically, with probability α, returning to x. Let q_{x} be a stationary distribution of a Markov chain describing this walker. From a definition of stationary distribution:

$$q_{x} = (1-\alpha)P^{T}q_{x} + \alpha e_{x},$$

where e_{x} is a unit vector with 1 in a position corresponding to node x, and P is a transition matrix describing an ordinary random walker, $P_{xy} = 1/k_{x}$ if $A_{xy} = 1$ and 0 otherwise. The solution for all nodes simultaneously is

$$q = [q_{1}|q_{2}| \dots |q_{n}] = \alpha(I - (1-\alpha)P^{T})^{-1}.$$

In order to achieve symmetry, the RWR index is defined as

$$s_{xy} = q_{xy} + q_{yx}.$$

L^+ directly (`l`)

(Fouss et al. 2007) L^+ provides a direct measure of proximity, as its elements are the inner products of vectors from a Euclidean space, which preserves Average Commute Time between nodes (see Fouss et al. 2007 for details).

$$S = L^+.$$

Matrix Forest Index (`mfi`)

(Chebotarev and Shamis 1997) Matrix Forest Index can be understood as the ratio of (1) the number of spanning rooted forests such that nodes x and y belong to the same tree rooted at x to (2) the number of all spanning rooted forests of the network. See Chebotarev and Shamis (1997) for detailed derivations.

$$S = (I + L)^{-1}.$$

3.4 Quasi-local methods

Geodesic distance (`dist`)

In this approach we expect edges to appear more likely between the vertices that are closer to each other in terms of geodesic distance. The proximity index becomes:

$$s_{xy} = \begin{cases} \infty & \text{if } x = y \\ 0 & \text{if } x \text{ and } y \text{ are not connected} \\ \dfrac{1}{p_{xy}} & \text{in other cases} \end{cases}$$

where $p_{xy} = \min\{l: path_{xy}^{<l>} \text{ exists}\}$ is the length of the shortest path connecting x and y. It is not implemented in ‘linkprediction’ package but available with code{igraph::distances()}. We list it here for the sake of completness.

Local Path Index (`lp`)

(Zhou, Lü, and Zhang 2009)

$$S = A^2 + \epsilon A^3,$$

where ϵ is a free parameter. This measure benefits from more information than simple common neighbors, as it looks at neighborhoods of second order.

4. USAGE

The main function in the package is `proxfun()`, which calculates scores of selected node proximity measure (argument `method`) based on the provided graph (argument `graph`). Let us use the following simple graph as an example (see Figure 4.1).

```
library(igraph)
g <- make_graph( ~ 1 -- 2:3, 4 -- 2:3:5)
```

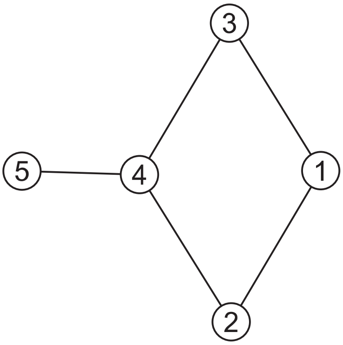

Figure 4.1: Simple example graph.

To calculate scores of Common Neighbors, for example, we call `proxfun()` with argument `method = “cn”`. By default, the result is a square matrix of scores. Rows and columns correspond to vertex ids in the graph object.

```
proxfun(g, method=”cn”)

##   1 2 3 4 5
## 1 0 0 0 2 0
## 2 0 0 2 0 1
## 3 0 2 0 0 1
## 4 2 0 0 0 0
## 5 0 1 1 0 0
```

The matrix is, by definition, symmetric. We see can that, for example, vertex 5 has one neighbor in common with vertices 2 and 3 (vertex 4 in both cases) while vertex 2 has two neighbors in common with vertex 3 (vertices 1 and 4) and one neighbor in common with vertex 5 (vertices 4 mentioned earlier).

Alternatively, and this can be controlled with argument `value`, the function can return a data frame containing an edge list (`value=”edgelist”`). This is a data frame with columns:

- `from`, `to` – vertex ids of the adjacent vertices
- `value` – scores of the selected proximity index

```
proxfun(g, method=”cn”, value=”edgelist”)

##   from to value
## 1    4  1     2
## 2    3  2     2
## 3    5  2     1
## 4    2  3     2
## 5    5  3     1
## 6    1  4     2
## 7    2  5     1
## 8    3  5     1
```

In this form, dyads that received a score of 0 are removed. Additionally, even though the scores are symmetric, the edge list contains all non-zero score values. This redundancy is deliberate as it facilitates joining such a data frame with possibly other dyadic data about the given network.

The third option is `value=”graph”`. The object returned is an “igraph” object with the same vertex set as the supplied graph and with edges in all dyads that received a non-zero score. The score itself is stored in an edge attribute `“weight”`. For example:

```
g.cn <- proxfun(g, method=”cn”, value=”graph”)
g.cn

## IGRAPH 576af01 UNW- 5 4 --
## + attr: name (v/c), weight (e/n)
## + edges from 576af01 (vertex names):
## [1] 3--5 2--5 2--3 1--4

E(g.cn)$weight

## [1] 1 1 2 2
```

5. ILLUSTRATIVE EXAMPLE

Let us consider the problem of predicting whether two researchers who did not collaborate in the past will co-author a publication together. We will use the data `uw` provided with the ‘linkprediction’ package.

5.1 Data

Data `uw` provided with the package 'linkprediction' is an `igraph` object representing an undirected graph of 1486 researchers (vertices) connected with 7505 edges. Two researchers are connected if they co-authored at least one publication in the period 2007-2012. The graph was assembled from bibliographic data extracted from the Polish Scholarly Bibliography (PBN 2017). A publication record was included if at least one of the authors was an employee of the University of Warsaw. The network in the `uw` object is a subgraph of that larger data consisting of researchers who (1) published at least once in 2007-2009, (2) published at least once in 2010-2012, and (3) are members of the largest connected component of the co-authorship graph based on publications in 2007-2009.

The network has additional vertex and edge attributes. In particular:

- `affiliation` – Vertex attribute identifying groups of departments by scientific field: natural sciences, social sciences, humanities, other, and external (co-authors who are not employees of UW).
- `p1` and `p2` – Logical edge attributes. If `p1` is `TRUE` then researchers incident on such an edge co-authored at least one publication in the first period (2007-2009). Analogously, if `p2` is `TRUE` then incident researchers co-authored at least one publication in the second period (2010-2012).

The network in the first period (i.e. with edges for which `p1` is not `TRUE` removed) is shown in Figure 5.1.

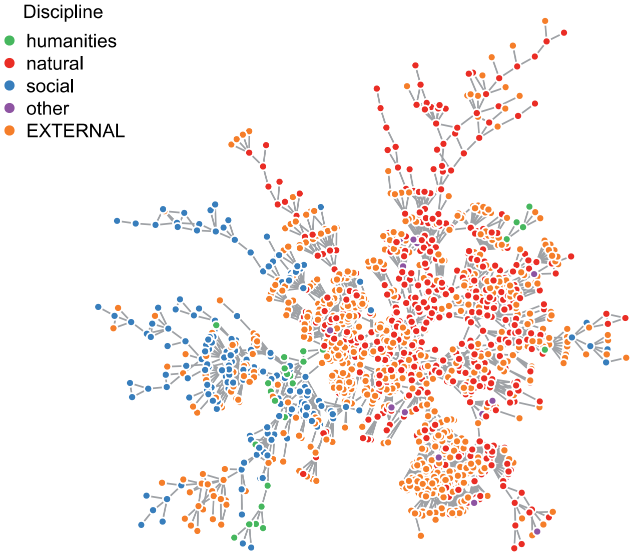

Figure 5.1: Co-authorship network in period 1.

5.2 Procedure

Our goal is to use the co-authorship network from period 1 to predict edges in period 2 in dyads that were disconnected in period 1. These are the *new* co-authorship ties. To this end we will use all of the methods implemented in the package and described in detail earlier in Section 3.

The procedure will consist of the following steps:

1. Calculate the scores of the measures for all the dyads in the network from period 1 (the “training data”)
2. Create the “test data”, consisting of dyads that were not connected in period 1 labelled with `TRUE` or `FALSE` depending whether there is a tie in that dyad in period 2.
3. Evaluate predictions using the scores from (1) vis a vis the labels from (2) using ROC curves (e.g. Fawcett 2006). Results are presented in the section 5.3 below.
4. Compare the measures empirically by looking at their correlations using Principal Component Analysis. Results are presented in the section 5.4 below.

Every dyad in the `uw` data is in one of four states. It can be

1. Disconnected in both periods – the two researchers did not work together at all.
2. Disconnected in period 1 but connected in period 2 – the two researchers started collaborating in period 2.
3. Connected in period 1 but disconnected in period 2 – the two researchers stopped collaborating in period 2.
4. Connected in both periods – the two researchers collaborated for the whole time under study.

The frequencies of those four states can be obtained by counting dyads depending on their connectedness in period 1 (variable `p1`), connectedness in period 2 (variable `p2`):

Table 5.1: Classification of all the dyads in the `uw` data.

Connected in period 1	Connected in period 2	Test data	New co-authorship	Frequency
FALSE	FALSE	TRUE	FALSE	1095850
FALSE	TRUE	TRUE	TRUE	1343
TRUE	FALSE	FALSE	FALSE	2069
TRUE	TRUE	FALSE	FALSE	4093

Thus, among 1103355 all possible pairs of researchers 2069 + 4093 = 6162 collaborated in period 1 and 1343 + 4093 = 5436 collaborated in period 2, of which 1343 are the *new* collaborations that we want to predict. For these dyads the logical variable "New co-authorship" is equal to `TRUE`. As mentioned earlier, we limit our predictive task to pairs of authors who did not collaborate in period 1 – for these dyads the value in the "Test data" column is `TRUE`. To recapitulate, the assembled dataset contains the following columns:

- `from` and `to` are vertex ids
- `p1` and `p2` are logical variables indicating whether researchers with ids `from` and `to` co-authored at least one publication in period 1 or period 2, respectively
- `new_coauthorship` variable is `TRUE` if researchers co-authored a publication in period 2, but not in period 1
- Remaining columns contain scores for dyads `from-to` computed with the measures described in Section 3.

At this point we can use standard tools for evaluating classifier performance – the ROC curves (Fawcett 2006) – to analyse scores vis a vis the true labels in variable `new_coauthorship`. This is presented in the next section.

5.3 Results: predictive performance

All the measures reviewed in Section 3 provide *uncalibrated* scores – they do not have comparable units and quite different value ranges. Using any one of them for formulating a predictions dyad requires establishing a threshold value. Dyads with values above the threshold will be predicted as *positives* (i.e. predict a network tie) and dyads with values below the threshold will be predicted as *negatives* (i.e. predict an absence of a tie). A standard tool for comparing such uncalibrated classification methods are Receiver Operating Characteristic (ROC) curves.

Figure 5.2 shows ROC curves for all 20 proximity measures computed with package ‘ROCR’ (Sing et al. 2005). A thorough description of ROC curves and their usage is beyond the scope of this article; therefore, we refer the reader to the work of Fawcett (2006). Here we provide only the most important elements for the presented context.

Let us consider the first panel of figure 5.2 showing the ROC curve for the Average Commute Time – other panels have the same structure. The vertical axis shows the True Positive Rate (TPR) – a probability of correctly predicting period 2 ties. The horizontal axis shows the False Positive Rate (FPR) – a probability of incorrectly predicting the period 2 ties. The values of the Average Commute Time vary between 0.000003411863 and 0.0022268, the higher the score the more likely a tie. Depending on the choice of the threshold, we will obtain different values of TPR and FPR. Should we classify all dyads as positives (with a threshold value of 3.4118632^{-6} or lower), the TPR will be equal to 1 and FPR will be equal to 1 – a point on the graph in the upper right corner. Should we classify all dyads as negatives (with a threshold value of 0.0022268 or higher), the TPR will be equal to 0 and FPR will be equal to 0 – a point on the graph in the lower left corner. Points

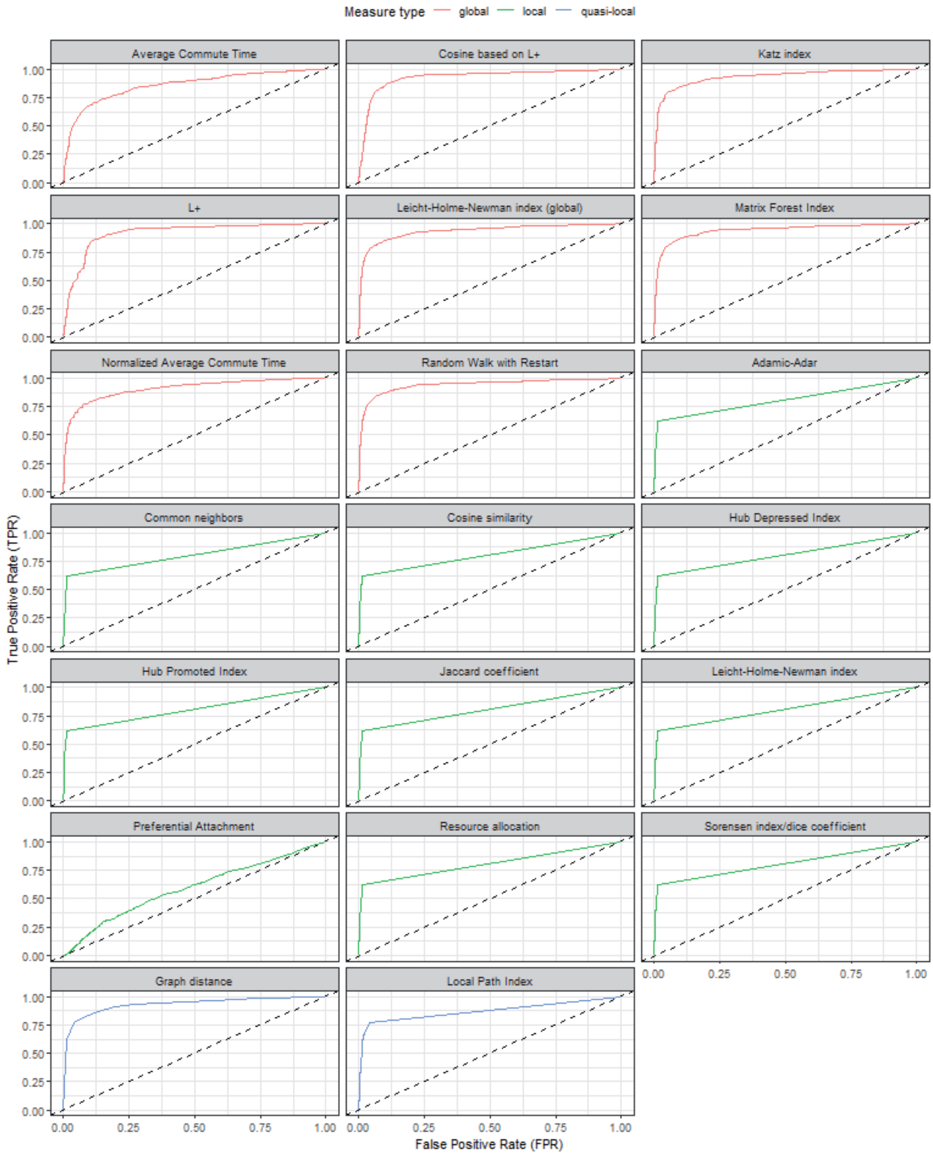

Figure 5.2: ROC curves for the 20 proximity measures used. Color corresponds to measure type: global, quasi-local and local.

on the diagonal (marked with a dashed line) correspond to performing prediction randomly but with a given probability of positives. For example, the point (0.5, 0.5) would correspond to making predictions by flipping a fair coin and predicting a network tie each time the coin falls on heads. Such random predictions essentially do not use any empirical information from the data, and thus are of no particular interest. A prediction model characterized with a point in the upper left corner of the graph performs perfectly: has a TPR of 1 and FPR of 0 – all ties are predicted correctly and none of the ties is predicted incorrectly. By varying the threshold, we will receive different pairs of TPR and FPR values, each corresponding to a point. These points jointly follow a curve shown on the plot. In general, the closer the curve to the upper-left corner, the better the predictive performance of the model.

A standard uni-dimensional way of comparing predictive performance is by calculating the Area Under Curve (AUC). The closer the value of AUC is to 1, the better the perfomance. Table 5.2 provides AUC values of the 20 measures.

Table 5.2: Measures and their AUC values.

Type	Measure	AUC
global	Random Walk with Restart	0.9417268
global	Matrix Forest Index	0.9405058
global	Leicht-Holme-Newman index (global)	0.9351180
global	Katz index	0.9322479
global	Cosine based on L+	0.9316619
quasi-local	Graph distance	0.9315297
global	L+	0.9169171
global	Normalized Average Commute Time	0.9102548
quasi-local	Local Path Index	0.8716868
global	Average Commute Time	0.8587340
local	Hub Promoted Index	0.8036433
local	Cosine similarity	0.8035110
local	Jaccard coefficient	0.8033008
local	Sorensen index/dice coefficient	0.8033008
local	Resource allocation	0.8032827
local	Adamic-Adar	0.8031710
local	Leicht-Holme-Newman index	0.8031322
local	Hub Depressed Index	0.8030719
local	Common neighbors	0.8028466
local	Preferential Attachment	0.5914478

Figure 5.2 and table 5.2 allow us to make the following observations:

1. Among the 20 measures compared, Random Walk with Restart performs best with Matrix Forest Index closely following.
2. In general, global methods perform better than local and quasi-local methods.
3. Local methods, while performing worse than the global methods, still provide quite high performance.
4. The ROC curves for most of the local methods follow a distinct pattern of rising abruptly in the beginning and then "rushing" in a straight line towards the upper-right corner. Preferential Attachment is a notable exception (with an equally notable poor predictive performance).

The presented methods used for similar tasks but on different data seem to have provided similar results. For example, in their experiments Backstrom and Leskovec (2011) also found that Random Walk with Restart performed best among those measures.

The distinct shape of the ROC curves for the local methods stems from the fact that all of them, apart from the Preferential attachment, are variations on the theme of Common neighbors. They implement different ways of counting and weighting the number of shared network partners of a dyad. If the network does is not characterized with a lot of clustering most of the dyads do not have any shared partners in common. For such dyads all these measures are equal to 0. The straight fragments of the ROC curves of the local methods correspond to these values, which do not bring any predictive information.

5.4 Results: comparing the measures

Detailed theoretical comparison of the implemented measures is beyond the scope of this article. We refer the reader to the sources cited in Section 3. In this section we provide an empirical perspective on the question of how the measures are (dis) similar to one another by comparing how similar or different the predictions they make are. This can be done by analyzing an inter-measure correlation matrix, which we do here using Principal Component Analysis. Such an analysis is of course limited, since it is conditioned on the particular dataset, which in turn documents a particular network formation process with all its particular sociological features. Nevertheless, we believe it is worthwhile, as it does show important differences between the measures.

Figure 5.3 shows variances of the components. We can clearly see that the first component accounts for most of the variance (57%). This is to be expected as all the measures share the same intention – link prediction through node proximity. The first three components account for 80% of variability cumulatively.

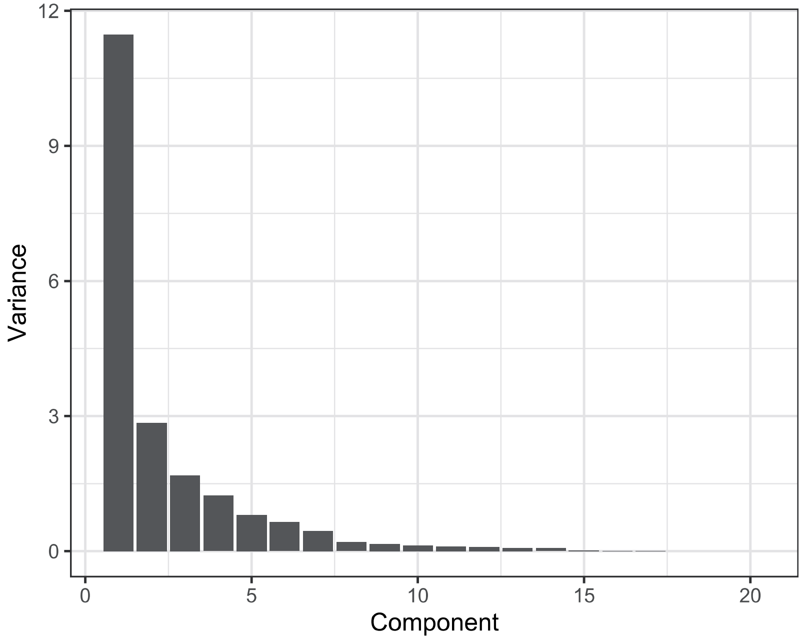

Figure 5.3: PCA component variances.

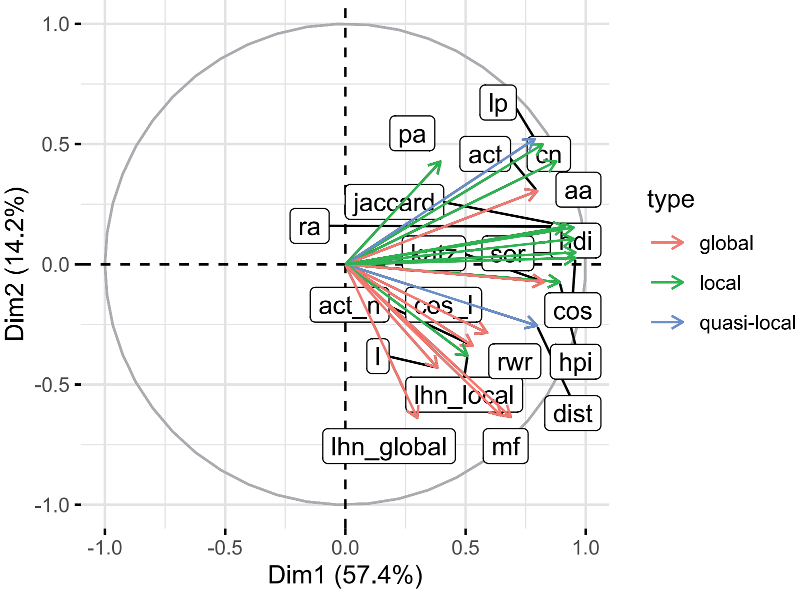

Figure 5.4: Proximity measures in the space of components 1 and 2. Labels correspond to acronyms from Section 3.

High variance of the first component suggests a largely uni-dimensional structure. This is also confirmed by looking at vectors for each measure in the space of the first two components, which is presented in figure 5.4. All the vectors point roughly in the same direction along the first component. However, we can see that local measures, such as Preferential Attachment (‘pa’), have the lowest correlations with (largest angle to) global measures, such as the global version of the Leicht-Holme-Newman index (‘lhn_global’). The measures cluster somewhat according to type as suggested by the colors. To inspect this further, we need to abstract away from the common variance attributable to the first component.

Figure 5.5 shows measure vectors in the space of components 2 and 3. Keeping in mind that the two components presented account for little over 22% of total variability of the measures, we can make the following observations:

1. Local methods, with the exception of Leicht-Holme-Newman (LHN) index, seem to form a separate group.
2. Global methods distinguish themselves from local methods and form two groups:
   – Random Walk with Restart, LHN, Matrix Forest Index
   – methods based on the Laplacian matrix of the graph and graph distance

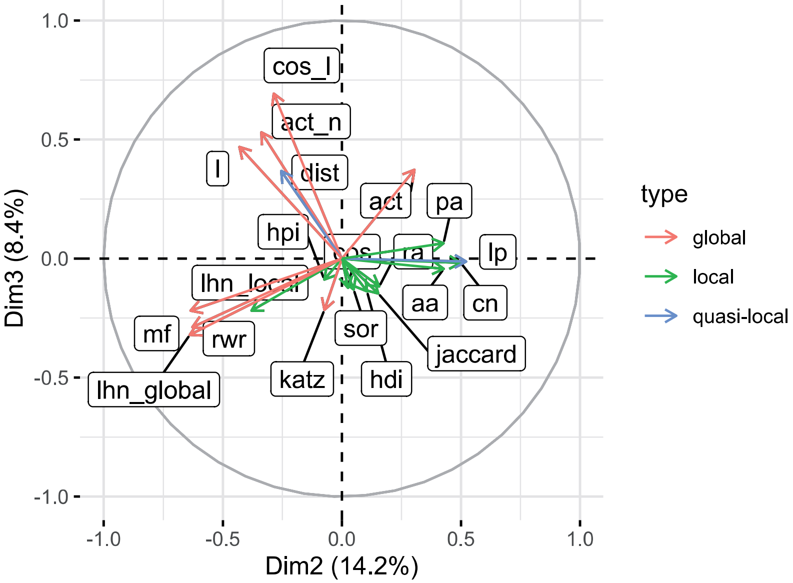

Figure 5.5: Proximity measures in the space of components 2 and 3. Labels correspond to acronyms from Section 3.

At this point, we leave more detailed comparative analysis for further research.

6. DISCUSSION

As signalled in the Introduction, the article presented a tool, an R package, which offers network analysis and prediction of network ties using node-proximity indices. Their advantages lie in relative simplicity and low computational costs. We have shown that their performance is quite high using a co-authorship network. Their advantages also become their limitations if the research questions go beyond pure prediction, for which these measures were designed.

First, the measures focus on a particular aspect of network topology – proximity. This is common to many social (and natural) settings, as social networks are usually characterized by high social closure following the “friend of a friend is likely to become a friend” motto. Incidentally, this is also observed in co-authorship networks. Nevertheless, there are a multitude of other aspects that play a role in social network formation, such as attribute-based heterogeneity and homophily, which are ignored by these measures.

Secondly, ignoring processes such as homophily is not only an “empistemological” problem, but also a disadvantage from a purely predictive point of view. The measures simply do not take into account any other potentially available information about the network nodes (e.g. researcher’s attributes, such as age, gender, affiliations, seniority etc.) or dyads (e.g. scientific degree similarity). Such information would definitely improve predictive performance.

NOTES

[1] Authors thank (Polish) National Science Centre for support through grant 2012/07/D/HS6/01971 for the project *Dynamics of Competition and Collaboration in Science: Individual Strategies, Collaboration Networks, and Organizational Hierarchies* (http://recon.icm.edu.pl).

[2] Corresponding author. Email: mbojanowski@kozminski.edu.pl

REFERENCES

Adamic, Lada, and Eytan Adar. 2001. “Friends and Neighbors on the Web.” *Social Networks* 25: 211–30. https://doi.org/10.1016/S0378-8733(03)00009-1

Backstrom, Lars, and Jure Leskovec. 2011. “Supervised Random Walks: Predicting and Recommending Links in Social Networks.” In *Proceedings of the Fourth Acm International Conference on Web Search and Data Mining*, 635–44. ACM. https://doi.org/10.1145/1935826.1935914

Barabási, Albert-László. 2016. *Network Science*. Cambridge University Press.

Barabási, Albert-László, and Réka Albert. 1999. “Emergence of Scaling in Random Networks.” *Science* 286 (5439): 509–12. https://doi.org/10.1126/science.286.5439.509

Bojanowski, Michał, and Rense Corten. 2014. "Measuring Segregation in Social Networks." *Social Networks* 39: 14–32. https://doi.org/10.1016/j.socnet.2014.04.001

Brandes, Ulrik, Garry Robins, Ann McCranie, and Stanley Wasserman. 2013. "What Is Network Science?" *Network Science* 1 (1): 1–15. https://doi.org/10.1017/nws.2013.2

Brin, Sergey, and Lawrence Page. 1998. "The Anatomy of a Large-Scale Hypertextual Web Search Engine." *Computer Networks and ISDN Systems* 30 (1–7): 107–17. https://doi.org/10.1016/S0169-7552(98)00110-X

Butts, Carter T. 2008. "Network: A Package for Managing Relational Data in R." *Journal of Statistical Software* 24 (2). https://doi.org/10.18637/jss.v024.i02

—. 2015. *Network: Classes for Relational Data*. The Statnet Project (http://statnet.org). http://CRAN.R-project.org/package=network.

Chebotarev, P. Yu., and E. V. Shamis. 1997. "The Matrix-Forest Theorem and Measuring Relations in Small Social Groups." *Automation and Remote Control* 58 (9): 1505–14.

Clauset, Aaron, Cristopher Moore, and Mark EJ Newman. 2008. "Hierarchical Structure and the Prediction of Missing Links in Networks." *Nature* 453 (7191): 98. https://doi.org/10.1038/nature06830

Csardi, Gabor, and Tamas Nepusz. 2006. "The Igraph Software Package for Complex Network Research." *InterJournal* Complex Systems: 1695. http://igraph.org.

Desmarais, Bruce A, and Skyler J Cranmer. 2013. "Forecasting the Locational Dynamics of Transnational Terrorism: A Network Analytic Approach." *Security Informatics* 2 (1): 8. https://doi.org/10.1186/2190-8532-2-8

Ferligoj, Anuška, Luka Kronegger, Franc Mali, Tom AB Snijders, and Patrick Doreian. 2015. "Scientific Collaboration Dynamics in a National Scientific System." *Scientometrics* 104 (3): 985–1012. https://doi.org/10.1007/s11192-015-1585-7

Fouss, Francois, Alain Pirotte, Jean-Michel Renders, and Marco Saerens. 2007. "Random-Walk Computation of Similarities Between Nodes of a Graph with Application to Collaborative Recommendation." *IEEE Transactions on Knowledge and Data Engineering* 19 (3): 355–69. https://doi.org/10.1109/TKDE.2007.46

Goyal, Sanjeev. 2012. *Connections: An Introduction to the Economics of Networks*. Princeton University Press.

Hanneke, Steve, Wenjie Fu, and Eric P. Xing. 2010. "Discrete Temporal Models of Social Networks." *Electronic Journal of Statistics* 4: 585–605. https://doi.org/10.1214/09-EJS548

Jaccard, Paul. 1912. "The Distribution of the Flora in the Alpine Zone." *New Phytologist* 11 (2): 37–50. https://doi.org/10.1111/j.1469-8137.1912.tb05611.x

Jackson, Matthew O. 2010. *Social and Economic Networks*. Princeton University Press.

Katz, Leo. 1953. "A New Status Index Derived from Sociometric Analysis." *Psychometrika* 18 (1): 39–43. https://doi.org/10.1007/BF02289026

Klein, Douglas J, and Milan Randić. 1993. "Resistance Distance." *Journal of Mathematical Chemistry* 12 (1): 81–95. https://doi.org/10.1007/BF01164627

Krivitsky, Pavel N, and Mark S Handcock. 2014. "A Separable Model for Dynamic Networks." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 76 (1): 29–46. https://doi.org/10.1111/rssb.12014

Krivitsky, Pavel N., and Mark S. Handcock. 2018. *Tergm: Fit, Simulate and Diagnose Models for Network Evolution Based on Exponential-Family Random Graph Models*. The Statnet Project (http://www.statnet.org). https://CRAN.R-project.org/package=tergm.

Leicht, E. A., Petter Holme, and M. E. J. Newman. 2006. “Vertex Similarity in Networks.” *Phys. Rev. E* 73 (2): 026120. https://doi.org/10.1103/PhysRevE.73.026120

Liben-Nowell, David, and Jon Kleinberg. 2007. “The Link-Prediction Problem for Social Networks.” *Journal of the American Society for Information Science and Technology* 58 (7): 1019–31. https://doi.org/10.1002/asi.20591

Lü, Linyuan, and Tao Zhou. 2011. “Link Prediction in Complex Networks: A Survey.” *Physica A* 390 (6): 1150–70. https://doi.org/10.1016/j.physa.2010.11.027

McPherson, Miller, Lynn Smith-Lovin, and James M Cook. 2001. “Birds of a Feather: Homophily in Social Networks.” *Annual Review of Sociology* 27 (1): 415–44. https://doi.org/10.1146/annurev.soc.27.1.415

Neuwirth, Erich. 2014. *RColorBrewer: ColorBrewer Palettes*. https://CRAN.R-project.org/package=RColorBrewer.

Newman, Mark. 2010. *Networks: An Introduction*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199206650.003.0001

Newman, M. E. J. 2001. “Clustering and Preferential Attachment in Growing Networks.” *Phys. Rev. E* 64 (2): 025102. https://doi.org/10.1103/PhysRevE.64.025102

PBN. 2017. “Polska Bibliografia Naukowa (Polish Scholarly Bibliography).” 2017. https://pbn.nauka.gov.pl/.

Ravasz, E., A. L. Somera, D. A. Mongru, Z. N. Oltvai, and Albert-László Barabási. 2002. “Hierarchical Organization of Modularity in Metabolic Networks.” *Science* 297 (5586): 1551–5. https://doi.org/10.1126/science.1073374

Michal Bojanowski and Bartosz Chrol (2020). linkprediction: Link Prediction Methods. R package version 1.0-1. https://github.com/recon-icm/linkprediction

Ripley, Ruth M., Tom A. B. Snijders, Zsofia Boda, András Vörös, and Paulina Preciado. 2018. “Manual for Siena Version 4.0.” Oxford: University of Oxford, Department of Statistics; Nuffield College.

Sing, T., O. Sander, N. Beerenwinkel, and T. Lengauer. 2005. “ROCR: Visualizing Classifier Performance in R.” *Bioinformatics* 21 (20): 7881. https://doi.org/10.1093/bioinformatics/bti623

Snijders, Tom A. B. 1996. “Stochastic Actor-Oriented Models for Network Change.” *Journal of Mathematical Sociology* 21 (1-2): 149–72. https://doi.org/10.1080/0022250X.1996.9990178

Sørensen, Thorvald. 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: 1–34.

Van de Bunt, Gerhard G, Marijtje AJ Van Duijn, and Tom AB Snijders. 1999. “Friendship Networks Through Time: An Actor-Oriented Dynamic Statistical Network Model.” *Computational & Mathematical Organization Theory* 5 (2): 167–92. https://doi.org/10.1023/A:1009683123448

Wasserman, Stanley, and Katherine Faust. 1994. *Social Network Analysis: Methods and Applications*. Vol. 8. Cambridge university press. https://doi.org/10.1017/CBO9780511815478

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2018. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Lionel Henry. 2018. *Tidyr: Easily Tidy Data with 'Spread()' and 'Gather()' Functions*. https://CRAN.R-project.org/package=tidyr.

Zhou, Tao, Linyuan Lü, and Yi-Cheng Zhang. 2009. "Predicting Missing Links via Local Information." *The European Physical Journal B* 71 (4): 623–30. https://doi.org/10.1140/epjb/e2009-00335-8