Notions of Similarity in Complex Networks

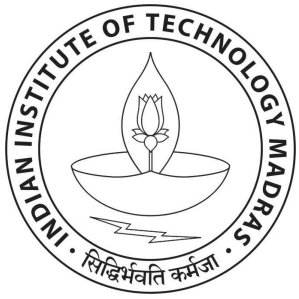
A THESIS

submitted by

ROHITH BHANDARU

in partial fulfilment of the requirements
for the award of the degree of

MASTER OF TECHNOLOGY



DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

MAY 2018

THESIS CERTIFICATE

This is to certify that the thesis titled **Notions of Similarity in Complex Networks**, submitted by **Rohith Bhandaru**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Venkatesh Ramaiyan
Research Advisor
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

I would like to thank Prof. Venkatesh Ramaiyan for believing in me and patiently guiding me through the ups and downs I faced while pursuing this project. His clear insights and sharp questions have showed me the path all through. I would also like to thank him for offering the introductory course on Complex Network Analysis, which provided a strong motivation and interest for me to pursue a project in network science.

I would like to thank Prof. Manikandan Narayanan for his valuable time. He has provided a new perspective to look at the problem and helped steer the project to its present stage. I would like to thank Prof. Balaraman Ravindran for his valuable time and rich insights during the weekly meetings. These meetings have helped in learning different problems that are being tackled in the networks area. It was an insightful and wonderful experience to attend the classes by some of the legendary teachers that the Institute has to offer. They will always remain a source of inspiration for me.

I would like to thank my friends Chandravadan and Chaturasan for being wonderful sources of inspiration. I have enjoyed several long discussions with them, which helped me to improve clarity of my thought. I would like to thank my batch mates for making these 5 years of stay at IITM, an enjoyable experience.

Last but not the least, I would like to express my heartfelt gratitude and love to my parents for their never ending support and constant encouragement.

ABSTRACT

Defining similarity between two nodes is a highly subjective area of research. We define similarity between two nodes in a network as the measure of how similar these two nodes are perceived to be, by every other node in the network. We call this similarity measure as *GSim*. We develop a random walk based metric to quantify the above relation. We provide a simple matrix formulation for *GSim*. We study various aspects of performance of *GSim* with other state-of-the-art similarity measures.

KEYWORDS: Complex networks ; Similarity; Random walk.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS i

ABSTRACT ii

LIST OF FIGURES iv

1 INTRODUCTION 1

2 Related Work 2

3 GSim 5

3.1 Preliminaries 5

3.2 Motivation 5

3.3 Formulation 6

3.4 Properties of GSim 11

3.4.1 Symmetry 11

3.4.2 Limited Information problem 11

3.4.3 Zero Similarity problem 11

3.4.4 Scalability 11

3.5 GSim computation algorithm 12

3.6 Time and space complexity analysis 13

4 Experimental results 14

4.1 Time analysis 14

4.2 Cluster analysis 15

5 GSim for multilayer networks 18

6 Conclusion and future work 20

LIST OF FIGURES

3.1 Description of a random walk 7

3.2 Example network 10

4.1 Time performance of *GSim*, *SimRank* and *PageSim* 15

4.2 Time performance of *GSim*, *SimRank* and *PageSim* as the probability of edge occurrence increases 16

4.3 Cluster performance of top $k\%$ similar node-pairs as given by *GSim*, *SimRank* and *PageSim* 17

5.1 Two scenarios of multilayer networks 19

CHAPTER 1

INTRODUCTION

A network is a mathematical construct with nodes, entities in a system, interconnected with edges based on pre-defined relationship. Consider the railway system of India. Players in the system are the stations. Suppose we define a relation as existence of a direct train between two stations, then, two nodes (stations) are connected if there is direct train between them with no intermediate station on the route. Thus a network is formed. Once we have this construct, we can perform various analyses to help design, maintain and modify the network. A famous among them is to quantize how similar two nodes in a network are. One needs to note that the answers to this question are highly subjective, i.e., the similarity between two nodes depends on how one defines the notion of similarity. As an illustration, consider the political network of Members of Parliament of India. Let us define two nodes to be similar if the measure of their importance in the network is similar. This definition will say that the Leader of the House and the Leader of the Opposition to be most similar nodes in the network. However, this is counter-intuitive as both of them will belong to rival parties. Thus one needs to carefully define the relation of similarity between nodes in a network. This definition will also depend on the kind of network we have at hand.

Similarity between two nodes in a network can be computed either purely based on the structure of the network or a combination of structure of the network (9; 1; 23) and machine learning techniques based on meta data of the network (2; 3). Through this work, we attempt to develop a structure based similarity measure called **GSim**. The relation that we use to define similarity between two nodes in a network is the possibility of reaching two nodes of interest in the network from every other node in the network. We use random walk based measure to quantize this similarity relation. The outline of this report is as follows: we present the prior research works in the field in Section 2, then discuss the motivation, formulation, properties and algorithm for *GSim* in Section 3. In Section 4, we present the performance of **GSim** as compared to other state-of-the-art measures.

CHAPTER 2

Related Work

Several similarity measures have been specifically designed for certain networks. However, these can be extended to other network types. Consider citation networks. The references among the body of academic publications form the citation network. The research papers are represented as nodes and a directed edge from a paper *i* to paper *j* exists if *i* cites *j*. The problem of finding similarity between two papers is found generally in the context of information retrieval and recommender systems. An intuitive similarity measure would incorporate both textual and structural properties of the papers in the network. Hybrid approaches of combining both textual and structural similarities have been tested in (13; 14). A text-based measure uses meta-data such as the title and the abstract as modeled in (15; 16) and also the full text as in (17). On the other end, finding similarity between papers in a citation network based purely on the structure of the network is widely studied and is found to give promising results.

Historical visualizations of citation network are *CoCitation* (9) and *Bibliographic Coupling* (10). In *CoCitation*, two papers *a* and *b* are similar to the extent of the number of papers that cite both *a* and *b*. In *Bibliographic Coupling*, two papers *a* and *b* are similar to the extent of the number of papers that are cited by both *a* and *b*. Neither *CoCitation* nor *Bibliographic Coupling* take the global structure of the network, i.e. overall paths between papers *a* and *b*, into account. *MatchSim* (11) is a local neighbor-based similarity measure built on the idea of measuring similarity of two nodes by the similarities of their pairwise matched neighbors instead of just calculating the number of common papers among the neighborhood of two papers. Thus, it is an iterative measure.

In (18), an iterative metric is proposed, which aims to define similarity between two nodes transitively, i.e., if papers (*a*, *b*) and (*b*, *c*) are similar then (*a*, *c*) are similar, known as *transitive node similarity*. In the formulation of *GSim*, it will be shown later that, as α, the decay parameter, tends to zero, *GSim* imitates the *transitive node similarity* in terms of formulation.

SimRank (1) is a widely reported global similarity measure due to its intuitive and sound mathematical basis. The main proposition of *SimRank* is that 'two objects are similar if they are referenced by similar objects'. Here, if there is a directed edge from a node i to node j, then i is said to reference j. Similarity between nodes a and b is calculated based on existence of equal length paths from nodes a and b to a node c, thus leaving paths that have unequal lengths. Having an objective relation for calculating similarity led to many extensions of this measure overcoming some of the serious drawbacks with *SimRank*, like, Limited Information problem, Zero Similarity problem and high computational complexity, which are later discussed in detail.

A variation of *SimRank* is pursued by *P-Rank* (4). *P-Rank* includes the effect of references made by the nodes of interest as well, i.e., 'two objects are similar if, (1) they are referenced by similar objects and (2) they reference similar objects'. Even *P-Rank* does not consider paths of unequal length. The natural extension of including paths that have unequal lengths in *SimRank* was pursued by *SimRank** (5) and *E-Rank* (6). *E-Rank* derives its formulation by considering two independent random surfers traversing paths to common node with any path lengths. *SimRank** brings in the notion of unequal lengths by modifying the mathematical expression of *SimRank* appropriately. *SimRank** also gives an approximate closed form expression to calculate similarity unlike *SimRank*. Other closed form expressions for *SimRank* algorithm have been given in (19; 20; 21).

All the above metrics are proposed on networks with a single type of edge. However, in reality, there can be multiple types of nodes and those nodes can be connected via multiple relations, for example, papers, authors, conferences, journals, web downloads can form the heterogeneous node set for citation network. *SimFusion*(22) attempts to make use of such heterogeneous data by defining a Unified Relationship Matrix to represent such heterogeneous data objects, and their interrelationships and measure similarity between two nodes. Thus, this method can be considered as extension of *SimRank* to multi-layer graphs.

There are attempts to connect centrality and similarity measures, for example, *CentSim*(23). In *CentSim*, two nodes with similar centrality vectors are defined to be similar. Centrality vector of a node includes various centrality measures like degree and PageRank as its components. Similarity between two centrality vectors is quantized

by using a Jaccard coefficient like formulation. A serious drawback arises from the above definition of *CentSim* as follows: even though two nodes have lower centrality measures, *CentSim* gives a high similarity score as their centrality values are nearby. This need not be true. Another interesting way of calculating similarities has been put forward in (24), where similarity is calculated from the perspectives of both query node and the candidate nodes. Here, query node is the node of interest for which we want to calculate similarity with respect to every node in the candidate list.

Another work that attempts connect centrality and similarity is *PageSim*(25), derived in the context of web page network. The central idea was that the centralities(feature vectors) of pages were propagated through the hyper-links and similarity between two pages was defined by a Jaccard coefficient like correlation among their feature vectors that are obtained after the end of propagation phenomenon. However, *PageSim* fails to provide intuition behind the formulation. *PageSim* also carries the drawback of *CentSim*, assigning high similarity for nodes with low propagation scores. This essentially arises from defining similarity using Jaccard coefficient based formulation. We overcome the high time complexities of similarity algorithm, drawbacks of *PageSim*, Limited Information problem, Zero Similarity problem through *GSim*.

Given the rise in size and scale of networks, a scalable model that efficiently quantifies similarity between nodes is required. We aim to devise a similarity relation with a strong theoretical backing and an efficient algorithm. We derive our inspiration for the similarity relation from transportation networks and this measure can be easily extended to other types of networks. We propose a random walk based quantization of similarity, similar to the ones proposed in *PageRank*(26) and *SimRank*.

CHAPTER 3

GSim

3.1 Preliminaries

Consider a transportation network as a directed graph $G_1(V, E)$, where *V* denotes the set of stops present in the network and *E* represents the set of edges based on a predefined relationship. Let the total number of stops in the network be *n*. Let *i* and *j* be two places (stops) of interest in the network. A directed edge starts from *i* and ends at *j*, if there exists a direct mode of transportation from *i* to *j* without any intermediary stops.

Let $\mathbf{A}_1$ denote the adjacency matrix of the network generated by the above relation rule with $[\mathbf{A}_1]_{ij}$ being 1 if an edge is directed from place *i* to place *j*. Let $\mathbf{W}_1$ denote the row normalized adjacency matrix. Let *O(i)* denote the set of out-link neighbors of node *i*, i.e., the set of immediate neighboring places of node *i* in the transportation network and let $|O(i)|$ denote its cardinality.

3.2 Motivation

Katz similarity (27) defines similarity measure between two nodes *i* and *j* by counting all paths between the nodes and damping them exponentially to favor short paths. This can be modeled as a random walk from *i* with an aim to reach *j*. For every path of length *l* traversed by a random surfer to reach *j*, a reward proportional to an exponential raised to *l* is given to similarity measure. However, in a probabilistic scenario, *Katz similarity* doesn't consider the number of times the event of a random surfer starting from *i* and reaching *j* via a path of length *l* occurs, and how many times the random surfer starts from *i*. Latter event can be related to the probability that a random surfer is present in state(node) *i* at a given time, π_i.

SimRank, however, approaches the problem of defining similarity in a different way. *SimRank* considers two independent random surfers starting their journey from *i*

and j one at each, and their meeting distance (as defined in (1)) is rewarded every time they meet at another node k with an amount proportional to an exponential raised to the length of traversal by both surfers. It can be observed that *SimRank* also doesn't account for recurrence of the entire process of rewarding the similarity measure as is the case with (6). We aim to bridge this gap of reasoning.

We use the insights from a transportation network perspective to define similarity based on reachability. We say that two places (nodes) i and j are similar if the two places are reachable from every other place in the network. In order to quantize this similarity rule, we consider an inverse random walk paradigm where an independent random surfer starts from a spectator node k. Steady state distribution for the random surfer is computed for every node. This distribution value at node i denotes the probability of the random surfer ending at node i in steady state. We quantize similarity between nodes i and j with respect to spectator node k as the product of distribution values of random surfer at nodes i and j. Thus, if one of the two distribution values is low, overall product will be lower, overcoming the problem of *PageSim*. However, in this probabilistic scenario, one needs to consider the frequency of the random surfer starting from node k. We assume that the more central a node is, the more frequent the random surfer starts at the node. So, we use the centrality value of a node i as its π_i. To compute the total similarity value of nodes i and j, we consider all the possible spectator nodes k in the network and sum over these nodes.

This inverse random walk paradigm leads to a compact matrix expression to compute similarity between any two pair of nodes. As there are many efficient matrix manipulation algorithms, overall time taken by the algorithm to compute all possible pair-wise similarities is drastically reduced.

3.3 Formulation

We aim to calculate similarity score between nodes i and j. We are primarily interested in calculating steady state probability distribution of a random surfer starting from node k in the network to nodes i and j.

We initially model a random surfer starting from a place k with an aim of reaching i. When the random surfer starts from k, the surfer chooses one of the out-going links

of *k* with a probability $1/|O(k)|$ and continues the surfing in a similar way as depicted in figure 3.1. Let *x* denote a possible path from place *k* to place *i* and let *l(x)* denote its length (total number of edges in the path *x*). Expected value of distance, *d(k, i)* between *k* and *i* can be formulated as:

$$d(k,i) \; = \sum_{x:\; k \to i} P(x).l(x) \tag{3.1}$$

where *P(x)* denotes the probability that the surfer undertakes path *x*. *P(x)* is given by:

$$P(x) = \prod_{a \in V_x} \frac{1}{|O(a)|} \tag{3.2}$$

where V_x denotes the set of papers present in the path *x* including *k* and excluding the end node *i*. It is assumed here that the surfer walk resembles a Markov process. This means that the surfer has no memory of the path traversed before.

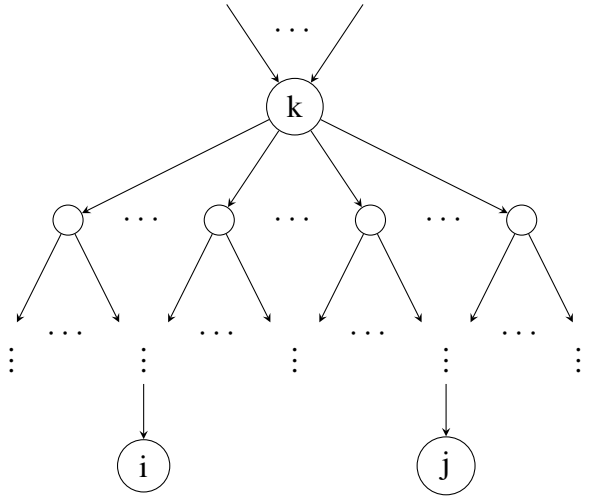


Figure 3.1: Description of a random walk

To circumvent the 'infinite expected distance problem' as discussed in (1), we define expected f-distance between *k* and *i* as follows:

$$d'(k,i) \; = \sum_{x:\; k \to i} P(x).\alpha^{l(x)} \tag{3.3}$$

We consider exponential function for the same reasons discussed in (1). α, a decay parameter is defined in the range of [0,1). It can be observed that the set of paths $\{x : k \rightarrow i\}$ have one-to-one correspondence with $\{x' : O(k) \rightarrow i\}$ and $l(x) = l(x') + 1$. Hence $P(x)$ can be expressed in a recursive form in terms of $P(x')$ as:

$$P(x) = \frac{1}{|O(k)|} \sum_{x' :\ O(k) \rightarrow i} P(x') \tag{3.4}$$

As a result, the following transformation for the distance metric can be worked out:

$$\begin{aligned} d'(k,i) &= \sum_{x :\ k \rightarrow i} P(x).\alpha^{l(x)} \\ &= \frac{1}{|O(k)|} \sum_{y=1}^{|O(k)|} \sum_{x' :\ O_y(k) \rightarrow i} P(x').\alpha^{l(x')+1} \\ &= \frac{\alpha}{|O(k)|} \sum_{y=1}^{|O(k)|} \sum_{x' :\ O_y(k) \rightarrow i} P(x').\alpha^{l(x')} \\ &= \frac{\alpha}{|O(k)|} \sum_{y=1}^{|O(k)|} d(O_y(x'), i) \end{aligned} \tag{3.5}$$

A recursive matrix expression for equation (3.5) is as follows:

$$\mathbf{D} = \alpha.\mathbf{W}_1^\top.\mathbf{D} + (1-\alpha)\mathbf{I}_n \tag{3.6}$$

where $[\mathbf{D}]_{ab}$ = *d(a,b)* and $\mathbf{I}_n$ is an n×n identity matrix. As we intend to calculate exponential raised to the length of path, the process can be viewed as propagation of influence of each node through the network. Thus **D** is initialized to $\mathbf{I}_n$ and the propagation from each node can be calculated iteratively using the above expression. To ensure that the score propagated over each iteration from a node is maximum, we add $(1-\alpha)\mathbf{I}_n$. A closed form expression can be obtained as well. By definition, each element in $\mathbf{W}_1$ is less than 1. So, each element in $\alpha\mathbf{W}$ is less than 1. Hence, $(\mathbf{I}_n - \alpha\mathbf{W}_1)$ is known to be invertible (28). Closed form expression of **D** is similar to the PageRank form and is given by:

$$\mathbf{D} = [\mathbf{I}_n - \alpha\mathbf{W}_1]^{-1} \tag{3.7}$$

Without loss of generality, we remove all the multiplied constants. We define the similarity metric between places *i* and *j* given that a surfer starts from place *k*, $s_k(i,j)$,

as the product of probabilities of the random surfer ending up at places *i* and *j* under the random walk with restart paradigm, i.e.,

$$\begin{aligned} s_k(i,j) &= d'(k,i).d'(k,j) \\ &= \left[(\mathbf{I}_n - \alpha\mathbf{W}_1)^{-1}\right]_{ki}\left[(\mathbf{I}_n - \alpha\mathbf{W}_1)^{-1}\right]_{kj} \\ &= \left[(\mathbf{I}_n - \alpha\mathbf{W}_1^\top)^{-1}\right]_{ik}\left[(\mathbf{I}_n - \alpha\mathbf{W}_1)^{-1}\right]_{kj} \end{aligned} \tag{3.8}$$

In a probabilistic scenario, for every occurrence of the following event, 'existence of two paths, one from *k* to *i* and the other from *k* to *j*, given the surfer is at state(paper) *k*' we have awarded a score proportional to the product of distance metrics of both paths, to the similarity measure and found the expected amount of the award over all the possible nodes *k*. We, however, didn't consider the occurrence of the surfer starting at node *k* in the first place. We propose that the occurrence of above event is proportional to place *k*'s centrality in the network. Suppose we know the centrality scores of the papers before hand, for example PageRank. Let c_a be the centrality score of place $a \in V$. We define a matrix $\mathbf{C}_1$ such that $[\mathbf{C}_1]_{aa} = c_a$ and off-diagonal elements to be zeros.

Hence *GSim* score between papers *i* and *j*, *s(i,j)* is defined as:

$$\begin{aligned} s(i,j) &= \sum_{\forall k} c_k.s_k(i,j) \\ &= \sum_{\forall k}\left\{\left[(\mathbf{I}_n - \alpha\mathbf{W}_1^\top)^{-1}\right]_{ik}\left[\mathbf{C}_1\right]_{kk}\left[(\mathbf{I}_n - \alpha\mathbf{W}_1)^{-1}\right]_{kj}\right\} \\ &= \left[(\mathbf{I}_n - \alpha\mathbf{W}_1^\top)^{-1}.\mathbf{C}_1.(\mathbf{I}_n - \alpha\mathbf{W}_1)^{-1}\right]_{ij} \end{aligned} \tag{3.9}$$

We define similarity score matrix, S_1, as follows:

$$\mathbf{S}_1 = \left[(\mathbf{I}_n - \alpha\mathbf{W}_1^\top)^{-1}.\mathbf{C}_1.(\mathbf{I}_n - \alpha\mathbf{W}_1)^{-1}\right] \tag{3.10}$$

where $[S_1]_{ij} = s(i,j)$. However, one must note the following issue with the above formulation. In an example network shown in figure 3.2, there is no node in the network, from which there exist paths to nodes 1 and 7, i.e., the above similarity measure can't capture similarities between recently added places onto the transportation network. Thus, to capture the above scenario, we'll have to consider destinations that we

can reach from nodes of interest as well, i.e., both 1 and 7 refer 2 as well as have an extended path to 6.

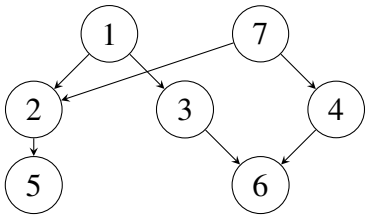


Figure 3.2: Example network

To address this issue, we propose an extension to the above similarity measure. We call this new measure as *GSim*. We consider a new network $G_2(V, E_2)$ where the directed edges in network *G(V,E)* are reversed and repeat the above mentioned procedure of computing similarity.

We can observe that adjacency matrix of network G_2, given by A_2, is the transpose of A_1. We obtain the following similarity score matrix for G_2:

$$\mathbf{S}_2 \;=\; \Big[(\mathbf{I}_n - \alpha\mathbf{W}_2^{\top})^{-1}.\mathbf{C}_2.(\mathbf{I}_n - \alpha\mathbf{W}_2)^{-1}\Big] \tag{3.11}$$

We define a comprehensive *GSim* matrix as follows:

$$\begin{aligned}\mathbf{S} \;&=\; \lambda.\mathbf{S}_1 + (1-\lambda)\mathbf{S}_2 \\ &= \lambda.\Big[(\mathbf{I}_n - \alpha\mathbf{W}_1^{\top})^{-1}.\mathbf{C}_1.(\mathbf{I}_n - \alpha\mathbf{W}_1)^{-1}\Big] \\ &+ (1-\lambda)\Big[(\mathbf{I}_n - \alpha\mathbf{W}_2^{\top})^{-1}.\mathbf{C}_2.(\mathbf{I}_n - \alpha\mathbf{W}_2)^{-1}\Big]\end{aligned} \tag{3.12}$$

where $\lambda \in$ [0,1] adjusts the relative weight between co-citation and bibliographic coupling influence on the similarity measure. Thus we provide a closed for expression for *GSim*.

3.4 Properties of GSim

3.4.1 Symmetry

As $\mathbf{C}$ is a symmetric matrix by definition, it can be observed that $\mathbf{S}_1$ and $\mathbf{S}_2$ are symmetric and thus making $\mathbf{S}$, a symmetric matrix.

3.4.2 Limited Information problem

One of the problems with *SimRank* is that the similarity between nodes with no incoming edges is defined to be zero, i.e., for newly added nodes to the network, similarity is given to be zero as there is limited information on these new nodes. This is called Limited Information problem. The main motivation for *P-Rank* was to overcome this problem. A similar version of formulation is also considered in *GSim* to come around the limited information on the new nodes.

3.4.3 Zero Similarity problem

As *SimRank* considers only the paths of equal length from nodes i and j, similarity between a parent and its child node is zero. This is called Zero Similarity problem. This problem is originally highlighted in *SimRank** (5). However, as *GSim* considers the two random walks independently, paths of different lengths are also taken into consideration, hence overcoming the issue of 'Zero Similarity'.

3.4.4 Scalability

Due to inherent parallel property of propagation of centrality of a node in the network, *GSim* can be effectively scale to networks with large set of nodes by efficiently using the memory space.

3.5 GSim computation algorithm

Algorithmic implementation of *GSim* has two steps - Propagation of centrality and calculation of similarity scores. Propagation of centrality of node over the network penetrating the centrality of a node through its neighborhood. We implement a limit on this penetration of neighborhood upto three hops. This penetration of influence of node *i* to a target node *j* can take place through multiple paths present between nodes *i* and *j*. Thus the overall influence of node *i* on node *j* is the sum of scores propagated through multiple paths. Thus every node *k* has a vector of these propagation scores from every node *j*. Once we have such scores, using equations 3.10 and 3.11, we calculate similarity scores between every pair of nodes in the network. Another implementation of *GSim* would be to directly use the readily available efficient matrix inversion packages for equation 3.12.

Algorithm 1 Propagation algorithm for a node *v*

```
1:  procedure SPREADPROCC(G, set1, hopNum, parentSet)
2:      while len(set1) > 0 do
3:          tempNode = set1.pop()
4:          if hopNum ≤ Thresh then
5:              set2 = G.neighbors(tempNode)
6:              parentSet(set2) = tempNode
7:              φ(tempNode)+ = α^hopNum / degree(parentSet(tempNode))
8:              φ = spreadProcc(G, set2, hopNum+1, parentSet)
9:          else
10:             break
11:     return φ
12:
13: global Thresh = 3                          ▷ Threshold for hops in neighborhood
14: procedure PROPAGATION(G, v)
15:     hopNum = 1
16:     for each w in V do
17:         φ(w) = 0                           ▷ Propagation value of node v
18:         parentSet(w) = v
19:     set1 = neighbors(v)
20:     φ = spreadProcc(G, set1, hopNum, parentSet)
```

The above algorithm 1 lays out the sequence for propagation of node *v*. *parentSet* is a dictionary with all the nodes as keys and their set of parent nodes is set as the value. We set the threshold over the number of hops to be considered for propagation to be three and initialize the node *v*'s spread component in the spread vector of each node to be zero. The graph **G**, one-hop neighbors of source node *v*, present hop number

and the *parentSet* corresponding to *set1* are given as inputs to function *spreadProcc*. *spreadProcc* is a recursive function which imitates Breadth First Search algorithm in its propagation traversal. Propagation value of every node in the neighborhood of node *v* is updated in *spreadProcc* function. This is recursed over every one-hop neighbor of node *v*. The number of recursion steps is limited to three. After all the recursions, *spreadProcc* function returns the three-hop propagation of node *v*'s influence.

3.6 Time and space complexity analysis

Let *k* be the maximum degree of a node in the network. The propagation step is similar to the propagation stage in *PageSim*. As we cap the number of recursions in the propagation stage of the algorithm to be 3 (in general, say, *r*), the expected size of the propagation vector at each node would be $O(k^r)$. Thus we can establish that the time complexity for performing propagation step for a node is $O(k^r)$. Once we have the *GSim* propagation scores and the centrality values of all the nodes, all we need to do is a simple multiplication (equation 3.11). In this multiplication stage, we know that we need not multiply an entire row of the first matrix with an entire column of the second matrix, as we know that only $O(k^r)$ values are non-zero. Thus, time complexity for this stage is $O(k^{2r})$. For computing *GSim* similarity values for all the n^2 node-pairs, time complexity would be $O(n^2k^r)$.

As we have discussed, space required to store propagation values of a node is $O(k^r)$. Thus, overall space complexity of *GSim* is $O(nk^r)$.

CHAPTER 4

Experimental results

In this section, we perform two experiments on *GSim* to understand its performance in comparison with *SimRank* and *PageSim*. We use optimized version of *SimRank* by employing several pruning techniques. As it is heavily time consuming to compute *SimRank* metric for every node pair on a network with more than 10000 nodes, we use synthetic datasets for the following experiments. Methods of network construction used for the two experiments are discussed in the respective subsections.

4.1 Time analysis

In this experiment, we measure the time performance of the three algorithms. We construct a simple direct Growing Network (GN) graph with number of nodes ranging from 100 to 2000. We use performance counter clock to measure time complexity of each metric. From Figure 4.1, we can observe that *GSim* performs very well compared to both *SimRank* and *PageSim* in terms of time taken to compute similarity measure for all possible node-pairs. The main reason for the better performance of *GSim* is it's simpler matrix formulation. As we have several high performing functions that can handle matrices, the overall time taken to compute similarity measures is lower. However, one needs to observe that as the number of nodes increases to the order of millions, the time measure shoots up drastically for all the algorithms.

We now consider a different graph generator model, namely, G(n,p) model. Here, every possible edge in an n-node network occurs with a probability of p. In this network, we study the affect of p on the time taken to compute all possible node-pair similarity measures, for all the three measures. Figure 4.2 shows the affect of increasing the probability value, p, on the time taken by all the three algorithms for a 200-node network. For every value of p, time complexity is averaged over 20 instances of graphs generated. It can be observed that the change in density of the network has a huge impact on the computational time complexity of *SimRank* compared to *PageSim* and not much

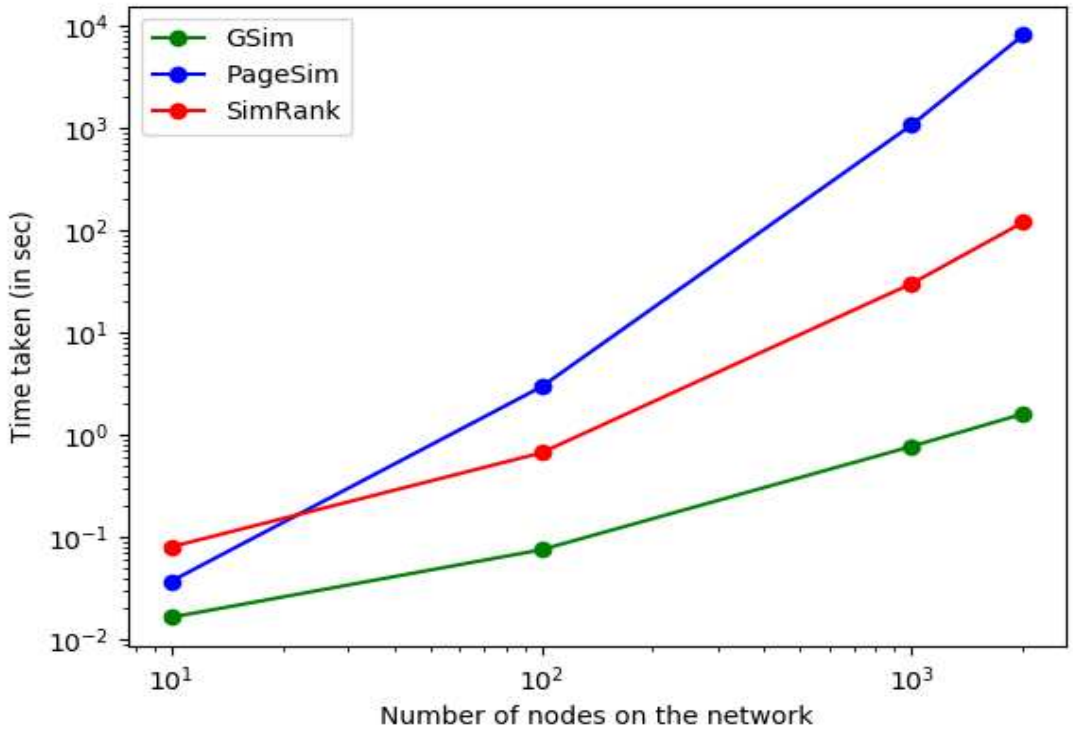


Figure 4.1: Time performance of *GSim*, *SimRank* and *PageSim*

visible affect on computation of *GSim*. One needs to note that the number of nodes in the network considered for this experiment is pretty small.

4.2 Cluster analysis

In order to establish that *GSim* actually traces the ground truth of similarity between the nodes, the following experiment is conducted. We consider the ground truth of similarity to be the belongingness of node pair to a single community. We generate a synthetic stochastic block model with 1000 nodes and 5 clusters. We assume to have the prior knowledge of the cluster to which each node belongs. Probability of occurrence of an edge is given by a stochastic matrix $\mathbf{[P]}_{5\times5}$, i.e., the probability of occurrence of an edge between node *i* belonging to cluster *a* and a node *j* belonging to community *b* is given by $\mathbf{[P]}_{ab}$. For this experiment, we assume that $\mathbf{[P]}_{aa} = \rho,\ \forall a$ and $\mathbf{[P]}_{ab} = (1-\rho)/4,\ \forall b \neq a$.

Given the stochastic block model, we calculate similarities between all possible

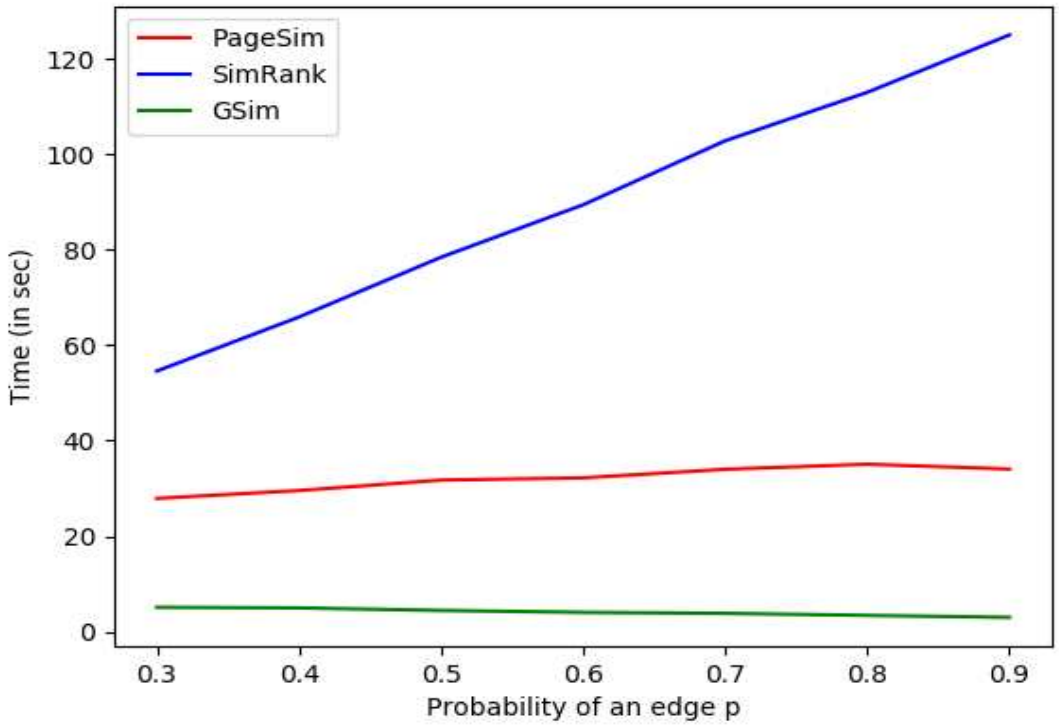


Figure 4.2: Time performance of *GSim*, *SimRank* and *PageSim* as the probability of edge occurrence increases

node-pairs for every measure. Now, we separate the node-pairs whose similarity value lies within the top $k\%$ of all values, for different values of k, as shown on the horizontal axis of Figure 4.3. Among these top $k\%$ similar node pairs, we count those node pairs (i, j), whose entities i and j belong to same cluster, as shown on the vertical axis of Figure 4.3. This counting is performed for various values of ρ. Each item in legend, for example, 'S0.4' in the Figure 4.3, corresponds to fraction of node-pairs with top $k\%$ *SimRank* similarity values that belong to same cluster, with parameter $\rho = 0.4$. Similarly, 'P' corresponds to *PageSim* and 'G' corresponds to *GSim*.

Two important observations can be noted from Figure 4.3. The first one is, as we expand the top similar node-pair net by increasing the value of k, one would assume that the similarity measure will output also those node-pairs which do not belong to same community. This can be seen in the performance of *GSim* and *SimRank* with an interesting exception by *SimRank*. For high values of ρ, *SimRank* is able to trace the true community structure of the network exceptionally well. However, for lower values of ρ, expected trend can be observed. *GSim* follows the expected trend throughout the range of ρ. It is important to note that the edges in real world networks have significant inter-

community links, i.e., possess low ρ values. So, we conclude that *GSim* will be able to follow the ground truth on par with established measures with an added advantage of low time complexity. *PageSim* does not seem to reflect the ground truth similarity measures very well. Particularly, the top 0.02% similar node-pairs seem to belong to different clusters.

From these three experiments, we have established that *GSim* has a considerable advantage over *SimRank* and *PageSim* in terms of time taken to compute all possible node-pair similarity measures. Also, in small networks, the increase in density of the network seems to be significantly impacting the time complexity of *SimRank* and not *PageSim* and *GSim*. Finally, we conclude that *GSim* and *SimRank* are able to better trace the ground truth similarity values as compared to *PageSim*. And, in a range of ρ, *SimRank* traces ground truth extremely well.

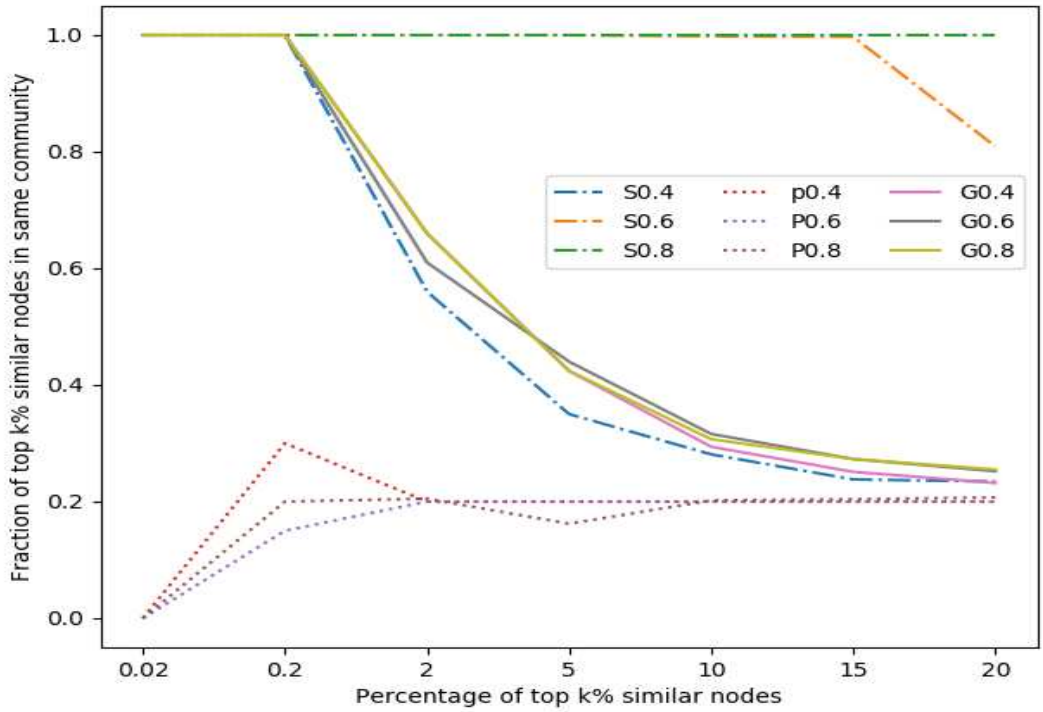


Figure 4.3: Cluster performance of top *k%* similar node-pairs as given by *GSim*, *SimRank* and *PageSim*

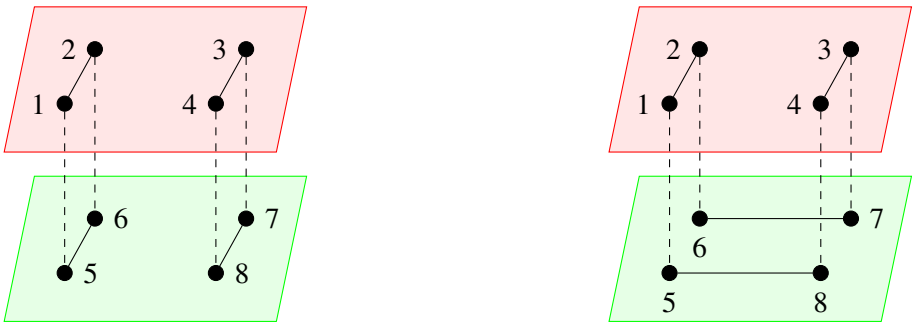
CHAPTER 5

GSim for multilayer networks

Increasingly, researchers are extending the theories and concepts for a single layer networks to multilayer networks. The motivation for this extension is predominantly that the real world networks can be better represented by multilayer networks. For example, consider transportation network of Manhattan, New York City. There are several modes of transportation, subway, public bus and cycle, available for a person to travel from place A to place B. Each mode of transport has its own network of pickup and drop points. Thus, transport network of Manhattan can be better represented by a combination of all these networks. Hence there is a need to extend all the theories that were proposed on single layer networks, to multilayer networks.

We try to extend *GSim* to multilayer networks for calculating similarity between any two nodes across the layers. We can club all the layers into a single network and compute similarities between nodes. However, we need to understand what this quantity reflects. We know that $GSim(i, j)$ gives the similarity between nodes i and j if both of them belong to same layer. We need to understand what does $GSim(i, j)$ mean if node i belongs to one layer and node j belongs to a different layer. In order to study this, let us define a new inter layer average similarity measure between two layers a and b, *avgInterLayerSim(a, b)* as the average of *GSim* similarity of evry possible node pair (i, j) such that node i belongs to layer a and node j belongs to layer b.

Consider a two scenarios of a 2-layer network with four nodes each with the edge structure as shown in Figure 5.1. Case (i) has same networks in both the layers, whereas, case (ii) has two complementary networks. The dashed lines, inter-layer edges, indicate the correspondence of nodes, i.e., node 1 and node 5 belong to same entity, but operate in two layers. The solid lines are the intra layer edges. When we consider the network as whole, the edge set includes both inter and intra layer edges. Now, consider the two layer network as a single network with 8 nodes and 8 edges. If we try to extend the random-walk based similarity measure *GSim* to the case (i), the similarity between



Case (*i*) Case (*ii*)

Figure 5.1: Two scenarios of multilayer networks

nodes 2 and 7 will be given as '0'. This is because, there is no other node in the entire network from which the random walker can traverse to both node 2 and node 7. However, in case (*ii*), a random walker starting from node 1 can reach both node 2 (path : 1→2) and node 7 (path : 1→5→8→4→3→7). Thus the *GSim* similarity between nodes 2 and 7 in case (*ii*) is non zero. With this insight, we can establish that *avgInterLayerSim(a, b)*, as defined above, is higher in case (*ii*) compared to case (*i*).

By definition, *GSim* uses the reachability of two nodes from every other node in the network as the relation to define similarity between two the two nodes. In the above multilayer construct, nodes 2 and 7 are reachable only if the structures of the network present in two layers are complementary to each other, thus improving the navigability of a random walker from one layer to other. This shows that *GSim* measures navigability between two layers. The phenomenon of navigability is very important in transportation networks and there are some past works which study navigability (29; 30; 31). A practical use case for this extension of *GSim* is, while designing a transportation network, the designer would want to design an under ground subway network in such a way that it augments the over ground bus network rather than end up as a redundant mode of transportation.

CHAPTER 6

Conclusion and future work

In this work, we provide a new random walk based similarity measure, *GSim*, between two nodes of a network. We study it's important properties and formulation. In many past works, similarity between two node *i, j* is defined from the perspective of them, for example, comparing the local structures around nodes *i* and *j*. We propose the similarity of two nodes *i* and *j* from the perspective of every other node in the network, i.e., two nodes are said to be similar if they are both reachable from another node *k* in the network. We observe that this inversion of defining similarity relation leads to a simple matrix formulation. We study the relative performance of *GSim* with other established similarity measures in terms of time and mimicking ground truth. We also provide motivation and use cases for extension of *GSim* to multilayer networks.

In this work, we have only considered small networks to study the performance of *GSim*. Hence, studying the relative performance of *GSim* on larger and a diverse set of networks would be an interesting future line of work. Another line would to study the impact of centrality measures that are used in the formulation of *GSim*. Study of *GSim* on large multilayer networks of different kind would be another line of future work.

REFERENCES

[1] Jeh, Glen, and Jennifer Widom. "SimRank: a measure of structural-context similarity." Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2002.

[2] Jiang, Jay J., and David W. Conrath. "Semantic similarity based on corpus statistics and lexical taxonomy." arXiv preprint cmp-lg/9709008 (1997).

[3] Becker, Hila, Mor Naaman, and Luis Gravano. "Learning similarity metrics for event identification in social media." Proceedings of the third ACM international conference on Web search and data mining. ACM, 2010.

[4] Zhao, Peixiang, Jiawei Han, and Yizhou Sun. "P-Rank: a comprehensive structural similarity measure over information networks." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009.

[5] Yu, Weiren, et al. "More is simpler: Effectively and efficiently assessing node-pair similarities based on hyperlinks." Proceedings of the VLDB Endowment 7.1 (2013): 13-24.

[6] Zhang, Mingxi, et al. "E-rank: A structural-based similarity measure in social networks." Web Intelligence and Intelligent Agent Technology (WI-IAT), 2012 IEEE/WIC/ACM International Conferences on. Vol. 1. IEEE, 2012.

[7] Shibata, Naoki, et al. "Comparative study on methods of detecting research fronts using different types of citation." Journal of the American Society for Information Science and Technology 60.3 (2009): 571-580.

[8] Küçüktunç, Onur, et al. "Recommendation on academic networks using direction aware citation analysis." arXiv preprint arXiv:1205.1143 (2012).

[9] Small, Henry. "Co-citation in the scientific literature: A new measure of the relationship between two documents." Journal of the American Society for information Science 24.4 (1973): 265-269.

[10] Kessler, Maxwell Mirton. "Bibliographic coupling between scientific papers." American documentation 14.1 (1963): 10-25.

[11] Lin, Zhenjiang, Michael R. Lyu, and Irwin King. "Matchsim: a novel neighbor-based similarity measure with maximum neighborhood matching." Proceedings of the 18th ACM conference on Information and knowledge management. ACM, 2009.

[12] Freidin, Robert, and Noam Chomsky. "The minimalist program." (1997): 571-582.

[13] Liu, Xinhai, et al. "Weighted hybrid clustering by combining text mining and bibliometrics on a large-scale journal database." Journal of the American Society for Information Science and Technology 61.6 (2010): 1105-1119.

[14] Janssens, Frizo, et al. "Hybrid clustering for validation and improvement of subject-classification schemes." Information Processing & Management 45.6 (2009): 683-702.

[15] Mihalcea, Rada, Courtney Corley, and Carlo Strapparava. "Corpus-based and knowledge-based measures of text semantic similarity." AAAI. Vol. 6. 2006.

[16] Hearst, Marti A., et al. "BioText Search Engine: beyond abstract search." Bioinformatics 23.16 (2007): 2196-2197.

[17] Manning, Christopher D., and Hinrich Schütze. Foundations of statistical natural language processing. Vol. 999. Cambridge: MIT press, 1999.

[18] Symeonidis, Panagiotis, and Eleftherios Tiakas. "Transitive node similarity: predicting and recommending links in signed social networks." World Wide Web 17.4 (2014): 743-776.

[19] Li, Cuiping, et al. "Fast computation of simrank for static and dynamic information networks." Proceedings of the 13th International Conference on Extending Database Technology. ACM, 2010.

[20] He, Guoming, et al. "Parallel SimRank computation on large graphs with iterative aggregation." Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2010.

[21] Yu, Weiren, Xuemin Lin, and Wenjie Zhang. "Fast incremental SimRank on link-evolving graphs." 2014 IEEE 30th International Conference on Data Engineering. IEEE, 2014.

[22] Xi, Wensi, et al. "Simfusion: measuring similarity using unified relationship matrix." Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2005.

[23] Chen, Mei, and Xiaoyun Chen. "Fast and Accurate Computation of Role Similarity via Vertex Centrality." Web-Age Information Management: 16th International Conference, WAIM 2015, Qingdao, China, June 8-10, 2015. Proceedings. Vol. 9098. Springer, 2015.

[24] Shi, Baoxu, Lin Yang, and Tim Weninger. "Forward Backward Similarity Search in Knowledge Networks." Knowledge-Based Systems (2016).

[25] Lin, Zhenjiang, Irwin King, and Michael R. Lyu. "Pagesim: A novel link-based similarity measure for the world wide web." Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence. IEEE Computer Society, 2006.

[26] Page, Lawrence, et al. "The PageRank citation ranking: bringing order to the web." (1999).

[27] Katz, Leo. "A new status index derived from sociometric analysis." Psychometrika 18.1 (1953): 39-43.

[28] Seneta, Eugene. Non-negative matrices and Markov chains. Springer Science & Business Media, 2006.

[29] Strano, Emanuele, et al. "Multiplex networks in metropolitan areas: generic features and local effects." Journal of The Royal Society Interface 12.111 (2015): 20150651.

[30] De Domenico, Manlio, et al. "Navigability of interconnected networks under random failures." Proceedings of the National Academy of Sciences 111.23 (2014): 8351-8356.

[31] De Domenico, Manlio, et al. "The physics of spreading processes in multilayer networks." Nature Physics 12.10 (2016): 901.