Studies in Computational Intelligence 597

Giuseppe Mangioni
Filippo Simini
Stephen Miles Uzzo
Dashun Wang *Editors*

Complex Networks VI

Proceedings of the 6th Workshop on Complex Networks CompleNet 2015

Springer

Studies in Computational Intelligence

Volume 597

Series editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland
e-mail: kacprzyk@ibspan.waw.pl

About this Series

The series "Studies in Computational Intelligence" (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the worldwide distribution, which enable both wide and rapid dissemination of research output.

More information about this series at http://www.springer.com/series/7092

Giuseppe Mangioni · Filippo Simini
Stephen Miles Uzzo · Dashun Wang
Editors

Complex Networks VI

Proceedings of the 6th Workshop on Complex Networks CompleNet 2015

Springer

Editors
Giuseppe Mangioni
Dip. di Ingegneria Elettrica, Elettronica e Informatica
Universita Catania
Catania
Italy

Filippo Simini
Faculty of Engineering
Dept. of Engineering Mathematics
University of Bristol
Clifton
United Kingdom

Stephen Miles Uzzo
New York Hall of Science
New York
USA

Dashun Wang
College of Information Sciences and Technology
Pennsylvania State University
University Park
USA

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-319-16111-2 ISBN 978-3-319-16112-9 (eBook)
DOI 10.1007/978-3-319-16112-9

Library of Congress Control Number: 2015933376

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

The International Workshop on Complex Networks – CompleNet (www.complenet.org) was initially proposed in 2008 with the first workshop taking place in 2009. The initiative was the result of efforts from researchers from the BioComplex Laboratory in the Department of Computer Sciences at Florida Institute of Technology, USA, and from the Dipartimento di Ingegneria Informatica e delle Telecomunicazioni, Università di Catania, Italia.

CompleNet aims at bringing together researchers and practitioners working on areas related to complex networks. In the past two decades we have been witnessing an exponential increase on the number of publications in this field.

From biological systems to computer science, from social systems and language to science of science, complex networks are becoming pervasive in many fields of science. It is this interdisciplinary nature of complex networks that CompleNet aims at addressing.

CompleNet 2015 was the sixth event in the series and was hosted by the New York Hall of Science, New York City, US, on March 25–27, 2015.

This book includes the some of the peer-reviewed works presented at CompleNet 2015. This year we received a record number of submissions in the history of CompleNet, 113 between abstracts and papers. Each submission was reviewed by at least 3 members of the Program Committee. Acceptance was judged based on the relevance to the symposium themes, clarity of presentation, originality and accuracy of results and proposed solutions. After the review process, 13 papers and 10 short papers were selected to be included in this book.

The 23 contributions in this book address many topics related to complex networks and can be categorized in the following major groups: (1) Analysis and models that focus on social systems, including social networks and social media; (2) Dynamical processes on networks such as diffusion, transportation and search processes; (3) New theory, models, and metrics of complex network, from network structure to growth to communities; (4) Biological networks and other health-related networks, and (5) Innovative applications of network science to other domains such as crime, terrorism, and more.

We would like to thank the Program Committee members for their time and effort during the refereeing process. This proceeding would not have been possible without their timely and careful reviews.

We deeply appreciate the efforts of our Keynote Speakers who greatly enriched CompleNet 2015 with their presentations and insights in the field of Complex Networks:

Hernàn Makse (City College of New York, US),
Chaoming Song (University of Miami, US),
Rèka Albert (Pennsylvania State University, US),
Alex Arenas (Universidad Rovira i Virgili, Spain),
Kathryn Coronges (West Point, US),
Cèsar Hidalgo (MIT Media Lab, US),
Mark Newman (University of Michigan, US),
Arun Sundararajan (Kaufman Management Center, US),
Ching-Yung Lin (IBM T.J. Watson Research Center)

Their presentations are among the reasons CompleNet 2015 was such a success.

Special thanks also go to the Local Organizers, Catherine Cramer and Marcia Rudy (New York Hall of Science, US), the Steering Committee, Giuseppe Mangioni (Universita Catania, Italy), Josè Mendes (University of Aveiro, Portugal) and Ronaldo Menezes (Florida Institute of Technology, USA).

New York City, USA
March 2015

Giuseppe Mangioni
Universita Catania, Italy

Filippo Simini
University of Bristol, UK

Stephen Miles Uzzo
New York Hall of Science, USA

Dashun Wang
Pennsylvania State University, USA

Contents

A Flexible Fitness Function for Community Detection in Complex Networks

Fabrício Olivetti de França[1] and Guilherme Palermo Coelho[2]

[1] Center of Mathematics, Computing and Cognition (CMCC), Federal University of ABC (UFABC), Santo André, SP, Brazil
folivetti@ufabc.edu.br

[2] Laboratory of Natural Computing (LCoN-FT), School of Technology (FT), University of Campinas (Unicamp), Limeira, SP, Brazil
guilherme@ft.unicamp.br

Abstract. Most community detection algorithms from the literature work as optimization tools that minimize a given *quality* (or *fitness*) *function*, while assuming that each node belongs to a single community. Although several studies propose fitness functions for the detection of communities, the definition of what a community is is still vague. Therefore, each proposal of fitness function leads to communities that reflect the particular definition of community adopted by the authors. Besides, such communities not always correspond to the real partition observed in practice. This paper proposes a new flexible fitness function for community detection that allows the user to obtain communities that reflect distinct characteristics according to what is needed. This new fitness function was combined with an adapted version of the immune-inspired optimization algorithm named cob-aiNet[C] and applied to identify (both disjoint and overlapping) communities in a set of artificial and real-world complex networks. The results have shown that the partitions obtained with the optimization of this new metric are more coherent (when compared to the real, known, partitions) than those obtained with one of the most adopted function from the literature: modularity.

1 Introduction

Complex problems from a wide range of fields can be theoretically modeled and described as complex networks [1–3]. In such networks, nodes that present similar properties often tend to be linked to each other, thus forming consistent subgraphs with dense interconnections that are called *communities* [4]. The detection of communities in complex networks is an important step in the multitude of possible analyses that can be performed to such models. When a complex problem is modeled as a network, the identification of communities may allow both the comprehension of characteristics that are specific to subgroups of nodes and of how such nodes interact with each other [5].

Several researchers devised theories to explain the emergence of communities in complex networks [6, 7]. However, in 2005, such studies about the emergence

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_1

of communities converged to a general theory known as *Preferential Attachment* [8]. According to Preferential Attachment, the emergence of a complex network starts with a single node, new nodes are introduced into the network iteratively and the probability that a given node will attach itself to another node is directly proportional to the importance of the latter (given by the number of connections it has).

Although communities play an important role in the network analysis, as they allow the identification of functional properties of a group of nodes and also of the ways complex behaviors emerge from simple individual functions, the formal definition of communities is still vague in the literature. Therefore, one of the challenges associated with the development of community detection algorithms is to select which metric should be used to properly evaluate whether a given set of nodes actually represent a community with characteristics that are relevant to the context of the problem [4]. Such metrics become even more important when considering that a large part of algorithms for community detection are based on the optimization of *quality functions* (or *fitness* functions, in the context of this paper).

Distinct quality functions were proposed in the literature, such as *Surprise* [9], the metric of Chira *et al.* [4] and *Modularity* [3]. From all metrics described in the literature, Modularity, which assumes that a community is a *module of the network* and that two nodes belonging to the same community tend to have much higher probability of being connected to each other than that of two nodes belonging to different communities [2], is one of the most adopted.

Given that each quality function may be intended to identify a set of communities according to one of the existing different definitions, the resulting partition of the original complex network invariably reflects the characteristics of such definitions. Besides, it is known that the optimization of some of these quality functions may lead to partitions of the network that do not correspond to the real partition observed in practice [10, 11].

The above scenario is even worse when *overlapping communities* are considered. Contrary to disjoint communities of complex networks, in which each node belongs to a single community, the partition of the network into overlapping communities allows some nodes (known as *bridge* nodes) to belong to different communities at the same time. In this context, aspects such as the *clustering coefficient*[1] of the network and the number of edges connecting a given node to its neighbors (both in and out of its community) may have different impacts when choosing a bridge node in complex networks that model different real-world situations.

Therefore, in this paper a new flexible fitness function for community detection is proposed. This new metric, named *Flex*, allows the user to predefine which characteristics should be present in the communities that will be obtained

[1] Clustering coefficient is a metric that evaluates the tendency of a given node being grouped with the other nodes of the network. It corresponds to the relation between the number of triangles formed by a given node and its neighbors and the number of all possible triangles that could be formed.

by the optimization process, thus allowing the identification of distinct sets of communities for the same complex network by simply adjusting a few intuitive parameters. Flex was combined with an adapted version of the immune-inspired optimization algorithm named cob-aiNet[C] [12] and applied to identify (both disjoint and overlapping) communities in a set of eight artificial and four real-world complex networks. The obtained results have shown that the partitions obtained with the optimization of this new metric are more coherent with the known real partitions than those obtained with the optimization of modularity.

This paper is organized as follows. The new flexible objective function for community detection will be presented in Section 2, together with some insights about how this objective function can be applied to identify overlapping communities and a brief description of the optimization algorithm adopted in this paper. The experimental methodology and the obtained results will be discussed in Section 3. Finally, some concluding remarks and indications for future work will be given in Section 4.

2 A Flexible Objective Function for Community Detection

In a broader definition, a community structure of a network is a partition of the nodes so that each partition is densely connected. The modularity metric [3] tries to capture this definition by analyzing the difference between the number of edges inside a community and the expected number of edges that would be observed if this community was formed in a random network. Although modularity is widely adopted by the complex network community, its structure may lead to the false assumption that the number of edges between two groups decreases as the network size increases. Therefore, for larger networks, a simple connection between two nodes of different communities may result in the merging of these two communities, in order to increase (maximize) modularity. This aspect is known as the *resolution limit* of the metric [13].

Additionally, in a situation in which a given node has few links connecting it to a small community and most of its links connecting it to a large community, the optimization of modularity will often include such node into the larger community, without considering the local contribution of this node to the smaller communitiy. This can be a drawback if, for example, the node has a higher clustering coefficient with respect to the smaller community than to the larger one.

With that in mind, a new quality function for community detection, hereby called *Flex*, is proposed. The optimization of Flex tries to balance two objectives at the same time: maximize both the number of links inside a community and the local clustering coefficient of each community. Additionally, it also penalizes the occurrence of open triangles (i.e., it minimizes the *random model* effect [14]).

The first step to calculate Flex for a given partition of the network is to define the *Local Contribution* of a node i to a given community c:

$$LC(i,c) = \alpha * \triangle(i,c) + (1-\alpha) * N(i,c) - \beta * \wedge(i,c), \tag{1}$$

where $\triangle(i,c)$ is the ratio between the transitivity of node i (number of triangles that i forms) inside community c and the total transitivity of this node in the full network, $N(i,c)$ is the ratio between the number of neighbors node i has inside community c and its total number of neighbors, and $\wedge(i,c)$ is the ratio between the number of open triangles in community c that contain node i and the total participation of i in the whole network. Variables α and β are weights that balance the importance of each term.

Since $\triangle(i,c)$ and $N(i,c)$ are related and their optimization tends to lead towards the generation of the same type of community, their contribution to Eq. 1 is balanced by an weighted average. By doing so, the user can specify which characteristics are more important when a node does not clearly belongs to any community. The penalization term avoids merging two communities that are connected by just a few edges.

Given the Local Contribution of all nodes to each community, it is also possible to define the *Community Contribution* (CC) of a community c in a given partition:

$$CC(c) = \sum_{i \in c} LC(i,c) - \frac{|c|}{|V|}^{\gamma}, \tag{2}$$

where $|.|$ is the number of elements in a set, V is the set of nodes of a network and γ is the penalization weight. The penalization in this equation is devised to avoid the generation of a trivial solution, in which the entire network forms a single community.

Finally, the Flex value of a given partition p is given by:

$$Flex(p) = \frac{1}{|V|} \sum_{c \in p} CC(c) \tag{3}$$

The weight parameter α directly dictates whether the optimization process will tend to insert a given node into a clustered community or into a community that contains the majority of this node's neighbors. It is important to consider both transitivity and the neighborhood of each node in the optimization, as both concepts are not necessarily related (i.e. a given node will not necessarily have high transitivity with the majority of its neighbors). Therefore, by weighting these two criteria, the user can emphasize each of them according to what is desirable in a given practical situation.

Finally, it is also important to highlight that the penalization term of Eq. 1 ensures that the convergence to the random model is penalized, even if α favors only the number of neighbors (i.e., $\alpha = 0$) in the definition of the communities.

2.1 Applying Flex to Identify Overlapping Nodes

The Flex fitness metric also provides insights about overlapping nodes. As the importance of transitivity and neighborhood is balanced by parameter α, this characteristic can be exploited to infer whether a given node should belong or not to more than one community.

When using Flex as an optimization function for community detection, some nodes may be more sensible than others to the weight α. This happens when, for example, a node has a fraction of its neighbors on a clustered community and the remaining neighbors spread across one or more communities with lower transitivity.

Therefore, a simple heuristic that allows the identification of overlapping nodes is to search for nodes that do not make significant contribution to one of the α-weighted factors (transitivity and neighborhood), i.e., nodes that are sensible to changes of α. After finding these nodes, we can allocate them to other communities that share a certain fraction of neighbors with them. This heuristic is summarized in Alg. 1.

Algorithm 1. Heuristic to find overlapping nodes

Data: thresholds $thr\triangle$ and $thrN$ for the contribution to transitivity and neighborhood, respectively, and threshold $thrSh$ of shared neighbors between communities.
Result: New set of communities with overlapping nodes.

```
for each node i do
    c = community that contains i
    if △(i, c) < thr△ or N(i, c) < thrN then
        for c_j ≠ c do
            if N(i, c_j) > thrSh then
                Add i to community c_j
```

2.2 The Cob-aiNet[C] Algorithm

As previously mentioned, an adaptation of the cob-aiNet[C] algorithm (*Concentration-based Artificial Immune Network for Combinatorial Optimization* – [12]) was adopted in this paper to obtain a set of communities for complex networks that maximize the new proposed quality function (Flex).

The cob-aiNet[C] algorithm, which was originally proposed to solve combinatorial optimization problems [12], was previously adapted to identify both disjoint and overlapping communities in complex networks [5]. As most of the adaptations proposed in [5] were adopted here as well, only a brief explanation of the general aspects of cob-aiNet[C] will be presented here, together with details about those aspects that differ from the adaptation proposed in [5]. For further details, the reader is referred to [12, 5].

The cob-aiNet[C] is a bioinspired search-based optimization algorithm that contains operators inspired in the natural immune system of vertebrates. Therefore, it evolves a population of candidate solutions of the problem (cells or possible partitions of the complex network), through a sequence of cloning, mutation and selection steps, guided by the fitness of each individual solution.

Besides these evolutionary steps, all the cells in cob-aiNet[C] population are compared to each other and, whenever a given cell is more similar to a better one than a given threshold, its concentration (a real value assigned to each cell) is reduced. This concentration can also be increased according to the fitness of the cell (higher fitness leads to higher concentration). Such concentration-based mechanism is an essential feature of the algorithm, as it controls the number clones that will be generated for each cell at each iteration, the intensity of the mutation process that will be applied to each clone and when a given cell should be eliminated from the population (when its concentration becomes null).

When compared to the adaptations made in [5], the only differences are associated with the new hypermutation operator, which will be discussed next, and the new approach to obtain overlapping communities described in Sect. 2.1.

The New Hypermutation Operator. To properly explain the new hypermutation operator, it is important to know that each cell in the population of the algorithm is represented as an array of integers with length equal to the number N of nodes of the complex network. Each position i of the array corresponds to a node of the network and assumes value $j \in \{1, 2, \ldots, N\}$ that indicates that nodes i and j belong to the same community.

The new hypermutation operator, which is applied to all cells in the population at a given iteration, is basically a random modification of the integer values in n_{mut} positions of the array that corresponds to a cell, being n_{mut} given by Eq. 4:

$$n_{mut} = \max\left[\text{round}(\beta(t) \cdot e^{-f_i^{Ag}(t) \cdot C^i(t)}), 1\right], \tag{4}$$

where $f_i^{Ag}(t) \in [0, 1]$ is the normalized fitness of cell i at iteration t, $C^i(t)$ is the concentration of cell i at iteration t, $\beta(t)$ is a parameter and round$(\cdot)$ returns the closest integer.

The n_{mut} positions of the cell that will suffer mutation are randomly selected, and so are the values that will be inserted into these positions. However, the probability that a given value k (associated with node k) replaces the current value in position i is directly proportional to $|N(i) \cap N(k)|$, where $N(i)$ is the set of nodes that are neighbors of i.

3 Experimental Results

In order to assess whether Flex is able to lead to gains in community detection, when compared to Modularity, an extensive experimental setup composed

of 8 artificial and 4 real world networks were devised here. The artificial networks, which were generated by the toolbox provided by Lancichinetti [15], are composed of 4 networks formed by high density communities (i.e. with a high number of internal edges), which facilitates the identification of the optimal partition, and 4 networks with noisy communities (i.e. with higher probability of presenting edges connecting them to other communities).

The artificial networks were generated with 50, 100, 200 and 500 nodes, being these nodes with average degree of 10 and maximum degree of 15. Such networks were generated with a maximum of 10 communities, 3 overlapping nodes belonging to an average of 2 communities and average clustering coefficient of 0.7. The mixing parameter, which introduces noise to the network structure, was set as 0.1 for the first set of networks (labeled Network $50 - 500$ in the tables that follow) and 0.3 for the second set (labeled Noise Network $50 - 500$).

The real world networks (with known partitions) that were chosen for the experiments were: Zachary's Karate Club, a social network of friendships between 34 members of a Karate Club [16]; Dolphins Social Network, a social network based on frequent associations between 62 New Zealandese dolphins [17]; American College Football, network of American College Football games during season Fall 2000 [18]; and a network of co-purchasing of books about US politics compiled by Krebs [19].

A total of 20 repetitions of the experiments were performed for each network for each fitness function adopted here (Flex and Modularity). The cob-aiNet[C] algorithm was empirically adjusted with the following parameters for all the experiments: $\sigma_S = 0.2$, maximum number of iterations equal to $1,500$, $\alpha^{Ini} = 10$, $\alpha^{End} = 1$, initial population with 4 candidate solutions and maximum population size of 6. After each run, the heuristic presented in Alg. 1 was applied to the best solution returned by cob-aiNet[C].

The results were evaluated by the average Normalized Mutual Information, which indicates how close a given partition of the network is from the real partition (ground truth) [5, 11]. In this work, the solutions with non-overlapping (labeled NMI in the tables that follow) and overlapping (labeled NMI OVER.) partitions were evaluated. Besides, the obtained results for overlapping partitions were also compared to those obtained with the technique proposed in [5] (labeled NMI MULTIMODAL).

The following tables (Tables 2 to 4) also report the evaluated fitness of the returned solutions (labeled FIT), the number of overlapping communities obtained with the proposed heuristic (labeled # Over.) and with the technique described in [5] (labeled # Over. Multimodal), the number of communities found (labeled # Comm.) and the total time taken to obtain the results (labeled TIME)[2].

3.1 Parameters

All the parameters required by Flex, presented in Table 1, were empirically defined here for groups of networks. Notice though that, in practice, these

[2] All the experiments were performed on an Intel Core i5 with 2.7GHz, 8GB of RAM and OSX 10.9.2.

Table 1. Weight parameters and heuristic thresholds for each dataset

Network	α	β	γ	$thr\triangle$	$thrN$	$thrSh$
Network 50, Karate	0.8	0.3	2	0.3	0.6	0.25
Network 100-500, Krebs	0.8	0.3	4	0.3	0.7	0.25
Noise Network 50-500	0.5	1.0	4	0.3	0.7	0.45
Football	0.8	0.6	4	0.3	0.6	0.25
Dolphins	0.4	0.3	4	0.3	0.6	0.25

parameters should be set depending on the goal of the network analysis (e.g. if partitions with highly clustered communities are required, α should be set with higher values).

Also, if the partition structure of the network is known *a priori*, the calibration of such parameters in order to obtain the known partition could indicate some characteristics of the network dynamics, such as, for example, the way that the connections of each node were established.

The results were statistically verified by means of the Kruskal paired test with significance < 0.05. Those that differ significantly are marked in bold in the tables.

3.2 Artificial Networks with Overlapping Communities

As expected, for the first set of networks (results given in Table 2), both Flex and Modularity obtained the same values of NMI for every experiment. This is due to the lower rate of noise adopted in the creation of such networks, which makes the identification of the real partitions trivial for most fitness functions. Notice though that the overlapping detection heuristic proposed here resulted in partitions with perfect NMI score (equal to 1.0) for every network considered, except Network 100. In this particular dataset, one of the overlapping nodes did not attend the criteria established by the heuristic, thus the obtained NMI was slightly lower than 1.0.

The results for the second set of networks, which were generated with higher noise, are reported in Table 3. In this scenario, the differences between Flex and Modularity become much more evident. In every situation the heuristic combined with Flex was able to find most of the overlapping nodes of each problem, as pointed out by the higher values of NMI. On the other hand, the method proposed in [5] obtained lower values of NMI by introducing many more (false) overlapping nodes into the partition. It is also noticeable that, in the presence of noise, Modularity also tends to find partitions with much less communities than Flex, which is due to the resolution limit discussed in Sect. 2.

3.3 Real-World Social Networks

Regarding the real-world social networks, the obtained results (given in Table 4) show that, again, Flex leads to a significant improvement over Modularity. However, it is important to notice that the ground truths of such networks are related to the classification of the nodes of these datasets according to their respective

Table 2. Results for the artificial networks with overlap and low level of noise

	Flex	Modularity
FIT:	0.81	0.60
NMI:	0.94	0.94
NMI OVER.:	1.00	1.00
NMI MULTIMODAL:	0.87	**0.93**
# Comm.:	5.00	5.00
# Over.:	3.00	3.00
# Over. Multimodal:	4.50	0.95
TIME (in seconds):	48.52	32.73

(a) Network 50

	Flex	Modularity
FIT:	0.81	0.59
NMI:	0.95	0.95
NMI OVER.:	0.98	0.98
NMI MULTIMODAL:	**0.96**	0.95
# Comm.:	4.00	4.00
# Over.:	2.10	2.00
# Over. Multimodal:	1.05	0.50
TIME (in seconds):	224.33	146.19

(b) Network 100

	Flex	Modularity
FIT:	0.85	0.64
NMI:	0.97	0.97
NMI OVER.:	1.00	1.00
NMI MULTIMODAL:	**0.98**	0.97
# Comm.:	5.00	5.00
# Over.:	3.00	3.00
# Over. Multimodal:	0.65	0.00
TIME (in seconds):	406.03	113.93

(c) Network 200

	Flex	Modularity
FIT:	0.87	0.80
NMI:	0.99	0.99
NMI OVER.:	1.00	1.00
NMI MULTIMODAL:	0.99	0.99
# Comm.:	11.95	12.00
# Over.:	2.95	3.00
# Over. Multimodal:	0.00	0.00
TIME (in seconds):	590.99	332.31

(d) Network 500

domains, so it does not necessarily mean that they actually correspond to the true partitions of the networks. Therefore, it is practically impossible to reach a perfect NMI score. It is also important to notice that those ground truths were originally devised without overlapping, so the NMI score with overlapping will always be smaller than the original NMI score.

Some interesting characteristics of each of these networks can be identified through a combination of visual inspection of the obtained partitions together with an analysis of the weights (α and β) adopted for Flex that led to the best values of NMI. From the obtained results, it is possible to infer that both the Karate Club and Krebs networks are formed by highly clustered communities ($\alpha = 0.8$), which makes sense as the Karate Club is a small social network prone to mutual friendships and the Krebs network, on the other hand, captures the interest of readers about particular subjects and, as such, they tend to buy only books that are related to their political views.

The Football network required the same value for α but a much higher value for β, which means that this particular network does not allow open triangles. The reason for that is due to the organization of tournaments that limit the occurrence of intra-cluster relationships. Finally, for the Dolphins network the required weights are more favorable to the establishment of inter-community relationships instead of clustering, which might be related to the hierarchy in their society that favors the creation of several hubs inside a community [17], thus raising the number of open triangles.

In order to illustrate the overlapping communities obtained by the combination of Flex and the proposed heuristic, Fig. 1 depicts the best partitions with overlapping nodes for each problem. In Fig. 1, the colors represent the communities found by the optimization of Flex, the shapes represent the communities

Table 3. Results for the artificial networks with overlap and high level of noise

	Flex	Modularity
FIT:	0.47	0.55
NMI:	**0.77**	0.43
NMI OVER.:	**0.78**	0.43
NMI MULTIMODAL:	**0.69**	0.42
# Comm.:	6.00	3.00
# Over.:	3.00	0.00
# Over. Multimodal:	12.80	1.60
TIME (in seconds):	58.77	73.34

(a) Noise Network 50

	Flex	Modularity
FIT:	0.51	0.49
NMI:	**0.93**	0.58
NMI OVER.:	**0.94**	0.58
NMI MULTIMODAL:	**0.82**	0.43
# Comm.:	6.00	4.30
# Over.:	0.40	1.45
# Over. Multimodal:	15.60	35.40
TIME (in seconds):	143.46	200.52

(b) Noise Network 100

	Flex	Modularity
FIT:	0.51	0.50
NMI:	**0.93**	0.60
NMI OVER.:	**0.94**	0.62
NMI MULTIMODAL:	**0.81**	0.47
# Comm.:	6.00	4.00
# Over.:	2.00	5.50
# Over. Multimodal:	34.65	40.55
TIME (in seconds):	655.99	657.93

(c) Noise Network 200

	Flex	Modularity
FIT:	0.50	0.63
NMI:	**0.89**	0.55
NMI OVER.:	**0.91**	0.56
NMI MULTIMODAL:	**0.73**	0.40
# Comm.:	13.95	7.65
# Over.:	10.30	6.70
# Over. Multimodal:	145.70	194.00
TIME (in seconds):	1818.17	2619.40

(d) Noise Network 500

Table 4. Results obtained for the real world networks

	Flex	Modularity
FIT:	0.82	0.41
NMI:	**0.95**	0.40
NMI OVER.:	**0.79**	0.45
NMI MULTIMODAL:	**0.91**	0.55
# Comm.:	1.95	3.95
# Over.:	1.90	2.90
# Over. Multimodal:	0.55	16.30
TIME (in seconds):	39.36	50.24

(a) Karate Club

	Flex	Modularity
FIT:	0.68	0.53
NMI:	**0.86**	0.46
NMI OVER.:	**0.82**	0.46
NMI MULTIMODAL:	**0.82**	0.46
# Comm.:	2.30	4.00
# Over.:	1.90	7.00
# Over. Multimodal:	4.60	0.00
TIME (in seconds):	63.88	27.61

(b) Dolphins

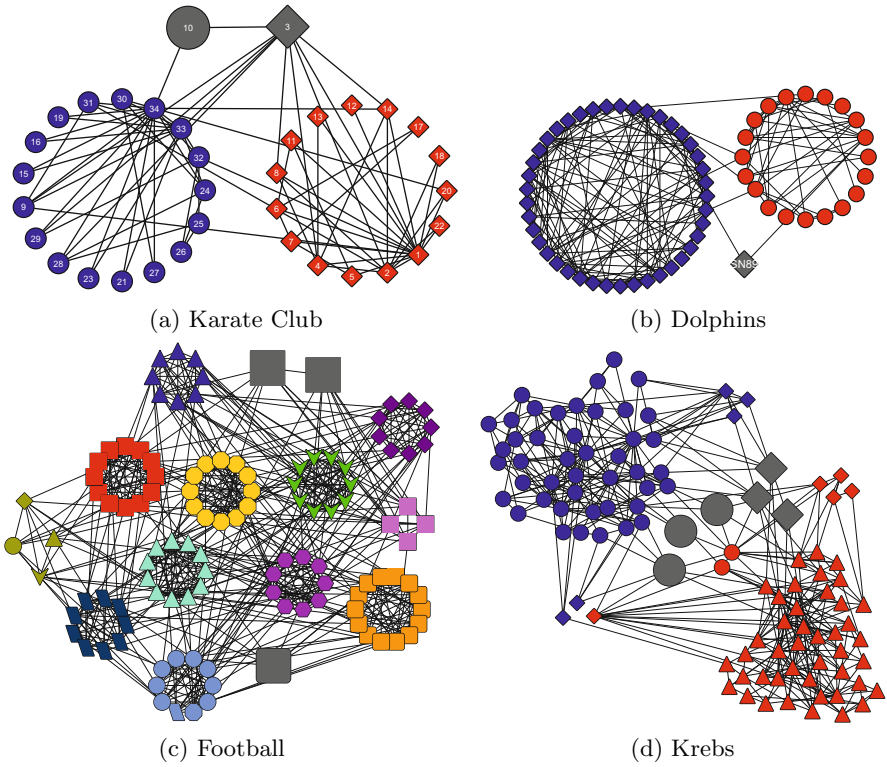
	Flex	Modularity
FIT:	0.77	0.60
NMI:	**0.74**	0.67
NMI OVER.:	**0.72**	0.66
NMI MULTIMODAL:	**0.74**	0.67
# Comm.:	11.25	9.60
# Over.:	2.25	2.05
# Over. Multimodal:	0.00	0.00
TIME (in seconds):	23.36	26.31

(c) Football

	Flex	Modularity
FIT:	0.72	0.53
NMI:	**0.45**	0.32
NMI OVER.:	**0.43**	0.37
NMI MULTIMODAL:	**0.45**	0.32
# Comm.:	2.25	4.95
# Over.:	7.30	11.90
# Over. Multimodal:	6.20	0.75
TIME (in seconds):	126.01	47.33

(d) Krebs

in the ground truth and the larger nodes are the overlapping nodes. It is visually noticeable that the structure of communities found using Flex makes sense, given the weights for each network. Also, every overlapping node is clearly positioned between two or more distinct groups.



(a) Karate Club (b) Dolphins

(c) Football (d) Krebs

Fig. 1. Results obtained by cob-aiNet[C] with Flex for the real-world networks

4 Conclusion

In this paper a novel fitness function for community detection in complex networks was introduced, together with a heuristic that allows the identification of overlapping nodes, based on particular characteristics of this function, and a novel mutation operator for the immune-inspired algorithm adopted in the optimization process. This new fitness function, called Flex, is parametrized in such a way that it can be adapted to obtain communities with different characteristics.

Through an extensive experimental setup, it was possible to verify that this new fitness function and heuristic are capable of leading to partitions close to the ground truth of a set of networks with different characteristics.

As for future investigations, we intend: (*i*) to explore Flex with other search-based algorithms, such as the Louvain Method; (*ii*) to perform further comparisons with other overlapping community detection algorithms, in order to evaluate possible differences among the identified overlapping nodes; and (*iii*) to devise a thorough complexity analysis of the metric and evaluate its performance in large-scale networks.

Acknowledgment. This research is funded by FAPESP 2014/06331-1.

References

1. Barabasi, A., Frangos, J.: Linked: The New Science of Networks. Perseus Books Group, New York (2002)
2. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. Physical Review E(99) (2002)
3. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Proceedings of the National Academy of Sciences of the United States of America 99(12), 7821–7826 (2004)
4. Chira, C., Gog, A., Iclanzan, D.: Evolutionary detection of community structures in complex networks: A new fitness function. In: Proc. of the 2012 IEEE Congress on Evolutionary Computation (CEC), pp. 1719–1725. IEEE (2012)
5. de França, F.O., Coelho, G.P.: Identifying overlapping communities in complex networks with multimodal optimization. In: Proc. of the 2013 IEEE Congress on Evolutionary Computation (CEC), pp. 269–276. IEEE (2013)
6. Simon, H.A.: On a class of skew distribution functions. Biometrika, 425–440 (1955)
7. Merton, R.K.: The Matthew effect in science. Science 159(3810), 56–63 (1968)
8. Newman, M.E.J.: Power laws, Pareto distributions and Zipf's law. Contemporary Physics 46(5), 323–351 (2005)
9. Aldecoa, R., Marín, I.: Deciphering Network Community Structure by Surprise. PLoS One 6(9), e24195 (2011)
10. Lancichinetti, A., Fortunato, S.: Community detection algorithms: a comparative analysis. Physical Review E 80(5), 056117 (2009)
11. Lancichinetti, A., Radicchi, F., Ramasco, J.J., Fortunato, S.: Finding statistically significant communities in networks. PloS One 6, e18961 (2011)
12. Coelho, G.P., de França, F.O., Von Zuben, F.J.: A concentration-based artificial immune network for combinatorial optimization. In: Proc. of the 2011 IEEE Congress on Evolutionary Computation, CEC (2011)
13. Fortunato, S., Barthelemy, M.: Resolution limit in community detection. Proceedings of the National Academy of Sciences 104(1), 36–41 (2007)
14. Eerdös, P., Rényi, A.: On random graphs I. Publ. Math. Debrecen 6, 290–297 (1959)
15. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. Physical Review E 78(4), 046110 (2008)
16. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)
17. Lusseau, D., Schneider, K., Boisseau, O.J., Haase, P., Slooten, E., Dawson, S.M.: The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behav. Ecology and Sociobiol. 54(4), 396–405 (2003)
18. Evans, T.: Clique graphs and overlapping communities. Journal of Statistical Mechanics: Theory and Experiment 2010(12), P12037 (2010)
19. Krebs, V.: The social life of books, visualizing communities of interest via purchase patterns on the WWW (2004)

Finding Network Motifs Using MCMC Sampling*

Tanay Kumar Saha and Mohammad Al Hasan

Department of Computer and Info. Science
Indiana University-Purdue University
Indianapolis, Indiana, USA
{tksaha,alhasan}@cs.iupui.edu

Abstract. Scientists have shown that network motifs are key building block of various biological networks. Most of the existing exact methods for finding network motifs are inefficient simply due to the inherent complexity of this task. In recent years, researchers are considering approximate methods that save computation by sacrificing exact counting of the frequency of potential motifs. However, these methods are also slow when one considers the motifs of larger size. In this work, we propose two methods for approximate motif finding, namely SRW-rw, and MHRW based on Markov Chain Monte Carlo (MCMC) sampling. Both the methods are significantly faster than the best of the existing methods, with comparable or better accuracy. Further, as the motif size grows the complexity of the proposed methods grows linearly.

1 Introduction

Studying the local topology is an important step for modeling the interaction among the entities in a network. In a seminal work around a decade ago, Shen-orr et al. [14] hypothesized that network motifs play an important role in carrying out the key functionalities that are performed by the entities in a biological network. Since then, researchers have also discovered that network motifs are building block for complex networks from many diverse disciplines including biochemistry, neurobiology, ecology, engineering [11], proteomics [1], social sciences [6] and communication [5].

Finding network motifs is computationally demanding. To identify whether a given subgraph topology is a motif, we need to count the topology's frequency in the input network as well as in many randomized networks. Counting a topology's frequency in a single network is a challenging task as it requires solving subgraph isomorphism, a known $\mathcal{NP}$-complete problem. As the size of the motif grows, the number of candidate motifs increases exponentially, and the task becomes more challenging. To cope with the enormous computation cost of exhaustive counting of the frequency of candidate motifs, researchers consider various sampling based methods that obtain an approximation of relative frequency measure (which we call concentration) over all the candidates of a given size. Most notable among these methods are MFinder [8], MODA [12], and RAND-ESU [16]. Besides these approximate methods, exact motif counting methods are also available, such as, GTrieScanner [13], ESU [16], Grochow-Kellis algorithm [4],

* This research is supported by Mohammad Hasan's NSF CAREER Award (IIS-1149851).

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_2

Kavosh [7], and NetMODE [9]; However, their application is limited to small networks only. In this work, our focus is on finding concentration of prospective motifs using a novel sampling based method.

The quality of a sampling based method depends on three critical performance metrics: accuracy, convergence, and execution time. Existing sampling based methods are poor in one or more of the above performance metrics. For instance, MFinder is costly and it scales poorly with the size of the desired motifs. Authors in [16] have shown that the cost of subgraph sampling of MFinder increases exponentially with the size (number of vertex) of the subgraph. It is also poor in terms of accuracy and convergence. A similar method, RAND-ESU [16] is significantly faster than MFinder and yet its scalability is also not that satisfactory. Besides, its sampling accuracy and convergence behavior are also poor.

Another important fact about the existing sampling based methods is that they require random access to any of the vertices or the edges in the networks. This becomes a severe limitation for networks for which such unrestricted access is not available. For an instance, consider the Web network or a hidden network, a user may not have access to any arbitrary vertex/edge in the input network for security reason; rather, the desired node can only be accessed from another node which is one-hop away from it; such scenarios are common in real-life and are considered in the task of snowball sampling [3]. None of the existing methods can be used for finding motifs in a graph that only allows restricted access, such as crawling.

In this work, we propose two random walk based methods, namely MHRW (Metropolis-Hastings random walk) and SRW-RW (Simple Random Walk with Reweighting) for approximating the concentration of arbitrary-sized pattern graphs in a large network. The underlying mechanism of both the methods is a Monte Carlo Markov Chain (MCMC) sampling over the candidate motif space, which is guaranteed to compute an unbiased estimate of concentration of all the candidate motifs of a given size simultaneously. Since, our methods are based on random walk over the edges of the input graph, they only require a restricted access over the network such that at any given time of the walk the one-hop neighboring nodes of currently visiting candidate are accessible. Besides, the methods are scalable and are significantly faster than the existing methods. They also have better convergence property and small memory footprint. While preparing for the final manuscript of this work, we have found another work [15], where the authors propose methodologies that are similar to our work.

2 Background

2.1 Graph, Subgraph, Induced Subgraph

Let $G(V,E)$ is a *graph*, where V is the set of vertex and E is the set of edges. Each edge $e \in E$ is denoted by a pair of vertices (v_i, v_j) where, $v_i, v_j \in V$. A graph without a self-loop or multi edge is a simple graph. In this work, we consider simple, connected, and undirected graphs.

A graph $G' = (V', E')$ is a *s*ubgraph of G (denoted as $G' \subseteq G$) if $V' \subseteq V$ and $E' \subseteq E$. A graph $G' = (V', E')$ is a *v*ertex-induced subgraph of G if G' is a subgraph of G, and for any pair of vertices $v_a, v_b \in V'$, $(v_a, v_b) \in E'$ if and only if $(v_a, v_b) \in E$. In other words,

a *vertex-induced* subgraph of G is a graph G' consisting of a subset of G's vertices together with all the edges of G whose both endpoints are in this subset. In this paper, we have used the phrase *induced subgraph* for abbreviating the phrase vertex-induced subgraph. If G' is an induced subgraph of G and $|V'| = p$, we call G' a *p-subgraph* of G. An *embedding* of a graph G' in another graph G is a subgraph S of G such that S and G' are isomorphic;

For a given vertex count, the number of distinct graph topologies is fixed. We use the symbol Λ_p to denote the set of all such topologies. To denote one specific topology in Λ_p we use the symbol $\omega_{p,q}$, where q is the order of that topology (considering an arbitrary but fixed ordering) among all the size p topologies. The set of induced embeddings of all graphs in Λ_p in graph G is the collection of p-subgraphs of G. Figure 1 shows all the elements of the sets Λ_3, Λ_4 and Λ_5. Using the order of the topologies in this figure, $\omega_{3,1}$ is the 3-node line graph.

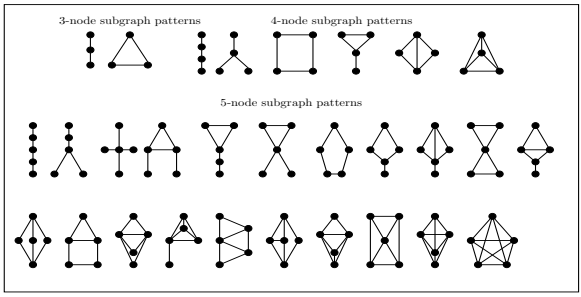


Fig. 1. All 3, 4 and 5 node topologies

2.2 Subgraph Concentration

The frequency of a particular p-subgraph topology g in an input graph G is the number of times it appears in G. We denote it by $f_G(g)$. The concentration of g in G is $C_G(g)$, which is defined as the normalized frequency over the cumulative frequency of all the subgraph topologies in the set Λ_p. Mathematically,

$$C_G(g) = \frac{f_G(g)}{\sum_{h \in \Lambda_p} f_G(h)} \tag{1}$$

2.3 Motif

A Motif is a subgraph topology which occurs in an input network at a significantly higher frequency than it occurs in a set of random networks with identical characteristics. For this purpose, the random networks are generated from the input network by imposing the constraint that the vertices of a random network has the identical degree distribution as that of the input network. There are several methods for generating random networks with identical degree distribution, but the most popular is the switching algorithm [10], which we use in this work. The significance of frequency deviation between the input network and the set of random networks is typically measured using z-score and p-value. If $\overline{f_{G_r}(g)}$ is the mean frequency of g in a set of randomized

graphs G_r (constructed from G), and $\sigma_{G_r}(g)$ is the corresponding standard deviation, then z-score of g for the input network G is defined as:

$$z_G(g) = \frac{f_G(g) - \overline{f_{G_r}(g)}}{\sigma_{G_r}(g)} \tag{2}$$

If the z-score of g is greater than some pre-specified threshold then we call g a motif. Since, setting this threshold requires domain expertise, all the existing motif finding methods consider it as a run-time parameter; we also follow the same in our work. For sampling based solution, we use concentration of subgraph instead of their frequency. Hence, z-score is defined as below:

$$\widehat{z}_G(g) = \frac{\widehat{C}_G(g) - \overline{\widehat{C}_{G_r}(g)}}{\widehat{\sigma}_{G_r}(g)} \tag{3}$$

In equation 3, we use $\widehat{C}_G$, and $\widehat{\sigma}_G$ to denote that they are statistics obtain from random sample of size-p embeddings.

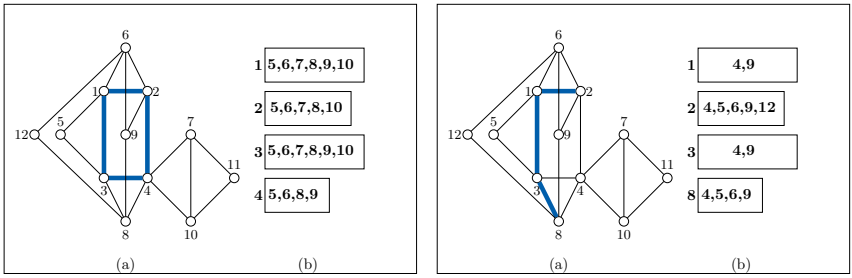
2.4 Markov Chains, and Metropolis-Hastings (MH) Method

A Markov chain is the sequence of Markov process over the state space S. The state-transition event is guided by a matrix, T, called *transition probability matrix*. The chain is said to reach a stationary distribution π, when the probability of being in any particular state is independent of the initial condition, it is reversible if it satisfies the *reversibility condition* $\pi(i)T(i,j) = \pi(j)T(j,i), \forall i,j \in S$ and it is *ergodic* if it has a stationary distribution. The main goal of the MH is to draw samples from some distribution $\pi(x)$, called the *target distribution*, where, $\pi(x) = f(x)/K$; here K is a normalizing constant which may not be known and difficult to compute. It can be used together with a random walk to perform MCMC sampling. For this, the MH algorithm calculates the *acceptance probability* using the following equation:

$$\alpha(x,y) = \min\left(\frac{\pi(y)q(y,x)}{\pi(x)q(x,y)}, 1\right) \tag{4}$$

3 Methods

Given a graph G (which we refer as input graph) and an integer p, a sampling based method samples a small set of p-subgraphs of G. From this set, it approximates the concentration of each topology in Λ_p as shown in section 2.3. To measure the exact concentration, one must perform unbiased sampling, where each of the p-subgraphs has an uniform probability to be sampled. This is not an easy task, as the sample space is very large. Besides, a direct sampling method is not applicable because for that we need to enumerate all the p-subgraphs (to obtain the size of the sample space), which we want to avoid. So, an indirect sampling strategy must be followed. Both MFinder [8] and RAND-ESU [16] adopt indirect sampling; however, they differ in the sampling



(a) Left: A graph G with the current state of random walk; Right: Neighborhood information of the current state (1,2,3,4)

(b) Left: The state of random walk on G (Figure 2(a)) after one transition; Right: Updated Neighborhood information

Fig. 2. Neighbor generation mechanism

methodologies. MFinder's sampling is biased which requires post-adjustment of concentration for correcting the bias; on the other hand, RAND-ESU guaranty a uniform sampling which requires no correction. For large p, both MFinder and RAND-ESU are costly.

In this paper, we propose MHRW, and SRW-RW for sampling p-subgraphs of a graph using Markov chain Monte Carlo (MCMC) sampling. As a Metropolis-Hasting based method (discussed in sec: 2.4), they perform a random walk over the state space so that the stationary distribution of the random walk converges to a desired target distribution. For our task, the state space are the set of p-subgraphs. Since, we want to approximate the concentration of each of the topologies in Λ_p, our target distribution is *uniform*, i.e., we want to sample each of the p-subgraphs with an identical probability. If $\mathcal{P}$ is the set of the p-subgraphs in the input graph G, and π is the target distribution, we want $\pi(g) = 1/|\mathcal{P}|, \forall g \in \mathcal{P}$.

```
MHRW(G, p, N)
    G: Input Graph
    p: Subgraph Size
    N: Size of the sample set, |S|

 1  g = Starting State
 2  M = φ
 3  i = 0
 4  d_g = Neighbor count of g
 5  while i < N
 6      h = Any neighbor of g chosen
            uniformly at random from (1, |d_g|)
 7      d_h = Neighbor count of h
 8      accp_val = d_g / d_h
 9      accp_probablility = min(1, accp_val)
10      if uniform(0,1) ≤ accp_probability
11          g = h
12          d_g = d_h
13      i = i + 1
14      Generate the Canonical code of g
15      Insert the code into the set M
        and update the count
16  Normalize the frequency using equation 5,
    ∀_i ω_{p,i} ∈ M
17  return M.
```

Fig. 3. MHRW Pseudocode

For the random walk of both MHRW and SRW-RW, a neighbor of a p-subgraphs (say, g) is obtained by simply replacing one of its existing vertices of g with another vertex which is not part of g and find the subgraph induced by the new vertex-set.

While replacement, the methods ensure that the new set of vertices induce a connected p-subgraph. At every iteration, all possible neighbors are populated using the above strategy. For a state, the number of neighboring states are called its *degree*.

Example: Suppose our sampling method (MHRW or SRW-RW) is sampling a 4-subgraph from the graph G shown in Figure 2(a)(Left). Let, the 4-subgraph $\langle 1,2,3,4\rangle$ (shown in bold lines) is the existing state of this random walk. One of it's neighbor state is $\langle 1,2,3,8\rangle$, which can be obtained by replacing the vertex 4 by the vertex 8. In Figure 2(a)(Right) we show the information of all its neighbors. Box labeled by x contains all the vertices that can be used as a replacement of vertex x to get a neighbor. If the random walk transition chooses to go to the neighbor state $\langle 1,2,3,8\rangle$, it can do so simply by adding the vertex 8 (a vertex in the box labeled by 4) and deleting the vertex 4. The updated state of the random walk along with the updated neighbor-list is shown in Figure 2(b). The degree of a state is the number of neighbors, which is simply the sum of the entries in each of the boxes; thus the degree of state $\langle 1,2,3,4\rangle$ is 21, and the degree of the state $\langle 1,2,3,8\rangle$ is 13. ■

To apply MH algorithm, we also need to decide on a proposal distribution, q. For MHRW random walk, we choose the proposal distribution to be uniform, i.e., in the proposal step MHRW chooses one of g's neighbors uniformly. If $h \in \mathcal{P}$ and h is a neighbor of g based on our neighborhood definition, using proposal distribution, the probability of choosing h from g, $q(g,h) = 1/d_g$, where d_g is the degree of the state g. Also note, if $m \in \mathcal{P}$, but m is not a neighbor of g, $q(g,m) = 0$, i.e., transitions are allowed among neighboring states only.

Using the proposal (q) and target (π) distributions, MHRW method is simply an implementation of the algorithm that we discussed in Section 2.4. A pseudo-code of MHRW is given in Figure 3. At the beginning of the sampling for each topology in Λ_p, we assign a counter which is initialized to 0. As the sampling progress, for each state we identify the specific topology that the state represents, and increment its counter by 1. Thus, if $\mathcal{S}$ is the sample set, the concentration equation defined in 1 for g where $g \in \Lambda_p$ becomes:

$$\widehat{C}(g) = \frac{1}{|\mathcal{S}|} \sum_{x \in \mathcal{S}} 1_{(x==g)} \tag{5}$$

At any iteration from the current stage g, the method chooses one of its neighbors, (say, h) using the proposal distribution (uniform), and either accept or reject the proposed move using Equation 4 i.e. MHRW adjusts the transition probability by accepting or rejecting the proposed transition so that the target distribution is guaranteed to be uniform.

On the other hand, an iteration of SRW-RW (simple random walk with re-weighting) simply chooses one of the neighbors uniformly and make this transition. Thus the difference between MHRW and SRW-RW is that the latter chooses the proposed transition with 100% probability. This does not guarantee uniform sampling of the states (p-subgraphs); rather the states are sampled in proportional to their degree values. In other words, the target distribution of simple random walk is directly proportional to the degree value of the p-subgraphs. So, the concentration of the topologies in Λ_p is also biased in proportional amount. To obtain an unbiased estimate of concentration, the estimated concentration should be re-weighted, which gives the name simple

random walk with re-weighting or in short SRW-RW. After re-weighting the concentration equation (Equation 1) of SRW-RW takes the following form:

$$\widehat{C}(g) = \frac{1}{W} \sum_{x \in \mathcal{S}} (1/d_x)_{(x==g)} \tag{6}$$

where, W is the sum of the total weights, i.e., $W = \sum_{x \in \mathcal{S}} (1/d_x)$. Such an idea of re-weighting has been used in [2] for approximating degree distribution of a large network by sampling.

Pseudo-code of SRW-RW is similar to the pseudo-code of Figure 3, the only difference is that, there is no acceptance rejection step and in Line 12, instead of incrementing the frequency count by 1, we increment the concentration by $1/d_g$. Finally, we normalize in Line 13 using equation 6 instead of equation 5.

Claim: For a given p and an input graph G, both MHRW and SRW-RW returns an unbiased estimate of the concentration of a topology in Λ_p.

Proof: Assume $g \in \Lambda_p$ is an arbitrary topology and $\mathcal{S}$ is a set of induced subgraph sampled from G. The expectation of g's concentration in G is $E\left[\widehat{C}(g)\right] = E\left[\frac{1}{|\mathcal{S}|}\sum_{x \in \mathcal{S}} 1_{(x \cong g)}\right] = E\left[P_u(x \cong g)\right]$. Here, $P_u(x \cong g)$ is the probability that a graph x in the sample set $\mathcal{S}$ is isomorphic to the topology g when it is sampled under uniform distribution. But, this value is the exact concentration value of g. So, $E\left[\widehat{C}(g)\right] = E\left[C_g\right] = C_G$. So, MHRW returns an unbiased estimate of the concentration of a topology in Λ_p.

By construction, the stationary distribution π for SRW-RW's random walk is proportional to the degree of a p-subgraph. Thus, for an arbitrary p-subgraph, w, its stationary probability $\pi(w) = d_w/K$ where K is a normalizing constant. For a topology $g \in \Lambda_p$, before re-weighting the expected value of its concentration is equal to $\sum_{w \in \mathcal{P}} \pi(w) \cdot 1_{(w \cong g)} = \sum_{w \in \mathcal{P}} \frac{d_w}{K} \cdot 1_{(w \cong g)}$. However if each sample w of type g contributes only $1/d_w$ instead of 1 in the counter of g, the expected value of concentration becomes $\sum_{w \in \mathcal{P}} \frac{d_w}{K} \cdot (\frac{1}{d_w})_{(w \cong g)} = \frac{1}{K}\sum_{w \in \mathcal{P}} 1_{(w \cong g)} = \frac{1}{K}C(g)$, which is the unbiased concentration scaled by a multiplicative constant. Since the concentration of all the topologies in Λ_p sums to 1, the expected value of the concentration returned by equation 6 after normalization is an unbiased estimate of the true concentration. ■

3.1 Implementation issues

Starting State. When we start the random walk on G, both MHRW, and SRW-RW starts from an arbitrary p-subgraph. To find it, the methods randomly choose an edge (of G) and include other adjacent edges to form an induced subgraph of desired size. As the input graph is connected, this process returns a p-subgraph of G.

Canonical Label of a Graph. We use *min-dfs-code* [17] for canonical labeling of the graph to unify different isomorphic forms of the same graph.

4 Results and Discussion

We implement MHRW and SRW-RW in C++ language and perform a set of experiments for evaluating their performance. We run all the experiments in a computer with 2.60 GHz processor and 4 GB RAM running Linux operating system. For experiments, we use graphs of different sizes from different domains. Table 1 lists the graphs along with the vertex count, the edge count and the average degree. Since the existing implementation of our methods only consider undirected graphs, all the input graphs are made undirected if necessary. The graphs are available from the following two web sites[1].

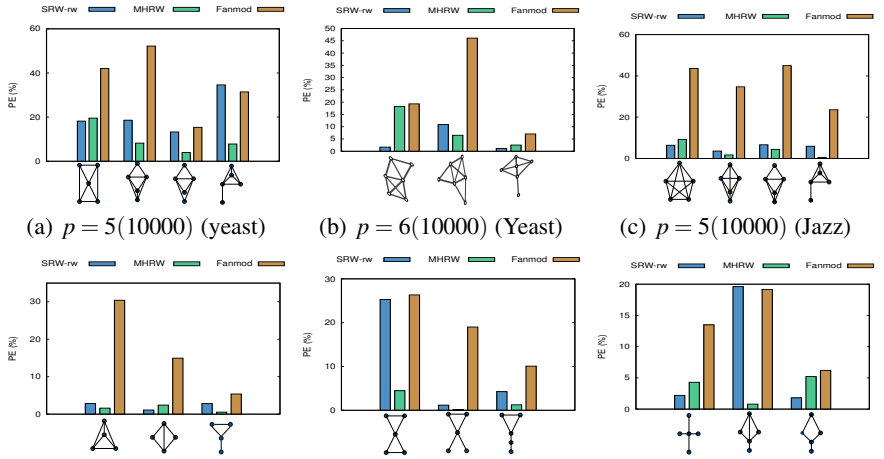
Table 1. Dataset Statistics

Graph	Vertex	Edge	Average Degree
Yeast	2,224	6,609	5.94
Jazz	198	2,742	27.49
ca-GrQc	4,158	13,422	6.43
ca-HepTh	8,638	24,806	5.74
ca-AstroPh	17,903	196,972	22.0

Experimental results in the earlier works show that RAND-ESU is the best among these three methods. In [16], Wernicke have shown that RAND-ESU is significantly faster than MFinder with a better accuracy. Another recent work [12] shows that RAND-ESU is the fastest among a set of methods including MODA. In this paper, we compare the performance of our methods with RAND-ESU to show that our methods are better than RAND-ESU in different performance metrics. We also considered MODA [12] for a comparison, but we found that its available implementation is unstable; the same fact was also reported by the authors of [9]. Note that we do not compare our methods with existing exact algorithm as they do not scale with the size of motif and also with the size of the input graph. For comparison with RAND-ESU, we use the implementation by authors that is available in the FANMOD library. Note that, in this implementation, the algorithm supports subgraph size up to 8. Besides a user need to set some probability values, which we set using the recommendation in FANMOD's documentation. In the result section, we will refer RAND-ESU as FANMOD following the convention in the earlier works.

We use three performance metrics: runtime, error, and convergence to compare our method with others. To compute the error value for a topology g, we first find the exact concentration of g using an exact method, then we find the approximate concentration using the sampling based method; the absolute difference between the above two concentration normalized by the actual concentration is the error for the topology g. However, since the sampling method is a randomized process, instead of using the approximate concentration of a single run, we take the average of the approximate concentration of 10 different runs. We represent the error as percentage and use the symbol $PE(g)$ (percentage error of g) for this metric.

[1] `http://snap.stanford.edu/data/index.html` and `http://www-personal.umich.edu/~mejn/netdata`



(a) $p = 5(10000)$ (yeast) (b) $p = 6(10000)$ (Yeast) (c) $p = 5(10000)$ (Jazz)

(d) $p = 4(40000)$ (Ca-AstroPh) (e) $p = 5(10000)$ (Ca-Hepth) (f) $p = 5(10000)$ (Ca-GrQC)

Fig. 4. Comparison of Percentage Error value for various methods. The dataset name, motif size, and the number of samples (in parenthesis) are given in figure sub-title.

4.1 Error Comparison

We compare the error percentage (PE) of various topologies using SRW-RW, MHRW, and FANMOD algorithms on all the datasets for different size values (p). Instead of showing the PE for all the topologies, we only show it for the topologies that are likely to be motifs, i.e., for these topologies, the $\widehat{z}_G(g)$ value in Equation 3 is the highest among all the topologies. For this experiment, we fixed the number of samples to 10000 for all of the experiments except for the experiment of Ca-AstroPh dataset, where we use 40000 samples.

For all the datasets, we see that our methods are significantly better than the FANMOD method based on the PE metric. Specifically, the performance gap between our method and FANMOD is very high for the Ca-AstroPh dataset, which is the largest among all our datasets. The performance of SRW-RW and MHRW are comparable. However, we observe that for topologies for which the concentration is high, MHRW's approximation is better than SRW-RW. On the other hand for graphs for which the concentration is small (see the dense topologies in Figure 4(b)), SRW-RW's approximation is better than MHRW. There are a few occasions where the PE of SRW-RW are as bad as FANMOD; nevertheless, the plots clearly demonstrate the superiority of Markov Chain based techniques over FANMOD in terms of percentage error.

4.2 Runtime Comparison

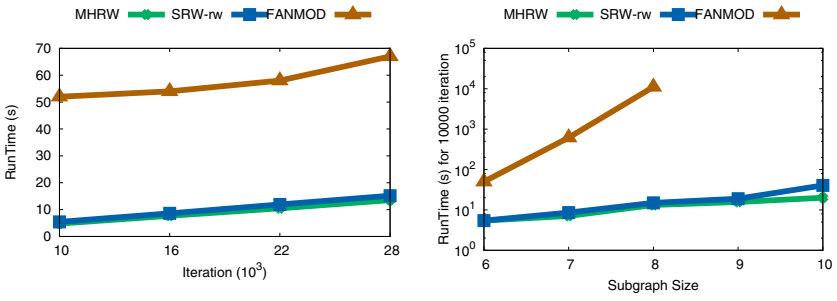
The runtime performance comparison of our methods with FANMOD is shown in Table 2. Here, we have fixed the sample count to 10000 for all the methods. To highlight the poor scalability of FANMOD with the size of the motif, we show some of the numbers in bold font. If we carefully observe the table we can see that as the size increases

by unity the runtime of FANMOD increases more than 10 times. For the Ca-AstroPh dataset which is the densest, for generating 10000 samples, FANMOD takes 180*s*, on the other hand both of our methods take about 5 seconds only. For this metric also, the performance gap between our methods and FANMOD increases as the dataset or the motif size increases.

We also show the runtime performance of the algorithms with the increasing number of samples in Figure 5(a) for yeast dataset and for subgraph size 5. The time increases mostly linearly for all the datasets; however, both of our methods have much smaller runtime than FANMOD. We also compare the runtime performance of the algorithms for motif sizes from 6 to 10. The result is shown in Figure 5(b) (note that *y*-axis is in logarithm scale). It is clear from the plot that our methods scale well with the increasing subgraph size. But, for FANMOD the runtime grows exponentially with the subgraph size; for example, to sample 10000 graphs from the yeast dataset, for subgraph size 7 and 8, it takes 616 seconds and 3 hours respectively. On the other hand, for size 8 our methods sample identical number of graphs in only 50 seconds. Also note that, FANMOD runs only for subgraph size up to 8.

Table 2. Runtime comparison of our methods with FANMOD

Dataset	Motif Size	MHRW (s)	SRW-rw (s)	FANMOD (s)
Yeast	5	2.73	3.13	2.73
	6	4.78	5.43	**50**
Jazz	5	5.08	5.71	3.45
	6	9.68	10.92	**52**
Ca-GrQC	3	0.79	1.06	0.026
	4	2.11	2.79	0.275
	5	7.03	10.53	2.79
	6	25.36	32.30	**34**
Ca-Hepth	3	0.60	0.75	0.43
	4	1.43	1.72	0.413
	5	3.03	3.30	5.37
	6	4.98	5.13	**70.41**
Ca-Astroph	3	3.20	4.48	3.35
	4	7.90	9.80	**180.38**



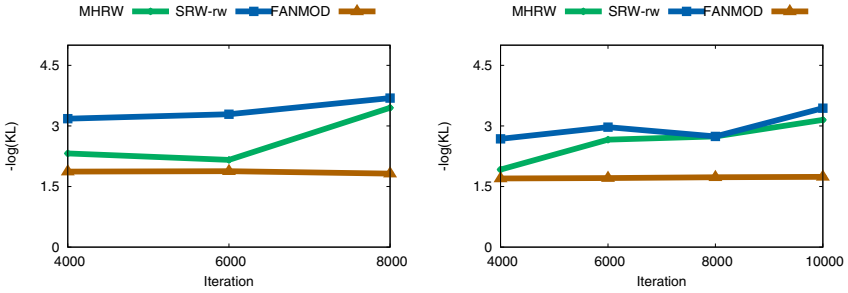
(a) Sample size vs Runtime (b) Subgraph Size vs Runtime

Fig. 5. Runtime performance for different sample sizes and for different subgraph sizes

4.3 Convergence Comparison

In this experiment, we study the convergence using the negative log (KL) metric by varying the number of samples. Figure 6(a) and 6(b) show that as we increase the

number of samples both the Markov chain based techniques approximate the concentration distribution more accurately (increasing value of $-\log(KL)$), on the other hand, for FANMOD the curve is almost flat, i.e. with an increasing number of samples FANMOD does not converge to the true concentration.



(a) Iteration vs KL (Size 5) (ca-Hepth) (b) Iteration vs KL (Size 6) (ca-Hepth)

Fig. 6. Comparison of convergence trend of our methods with FANMOD using KL Divergence

5 Conclusion

In this paper, we propose two methods MHRW, and SRW-RW for approximating the concentration of p-subgraphs in a host network for any given value of p. Our experimental results demonstrates that both of our proposed methods are significantly faster than the best of the existing methods. Moreover, our methods do not require full access over the networks. This makes our method useful for very large network (such as, Web) which can only be crawled.

References

1. Albert, I., Albert, R.: Conserved network motifs allow protein–protein interaction prediction. Bioinformatics 20(18), 3346–3352 (2004)
2. Gjoka, M., Kurant, M., Butts, C.T., Markopoulou, A.: Walking in Facebook: A Case Study of Unbiased Sampling of OSNs. In: Proc. of IEEE INFOCOM, pp. 1–9 (2010)
3. Goodman, L.A.: Snowball sampling. Ann. Math. Statist. 32, 148–170 (1961)
4. Grochow, J.A., Kellis, M.: Network motif discovery using subgraph enumeration and symmetry-breaking. In: Speed, T., Huang, H. (eds.) RECOMB 2007. LNCS (LNBI), vol. 4453, pp. 92–106. Springer, Heidelberg (2007)
5. Itzkovitz, S., Alon, U.: Subgraphs and network motifs in geometric networks. Physical Review E, Statistical, Nonlinear, and Soft Matter Physics
6. Juszczyszyn, K., Kazienko, P., Musiał, K.: Local topology of social network based on motif analysis. In: Lovrek, I., Howlett, R.J., Jain, L.C. (eds.) KES 2008, Part II. LNCS (LNAI), vol. 5178, pp. 97–105. Springer, Heidelberg (2008)
7. Kashani, Z., Ahrabian, H., Elahi, E., Nowzari-Dalini, A., Ansari, E., Asadi, S., Mohammadi, S., Schreiber, F., Masoudi-Nejad, A.: Kavosh: a new algorithm for finding network motifs. BMC Bioinformatics 10(1), 318 (2009)

8. Kashtan, N., Itzkovitz, S., Milo, R., Alon, U.: Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. J. Bioinformatics 20(11), 1746–1758 (2004)
9. Li, X., Stones, D.S., Wang, H., Deng, H., Liu, X., Wang, G.: Netmode: Network motif detection without nauty. PLoS One 7(12) (December 2012)
10. Milo, R., Kashtan, N., Itzkovitz, S., Newman, M.E.J., Alon, U.: On the uniform generation of random graphs with prescribed degree sequences (May 2004)
11. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science 298, 824–827 (2002)
12. Omidi, S., Schreiber, F., Masoudi-Nejad, A.: MODA: an efficient algorithm for network motif discovery in biological networks. Genes and Genetic Systems 84(5), 385–395 (2009)
13. Ribeiro, P., Silva, F.: G-tries: an efficient data structure for discovering network motifs. In: Proc. ACM Symp. on Applied Computing, pp. 1559–1566 (2010)
14. Shen-Orr, S.S., Milo, R., Mangan, S., Alon, U.: Network motifs in the transcriptional regulation network of escherichia coli. Nature Genetics 31, 1061–4036 (2002)
15. Wang, P., Lui, J., Ribeiro, B., Towsley, D., Zhao, J., Guan, X.: Efficiently estimating motif statistics of large networks. ACM Trans. Knowl. Discov. Data 9(2) (2014)
16. Wernicke, S.: Efficient detection of network motifs. IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(4), 347–359 (2006)
17. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: Proc. of 2nd International Conference on Data Mining, pp. 721–724. IEEE Computer Society (2002)

Analysis of the Robustness of Degree Centrality against Random Errors in Graphs

Sho Tsugawa[1] and Hiroyuki Ohsaki[2]

[1] University of Tsukuba, Tsukuba, Ibaraki 305-8573, Japan
s-tugawa@cs.tsukuba.ac.jp
[2] Kwansei Gakuin University, Sanda, Hyogo 669-1337, Japan
ohsaki@kwansei.ac.jp

Abstract. Research on network analysis, which is used to analyze large-scale and complex networks such as social networks, protein networks, and brain function networks, has been actively pursued. Typically, the networks used for network analyses will contain multiple errors because it is not easy to accurately and completely identify the nodes to be analyzed and the appropriate relationships among them. In this paper, we analyze the robustness of centrality measure, which is widely used in network analyses, against missing nodes, missing links, and false links. We focus on the stability of node rankings based on degree centrality, and derive Top_m and $\mathrm{Overlap}_m$, which evaluate the robustness of node rankings. Through extensive simulations, we show the validity of our analysis, and suggest that our model can be used to analyze the robustness of not only degree centrality but also other types of centrality measures. Moreover, by using our analytical models, we examine the robustness of degree centrality against random errors in graphs.

1 Introduction

Research on network analysis, which is used to analyze large-scale and complex networks such as social networks, protein networks, and brain function networks, has been actively pursued [1,6,8,20–22]. In network analysis, relationships among entities in the real world are represented by a graph. In social network analysis (SNA), individuals are represented as nodes in a graph, and the social ties among them, such as similarities, social relations, interactions, and flows, are represented as links [6, 22]. In brain function network analysis, brain regions are represented as nodes, and temporal correlations in activity among them are represented as links [20].

Among various indices proposed for network analysis, centrality measures (e.g., degree centrality, betweenness centrality, closeness centrality, and eigenvector centrality) [4, 11] have been widely used in actual analyses [3, 5, 25]. Centrality measures are indices that express the influence of one node on others, and such measures have been used for various purposes, such as discovering which person plays a central role in a community [3, 5] and inferring which brain regions are important for the task of interest [25].

Typically, the graphs used for network analyses will contain multiple errors because it is not easy to accurately and completely identify the entities to be analyzed and the appropriate relationships among them [7, 9, 14–16, 19]. For instance, graphs used in SNA

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_3

can contain several errors of different types, such as *missing nodes*, *missing links*, and *false links*. In traditional SNA, graphs are generated from the results of questionnaires, and so non-responses and inaccurate answers will cause such errors [24]. Even in recent SNA used for analyzing online social networks, such errors can be present due to sampling bias and restrictions on social network data, which is typically accessed by means of application programming interfaces. In biological network analyses, such as analyses of protein interaction networks and gene regulatory networks, graphs often contain errors such as missing links and false links as a result of measurement errors [19, 23].

Several analyses on the robustness of centrality measures used for network analyses against errors in the graphs (simulated as noise created by random addition and deletion of nodes and links) have been performed [7,9,12–17,19]. In [7,16], how centrality measures of nodes in networks are affected by the random addition and deletion of nodes and links is experimentally investigated. Robustness of centrality measures against link weight noises has also been experimentally investigated, such as in [13, 17].

Most existing studies use an experimental approach to understand the robustness of centrality measures, but some recent studies adopt a theoretical approach. Ghoshal *et al.* [12] analyze node-ranking stability based on the PageRank algorithm against random rewiring of links. Platig *et al.* [19] develop an analytical model to quantify the robustness of degree centrality against link errors (i.e., missing links and false links). They derive correlation coefficients r between the degree measures of the ground-truth graph and those of graphs with errors.

Our study builds on prior work and contributes to developing an analytical model that can be used to quantify the robustness of centrality measures. Since one of the most typical errors in network analysis is missing nodes [7, 9, 24], we extend the model of [19] to include these, and analyze the robustness of degree centrality against missing nodes as well as against missing links and false links. As discussed in the previous works [7, 19], centrality measures are used mainly for node ranking. We therefore focus on the stability of node ranking and derive Top_m and $\mathrm{Overlap}_m$, which evaluate the robustness of node rankings [7, 16, 17, 19]. Through extensive simulations, we show the validity of our analysis. Moreover, by using our analytical models, we examine the robustness of centrality measures against random errors in graphs.

The remainder of this paper is organized as follows. Section 2 introduces related work. In Section 3, we analyze the robustness of degree centrality against three types of errors (i.e., missing nodes, missing links, and false links). Section 4 examines the validity of our analysis through comparison between numerical examples of our analysis and results of simulations, and also discusses the robustness of centrality measures against random errors in graphs. Finally, Section 5 contains our conclusions and a discussion of future work.

2 Related Work

Most existing studies use a simulation to understand the robustness of centrality measures by adding errors to a ground-truth graph and investigating the relation between the centrality measures of the ground-truth graph and those of the graphs with errors [7, 9, 14–16]. In contrast, some recent studies use a theoretical approach [12, 19].

Ghoshal *et al.* [12] analytically derive the conditions under which node ranking according to PageRank is stable against random rewiring of links. Platig *et al.* [19] investigate the robustness of centrality measures against link errors (i.e., missing links and false links) through simulations and theoretical analysis. In their analysis, the correlation coefficients r of degree centrality between a ground-truth graph and graphs with errors are derived.

The only type of error studied in Ghoshal *et al.* [12] is link rewiring, and typical errors such as node and link addition and deletion are not considered. Platig *et al.* [19] investigate the robustness of centrality measures against link errors typical in network analysis, but the effect of node deletion, which is also a typical error in network analysis [7, 9, 24], is not studied. Moreover, the stability of node ranking based on centrality measures is investigated in their simulations, but a theoretical analysis of the node ranking stability is not performed. The correlation coefficients r of centrality measures, which are theoretically analyzed in [19], and Top_m and $\mathrm{Overlap}_m$, which are studied in this paper, exhibit different tendencies [19]. In this paper, we extend the model of [19] and use this extended model to analyze the robustness of centrality measures, as measured by node ranking stability based on degree centrality, against missing nodes, missing links, and false links.

3 Analysis

We analyze the consistency of node ranking based on degree centrality, comparing an undirected unweighted graph $G = (V,E)$ with a graph G_e that is a copy of G with random errors introduced. We analyze the robustness of degree centrality against three types of errors, which correspond to the following operations: link deletion, node deletion, and link addition. The link deletion error independently deletes each link of graph G with probability α; the node deletion error independently deletes each node in graph G and all links associated to that node with probability β; and the link addition error randomly adds $\gamma|E|$ links to graph G, where $|E|$ is the number of links in graph G. We assume that the graph G has an arbitrary degree distribution [18], and that the degree of each node in graph G $(k_1, k_2, \ldots, k_{|V|})$ is known, where $|V|$ is the number of nodes in graph G.

We rank all the nodes in graphs G and separately in G_e by sorting the nodes in descending order of their degree centrality, and we analyze the node ranking consistency between graphs G and G_e. We particularly focus on the ranking of highly ranked nodes, and derive expected values of Top_m and $\mathrm{Overlap}_m$, which are used to evaluate the robustness of node ranking. Top_m is the probability that the most central node in graph G is ranked in the top m most central nodes in graph G_e [7, 16, 17]. $\mathrm{Overlap}_m$ is the overlap between the top m most central nodes in graph G and those in graph G_e. More specifically, let $U_m(G)$ be the set of the m most central nodes in graph G; then, $\mathrm{Overlap}_m$ is defined as $|U_m(G) \cap U_m(G_e)|/m$ [7, 16, 17, 19]. These measures are used in the simulation studies [7, 16, 17, 19] to evaluate the robustness of centrality measures. Table 1 shows the definitions of symbols used in this paper.

Let $p(l|k)$ be the probability that a node with degree k in graph G has degree l in graph G_e. We derive Top_m and $\mathrm{Overlap}_m$ by using $p(l|k)$. In what follows, v_i denotes a node whose degree is the ith largest in graph G, and k_i denotes the degree of node v_i.

Table 1. Definitions of symbols used in this paper

G	Unweighted undirected graph
G_e	Unweighted undirected graph with errors
V	Set of nodes in graph G
E	Set of links in graph G
k_i	Degree of a node whose degree is the ith largest in graph G
v_i	Node whose degree is the ith largest in graph G
V_i	Subset of V defined as $V - \{v_i\}$
α	Probability of deleting each link in graph G
β	Probability of deleting each node in graph G
γ	Ratio of links added to graph G
$p(l\|k)$	Probability that a node with degree k in graph G has degree l in graph G_e
$P(l\|k)$	Probability that a node with degree k in graph G has degree l or less in graph G_e
$\overline{P}(l\|k)$	Probability that a node with degree k in graph G has degree more than l in graph G_e
$t_{i,j}$	Probability that node v_i has the jth largest degree in graph G_e
Top_m	Probability that node v_1 is ranked in the top m most central nodes in graph G_e
$\mathrm{Overlap}_m$	Overlap between the top m most central nodes in graph G and those in graph G_e

To obtain Top_m and $\mathrm{Overlap}_m$, we first obtain the probability that node v_i has the jth largest degree in graph G_e, which is denoted as $t_{i,j}$. First, let us consider $t_{i,1}$, which is the probability that node v_i has the largest degree in graph G_e. Node v_i has the largest degree in graph G_e if and only if the degree of each node is less than or equal to k_i, and therefore $t_{i,1}$ is given by

$$t_{i,1} = \sum_{l=0}^{|V|-1} p(l|k_i) \prod_{r \neq i} P(l|k_r), \tag{1}$$

where $P(l|k)$ is the probability that a node with degree k in graph G has degree l or less in graph G_e; this is given by the following equation.

$$P(l|k) = \sum_{s=0}^{l} p(s|k) \tag{2}$$

Next, let us consider the case with $j > 1$. Node v_i has the jth largest degree in graph G_e if and only if $(j-1)$ nodes have a higher degree than node v_i in graph G_e and all other nodes have a weakly lower degree than node v_i. Here, we define the following symbols.

$$\overline{P}(l|k) = 1 - P(l|k) \tag{3}$$

$$Q(l, k_i, S) = \begin{cases} \overline{P}(l|k_i) & v_i \in S \subset V \\ P(l|k_i) & \text{otherwise} \end{cases} \tag{4}$$

$$V_i = V - \{v_i\} \tag{5}$$

Then, $t_{i,j}$ is given by

$$t_{i,j} = \sum_{l=0}^{|V|-1} p(l|k_i) \sum_{X \in \binom{V_i}{j-1}} \prod_{r \neq i} Q(l, k_r, X), \tag{6}$$

where $\binom{V_i}{K}$ is the set of all subsets of V_i which have a given size K.

Top_m is the sum of the probability that node v_1 has the largest degree in graph G_e, the probability that node v_1 has the second largest degree in graph G_e, …, and the probability that node v_1 has the mth largest degree in graph G_e. Symbolically,

$$\mathrm{Top}_m = \sum_{j=1}^{m} t_{1,j}. \tag{7}$$

Additionally, we define $T_{i,j}$ as follows.

$$T_{i,j} = \sum_{s=1}^{j} t_{i,s} \tag{8}$$

Since the expected number of overlapping nodes between the top m most central nodes in graph G and those in graph G_e is $\sum_{i=1}^{m} T_{i,m}$, $\mathrm{Overlap}_m$ is then given by

$$\mathrm{Overlap}_m = \frac{\sum_{i=1}^{m} T_{i,m}}{m}. \tag{9}$$

We next derive $p(l|k)$, the probability that a node with degree k in graph G has degree l in graph G_e.

First, let us consider the case with *link deletion*. A node with degree k in graph G has degree l in graph G_e if and only if $(k-l)$ links are deleted from the node. The probability distribution of the number of deleted links follows the binomial distribution, and therefore, as also shown in [19], the probability that a node with degree k in graph G has degree l in graph G_e is given by

$$p_D(l|k) = \binom{k}{k-l} (1-\alpha)^l \alpha^{k-l}. \tag{10}$$

Next, let us consider the case with *node deletion*. In this case, similarly to the case with *link deletion*, the probability that s links are deleted from a node with degree k follows the binomial distribution. Hence, the probability that a node with degree k in graph G has degree l in graph G_e is given by

$$p_v(l|k) = \begin{cases} (1-\beta)\binom{k}{k-l}(1-\beta)^l \beta^{k-l} & l > 0 \\ \beta + (1-\beta)\beta^k & l = 0. \end{cases} \tag{11}$$

Next, let us consider the case with *link addition*. The probability that s links are added to a node with degree k is approximated by the Poisson distribution when the number of nodes $|V|$ is sufficiently large. Hence, as shown in [19], the probability that a node with degree k in graph G has degree l in graph G_e is approximated by

$$p_a(l|k) \simeq \frac{u^{l-k}}{(l-k)!} e^{-u}, \tag{12}$$

where u is the average number of added links per node, and is defined as $u = 2|E|\gamma/|V|$.

Next, let us consider the case with both *link deletion* and *link addition*. The probability that a node with degree k in graph G has degree l in graph G_e is derived in [19], and given by

$$p_{da}(l|k) = \sum_r \frac{u^r e^{-u}}{r!} \binom{k}{k+r-l} (1-\alpha)^{l-r} \alpha^{k+r-l}. \tag{13}$$

Finally, we consider the case with all of *link deletion*, *link addition*, and *node deletion*. A node with degree k in graph G has degree l in graph G_e if and only if r links are added, s adjacent nodes are deleted, and $(k+r-s-l)$ links are deleted from the node. Hence, combining Eqs. (11) and (13), the probability that a node with degree k in graph G has degree l in graph G_e is given by the following equation.

$$p(l|k) = \begin{cases} (1-\beta) \sum_r \frac{u^r e^{-u}}{r!} \sum_s \binom{k}{s} (1-\beta)^{k-s} \beta^s \\ \times \binom{k-s}{k+r-s-l} (1-\alpha)^{l-r} \alpha^{k+r-s-l} & l > 0 \\ \beta + \\ (1-\beta) e^{-u} \sum_s \binom{k}{s} (1-\beta)^{k-s} \beta^s \alpha^{k-s} & l = 0 \end{cases} \tag{14}$$

Note that we can also obtain a correlation coefficient r between degrees in graph G and those in graph G_e by using Eq. (14) and the model in [19].

4 Numerical Examples and Simulation Results

In this section, we examine the validity of our analysis by comparison between numerical examples of our analysis and the results of simulations. Moreover, we also discuss the effects of missing nodes, missing links, and false links on node rankings that are based on degree centrality.

As the ground-truth graph G, we use random graphs generated with the ER (Erdös–Rényi) model [10] and scale-free graphs generated with the BA (Barabási–Albert) model [2]. The number of nodes is 200, and the average degree of a node is 5 in the ER model and 2 in the BA model. In our simulations, we obtain graph G_e by deleting each link with probability α, deleting each node with probability β, and adding $\gamma|E|$ links between randomly selected pairs of unlinked nodes. For each graph G, we obtain 200 graphs for G_e, and calculate Top_m and $\mathrm{Overlap}_m$. We generate 100 different initial graphs G, and obtain averages of Top_m and $\mathrm{Overlap}_m$. We also obtain Top_m and $\mathrm{Overlap}_m$ by using our analytical models from degrees of nodes in graph G and the parameters α, β, and γ. In what follows, lines in the figures represent the results of analysis, and dots represent results of simulation.

We first investigate Top_m and $\mathrm{Overlap}_m$ when only a single type of error is contained in the graphs. Namely, we obtain Top_m and $\mathrm{Overlap}_m$ while two of α, β, and γ are fixed at 0 and the other parameter is changed. Figures 1, 2, and 3 show the results when changing α, β, and γ, respectively; Top_1, Top_3, and $\mathrm{Overlap}_3$ are used to characterize the results. From these results, we can confirm that the results of analysis are in good agreement with the simulation results. These results show the validity of our analysis.

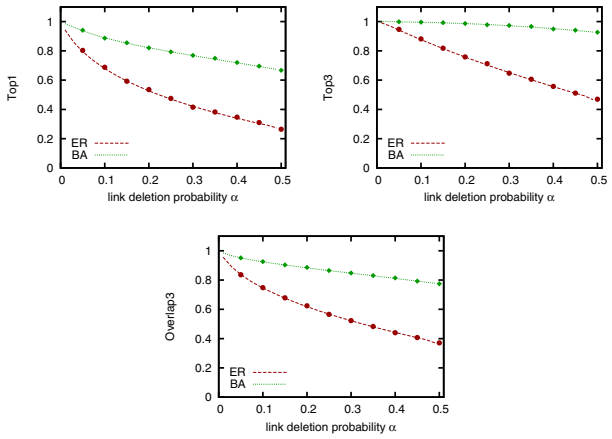


Fig. 1. Link deletion probability α vs. Top_1, Top_3, and $\mathrm{Overlap}_3$

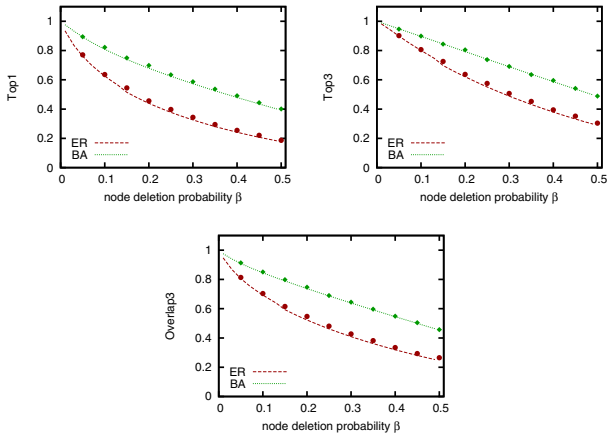


Fig. 2. Node deletion probability β vs. Top_1, Top_3, and $\mathrm{Overlap}_3$

We next investigate Top_m and $\mathrm{Overlap}_m$ when multiple types of errors are contained in graphs. We focus on two cases: a case with both missing links and false links, and a case with missing links and missing nodes. A typical example of the first case is the case of constructing protein interaction networks, where measurement error causes both missing links and false links. A typical example of the latter case is the case of constructing a social network, where incomplete data causes both missing links and missing nodes. Figure 4 shows Top_1, where errors of both missing links and false links are contained in graphs, and Fig. 5 shows Top_1, where errors of both missing links and missing nodes are contained in graphs. These figures show that analytical results and simulation results coincide closely. These results show the validity of our analysis when multiple types of errors are contained in graphs.

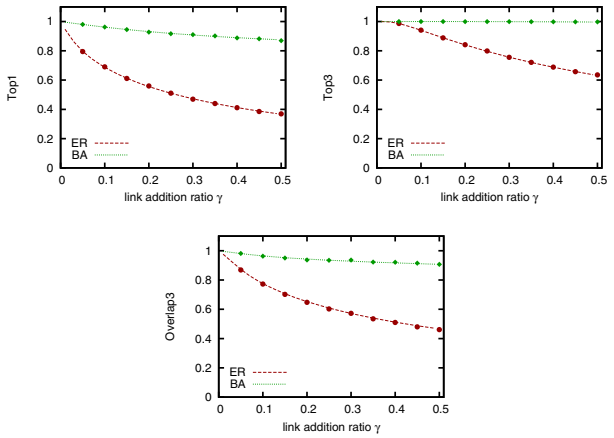


Fig. 3. Link addition ratio γ vs. Top_1, Top_3, and $\mathrm{Overlap}_3$

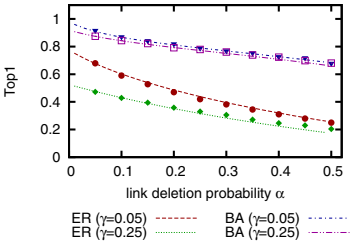


Fig. 4. Top_1 when errors of both missing links and false links are contained in graphs

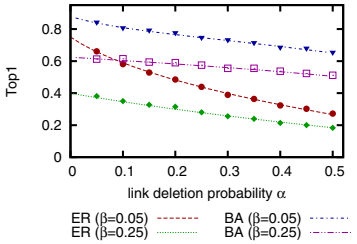
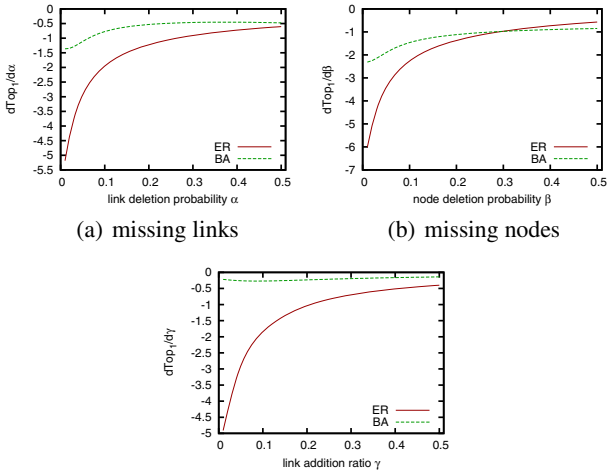


Fig. 5. Top_1 when errors of both missing links and missing nodes are contained in graphs

From these results, as previously shown in the simulation studies [7, 16, 19], we can observe non-negligible effects of random errors in graphs on node rankings based on degree centrality. Graphs generated according to the BA model are more robust than those generated according to the ER model, but in the particular case with missing nodes, Top_1, Top_3, and $\mathrm{Overlap}_3$ decrease almost linearly. Thus, our analysis gives theoretical confirmation of the results from previous works.

We further analyze the robustness of degree centrality by using analytical models. We differentiate Top_1 with respect to the error rates α, β, and γ, and investigate the effects of each type of error on the node ranking. Figure 6 shows the derivation of Top_1 with respect to the error rates α, β, and γ. These figures show the relation between an increase in the error rate and a decrease in the accuracy of detecting most central node according to degree centrality. From these figures, we can find that, for instance,



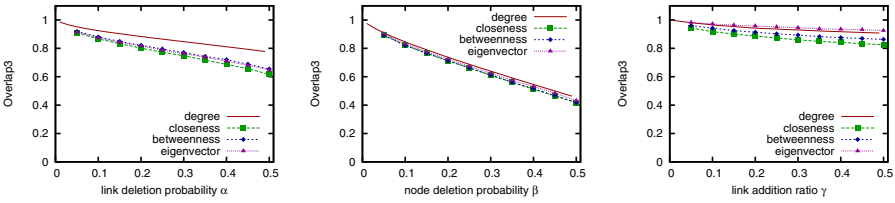
(a) missing links (b) missing nodes

(c) false links

Fig. 6. Derivation of Top_1 with respect to α, β, and γ: in the panels, one parameter is changed and the other two parameters are fixed to 0

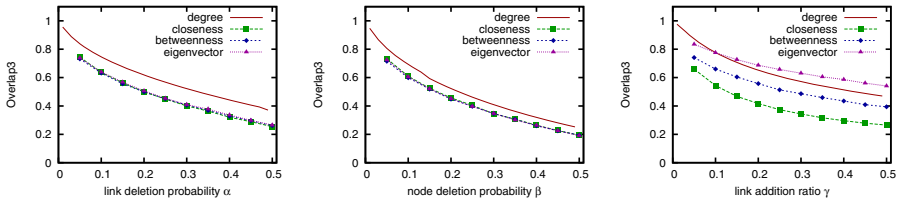
in graphs generated from the BA model, a 1% increase in the rate of missing nodes causes an approximately 2% decrease in Top_1 when the node deletion probability is less than 0.1. Our models reveal the relation between the increase of the error ratio and the decrease in accuracy of centrality.

Finally, we investigate the robustness of other types of centrality measures (specifically, betweenness, closeness, and eigenvector centralities) through simulations. Due to space limitation, we show the results of $\mathrm{Overlap}_3$ only. Figures 7 and 8 show $\mathrm{Overlap}_3$ of different types of centrality measures when α, β, and γ are changed in the BA model (Fig. 7) and the ER model (Fig. 8), respectively. For comparison purposes, the analytical results of $\mathrm{Overlap}_3$ of degree centrality are also shown on the graphs.



(a) missing links (b) missing nodes (c) false links

Fig. 7. $\mathrm{Overlap}_3$ of the four types of centrality measures (degree, closeness, betweenness, and eigenvector centralities) in graphs generated according to the BA model: $\mathrm{Overlap}_3$ of degree centrality is obtained by our analysis, and values with the other measures are obtained by simulation



(a) missing links (b) missing nodes (c) false links

Fig. 8. Overlap_3 of the four types of centrality measures (degree, closeness, betweenness, and eigenvector centralities) in graphs generated according to the ER model: Overlap_3 of degree centrality is obtained by our analysis, and values with the other measures are obtained by simulation

Figure 7 shows that Overlap_3 of the four types of centrality is similar among graphs generated with the BA model. Figure 8 shows that in the ER model, the magnitudes of Overlap_3 are different, but the curves of Overlap_3 are of similar shape for the four types of centrality measures. We observed (not shown here) that Top_1 and Top_3 also exhibit similar tendencies. These results indicate that the four types of centrality measures have similar robustness, particularly in graphs generated according to the BA model. This suggests that analytical models of degree centrality can be used to predict the robustness of other types of centrality measures. The cause of the similar robustness among the four types of centrality measures can be attributed to the high correlation among the centrality measures.

5 Conclusion and Future Works

We analyzed the robustness of degree centrality against missing nodes, missing links, and false links. We extended the model of [19], and derived Top_m and Overlap_m, which were used to evaluate the robustness of node rankings, and showed the validity of the analysis. Moreover, through extensive simulations, we showed that the four types of popular centrality measures (degree, closeness, betweenness, and eigenvector centralities) exhibit similar robustness, which suggests that our model can be used to analyze the robustness of not only degree centrality but also other types of centrality measures.

As future work, we plan to analyze the robustness of centrality measures other than degree centrality. Investigating the effects of other types of errors is also important future work. This paper focuses on uniform errors, but in actual network analyses, non-uniform errors arise. As an example, biased sampling is a known cause of non-uniform errors, and such types of errors are of interest to network researchers.

Acknowledgments. This work was partly supported by JSPS KAKENHI Grant Number 25280030 and 26870076.

References

1. Ball, B., Newman, M.E.J.: Friendship networks and social status. Network Science 1(01), 16–30 (2013)
2. Barabási, A., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
3. Batallas, D., Yassine, A.: Information leaders in product development organizational networks: Social network analysis of the design structure matrix. IEEE Transactions on Engineering Management 53(4), 570–582 (2006)
4. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. Journal of Mathematical Sociology 2(1), 113–120 (1972)
5. Borgatti, S.: Identifying sets of key players in a social network. Computational & Mathematical Organization Theory 12(1), 21–34 (2006)
6. Borgatti, S., Mehra, A., Brass, D., Labianca, G.: Network analysis in the social sciences. Science 323(5916), 892–895 (2009)
7. Borgatti, S.P., Carley, K.M., Krackhardt, D.: On the robustness of centrality measures under conditions of imperfect data. Social Networks 28(2), 124–136 (2006)
8. Chen, J., Yuan, B.: Detecting functional modules in the yeast protein–protein interaction network. Bioinformatics 22(18), 2283–2290 (2006)
9. Costenbader, E., Valente, T.: The stability of centrality measures when networks are sampled. Social Networks 25(4), 283–307 (2003)
10. Erdös, P., Rényi, A.: On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci. 5, 17–61 (1960)
11. Freeman, L.: Centrality in social networks conceptual clarification. Social Networks 1(3), 215–239 (1979)
12. Ghoshal, G., Barabási, A.: Ranking stability and super-stable nodes in complex networks. Nature Communications 2(394), 1–7 (2011)
13. Ishino, M., Tsugawa, S., Ohsaki, H.: On the robustness of centrality measures against link weight quantization in real weighted social networks. In: Proceedings of the the 2nd International Workshop on Ambient Information Technologies (AMBIT 2013), pp. 25–28 (March 2013)
14. Kim, P., Jeong, H.: Reliability of rank order in sampled networks. The European Physical Journal B-Condensed Matter and Complex Systems 55(1), 109–114 (2007)
15. Lee, S.H., Kim, P.J., Jeong, H.: Statistical properties of sampled networks. Physical Review E 73(1), 016102 (2006)
16. Frantz, T.L., Cataldo, M., Carley, K.: Robustness of centrality measures under uncertainty: Examining the role of network topology. Computational and Mathematical Organization Theory 15(4), 303–328 (2009)
17. Matsumoto, Y., Tsugawa, S., Ohsaki, H., Imase, M.: Robustness of centrality measures against link weight quantization in social network analysis. In: Proceedings of the 4th Annual Workshop on Simplifying Complex Networks for Practitioners (SIMPLEX 2012), pp. 49–54 (April 2012)
18. Newman, M.E.J., Strogatz, S.H., Watts, D.J.: Random graphs with arbitrary degree distributions and their applications. Physical Review E 64(2), 026118 (2001)
19. Platig, J., Ott, E., Girvan, M.: Robustness of network measures to link errors. Physical Review E 88(6), 062812 (2013)
20. Rubinov, M., Sporns, O.: Complex network measures of brain connectivity: uses and interpretations. Neuroimage 52(3), 1059–1069 (2010)

21. Tsugawa, S., Ohsaki, H.: Emergence of fractals in social networks: Analysis of community structure and interaction locality. In: Proceedings of the 38th Annual IEEE International Computers, Software, and Applications Conference (COMPSAC 2014), pp. 568–575 (July 2014)
22. Watts, D.J.: A twenty-first century science. Nature 445(7127), 489 (2007)
23. Yu, H., Braun, P., Yıldırım, M.A., Lemmens, I., Venkatesan, K., Sahalie, J., Hirozane-Kishikawa, T., Gebreab, F., Li, N., Simonis, N., et al.: High-quality binary protein interaction map of the yeast interactome network. Science 322(5898), 104–110 (2008)
24. Žnidaršič, A., Ferligoj, A., Doreian, P.: Non-response in social networks: The impact of different non-response treatments on the stability of blockmodels. Social Networks 34(4), 438–450 (2012)
25. Zuo, X.N., Ehmke, R., Mennes, M., Imperati, D., Castellanos, F.X., Sporns, O., Milham, M.P.: Network centrality in the human functional connectome. Cerebral Cortex 22(8), 1862–1875 (2012)

A Model for Ambiguation and an Algorithm for Disambiguation in Social Networks*

Janaína Gomide, Hugo Kling, and Daniel Figueiredo

Systems Engineering and Computer Science Program (PESC)
Federal University of Rio de Janeiro (UFRJ), Brazil
{janaina,hugo,daniel}@land.ufrj.br

Abstract. A common assumption when collecting network data is that objects can be uniquely identified. However, in many scenarios objects do not have a unique label giving rise to ambiguities since the mapping between observed labels and objects is not known. In this paper we consider the ambiguity problem that emerges when objects appear with more than one label in the context of social networks. We first propose a probabilistic model to introduce ambiguity in a network by duplicating vertices and adding and removing edges. Second, we propose an simple label-free algorithm to remove ambiguities by identifying duplicate vertices based only in structural features. We evaluate the performance of the algorithm under two classical random network models. Results indicate that network structure can indeed be used to identify ambiguities, yielding very high precision when local structure is preserved.

Keywords: network ambiguity, social networks, network structure, disambiguation.

1 Introduction

During the past decade, networks have increasingly been used to encode relationships between objects, from interactions among proteins, to friendship among people, to hyperlinks between webpages. Underlying this abstraction is the premise that objects can be uniquely identified when observing relationships among them. For example, user accounts in Facebook have a unique number identifier that is used when crawling the friendship graph.

However, in many scenarios objects do not exhibit a unique identifier when relationships among them are observed. In particular, a single object may have different labels that appear in reference to the object, or alternatively, a single label may appear in reference to different objects. For example, in the context of social networks, a person (object) may be known by various names (labels), or a single name (label) may be given to different people (objects). Thus, when observing relationships among labels of objects we are faced with ambiguity, since the mapping between observed labels and objects may not be known a priori. In a nutshell, network disambiguation refers to the problem of removing ambiguities

* This research received financial support through grants from FAPERJ and CNPq (Brazil).

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_4

among nodes of network that is constructed by observing relationships among ambiguous labels. A more precise formulation is given in Section 3.

In this work we are interested in understanding ambiguity arising when a single object can appear with different labels in the context of social networks. We call this the "Brazilian Ambiguity Problem" (BAP) in allusion to the fact that Brazilians tend to have many first and last names which then appear in many different forms and combinations. Towards this direction, we make the following contributions:

1. Ambiguity model for BAP: based on intuition and empirical observations of real data, we propose a probabilistic model that introduces ambiguity in a social network. The model has three intuitive parameters used for tuning the desired amount and structure of ambiguity and can operate over any original social network. This model is presented in Section 4.
2. Disambiguation algorithm for BAP: again, based on intuition and empirical observations of real data, we propose a simple and efficient label-free algorithm for removing ambiguity in the context of BAP. Our algorithm uses only the structure of the network of observed labels but not the labels themselves to identify nodes (labels) that refer to the same person. We present an extensive analysis of the performance (precision and recall) of algorithm when applying the proposed ambiguity model to random graph models. The algorithm and its evaluation are presented in Sections 5 and 6, respectively.

Identifying ambiguities among nodes of a network of observed labels is an important problem, as one is usually interested in the network of objects. In particular, the network of objects and not labels is the one that is used to characterize and make statements about relationships or other phenomena that depends on the structure. Nevertheless, the problem of name disambiguation has been studied for more two decades, as discussed in Section 2. Our contributions as enumerated above indicates that structure alone in the network of observed labels can contribute to addressing the BAP.

2 Related Work

The problem of network disambiguation is considered a difficult and relatively open problem [2,4]. Author name disambiguation was initially studied in the Information Sciences using manual and intuitive methods [2], but also in Computer Science using sophisticated algorithms [3,4].

Most approaches found in literature consider label and textual information as main features to remove ambiguities in the network, which might not be available in several contexts. We believe that structural features are fundamental to solve ambiguity in networks in agreement with other recent works [1,5,7].

The problem of more than two people being represented in one node (appear with the same name) has been addressed using a supervised classification algorithm (SVM) considering as features the structural information of the network [5], and also using an unsupervised learning algorithm [7]. The BAP (one person

appearing with multiple names) has also been addressed using a machine learning approach with structure and textual features [1]. Our work contribution is an ambiguity model for the BAP and a disambiguation algorithm that does not use machine learning.

3 Problem Statement

In this section we formalize the network ambiguity problem. Consider a graph $G = (O, E)$ where the vertex set $O = \{o_1, \ldots, o_n\}$ represents objects and the edge set E represents pairwise relationships among the objects. Lets assume that objects have labels and in particular, let $L_i = \{l_{i,1}, \ldots, l_{i,s_i}\}$ denote the set of labels that can be assigned to object o_i. Note that objects have one or more label that are not necessarily unique. Thus, labels of different objects can be identical.

Consider an observation process of relationships among objects that reveals object labels. Thus, a relationship $(o_i, o_j) \in E$ is observed as (l_i, l_j) where $l_i \in L_i$ and $l_j \in L_j$. Let $L = \bigcup_{i=1}^{n} L_i$ denote the set of all different labels. The observation process applied to many (possibly all) relationships $(o_i, o_j) \in E$ will then yield a graph $G' = (L', E')$ where the vertex set $L' \subset L$ represents all observed labels and the edge set E' represents all observed relationships among labels. Note that a given $l \in L'$ can refer to two or more objects while a given $l_1, l_2 \in L'$ can refer to the same object.

The network disambiguation problem is to recover G (network of objects) having observed G' (network of labels). In the context of the "Brazilian Ambiguity Problem" (BAP) studied in this paper, labels of different objects are different, thus, $l_i \neq l_j$ for any $l_i \in L_i$ and $l_j \in L_j$ and for any $i \neq j$. However, we also assume there is no information on the labels themselves (i.e., labels are random numbers), and no information on the number of labels assigned to each object.

4 Ambiguation Model

In this section we present a novel probabilistic model that introduces ambiguity in a network. The model is mostly tailored for social networks and its workings are based on intuition and empirical observations. The idea is to duplicate nodes and add and remove edges to neighbours of the original node. A duplicated node represents a second label for the original node. Therefore, one object (node) of the original network can be represented by two nodes (labels) in the ambiguous network and relationships among the original object (node) can be copied to its duplicate and removed from itself.

Consider a network represented as a graph $G = (V, E)$ in which V is the vertices set (e.g. people), and E is the set of edges (e.g. friendship relationship). In this graph, each vertex uniquely identifies an object in the network. The proposed model has three phases, each with a parameter:

1. **Vertex duplication:** with probability p a vertex is duplicated;
2. **Edge addition:** with probability q an edge between a neighbour of the original vertex and the duplicated vertex is created;
3. **Edge removal:** with probability r an original edge that was copied to a duplicated vertex is removed.

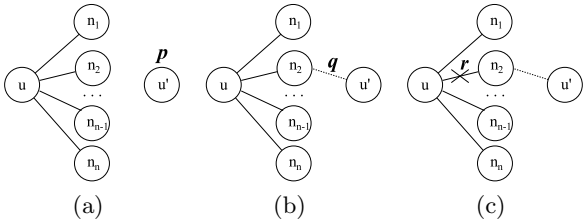


Fig. 1. Parameters of the probabilistic model for create ambiguity in a network. In (a) vertex duplication phase, (b) edge addition phase, and (c) edge removal phase.

In the vertex duplication phase, the vertices are duplicated creating ambiguity. Each vertex $u \in V$, sampled with probability p independently, to generate another graph with a duplicate vertex, u', as shown in Figure 1(a). Note that p controls the amount of ambiguity introduced in the network, so that with $p = 1$ all vertices will have a duplicate in a network.

In the edge duplication phase, the neighbours from the original vertex are copied to the duplicated vertex. For each neighbour $v \in N_u$(neighbours of u) of an original vertex u that has been duplicated, with probability q independently, an edge $e = (u', v)$ is created as illustrated in Figure 1(b). Note that with $q = 1$ all neighbours from u will become neighbours of u'.

In the edge removal phase, edges between an original vertex u and a neighbour v, that has become a neighbour of u' is removed with probability r, independently, as shown in Figure 1(c). Note that for $r = 1$ all edges between the original vertex u and its neighbours that became neighbours of the duplicate vertex u' will be removed. The algorithm for this ambiguity model is described in Algorithm 1.

5 Algorithm for Removing Ambiguities

In this section we present a simple algorithm to identify ambiguities in the context of BAP in a social network. In particular, we consider just the case where a single object, due to ambiguities, can be represented in the observed label network by more than one vertex. Our algorithm will identify network nodes that represent the same entity without resorting to label information - thus, only structure information will be used.

We develop several structure-based heuristics to identify nodes in the label network that might represent the same entity. For example, we consider that two

nodes might refer to the same entity if they are at distance 2, since it is unlikely that a node will have a relationship with itself using two different labels. Moreover, the same is considered if the common neighbourhood between two vertices strongly overlaps, and is contained in one another. We aim in developing a conservative approach to merge nodes, in order to minimize false-positives, allowing greater applicability of the algorithm. The proposed algorithm is described in Algorithm 2.

Algorithm 1: Model to introduce ambiguity with parameters: p, q, r.

Data: $G = (V, E)$, p, q, r
Result: $G' = (V', E')$
$E' \longleftarrow E$; $V_d \longleftarrow \emptyset$; $E_d \longleftarrow \emptyset$
for *v in V* **do**
  with probability p, duplicate v into v' and $V_d \longleftarrow V_d \cup v'$
for *v' in* V_d **do**
  $v \longleftarrow$ original(v')
  $N \longleftarrow$ neighbours(v)
  for *u in N* **do**
    with probability q, create $e' = (v', u)$ and $E_d \longleftarrow E_d \cup e'$
    if e' *in* E_d **then**
      with probability r, remove $e = (v, u)$ from E'
$V' \longleftarrow V \cup V_d$; $E' \longleftarrow E' \cup E_d$

Algorithm 2: Algorithm - Remove ambiguity

Data: $G = (V, E)$, α
for *v in V* **do**
  $P \longleftarrow \emptyset$; $N_v \longleftarrow$ neighbours(v) ; $D_v^2 \longleftarrow \{u | distance(u, v) = 2\}$
  for *u in* D_v^2 **do**
    if $degree(v) \geq \alpha$ *and* $degree(v) \leq$ *degree(u)* **then**
      $N_u \longleftarrow$ neighbours(u)
      if $N_v \subseteq N_u$ **then**
        $P \longleftarrow P \cup u$
  if *sizeOf(P) = 1* **then** `/* Ambiguity found! Unify` v `and P.first() */`
    merge(v,P.first())

6 Evaluation

In this section we present an extensive evaluation of the performance of the proposed algorithm to remove ambiguities when applied to networks generated by the ambiguity model.

The steps evaluation has the following steps: (i) generate the networks, (ii) introduce ambiguity using the model proposed in Section 4, (iii) apply the

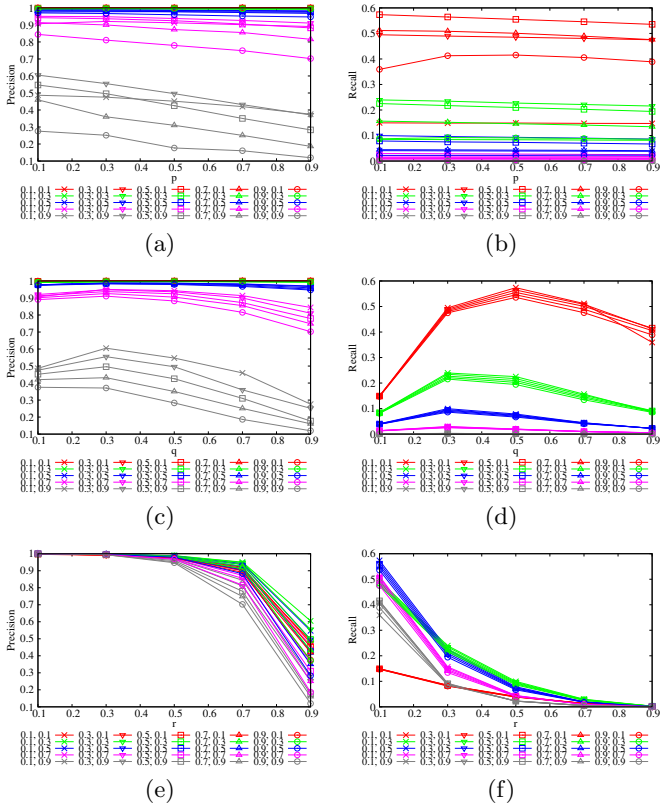


Fig. 2. Evaluation in Erdos-Renyi network with ambiguity. In (a,c,e) precision and in (b,d,f) recall. The pair of values in the legend correspond to p, q, r with the exception of the value appearing in x-axis.

algorithm to remove ambiguity proposed in Section 5 and (iv) measure the precision and recall of the algorithm.

In order to generate the networks, we use two models, Erdos-Renyi model, that generates graphs connecting nodes randomly, and Watt-Strograts model, that generates graphs with small-world properties [6]. Both networks were generated with $n = 100,000$ vertices and average degree of eight (rewiring probability of two percent was used in the Watts-Strogats model).

Next, we introduce ambiguity into the two networks created. We apply the probabilistic model with different values for the parameters p, q and r aiming to evaluate how these parameters affect the identification of duplicated vertices. The values used for each parameter are 0.1, 0.3, 0.5, 0.7 and 0.9. We apply the algorithm to remove the duplicated vertices, with parameter $\alpha = 0$, and we evaluate the performance by measuring the precision and recall of the algorithm. For each parameter configuration, we perform thirty independent runs and report the sample average of performance metrics. The algorithm performance in

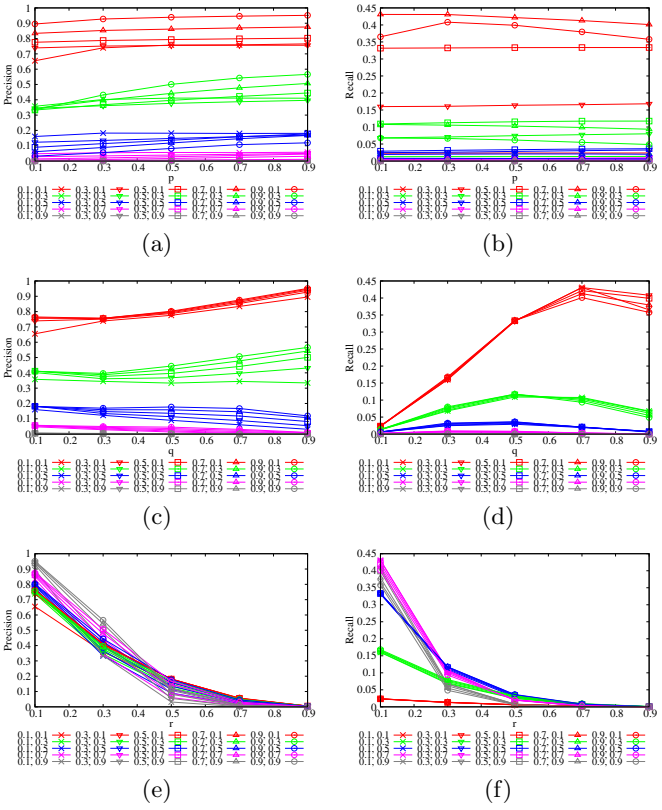


Fig. 3. Evaluation in Watts-Strogatz network with ambiguity. In (a,c,e) precision and in (b,d,f) recall. The pair of values in the legend correspond to p, q, r with the exception of the value appearing in x-axis.

the Erdos-Renyi and in the Watts-Strogatz network models with ambiguity are shown in Figures 2 and 3 respectively for all combinations of model parameters.

The precision and recall for the Erdos-Renyi model are shown in Figures 2(a) and 2(b), respectively. Note that the parameter p is not critical to the algorithm, when ten or ninety percent of the vertices are duplicated the performance of the algorithm remains roughly the same. This occurs because in the Erdos-Renyi network model lacks local structure and, therefore, any duplication of vertices and edges creates a local structure that is detected by the algorithm. In these Figures the lines are grouped by the parameter r, so that with smaller values of r we get around 100% of precision and 50% of recall.

In Figures 2(c) and 2(d) we observe an inflexion point with respect to parameter q, with precision and recall growing and then o decrease. This occurs because the number of edges that are removed from the original grows with q. However, for lower values of q the duplicated vertex has a small degree and thus there are many vertices that are candidates to be its original and the algorithm

fails to make a decision yielding a lower precision and recall. The inflexion point changes with the value of r because the expected number of removed edges is $d_u qr$ where d_u is the degree of the node u.

Figures 2(e) and 2(f) shows the precision and the recall as a function of parameter r, respectively. Clearly, r is the most sensitive parameter for the performance of the algorithm. Note that precision is more than 90% for values of r lower than 0.5, independent of the other parameters p and q. As r grows the precision and the recall decrease as more original edges are removed and the algorithm fails to find the original vertex that corresponds to the duplicated one.

Results under the Watts-Strogatz network model is shown in Figure 3. In general, results have the same qualitative trends as for the Erdos-Renyi model, with a higher sensitivity in the parameter r. For example Figures 3(e) and 3(f) illustrate that performance degrades quickly as r increses. This occurs due to the local structure present in the Watts-Strogatz model, which makes the algorithm fail if few edges are removed.

7 Conclusion

In this work we addressed the problem of disambiguation in networks when different labels (vertices) can represent the same object. We proposed a probabilistic model that introduces ambiguity in the context of social networks using three parameters for tuning the desired amount of structural ambiguity. We also propose a simple disambiguation algorithm that uses only structure to identify duplicate nodes. Through simultaneous, we extensively evaluate the performance of the algorithm using random graphs subject to ambiguation introduced by the proposed ambiguity model. Results indicate that the structure of a network can successfully be used to identify ambiguities and does not strongly depend on the amount (fraction) of objects with double identity (duplicated nodes), but on the local structure between the main and the alternative labels. In particular, local network features such as absence of direct edge and common neighbourhood play a key role in disambiguation of social networks.

References

1. Amancio, D., Oliveira Jr., O., Costa, L.: On the use of topological features and hierarchical charac. for disambiguating names in collab. networks. In: EPL (2012)
2. Elliot, S.: Survey of author name disambiguation: 2004 to 2010. Library Philosophy and Practice 473 (2010)
3. Fan, X., Wang, J., Pu, X., Zhou, L., Lv, B.: On graph-based name disambiguation. J. Data and Information Quality 2(2), 10:1–10:23 (2011)
4. Ferreira, A.A., Gonçalves, M.A., Laender, A.H.: A brief survey of automatic methods for author name disambiguation. SIGMOD Rec. 41(2), 15–26 (2012)
5. Hermansson, L., Kerola, T., Johansson, F., Jethava, V., Dubhashi, D.: Entity disambiguation in anonymized graphs using graph kernels. In: CIKM (2013)
6. Newman, M.: Networks: An Introduction. Oxford University Press (2010)
7. Zhang, B., Saha, T.K., Hasan, M.A.: Name disambiguation from link data in a collaboration graph. In: ASONAM (2014)

Measuring the Generalized Friendship Paradox in Networks with Quality-Dependent Connectivity

Naghmeh Momeni and Michael G. Rabbat

Department of Electrical and Computer Engineering
McGill University, Montréal, Canada
naghmeh.momenitaramsari@mail.mcgill.ca, michael.rabbat@mcgill.ca

Abstract. The friendship paradox is a sociological phenomenon stating that most people have fewer friends than their friends do. The generalized friendship paradox refers to the same observation for attributes other than degree, and it has been observed in Twitter and scientific collaboration networks. This paper takes an analytical approach to model this phenomenon. We consider a preferential attachment-like network growth mechanism governed by both node degrees and 'qualities'. We introduce measures to quantify paradoxes, and contrast the results obtained in our model to those obtained for an uncorrelated network, where the degrees and qualities of adjacent nodes are uncorrelated. We shed light on the effect of the distribution of node qualities on the friendship paradox. We consider both the mean and the median to measure paradoxes, and compare the results obtained by using these two statistics.

1 Introduction

The friendship paradox, introduced by Feld [1], is a sociological observation that says most people are less popular than their friends on average. It is called a 'paradox' because, while most people believe that they are more popular than their friends [2], Feld observed that the converse is actually true. There are more recent observations agreeing with Felds', that study online environments. For example on Twitter, people you follow and also your followers have, on average, more followers than you do. They also follow more people than you do [3]. On Facebook, your friends have, on average, more friends than you do [4].

The friendship paradox is about the inter-nodal inequality of the degrees. What happens if we consider other attributes? This is the focus of the 'Generalized Friendship Paradox' [5,6]. For example on Twitter, your friends on average tweet more and also share more viral content than you [3,7]. In the scientific collaboration networks your collaborators have on average more publications, more citations and more collaborators than you do [5].

The friendship paradox has applications in spotting influential nodes. In [8], it is used for finding high-degree nodes for efficient vaccination. In order to sample a node with above average degree, a node is chosen uniformly at random and

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_5

one of their neighbours will be sampled. In [9], the friendship paradox is used for the early detection of flu outbreaks among college students. In [10], it is utilized to derive early-warning sensors during catastrophic events such as hurricanes.

In this paper, first we explain a quality-dependent preferential attachment scheme introduced in [12]. Then, we introduce measures to quantify the mean and the median paradoxes. In Section 4 these measures are computed numerically on the networks generated with the quality-dependent model and also uncorrelated networks. We compare the results obtained in these networks using both the mean and the median statistics. Furthermore, we study the effect of node quality distribution on the quality and friendship paradoxes.

2 Model, Notation and Terminology

We consider a *quality-based preferential attachment* (QPA) model, identical to the model proposed and analysed in [12]. It is similar to the Barabasi-Albert model [11], but incorporates node qualities. Each incoming node has β links, and a discrete quality θ drawn from a distribution $\rho(\theta)$ that is assigned to it upon birth. The probability of an existing node x with degree k_x and quality θ_x (at the instant) receiving a new link is proportional to $k_x + \theta_x$.

Once assigned, the quality of a node does not change. We denote the mean of the quality distribution by μ. Following [12], as the number of nodes tends to infinity, $P(k,\theta)$, the fraction of nodes with degree k and quality θ is given by:

$$P(k,\theta) = \rho(\theta)\big(2+\frac{\mu}{\beta}\big)\frac{\Gamma(k+\theta)}{\Gamma(\beta+\theta)}\frac{\Gamma\big(\beta+\theta+2+\frac{\mu}{\beta}\big)}{\Gamma\big(k+\theta+3+\frac{\mu}{\beta}\big)}u(k-\beta). \tag{1}$$

In [12] the nearest-neighbor distribution, i.e., the fraction of neighbors of a node with degree k and quality θ who has degree ℓ and quality ϕ is given by:

$$P(\ell,\phi|k,\theta) = \frac{\rho(\phi)}{k}\frac{\Gamma\left(k+\theta+3+\frac{\mu}{\beta}\right)}{\Gamma\left(k+\theta+3+\frac{\mu}{\beta}+\ell+\phi\right)}\frac{(\ell-1+\phi)!}{(\beta-1+\phi)!}\Gamma\left(\beta+2+\phi+\frac{\mu}{\beta}\right)\times$$
$$\left[\sum_{j=\beta+1}^{k}\frac{\Gamma\left(j+\theta+2+\frac{\mu}{\beta}+\beta+\phi\right)\binom{k-j+\ell-\beta}{\ell-\beta}}{\Gamma\left(j+\theta+2+\frac{\mu}{\beta}\right)\Gamma\left(\beta+2+\phi+\frac{\mu}{\beta}\right)}+\sum_{j=\beta+1}^{\ell}\frac{\Gamma\left(j+\theta+2+\frac{\mu}{\beta}+\beta+\phi\right)\binom{\ell-j+k-\beta}{k-\beta}}{\Gamma\left(j+\phi+2+\frac{\mu}{\beta}\right)\Gamma\left(\beta+2+\theta+\frac{\mu}{\beta}\right)}\right]. \tag{2}$$

3 Measures of Friendship and Quality Paradoxes

By marginalizing the joint distribution $P(k,\theta)$ we can find the degree distribution, denoted by $P(k)$. Also, from the nearest-neighbor distribution (2), we can find the expected value of the qualities of neighbors of a node with quality θ and also the expected value of the degrees of neighbors of a node with degree k.

This allows us to investigate when the quality paradox (hereinafter QP) and the friendship paradox (hereinafter FP) are in force, and which nodes in the network exhibit the paradox.

Let us also define the 'median' version of the paradoxes, following [7]. In the median version, instead of the average values of quality or degree of neighbors, we use the median values. A node experiences the median QP (FP), if its quality (degree) is less than the quality (degree) of at least half of its neighbors.

Throughout the paper, the superscript NN denotes Nearest-Neighbor. Let us denote the median operator by $M\{\cdot\}$. For example, $M\{\phi^{NN}|\theta\}$ denotes the median value of ϕ under the distribution $P(\phi|\theta)$, and is a function of θ. Also note that every measure we introduce here is by nature a function of the parameters of the quality distribution. For example, if the exponential decay quality distribution is considered, the measures will depend on the decay factor. We denote the parameter of the quality distribution by x. Using this notation, we define the critical values for the mean and the median paradoxes as follows:

$$\text{mean:} \begin{cases} \widetilde{\theta}_c(x) \stackrel{\text{def}}{=} \max\left\{\theta \middle| \theta < E\{\phi^{\text{NN}}|\theta\}\right\} \\ \widetilde{k}_c(x) \stackrel{\text{def}}{=} \max\left\{k \middle| k < E\{\ell^{\text{NN}}|k\}\right\} \end{cases} , \text{median:} \begin{cases} \hat{\theta}_c(x) \stackrel{\text{def}}{=} \max\left\{\theta \middle| \theta < M\{\phi^{\text{NN}}|\theta\}\right\} \\ \hat{k}_c(x) \stackrel{\text{def}}{=} \max\left\{k \middle| k < M\{\ell^{\text{NN}}|k\}\right\} \end{cases} . \quad (3)$$

In other words, $\widetilde{\theta}_c(x)$ is the highest quality that a node can have, given that its quality is lower than the average quality of its neighbors. Similarly, $\widetilde{k}_c(x)$ is the highest degree that a node can have, given that it exhibits the mean FP. For the median version of the paradox, we have $\hat{\theta}_c(x)$ and $\hat{k}_c(x)$. So $\hat{\theta}_c(x)$ is the highest quality that a node exhibiting the median QP can have. Let us also emphasize that we use the following convention with regards to the median throughout the paper: the median of the probability distribution $g(x)$ (with CDF $G(x)$) is the minimum value of x for which $G(x) \geq \frac{1}{2}$. For example, for $g(x) = \frac{1}{2}\delta[x] + \frac{1}{2}\delta[x-5]$, the median is $x = 0$.

We now define similar quantities for an 'uncorrelated network'. In this network the qualities are assigned to nodes in an identical way to the QPA model, but the attachment of new nodes to existing nodes depends on neither the degrees nor the qualities of the existing nodes. In this network the properties of a node are uncorrelated with the properties of its neighbors. We denote this case by superscript u. For this network we have $P^u(\ell, \phi|k, \theta) = P(\ell, \phi)$ and $P^u(\phi|\theta) = \rho(\phi)$. For the critical values of the mean and the median paradoxes, we have:

$$\begin{cases} \widetilde{\theta}^u_c(x) \stackrel{\text{def}}{=} \max\left\{\theta \middle| \theta < E\{\phi^{\text{NN}}|\theta\}\right\} = \max\left\{\theta \middle| \theta < \underbrace{E\{\phi\}}_{=\mu}\right\} = \mu(x) - 1 \\ \hat{\theta}^u_c(x) \stackrel{\text{def}}{=} \max\left\{\theta \middle| \theta < M\{\phi^{\text{NN}}|\theta\}\right\} = \max\left\{\theta \middle| \theta < \underbrace{M\{\phi\}}_{=\hat{\theta}}\right\} = \hat{\theta}(x) - 1 \end{cases} . \quad (4)$$

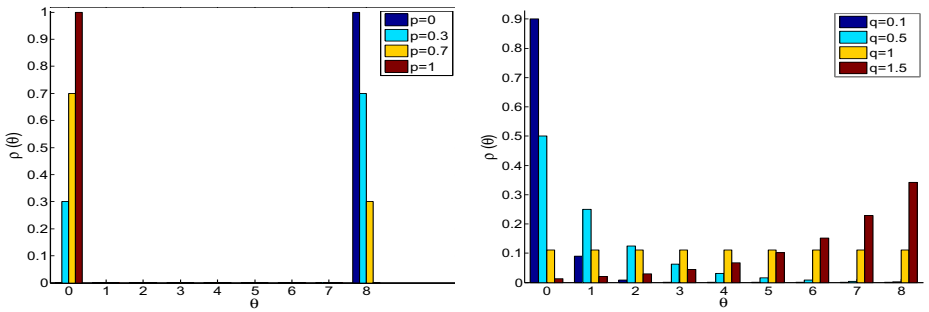
Similarly, for degrees we have: $\widetilde{k}^u_c(x) = \overline{k}(x) - 1$ and $\hat{k}^u_c(x) = \hat{k}(x) - 1$.

We are also interested in the fraction of all nodes that experience each type of paradoxes. This is equal to the fraction of nodes with their attribute below the corresponding critical value. We denote these quantities by:

$$\text{mean:} \begin{cases} \widetilde{F}_\theta(x) = \sum\limits_{\theta \le \tilde{\theta}_c(x)} \rho(\theta) \\ \widetilde{F}_k(x) = \sum\limits_{k \le \tilde{k}_c(x)} P(k) \end{cases}, \qquad \text{median:} \begin{cases} \hat{F}_\theta(x) = \sum\limits_{\theta \le \hat{\theta}_c(x)} \rho(\theta) \\ \hat{F}_k(x) = \sum\limits_{k \le \hat{k}_c(x)} P(k) \end{cases}. \quad (5)$$

4 Results and Discussion

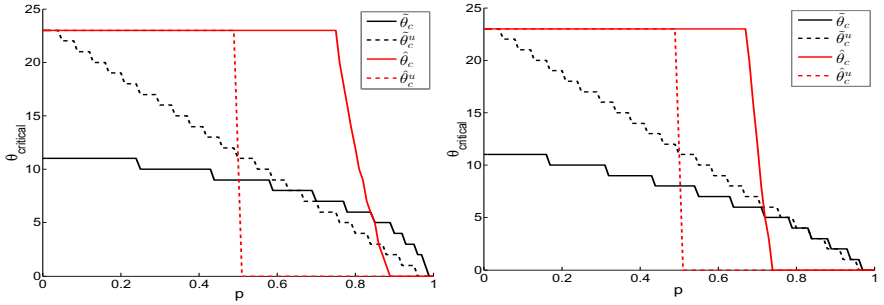
In this paper we consider two quality distributions for expository purposes. The first one is the Bernoulli distribution, where nodes have quality 0 (with probability p) or quality $\theta_{\max}$ (with probability $1-p$). The other one is the discrete exponential distribution, with decay factor q. The probability of quality θ is proportional to q^θ, and the maximum value of θ is denoted by $\theta_{\max}$. Figure 1 depicts these quality distributions for four example values of p and q. Note that for $q < 1$, the exponential distribution is a decreasing function of quality and $\mu > \hat{\theta}$, and for $q > 1$, the distribution is increasing function of quality and $\mu < \hat{\theta}$. Also for the Bernoulli distribution note that, with the convention we use for the median, the value of the median is zero if $p \ge \frac{1}{2}$, and the median is equal to $\theta_{\max}$ if $p < \frac{1}{2}$. For each distribution, we have numerically computed all the introduced measures for four different values of β and four different values of $\theta_{\max}$.



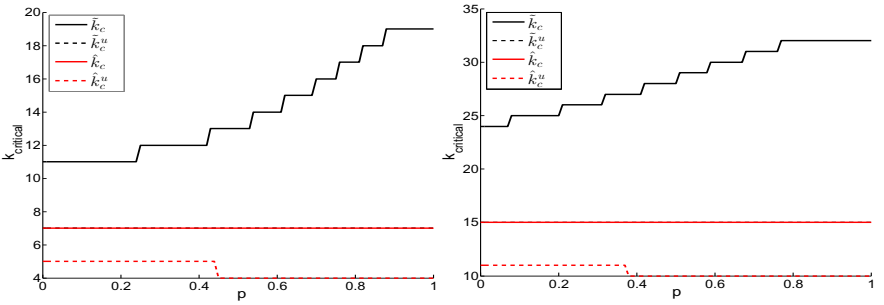
(a) Bernoulli distribution with $p = 0, 0.3, 0.7, 0.1$. The cases of $p = 0$ and $p = 1$ correspond to conventional Barabasi-Albert and shifted-linear preferential attachment networks, respectively.

(b) Exponential distribution for decay factor $q = 0.1, 0.5, 1, 1.5$. The special case of $q = 1$ corresponds to a uniform distribution supported in the interval $0 \le \theta \le \theta_{\max}$.

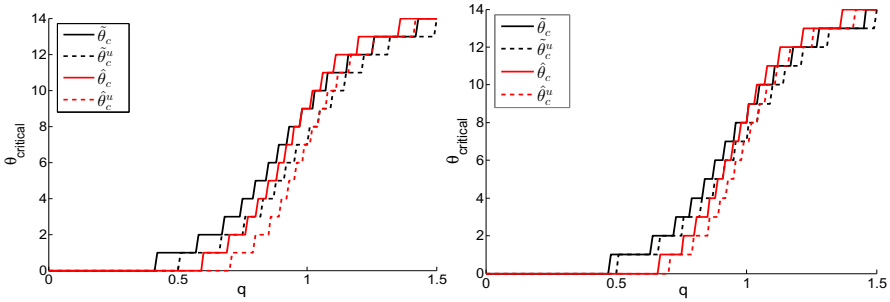
Fig. 1. Examples of the quality distributions used in this paper with $\theta_{\max} = 8$. Four instances of each type is depicted.



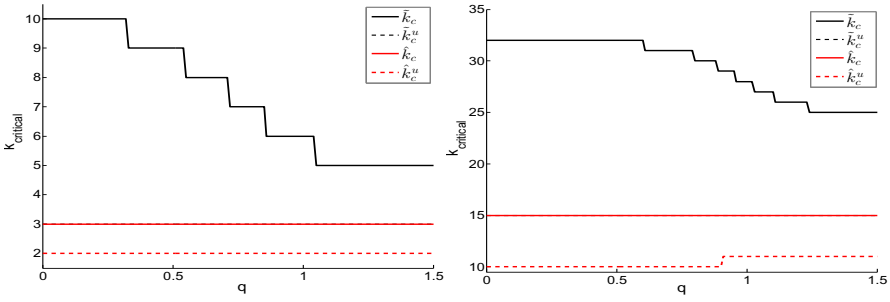
(a) Bernoulli, $\beta = 2$ $\theta_{max} = 24$ (b) Bernoulli, $\beta = 8$ $\theta_{max} = 24$



(c) Bernoulli, $\beta = 4$ $\theta_{max} = 16$ (d) Bernoulli, $\beta = 8$ $\theta_{max} = 16$



(e) Exponential, $\beta = 2$ $\theta_{max} = 16$ (f) Exponential, $\beta = 8$ $\theta_{max} = 16$



(g) Exponential, $\beta = 2$ $\theta_{max} = 16$ (h) Exponential, $\beta = 8$ $\theta_{max} = 16$

Fig. 2. Critical values for quality and degree as defined in (3) and (4) computed for Bernoulli and exponential quality distributions

Critical values obtained for two distributions are presented in Figure 2. These values are computed using the closed form expressions mentioned in Section 2. From Figure 2a we can learn about the differences between the networks that the QPA model generates and an uncorrelated network. In an uncorrelated network the probabilities of a random node being connected to a neighbor with quality 0 and $\theta_{\max}$ are equal to p and $1-p$, respectively (regardless of the quality of the node). If the majority of the neighbors have quality zero ($p \geq 0.5$), the median is zero. Similarly, if the majority have quality $\theta_{\max}$ ($p < 0.5$), the median is $\theta_{\max}$. So if $p < 0.5$, nodes with qualities up to $\theta_{\max} - 1$ experience the median QP and $\hat{\theta}_c^u = \theta_{\max} - 1$. Conversely, if $p \geq 0.5$, $\hat{\theta}_c^u = 0$. This explains the abrupt drop in $\hat{\theta}_c^u$ in Figure 2a. On the other hand, in the QPA model, this transition takes place at a p greater than 0.5. This means that upto some point beyond $p = 0.5$, although the probability of $\theta = 0$ is higher than that of $\theta = \theta_{\max}$, the majority of the friends of each node have quality $\theta_{\max}$. There is a region for $p > 0.5$, where the majority of the network have quality zero, but the majority of the neighbors of most nodes have quality $\theta_{\max}$. This indicates quality disassortativity, since low quality nodes are mostly connected to nodes with high qualities.

For the mean version of the QP, we consider the example case of $p = 0.2$ for discussion. In an uncorrelated network, each node (with any quality) is connected to neighbors with quality 0 and $\theta_{\max}$ with probabilities 0.2 and 0.8, respectively. So the average of the qualities of its neighbors is $0.8\,\theta_{\max}$. So nodes with quality less than $0.8\,\theta_{\max}$ experience the mean QP. On the other hand, in the QPA model $\widetilde{\theta}_c < \widetilde{\theta}_c^u$ at $p = 0.2$. This means that nodes whose qualities are between $\widetilde{\theta}_c$ and $\widetilde{\theta}_c^u$, do not experience the mean QP in the QPA model (while they do experience this paradox in the uncorrelated case). We deduce that these nodes are connected to quality zero nodes with a higher probability than 0.2. This reduces the average quality of their niehgbors. Now consider the example case of $p = 0.8$. In this case, $\widetilde{\theta}_c^u < \widetilde{\theta}_c$. This means that nodes with quality between $\widetilde{\theta}_c$ and $\widetilde{\theta}_c^u$ experience the mean QP in the proposed model, while they do not experience it in the uncorrelated case. In an uncorrelated network these nodes would be connected to zero and $\theta_{\max}$ quality nodes probabilities 0.8 and 0.2, respectively. However, in the QPA model, these nodes are connected to nodes with quality $\theta_{\max}$ with a probability higher than 0.2, and this increases the average quality of their neighbors, making them subject to the mean QP.

Comparing Figure 2b with 2a we observe the curves are similar, but the difference between the QPA model and the uncorrelated case is smaller in Figure 2b. For example, the drop in the $\hat{\theta}_c$ curve is closer to the drop in $\hat{\theta}_c^u$ for the uncorrelated case. We conclude that increasing β decreases the difference between the QPA model and the uncorrelated case.

In Figure 2c, critical degrees are depicted. It can be observed that as p increases, $\widetilde{k}_c$ increases. Comparing Figures 2c and 2d, we observe that all the critical degrees are greater in the case of $\beta = 8$ than $\beta = 4$. Also the range of node degrees experiencing any type of paradoxes is wider in the $\beta = 8$ case.

From Figure 2e, we observe that for fixed decay factor, $\widetilde{\theta}_c \geq \widetilde{\theta}_c^u$ and $\hat{\theta}_c \geq \hat{\theta}_c^u$. This means that there exist values of θ that in the uncorrelated network

experience QP, but in the proposed model they do not. So the range of possible values of quality that experience the QP is wider in the QPA model than in uncorrelated networks. This argument holds for both mean and median paradoxes.

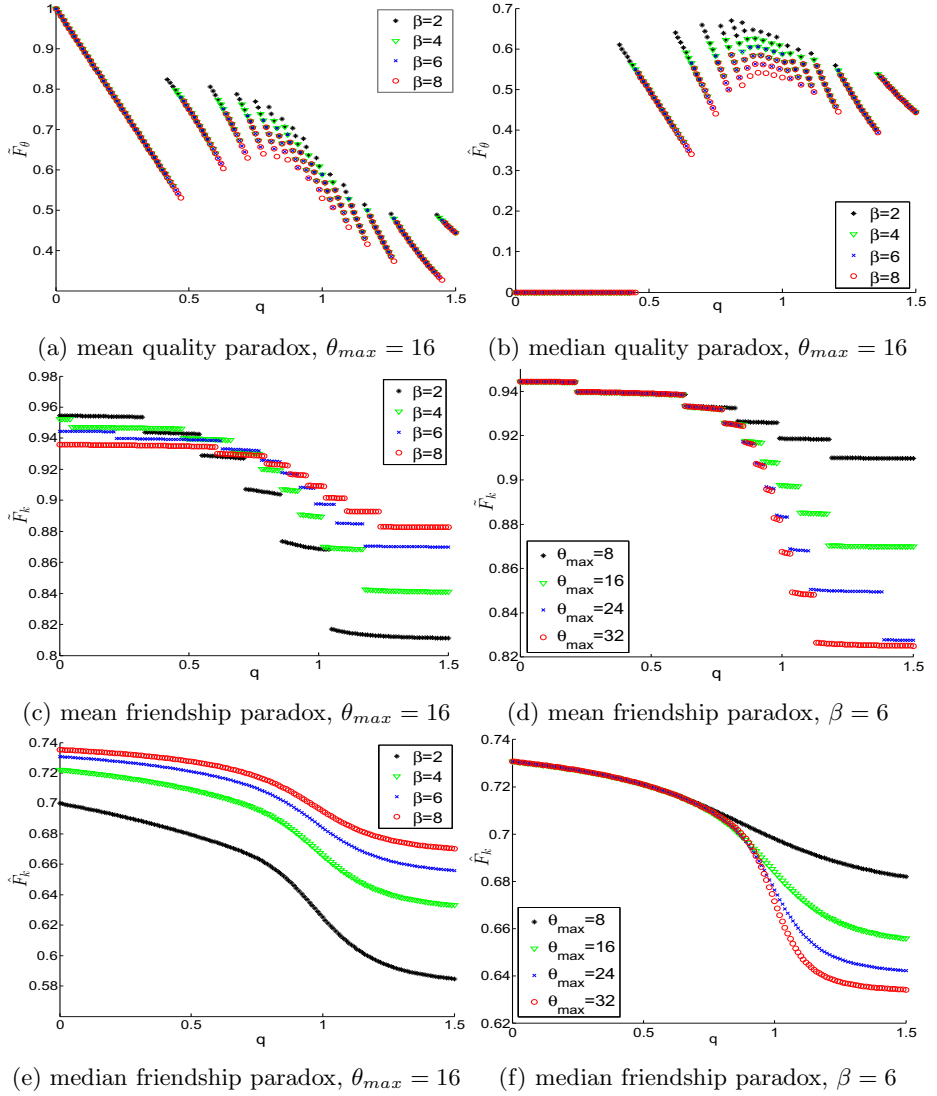
We also observe from Figure 2e that for $q < 1$, $\widetilde{\theta}_c \geq \hat{\theta}_c$ and $\widetilde{\theta}_c^u \geq \hat{\theta}_c^u$. Both of these inequalities flip in the case of $q > 1$. The main cause of this change of regime is the difference between the shape of the quality distribution for $q > 1$ and $q < 1$. When $q < 1$, the median paradox is stronger (using the terminology of [7]), that is, the median paradox applies to a smaller range of qualities than the mean paradox (for both the uncorrelated network and the QPA model). However, when $q > 1$, the median of the distribution is greater than the mean. As it can be observed in Figure 2e, there are values of θ that are subject to the median version of the paradox, but not to the mean version. This means that the term 'strong paradox' introduced in [7] is not applicable to this case, because the mean version provides a tighter range of qualities in paradox, as compared to the median version.

Another observable trend in Figure 2e is that the critical values of quality are a non-decreasing functions of q. This can be intuitively explained as follows. When q is low, the majority of the network is constituted by low quality nodes. The majority of the neighbors of a low quality node will also have low quality. So the node does not experience the paradox with high probability. When q increases, the number of nodes with higher quality increases, and a low quality node has a higher probability of being connected to those high quality nodes, which gives it a higher probability of experiencing paradox. Comparing Figure 2f with Figure 2e, we observe that as β varies $\widetilde{\theta}_c^u$ and $\hat{\theta}_c^u$ do not change, while the critical values of the QPA model get closer to those of the uncorrelated case. These figures only depict the results for two values of β, due to space limitations. The trend holds for the omitted figures. We conclude that *as β gets larger, the correlation of the quality of a node with the quality of its neighbors diminishes.*

In Figure 2g, the critical degrees (as defined in (3) and (4)) are depicted. It can be observed that as q increases, $\widetilde{k}_c$ decreases. Comparing Figures 2g and 2h, we observe that all the critical degrees are greater in the case of $\beta = 8$ than $\beta = 2$. Also the range of the degrees who experience paradox (of any type) is wider when $\beta = 8$. In both figures, we observe that the mean FP is more sensitive to changes in the quality distribution than the median FP.

Figure 3 depicts the fraction of nodes in the quality and friendship paradoxes (as defined in (5)) when quality distribution is exponential. From Figure 3a we observe that, as q increases in the vicinity of zero, $\tilde{F}_\theta$, the fraction of nodes experiencing the mean QP (with qualities lower than $\widetilde{\theta}_c$) decreases, because increasing q increases the fraction of nodes with high qualities. The fraction $\tilde{F}_\theta$ has discontinuities at the values of q at which $\widetilde{\theta}_c$ is incremented by one. So all the nodes whose qualities where equal to the new $\widetilde{\theta}_c$ are taken into account as those who experience the mean QP, hence the abrupt jump.

The fraction of nodes in the median QP is depicted in Figure 3b. It can be seen that $\hat{F}_\theta$ has a similar behavior to that of $\tilde{F}_\theta$. Each discontinuity pertains to a value of q at which $\hat{\theta}_c$ increments. The main difference between Figures 3a and 3b is the



(a) mean quality paradox, $\theta_{max} = 16$

(b) median quality paradox, $\theta_{max} = 16$

(c) mean friendship paradox, $\theta_{max} = 16$

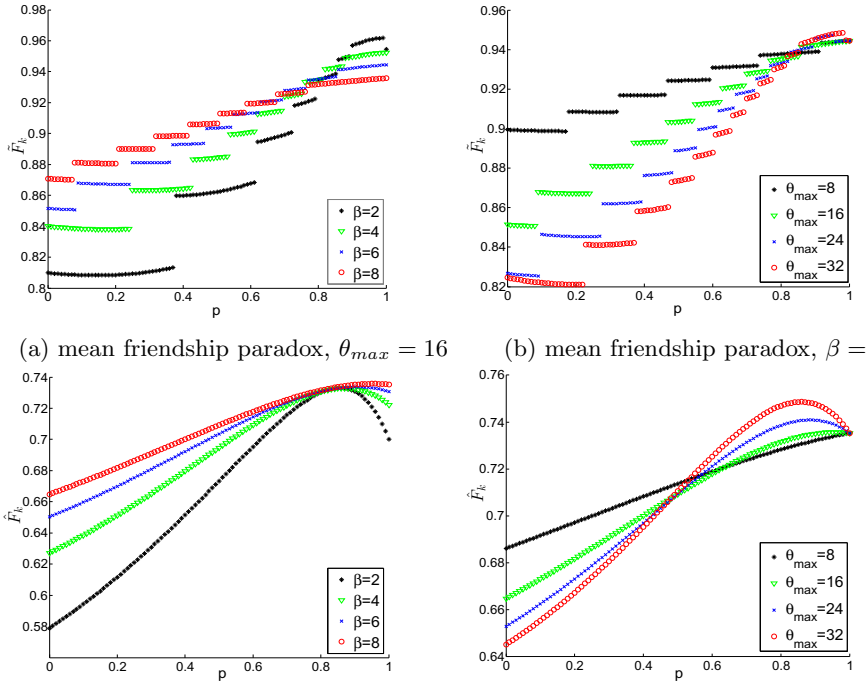
(d) mean friendship paradox, $\beta = 6$

(e) median friendship paradox, $\theta_{max} = 16$

(f) median friendship paradox, $\beta = 6$

Fig. 3. The fraction of nodes in the quality and friendship paradoxes when the quality distribution $\rho(\theta)$ is exponential

behavior near $q = 0$. In the mean QP, when almost all nodes have quality zero, even one non-zero quality neighbor elevates the average above zero, so all those zero-quality nodes experience the mean QP. However, in the median version, at least half of the friends of a zero-quality node must have non-zero quality. Also observe that for $q < 1$, we have $\widetilde{F}_\theta \geq \hat{F}_\theta$, i.e, the fraction of nodes in the mean QP is higher than the fraction of nodes in the median QP. But, for $q > 1$ the inequality changes sides.



(a) mean friendship paradox, $\theta_{max} = 16$ (b) mean friendship paradox, $\beta = 6$

(c) median friendship paradox, $\theta_{max} = 16$ (d) median friendship paradox, $\beta = 8$

Fig. 4. The fraction of nodes in the friendship paradox when the node quality distribution $\rho(\theta)$ is Bernoulli

In Figures 3c and 3d, it can be observed that *for all values of* β *and* $\theta_{\max}$*, the majority of the nodes (over* 80%*) experience the mean FP.* Also, as q increases, $\widetilde{F}_k$ decreases. It means that the quality distribution affects the FP that depends solely on degrees. Through the quality-dependant network growth mechanism, the degree distribution, and hence the conditions under which a node experiences the FP, depend on the quality distribution. Also, it is observed in Figure 3c that as β increases, the sensitivity of $\widetilde{F}_k$ to variations of q decreases. This means that as the initial degree of nodes increases, the effect of the quality distribution on the FP diminishes. Because as β increases the final degrees of nodes increase, and for larger degrees $k + \theta$ is dominated by k; varying θ has less of an effect. Conversely, in Figure 3d, as $\theta_{\max}$ increases, the sensitivity of $\widetilde{F}_k$ to variations of q increases. As the range of possible qualities becomes wider, the probability of having high values of θ that have significant roles in $k + \theta$ increases.

In Figures 3e and 3f, we observe that as q increases, $\hat{F}_k$ (the fraction of nodes experiencing the median FP) decreases. This is similar to the trend observed for $\widetilde{F}_k$ in Figures 3c and 3d. From Figure 3e we observe that $\hat{F}_k$ increases as β increases. From Figure 3f we observe that for a range of decay factors (up to around $q = 0.7$), $\theta_{\max}$ does not have a significant effect on $\hat{F}_k$, but beyond that

point, $\hat{F}_k$ decreases as $\theta_{\max}$ increases. Also, comparing Figures 3e and 3f with Figures 3c and 3d, we assert that $\hat{F}_k \leq \widetilde{F}_k$. In other words, *the median FP is always stronger than the mean FP, regardless of the quality distribution.*

The fraction of nodes experiencing the FP when the quality distribution is Bernoulli are depicted in Figure 4. From Figures 4a and 4b we observe that as p increases, $\widetilde{F}_k$ (the fraction of nodes experiencing the mean FP) increases. From Figure 4a we deduce that as β increases, the sensitivity of $\widetilde{F}_k$ to variations of p decreases. Also, in Figure 4b it is observed that as $\theta_{\max}$ increases, the sensitivity of $\widetilde{F}_k$ to variations of p increases (similar to Figures 3c and 3d).

From Figure 4c we observe that as β increases, $\hat{F}_k$ (the fraction of nodes experiencing the median FP) increases. From Figure 4d we observe that as $\theta_{\max}$ increases, the sensitivity of $\hat{F}_k$ to the variations of p increases. Comparing Figures 4a and 4b with Figures 4c and 4d we deduce that for each value of p, we have $\hat{F}_k \leq \widetilde{F}_k$, i.e., *the fraction of nodes experiencing the mean FP is higher than nodes in the median FP regardless of the quality distribution.*

5 Summary and Future Work

In this paper we studied the friendship and the generalized friendship paradoxes on networks grown under a quality-based preferential attachment scheme. To this end, we introduced measures, such as quality and degree critical values, and fraction of nodes that experience each paradox. In each case, we considered the mean and the median to characterize the paradox. We compared the results to the uncorrelated network where the qualities and degrees of neighbors are uncorrelated. We considered Bernoulli and exponential distributions for qualities.

For the exponential quality distribution, the critical quality of the uncorrelated case is always smaller than that of the QPA model. This means that the range of possible values of the quality that experience paradox is wider in the QPA model than in the uncorrelated case. We also observed that as β increases, the nearest-neighbor quality correlation decreases. In other words, the critical values of the proposed model converge to those of the uncorrelated case. For the exponential quality distribution we also observe that when $q < 1$ (which makes the median smaller than the mean), the median QP is stronger than the mean QP for both the QPA model and the uncorrelated case. The converse is true for $q > 1$. For all values of β, $\theta_{\max}$, over 80% of nodes experience the mean FP. We observed that changing the distribution of qualities affects the FP (in addition to the QP). This effect is strengthened when β decreases or when $\theta_{\max}$ increases. Also, it was observed that regardless of the quality distribution, the median FP is always stronger than the mean FP.

Plausible extensions of the present contribution are as follows. We can apply the measures introduced here to real networks, and compare the results, and also compare them with networks synthesized with arbitrary quality distributions. This enables us to investigate what type of quality distribution best characterizes a given network.

References

1. Feld, S.L.: Why Your Friends Have More Friends than You Do. American Journal of Sociology 96(6), 1464–1477 (1991)
2. Ezar, W., Zuckerman, J.T.: What Makes You Think You're So Popular? Self-evaluation Maintenance and the Subjective Side of the "Friendship Paradox". Social Psychology Quarterly 64(3), 207–223 (2001)
3. Hodas, N.O., Kooti, F., Lerman, K.: Friendship Paradox Redux: Your Friends Are More Interesting Than You. In: Proc. 7th Int. Conf. on Weblogs and Social Media, ICWSM 2013, pp. 1–8. ACM, New York (2013)
4. Ugandre, J., Karrer, B., Backstorm, L., Marlow, C.: The Anatomy of the Facebook Social Graph. arXiv preprint arxiv:1111.4503 (2011)
5. Eom, Y.H., Jo, H.H.: Generalized Friendship Paradox in Complex Networks: The Case of Scientific Collaboration. Nature Scientific Reports 4 (2014)
6. Jo, H.H., Eom, Y.H.: Generalized Friendship Paradox in Networks with Tunable Degree-attribute Correlation. Physical Review E 90(2) (2014)
7. Kooti, F., Hoda, N.O., Lerman, K.: Network Weirdness: Exploring the Origins of Network Paradoxes. In: Proc. 8th Int. Conf. on Weblogs and Social Media, ICWSM 2014, pp. 266–274 (2014)
8. Cohen, R., Havlin, S., Ben-Avraham, D.: Efficient Immunization Strategies for Computer Networks and Populations. Physical Review Letters 91(24), 247901 (2003)
9. Gracia-Herranz, M., Moro, E., Cebrian, M., Christakis, N., Fowler, J.: Using Friends as Sensors to Detect Global-Scale Contagious Outbreaks. PLos One 9, e92413 (2014)
10. Kryvasheyeu, Y., Chen, H., Moro, E., Van Hentenryck, P., Cebrian, M.: Performance of Social Network Sensors During Hurricane Sandy. arXiv preprint arXiv:1402.2482 (2014)
11. Barabási, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)
12. Fotouhi, B., Momeni, N., Rabbat, M.G.: Generalized Friendship Paradox: An Analytical Approach. Appearing in Proc. 6th Int. Conf. on Social Informatics (workshops), SocInfo 2014, arXiv preprint arXiv:1410.0586 (2014)

Expected Nodes: A Quality Function for the Detection of Link Communities

Noé Gaumont[1], François Queyroi[2], Clémence Magnien[1], and Matthieu Latapy[1]

[1] Sorbonne Universités, UPMC Univ Paris 06, UMR 7606, LIP6, F-75005, Paris, France
CNRS, UMR 7606, LIP6, F-75005, Paris, France
noe.gaumont@lip6.fr
[2] Géographie-Cités, CNRS - Univ Paris 01/07

Abstract. Many studies use community detection algorithms in order to understand complex networks. Most papers study node communities, *i.e.* groups of nodes, which may or may not overlap. A widely used measure to evaluate the quality of a community structure is the *modularity*. However, sometimes it is also relevant to study link partitions rather than node partitions. In order to evaluate a link partition, we propose a new quality function: *Expected Nodes*. Our function is based on the same inspiration as the modularity and compares, for a given link group, the number of incident nodes to the expected one. In this short note, we discuss the advantages and drawbacks of our quality function compared to other ones on synthetics graphs. We show that *Expected Nodes* is able to pass some fundamental sanity criteria and is the one that best identifies the most relevant partition in a more realistic context.

Keywords: complex networks, community detection, link partition, quality measure.

1 Introduction

In the past years, complex networks were extensively studied because of the broad range of systems they can model, from protein-protein interactions to social networks. One question of interest is the detection of communities. Despite the important literature that covers the detection of classical, overlapping or even dynamic communities, most works focus on grouping nodes. On the other hand, the question of link communities has received less attention [4,1,8]. Intuitively, partitioning a network's links is very relevant in some contexts. For example, in a social network, most individuals belong to multiple communities such as families, friends, and co-workers, while the links between individuals usually exist for a dominant reason. In this context, a link community would be a group of interactions on one topic.

In this paper, we address the problem of evaluating the quality of a link partition. After a review of previous works (Section 2), we introduce a novel measure: *Expected Nodes* (Section 3). It is based on the assumption that a link community corresponds to less individuals than expected while its surroundings links correspond to more individuals than expected. We use several test cases (Section 4) to study how this measure behaves when compared to other quality functions.

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_6

2 Related Work

We introduce some notations used throughout this paper. Let $G = (V, E)$ ge a graph, $d(u)$ denotes the degree of node u in G. A link partition in k groups is noted $\mathscr{L} = (L_1, L_2, \ldots, L_k)$ with $L_i \subseteq E \ \forall i$, $L_i \cap L_j = \emptyset \ \forall i \neq j$ and $\bigcup_i L_i = E$. For a given link group $L \in \mathscr{L}$, let $V_{in}(L) = \{u \in V, \exists (u,v) \in L\}$ be the group of nodes inside L and $V_{out}(L) = \{u \in V \setminus V_{in}(L), (u,v) \in E \setminus L \wedge v \in V_{in}(L)\}$ be the nodes adjacent to L.

Ahn *et al.* [1] were among the first to propose a method to detect link communities. Their method *link clustering* is a hierarchical clustering method constructing a dendrogram by iteratively merging groups of links according to a similarity measure based on the Jaccard index. To decide where to cut the dendrogram in order to create a partition, they use a density based measure: the *partition density*. For a given link partition $\mathscr{L}$, the *partition density* is given by:

$$D(\mathscr{L}) = \frac{2}{|E|} \sum_{L \in \mathscr{L}, |L|>2} |L| \frac{|L| - (|V_{in}(L)| - 1)}{(|V_{in}(L)| - 1)(|V_{in}(L)| - 2)}. \tag{1}$$

However, the *partition density* cannot be easily generalized to weighted networks. An attempt in this direction has been made by Kim [5].

Evans *et al.* [4] propose three quality functions to evaluate link partitions. Their quality functions can be computed and optimized on the original graph but also on specific weighted line graphs (LG_1, LG_2, LG_3) using existing algorithms such as the *Louvain* method [3]. A line graph of an undirected graph is a graph where each node represents a link from the original graph and two nodes are connected if the corresponding links share a node.

To define these particular line graphs LG_1, LG_2 and LG_3, let B denote the incidence matrix of a network G: the elements $B_{i\alpha}$ of this $|V| \times |E|$ matrix are equal to 1 if link α is connected to node i and 0 otherwise. Matrices LG_1, LG_2 and LG_3 are defined as:

	$x = 1$	$x = 2$	$x = 3$
$LG_x(\alpha, \beta)$	$B_{i\alpha} B_{i\beta} (1 - \delta_{\alpha\beta})$	$\sum_{i \in V, d_G(i)>1} \frac{B_{i\alpha} B_{i\beta}}{d(i) - 1}$	$\sum_{i,j \in V, d(i) d_G(j)>0} \frac{B_{i\alpha} A_{ij} B_{j\beta}}{d(i) d(j)}$

Let $k_x(\alpha) = \sum_\beta LG_x(\alpha, \beta)$ be α's weighted degree in LG_x and $W_x = \sum_{\alpha,\beta \in |E|} LG_x(\alpha, \beta)$. For $x \in \{1, 2, 3\}$, the quality function $Evans_x$ is:

$$Evans_x(\mathscr{L}) = \frac{1}{W_x} \sum_{L_i \in \mathscr{L}} \sum_{e_1, e_2 \in L_i^2} LG_x(e_1, e_2) - \frac{k_x(e_1) k_x(e_2)}{W}. \tag{2}$$

Kim *et al.* [6] explored the extension of the concept of Minimum Length Description introduced by Rosvall *et al.* [10] which is an information-theoretic framework. This extension directly considers link partitions. An advantage of their method is the ability to compare link and node partitions.

3 Our Quality Function: *Expected Nodes*

One commonly accepted assertion for node communities is: a community should have more internal connections than the expected number of connections in a random null

model where no community structure exists. This assertion is at the core of the modularity introduced by Newman and Girvan [9]. In the same way, to evaluate a link community, we compare the number of nodes to its expected number of nodes. Like *modularity*, the measure can be decomposed, for each group L, into two components: internal quality and external quality. Like *modularity*, we use the configuration model [2] for a null model. In this model, the links are created by choosing random pairs of half-link (or stubs), each node having as many stubs as is degree in the original graph.

We start by describing the internal quality. Intuitively, a group of links L is a relevant community if it consists of a large number of links adjacent to few nodes, *i.e.* if V_{in} is small compared to what would be expected in the configuration model. By definition, a node is an internal node of L if one of its stubs (half-links) is in L. Therefore, to compute the expected number of internal nodes in the configuration model, we choose randomly $2|L|$ stubs among a total of $2|E|$ stubs. A node u has therefore $d(u)$ ways to be picked. The expected number of internal nodes for a given link group L, denoted by $\mu_G(|L|)$, is then:

$$\mu_G(|L|) = \sum_{u \in V} \mathbb{P}(u \textit{ picked at least once}) = \sum_{u \in V} 1 - \frac{\binom{2|E|-d(u)}{2|L|}}{\binom{2|E|}{2|L|}}. \tag{3}$$

Note that the function μ_G only depends on the degree sequence $\{d(v)\}_{v \in V}$. Note also that if $|L| = 1$, then $\mu_G(|L|) \leq 2$; this is because the configuration model allows self loops. A group L has a good internal quality if it has less internal nodes than expected. We therefore choose to define the internal quality function Q_{in} for a given group L as the variation between the actual number of internal nodes and its expectation:

$$Q_{in}(L) = \frac{\mu_G(|L|) - |V_{in}(L)|}{\mu_G(|L|)}. \tag{4}$$

With this definition, for a given $|L|$, the fewer nodes a group of links involves, the higher Q_{in} will be.

We now describe the external quality of a group L. The process to evaluate the neighbourhood $V_{out}(L)$ of a group L is similar to the process for the internal nodes. However in this case, we consider that L has a bad neighbourhood if it has fewer external nodes than expected. Indeed if there are many external links and few external nodes, these external links should be included in the community. Let $\bar{d}(L,u) = \sum_{v \in V} \mathbb{1}_{(u,v) \in E \setminus L}$ be the degree of u restricted to links not in L and $\bar{d}(L) = \sum_{u \in V_{in}(L)} \bar{d}(L,u)$. The expectation of the number of adjacent nodes is evaluated as the number of nodes that are picked when $\bar{d}(L)$ stubs are chosen randomly in the configuration model where the links of L have been removed. The corresponding degree sequence is $\{d_{G \setminus L}(u)\}_{u \in V}$ where $G \setminus L = (V, E \setminus L)$. Only one half link is chosen randomly because the other half has to remain attached to an internal node of L. Thus, we have the following equation:

$$\mathbb{E}[\bar{d}(L)] = \mu_{G \setminus L}(\bar{d(L)}/2). \tag{5}$$

Since we are interested in penalizing groups that have few external nodes, but do not consider that a group is particularly good if it has a large number of external nodes, we

bound the external quality by 0:

$$Q_{ext}(L) = min\left(0, \frac{|V_{out}(L)| - \mu_{G\setminus L}(\bar{d}(L)/2)}{\mu_{G\setminus L}(\bar{d}(L)/2)}\right). \tag{6}$$

Finally, we define *Expected Nodes* for a group L as:

$$Q(L) = 2\frac{|L|Q_{in}(L) + |L_{out}|Q_{ext}(L)}{|L| + |L_{out}|}. \tag{7}$$

Notice that the trivial group containing all links has a null quality because Q_{in} and Q_{out} will be equal to 0. The other trivial decomposition where each link belongs to its own group has a negative quality. However, in some cases a group containing a single link might be the best choice. It is the case when the link is a bridge between dense groups. Finally, we define *Expected Nodes* for a given link partition $\mathscr{L}$ as the weighted sum of the quality of each group:

$$Q_G(\mathscr{L}) = \frac{\sum_{L\in\mathscr{L}} |L|Q(L)}{|E|}. \tag{8}$$

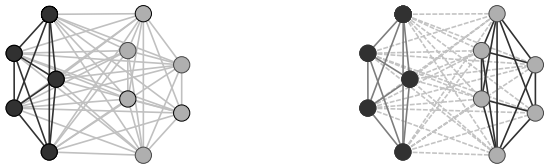
4 Comparison with Existing Methods

In order to study the relevance of *Expected Nodes*, we use two test cases.We also compare it to acknowledged quality functions: *partition density* [1] and the quality functions developed by Evans *et al.* [4] denoted by $Evans1$, $Evans2$ and $Evans3$.

4.1 Complete Graph

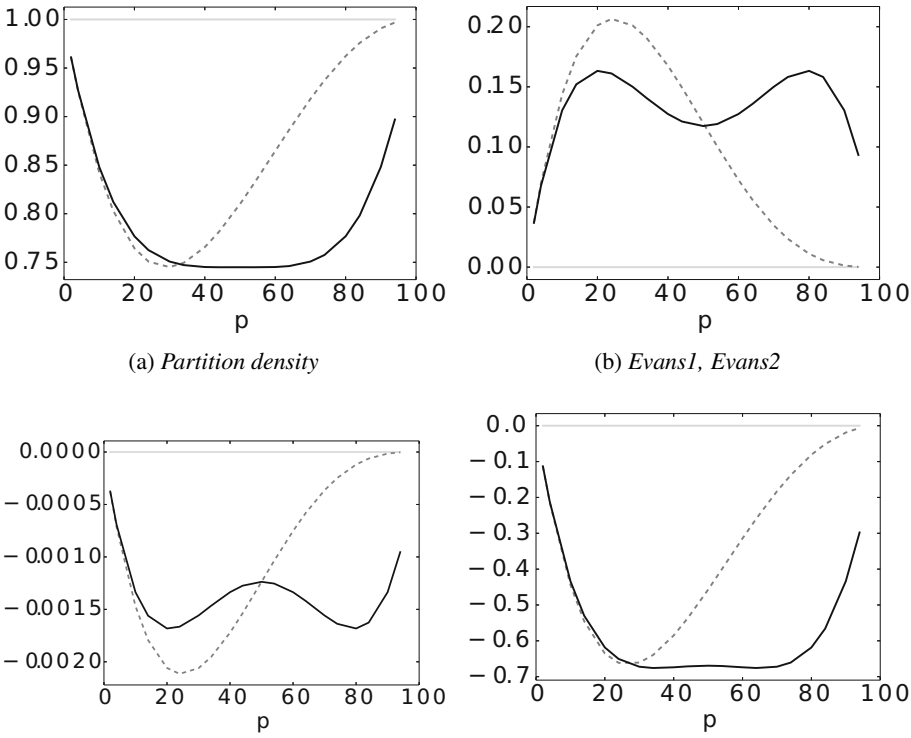
We start with a simple case in order to check that *Expected Nodes* satisfies some important and fundamental properties. We study a complete graph of 100 nodes (we obtained similar results with different sizes). On this graph, we define the trivial partition with one group containing all links, and two partitions families: one with two groups and one with three groups. Given a parameter $p < |V|$, let V' be a set of p nodes. Both partitions place all links in $V' \times V'$ in one group. The 2-groups partition places all other links in the second group. The 3-groups partition places all links in $V \times V \setminus V'$ in a second group and all remaining links in the third. These assignment rules are illustrated in Figure 1

As the graph is a single complete graph, the best solution is to capture only one group with all the links, *i.e.* the trivial partition should have the highest ranking. Figure 2 shows the results. For each value of p and each quality function, we present the values for the corresponding partitions in 2 and 3 groups and for the trivial partition. Surprisingly, quality functions $Evans1$ and $Evans2$ fail this simple test because they evaluate the 2- or 3-groups partitions as better than the trivial one. According to *partition density*, *Expected Nodes* and E_3, the trivial partition is best. The quality function $Evans3$ differs because of its small amplitude ($\approx 10^{-3}$).



(a) (b)

Fig. 1. Two different link partitions of a complete graph with $p = 5$: (a) in two link groups and (b) in three link groups. The dark nodes corresponds to V' and the color of a link corresponds to its group.



(a) *Partition density* (b) *Evans1, Evans2*

(c) *Evans*3 (d) Expected Nodes

Fig. 2. Evaluation of 5 quality functions on a complete graph of 100 nodes for 3 different partitions. The tested partitions are presented in Section 4.1. The results for quality functions *Evans*1 and *Evans*2 are identical. The gray line, black line and dashed line represent respectively the 1-group partition, the 2-groups partition and the 3-groups partition.

4.2 Overlapping LFR Benchmark

We now discuss results obtained by comparing the quality functions on random networks with a known community structure. To the best of our knowledge, there is no graph generator based on link partitions. We use the benchmark proposed by Lancichinetti *et al.* [7] which generates graphs based on a known node cover. We introduce two transformations of this overlapping community structure into link partitions denoted by TA and TB (see Figure 3). Given $u, v \in V$, let $C_{u,v}$ denote the intersection between the communities of u and v in the node cover and $U_{u,v}$ their union. We define the group of a link $(u,v) \in E$ in the partitions as follows:

intra-community if $|C_{u,v}| = 1$ then (u,v) is in community $C_{u,v}$;
inter-community if $|C_{u,v}| = 0$ then in TA, (u,v) belongs to its own community. In TB it belongs to community $U_{u,v}$ which contains all links (u',v') such that $U_{u',v'} = U_{u,v}$;
overlapping if $|C_{u,v}| > 1$ then (u,v)'s community is chosen randomly in $C_{u,v}$.

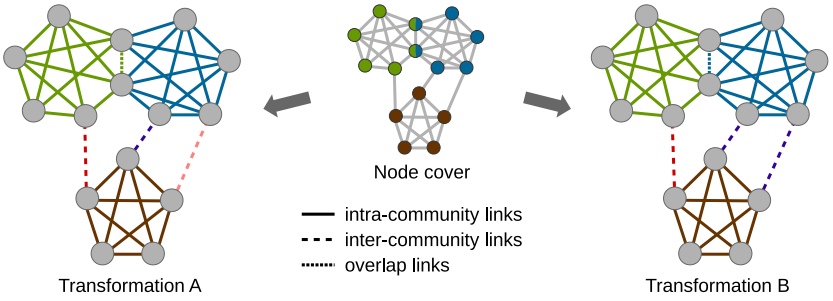


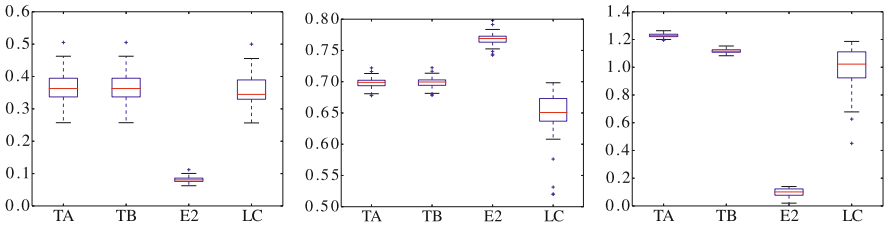
Fig. 3. Construction of TA and TB from a node cover. Link colours denote groups.

We describe the results averaged over 30 graph generations with 500 nodes, an average degree of 25, 10 overlapping nodes and a mixing parameter of 0.1[1]. There are on average 5620 intra-community links, 625 inter-community links and only 5 overlapping links. For each generation, the partition TA, TB, the partition LC found by *link clustering* [1] and the partition $E2$ found by the second method of Evans *et al.* [4] (based on the optimization of *Evans*2)[2] are evaluated using *Partition Density*, *Evans*2 and *Expected Nodes*.

In TA (resp. TB), there are 650 (resp. 70) groups on average. Manual investigations show that the $E2$ partitions are very close to the ground truth (TA or TB) if inter-community links are not considered. Indeed in $E2$, inter-community links are randomly distributed among adjacent larger link communities. The LC partitions contain 720 groups on average and intra-community links are split into many small groups. Notice that neither TA nor TB get the best evaluation according to *Evans*2 and *Partition density* even though they are considered as ground truth.

[1] Remaining parameters with original notations: $k_{max} = 50, t_1 = -2, t_2 = -1, C_{min} = 20, C_{max} = 100$.

[2] The results are similar for the algorithms using $E1$ and $E3$.



(a) Partition density (b) Evans2 (c) Expected Nodes

Fig. 4. Boxplots of the three quality functions values for the different link partitions. The box shows lower and upper quartiles and the median. The whiskers extend to 1.5 time the interquartile range. Flier points are those past the end of the whiskers.

The following observations can be made. First, *Expected Nodes* (Fig. 4c) behaves differently than both other measures (Fig. 4a and 4b). This shows that our measure brings something new to the picture. Moreover, its values are usually higher for TA and TB than for the partitions found using the two algorithms, which corresponds to our expectations. Second, the *Expected Nodes* values are significantly different for TA and TB. It is not the case for quality functions $E2$ and *Partition density*. Indeed, external links between the same community are likely to form a group of isolated links in TB. This situation is highly penalized by our measure. It also explains why *Expected Nodes* evaluates LC partitions better than $E2$ partitions. For those reasons, we believe that maximizing *Expected Nodes* would result in partitions close to TA in this benchmark.

4.3 Conclusion

In this paper, we propose a new quality function, *Expected Nodes*, to evaluate the quality of a link partition of a graph[3]. It compares the number of nodes adjacent to a link group to its expectation, in the same way as the modularity evaluates the relevance of a node group by comparing the number of adjacent links to its expected value. To show the relevance of *Expected Nodes*, we compared it to existing quality functions. The main perspective of our work is to design an algorithm for maximizing *Expected Nodes* in order to detect relevant link partitions. More detailed comparisons between quality functions may also be performed. For instance, it would be interesting to evaluate their behaviour to detect whether they are likely to present local maxima such as the one observed in Figure 2c or not.

Acknowledgements. This research was supported by a DGA-MRIS scholarship, by a grant from the French program "*PIA – Usages, services et contenus innovants*" under grant number $O18062 - 44430$ and by the CODDDE project ANR-13-CORD-0017-01.

[3] The code used to compute each quality function is available:
`https://github.com/ksadorf/ExpectedNodes`

References

1. Ahn, Y.-Y., Bagrow, J.P., Lehmann, S.: Link communities reveal multiscale complexity in networks. Nature 466(7307), 761–764 (2010)
2. Bender, E.A., Canfield, E.: The asymptotic number of labeled graphs with given degree sequences. Journal of Combinatorial Theory, Series A 24(3), 296–307 (1978)
3. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (2008)
4. Evans, T.S., Lambiotte, R.: Line graphs, link partitions, and overlapping communities. Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics 80(1), 016105 (2009)
5. Kim, S.: Community Detection in Directed Networks and its Application to Analysis of Social Networks. PhD thesis, Ohio State University (2014)
6. Kim, Y., Jeong, H.: Map equation for link communities. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 84(2), 026110 (2011)
7. Lancichinetti, A., Fortunato, S.: Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics 80, 016118 (2009)
8. Lim, S., Ryu, S., Kwon, S., Jung, K., Lee, J.-G.: LinkSCAN*: Overlapping community detection using the link-space transformation. In: 2014 IEEE 30th International Conference on Data Engineering, pp. 292–303. IEEE (March 2014)
9. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E - Statistical, Nonlinear, and Soft Matter Physics 69 (2004)
10. Rosvall, M., Bergstrom, C.T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences of the United States of America 105(4), 1118–1123 (2008)

Core-Periphery Models for Graphs Based on their δ-Hyperbolicity: An Example Using Biological Networks

Hend Alrasheed and Feodor F. Dragan

Department of Computer Science, Kent State University, Kent, OH 44242, USA
halrashe@kent.edu, dragan@cs.kent.edu

Abstract. Hyperbolicity is a global property of graphs that measures how close their structures are to trees in terms of their distances. It embeds multiple properties that facilitate solving several problems that found to be hard in the general graph form. In this paper, we investigate the hyperbolicity of graphs not only by considering Gromov's notion of δ-hyperbolicity but also by analyzing its relationship to other graph's parameters. This new perspective allows us to classify graphs with respect to their hyperbolicity, and to show that many biological networks are hyperbolic. Then we introduce the *eccentricity-based bending property* which we exploit to identify the core vertices of a graph by proposing two models: *the Maximum-Peak model* and *the Minimum Cover Set model*.

1 Introduction

Using graph-theoretical tools for analyzing complex networks aids identifying multiple key properties as well as explaining essential behaviors of those systems. A common structure in many network disciplines is the core-periphery structure which suggests partitioning the graph into a dense core and sparse periphery. Vertices in the periphery interact through a series of core vertices. This pattern of communication (where traffic tends to concentrate on a subset of vertices) has been observed in trees where distant nodes communicate via the central nodes. δ-Hyperbolicity, which is a measure that shows how close a graph is to a tree, suggests that any shortest path between any pair of vertices bends (to some extent) towards the core. This phenomenon has been justified by the negative curvature which in case of graphs can be measured using hyperbolicity [24].

Multiple complex networks such as the Internet [28,14], data networks at the IP layer [24], and social and biological networks [4,2] show low δ-hyperbolicity (low hyperbolicity suggests a structure that is close to a tree structure [14,3]). Also, it has been observed that networks with this property have highly connected cores [24]. Generally, the core of a graph is specified according to one or more centrality measures. For example, the betweenness centrality and the eccentricity centrality. The δ-hyperbolicity of graphs embeds multiple properties that facilitate solving several problems that found to be difficult in the general graph form; for example, diameter estimation [9] and compact distance and

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_7

routing labeling schemes [10,12]. In this paper, we investigate implications of the δ-hyperbolicity of a graph and exploit them for the purpose of partitioning the graph into core and periphery parts. Our main contributions can be summarized as follows.

(a) We study the hyperbolicity of several biological networks and show that the hyperbolicity of almost all the networks in our dataset is small. This confirms the results in [4]. However, unlike previous efforts, we analyze the relationship between the hyperbolicity and other global parameters of the graph. We find in most of our networks that the hyperbolicity is bounded by the logarithm of the graph's diameter and the logarithm of the graph's size. Based on this we classify graphs into: *strongly-hyperbolic*, *hyperbolic*, and *non-hyperbolic*.

(b) We formalize the notion of the *eccentricity layering* of a graph and employ it to introduce a new property that we find to be intrinsic to hyperbolic graphs: *the eccentricity-based bending property*. Unlike previous work, we investigate the essence of this bending in shortest paths by studying its relationship to the distance between vertex pairs.

(c) We exploit the eccentricity-based bending property by proposing two core-periphery separation models. We apply both models to our datasets. In contrast to what was observed in [18], we find that biological networks exhibit a clear-cut core-periphery structure. Some details were omitted in this conference version of the paper. Interested readers can refer to [1].

2 Theoretical Background and Related Work

Preliminaries on Graph Theory. A simple undirected graph $G = (V, E)$ naturally defines a metric space (V, d) on its vertex set V. The distance $d(u, v)$ is defined as the number of edges in a shortest path $\rho(u, v)$ that connects two vertices u and v. We define the *size* of the graph denoted as $size(G)$ as $size(G) = |V|+|E|$. The *diameter* of the graph $diam(G)$ is the length of the longest shortest path between any two vertices u and v, i.e., $diam(G) = \max_{u,v \in V}\{d(u, v)\}$. The *eccentricity* of a vertex u is $ecc(u) = \max_{v \in V}\{d(u, v)\}$, i.e., the distance between u and any of its farthest neighbors v. The minimum value of the eccentricity represents the graph's *radius*: $rad(G) = min_{u \in V}\{ecc(u)\}$. The set of vertices with minimum eccentricity are considered the *center* of the graph $C(G)$. In other words, $C(G) = \{u \in V : ecc(u) = rad(G)\}$.

δ-Hyperbolicity. The δ-hyperbolicity measure of a metric space was proposed by Gromov [17]. It measures how close the metric structure is to a tree structure. A connected graph G can be viewed as a metric space with the graph distance metric d. There are multiple equivalent definitions (up to constant factors [9]) for Gromov's hyperbolicity. Here we use the four-point condition definition.

Given a graph $G = (V, E)$, x, y, u, and $v \in V$ are four distinct vertices, and the three sums: $d(x, y) + d(u, v)$, $d(x, u) + d(y, v)$, and $d(x, v) + d(y, u)$ sorted in a non-increasing order, the hyperbolicity of the quadruple x, y, u, v is defined as: $\delta(x, y, u, v) = ((d(x, y) + d(u, v)) - (d(x, u) + d(y, v)))/2$. The δ-hyperbolicity of the graph G denoted as $\delta(G)$ (or simply δ) is $\delta(G) = \max_{x,y,u,v \in G} \delta(x, y, u, v)$.

For finite graphs δ-hyperbolicity is finite. Consequently, one can think of all finite graphs as hyperbolic except that the value of δ decides how *hyperbolic* the graph is. On the other hand, when no finite δ exists (which may be the case for infinite graphs), the graph is considered non-hyperbolic [3]. Generally, the smaller the value of δ the closer the graph is to a tree (metrically).

Core-Periphery and Network Centrality in Complex Networks. In [6], the authors formalize the core-periphery structure by developing two models: the discrete model where vertices belong to one of two classes (core and periphery) and the continuous model which includes three classes or more of vertices. Holme in [18] introduces a coefficient that measures if a network has a clear core-periphery structure based on the closeness centrality. Structure analyses of some biological networks have detected the presence of the core-periphery organization. [13] proposes a parameter that detects the existence of a core-periphery structure in a metabolic network based on the closeness centrality. [16] studies recognizing the central metabolites in a metabolic network. In [21], the authors identify the central metabolites using degree and closeness centrality.

In the study of communication networks, the core is usually identified by the small dense part that carries out most traffic under shortest path routing [5,24]. It is quite natural to associate the concepts of the network's core and its center. In [6], the authors argue that each central vertex is a core vertex; consequently, all coreness measures are centrality measures. The notion behind centrality is identifying vertices that are high contributors. There are multiple centrality measures in the literature. The betweenness centrality expresses how much effect each vertex has in the communication. Given a connected finite graph $G = (V, E)$, the betweenness centrality of a vertex $u \in V$ measures the total number of shortest paths between every pair of vertices x and y that pass through u. The eccentricity centrality suggests that the center of the graph includes the vertex (or vertices) that has the shortest distance to all other vertices.

3 Datasets

We analyze the protein interaction networks of Budding yeast [7], Escherichia coli [8], Yeast [11], Saccharomyces cerevisiae [19], and Helicobacter Pylori [26]. Also, we analyze two brain area networks of the macaque monkey [25] [23]; and the metabolic networks of the Escherichia coli [20] and the Caenorhabditis elegans [15]. Finally, we analyze the yeast transcription network [22]. In this work, we consider unweighted graphs, and we only consider the largest connected component of each network. The size of this component for each network is presented in Table 1. We also ignore the directions of the edges.

4 δ-Hyperbolicity of Networks

For the purpose of investigating the hyperbolicity of networks, it seems natural to analyze and classify them based on their hyperbolicity. The classification should reflect how strong (evident) the tree-likeness is in the graph's structure.

Table 1. Graph datasets and their parameters: number of vertices $|V|$; number of edges $|E|$; graph's size $size(G)$; average degree $\bar{d}$; diameter $diam(G)$; radius $rad(G)$; hyperbolicity $\delta(G)$; and the average hyperbolicity $\delta^{'}(G)$

Network Category	Network	$\|V\|$	$\|E\|$	$\log_2(size(G))$	$\bar{d}$	$diam(G)$	$rad(G)$	$\delta(G)$	$\delta^{'}(G)$
PI Networks	B-yeast-PI	1465	5839	12.8	7.97	8	5	2.5	0.299
	E-coli-PI	126	581	9.5	9.2	5	3	2	0.251
	Yeast-PI	1728	11003	13.6	12.7	12	7	3.5	0.322
	S-cerevisiae-PI	537	1002	10.5	3.7	11	7	4	0.419
	H-pylori-PI	72	112	7.5	3.1	7	5	3	0.368
Neural Networks	Macaque-brain-1	45	463	9	11.3	4	2	1.5	0.231
	Macaque-brain-2	350	5198	12.4	29.7	4	3	1.5	0.203
Metabolic Networks	E-coli-metabolic	242	376	9.3	3.1	16	9	4	0.483
	C-elegans-metabolic	453	4596	12.3	8.9	7	4	1.5	0.133
Transcription Networks	Yeast-transcription	321	711	10	4.4	9	5	3	0.365

Hyperbolicity of Biological Networks. We measure δ-hyperbolicity using Gromov's four-point condition. For each network, we identify a bi-connected component with the maximum value of δ since the hyperbolicity of a graph equals the maximum hyperbolicity of its bi-connected components [14].

Table 1 shows that almost all networks in our datasets have small hyperbolicity. Even though the definition of δ-hyperbolicity considers the difference between the largest two distance sums among any quadruples and takes into account only the maximum one, this absolute analysis is deficient. Similar to [14,3], closer analysis to the distribution of the value of δ (see Figure 1) shows that only a very small percent of the quadruples have the maximum value of δ while most quadruples have $\delta = 0$. This observation makes it equally important to calculate the value of the average delta $\delta'(G)$ (see Table 1).

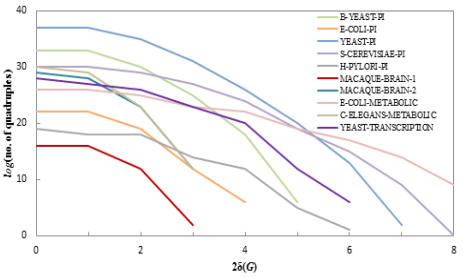


Fig. 1. The distribution of the quadruples over different values of δ

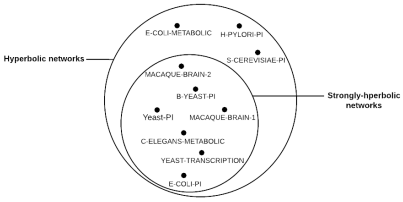


Fig. 2. Classification of the graph datasets based on their hyperbolicity

Analysis and Discussion. Our goal is to categorize graphs with respect to their hyperbolicity into three classes: *strongly-hyperbolic, hyperbolic*, and *non-hyperbolic*. Studying the tree-like structure of graphs based solely on the value

of the hyperbolicity may not be sufficient for two reasons. First, the hyperbolicity is a relative measure. For example, for a given graph, a value of $\delta(G) = 10$ can be seen as too large when $size(G) \simeq 10^2$. However, when $size(G) \simeq 10^7$, the hyperbolicity $\delta(G) = 10$ looks much smaller. Second, small graph size and (or) small diameter directly yield low hyperbolicity. In other words, small $\delta(G)$ does not always suggest a graph with a tree-like structure; other graph attributes that might impact the hyperbolicity must be investigated. We find $size(G)$ and $diam(G)$ play an important role in deciding how hyperbolic a given graph is.

Since finite graphs will always have a finite value for δ such that the four-point condition is true, it is natural to think that the non-hyperbolic class includes only infinite graphs. However, in this study, we only consider finite graphs; accordingly, a non-hyperbolic graph in our sense is a graph with too large δ with respect to the logarithm of the graph's size, i.e., when it violates $\delta(G) \leq \log_2(size(G))$.

In cases where $\delta(G) \leq \log_2(size(G))$, we move on and compare δ with the diameter. To guarantee that the value of the diameter is not directly impacted by the graph's size, first we require that $diam(G) \leq \log_2(size(G))$. Multiple previous works have analyzed the relationship between $\delta(G)$ and the diameter.

Lemma 1 ([27]). *For any graph G with diameter $diam(G)$ and hyperbolicity $\delta(G)$, $\delta(G) \leq diam(G)/2$.*

Interestingly, for most of the networks in our graph datasets, we find that $\delta(G) \leq \log_2(diam(G))$. Therefore, we say that a graph is *strongly-hyperbolic* if it exhibits (1) $diam(G) \leq \log_2(size(G))$ and (2) $\delta(G) \leq \log_2(diam(G))$ (small-world), *hyperbolic* when it violates either (1) or (2), and *non-hyperbolic* whenever it has a large δ, i.e., $\delta > \log_2(size(G))$. As Table 1 shows, all networks in the datasets, with the exception of S-CEREVISIAE-PI and E-COLI-METABOLIC, exhibit the small-world property. Also, it shows that $\delta(G) \leq \log_2(diam(G))$ in all graphs except for the S-CEREVISIAE-PI and the H-PYLORI-PI networks. As a result, those three graphs have been classified as hyperbolic graphs, and their $\delta(G)$ and $\delta^{'}(G)$ values are on the larger side. In Figure 2, we show this classification.

Quantifying "small" and "large" for δ is not straightforward simply because it is relative. Therefore, we judge according to the difference between δ and $\log_2(\log_2(size(G)))$. The more substantial this difference is the closer the graph's structure to a tree structure. For example network C-ELEGANS-METABOLIC is metrically closer to a tree than network YEAST-PI.

5 Core-Periphery Models Based on δ-Hyperbolicity

In this section, we formalize the notion of bending in shortest paths by introducing the *eccentricity-based bending property*. Then we use the implication of this property to aid the partitioning of a graph into core and periphery parts.

Eccentricity Layering of a Graph. The *eccentricity layering* of a graph $G = (V, E)$ denoted as $\mathcal{EL}(G)$ partitions its vertices into concentric circles or layers $\ell_r(G)$, $r = 0, 1, \ldots$. Each layer r is defined as $\ell_r(G) = \{u \in V :$

$ecc(u) - rad(G) = r\}$. Here r represents the index of the layer. The inner-most layer (layer 0) encloses the graph's center $C(G)$. Then the first layer includes all vertices with eccentricities equal to $rad(G) + 1$, and so on. The vertices in the last layer (outer-most) have eccentricities equal to the diameter. Any vertex $v \in \ell_r(G)$ has *level* (or layer) $level(v) = r$. Figure 3 gives an illustration. We noticed that the vertices' population is denser in the middle layers in almost all networks.

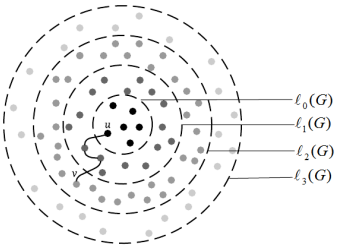


Fig. 3. Eccentricity layering of a graph. Darker vertices belong to lower layers.

Table 2. The effect of the distance k between vertex pairs on the bending property. Out of all vertex pairs with distance at least k, we show the percentage of those that bend for three networks.

k	C-elegans -metabolic ($diam(G) = 7$)	B-yeast -PI ($diam(G) = 8$)	Yeast -transcription ($diam(G) = 9$)
2	96.99%	93.10%	96.65%
3	99.89%	94.87%	97.77%
4	100%	98.43%	99.11%
5	100%	99.93%	99.88%
6	100%	100%	100%
7	100%	100%	100%
8		100%	100%
9			100%

5.1 Eccentricity-Based Bending Property of δ-Hyperbolic Networks

Let $G = (V, E)$ be a δ-hyperbolic graph, $\mathcal{EL}(G)$ be its eccentricity layering, and $C(G)$ be its center. In [9], the following useful metric property of δ-hyperbolic graphs was proven.

Lemma 2 ([9]). *Let G be a δ-hyperbolic graph and x, y, v, u be its four arbitrary vertices. If $d(v, u) \geq \max\{d(y, u), d(x, u)\}$, then $d(x, y) \leq \max\{d(v, x), d(v, y)\} + 2\delta$.*

We use this property to establish the following few interesting statements. The proofs are omitted in this version. We direct interested readers to [1].

Proposition 1. *Let G be a δ-hyperbolic graph and x, y, s be arbitrary vertices of G. If $d(x, y) > 4\delta + 1$, then $d(w, s) < \max\{d(x, s), d(y, s)\}$ for any middle vertex w of any shortest (x, y)-path.*

Proposition 2. *Let G be a δ-hyperbolic graph and x, y be arbitrary vertices of G. If $d(x, y) > 4\delta + 1$, then on any shortest (x, y)-path there is a vertex w with $ecc(w) < \max\{ecc(x), ecc(y)\}$.*

We define the bend in shortest paths between two distinct vertices u and v with $d(u, v) \geq 2$, denoted by $bend(u, v)$, as follows $\forall\ u, v \in V$ $bend(u, v) =$

$min\{level(z) : z \in V \text{ and } d(u,z) + d(z,v) = d(u,v)\}$. Here $level(z) = r$ iff $z \in \ell_r(G)$. We say that shortest paths between u and v bend if and only if a vertex z with $ecc(z) < \max\{ecc(u), ecc(v)\}$ exists in a shortest path between them. In this case we say also that *pair of vertices u and v bends*. The parameter *bend* decides the extent (or the level) to which shortest paths curve towards the center. Note that in some cases $bend(u,v)$ will be assigned either $ecc(u)$ or $ecc(v)$, whatever is smaller. For example, see $\rho(u,v)$ in Figure 3.

Now we investigate the effect of the distance between a vertex pair on its bend. Our findings in this context are summarized in the following statements.

(A) Despite their distances, most vertex pairs bend. Moreover, among those bending pairs, the majority are sufficiently far from each other.

(B) There is a direct relation between the distance among vertex pairs and how close to the center a shortest path between them bends.

Motivation and Empirical Evaluation of (A). In light of Proposition 2, we investigate how vertex pairs of various distances act with respect to the eccentricity-based bending property. Interestingly, we noticed the bend in the majority of shortest paths. A quick look at Table 2 shows that a big percent of vertex pairs of distance at least two bend.

To quantify the distances at which the bend happens, we define two parameters: the *absolute curvity* and the *effective curvity*. Let k be the distance between a pair of vertices ($2 \leq k \leq diam(G)$), the *absolute curvity* k^* is the minimum k such that all pairs with distance $\geq k$ bend. The *effective curvity* $\tilde{k}$ is the minimum k such that more than 90% of the pairs with distance $\geq k$ bend. When the values of k^* and $\tilde{k}$ of each graph are represented as a function of δ to compare it with the upper bound $4\delta + 1$, we find that the networks have their k^* either equal to $2\delta + 1$ or to 2δ, and $\delta - 2 \leq \tilde{k} \leq 2\delta$. Also, all networks (except for MACAQUE-BRAIN-1) have their $\tilde{k}$ less than their k^*.

Motivation and Empirical Evaluation of (B). Here we examine the impact of the distance on the level to which vertex pairs bend. Let k be the distance between two vertices such that $2 \leq k \leq diam(G)$. Consider μ_k as the lowest layer that all vertex pairs of distance $\geq k$ bend to. We define it as: $\mu_k = \max\{bend(u,v) : \forall u, v \in V \text{ with } d(u,v) \geq k\}$. This allows us to look at how the bends of the vertex pairs behave with respect to different distances (see Figure 4). As expected, we found a direct relation between the distance of vertex pairs and their bend. For example, in network YEAST-PI, vertex pairs with distances 3, 6, and 9 bend to layers 4, 3, and 2 respectively.

5.2 Core-Periphery Identification Using the Eccentricity-Based Bending Property

A well-defined center of a graph is a good starting point for locating its core. According to the pattern of data exchange discussed earlier, we identify the core using the eccentricity centrality measure. Even though the center contains all vertices that are closer to other vertices, this subset is not sufficient. More vertices should be added to the core according to their participation in routing

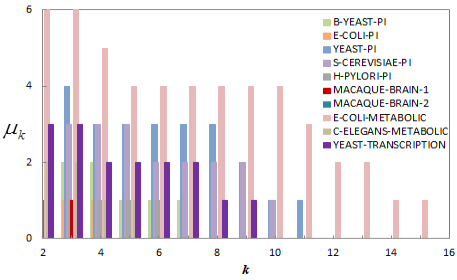


Fig. 4. μ_k values for each network in the graph datasets

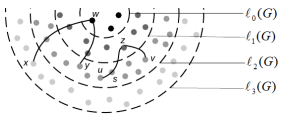


Fig. 5. Illustration of the eccentricity layering of a graph and the Maximum-Peak model. $\ell_r(G)$ represents each layer r. The peaks of $\rho(x, y)$ and $\rho(u, v)$ are w and z.

the traffic. We decide the participation of each vertex based on its eccentricity and whether or not it lies on a shortest path between a vertex pair.

Graphs follow the core-periphery structure with different extents with respect to the quality of their cores. We identify a good graph's core as the one that (1) includes a small number of layers with respect to the eccentricity layering; and (2) has a size (with respect to the number of vertices) that is small compared to the total number of vertices in the graph. The core should also contain vertices that participate in the majority of interactions among other vertices. In the following subsections, we discuss two core-periphery separation models.

Model I: The Maximum-Peak Model. Given a δ-hyperbolic graph $G = (V, E)$ along with its eccentricity layering $\mathcal{EL}(G)$, the Maximum-Peak model identifies a separation layer index $p \geq 0$ and defines the core as the subset of vertices formed by layers $\ell_0(G), \ell_1(G), ..., \ell_p(G)$.

In light of the eccentricity-based bending property, each $bend(x, y)$ for a pair of vertices x and y represents a *peak* for $\rho(x, y)$. In this model, we are locating the index of the lowest layer p over all layers that vertex pairs bend to. Index p represents the separation point where the layers can be partitioned to a core and a periphery. See Figure 5 for an illustration. After identifying all peaks, the core will include all vertices starting at $\ell_0(G)$ until $\ell_p(G)$, i.e., $core(G) = \bigcup_{r=0}^{p} \ell_r(G)$. Then the periphery will include the vertices in the remaining layers.

Again, to avoid the impact that outlier vertices may impose, we define two types of p. The *absolute separation index* p^* is the lowest layer that all vertex pairs bend to; we call the core defined by this index the absolute core set C^*_{core}. The effective separation index $\tilde{p}$ is the lowest layer where 90% of the vertex pairs bend to, and the core defined by this index is the effective core set $\tilde{C}_{core}$. Table 3 shows the cores for the networks in our datasets according to this model.

Table 3 shows a big difference in the sizes of the absolute core and the effective core in the majority of the networks. Closer analysis to $\tilde{C}_{core}$ suggests that deciding the core according to this notion generates good cores (number of layers in the core is small and the number of vertices is about 25% of the total number of vertices) for some networks such as the YEAST-PI. Also, networks with core sizes between 25% - 50% can be considered good as well; such as the core of

the S-CEREVISIAE-PI. On the other hand, networks like E-COLI-PI have too large core sizes compared to the overall graph size. This model is highly affected by the distribution of vertices over the layers. For example, the core of graph B-YEAST-PI has two layers (out of four). This can be considered as a balanced core-periphery separation. However, considering the distribution of the vertices in the four layers, which is 90, 902, 465, and 17, explains the increase in the size of the core. This issue can be resolved by using the second model.

Table 3. The cores of the graph datasets based on the Maximum-Peak model. $|V|$ is the number of vertices; $|Layers|$ is the number of layers; C^*_{core}-lyr and $|C^*_{core}|$ are the number of layers and number of vertices in the absolute core set; $\tilde{C}_{core}$-lyr and $|\tilde{C}_{core}|$ are the number of layers and number of vertices in the effective core set.

Network	$\|V\|$	$\|Layers\|$	C^*_{core}-lyr	$\|C^*_{core}\|$	$\|C^*_{core}\|$ to $\|V\|$	$\tilde{C}_{core}$-lyr	$\|\tilde{C}_{core}\|$	$\|\tilde{C}_{core}\|$ to $\|V\|$
B-YEAST-PI	1465	4	3	1448	$\approx 99\%$	2	902	$\approx 62\%$
E-COLI-PI	126	3	2	93	$\approx 74\%$	2	93	$\approx 74\%$
YEAST-PI	1728	6	5	1725	$\approx 100\%$	2	472	$\approx 27\%$
S-CEREVISIAE-PI	537	5	5	537	100%	2	223	$\approx 42\%$
H-PYLORI-PI	72	3	2	56	$\approx 78\%$	2	56	$\approx 78\%$
MACAQUE-BRAIN-1	45	3	2	31	$\approx 69\%$	2	31	$\approx 69\%$
MACAQUE-BRAIN-2	350	2	2	350	100%	2	350	100%
E-COLI-METABOLIC	242	8	7	240	$\approx 99\%$	3	102	$\approx 42\%$
C-ELEGANS-METABOLIC	453	4	3	439	$\approx 97\%$	1	17	$\approx 4\%$
YEAST-TRANSCRIPTION	321	5	4	314	$\approx 98\%$	2	62	$\approx 19\%$

Model II: The Minimum Cover Set Model. Consider a graph $G = (V, E)$ with the eccentricity layering $\mathcal{EL}(G)$ and with the center $C(G)$. The way this model works is to start the core as an empty set and expand it to include vertices which have smaller eccentricity, are closer to the center, and participate in the traffic. This expansion should be orderly, first incorporating the vertices that have higher priority, and then vertices who are less eligible. For each vertex $v \in V$, we define three parameters according to which we prioritize the vertices.

- The *eccentricity* $ecc(v)$. Vertices with smaller eccentricities have higher priority to be in the graph's core.
- The *distance-to-center*, denoted as $f(v)$, which expresses the distance between v and its closest vertex from the center $C(G)$, i.e., $f(v) = d(v, C(G))$. Vertices with small $f(v)$ have higher priority of being in the core. For example, in Figure 5, vertex y is closer to the center than u.
- The *betweenness* $b(v)$. The betweenness measures how many pairs of distant vertices x and y have v in one of their shortest paths (versus counting all shortest paths in the classic definition of the betweennes). It quantifies the participation of a vertex v in the traffic flow process, and we define it as: $b(v) =$ number of pairs $x, y \in V$ with $v \neq x, v \neq y, d(x, y) \geq 2$ and $d(x, v)+d(v, y) = d(x, y)$. According to the core-periphery organization, the betweenness of a vertex should increase as its eccentricity decreases.

Our goal in this model is to identify the smallest subset of vertices that participate in all traffic throughout the network. The algorithm for this model comprises two stages. First, in a priority list T we lexicographically sort the vertices according to the three attributes: $ecc(v)$, $f(v)$, and $b(v)$. T now has the vertices in the order that they should be considered to become part of the core. The goal is to ensure that there exists at least one vertex $v \in core(G)$ such that $v \in \rho(x, y)$ for each pair of vertices $x, y \in V$. In such case, we say that a shortest path $\rho(x, y)$ is covered by v (a shortest path from y to x is also covered by v since we are dealing with undirected graphs).

The second stage starts with a vertex v at the head of T being removed from T and added to an initially empty set C^*_{core} that represents the absolute core set. This vertex must cover at least one pair. After this initial step, the process continues by repeatedly removing the vertex v at the head of T and adding it to C^*_{core} if and only if v covers an uncovered yet pair x and y (when there is at least one vertex $v \in C^*_{core}$ that covers a pair (x, y), then it becomes covered). This step should run until all pairs are covered. Note that we consider the core set C^*_{core} as absolute since all vertex pairs must be covered by a vertex in it. Now the vertices in set C^*_{core} represent the core of the graph while the remaining vertices represent the periphery. The number of vertices in the absolute and the effective core sets of each graph in our datasets is listed in Table 4.

Table 4. The cores of the graph datasets based on the Minimum Cover Set model. $|V|$ is the number of vertices; $\delta(G)$ is the hyperbolicity; $|C^*_{core}|$ is the number of vertices in the absolute core set; $|\tilde{C}_{core}|$ is the number of vertices in the effective core set; C^*_{MaxLyr} is the largest index layer found among vertices in C^*_{core}; and $\tilde{C}_{MaxLyr}$ is the largest index layer found among vertices in $\tilde{C}_{core}$.

Network	$\|V\|$	$\delta(G)$	$\|C^*_{core}\|$	$\|C^*_{core}\|$ to $\|V\|$	C^*_{MaxLyr}	$\|\tilde{C}_{core}\|$	$\|\tilde{C}_{core}\|$ to $\|V\|$	$\tilde{C}_{MaxLyr}$
B-YEAST-PI	1465	2.5	1117	$\approx$ 76 %	3	117	$\approx$ 8 %	1
E-COLI-PI	126	2	65	$\approx$ 52 %	2	13	$\approx$ 10 %	1
YEAST-PI	1728	3.5	902	$\approx$ 52 %	5	318	$\approx$ 18 %	2
S-CEREVISIAE-PI	537	4	438	$\approx$ 82 %	4	114	$\approx$ 21 %	1
H-PYLORI-PI	72	3	54	$\approx$ 75 %	2	15	$\approx$ 21 %	1
MACAQUE-BRAIN-1	45	1.5	20	$\approx$ 44 %	2	7	$\approx$ 16 %	1
MACAQUE-BRAIN-2	350	1.5	197	$\approx$ 56 %	1	31	$\approx$ 9 %	0
E-COLI-METABOLIC	242	4	208	$\approx$ 86 %	7	66	$\approx$ 27 %	2
C-ELEGANS-METABOLIC	453	1.5	202	$\approx$ 45 %	2	12	$\approx$ 3%	0
YEAST-TRANSCRIPTION	321	3	155	$\approx$ 48 %	4	40	$\approx$ 12%	1

Close analysis of Table 4 shows that each produced absolute core C^*_{core} is of a size between 44% to 86% of the original number of vertices in the graph. It is important to note that vertices in the core are expected to have different contributions (some vertices cover more vertex pairs than others). Figure 6 shows how many vertex pairs are remained uncovered after the orderly addition of vertices to the absolute core. For example, in the network B-YEAST-PI, 80% of vertex pairs are uncovered after adding the first vertex to C^*_{core}. However, after adding 20 vertices, only 35% of the vertex pairs are uncovered. It is also clear

that many of the vertices that have been added later to the absolute core set cover a very small percentage of vertex pairs.

To keep only vertices that are considered higher contributors we define the effective core set $\tilde{C}_{core}$. The effective core is the subset of the core that is sufficient to cover 90% of the vertex pairs in the graph. To obtain $\tilde{C}_{core}$, we examine the vertices of the core C^*_{core} in the same order in which they were added. A new vertex is added to current $\tilde{C}_{core}$ only if more than 10% of the vertex pairs remain uncovered. The results on the core according to both concepts in this model are presented in Table 4. Note that the index of the layer of the last vertex added to the core in each network has significantly decreased.

Because hyperbolic graphs adhere to the property of having shortest paths that bend to the core, it was natural to think that hyperbolic graphs with lower $\delta(G)$ should have even smaller cores. A quick comparison between the $\tilde{C}_{core}$ of each graph with its $\delta(G)$ supports this idea.

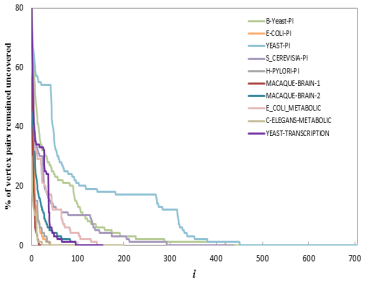


Fig. 6. The percentage of the uncovered vertex pairs after the orderly addition of vertices to the core set C^*_{core}. Number i indicates the cardinality of the current core.

Table 5. Summary of the graph datasets' parameters and cores. $\tilde{C}_{core}$ is the effective core according to the Minimum Cover Set model.

		Network	$\log_2(size)$	$diam$	δ	δ'	$\tilde{C}_{core}$
Strongly-hyperbolic Networks	1	C-ELEGANS-METAB.	12.3	7	1.5	0.133	3%
		B-YEAST-PI	12.8	8	2.5	0.299	8%
	2	MACAQUE-BRAIN-2	12.4	4	1.5	0.203	9%
		E-COLI-PI	9.5	5	2	0.251	10%
		YEAST-TRANSCR.	10	9	3	0.365	12%
		MACAQUE-BRAIN-1	9	4	1.5	0.231	16%
		YEAST-PI	13.6	12	3.5	0.322	18%
Hyperbolic Networks		S-CEREVISIAE-PI	10.5	11	4	0.419	21%
		H-PYLORI-PI	7.5	7	3	0.368	21%
		E-COLI-METAB.	9.3	16	4	0.483	27%

6 Concluding Remarks

The structure of several biological networks has been often described as a tree-like topology in molecular biology [4]. This motivates investigating if those networks also admit tree-like structures based on their distances. In Section 4, we observed that most biological networks appear to have low hyperbolicity. Since strongly-hyperbolic graphs have a structure that is closer to a tree, this motivates the following hypothesis: do strongly-hyperbolic graphs have more concise cores compared to other hyperbolic graphs? It is clear from Tables 5 that hyperbolic networks have larger cores when compared to strongly-hyperbolic networks (which confirms our hypothesis). Here we only consider cores according to the Minimum Cover Set model. The sizes of the cores in strongly-hyperbolic graphs are less than 20% of the number of vertices of each network.

We also observed two patterns in strongly-hyperbolic networks named groups 1 and 2 in Table 5. The networks in group 1 have $\delta(G) < 3$ and in the same time $\delta(G)$ is sufficiently smaller than the value of half the diameter. The cores for those networks are very small. The second group has networks that are either with higher hyperbolicity, or low hyperbolicity with value of $\delta(G)$ very close to $diam(G)/2$. The cores for group 2 are larger than those in group 1.

References

[1] http://www.kent.edu/~dragan/FullVersionComplexNet.pdf
[2] Abu-Ata, M., Dragan, F.F.: Metric tree-like structures in real-life networks: an empirical study. arXiv preprint 3364(1402) (2014)
[3] Adcock, A., Sullivan, B., Mahoney, M.: Tree-like structure in large social and information networks. In: ICDM 2013 (2013)
[4] Albert, R., DasGupta, B., Mobasheri, N.: Topological implications of negative curvature for biological and social networks. Physical Review E 89(3) (2014)
[5] Baryshnikov, Y., Tucci, G.: Asymptotic traffic flow in a hyperbolic network. In: ISCCSP (2012)
[6] Borgatti, S., Everett, M.: Models of core/periphery structures. Social Networks 21(4), 375–395 (2000)
[7] Bu, D., Zhao, Y., et al.: Topological structure analysis of the protein-protein interaction network in budding yeast. Nucleic Acids Research 31(9), 2443–2450 (2003)
[8] Butland, G., Manuel, J., et al.: Interaction network containing conserved and essential protein complexes in escherichia coli. Nature 7025(433), 531–537 (2005)
[9] Chepoi, V., Dragan, F., et al.: Diameters, centers, and approximating trees of delta-hyperbolic geodesic spaces and graphs. In: SoCG 2008 (2008)
[10] Chepoi, V., Dragan, F., et al.: Additive spanners and distance and routing labeling schemes for hyperbolic graphs. Algorithmica 62(3), 713–732 (2012)
[11] Christian, V.M., Krause, R., et al.: Comparative assessment of large-scale data sets of protein-protein interactions. Nature 6887(417), 309–403 (2002)
[12] Cvetkovski, A., Crovella, M.: Hyperbolic embedding and routing for dynamic graphs. In: INFOCOM 2009 (2009)
[13] Da, S., Rosa, M., et al.: Centrality, network capacity, and modularity as parameters to analyze the core-periphery structure in metabolic networks. Proceedings of the IEEE 96, 1411–1420 (2008)
[14] de Montgolfier, F., Soto, M., Viennot, L.: Treewidth and hyperbolicity of the internet. In: 10th IEEE International Symposium, NCA, pp. 25–32. IEEE (2011)
[15] Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. Physical Review 72(2) (2005)
[16] Fell, D., Wagner, A.: The small world of metabolism. Nature Biotechnology 18(11), 1121–1122 (2000)
[17] Gromov, M.: Hyperbolic Groups. Springer, New York (1987)
[18] Holme, P.: Core-periphery organization of complex networks. Physical Review E 72(4) (2005)
[19] Jeong, H., Mason, S., et al.: Lethality and centrality in protein networks. Nature 6833(411), 41–42 (2001)
[20] Ma, H., Zeng, A.-P.: Reconstruction of metabolic networks from genome data and analysis of their global structure for various organisms. Bioinformatics 19(2), 270–277 (2003)

[21] Ma, H.-W., Zeng, A.-P.: The connectivity structure, giant strong component and centrality of metabolic networks. Bioinformatics 19(11), 1423–1430 (2003)
[22] Milo, R., Shen-Orr, S., et al.: Network motifs: simple building blocks of complex networks. Science 5594(298), 824–827 (2002)
[23] Modha, D., Singh, R.: Network architecture of the long-distance pathways in the macaque brain. Proceedings of NAS 107
[24] Narayan, O., Saniee, I.: The large scale curvature of networks. Physical Review E 84(6) (2011)
[25] Négyessy, L., Nepusz, T., et al.: Prediction of the main cortical areas and connections involved in the tactile function of the visual cortex by network analysis. European Journal of Neuroscience 23(7), 1919–1930 (2006)
[26] Rain, J.-C., Selig, L., et al.: The protein-protein interaction map of helicobacter pylori. Nature 6817(409), 211–215 (2001)
[27] Rodríguez, J., Sigarreta, J., et al.: On the hyperbolicity constant in graphs. Discrete Mathematics 311(4) (2011)
[28] Shavitt, Y., Tankel, T.: On the curvature of the internet and its usage for overlay construction and distance estimation. In: INFOCOM 2004. IEEE (2004)

Fast Optimization of Hamiltonian for Constrained Community Detection

Keisuke Nakata and Tsuyoshi Murata

Department of Computer Science, Graduate School of Information Science and Engineering
Tokyo Institute of Technology
W8-59 2-12-1 Ookayama Meguro Tokyo 152-8552 Japan

Abstract. Various methods for analyzing networks have been proposed. Among them, methods for community detection based on network structures are important for making networks simple and easy to understand. As an attempt to incorporate background knowledge of given networks, a method known as constrained community detection has been proposed recently. Constrained community detection shows robust performance on noisy data since it uses background knowledge. In particular, methods for community detection based on constrained Hamiltonian have advantages of flexibility in output results. In this paper, we propose a method for accelerating the speed of constrained community detection based on Hamiltonian. Our optimization method is a variant of Blondel's Louvain method which is well-known for its computational efficiency. Our experiments showed that our proposed method is superior in terms of computational time, and its accuracy is almost equal to the existing method based on simulated annealing under the same conditions. Our proposed method enables us to perform constrained community detection in larger networks compared with existing methods. Moreover, we compared the strategies of adding constraints incrementally in the process of constrained community detection.

1 Introduction

There are emerging needs for understanding the structures of huge data due to the growing advancement of information technologies. Many of them can be represented as networks, such as friendship networks of social media or hyperlink networks of Web pages. Several attempts have been made for community detection [POM09][For10]; extracting dense subnetworks from given networks. Community detection is important for analyzing and visualizing given networks from mesoscopic viewpoints.

One of the most popular metrics for community detection is modularity [NG04]. It is often used for evaluating the qualities of detected communities compared with the null model. Many community detection methods optimize modularity in order to search for partitions of given networks [CNM04][For10][PKVS12]. As the method for optimizing modularity of large-scale networks, Louvain method [BGLL08] is often employed.

One of the promising directions of community detection is to incorporate constraints on communities to be detected, which is called constrained community detection. In many cases, humans already have some background knowledge on the structure of given networks. Such knowledge should be incorporated in the process of community detection in order to find better communities.

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_8

Among the approaches of constrained community detection, Reichardt and Bornholdt [RB06] introduced Hamiltonian as a generalization of modularity. Eaton et al. proposed a method for optimizing constrained Hamiltonian [EM12]. Although the method is theoretically good, it is slow since it employs simulated annealing [KJV83] for optimizing constrained Hamiltonian.

This paper extends Louvain method, and proposes a method for fast optimization of constrained Hamiltonian. It is often said that there is a tradeoff between accuracy and speed, but our optimization method satisfies both. It is effective not only for processing large-scale networks but also for performing interactive community detection since users often put some additional constraints after they watched the results of obtained communities. There are many strategies for giving constraints incrementally in the process of community detection, hence we performed experiments comparing some of them.

2 Related Works

This section introduces some basic metrics and notations that are necessary for explaining our proposed method.

2.1 Modularity

Modularity introduced by Newman and Girvan [NG04] is one of the most popular metrics for evaluating the quality of communities extracted from a given network. The metric is computed from the difference between the number of actual edges within communities in a network and the expected value of its null model. Null model of a network is generated by rewiring edges of the network while degrees of all vertices are kept the same as those of the original network. Modularity shows the amount of deviation of the number of edges within communities from random partitions. Therefore, partitions of high modularity are regarded as good from the viewpoint of community detection. The value of modularity Q is defined as follows:

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - P_{ij}\right) \delta\left(C_i, C_j\right), \tag{1}$$

where i and j are indices of nodes, A is an adjacency matrix of the network, $P_{ij} = (k_i k_j)/2m$ is a null model of the network, k_i is the degree of node i, $m = \sum_i k_i/2$ is the number of edges in the network, C_i is the index of the community which node i belongs to, and δ is the Kronecker's delta. In order to detect communities, partitions of high Q values are searched, and it is often called modularity optimization.

3 Generalization of Modularity

Hamiltonian $\mathscr{H}$ [RB06], which is a generalization of modularity (expression (1)), is expressed as follows:

$$\begin{aligned}\mathscr{H} = &-\sum_{i,j} a_{ij}A_{ij}\delta(C_i,C_j) \\ &+\sum_{i,j} b_{ij}(1-A_{ij})\delta(C_i,C_j) \\ &+\sum_{i,j} c_{ij}A_{ij}(1-\delta(C_i,C_j)) \\ &-\sum_{i,j} d_{ij}(1-A_{ij})(1-\delta(C_i,C_j)).\end{aligned} \tag{2}$$

We have to keep in mind that in contrast to modularity, smaller Hamiltonian value means better network partition. In expression (2), Hamiltonian (a) rewards intra-community edges (the first term), (b) penalizes the lack of intra-community edges (the second term), (c) penalizes inter-community edges (the third term), and (d) rewards the lack of inter-community edges (the fourth term), and each is weighted by parameters a, b, c and d, respectively.

If the parameters are set appropriately ($a_{ij} = c_{ij} = 1-\gamma P_{ij}$, $b_{ij} = d_{ij} = \gamma P_{ij}$), expression (2) can be transformed as follows:

$$\mathscr{H} = -2\sum_{i,j}(A_{ij} - \gamma P_{ij})\delta(C_i,C_j) + 2m(1-\gamma). \tag{3}$$

The second term on the right side, $2m(1-\gamma)$, can be ignored because it is independent of the result of community detection. Then the expression is equal to the definition of modularity (expression (1)) times constant value. This means that Hamiltonian is a generalization of modularity.

3.1 Constrained community detection

As a method for performing constrained community detection, Eaton et al. [EM12] proposed an optimization for constrained Hamiltonian, in which a constrained term is added to the above-mentioned Hamiltonian (expression (3)). Constrained term U is composed of (a) u_{ij} which means that a pair of nodes should be in the same community, and (b) $\overline{u}_{ij}$ which means that a pair of nodes should be in different communities:

$$U = \sum_{i,j}\left(u_{ij}\left(1-\delta\left(C_i,C_j\right)\right) + \overline{u}_{ij}\delta\left(C_i,C_j\right)\right). \tag{4}$$

Settings for the values of u_{ij} and $\overline{u}_{ij}$ are discussed in section 5. Constrained Hamiltonian $\mathscr{H}'$ is expressed as follows:

$$\begin{aligned}\mathscr{H}' &= \mathscr{H} + \mu U \\ &= -2\sum_{i,j}\left(\left(A_{ij} - \gamma P_{ij} + \mu\Delta U_{ij}\right)\delta\left(C_i,C_j\right)\right) + K,\end{aligned} \tag{5}$$

where μ is a parameter for balancing Hamiltonian $\mathscr{H}$ and constrained term U, $\Delta U_{ij} = (u_{ij} - \overline{u}_{ij})/2$, $K = 2m(1-\gamma) + \mu\sum_{i,j} u_{ij}$, respectively. K is a constant independent from extracted communities.

Eaton et al. employed simulated annealing [KJV83] in order to optimize expression (5). They claimed that noise-tolerant and accurate constrained community detection is achieved [EM12].

3.2 Louvain Method

Louvain method [BGLL08] is a method known for its fast optimization of modularity. Although Louvain method is a straightforward greedy method, it experimentally showed high accuracy. Louvain method consists of the following two phases:

1. Each node is moved to one of its adjacent communities, and the gain of modularity value after the move is computed. The move that will increase modularity the most will be employed and the node is assigned to the new community, but only if the gain is positive. This process is repeated for every node until no more increase of modularity can be obtained.
2. Each community obtained in step 1 is aggregated to a node, and a new network of aggregated nodes is generated.

The above two phases are repeated iteratively until convergence. In phase 1, only the difference of modularity before and after the move (ΔQ) is computed in order to speedup the computation. When node x is moved from community Y to community Z, the difference of modularity value ΔQ is as follows:

$$\Delta Q = \frac{1}{m}\left(\sum_{i\in Z}(A_{ix}-P_{ix}) - \sum_{i\in Y}(A_{ix}-P_{ix})\right), \tag{6}$$

where k_i in $P_{ij} = (k_i k_j)/2m$ is the sum of weights of all edges that are connected to node i.

In phase 2, each community obtained in phase 1 is regarded as a node and a new network of the nodes is generated. The weight of an edge that connect two nodes in the new network is the sum of the weights of all edges that connect nodes between corresponding two communities before aggregation. The weight of self-loop edge in a new network is equal to the double of the sum of all edges within the community.

4 Fast Optimization of Hamiltonian for Constrained Community Detection

Eaton et al. claimed that optimization of constrained Hamiltonian is good for constrained community detection, although they used slow simulated annealing for the optimization. We extended Louvain method (which was originally for optimizing modularity) for the optimization of constrained Hamiltonian in order to speedup constrained community detection.

Our method for optimization is similar to Louvain method, except $\Delta\mathscr{H}'$ is computed in phase 1 in section 3.2 instead of ΔQ. The difference of constrained Hamiltonian $\mathscr{H}'$ before and after node x is moved from community Y to community Z ($\Delta\mathscr{H}'$) is

represented as follows; where C^y is the network partition before the move (when node x belongs to community Y), and C^z is the network partition after the move (when node x belongs to community Z):

$$\Delta\mathscr{H}' = \left(-2\sum_{i,j}\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{ij})\,\delta(C_i^z,C_j^z)\right)+K\right) - \left(-2\sum_{i,j}\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{ij})\,\delta(C_i^y,C_j^y)\right)+K\right). \tag{7}$$

Since the communities of other nodes except x is the same (if $i \neq x$ and $j \neq x$ then $\delta(C_i^z,C_j^z) = \delta(C_i^y,C_j^y)$), the following equation holds:

$$\begin{aligned}\frac{\Delta\mathscr{H}'}{2} = &-\sum_i\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{ix})\,\delta(C_i^z,C_x^z)\right)\\ &-\sum_j\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{xj})\,\delta(C_x^z,C_j^z)\right)\\ &+\sum_i\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{ix})\,\delta(C_i^y,C_x^y)\right)\\ &+\sum_j\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{xj})\,\delta(C_x^y,C_j^y)\right).\end{aligned} \tag{8}$$

Since A, P and ΔU are symmetric matrices[1], the following equation holds:

$$\begin{aligned}\frac{\Delta\mathscr{H}'}{2} = &-2\sum_i\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{ix})\,\delta(C_i^z,C_x^z)\right)\\ &+2\sum_i\left((A_{ij}-\gamma P_{ij}+\mu\Delta U_{ix})\,\delta(C_i^y,C_x^y)\right).\end{aligned} \tag{9}$$

If nodes i and x are in different communities, $\delta(C_i,C_x) = 0$. Otherwise, if they are in the same community, $\delta(C_i,C_x) = 1$. Therefore the following equation holds:

$$\Delta\mathscr{H}' = -4\left(\sum_{i\in Z}(A_{ij}-\gamma P_{ij}+\mu\Delta U_{ix}) - \sum_{i\in Y}(A_{ij}-\gamma P_{ij}+\mu\Delta U_{ix})\right). \tag{10}$$

Expression (10) is computed in our proposed method in order to perform constrained community detection. If the parameter μ is set to $\mu = 0$, the term ΔU is cancelled out in expressions (5) and (10), and our method is the same as the normal community detection without considering constraints. If the parameter μ is set to a large value, ΔU dominates the behavior of $\mathscr{H}'$, and the communities that only focus on constraints will be extracted.

Since computational cost of expression (10) is almost the same as that of expression (6), the efficiency of our proposed method for optimizing constrained Hamiltonian is expected to achieve the same level as Louvain method.

[1] In the case of an undirected network, A and P are always symmetric. Blondel's original Louvain method is basically for undirected networks.

Table 1. Networks used in our experiments

Network	#nodes	#edges	#communities
Karate [Zac77]	34	78	2
Polbooks [Kre]	105	441	3
Polblogs [AG05]	1,222	16,714	2

5 Experiments

Table 1 shows the networks that were used for our experiments. Correct communities are known in advance as the ground-truth labels for each of them. Parameters are set as follows: $\mu = 2$, $\gamma = 1$, and $P_{ij} = k_i k_j / 2m$.

We focus on the constraints of assigning a positive integer l_i (as community label) to node i. A label of an unconstrained node is assigned as $l_i = -1$. Values of u_{ij} and $\overline{u}_{ij}$ are set as follows:

$$u_{ij} = \begin{cases} 1 & (\text{when } l_i = l_j \neq -1), \\ 0 & (\text{otherwise}), \end{cases} \tag{11}$$

$$\overline{u}_{ij} = \begin{cases} 1 & (\text{when } l_i \neq l_j \wedge l_i \neq -1 \wedge l_j \neq -1), \\ 0 & (\text{otherwise}). \end{cases} \tag{12}$$

As a metric for measuring the similarity between extracted communities C and correct communities C', normalized mutual information (NMI) [SG03] is used:

$$\mathrm{NMI}(C, C') = \frac{\sum_c \sum_{c'} n_{cc'} \log \frac{n_{cc'} \cdot n}{n_c \cdot n_{c'}}}{\sqrt{\left(\sum_c n_c \log \frac{n_c}{n} \right) \left(\sum_{c'} n_{c'} \log \frac{n_{c'}}{n} \right)}}, \tag{13}$$

where c and c' are indices of communities C and C', n is the number of nodes, $n_{cc'}$ is the number of nodes that belong to both c and c', and n_c and $n_{c'}$ are the number of nodes that belong to c and c', respectively. The more C and C' are similar, the larger their NMI is. C is set to the extracted communities and C' is set to the correct communities in order to measure the accuracy of community detection.

5.1 Comparison of Our Proposed Method, Simulated Annealing Method and Louvain Method

We can consider two cases for constrained community detection: (1) all constraints are given in advance, and (2) constraints are given incrementally. This subsection discusses the former case for comparing our proposed method, simulated annealing method, and Louvain method.

Figure 1 shows comparisons of accuracy using Karate network, Polbooks network and Polblogs network. X axis is the ratio of randomly added/deleted edges (as noise)

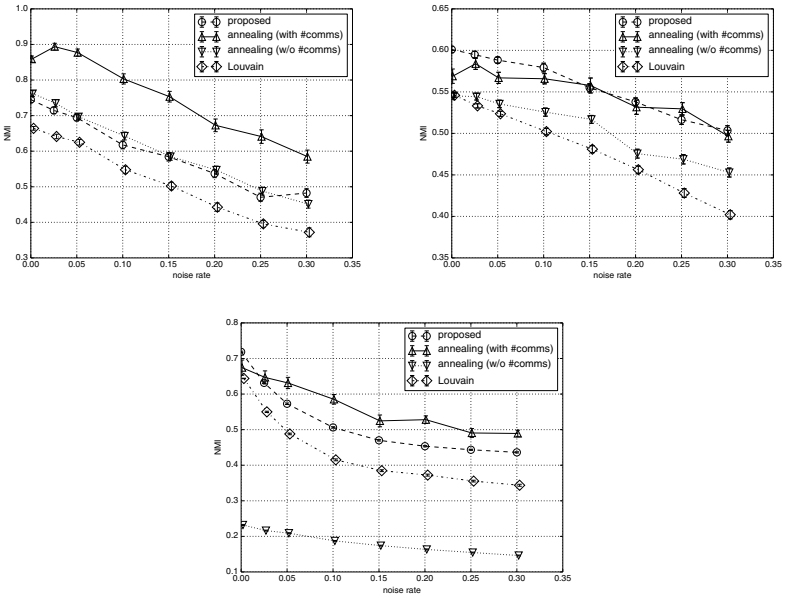


Fig. 1. Accuracies of our proposed method, simulated annealing method and Louvain method using Karate network (top left), Polbooks network (top right), and Polblogs network (bottom).

with keeping the degree distributions, and Y axis is NMI. In our proposed method and simulated annealing method, 20% of nodes are randomly selected and their ground-truth labels are given as constraints. Error bars show standard errors.

As Figure 1 shows, our proposed method is almost as accurate as simulated annealing method, if the number of communities is not given. In [EM12], the number the of ground-truth communities is given to simulated annealing method (triangular solid line in Figure 1). We also performed experiments with simulated annealing method without giving the number of communities (reversed-triangular dotted line in Figure 1), in order to compare it with our proposed method in the same condition. It was already pointed out that Louvain method is effective for optimizing modularity compared with other optimization methods [BGLL08], which is consistent with this result.

Figure 2 shows the comparisons of computational times of three methods. X axis is the same as Figure 1, and Y axis is the computational time (seconds). This showed that our proposed method is significantly faster than simulated annealing.

These results showed that our proposed method is almost as accurate as simulated annealing, and is much faster. This enables us to process large-scale networks.

5.2 Experiments on Large-Scale Networks

Table 2 shows the large-scale networks which we experimented with. Because there was no ground-truth label for them, it is impossible to give constraints from ground-truth labels or to measure the accuracy with NMI. However we tried to detect communities

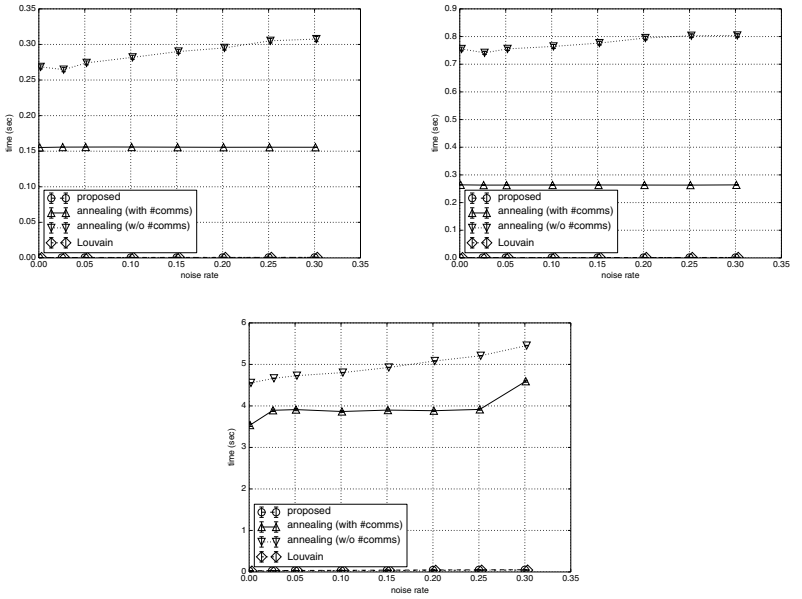


Fig. 2. Computational times of our proposed method, simulated annealing method and Louvain method using Karate network (top left), Polbooks network (top right) and Polblogs network (bottom)

from them with our proposed method and simulated annealing method without giving constraints in order to check the computational costs of them.

The results are shown in Table 3. It implies that our proposed method is very fast on large-scale networks.

5.3 Incremental Constrained Community Detection

This section discusses how to give constraints incrementally during the optimization of constrained Hamiltonian. Suppose there are no constraints at the initial stage, and constraints are given one by one and then constrained community detection is performed based on the constraints given so far. Since giving too many constraints manually is

Table 2. Large-scale networks used in our experiments

Network	#nodes	#edges	#communities
Power [WS98]	4,941	6,594	unknown
Dblp [YL12]	317,080	1,049,866	unknown[2]

[2] Dblp network has the overlapping and nested ground-truth communities, but that is not suitable because we assume that each node must belong to exactly one community.

Table 3. Computational times (second) for large-scale networks of our proposed method and simulated annealing method

	Annealing	Proposed
Power	3.016	0.056
Dblp	143.111	12.720

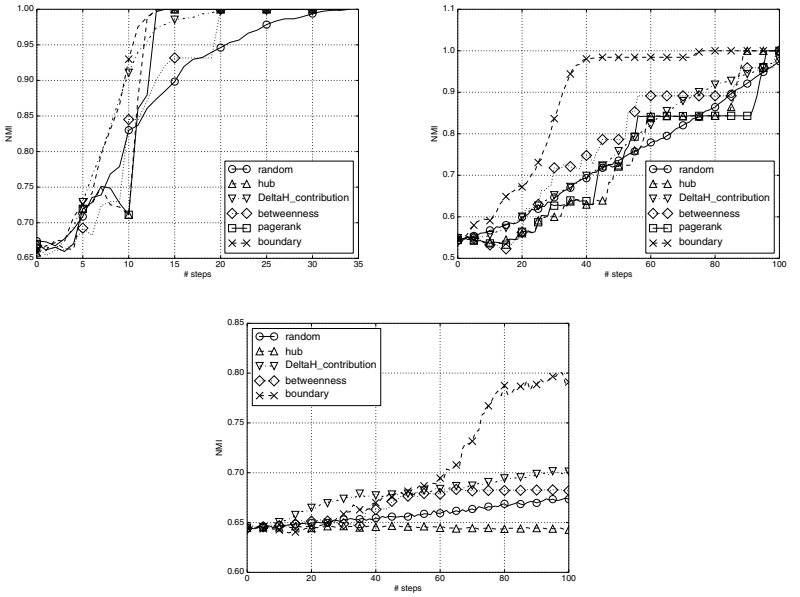


Fig. 3. Incremental addition of constraints and corresponding NMI using Karate network (top left), Polbooks network (top right) and Polblogs network (bottom)

unrealistic, we have to think about the strategies for selecting nodes that should be constrained.

Figure 3 shows the results of incremental addition of constraints and corresponding NMI values after constrained community detection was performed with our proposed method. X axis is the number of constraints, and Y axis is NMI. Lines in the Figure correspond to the following strategies for giving constraints:

random: Nodes are selected randomly.
hub: Nodes are selected in descending order of their degrees.
DeltaH_contribution: Nodes are selected in descending order of expression (10).
betweenness: Nodes are selected in descending order of betweenness.
pagerank: Nodes are selected in descending order of PageRank[PBMW99].
boundary: Nodes adjacent to different communities are selected.

The top left of Figure 3 shows that the performances of DeltaH_contribution and boundary are good when the number of constraints are less than ten. Among them,

boundary strategy is the best since it quickly reaches the highest NMI value. The top right and bottom of Figure 3 also shows that boundary is the best strategy. The results show that the order of adding constraints matters for an accurate constrained community detection. Based on the above results, we can conclude that the boundary strategy is the best in our list of surveyed strategies. This strategy gives constraints to the nodes that are located at the boundaries of different communities. It makes sense because giving constraints to such marginal nodes is expected to enhance the accuracies of community detection.

As for the strategy for adding constraints to nodes, the uncertain sampling [LG94] is often employed. The strategy is to select nodes whose degree of "wrongness" are the biggest. It has been pointed out that humans' strategies are often superior to uncertain sampling. This means that the performance of humans' interactive constrained community detection is expected to be better than the results shown in Figure 3.

6 Conclusion

This paper extends Louvain method for optimizing constrained Hamiltonian. Our proposed method is much faster than the existing simulated annealing method, without any compromise in accuracy. In addition, we performed some experiments on incremental constrained community detection and compare the strategies for giving constraints.

The followings are left for our future work.

Firstly, appropriate values of parameters such as γ, μ, u, $\overline{u}$ should be discussed further. We have used the same values that are used in Eaton's paper [EM12]. But theoretical and experimental optimization for these parameters have yet to be solved. μ controls the strength of overall constrained term, and u, $\overline{u}$ controls the strength of each constraint. Hence u, $\overline{u}$ can be set to the degree of user's confidence on each constraint. Another direction of this research is to set the weight of each constraint automatically.

Secondly, good strategies for giving constraints should be discussed further. It might be good to observe and imitate humans' heuristic strategies for accurate constrained community detection. The final goal of our research is to develop an environment of network analysis that would allow an interactive feedback from users, and this would give more insights into the performance of interactive community detection.

References

AG05. Adamic, L.A., Glance, N.: The political blogosphere and the 2004 u.s. election: Divided they blog. In: Proceedings of the 3rd International Workshop on Link Discovery, LinkKDD 2005, pp. 36–43. ACM, New York (2005)

BGLL08. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (2008)

CNM04. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. Phys. Rev. E 70, 066111 (2004)

EM12. Eaton, E., Mansbach, R.: A spin-glass model for semi-supervised community detection. In: Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI 2012), July 22-26, pp. 900–906. AAAI Press (2012)

For10. Fortunato, S.: Community detection in graphs. Physics Reports 486(3-5), 75–174 (2010)

KJV83. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)

Kre. Krebs, V.: Books about us politics. Nodes represent books about US politics sold by the online bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers, as indicated by the "customers who bought this book also bought these other books" feature on Amazon

LG94. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994, pp. 3–12. Springer-Verlag New York, Inc., New York (1994)

NG04. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Phys. Rev. E 69, 26113 (2004)

PBMW99. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab (November 1999) Previous number = SIDL-WP-1999-0120

PKVS12. Papadopoulos, S., Kompatsiaris, Y., Vakali, A., Spyridonos, P.: Community detection in social media. Data Mining and Knowledge Discovery 24(3), 515–554 (2012)

POM09. Porter, M.A., Onnela, J.-P., Mucha, P.J.: Communities in networks. Notices of the AMS 56(9) (2009)

RB06. Reichardt, J., Bornholdt, S.: Statistical mechanics of community detection. Phys. Rev. E 74, 016110 (2006)

SG03. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research 3, 583–617 (2003)

WS98. Watts, D., Strogatz, S.: Collective dynamics of 'small-world' networks. Nature 393, 440–442 (1998)

YL12. Yang, J., Leskovec, J.: Defining and evaluating network communities based on ground-truth. CoRR, abs/1205.6233 (2012)

Zac77. Zachary, W.W.: An information flow model for conflict and fission in small groups. Journal of Anthropological Research 33, 452–473 (1977)

Selecting Seed Nodes for Influence Maximization in Dynamic Networks

Shogo Osawa and Tsuyoshi Murata

Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology, W8-59 2-12-1 Ookayama Meguro Tokyo 152-8552 Japan
{s_osawa,murata}@ai.cs.titech.ac.jp

Abstract. This paper proposes a method for solving influence maximization problem in a dynamic network. In our method, a node that increases its influence most will be searched and it is added to the seed nodes incrementally. Since exact computation of influence of a node is #P-Hard, we employ heuristics for approximate computation. The results of our experiments show that our method is more effective than the methods based on centralities for dynamic networks, especially when the networks exhibit community structures.

1 Introduction

Influence maximization problem is a problem of selecting the set of k nodes that is the most influential for propagating information (or diseases) to other nodes in a network. Solving this problem is important for minimizing disease propagation or maximizing the effect of advertisement in viral marketing. Since this problem is proved to be NP-Hard[KKT03], obtaining exact answer to the problem is intractable for large networks. Therefore, several methods such as Monte-Carlo simulation and heuristic-based methods have been proposed [CSH$^+$13] [CWW10] [JSC$^+$11] [JHC12]. These research are basically for static networks. Only few attempts have been made for influence maximization on dynamic networks whose edges are dynamically added or deleted.

Naive methods for solving influence maximization problem in dynamic networks are centrality-based methods, which select top k nodes of high centrality values. There are several definitions of centrality for dynamic networks, such as closeness centrality [HS12] and broadcast centrality [GPHE11]. One of the weaknesses of centrality-based methods is that nodes of high centrality might propagate information to adjacent nodes that overlap with each other.

Suppose we are going to select two nodes that are the most influential to the network shown in Figure 1. The number shown at the upper left of each node is its closeness centrality. For the sake of convenience, selected nodes will propagate information to all reachable nodes. Although two nodes of the largest closeness centralities in Figure 1 are nodes D and B, reachable nodes from them are exactly the same (A, B, C, D, and E). This means that selecting node B in addition to node D does not increase the power of influence of seed nodes. In this example, selecting node D and F will be a good choice because all other nodes in the network are reachable from these two. Therefore, just selecting nodes of high centrality values may not be a good method. This is also true in a dynamic network.

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_9

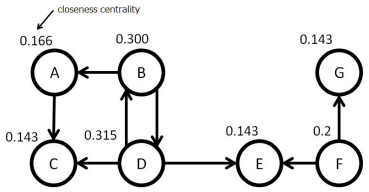


Fig. 1. An example of influence maximization

This paper proposes a method for solving influence maximization problem in dynamic networks. Our proposed method starts with an empty node set $\mathscr{S} = \emptyset$. Then a node n is added to $\mathscr{S}$ incrementally so that the influence of $\mathscr{S} \cup \{n\}$ will be maximized. Since the computation of exact influence is time consuming, the approximated power of influence of node set is computed. Experimental results show that our method is effective especially when a network exhibits community structures.

2 Related Works

2.1 Dynamic Networks

We focus on a network whose edges will appear or disappear dynamically and its nodes are static throughout its period. Such a dynamic network can be represented as a list of adjacency matrices: $G = (A_1, A_2, \ldots, A_T)$ where A_t is an adjacency matrix of a network at time t. T is the period of the dynamic network, and we assume that T is finite. An edge between node i and j at time t is represented as a triplet (t, i, j). A walk of length $k-1$ from node n_1 to node n_k is defined as a sequence of edges: $(t_1, n_1, n_2), (t_2, n_2, n_3), \ldots, (t_{k-1}, n_{k-1}, n_k)$, where $t_1 < t_2 < \cdots < t_{k-1}$ should be satisfied. A walk of no node revisit $(\forall i, j (i \neq j) \;\; n_i \neq n_j)$ is called as a path. The period of a path is the duration of time from the start to the end of the path, which is defined as $t_{k-1} - t_1 + 1$. A path of minimum period is the shortest path, and its period is the shortest period.

An aggregate network G_{agg} of a dynamic network $G = (A_1, A_2, \ldots, A_T)$ is a static network: $G_{agg} = \sum_{t=1}^{T} A_t$, in which times of all edges in G are ignored.

2.2 SI Model for Information Propagation

We focus on SI model [BZW07] as a model for information propagation. In SI model, state S (susceptible) or state I (infected) is assigned to each node. A node in state S does not have information, and a node in state I has information and is ready to propagate. At the initial stage of information propagation ($t = 1$), only seed nodes are assigned to state I and others are assigned to state S. At $t = 1, 2, \ldots, T$, information is propagated in the following steps:

1. For each edge (t, i, j) at time t, the following operation is done:
   a. If node i is in status I and if node j is in status S, then node j will be in status I with probability λ at time $t + 1$.

b. If the network is undirected, information is propagated to both directions. In other words, if node j is in status I and if node i is in status S, then node i will be in status I with probability λ at time $t+1$.

2. Information propagation is terminated at time $T+1$.

λ is a parameter for the ratio of infection. We assume that T is finite so the above steps will be terminated within finite time.

2.3 Formalization of an Influence Maximization Problem

For SI model, we define the power of influence of node set $\mathscr{S}$ as the expected number of nodes in status I at time $T+1$ when seed nodes are given as $\mathscr{S}$, and express it as $\sigma(\mathscr{S})$. Influence maximization problem is a problem of selecting the node set of size k that maximize $\sigma(\mathscr{S})$. In SIR model, which is a generalization of SI model, exact computation of σ for static networks is proved to be #P-Hard[PS12]. Based on this result, we can assume that exact computation of σ for dynamic networks is also #P-Hard.

2.4 Selecting Seed Nodes of the Maximum Influence

2.4.1 Centrality-Based Method

As a naive method for influence maximization, we can compute centralities of all nodes and select k biggest nodes. Closeness centrality in a dynamic network is defined based on an assumption that a node is central if the shortest periods from the node to all other nodes are small, which is expressed as follows[HS12]: $C_i^C = \frac{N-1}{\sum_j d_{ij}}$, where N is the number of nodes, d_{ij} is the shortest period from node i to node j, respectively. In the process of information propagation, not only the shortest path but also other longer paths will play important roles. Since closeness centrality focuses on the shortest path only, it may not be a good metric for information propagation.

Grindrod et al. extend Katz centrality[Kat53] to dynamic networks, and propose broadcast centrality[GPHE11]. Broadcast centrality takes all walks between two nodes into consideration, which is defined as follows: $C_i^B = \sum_{k=1}^{N} Q_{ik}$,where $Q_{ik} = \left[(I-aA_1)^{-1}(I-aA_2)^{-1}\cdots(I-aA_T)^{-1}\right]_{ik}$ and a is an attenuation parameter for discounting longer walks. If the maximum value of the largest eigenvalue of all adjacency matrices is $\lambda_{\max}$, parameter a has to satisfy $a < \frac{1}{\lambda_{\max}}$. The definition of walks by Grindrod et al. is a little bit different from the definition in the last section. In the last section, a walk $(t_1,n_1,n_2),(t_2,n_2,n_3),\ldots,(t_{k-1},n_{k-1},n_k)$ should satisfy $t_1 < t_2 < \cdots < t_{k-1}$, whereas a walk by Grindrod's definition should satisfy $t_1 \leq t_2 \leq \cdots \leq t_{k-1}$ only. In other words, the number of move at each time step in a walk in the last section is limited up to one, whereas there is no such limitation to a walk by Grindrod's definition. Grindrod's definition allows walks that cannot be the paths for information propagation of SI model, so it may not be a good metric for information propagation, either.

2.4.2 A Method Based on Monte-Carlo Simulation

Berger-wolf et al. propose a greedy method for solving influence maximization problem which approximates the power of influence of node set in SI model by Monte-Carlo

simulation[BW07]. However, the method needs much computational time for better approximation. Our proposed method uses fast heuristic instead of Monte-Carlo simulation to approximate the power of influence of node set.

3 Proposed Method for Selecting Seed Nodes

This section proposes a method for selecting seed nodes that starts from empty node set $\mathscr{S} = \emptyset$. In our method, node n that maximizes $\hat{\sigma}(\mathscr{S} \cup \{n\})$, where $\hat{\sigma}(\cdot)$ is approximated power of influence of node set, is added to $\mathscr{S}$ incrementally. $\hat{\sigma}(\mathscr{S})$ is calculated in the following way.

1. Let $\hat{p}_i(t)$ the approximated probability that node i is in status I at time t. $\hat{p}_i(1)$ is initialized as follows:
$$\hat{p}_i(1) = \begin{cases} 1 & i \in \mathscr{S} \\ 0 & i \notin \mathscr{S}. \end{cases}$$
2. At time $t = 2,3,\ldots,T+1$, $\hat{p}_i(t)$ is computed in the following way:$\hat{p}_i(t) = 1 - (1 - \hat{p}_i(t-1))R_i(t-1)$, where $R_i(t)$ is the approximated probability that none of the neighbors of node i at time t propagates information, which are expressed as follows: $R_i(t) = \prod_{j \in \text{neighbors}(i,t)}(1 - \hat{p}_j(t)\lambda)$, where neighbors$(i,t)$ is the set of neighbors of node i at time t.
3. $\hat{\sigma}(\mathscr{S})$ is calculated as the expected number of I nodes at time $T+1$ in terms of approximated probability $\hat{p}_i(T+1)$, i.e. $\hat{\sigma}(\mathscr{S}) = \sum_{i=1}^{N} \hat{p}_i(T+1)$.

An example of exact value and its approximate value of σ are shown in Figure 2. A label of an edge in Figure 2 shows the time that the edge appears. Suppose the seed nodes at time 1 is $\mathscr{S} = \{A\}$, and we are going to compute the probability $p_B(4)$ that node B is in status I at $T = 4$. In exact computation, p_B is affected by edge $(1,A,B)$ only, so the final probability is $p_B(4) = \lambda$. It seems that p_B is also affected by edge $(3,C,B)$, but this is not true. If node C is in status I at time $t = 3$, node B is already in status I, so p_B will not be affected with the edge from C to B. In this way, we have to judge whether each edge actually affect the probability in status I in order to perform exact computation. However, this procedure is computationally expensive.

We propose a method for approximating this computation shown above. In this method, all edges that are connected to a node are assumed to affect the probability that the node is in status I. Based on this method, the above probability $p_B(4)$ in Figure 2 is computed as $p_B(4) = \lambda + (1-\lambda)\lambda^3$, which is $(1-\lambda)\lambda^3$ more than true probability. Our approximation method overestimates the probability of a node to be in status I on networks having cycles.

As for computational complexity of our method, computational time for updating $\hat{p}_i$ needs time that is proportional to the number of edges m in a network. So the computational time for the update is $\mathscr{O}(m)$. In order to select k nodes that should be added to $\mathscr{S}$, approximate computation of σ is repeated N times, where N is the number of nodes in the network. Therefore, the total computational time will be $\mathscr{O}(Nmk)$.

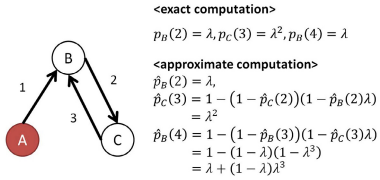


Fig. 2. Approximate computation of $p_B(4)$

Table 1. Statistics of dynamic networks

	nodes	edges	time period	modularity	density
Hospital	75	2,424	5,792	0.367	0.410
Infectious	200	943	469	0.883	0.036
TI model	500	308,000	3,000	0.892	0.006

4 Experiments

4.1 Experimental Settings

We have performed experiments using some dynamic networks and compare the performances of proposed method and some other methods. Three dynamic networks that we used for our experiments are shown in Table 1. Hospital network [VBC$^+$13] shows dynamic proximities of patients and workers in a French hospital. Infectious network [ISB$^+$11] also shows dynamic proximities at a science gallery in Ireland. TI model network is a synthetic network generated by Triad-enhanced Interaction model which is proposed by Jo et al. [JPK11].

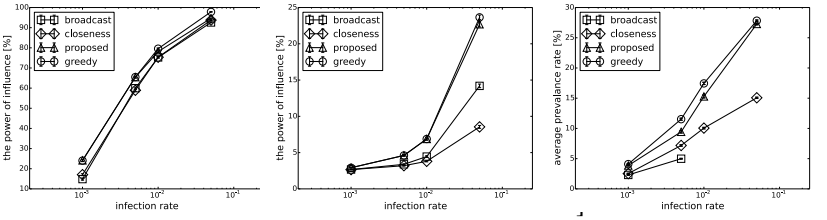
As baseline methods, the following three methods are attempted: (1) a method of selecting nodes of top-k closeness centrality values (closeness method), (2) a method of selecting nodes of top-k broadcast centrality values (broadcast method) and (3) the greedy method based on Monte-Carlo simulation proposed by Berger-wolf et al. (greedy method).

In this experiment, we fix the number of seed nodes $k = 5$ and set infection rate $\lambda = 0.001, 0.005, 0.01, 0.05$ to observe behaviors of methods for values of λ. As for the parameters for broadcast centrality a, for our proposed method λ and for greedy method λ, the same value as infection rate λ is used.

Based on the seed nodes that are selected with our proposed method and the baseline methods, simulations of information propagation based on SI model are performed 1,000 times to calculate the power of influence of seed nodes selected by methods, which is used to evaluate the quality of them.

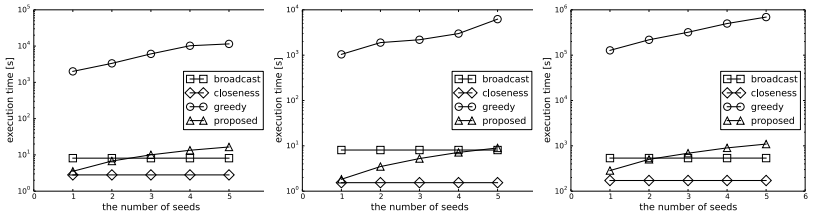
4.2 Results

Results for each network are shown in Figure 3. In this Figure, X axis is infection rate λ, and Y axis is the power of influence of seed nodes selected by each method. For most of values of λ, our proposed method successfully select seed nodes that are more



(a) Hospital Network (b) Infectious Network (c) TI model Network

Fig. 3. The power of influence for values of λ in each network



(a) Hospital network (b) Infectious network (c) TI model network

Fig. 4. Computational time for each network

influential than those selected by two centrality-based methods. But in Hospital network (Figure 3(a)), the power of influence of them are almost equal when $\lambda = 0.05$, and all methods can propagate the information to more than 90% of nodes in the network. On the other hand, in Infectious network and TI model network, the advantage of proposed method becomes larger as the value of λ increases. In TI model network, broadcast centrality cannot be calculated because of the irregularity of matrix $(I - aA_t)$. Compared with greedy method, our proposed method can select seed nodes as influential as the one selected by greedy method even though our proposed method is quite faster than it as shown below.

Computational times for all methods are shown in Figure 4. X axis of the Figure is the number of seed nodes, and Y axis is the computational time. Since closeness method and broadcast method need to compute centralities of all nodes in a network, their computational times are the same regardless of the value of k. On the other hand, computational times of our proposed method and greedy method are proportional to the number of seed nodes k. In all of our cases, the closeness method is the fastest and greedy method is the slowest. Our proposed method is the second or third slowest, but its computational time is still practical even though its performance is almost equal to greedy method which is 500 times slower than the proposed method.

In summary, we can claim that our proposed method can select seed nodes that is as influential as the one obtained with greedy method, which is the most accurate method in the comparison and more accurate than two centrality-based methods, in most of our parameter settings. Computational time of proposed method is slower than two centrality-based methods but is still practical and 500 times faster than greedy method.

5 Discussion

Experimental results in the last section show that for some networks and parameter settings, our proposed method does not outperform two centrality-based methods. One of the reasons for this is that such networks are too dense and they have no community structures. There is no clear definition of community structures especially for dynamic networks. For the sake of convenience, we define "the existence of community structures in a dynamic network" as "the existence of partitions of high modularity[New06] for its aggregated static network". Modularity Q is a function that takes a network and its partition as its input, and a value for showing the goodness of the partition as it output, which is defined as follows:

$$Q = \frac{1}{2m} \sum_{i,j} (A_{ij} - \frac{k_i k_j}{2m}) \delta(C_i, C_j),$$

where A is an adjacency matrix of a network, k_i is the degree of node i, C_i is a community that node i belongs to, $m = \frac{1}{2} \sum_{i,j} A_{ij}$ is the number of edges, respectively. High modularity values will be obtained for the partitions whose intra-community densities are high and whose inter-community densities are low. As a method for optimizing modularity, Louvain method [BGLL08] is used.

Modularity values and density for each static aggregate network are shown in Table 1. Modularities of Infectious network and TI model network are very high, while that of Hospital network is not. As for the densities of these static aggregated networks, density of Hospital network is quite high compared with those of other two networks.

If a network is dense and exhibits no community structure, each node in the network can propagate information to many others especially when λ is high. Therefore all methods including centrality-based methods can select very influential seed nodes. On the other hand, if a network is sparse and exhibits community structure, information tends to stay within the communities in which the seed nodes are. In this case, selecting seed nodes from the same communities will be ineffective for information propagation because they may have many overlapping adjacent nodes as we pointed it out as one of the problems of centrality-based influence maximization methods.

6 Conclusion

This paper proposes a method for selecting seed nodes in a dynamic network that are the most influential in information propagation. Experimental results show that our proposed method is effective for some networks compared with the strategies based on centralities for dynamic networks. In comparison between proposed method and greedy method, it is shown that proposed method is as effective as greedy method for some networks, and consistently 500 times faster than it. Our proposed method is especially good for the networks exhibiting community structures.

References

BGLL08. Blondel, V.D., Guillaume, J.-L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment 2008(10), P10008 (2008)

BW07. Berger-Wolf, T.Y.: Maximizing the extent of spread in a dynamic network. DIMACS Technical Report 2007-20, 10 pages (2007)

BZW07. Bai, W.-J., Zhou, T., Wang, B.-H.: Immunization of susceptible–infected model on scale-free networks. Physica A: Statistical Mechanics and its Applications 384(2), 656–662 (2007)

CSH+13. Cheng, S., Shen, H., Huang, J., Zhang, G., Cheng, X.: Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In: Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management, pp. 509–518 (2013)

CWW10. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029–1038 (2010)

GPHE11. Grindrod, P., Parsons, M.C., Higham, D.J., Estrada, E.: Communicability across evolving networks. Physical Review E 83(4), 046120 (2011)

HS12. Holme, P., Saramäki, J.: Temporal networks. Physics Reports 519(3), 97–125 (2012)

ISB+11. Isella, L., Stehlé, J., Barrat, A., Cattuto, C., Pinton, J.-F., Van den Broeck, W.: What's in a crowd? analysis of face-to-face behavioral networks. Journal of Theoretical Biology 271(1), 166–180 (2011)

JHC12. Jung, K., Heo, W., Chen, W.: Irie: Scalable and robust influence maximization in social networks. In: ICDM, pp. 918–923 (2012)

JPK11. Jo, H.-H., Pan, R.K., Kaski, K.: Emergence of bursts and communities in evolving weighted networks. PloS One 6(8), e22687 (2011)

JSC+11. Jiang, Q., Song, G., Cong, G., Wang, Y., Si, W., Xie, K.: Simulated annealing based influence maximization in social networks. In: AAAI, pp. 127–132 (2011)

Kat53. Katz, L.: A new status index derived from sociometric analysis. Psychometrika 18(1), 39–43 (1953)

KKT03. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146 (2003)

New06. Newman, M.E.J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103(23), 8577–8582 (2006)

PS12. Peyrard, N., Sabbadin, R.: Evaluation of the expected size of a sir epidemics on a graph. UBIAT Resarch Report, RR-2012-1 (2012)

VBC+13. Vanhems, P., Barrat, A., Cattuto, C., Pinton, J.-F., Khanafer, N., Régis, C., Kim, B.-A., Comte, B., Voirin, N.: Estimating potential infection transmission routes in hospital wards using wearable proximity sensors. PloS One 8(9), 73970 (2013)

Neighbourhood Distinctiveness: An Initial Study

A. Hecker, C.J. Carstens, and K.J. Horadam

RMIT University, Melbourne, Australia
{corriejacobien.carstens,kathy.horadam}@rmit.edu.au

Abstract. We investigate the potential for using neighbourhood attributes alone, to match unidentified entities across networks, and to classify them within networks. The motivation is to identify individuals across the dark social networks that underly recorded networks. We test an Enron email database and show the out-neighbourhoods of email addresses are highly distinctive. Then, using citation databases as proxies, we show that a paper in CiteSeer which is also in DBLP, is highly likely to be matched successfully, based on its (uncertainly labelled) in-neighbours alone. A paper in SPIRES can be classified with 80% accuracy, based on classification ratios in its in-neighbourhood alone.

Keywords: local structure, neighbourhood matching, instance matching.

1 Introduction

There are many sets of large databases that contain overlapping and complementary information. For instance, a social network example is the Twitter, Facebook and LinkedIn databases and a bibliographic example is the CiteSeer, DBLP, Google Scholar and Scopus databases.

Sometimes the same entity appears in multiple databases but with a different description, either due to errors or to the data having inherent differences, such as user names within different social media databases. Matching across databases at the instance level (also termed reference reconciliation), that is, matching different individual descriptions referring to the same real-world entity, is important for both discovery and database management.

Our focus here is discovery: sometimes entities represent humans or organisations operating incognito or under several aliases, for either legal or illegal reasons. From this perspective it is natural to investigate the *context* of an entity: those entities in a database that are directly linked or related to it [6], and further, to investigate their interlinkages (eg. [14, Figure 1]). In network terms this is the set of nodes directly surrounding such a node, together with the edges between them. In an undirected network this is the (open) neighbourhood of a node: the subgraph induced by the neighbours of the node. In social network terms, this is the ego-network without the ego.

Our interest in this idea arises from a security analysis problem in communication networks. An example of a real scenario where discovery is important and the neighbourhood may help, is when a person of interest to authorities uses

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_10

an unidentified mobile phone in order to remain untraceable, but still makes calls to other phones that are identified. Similarly, by using a publicly accessible Internet terminal the individual may intend to remain unidentifiable, but is likely to access sites that are identified. The two dark social networks underlying these communication networks will have other people in common. Can we match an individual in both communication networks if sufficiently many nodes in his neighbourhoods can be identified across both networks, or, failing that, if the two neighbourhoods match structurally sufficiently well? A less specific question to ask is whether nodes can be classified into broad types according to features of their neighbourhoods. For instance in the affiliation subnetwork of the Noordin Top terrorist group [10], actors are classified into six categories (operations, logistics, organizations, training, finance and meetings) inferred from their mention together in public reports in newspapers and elsewhere.

There is a long history of characterising and modelling networks by the local structure around their nodes. In social network analysis the triad census for a network was introduced in [12] and counts of connected triads, and 4-node motifs, are used as summary statistics in the seminal motifs paper [16]. Motif profiling in networks is now a well accepted technique, though there is criticism of its reliance on a null-model [3]. More general subgraph features have been used to describe the emergence of symmetry [22] and as explanatory variables in the exponential random graph model, which seeks to model global network structure better, as a function of local features [20].

Similarly, there is an extensive literature on database alignment, on name disambiguation and on approximate graph matching for de-anonymising social networks. In [17] it is shown that if a seed set of nodes has been pre-matched between two networks, structural features of node neighbourhoods such as number of nodes, number of edges, and clustering coefficient, that are independent of node labels, can be used to propagate further node matches between the networks with good accuracy. The algorithm in [18] avoids using a seed set by starting from the highest degree node in each of two equal-sized networks.

If we focus on the local level alone, the potential for characterising the ego-network or the two-hop neighbourhood of a single node by using subgraph measures is assessed in [8]. The Wikipedia edit network example suggests quality classification of article nodes based on features of their two-hop networks is possible. In [13] the in-neighbourhood of papers in a CiteSeer citation network is shown to be highly distinctive.

We report here case studies of whether an arbitrary node can be matched ab initio across two unequal networks, or typed within a network, based solely on features of its neighbourhood. For the first problem we require networks in which we can be certain some of the the same entities will appear, even if they are uncertainly labelled, and for the second we require networks in which nodes are assigned to types. We use bibliographic databases for our experiments, since for both problems we require publicly available databases in which information, an idea or influence travels from a node to its neighbourhood. In the derived citation

networks, each node corresponds to a paper and each directed edge corresponds to a citation. An edge points from the citing paper to the cited paper.

We first check our approach with an Enron email database, since it is a publicly available directed communication network in which a subset of people in the underlying social network were acting illegally. We select the CiteSeer and DBLP networks for testing the first question, since both mostly consist of computer science papers, so we can hope to find many nodes in common, even if they are not identically labelled. They also have nodes in common with other databases, such as Google Scholar and Scopus. We select the high-energy physics network SPIRES for testing the second question, since in it most nodes are typed into one of five categories.

There are other useful similarities of citation networks as proxies for the dark social networks of interest, though we do not take advantage of them here. Reliable information about nodes can often be extracted from their neighbourhood in the presence of missing information or significant errors. For instance, if the publication date for a paper is missing or wrong, an approximate publication date can be inferred as being slightly earlier than the publication dates of papers that cite it. As another example, it is often difficult to determine the subject area of a paper from its title, but the papers which cite it can provide this information by repetition of keywords or classifications.

In the citation networks we concentrate on the subgraph induced by the in-neighbours. Our first contribution is to show that the node sets of in-neighbourhoods of papers in CiteSeer and in DBLP are sufficiently distinctive that it is very likely that a paper which appears in both databases will be matched successfully. It is very likely that the node sets of the in-neighbourhoods of two different papers appearing in both databases will not match. Our second contribution is to show that a node in SPIRES can be correctly categorised by features of its in-neighbourhood alone about 80% of the time.

2 Neighbourhoods

In this section we briefly describe the communication and citation networks we use and detail basic properties of the directed neighbourhoods that were extracted.

The cleaned, directed Enron network that we use is accessible online [9]. It has 22,477 nodes and 53,285 edges. The complete CiteSeer archive database is accessible online [7]. Cleaning and processing involves removal of all papers with no references. The resulting citation network has $383,535$ vertices and $1,740,303$ edges. The average indegree is 4.5. The DBLP V3 dataset is already cleaned and publicly available in citation network form [21]. Node labels in DBLP and CiteSeer for the same paper often have slightly different syntax, so a pure string match on paper titles will fail to capture the match between them. This is a case of uncertain labelling. The SPIRES dataset we use is that studied in [15], where papers are assigned to five categories: Theoretical, Experimental, Phenomenology (papers coded as being both Theoretical and Experimental), Review and

Instrumentation, or else are unassigned. After cleaning and conversion, a network with 353,954 vertices and 3,921,382 edges is created.

Formally, the *in-neighbourhood* of a node $v \in V$ in a directed network $G = (V, E)$ is the subgraph $N_{in}(v) := (V_{in}(v), E_{in}(v))$ induced by the in-neighbours of v, where

$$V_{in}(v) = \{w \in V \,|\, (w, v) \in E, w \neq v\}, \quad E_{in}(v) = \{(u, w) \in E \,|\, u, w \in V_{in}(v)\}.$$

The *out-neighbourhood* $N_{out}(v)$ of a node v is analogously defined. We explicitly exclude the node v itself. In a citation network this condition is superfluous, since a paper cannot cite itself. In a citation network the in-neighbourhood of a paper corresponds to the subnetwork of papers that were directly influenced by the paper. In a communication network the out-neighbourhood of a sender corresponds to the subnetwork of receivers that were directly influenced by the communication.

Close to half (49%) the nodes in the CiteSeer database and over one third (35%) of the nodes in the SPIRES database have no citations and will be indistinguishable via their in-neighbourhoods. They are removed from study. The remaining papers were partitioned by citation number (indegree) into ranges that increased exponentially, see Table 1. The smaller Enron database was similarly partitioned by outdegree, for comparison.

From now on, "neighbourhood" will mean *out*-neighbourhood for the Enron network and *in*-neighbourhood for the citation networks.

Table 1. Partitioning of the databases by number of recipients (Enron) or citations (CiteSeer and SPIRES)

Partition	k **Range**	**Enron**	**CiteSeer**	**SPIRES**
1	1	8711	51949	44652
2	2-3	3270	50823	46928
3	4-7	1320	40313	45010
4	8-15	528	26669	38152
5	16-31	214	14510	27104
6	32-63	98	6700	15663
7	64-127	56	2543	7549
8	128-255	41	793	3023
9	256-511	9	207	945
10	512-1023	1	43	273
11	$\geq$ 1024	0	8	81

We used the igraph package in R to extract neighbourhoods. Examination of visualisations of neighbourhoods reveals that they have a large range of different structural features. For example, in Figure 1 we illustrate the neighbourhoods of two papers classified in different categories in SPIRES. Visually their structures are very different. Some examples for CiteSeer appear in [13].

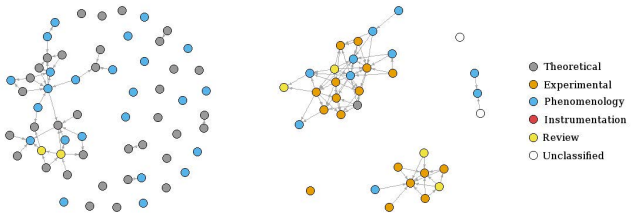


Fig. 1. SPIRES neighbourhoods of Phenomenology paper [11] (left) and Experimental paper [1] (right)

In order to measure the distinctiveness of neighbourhoods it is necessary to have a measure of either the similarity or difference of graphs. If the nodes are labelled, as here, then the simplest measure of difference is a node-based score. In essence this is the ground truth. If the nodes are uncertainly labelled then metrics which are structural or fuzzy would be necessary.

We use the Jaccard distance of the neighbours (the relative set difference) to measure dissimilarity of neighbourhoods. That is, if A and B denote the neighbour sets of nodes v_a and v_b respectively, their Jaccard distance d is:

$$d(A, B) = 1 - |A \cap B|/|A \cup B|$$

where $|\ |$ is set cardinality. Compared with other strictly node-based scores, the Jaccard distance has relatively good discriminatory performance on a selection of databases [23].

To test the likelihood that neighbourhoods in a communication network and in a proxy citation network are distinctive, we first ran two experiments on both the Enron network and the CiteSeer network. For the first experiment, 100 nodes were selected from each of the partitions listed in Table 1 (with replacement if necessary). Each node was paired with 1,000 nodes, randomly selected from the whole database excluding the node itself, and the Jaccard distance between them was calculated. In both cases, the cumulative relative frequency drops from 1 very rapidly. For the Enron network, out of the 1,000,000 random pairings only 15 pairs are found with Jaccard distance 0, all in Partition 2, only 858 pairs with Jaccard distance below 0.7 and only 8968 pairs below 0.9. For the CiteSeer network, out of the 1,100,000 random pairings 0 pairs are found with Jaccard distance 0, only 6 below 0.7 and only 46 pairs below 0.9.

For our second experiment we look at worst-case matching, where neighbourhoods are guaranteed to overlap. Again, we chose 100 nodes randomly from each partition, and then matched each of the nodes to all of the nodes in the database that had at least one common neighbour with the selected node. This means only node pairs which are most likely to have a low Jaccard distance are tested. For the Enron network, out of the 531,714 nearby pairings only 759 pairs are found with Jaccard distance of 0, and all of these are in Partitions 1 and 2. There are only 3781 pairs with Jaccard distance below 0.7 and 30,957 pairs with Jaccard distance below 0.9. For the CiteSeer network, out of the 537,932 nearby pairings

0 pairs are found with Jaccard distance of 0, only 54 pairs with Jaccard distance below 0.7 and 1163 pairs with Jaccard distance below 0.9.

The cumulative distributions are shown in Figure 2. We conclude the neighbourhoods are highly distinctive in each of these databases, and distinctive in similar ways.

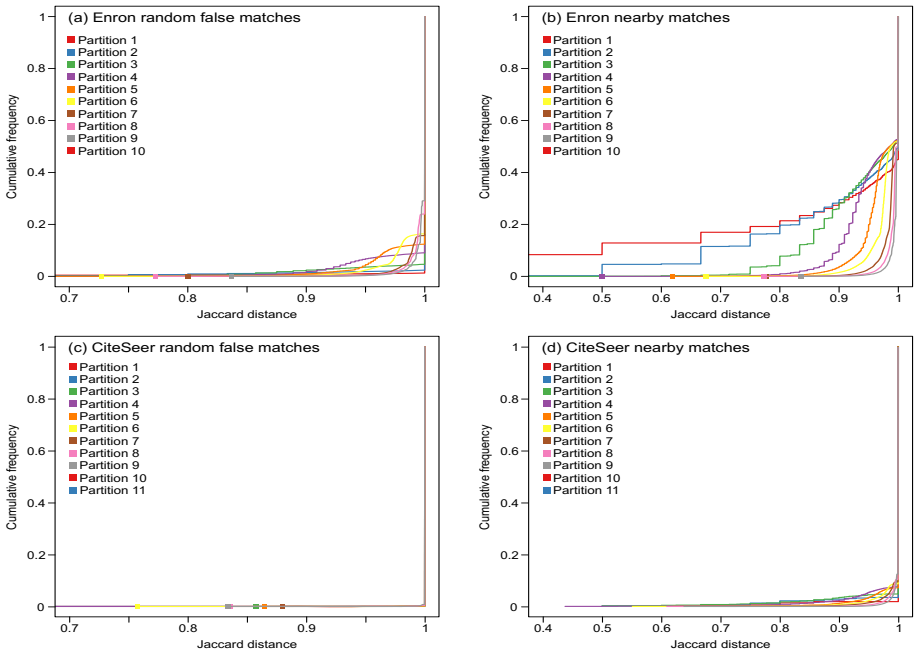


Fig. 2. Distinctiveness of neighbourhoods within Enron and CiteSeer networks

3 Matching CiteSeer Neighbourhoods in DBLP

For this experiment, we first corrected for uncertain labelling. All the paper titles in both the DBLP and Citeseer databases were converted to lower case only, stripped of all punctuation and stripped of white spaces at each end. For example, the title "Computer Science: Science of the Future! " became "computer science science of the future". After this processing and string matching, the number of nodes with identical titles in Citeseer and DBLP is 115,803. This is 7% of the DBLP node set and 16.84% of the Citeseer node set. This processing did not remove all syntactical differences between representations of the same paper in the two databases, so we can expect that some true matches have not been included.

We extracted neighbourhoods for all papers in CiteSeer, and restricted to the 51,473 with neighbourhoods of size ≥ 8 (see Table 1). We found 13,555 matching papers in DBLP and extracted the neighbourhoods for these.

Any nodes in a neighbourhood from DBLP that did not appear in Citeseer were removed when calculating scores. We did not perform the symmetric restriction of CiteSeer neighbourhoods with respect to DBLP, because our focus is on whether a node in CiteSeer can be identified with one in DBLP, though this would be expected to improve matching scores. If a DBLP node had an empty neighbourhood as a result of this node removal, it was excluded from the DBLP set. After this restriction, there were 12,582 DBLP nodes, all of which matched nodes in CiteSeer with neighbourhoods of size ≥ 8.

For each of the 12,582 papers appearing in both CiteSeer and DBLP, we calculated the Jaccard distance between its in-neighbour set in CiteSeer and its in-neighbour set in DBLP. This gave a distribution of genuine scores.

A distribution of imposter scores was generated by taking 200,000 random different pairings between the databases. A node was chosen at random from the 12,582 DBLP nodes, then paired at random with a node from the 51,473 Citeseer nodes. The paired node in Citeseer need not have a match in DBLP. The Jaccard distance was calculated between the in-neighbour set of a node in DBLP and that of its paired node in CiteSeer.

The separation of genuine from imposter scores is very good, and we report only summary results of correct decision and error rates at a threshold of 0.95 in Table 2, representing a choice to minimise the False Match Rate.

Table 2. Matching rates at Jaccard distance threshold = 0.95

	True (T)	**False (F)**
Match Rate (MR)	0.99993	0.00007
Non-Match Rate (NMR)	0.85670	0.14330

Increasing the decision threshold will shift choice towards minimising the False Non-Match Rate. A threshold of 0.999 minimises the sum of errors (FNMR = 0.001, FMR = 0.051).

This shows that (uncertainly labelled) nodes in CiteSeer can be reliably identified with nodes in DBLP, based on their neighbours alone. Furthermore, nodes in CiteSeer can reliably be distinguished from non-matching nodes in DBLP, based on their neighbours alone. Because the nodes are labelled, it was always possible that the True Match and True Non-Match rates based on neighbour sets alone would be as good as this, without requiring any more complex graph-matching techniques to be applied to the neighbourhoods. These are very encouraging results for the instance matching problem, in cases where enough nodes in each neighbourhood can be identified.

In the next section we remove node labels and try matching based on more general neighbourhood attributes.

4 Classifying by Neighbourhood in SPIRES

In this section we describe SVM (support vector machine) experiments to determine whether neighborhoods for papers are sufficiently distinctive to permit

classification of papers into categories. In the SPIRES database 56,116 (15.4%) of the neighbourhoods have ≥ 16 nodes, and their distribution into categories is given in Table 3.

Table 3. Distribution of papers with neighbourhood size ≥ 16 in SPIRES categories

Category	#	%	**Category**	#	%
Theoretical	28244	51.69	Experimental	5495	10.06
Phenomenology	17618	32.25	Instrumentation	444	0.82
Review	3059	2.89	Unclassified	1256	2.29

We inspected visualisations of many neighbourhoods to inform our feature selection, and eventually selected 14 features: five based solely on attributes of the set of neighbours (the ratios of the number of nodes labelled Theoretical, Experimental, Phenomenology, Instrumentation and Review, respectively, to the number of nodes in the neighbourhood); and 9 based on structural properties of the neighbourhoods.

The first five structural features are: edge density (for a directed network); transitivity ratio (or global clustering coefficient); ratio of isolated nodes in the neighbourhood; ratio of the number of nodes in the largest connected component in the neighbourhood; and *disconnectedness*, a measure introduced here. The disconnectedness $\Delta(G)$ of a non-empty graph $G = (V, E)$ is $\Delta(G) = 0$ if G is connected and $\Delta(G) = \frac{G_c}{|V|}$ otherwise, where G_c be the number of connected components in the graph. It takes a value between 0 and 1 where 0 corresponds to a connected network and 1 to a collection of isolated vertices.

The remaining structural features are the four possible 3-node motifs (pictured in Figure 3). Three of them (Motifs 2, 3 and 4) have been shown to be distinctive within the two largest categories (Theoretical and Phenomenology) of the SPIRES network as a whole [5]. The motif features were computed as follows. For each neighbourhood H, we counted the number of occurrences of each of the four subgraphs in Figure 3. We generated 1,000 random directed acyclic networks with the same number of nodes and edges as H (degree distribution was not taken into account) and computed the z-score for each subgraph. Computational constraints meant that it was only possible to use a linear kernel when building a multiclass SVM model. We used 5-fold cross validation once, on all 56,116 neighbourhoods using all 14 features, and again using only the 9 structural features. The corresponding Test set misclassification rates for three standard

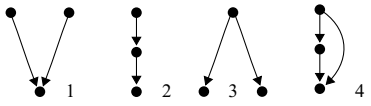


Fig. 3. Structural features 11, 12, 13, 14: the possible 3-node motifs in a citation network

parameter values are given in Table 4. Feature reduction may further improve classification accuracy, as may using a different kernel. It seems clear that any feature reduction should be through removal of structural features since, using them alone, the classification performance for a binary decision deteriorates to little better than chance. To double-check this, Information Gain (IG) [19] was calculated for each of the 14 features for the entire database of neighbourhoods, see Table 5. Of the motif features, only the feedforward loop (Motif 3) shows any possibility of contributing to classification; the low IG score for the others indicates they could be removed from the model.

Table 4. Misclassification rates for all in-neighbourhoods, using a linear kernel

Parameter	Features 1–14	Features 6–14
$C = 0.1$	0.2085	0.4705
$C = 1$	0.2084	0.4704
$C = 10$	0.2084	0.4705

Table 5. Information Gain (IG) for 14 features for all SPIRES in-neighbourhoods

Feature	IG	Feature	IG
1. Theoretical Ratio	0.44595	8. Edge Density	0.02565
2. Experimental Ratio	0.26950	9. Disconnectedness	0.02160
3. Phenomenology Ratio	0.33539	10. Largest Component Ratio	0.02284
4. Instrumentation Ratio	0.05903	11. z-score of Motif 1	0.00259
5. Review Ratio	0.03638	12. z-score of Motif 2	0.00723
6. Isolated Nodes Ratio	0.02543	13. z-score of Motif 3	0.01914
7. Clustering Coefficient	0.01186	14. z-score of Motif 4	0.00647

After removing unclassified papers, we repeated the experiment on the first 10 features with a Gaussian kernel, again using 5-fold cross-validation, training using several smaller training sets sizes and optimising with $n = 500$ in the training set and 100 in the validation subset. Random selection ensures distribution of the papers belonging to each category is approximately represented in the training set and validation subset. In order to reduce sample variation, the model was trained and validated 100 times. For each training set, an independent Test set of size 2,000 was chosen randomly from the set of available neighbourhoods. Again, testing was repeated 100 times. A simple wrapper was run to find the smallest, best performing feature set. IG scores applied to each feature confirmed the feature selection. The models trained using features 2, 3, 4 slightly outperform the models trained using the first ten features ($t(198) = 6.77, p < 0.001$). Results appear in Table 6. They suggest that classification using attributes of the in-neighbour set as a whole is successful in 78.3% of cases, when applied to the test set.

Table 6. Performance comparison of the smallest, best performing feature set

Feature Set	V Misclass	T Misclass
1–10	0.1652 ± 0.0239	0.2262 ± 0.0093
2, 3, 4	0.1963 ± 0.0185	0.2173 ± 0.0093

In this experiment the category of the paper does not seem to be related to the structure of the neighbourhood, nor to the non-structural feature which had highest IG (Theoretical paper ratio), but this could be because this local citation behaviour is typical of scholarly articles in general and is independent of category.

5 Conclusions and Future Work

On the small Enron network which represents the type of network of interest, we have demonstrated that neighbourhoods are very distinctive, especially if they contain 4 or more out-neighbours. In the much larger CiteSeer network, we have demonstrated that neighbourhoods are very distinctive, especially if they contain 8 or more in-neighbours. Figure 2 shows very similar behaviour of neighbourhood distinctiveness between the two networks, so our use of citation networks as proxies is reasonable.

We have demonstrated that the neighborhood of a paper in CiteSeer is likely to match the neighbourhood of the same paper in the DBLP database, based on its node labels alone. The neighbourhood shows promise for good matching performance across databases, but requires further study. We have yet to test how the False Matches and False Non-Matches arise, and whether structural information can separate True Match and True Non-Match scores further.

The features that were used to classify a node successfully within the SPIRES network were the classification ratios of the in-neighbours of the node. A paper's category does not seem to be related to the structures in its neighbourhood that we tested, but perhaps other more useful structural features could be found. In other networks or for other categories the structure might matter more.

The distinctiveness of different node-based scores could be compared with that of Jaccard distance. Other similarity scores, such as the Adamic-Adar score [2] and RA and LP scores [23], which have somewhat better performance than the Jaccard similarity score on a selection of databases [23], can be tested. A natural generalisation of Jaccard distance which could also be used is the graph edit distance [4]. This metric measures the minimum cost to alter one graph to another where there are weighted costs to adding or removing nodes and edges. The Jaccard distance is the special case of the normalised graph edit distance where the cost of adding or removing edges is zero. Graph-edit distance based matching scores [4] allow us to compare the topology of two in-neighbourhoods directly. For instance, the maximum common subgraph of two in-neighbourhoods can be found using a graph edit algorithm, and properties based on subgraphs of the maximum common subgraph can be measured.

We have demonstrated that neighbourhoods have promise to discover or categorise uncertainly labelled nodes in citation networks. Ideally, we will be able to apply these ideas to communication networks containing uncertainly labelled nodes.

In general there is a continuum of problem types between uniquely labelled nodes through to unlabelled nodes, and we expect many useful applications to lie somewhere between the extremes. Many examples are likely to involve nodes that are partially labelled or labelled with errors, such as we consider here, and neighbourhood matching is a promising approach to the problem.

Acknowledgements. This work was supported by Commonwealth of Australia Department of Defence Research Agreement number 4500785154. We thank Sune Lehmann for providing the SPIRES database we use. We thank the anonymous referees for bringing several local neighbourhood studies to our attention.

References

1. Aalseth, C.E., et al.: Neutrinoless double-β decay of 76Ge: First results from the International Germanium Experiment (IGEX) with six isotopically enriched detectors. Phys. Rev. C 59, 2108–2113 (1999)
2. Adamic, L.A., Adar, E.: Friends and neighbours on the Web. Social Networks 25, 211–230 (2003)
3. Artzy-Randrup, Y., Fleishman, S., Ben-Tal, N., Stone, L.: Comment on "Network motifs: simple building blocks of complex networks" and "superfamilies of evolved and designed networks". Science 305(5687), 1107 (2004)
4. Bunke, H., Dickinson, P.J., Kraetzl, M., Wallis, W.D.: A graph-theoretic approach to enterprise network dynamics. Birkhäuser (2007)
5. Carstens, C.J.: A uniform random graph model for directed acyclic networks and its effect on finding motifs. J. Complex Networks 2, 419–430 (2014)
6. Castano, S., Ferrara, A., Montanelli, S., Varese, G.: Ontology and instance matching. In: Paliouras, G., Spyropoulos, C.D., Tsatsaronis, G. (eds.) Multimedia Information Extraction. LNCS, vol. 6050, pp. 167–195. Springer, Heidelberg (2011)
7. CiteSeer Archive, `http://citeseer.ist.psu.edu/oai.html`
8. Cunningham, P., Harrigan, M., Wu, G., O'Callaghan, D.: Characterizing ego-networks using motifs. Network Science 1(2), 170–190 (2013)
9. Enron email database, `http://sociograph.blogspot.com.au/2011/04/communication-networks-part-1-enron-e.html`
10. Everton, S.F.: Disrupting dark networks. Cambridge University Press (2012)
11. Gunion, J.F., Willey, R.S.: Hadronic spectroscopy for a linear quark containment potential. Phys. Rev. D 12(1), 174–186 (1975)
12. Holland, P., Leinhardt, S.: Local structure in social networks. Sociological Methodology 7(1), 1–45 (1976)
13. Jeffers, J., Horadam, K.J., Carstens, C.J., Rao, A., Boztaş, S.: Influence neighbourhoods in CiteSeer: a case study. In: Proc. SITIS 2013, pp. 612–618. IEEE/ACM (2013)
14. Lacoste-Julien, S., et al.: SiGMa: Simple greedy matching for aligning large knowledge bases. In: KDD 2013, pp. 572–580 (2013)

15. Lehmann, S., Lautrup, B., Jackson, A.D.: Citation networks in high energy physics. Phys. Rev. E 68(2), 026113 (2003)
16. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science 298(5594), 824–827 (2002)
17. Narayana, A., Shmatikov, V.: De-anonymizing social networks. In: 2009 IEEE Symposium on Security and Privacy, pp. 173–187 (2009)
18. Pedarsani, P., Figueiredo, D., Grossglauser, M.: A Bayesian method for matching two similar graphs without seeds. In: IEEE 51st Allerton Conference, pp. 1598–1607 (2013)
19. Roobaert, D., Karakoulas, G., Chawla, N.: Information gain, correlation and support vector machines. In: Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A. (eds.) Feature Extraction. STUDFUZZ, vol. 207, pp. 463–470. Springer, Heidelberg (2006)
20. Saul, Z.M., Filkov, V.: Exploring biological network structure using exponential random graph models. Bioinformatics 23(19), 2604–2611 (2007)
21. Tang, J., Zhang, J., Yao, L., Li, J., Zhang, L., Su, Z.: ArnetMiner: Extraction and Mining of Academic Social Networks. In: KDD 2008, pp. 990–998 (2008), `http://arnetminer.org/citation`
22. Xiao, Y., Xiong, M., Wang, W., Wang, H.: Emergence of symmetry in complex networks. Phys. Rev. E 77, 066108 (2008)
23. Zhou, T., Lü, L., Zhang, Y.-C.: Predicting missing links via local information. Eur. Phys. J. B 71, 623–630 (2009)

An Efficient Estimation of a Node's Betweenness

Manas Agarwal[1], Rishi Ranjan Singh[2], Shubham Chaudhary[3], and S.R.S. Iyengar[4]

[1] Department of Mathematics, Indian Institute of Technology Roorkee, Uttarakhand, India
manasuma@iitr.ac.in
[2] Department of Computer Science and Engineering,
Indian Institute of Technology, Ropar, Punjab, India
rishirs@iitrpr.ac.in
[3] Department of Mathematics, Indian Institute of Technology Roorkee, Uttarakhand, India
shubhuma@iitr.ac.in
[4] Department of Computer Science and Engineering,
Indian Institute of Technology, Ropar, Punjab, India
sudarshan@iitrpr.ac.in

Abstract. Betweenness Centrality measures, erstwhile popular amongst the sociologists and psychologists, have seen wide and increasing applications across several disciplines of late. In conjunction with the big data problems, there came the need to analyze large complex networks. Exact computation of a node's betweenness is a daunting task in the networks of large size. In this paper, we propose a non-uniform sampling method to estimate the betweenness of a node. We apply our approach to estimate a node's betweenness in several synthetic and real world graphs. We compare our method with the available techniques in the literature and show that our method fares several times better than the currently known techniques. We further show that the accuracy of our algorithm gets better with the increase in size and density of the network.

1 Introduction

Centrality of a node in a network is the quantification of the intuitive notion of importance of a node in a network. Centrality measures have been extensively used in the analysis of large data available from real world networks. Amongst a plethora of application specific definitions available in the literature to rank the vertices, closeness centrality, betweenness centrality and eigenvector centrality (page-rank) have been the most important and widely applied ones. For a detailed study of centrality indices and their applications, one can refer to the books by Newman [23] and Brandes and Erlebach [6]. We focus on betweenness centrality. It was proposed by Freeman [12] and Anthonisse [1] independently. Betweenness centrality of a node v is defined as the relative fraction of shortest paths passing through v. Betweenness centrality has found several important applications in diverse fields. One can refer [23,6] and the literature cited in related work section to explore the applications of betweenness centrality.

Real-world networks are generally very large in size, dynamic in nature and keep changing at a very high rate. In such networks, estimating betweenness centrality score of a node is of great importance. Consider for example, in the city network of a state, where nodes are the cities and edges are the roads connecting cities, and government has limited resources that can be used for the development of only one city. Then, out

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_11

of two cities, the government might want to pick one, developing which will benefit more to the state. This requires the government to compute and compare the individual importance (in this case betweenness) of the cities.

There are two reasons why the current state of the art algorithms for exact computation of a node's betweenness are not time efficient. Firstly because of the large size and the dynamic nature of networks. In large dynamic networks, we have to recompute the centrality scores each time the network changes, which is evidently expensive. Secondly because of the global characteristics of betweenness centrality. Unlike degree and closeness centralities, computing betweenness centrality of a node is conjectured to be as expensive as computing it for all the nodes in any network [18]. Thus, we are motivated to efficiently estimate a node's betweenness without computing betweenness of all nodes.

2 Related Work

Algorithms for exact betweenness computation are based on either single source shortest path (SSSP) computation algorithms from all nodes or all pair shortest path computation algorithms. The most trivial algorithm is a modified version of the all pair shortest path computation (APSP) algorithm to compute the betweenness scores for all nodes [12]. But this takes $O(n^3)$ time where n is the number of nodes. In year 2001, Brandes [5] introduced an algorithm based on the computation of single source shortest path (SSSP) that computes the exact betweenness score of all nodes in unweighted graphs in $O(mn)$ time, where m is the number of edges. Due to the size of current real world networks, even the state of art (Brandes') algorithm was very expensive in terms of time. This motivated the researchers to develop faster exact or approximation algorithms. Several exact algorithms for large graphs (Sariyüce et al. [25]) and dynamic graphs (Lee et al. [20], Green et al. [16], Kas et al. [17], Goel et al. [15], Nasre et al. [22]) have been developed. These algorithms improved the computation time experimentally on special type of graphs but in worst case they all were as expensive as Brandes'[5]. Several approximation algorithms were also proposed. These algorithms ran much faster and estimated the centrality scores close to the exact centrality scores. The approximation approaches in the literature can be grouped into two categories. The First category consists of algorithms that focus on computing the approximate betweenness of all nodes together (Brandes and Pich [7], Geisberger et al. [13], Gkorou et al. [14], Riondato and Kornaropoulos [24]). The second category comprises of algorithms that estimate the betweenness score of a given node (Bader et al. [2], Chehreghani[9]). Our goal is to develop an approximation algorithm of second category.

In this paper, we propose a novel non-uniform sampling technique that approximates very closely the optimal sampling explained in [9]. Our approach outperforms the estimations provided by Chehreghani's [9] work that already surpasses the uniform sampling based approaches ([7,2]). We organize the rest of the paper as follows. In next section we define basic terms used in the paper and briefly discuss Chehreghani's work. In section 4, we develop our model based on the analysis of random networks and some observations. All the details about simulations, data sets used in simulations, performance tools used for evaluation and extensive results in the form of plots and tables are compiled in section 5. We discuss the possible future directions of work and conclude the paper in section 6.

3 Preliminary

In this section, we introduce some basic terms related to the betweenness centrality that have been used throughout the paper. We also discuss the previous concepts that have motivated our sampling technique.

3.1 Terminology

We use following terms interchangeably; node or vertex and graph or network. For simplicity, we consider only unweighted undirected graphs until mentioned explicitly. All the concepts discussed in this paper can be easily generalized for weighted or directed graphs. Given a graph $G = (V,E)$, V is the set of nodes with $|V| = n$ and E is the set of edges with $|E| = m$. A (simple) *path* is a sequence of edges connecting a sequence of vertices without repetition of any vertex. . The *length* of a path is the number of edges in the path. *Shortest paths* between two vertices are the smallest length paths between them. *Distance* between two nodes i and j, $d(i,j)$, is the length of shortest path between i and j.

Let σ_{st} be the number of shortest paths between s and t, for $s,t \in V$. Let $\sigma_{st}(v)$ be the number of shortest paths between s and t passing through v, for $v \in V$. Betweenness centrality score of a node $v \in V$ is calculated as $BC(v) = \sum_{s \neq t \neq v \varepsilon V} \frac{\sigma_{st}(v)}{\sigma_{st}}$. *Pair dependency* of a pair of vertices (s,t) on a vertex v is defined as: $\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}$. Betweenness centrality of a vertex v can be defined in terms of pair dependency as $BC(v) = \sum_{s \neq v \neq t \in V} \delta_{st}(v)$. Let BFT_r be the breadth-first traversal (BFT) of the graph rooted on vertex r. In BFT_r, we assume that r is at level 0 and the next levels are labelled by natural numbers in an increasing order. *Dependency* of a vertex s on a vertex v is defined as: $\delta_{s\bullet}(v) = \sum_{t \in V \setminus \{s,v\}} \delta_{st}(v)$. Let us define a set $P^s(w) = \{v : v \in V,\ w \text{ is a successor of } v \text{ in } BFT_s\}$. Brandes [5] proved that:

$$\delta_{s\bullet}(v) = \sum_{w:v \in P^s(w)} \frac{\sigma_{sv}}{\sigma_{sw}} (1 + \delta_{s\bullet}(w)). \tag{1}$$

Algorithm 1. Approximation algorithm to compute betweenness score of a given node v [9]

1. **Input.** Graph G, probabilities $P = \{p_1, p_2, \cdots, p_n\}$, node v.
2. $BC(v) = 0$.
3. **for** i=1 to T **do**
4. Select a node i with probability p_i.
5. Compute $\delta_{i\bullet}(v)$ in the BFT_i using equation (1).
6. $BC(v) \leftarrow BC(v) + \frac{\delta_{i\bullet}(v)}{p_i}$.
7. **end for**
8. $BC[v] \leftarrow BC(v)/T$.
9. **Return.** $BC(v)$.

3.2 A Betweenness Approximation Technique Based on Non-uniform Sampling

In this section we will briefly describe the recent work of Chehreghani [9] that provides motivation for our work. He gave an approximation algorithm to compute betweenness score of a given node v. The algorithm is summarized as Algorithm 1. For a given node v, the algorithm takes the sampling probabilities as input and outputs the approximate betweenness score of node v. Step 2 initializes the betweenness score to 0. The algorithm estimates the betweenness score of node v, T number of times and takes the average of all T estimations. In each iteration of the algorithm, it samples a pivot node and computes the dependency of the pivot node on node v using a single iteration of Brandes' algorithm [5]. Then it estimates the betweenness score of node v by, dividing (scaling) the computed dependency by the sampling probability of that pivot node. He has motivated his paper with the idea of optimal sampling that is stated in the following theorem.

Theorem 1. *[9] Let the sampling probability assigned to each node i be*

$$p_i = \frac{\delta_{i\bullet}(v)}{\sum_{j=1}^{n} \delta_{j\bullet}(v)}$$

then, betweenness score of node v can be exactly calculated in $O(m)$ time using single iteration of Algorithm 1.

We refer the probability defined in Theorem 1 as *optimal probability* and call a model *optimal model (OPT)* if it can generate optimal probabilities. Calculating optimal probabilities is as expensive as computing exact betweenness using Brandes' algorithm [5]. Thus, a model was desired that can efficiently estimate sampling probabilities close to the optimal. Chehreghani noted that any such model should satisfy at least the following relation for most of the vertex pairs (i, j):

$$p_i < p_j \iff \delta_{i\bullet}(v) < \delta_{j\bullet}(v) \tag{2}$$

Chehreghani has given a simple distance based model (DBM)[9] to generate the sampling probabilities. He proposed to take the probabilities as the normalized value of the inverse of distance from node v to node i, $p_i \propto \frac{1}{d(v,i)}$. He has shown experimentally that his nonuniform sampling technique reduces the error in the computation of betweenness score as compared to uniform sampling technique([7,2]). But, he was unable to provide a theoretical derivation for DBM. In DBM, many of the nodes j with $\delta_{j\bullet}(v) = 0$ get same probabilities as nodes i with $\delta_{i\bullet}(v) \neq 0$ because of being at the same level in BFT_v. We propose a new probability estimation model for nodes that efficiently approximates the optimal probabilities and outperforms DBM.

4 A New Non-uniform Sampling Model

In this section, for a given node v, we propose a model that generates non-uniform probabilities for sampling the nodes very close to optimal probabilities. This model can be incorporated with Algorithm 1 to solve the considered problem. Our model is based on the inverse of degree and an exponential function in the power of distance, thus, we

refer it as EDDBM (exponential in distance and inverse of degree based model). We try assigning larger probability values to the vertices contributing more to the betweeness of a given node v and smaller to those that contribute less. We perform the analysis on the random graphs to establish the relation between probabilities and distance. Then, on the basis of few observations, we tune the probability function. Finally, we describe the steps to generate the probabilities by our model.

4.1 Analysis of Random Graphs

Let G be a random graph that is generated based on the $G(n,p)$ model given by Erdos Renyi [11]. We are given a vertex v to compute its betweenness score. We first analyze how the dependency of a node i on the node v, $\delta_{i\bullet}(v)$ varies when v lies on different levels in BFT_i. This will help us to establish a relation between $\delta_{i\bullet}(v)$ and the distance between i and v. For this, first we need to compute the expected number of nodes at any level m of a BFS traversal. We present a simpler version of the estimation technique given by Wang [29] to estimate the number of nodes at any level in BFS traversal. Let λ be the average degree of the given graph and let p be the probability of an edge's existence. The proof of all the Lemmas and Theorem 2 are available in the full version on the paper. The first lemma approximately estimates the number of nodes at a given level in a BFS traversal by a recurrence relation.

Lemma 1. *Let α_j be the number of nodes at level j in the BFS_i. Then, based on the exploration technique in random graphs by Van Der Hofstad [28], the number of nodes at level $m+1$, α_{m+1} can be given as:*

$$\alpha_{m+1} \approx np(1 - \frac{\sum_{j=0}^{m} \alpha_j}{n})\alpha_m. \tag{3}$$

Equation (3) is a recurrence relation to estimate the number of nodes at some level $m+1$. Using Lemma 1, we can estimate the ratio between the expected number of nodes at two consecutive levels. The ratio is given in Lemma 2.

Lemma 2. *Let α_m and α_{m+1} be the number of nodes at level m and $m+1$ respectively. Then we have $\frac{\alpha_{m+1}}{\alpha_m} \approx c_{m+1}\lambda$ where $c_{m+1} = (1 - \frac{\sum_{j=0}^{m} \alpha_j}{n})$ and $c_{m+1} \in [0,1)$.*

Based on lemma 2, we derive the formula to calculate the expected dependency of a node i on node v, $E[\delta_{i\bullet}(v)]$ in next lemma.

Lemma 3. *Let v be a node at level m in the BFS_i. Then the expected dependency of node i on node v can be given as*

$$E[\delta_{i\bullet}(v)] = (\frac{\alpha_{m-1}}{\alpha_{m-2}})(1 + c_m\lambda) \approx c_{m-1}\lambda(1 + c_m\lambda). \tag{4}$$

Now, we can give the theorem stating the ratio between dependencies of root node on two nodes positioned at two consecutive levels.

Theorem 2. *Let l be the last level in BST_i. Let $\delta_{i\bullet}(v_{l-k})$ be the dependency of node i at a node v_{l-k} at level $l-k$ and let $\delta_{i\bullet}(v_{l-k+1})$ be the dependency of node i at a node v_{l-k+1} at level $l-k+1$. Then we have*

$$\frac{E[\delta_{i\bullet}(v_{l-k})]}{E[\delta_{i\bullet}(v_{l-k+1})]} = c_{l-k+1}(\frac{1}{\phi}+\lambda) \tag{5}$$

where $\phi = (c_{l-k+2})(1+c_{l-k+3}\lambda(1+c_{l-k+4}\lambda(1+c_{l-k+5}\lambda(1+\cdots(1+c_l\lambda))\cdots)$.

It is simple to observe that c_m decreases continuously as m increases. So as v becomes closer to i, $E[\delta_{i\bullet}(v)]$ increases steeply, proportional to the average degree λ. Therefore, on the basis of lemma 2, we can assign a probability p_i to the node i as in the following theorem.

Theorem 3. *Suppose, we have to compute the betweenness score of node v. Then the sampling probability assigned to node i is :*

$$p_i \propto (\lambda)^{-d(i,v)} \tag{6}$$

where $d(i,v)$ is the distance between v and i.

4.2 EDDBM

Based on few observations we tweak our model. The observations and proposed tweak are available in the full version on the paper. We generate the final probabilities as following. First, we generate the probabilities on the basis of distance relation given in equation (6). Each node i at level d in the BFT_v will get following probability value: $p^d = \frac{(\lambda)^{-d}}{\sum_{j\in V\setminus\{v\}}(\lambda)^{-d}}$. Let V_d be the set of nodes at level d in the BFT_v and $|V_d|$ denotes the number of nodes in set V_d. Then to resolve the problem stated in Observation 1 to best extent, at each level d, we further tweak the formula and change the assigned probability to node i at d^{th} level to:

$$p_i = \frac{p^d|V_d|\cdot deg(i)^{-1}}{\sum_{j\in V_d} deg(j)^{-1}}.$$

5 Experimental Results

In this section, we discuss the experimental results achieved on extensive real world graphs and synthetic graphs. We have implemented the algorithms in Python Version 2.7.3 and used Networkx library for graph functions. All the simulations were performed on a 32 bit Ubuntu machine with 3.00GHz Intel Core 2 Duo E8400 processor and 3.4 GB RAM. We have not discussed the execution time of our approach. It is $O(Tm)$, same as for the approach in [9].

5.1 Data Sets

Real Networks. We have picked several real world networks. [4,21,10] can be referred for the description about the considered networks and data. We have shown

the betweenness computation results on real graphs in the Table 2. Details about the considered graphs and source of the data set are mentioned in the table. We have considered power grid networks, city networks, airline network, road network, random benchmarked networks and many more.

Synthetic Networks. For synthetic graphs we considered following three types of graphs: *Random Graphs (ER)* generated based on the $G(n,p)$ model given by Erdos Renyi [11]; *Scale-free Random Graphs (BA)* generated by the Albert Barabasi graph generation $G(n,k)$ model [3]; *Small World Graphs (WS)* generated by Watts Strogatz $G(n,k,p)$ model [30].

5.2 Performance Measurement Tool: Average Error

We use average error as a performance tool for our approach. It is defined as follows. Let $G=(V,E)$ be a given graph with $|V|=n$. Let $BC^e(v)$ be the exact betweenness score of node v in the graph G. Let $BC^a(v)$ be the betweenness score of the same node v computed by Algorithm 1 using probabilities generated by our model. Then, we define error in computation of betweenness score on node v same as Chehreghani[9]: $Er(v)=\frac{|BC^e(v)-BC^a(v)|}{BC^e(v)}\times 100$. We can define *average error* E in the computation of betweenness score of a set of nodes U, $U\subseteq V$, over a graph G as $E=\frac{\sum_{i\in U} Er(i)}{|U|}$ where $|U|$ denotes the number of nodes in set U. *Number of iterations* used for computation of betweenness score is also referred as number of sampled nodes. We denote it by T, where T is the percentage of total number of nodes n.

We take $T=10\%$ for all the simulations. To find the average error in the betweenness computation for a node, we take mean of the error over five iterations. We consider every node of a graph to measure the average error in the betweenness computation on that graph. For synthetic graphs, we take mean of the average error over five such artificial graphs.

5.3 Plots

In this section we evaluate the performance of EDDBM through various plots. With the help of different plots on synthetic networks, we compare the accuracy of EDDBM with DBM.

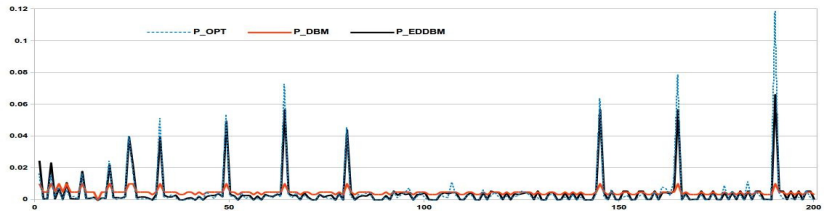
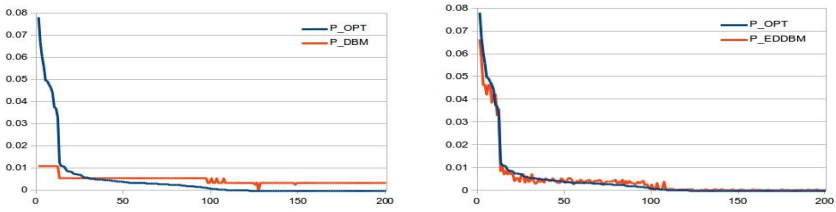


Fig. 1. Comparison of probabilities assigned by DBM, EDDBM and optimal model in a Barabasi Albert graph with $n=200$ and $k=5$



(a) DBM vs Optimal Model (b) EDDBM vs optimal model

Fig. 2. Comparison of probabilities assigned by DBM and EDDBM vs the optimal model in a random graph with $n = 200$ and $p = .05$

Comparison of Probabilities Assigned by DBM, EDDBM and Optimal Model. Plots in this subsection show that EDDBM generates probabilities very close to the optimal probabilities. These plots also compare EDDBM with DBM. The first plot in Fig. 1 is drawn for a synthetic scale free graph with $n = 200$ and $k = 5$. Here, the x-axis represents the nodes labelling and the y-axis represent the probabilities assigned by DBM, EDDBM and the optimal model. The next two plots in Fig. 2 are drawn for a random synthetic graph with $n = 200$ and $p = 0.05$. In both the plots in Fig. 2, the x-axis represents the nodes sorted in the order of their optimal probabilities and the y-axis represents the probabilities assigned by DBM / EDDBM and the optimal probabilities. In both the figures Fig. 1 and Fig.2, it is easy to observe that EDDBM is much better than DBM. We plot the average error in the computation of betweenness centrality using EDDBM when the number of sampled nodes were $T = X\%$ of the total number of nodes. We note that the average error reduces very sharply for smaller X. After $X = 10$, there is very small change in the average error. Thus, we have set $X = 10$ to get the experimental results in this paper. We also plot the average error in the computation of betweenness score in a graph with respect to the size of graph (number of nodes in that graph). We note that the average error in computation of betweenness score decreases with increase in the number of nodes in both cases (constant average degree and constant graph-density). The more details about the above plots are available in the full version on the paper.

Table 1. Average error in synthetic graphs

Instance	AD	DBM	EDDBM
Barabasi(500,2)	3.98	41.93%	14.6%
Barabasi(500,4)	7.94	39.82%	9.23%
Barabasi(1000,2)	3.99	37.81%	11.42%
Barabasi(1000,4)	7.97	34.65%	6.95%
Erdos_renyi(500, 0.016)	7.85	26.5%	4.34%
Erdos_renyi(500, 0.008)	3.99	22.5%	6.77%
watts_strogatz(500,4,.2)	4	18.41%	20.68%
watts_strogatz(500,6,.3)	6	21.12%	8.69%

Table 2. Average error in real graphs

Instance	n	m	GD	AD	DBM	EDDBM
Gset/G10 [10]	800	19176	.060	47.94	24.62%	3.88%
Gset/G14 [10]	800	4694	.0147	11.74	33.48%	7.13%
Florida Food Web [27, 4]	128	2075	.2553	32.42	38.90%	10.89%
Baydry Food Web [27, 4]	128	2106	.2591	32.91	35.22%	11.45%
Bai/rbsa480 [10]	480	16408	.1427	68.37	22.53%	4.99%
US Air lines [4]	332	2126	.0387	12.81	37.00%	14.50%
World Cities [26]	415	7518	.0875	36.41	24.04%	10.07%
Pajek/Roget [19, 4]	1022	4643	.0089	9.19	69.54%	32.19%
Pajek/SmaGri [4]	1024	4916	.0094	9.6	34.01%	9.48%
Pajek/GD06_C JAVA [4]	1538	7817	.0066	10.17	40.68%	12.08%
Pajek/Yeast [8, 4]	2284	6646	.0025	5.82	19.09%	5.59%
HB/bcsstk08 [10]	1071	5943	.0104	11.1	38.23%	14.14%
Arenas/Email [10]	1133	5451	.0085	9.62	29.14%	7.99%

5.4 Average Error in Graphs

In this section, we discuss and compare the results achieved by Algorithm 1 when it takes probabilities from our model (EDDBM) and Chehreghani's model (DBM) on some synthetic graphs and several small and moderate size real graphs. We have not compared our approach with the other uniform sampling approaches as Chehreghani's work has already surpassed them. We considered small and moderate size graphs as we have already showed that the performance of our approach increases in large size graphs.

Average Error in Synthetic Graphs. In this section, we will analyze the results over some synthetic graphs. The results are summarized in Table 1. The first column gives the description of graph considered. Second column contains the average degree (AD) of nodes in the instance graph. The last two columns contain the average error in computation of betweenness due to our model (EDDBM) and average error due to Chehreghani's model (DBM). We considered four scale free Barabasi Albert graphs, two random Erdos Renyi graphs and two small world Watts Strogatz graphs. We generated at least one sparse and one dense graph from each model. It is easy to observe that EDDBM performed much better than DBM. Our model performs better in denser graph than in sparser graphs. In the considered instances we reduced the error due to DBM by a maximum of 6.11 times and an average of 3.53 times.

Average Error in Real Graphs. This section presents and discusses the extensive simulation on real networks. The real networks were picked from [4,10,21]. After extracting the networks, we converted the networks into unweighted undirected networks. Then we removed multi-edges, self-loops and isolated nodes. The results obtained are summarized in the Table 2. The columns are in similar order as in the Table 1 except there are three more new column entries after the first column. The first new column contains the number of nodes (n), the second one lists the number of edges (m) in the corresponding networks and the third one lists the graph-density (GD) of the networks. EDDBM performed again better than DBM. In some considered data sets, we reduced the error by a maximum of more than 6.3 times and on a average reduction of 3.54 times.

The formulation of EDDBM is based on the analysis of random graphs. Random graphs do not posses high clustering coefficient and thus, this model does not perform well on the graphs with high clustering coefficient.

6 Conclusion and Further Work

In this paper, we considered the problem of approximating the betweenness score of a given node and provided a better feasible and practical solution to it than the existing one in literature. We presented the proof of concept of our technique by applying it to real world graphs as well as synthetic ones. To the best of our knowledge, it is the first attempt to theoretically derive a sampling function for betweenness computation. An interesting problem would be to tune our algorithm, so that, the clustering has no effect on the results. A more efficient approximation of the optimal sampling is another direction to look.

Acknowledgement. This project was partially supported by the SRF grant from the Indian Academy of Sciences.

References

1. Anthonisse, J.M.: The rush in a directed graph. Stichting Mathematisch Centrum. Mathematische Besliskunde (BN 9/71), 1–10 (1971)
2. Bader, D.A., Kintali, S., Madduri, K., Mihail, M.: Approximating betweenness centrality. In: Bonato, A., Chung, F.R.K. (eds.) WAW 2007. LNCS, vol. 4863, pp. 124–137. Springer, Heidelberg (2007)
3. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. Science 286(5439), 509–512 (1999)
4. Batagelj, V., Mrvar, A.: Pajek datasets (2006), `http://vlado.fmf.uni-lj.si/pub/networks/data`
5. Brandes, U.: A faster algorithm for betweenness centrality. The Journal of Mathematical Sociology 25(2), 163–177 (2001)
6. Brandes, U., Erlebach, T. (eds.): Network Analysis. LNCS, vol. 3418. Springer, Heidelberg (2005)
7. Brandes, U., Pich, C.: Centrality estimation in large networks. International Journal of Bifurcation and Chaos 17(07), 2303–2318 (2007)
8. Bu, D., Zhao, Y., Cai, L., Xue, H., Zhu, X., Lu, H., Zhang, J., Sun, S., Ling, L., Zhang, N., et al.: Topological structure analysis of the protein–protein interaction network in budding yeast. Nucleic Acids Research 31(9), 2443–2450 (2003)
9. Chehreghani, M.H.: An efficient algorithm for approximate betweenness centrality computation. The Computer Journal, page bxu003 (2014)
10. Davis, T.A., Hu, Y.: The university of florida sparse matrix collection. ACM Transactions on Mathematical Software (TOMS) 38(1), 1 (2011)
11. Erdos, P., Renyi, A.: On random graphs i. Publ. Math. Debrecen 6, 290–297 (1959)
12. Freeman, L.C.: A set of measures of centrality based on betweenness. Sociometry 40(1), 35–41 (1977)
13. Geisberger, R., Sanders, P., Schultes, D.: Better Approximation of Betweenness Centrality, ch. 8, pp. 90–100 (2008)

14. Gkorou, D., Pouwelse, J., Epema, D., Kielmann, T., van Kreveld, M., Niessen, W.: Efficient approximate computation of betweenness centrality. In: 16th Annual Conf. of the Advanced School for Computing and Imaging, ASCI 2010 (2010)
15. Goel, K., Singh, R.R., Iyengar, S., Sukrit: A faster algorithm to update betweenness centrality after node alteration. In: Bonato, A., Mitzenmacher, M., Prałat, P. (eds.) WAW 2013. LNCS, vol. 8305, pp. 170–184. Springer, Heidelberg (2013)
16. Green, O., McColl, R., Bader, D.A.: A fast algorithm for streaming betweenness centrality. In: 2012 International Conference on Privacy, Security, Risk and Trust (PASSAT) and 2012 International Confernece on Social Computing (SocialCom), pp. 11–20 (September 2012)
17. Kas, M., Wachs, M., Carley, K.M., Carley, L.R.: Incremental algorithm for updating betweenness centrality in dynamically growing networks. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM 2013, pp. 33–40. ACM, New York (2013)
18. Kintali, S.: Betweenness centrality: Algorithms and lower bounds. arXiv preprint arXiv:0809.1906 (2008)
19. Knuth, D.E.: The Stanford GraphBase: a platform for combinatorial computing, vol. 37. Addison-Wesley, Reading (1993)
20. Lee, M.-J., Lee, J., Park, J.Y., Choi, R.H., Chung, C.-W.: Qube: A quick algorithm for updating betweenness centrality. In: Proceedings of the 21st International Conference on World Wide Web, WWW 2012, pp. 351–360. ACM, New York (2012)
21. Leskovec, J.: Stanford large network dataset collection (2010)
22. Nasre, M., Pontecorvi, M., Ramachandran, V.: Betweenness centality–incremental and faster. arXiv preprint arXiv:1311.2147 (2013)
23. Newman, M.: Networks: An Introduction. Oxford University Press, Inc., New York (2010)
24. Riondato, M., Kornaropoulos, E.M.: Fast approximation of betweenness centrality through sampling. In: Proceedings of the 7th ACM International Conference on Web Search and Data Mining, pp. 413–422. ACM (2014)
25. Sariyüce, A.E., Saule, E., Kaya, K., Çatalyürek, Ü.V.: Shattering and compressing networks for betweenness centrality. In: SIAM Data Mining Conference (SDM). SIAM (2013)
26. Taylor, P.J.: World city network: a global urban analysis. Psychology Press (2004)
27. Ulanowicz, R.E., DeAngelis, D.L.: Network analysis of trophic dynamics in south florida ecosystems. In: FY97: The Florida Bay Ecosystem, pp. 20688–20038 (1998)
28. Van Der Hofstad, R.: Random graphs and complex networks (2009), `http://www.win.tue.nl/rhofstad/NotesRGCN.pdf`
29. Wang, X.: Deciding on the type of the degree distribution of a graph (network) from traceroute-like measurements (2011)
30. Watts, D.J., Strogatz, S.H.: Collective dynamics of small-world networks. Nature 393(6684), 440–442 (1998)

Sentiment Classification Analysis of Chinese Microblog Network

Xiaotian Wang, Chuang Zhang, and Ming Wu

Beijing University of Posts and Telecommunications, Beijing 100876, China
wangxiaotianbe@163.com, {zhangchuang,wuming}@bupt.edu.cn

Abstract. In recent years, more and more people begin to publish information on online social platforms like Sina Weibo. Via the facilities like posting tweets, retweeting tweets and making comments provided by Weibo service, users can easily express their feelings, giving opinions and make interactions with their friends in real time. Sentiment analysis of Weibo messages is important for the analysis of human sentiment. The characteristics of Chinese microblogs bring difficulty in sentiment classification. In this paper, an effective Chinese microblogs sentiment classification model based on Naive Bayes is proposed. Two strategies to do the three sentiment polarities classification are compared and the two-step strategy performs better than the one-step strategy.

Keywords: Sentiment classification, Sina Weibo, Machine learning, Social network.

1 Introduction

Social networks like Twitter and Weibo play vital roles in people daily online activities. [1] More and more Chinese people use Sina Weibo network to express their feelings and interact with friends. Besides common users, many organizations, such as enterprises, news agencies and government institutions use Sina Weibo to spread information.

Information on Weibo involves many areas. Online social networks are the reflection of the real networks of people. The contents of the microblogs indicate the sentiment of users. [2] Therefore, sentiment information is of much value. Analyzing the sentiment of specific microblogs in Weibo, we can obtain the opinions of participated users, which can be used to monitor or predict the trend of events, such as political elections prediction [3, 4, 5] and user's review prediction for products marketing [6].

However, some difficulties exist in the sentiment analysis of Chinese microblogs. The nature of Chinese language brings in the problem of words segmentation, which makes the sentiment analysis difficult. Meanwhile, Chinese microblogs with short-text nature and novelty of the new cyberspeak terms are quite different with normal written texts. [7] All of these increase the difficulty of sentiment analysis. The sentiment classification model needs to fit the Chinese microblogs characteristics.

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_12

In this work, we focus on the sentiment classification of Chinese microblogs. First, considering the characteristics of Weibo messages, an effective classification model based on Naive Bayes is proposed, which fits the characteristics of Chinese microblogs. Second, the two-step classification strategy is proposed for the classification process, that is subjective-objective classification in step 1 and positive-negative classification in step 2. Third, two classification strategies are compared. Two-step classification obtains better performance than that of one-step strategy.

2 Related Work

There has been much work on the sentiment analysis of texts of which two classes of methods are frequently used. The first one is the machine learning techniques. Pang B et al. [8] used Naive Bayes, Maximum Entropy and SVM to classify the film reviews into positive and negative which are reported to be superior to the human-produced baselines in document level. In their later study, they reviewed the machine learning techniques and methods which can be used in opinion-oriented information-seeking systems. [9] For Chinese sentiment analysis, Jun LI et al. [10] compared common machine learning methods in sentiment classification of Chinese hotel reviews. They found Naive Bayes performs best in their study and the feature schemes could affect the classification performance.

The other main method is the lexicon-based approach. Vasileios el al. [11] studied the log-linear regression model using the positive and negative semantic orientation lexicon to predict the polarity of adjectives. Turney and Littman [12] used SO-PMI and SO-ISA to calculate the semantic orientation of terms based on word co-occurrence in corpus. Isa Maks and Vossen [13] presented a lexicon model for the description of verbs, nouns and adjectives to do sentiment analysis and opinion mining.

Some researchers focus on sentiment analysis in Twitter or Weibo by considering the social networks. Tan et al. [14] found that the performance of sentiment classification can be improved significantly by incorporating social network information based on SVM. Tang et al. [15] proposed a graphical model to predict users' sentiment in the social network by studying how users' opinions are influenced by the people they follow on Tencent Weibo platform.

3 Sentiment Classification Model

Weibo messages have two important characteristics compared to the normal written texts. The first one is short-text nature. There is a 140-words limit in Sina Weibo networks. Because of the short attention of network activities, most Weibo messages are much shorter than 140 words, which is quite different to the normal written texts. This will result in the problems of information shortage and the lack of correlation structure in the sentences. The second characteristic is novelty. The words of Weibo messages come from daily online writing. The sentence structures of expression vary a lot and new online terms emerge continuously, such as new cyberspeak and emoji.

This requires the features of machine learning classification methods fit the development of the online expression.

To solve the problems, we built a machine learning model based on Naive Bayes to classify sentiment of Sina Weibo messages into three categories, namely positive, negative, and neutral. To construct a valid feature space, we used the Hownet sentiment dictionary and Ntusd sentiment dictionary as basic features. Since we merged two large dictionaries together, the dimension of feature space is too high compared with our handedly labeled training cases. And many of the words in the dictionary are rarely used in the contexts of tweets, which leads to sparse learning space.

In this work, ICTCLAS 2013 is used as the word segmentation tool to do the text processing. ICTCLAS is one of the most popular Chinese word segmentation tools which provide the functions of word segmentation, POS tagging and user defined dictionary. [16]

To reduce the dimension of feature space, we used a validation set to filter out infrequent features which are regarded as less important and could be noise that leads to sparsity problem. We used a corpus of 400K tweets from Sina Weibo to calculate the frequency of the words in our dictionary. Figure 1 gives the distribution of feature counts.

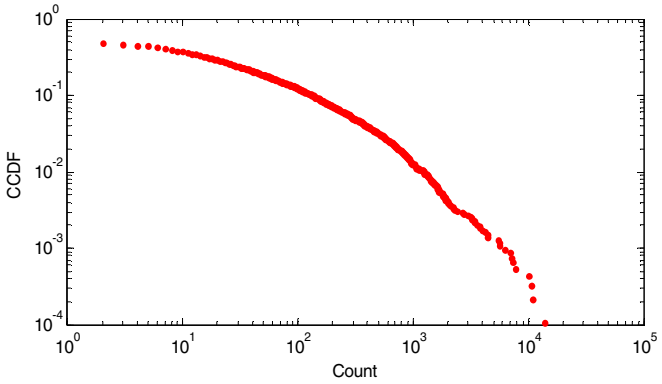


Fig. 1. Complementary cumulative distribution function (CCDF) of the feature frequency

As shown in Figure 1, approximately 90% of the words in feature space have a frequency lower than 100. In practice, feature words of frequency lower than 50 are filtered out.

As the language used in online communities is quite different from that of written text. One prominent characteristic is that tweets use more informal words with abbreviations and emoji. This lowers the performance of both segmentation of Chinese texts and feature selection. To solve this problem, we first crawled many of the popular online expressions to add to our dictionary. Then we extracted the unidentified words in the segmentation of Chinese texts and added them to our feature space. Also, the words of text expression for emoji defined by Weibo were added in feature space.

As a result, 2522 features are selected to build learning models.

4 Experiments and Analysis

To perform a supervised learning, 1584 microblog entries labeled manually are as the training set to train the machine learning model. The entries are randomly selected from the tweets posted for Weibo messages posted during 2013 and 2014.

Table 1. Sentiment distribution of training set

	Positive	Neutral	Negative	Total
Number	576	469	539	1584
Percentage	36.43%	29.61%	33.96%	100%

To do the classification, we proposed two classification strategies for the classification process:

Strategy 1: One-Step Classification. This strategy classifies messages into three polarities: positive, neutral and negative with one step.

10-fold cross-validation technique is utilized to train and test the Naive Bayes model. In 10-fold cross-validation, the original sample is randomly divided into ten subsamples. One subsample is the validation data used to test model, and the other nine subsamples are training data. Then repeat the process ten times with each of the subsample as the validation data. The final performance of the classification is the average scores of the results of the ten times. After the training and testing of 10-fold cross-validation, the classification model and results were obtained.

The precision, recall and F_1-measure are used as model evaluation metrics. F_1-measure is a measure of a test's accuracy considering both the precision and the recall. It is defined as follows:

$$F_1\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \tag{1}$$

The classification performance of Naive Bayes model in one-step strategy is shown in Table 2.

Table 2. Classification performance of one-step strategy

Sentiment	Precision	Recall	F_1-Measure
Positive	0.786	0.699	0.74
Neutral	0.709	0.728	0.719
Negative	0.755	0.809	0.781
Average	0.751	0.75	0.749

From the table we can see that the average F_1-measure is about 75%. The performance of most sentiment classification methods, which classify the texts into positive, negative and neutral, ranges from 0.7 to 0.8 for Chinese microblogs. Analyzing the

classification result, we find that microblog entries with weak sentiment polarity and entries with only a few words are easier to be classified into incorrect polarity.

Strategy 2: Two-Step Classification. This strategy consists of two steps: step-1, the entries are classified into subjective and objective. The objective entries are messages with neutral sentiment. They may be the statements of the facts or the tweets which just express the neutral sentiment without bias. However, the subjective are the entries with positive or negative sentiment. In step-2, the subjective entries are classified into positive and negative.

In **step-1**, the 1584 microblog entries are labeled as objective or subjective, where neutral entries are labeled as objective, positive and negative are labeled as subjective. 10-fold cross-validation technique is utilized to train and test the first-step Naive Bayes model. The performance is shown in Table 3.

Table 3. Classification performance of step one for two-step strategy

Sentiment	Precision	Recall	F_1-Measure
Subjective	0.878	0.901	0.889
Objective	0.893	0.868	0.88
Average	0.885	0.885	0.885

We can see that the performance is near to 0.9. The model can effectively classify microblogs in two polarities.

In **step-2**, the subjective entries are labeled as negative or positive. After 10-fold cross-validation, the result of Naive Bayes model is shown in Table 4.

Table 4. Classification performance of step two for two-step strategy

Sentiment	Precision	Recall	F_1-Measure
Positive	0.831	0.92	0.873
Negative	0.91	0.812	0.858
Average	0.871	0.866	0.866

We can see that the classification of positive and negative is a little higher than 0.85.

To compare the performance, we calculate the final result of **Strategy 2** as Figure 2 shows:

Except that negative polarity classification is almost same, we can see that the performance of **Strategy 2** is much higher than that of **Strategy 1** in positive and neutral polarity classification. The average F_1-measure of **Strategy 2** is 0.75 higher than that of **Strategy 1.** This proves that two-step strategy performs better to classify Chinese microblogs entries than one-step strategy.

On the whole Naive Bayes is an effective method to do the sentiment classification of Chinese microblog messages. And two-step strategy is the better choice when doing the sentiment classification.

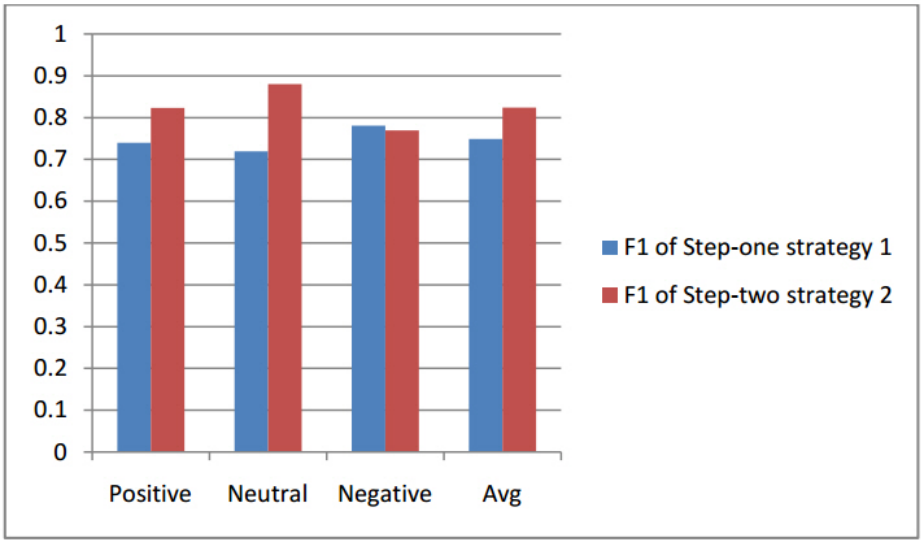


Fig. 2. Performance comparison between **Strategy 1** and **Strategy 2**. The vertical axis is the F_1-measure of for polarities.

5 Conclusions and Future Work

This paper focuses on the sentiment classification analysis of Chinese microblog network. Considering the characteristics of Weibo messages, we propose a sentiment classification model based on Naive Bayes. This model obtains good classification performance by fitting the characteristics of Chinese. A two-step strategy for the classification process is proposed. That is subjective-objective classification in step 1 and positive-negative classification in step 2. Compared to the one-step classification strategy, two-step classification performs better with the measurement of precision, recall and F1-measure.

In future work, more features of microblogs can be introduced to improve the classification performance. The structure of networks and user profile may influence the sentiment expression of users, these features can be utilized. And the analysis of Chinese language context can help to adjust the machine learning model.

Acknowledgement. The authors gratefully acknowledges the generous support from the National Science Foundation of China (61273217) and Advanced Intelligence and Network Service, Chinese 111 program(B08004).

References

1. Huberman, B.A., Romero, D.M., Wu, F.: Social networks that matter: Twitter under the microscope. arXiv preprint arXiv:0812.1045 (2008)
2. Gao, Q., Abel, F., Houben, G.-J., Yu, Y.: A comparative study of users' microblogging behavior on Sina Weibo and Twitter. In: Masthoff, J., Mobasher, B., Desmarais, M.C., Nkambou, R. (eds.) UMAP 2012. LNCS, vol. 7379, pp. 88–101. Springer, Heidelberg (2012)
3. Pak, A., Paroubek, P.: Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: LREC (2012)
4. Tumasjan, A., Sprenger, T.O., Sandner, P.G., et al.: Predicting Elections with Twitter: What 140 Characters Reveal about Political Sentiment. In: ICWSM 2010, pp. 178–185 (2010)
5. Choy, M., Cheong, M.L.F., Laik, M.N., et al.: A sentiment analysis of Singapore Presidential Election, using Twitter data with census correction. arXiv preprint arXiv 1108: 5520 (2011)
6. O'Connor, B., Balasubramanyan, R., Routledge, B.R., et al.: From tweets to polls: Linking text sentiment to public opinion time series. ICWSM 11, 122–129 (2010)
7. Lu, W., Wang, Y.: Review of Chinese text sentiment analysis. Application Research of Computers 29(6) (2012)
8. Pang, B., Lee, L., Vaithyanathan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, vol. 10, pp. 79–86. Association for Computational Linguistics (2002)
9. Pang, B., Lee, L.: Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2(1-2), 1–135 (2008)
10. Li, J., Sun, M.: Experimental study on sentiment classification of Chinese review using machine learning techniques. In: International Conference on IEEE Natural Language Processing and Knowledge Engineering, pp. 393–400 (2007)
11. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, pp. 174–181. Association for Computational Linguistics (1997)
12. Turney, P.D., Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS) 21(4), 315–346 (2003)
13. Maks, I., Vossen, P.: A lexicon model for deep sentiment analysis and opinion mining applications. Decision Support Systems 53(4), 680–688 (2012)
14. Tan, C., Lee, L., Tang, J., et al.: User-level sentiment analysis incorporating social networks. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1397–1405. ACM (2011)
15. Tang, J., Fong, A.C.M.: Sentiment diffusion in large scale social networks. In: 2013 IEEE International Conference on IEEE Consumer Electronics (ICCE), pp. 244–245 (2013)
16. Zhang, H.: NIPIR/ICTCLAS (2014), `http://ictclas.nlpir.org/` (accessed September 12, 2014)

Techniques for Brain Functional Connectivity Analysis from High Resolution Imaging

A.C. Leitão[1], A.P. Francisco[1], R. Abreu[2], S. Nunes[2], J. Rodrigues[2], P. Figueiredo[2], L.L. Wald[3], M. Bianciardi[3], and L.M. Silveira[1]

[1] INESC-ID/ Instituto Superior Técnico, Universidade de Lisboa, Portugal
{andrechambel,aplf,lms}@tecnico.ulisboa.pt
[2] ISR/ Instituto Superior Técnico, Universidade de Lisboa, Portugal
{rodolfo.abreu,sandro.nunes,juliana.rodrigues,
patricia.figueiredo}@tecnico.ulisboa.pt
[3] Athinoula A. Martinos Center for Biomedical Imaging, Charlestown, MA USA
{wald,martab}@nmr.mgh.harvard.edu

Abstract. Several methods have previously been proposed for mapping and enabling the understanding of the brain's organization. A widely used class of such methods consists in reconstructing brain functional connectivity networks from imaging data, such as fMRI data, which is then analysed with appropriate graph theory algorithms. If the imaging datasets are acquired at high resolution, the complexity of the problem both in spatial as well as temporal terms becomes very high. In this work, brain images were acquired using high-field scanners that produce very high resolution fMRI datasets. In order to address the resulting complexity issues, we developed a tool that is able to reconstruct the brain connectivity network from the high resolution images and analyse it in terms of the network's information flowing efficiency and also of the network's organization in functional modules. We were able to see that, although the networks are very complex, there is an apparent underlying organization. The corresponding structure allows the information to flow from one point to another in a very efficient manner. We were also able to see that these networks have a modular structure, which is in accordance with previous findings.

Keywords: functional Magnetic Resonance Imaging, brain, network mining, high resolution, graph theory, functional connectivity.

1 Introduction

The human brain is known to be the most complex organ of the human body. Over time its study has attracted considerable attention and researchers have come up with multiple ways to analyse it. One possible way to do that is to build and analyse the brain functional connectivity (BFC) network from the data provided by functional Magnetic Resonance Imaging (fMRI). This BFC network allows to study the brain using standard graph theory algorithms.

This work intends to address this analysis and mapping problem, by starting with high resolution resting state fMRIs obtained from experimental 7T machine scans, extracting from them the BFC network and applying network mining techniques to analyse them. Having a high resolution image of the brain we hope will make it possible to

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_13

extract a more accurate and more detailed network. However, the increase in data size is also a problem as the amount of data can easily be hundreds of times larger than usual fMRI. Therefore one of the challenges of this work is to find efficient ways to build, represent and analyse these networks.

fMRI

The fMRI is one of the most widely used brain imaging techniques. It relies on the magnetic properties of the hemoglobin measuring the Blood-Oxygen Level-Dependent (BOLD) signal. The brain activity is measured based on the changes in the blood flow and on the fact that the blood flow in the brain is strongly correlated with neuronal activity [1]. The BOLD signal will be more intense in the areas of the brain that are active at a given time. Thus, the fMRI will provide a spatial map of the 3D brain where each volume division (voxel) will have associated to it a different BOLD signal fluctuation. This allows us to know how active that specific volume unit of the brain was through the time course of the test.

Using a stronger magnetic field makes it possible to get higher quality spatial resolution. That property is consequentially reflected on the size and number of voxels, i.e., higher resolution yields more and smaller voxels.

Functional Connectivity

The most commonly accepted definition of functional connectivity describes it as the temporal correlation between spatially remote neurophysiological events [2, 3]. In other words the brain functional connectivity network will give us an insight on how the different brain regions are functionally related. Several methods may be used to evaluate functional connectivity. The evaluated functional connectivity may differ depending on whether the complete time series is used or just part of it and also on whether one uses the data from a single subject or the data obtained by averaging across subjects. All these different approaches may yield different functional connectivity networks even though the same datasets are being used. The basic elements of this network will be the voxels the information about voxels functional connectivity will determine if they are connected or not.

Graph Theory

To perform all the network mining analysis that are required to obtain the previously described functional network, we resorted to graph theory. A generic graph G consists of a set of nodes, or vertices, (V) connected to each other through a set of edges (E), i.e., G = (V,E). These connections can either exist or not based on the pairwise relation between the nodes.

Often a graph that models the brain functional network can have as vertices the brain's regions of interest (ROI), that are usually known from a brain atlas, which is a three-dimensional map of the human brain. If a more detailed analysis is desired the vertices can be the voxels that come directly from the fMRI.

In graph theory there are several metrics that can be computed for a given graph. In order to understand them there are some baseline concepts that need to be defined first. One simple concept is that of degree of a vertex, which is the number of edges that are connected to it. Another important concept is that of a path, that is the sequence of vertices and edges that are crossed to get from a vertex of the graph to another. The length of a path can be measured by the number of edges that are crossed and this yields the concept of distance between two vertices, as the shortest of all paths that connect them.

2 Methods

In order to analyse the BFC networks using graph theory concepts several metrics were used to give us an insight on the network's structure such as the degree distribution function, clustering coefficient, modularity and small world coefficient, that are defined in [4, 5, 6, 7].

Building a Network from fMRI Data

In the BFC network each voxel will be a vertex and their pairwise functional connectivity will be an edge. The most commonly used way to determine if there is an edge between two vertices is to measure the correlation between them [8]. Having the correlation between all pairs of voxels a threshold is set and only pairs with a correlation above that level are accepted as functionally connected.

The amount of data that we are dealing with when we compute a matrix that correlates every pair of voxels is a challenging problem, therefore we are going to do some pre-processing before starting the computation. The most obvious step to do first is, on each slice, to only consider the voxels that actually belong to the brain, i.e., voxels that do not have any BOLD signal are no considered. Additionally, we also want to avoid making the computation of the whole correlation matrix at once, an instead do it in chunks. Each of those chunks is then processed to extract the pairs of voxels whose correlation is above the chosen threshold. Those are stored and all other data is discarded. The procedure is as follows: Initially the data matrix is divided in chunks of equal size where each matrix Y_i has dimensions $m \times t$, with m being the number of voxels and t the number of different time points (Equation 1); Then each of the chunks is correlated with all the other chunks yielding the correlation matrix that is formed by the sub-matrices of the chunks pairwise correlations (Equation 2). All of these sub-matrices will have the same size $m \times m$, m being the number of voxels that are present in each of the data chunks Y_i.

$$Y = \begin{bmatrix} Y_1 & Y_2 & \dots & Y_n \end{bmatrix} \qquad (1)$$

$$R = \begin{bmatrix} R_{11} & R_{12} & \dots & R_{1n} \\ R_{21} & \ddots & & \\ \vdots & & \ddots & \\ R_{n1} & R_{n2} & \dots & R_{nn} \end{bmatrix} \qquad (2)$$

Community Detection

We performed a modularity analysis that was intended to find separate modules on the network. The best partition is the one that concentrates more edge density within its modules. The optimization of modularity measure defined by Newman [5] is a hard problem, thus we must rely on efficient greedy algorithms. Even though we are not able to achieve optimal partitionings, we obtain reasonably good ones in linear time. To do so we used a parallel algorithm developed by Boldi et al. [9] called Layered Label Propagation (LLP) that is based on well known label propagation algorithms but with the ability to tune the vertex resistance to change label leading to a hierarchical clustering. The conclusions regarding the different functional modules drawn from these datasets were then compared with a different state of the art analysis. The most commonly used approach to analyse resting state fMRI is the Independent Component Analysis (ICA), and we will therefore be interested in comparing with it. These Independent Components (ICs) can then be compared with our results, to check for their validity. This validation was made by measuring the overlap between the ICs and the modules found by the community detection algorithm. The ICA analysis was conducted using FSL version 5.0.6 with MELODIC version 3.14 [10] generating 20 ICs.

3 Results

The resting-state fMRI datasets were collected from a group of six healthy volunteers on a 7T Siemens machine yielding data with 1.1mm^3 isotropic voxels. As the size of the brain varies from one person to another, each of the subjects has a different number of nodes in their BFC network. These are presented in Table 1.

Table 1. Total number of nodes in the BFC network of each subject

Subject	Slices	Number of nodes
1	144	1 365 082
2	120	1 080 702
3	133	1 282 836
4	138	1 305 160
5	145	1 365 120
6	135	1 262 244
Average	136	1 276 857

Table 2. Number of created edges for each subject using different thresholds

	Edge average	Edge density
0.40	668 989 847	8.2067e-04
0.45	302 385 481	3.7095e-04
0.50	136 909 022	1.6795e-04
0.55	61 017 717	7.4852e-05
0.60	26 350 364	3.2325e-05
0.65	10 887 184	1.3356e-05
0.70	4 263 328	5.2300e-06

For each subject, a different correlation threshold yields a different BFC network. This difference can be easily observed when computing the number of edges. The lower the threshold the more voxel pairs are considered as functionally connected thus resulting in a higher number of edges for the lower thresholds.

As one can observe from Table 2, for all the thresholds the edge density is very low, which makes the network sparse. This was an expected result since in previous state of the art works all the BFC networks were found to be sparse. [11, 7]

Connected Components and Degree Distribution

In order to choose an appropriate threshold it is required to check how much information about the network is lost when going from a low threshold to a higher one. In order to evaluate this, the size of the giant connected component of the network was computed and compared with the total number of nodes in the network. The results regarding these computations are presented in Table 3.

Table 3. Percentage of the total nodes that are in the giant component of the network

	Subject					
T	1	2	3	4	5	6
0.40	100%	100%	100%	100%	100%	100%
0.45	98%	100%	99%	100%	99%	100%
0.50	77%	100%	91%	100%	88%	99%
0.55	53%	96%	58%	99%	58%	92%
0.60	37%	77%	44%	94%	35%	71%
0.65	26%	48%	32%	74%	23%	55%
0.70	16%	28%	22%	46%	14%	42%

From the results presented it is easy to conclude that if the threshold is too high then the network loses its connectivity and the amount of information lost is also too high. We could infer that, on average, for a correlation threshold between 0.4 and 0.5 little information seems to be lost, whereas above that we will start to have is a significant loss of information.

Regarding the vertex degree distribution for the BFC networks it is also dependent on the chosen correlation threshold. To check if our BFC networks exhibit properties similar to the ones already studied in other state of the art works, their degree distribution should follow a power law, with an exponent between 2 and 3. For each subject and for each threshold the degree distribution function was computed and fitted with a power law and the results of the power law exponent that fits each degree distribution function are shown on Table 4.

Table 4. Value of the exponent from the fitting function of the degree distribution

	Subject					
T	1	2	3	4	5	6
0.40	2.09	2.04	2.15	2.16	2.04	2.28
0.45	2.00	1.92	2.04	2.06	1.97	2.18
0.50	1.92	1.83	1.95	1.95	1.89	2.09
0.55	1.85	1.66	1.86	1.84	1.77	1.96
0.60	1.80	1.64	1.79	1.72	1.77	1.88
0.65	1.77	1.63	1.74	1.66	1.75	1.70
0.70	1.73	1.60	1.66	1.59	1.73	1.65

It is possible to see from Table 4 that the networks whose degree distribution is closer to the ones reported in other state of the art works are the ones corresponding to lower thresholds. This is expected as the edge density for the networks with higher correlation thresholds is very low.

Small Worldness

Based on the obtained results, the only networks that were considered for further analysis were the ones obtained with a correlation threshold between 0.4 and 0.5.

To prove the small-world topology we need to compute the minimum average path in all the BFC and respective random equivalent networks, and also the clustering coefficient for both cases. With this information we are now able to compute the λ and γ coefficients as presented in [7]. All the results regarding these computations are presented on Table 5 and Table 6.

Table 5. Average characteristic path for the BFC networks, their respective random equivalents and value of the λ coefficient

T	BFC	rand	λ
0.4	4.113	3.251	1.265
0.45	5.650	3.509	1.610
0.5	7.487	4.698	1.498

Table 6. Average clustering coefficient of the BFC networks, their respective random equivalent networks and value of the γ coefficient

T	BFC	rand	γ
0.4	0.213	0.053	4.102
0.45	0.197	0.042	4.690
0.5	0.173	0.039	4.436

As one can see from Table 5 the minimum average path of all the BFC networks is almost as low as the one from their random equivalents, which is exactly what usually happens in small-world networks [4]. This is an important property of the networks that have a small-world topology, it is possible to go from any vertex to any other with a small number of steps. Regarding the clustering coefficient results, presented in Table 6, we were able to see that these networks have a higher cluster coefficient than its random equivalent. From the previous results it is possible to estimate the σ coefficient, presented in [7] with the results shown in Table 7.

Table 7. Average small-worldness coefficient of all the BFC networks

T	γ	λ	σ
0.4	4.102	1.265	3.243
0.45	4.690	1.610	2.913
0.5	4.436	1.498	2.961

With these final results of the σ coefficient we are now able to postulate that all the studied BFC networks have a small-world topology, since for all of them the σ coefficient is higher than 1, which, as shown in the work of van den Heuvel et al. [7], is enough to prove our assertion of small-worldness.

Community Detection

For each of the three different BFC networks of all the subjects a community detection algorithm was applied with the purpose of finding functional modules of the brain. In order to validate these results, the found clusters were compared with the resulting data provided by independent component analysis (ICA).

For the graph cluster analysis only the six major modules were represented because on average the other modules were very small when compared with the average size of the IC. There was some significant overlap between some modules found by LLP and IC found by ICA, with some of these values up to 90%. This is a very relevant result as it proves that our analysis made with the LLP algorithm has very likely found relevant modules of the brain because it is supported by the results of ICA. It was also possible to see that there is a significant overlap of the modules with the ICs in almost every subjects' networks at all three chosen threshold levels; however some thresholds had better results than others. In Figure 1 three modules from the BFC network of subject 3 are represented and also the ICs where those modules are contained. As can be seen, both images are very similar, with the modules that have a higher overlap with the IC being the ones that seem almost the same.

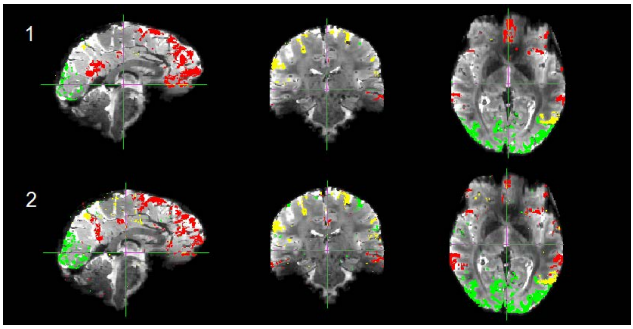


Fig. 1. 1 - Three modules found with LLP for subject 3 at a correlation threshold of 0.45; 2 - Three IC found with ICA for subject 3 at a correlation threshold of 0.45

After measuring the overlap between the modules found with LLP and the ICA, we computed their normalized mutual information (NMI). The results showed that almost all the modules and ICs that were chosen have an NMI between 0.3 and 0.5. This may seem an unexpected result because of the high percentage of vertices from the modules that are contained in the ICs. However it is important to stress the fact that the size of the modules sometimes is quite different from the size of the ICs, which means that although the majority of the vertices from the module overlaps the IC there is still a number of vertices from the IC that is outside of that given module.

4 Conclusions and Future Work

The results of our work were very interesting, as far as reconstructing the BFC network from high resolution fMRIs is concerned, because to the best of our knowledge no tool has been presented that allows a reconstruction of such high resolution networks. Furthermore, our results also showed that the structural properties of the networks are similar to the ones found in low resolution networks. Thus, even at high resolution, we found that there is an evident ability of the brain's network to flow information in a very efficient way.

Regarding the BFC network analysis, more advanced metrics can be computed and more detailed modularity analysis can be made. For instance, for each cluster that we found another modularity analysis can be performed and checked for clusters within the clusters, i.e., check for some hierarchical information.

It is also important to stress that better and more advanced methods to pre-process the data will yield more interesting and accurate the results. However, all these techniques are also complex specially in datasets that cover areas such as the brainstem, that are very exposed to noise.

Acknowledgement. This work was partially supported through FCT, Fundação para a Ciência e Tecnologia, under projects HiFi-MRI, PTDC/EEI-ELC/3246/2012 and PEst-OE/EEI/LA0021/2013, and also by the National Institutes of Health NIH-NIBIB P41EB015896 grant.

References

[1] Huettel, S.A., Song, A.W., McCarthy, G.: Functional magnetic resonance imaging, vol. 1. Sinauer Associates, Sunderland (2004)
[2] Friston, K.J.: Human Brain Mapping 2(1-2), 56 (1994)
[3] Horwitz, B.: Neuroimage 19(2), 466 (2003)
[4] Watts, D.J., Strogatz, S.H.: Nature 393(6684), 440 (1998)
[5] Newman, M.E., Girvan, M.: Physical Review E 69(2), 026113 (2004)
[6] Bigg, N.L., Lloyd, E.K., Wilson, R.J.: Graph Theory: 1736-1936. Oxford University Press (1976)
[7] van den Heuvel, M.P., Stam, C.J., et al.: Neuroimage 43(3), 528 (2008)
[8] Smith, S.M.: Neuroimage 62(2), 1257 (2012)
[9] Boldi, P., Rosa, M., Santini, M., Vigna, S.: In: Proceedings of the 20th International Conference on World Wide Web, pp. 587–596. ACM (2011)
[10] Beckmann, C.F., Smith, S.M.: IEEE Transactions on Medical Imaging 23(2), 137 (2004)
[11] Ferrarini, L., Veer, I.M., et al.: Human Brain Mapping 30(7), 2220 (2009)

A Two-Parameter Method to Characterize the Network Reliability for Diffusive Processes

Madhurima Nath[1,2], Stephen Eubank[1,3], Mina Youssef[1,⋆], Yasamin Khorramzadeh[1,2], and Shahir Mowlaei[1,2]

[1] Network Dynamics and Simulation Science Laboratory, Virginia Bioinformatics Institute, Virginia Tech, Blacksburg, Virginia 24061, USA
[2] Department of Physics, Virginia Tech, Blacksburg, Virginia 24061, USA
[3] Department of Population Health Sciences, Virginia Tech, Blacksburg, Virginia 24061, USA
{nmaddy,seubank,myoussef,yasi,shahir}@vbi.vt.edu

Abstract. We introduce a new method to characterize the network reliability polynomial of graphs – and hence the graph itself – using only a few parameters. Exact evaluation of the reliability polynomial is almost impossible for large graphs; estimating the polynomial's coefficients is feasible but requires significant computation. Furthermore, the information required to specify the polynomial scales with the size of the graph. Thus, we aim to develop a way to characterize the polynomial well with as few parameters as possible. We show that the error function provides a two-parameter family of functions that can closely reproduce reliability polynomials of both random graphs and synthetic social networks. These parameter values can be used as statistics for characterizing the structure of entire networks in ways that are sensitive to dynamical properties of interest.

Keywords: Network reliability, Error function, synthetic social networks.

1 Introduction

1.1 Motivation

It has been more than 50 years since Moore and Shannon introduced the network reliability polynomial to study the performance of electronic circuits with "crummy" relays [1]. Since then, the concept has been widely applied in designing reliable circuits and other networks delivering commodities between source and destination locations. Early studies showed the effect of network topology on the overall performance of the network for simple commodity flow between a source vertex and a sink vertex [2,3]. This problem is well-known as the *Two-Terminal reliability rule*. Another common performance measure is the probability that a

⋆ Corresponding author.

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_14

randomly selected set of edges connects all the vertices of the original graph, often referred to as the *All-Terminal reliability rule*. Furthermore, the *K-Terminal reliability rule* – the probability that a randomly selected subgraph contains a predefined set of vertices of size K – has been also studied. For a comprehensive review of the reliability polynomial, we refer the reader to the book by C. J. Colbourn [4].

Network reliability polynomials are not limited to Two-Terminal reliability, K-Terminal reliability or All-Terminal reliability rules. There are many features of percolation processes that the network reliability polynomial can reflect. In previous studies, the concept of reliability polynomial was successfully applied to study the spread of the infectious diseases in social networks [5,6].

The classical concept of network reliability provides a rich theoretical basis, supported by computational estimation procedures, to study the effect of structural properties on the diffusion of dynamics. Although evaluation of the reliability polynomial coefficients is usually intractable and its complexity is #-P hard [4], estimating the coefficients to within a practically important confidence interval is feasible. This paper aims at shrinking the wide gap between theoretical analysis of reliability problems and our ability to apply the conceptual framework to practical problems for large and non-trivial graphs. The estimation procedure relies on the random selection of subgraphs from the main graph under study. The reliability rule, which is chosen based on the dynamical features of interest, is applied to every subgraph to determine whether it exhibits the desired feature. For example, in this paper, the feature we are interested in is the probability that a certain fraction α (the "attack rate")of the population will be infected during an outbreak of disease, as a function of its person-to-person transmissibility.

The reliability polynomial describes the system's behavior. We would like to use it to characterize the system itself, and to that end, we need a way to summarize the information it contains. We can take advantage of the fact that, for rules that satisfy a simple criterion, the reliability polynomials are monotonic increasing sigmoidal functions from the interval $[0, 1]$ to itself. This suggests representing the polynomial as the cumulative distribution function of a continuous probability density.

1.2 Contribution

In this paper, we represent the network reliability polynomial in terms of well-known two-parameter functions. Here, we test two functions, the error function and the binomial cumulative distribution function. We fit reliability polynomials for several random graphs and synthetic social networks to these functions, note the values of the best-fit parameters, and evaluate the goodness of fit using the statistical coefficient of determination. The error function provides better fits than the binomial CDF fits and provides a very close fit to all the examples. Thus the two parameters of the error function, the mean and deviation of the corresponding Gaussian, were sufficient to characterize the reliability polynomial.

We observe that, for random graphs, the values of these two parameters are weakly correlated with the size of the graph. The values also depend on the reliability rule. Lastly we exploit this method to characterize differences among synthetic social networks for the New River Valley in Virginia, Mexico City, Sierra Leone and Liberia [7]. We conclude that we can reconstruct the network's reliability using just a few parameters.

The paper is organized as follows: Section 2 introduces the definition of network reliability and reliability rules. Section 3 elaborates on fitting the reliability polynomial to two parameterized functions. The numerical evaluations are described in Section 4. Finally, the conclusions are discussed in Section 5.

2 Network Reliability Polynomial

Moore and Shannon introduced the concept of the network reliability polynomial in the 1950's to evaluate the performance of electrical circuits composed of crummy relays. Given that every relay has a probability of failure, Moore and Shannon showed that the probability the circuit functioned as desired could be expressed as a polynomial. In addition, they evaluated the circuit reliability given that the relays are connected in series, in parallel, and in certain combinations of series and parallel. In this paper, we use the reliability concept to analyze social networks. In particular, we tie the concept of network reliability to network epidemiology by evaluating the probability of obtaining a given attack rate as a function of transmissibility.

2.1 Mathematical Definition

Given a graph composed of N vertices and E edges and a criterion that clearly defines the acceptance or the rejection of a subgraph–represented as the *reliability rule r:* $r(g) \in \{0, 1\}$–in a binary form, we introduce a *damage model* that assigns a probability to each subgraph. The network reliability is then:

$$R_G(\boldsymbol{x}) \equiv \sum_{g \subset G} r(g) p_{\boldsymbol{x}}(g) \tag{1}$$

where g is a subgraph, $r(g)$ is 1 if the subgraph g is accepted by the rule r, and $p_{\boldsymbol{x}}(g)$ is the probability to obtain the subgraph g under the damage model. In this paper, the damage model includes each edge with probability x, corresponding to bond percolation. The probability of obtaining a subgraph with k edges is $x^k(1-x)^{E-k}$. We denote the number of subgraphs with k edges that are accepted by the reliability rule $r(g)$ as R_k. The network reliability can be written as a polynomial:

$$R_G(x) = \sum_{k=0}^{k=E} R_k x^k (1-x)^{E-k}. \tag{2}$$

The term R_k is called the reliability coefficient, and its computation is, in general, intractable. However, we know that there are $\binom{E}{k}$ subgraphs with k

edges in the graph, and that some fraction between 0 and 1 of them is accepted by the reliability rule. Therefore, the reliability coefficient can be written as follows:

$$R_k = P_k \binom{E}{k} \tag{3}$$

where P_k is the fraction of with k edges that is accepted. Hence:

$$R_G(x) = \sum_{k=0}^{k=E} P_k \binom{E}{k} x^k (1-x)^{E-k}. \tag{4}$$

Computing the coefficients P_k is straightforward. Simply select a subgraph with k edges randomly and evaluate the reliability rule. The estimate for P_k is the number of accepted subgraphs divided by the total number of subgraphs sampled. The random selection of subgraphs is repeated until the sampling error for P_k is within the desired confidence interval. The number of edges in the smallest accepted subgraph is called k_{min}, while the number of edges in the largest unaccepted subgraph is called k_{max} [8]. Thus the P_k curve has values between 0 and 1 between k_{min} and k_{max}, respectively. The reliability $R(x)$ is a smoothed version of P_k, as can be seen from Equation 4, where $\binom{E}{k} x^k (1-x)^{E-k}$ plays the role of a sharply peaked smoothing kernel.

We use the following reliability rule: *a graph g is accepted if and only if the mean square size of connected components in g is greater than* αN. This rule creates a mapping between an interesting epidemiological problem and the reliability polynomial as follows: x represents the transmission probability, α represents the attack rate, and $R(x)$ represents the probability that the attack rate is at least α. We denote this rule as $ExpX - \alpha$.

The motivation of this work was to characterize the reliability polynomial $R(x)$ using a small set of parameters. Since $R(x)$ has the properties of a cumulative distribution function (CDF), we propose to fit the P_k values to the binomial CDF and the error function.

3 Two-Parameter Characterization of Network Reliability

3.1 Binomial CDF Method

The cumulative distribution function (CDF) of a binomial distribution is given as

$$CDF(k) = \sum_{m=0}^{k} \binom{N}{m} p^m (1-p)^{N-m}. \tag{5}$$

$R(x)$ has a similar form (Eqn. 4). Thus, we fit the right hand side of Eqn.(5) to the P_k of Eqn.(4) and plot it with respect to the k estimates P_k by taking p as a parameter. They are truncated between the k_{min} and k_{max}. The k data points are re-scaled to give the values of m such that it runs from 0 to N as

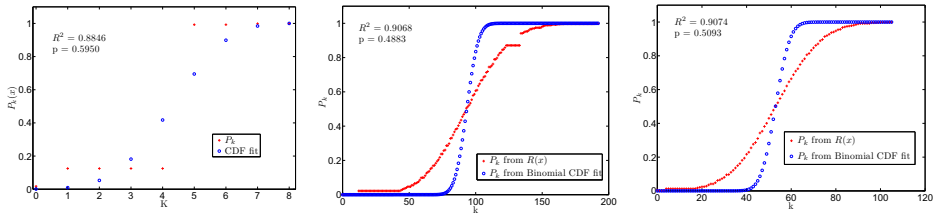


Fig. 1. Plot of P_k and binomial CDF fit with k^* for (a)$N = 20$, $M = 50$ (left panel), (b)$N = 2000$, $M = 50,000$ (middle panel) and (c)$N = 153,036$, $M = 4,152,739$ (right panel)

integers with an increment of 1. That is it runs from 0 to the total number of data points. Here, N is chosen to be the maximum of m. We find the value of $p \in [0,1]$ for which the binomial CDF best fits the P_k values. We use the coefficient of determination R^2 as a measure of goodness-of-fit as follows:

$$R^2 = 1 - \sum_i \frac{(y_i - y_{fit})^2}{(y_i - y_{mean})^2} \tag{6}$$

where y_i are the data points, y_{mean} is the mean of the data points and y_{fit} are the fitted data. The best fit is obtained when R^2 is closest to 1.

Figure 1 shows the binomial CDF fit to three sets of random networks generated by choosing a specific number of edges M uniformly at random over a specific number of vertices N. These are (a)$N = 20$, $M = 50$, (b)$N = 2000$, $M = 50,000$ and (c)$N = 153036$, $M = 4,152,739$. As the size of the graph increases the fit becomes better as observed from the R^2 value, which increases from 0.8846 to 0.9074.

3.2 Error Function Method

We show the fit of $R(x)$ to the error function:

$$erf(X) = \frac{1}{a}\int_0^X e^{-\left(\frac{t+b}{a}\right)^2} dt \tag{7}$$

Here, a changes the width of the underlying Gaussian and is related to the variance whereas b shifts the position of the mean of the Gaussian. The error function is defined between -1 and 1 for positive and negative values of X. The $R(x)$ and P_k have values only between 0 and 1 and x lies between 0 and 1. Thus, we rescale the error function such that both of them are in the same range. For this we fit $R(x)$ and P_k to $\frac{1}{2}(erf(ax - b) + 1)$. Also, we normalize the x values to x^* given by

$$x^* = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{8}$$

Table 1. Comparison between fits to the error function method and the Binomial CDF

Size of Graph	Two-parameter method	Binomial CDF method
$N = 20,\ M = 50$	0.9830	0.8846
$N = 100,\ M = 450$	0.9508	0.8808
$N = 500,\ M = 35750$	0.9976	0.8983
$N = 2000,\ M = 50,000$	0.9993	0.9068
$N = 10,000,\ M = 40,000$	0.9916	0.9627
$N = 35,000,\ M = 750,000$	0.9975	0.9142
$N = 125,000,\ M = 500,000$	0.9986	0.8843
$N = 153,036,\ M = 4,152,739$	0.9999	0.9075

$R(x)$ shows a sharp transition when plotted against x for large systems. This means that the values of x_{min} and x_{max} are closer for larger systems compared to a small size network. To look at the behaviour of $R(x)$ in the region of the sharp transition we re-scale the axis from x to x^*.

4 Numerical Evaluation

We generate random Erdős-Rényi $G(N, M)$ graphs each having N vertices and M edges. We estimate the reliability polynomial for these graphs, and we use the coefficient of determination R^2 as a metric to determine the closeness of fitting the reliability curve using the error function and the Binomial CDF method. The graphs that have been used in this analysis are summarized in Table 1. The last GNM graph in the table is generated with the same number of vertices and edges as an estimated social contact for the New River Valley region near Blacksburg, Virginia. We use the reliability rule $ExpX = 0.2$. We use both k, the number of edges in the sub-graph and x, the ratio of k to the total number of edges E in our analysis. We also use the k_{min} or alternatively x_{min} and k_{max} or x_{max} values to normalize our data. Table 1 shows a comparison between the two-parameter method based on the error function and the Binomial CDF method for different graphs. Based on R^2 values, the error function is a better fit than the Binomial CDF. Therefore, in the rest of the numerical evaluation, we use the error function to represent the reliability polynomial.

An exhaustive search was done in the parameter space to find out the values of a and b with R^2 as a metric of goodness-of-fit for both $R(x)$ and P_k. Figures 2 and 3 show the fits to the error function for both P_k and $R(x)$ for the GNM graphs.

We clearly observe that the parameters a and b decrease as the graph size increases. Meanwhile, the value of R^2 indicates that a better fit is obtained as the graph size increases for fitting P_k and $R(x)$. Next, we change the reliability rule from $ExpX = 0.2$ to 0.4, 0.6, and 0.8 and repeat the analysis. The parameters a and b and R^2 are reported in Table 2.

We evaluate the reliability polynomials for the NRV, Mexico City, Liberia, and Sierra Leone synthetic social networks. The Sierra Leone and Liberia synthetic

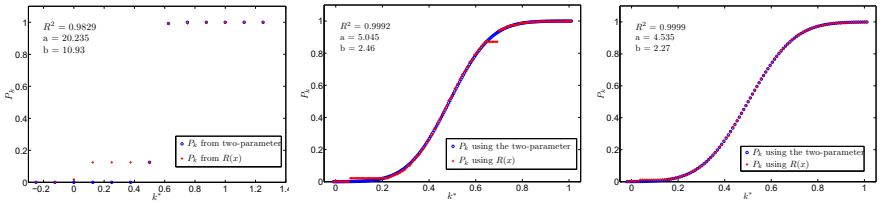


Fig. 2. Plot of P_k and fit to the error function for GNM graphs with k^* for (a)$N = 20$, $M = 50$ (left panel), (b)$N = 2000$, $M = 50,000$ (middle panel) and (c)$N = 153,036$, $M = 4,152,739$ (right panel)

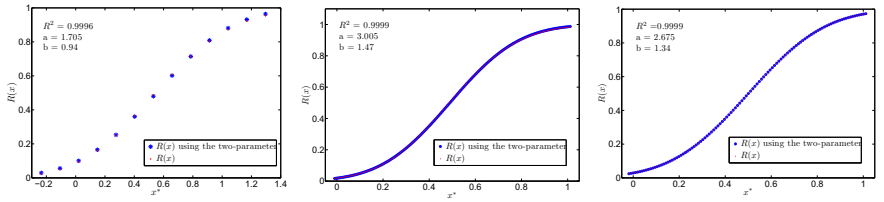


Fig. 3. Plot of $R(x)$ and the fit to the error function for GNM graphs with x^* for (a)$N = 20$, $M = 50$ (left panel), (b)$N = 2000$, $M = 50,000$ (middle panel) and (c)$N = 153,036$, $M = 4,152,739$ (right panel)

Table 2. Fitting the reliability polynomial $R(x)$ of GNM graphs for different reliability rule $ExpX - \alpha$

Graph	$\alpha = 0.4$: R^2,a,b	$\alpha = 0.6$: R^2,a,b	$\alpha = 0.8$: R^2,a,b
$N = 20,\ M = 50$	0.9992,1.81,0.52	0.9999,2.21,0.70	0.9999,2.69,0.70
$N = 100,\ M = 450$	0.9998,1.53,1.24	0.9997,1.27,1.01	0.9994,3.67,1.28
$N = 500,\ M = 35750$	0.9999,2.28,1.16	0.9994,2.22,1.03	0.9986,3.28,1.43
$N = 2000,\ M = 50,000$	0.9996,1.88,1.09	0.9999,2.43,1.07	0.9999,3.57,1.63
$N = 10,000,\ M = 40,000$	0.9999,2.61,1.20	0.9999,3.14,1.60	0.9999,3.60,1.58
$N = 35,000,\ M = 750,000$	0.9999,2.32,1.24	0.9999,2.61,1.33	0.9999,3.24,1.47
$N = 125,000,\ M = 500,000$	0.9999,2.56,1.21	0.9999,2.73,2.32	0.9999,3.39,1.63
$N = 153,036,\ M = 4,152,739$	0.9999,2.39,1.14	0.9999,2.65,1.41	0.9999,3.46,1.75

social networks are available for public use[1] as part of studying the spread of Ebola in Africa. The fitting of $R(x)$ is reported in Figures 4-7. The R^2 values are close to 1 with minimum value of 0.9984 and maximum value of 0.9999 showing that the error function matches the reliability polynomial well and the parameters a and b are thus suitable for characterizing the reliability polynomial and, by extension, the structure of graphs.

[1] `http://vbi.vt.edu/ndssl/ebola`

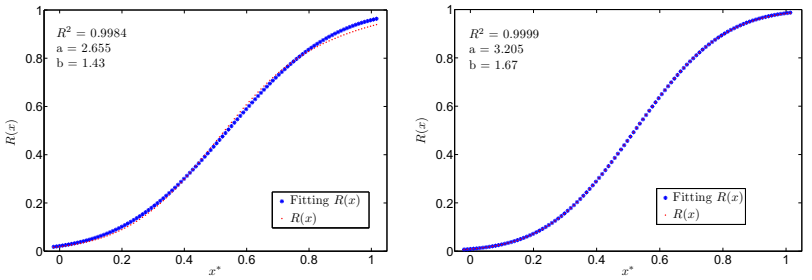


Fig. 4. Plot of $R(x)$ and the error function fit with x^* for the NRV social network for reliability rules (a) $ExpX - 0.1$ (left panel), and (b) $ExpX - 0.2$ (right panel)

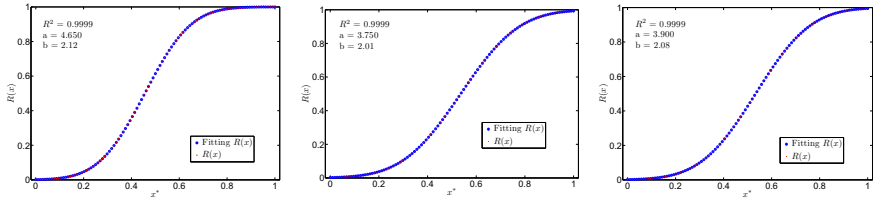


Fig. 5. Plot of $R(x)$ and the error function fit with x^* for Mexico City social network for reliability rules (a) $ExpX - 0.05$ (left panel), (b) $ExpX - 0.1$ and (c) $ExpX - 0.2$ (right panel)

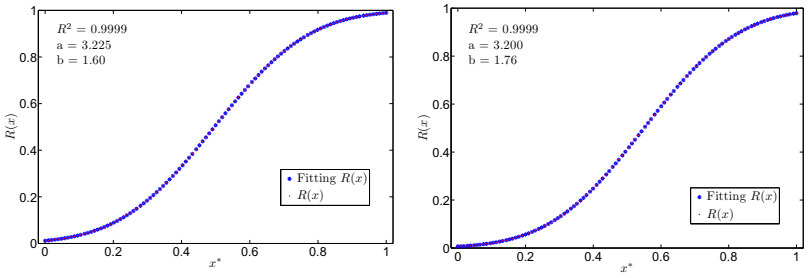


Fig. 6. Plot of $R(x)$ and the error function fit with x^* for the Liberia social network for reliability rules (a) $ExpX - 0.05$ (left panel), and (b) $ExpX - 0.1$ (right panel)

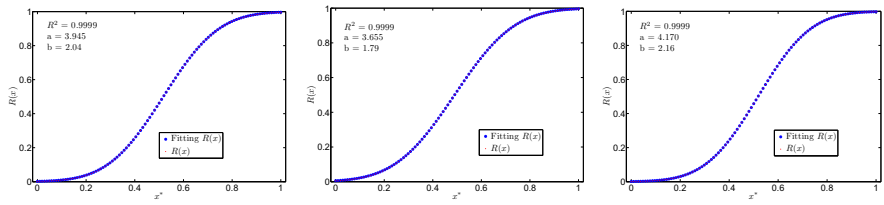


Fig. 7. Plot of $R(x)$ and the error function fit with x^* for Sierra Leone social network for reliability rules (a) $ExpX - 0.05$ (left panel), (b) $ExpX - 0.1$ and (c) $ExpX - 0.2$ (right panel)

5 Conclusions

We have compared a set of reliability polynomials with both the cumulative distribution function (CDF) of a binomial and the error function. We have reported that the error function yields a better fit to the P_k values than the binomial CDF. We suggest using the parameters of the best-fit error function to characterize $R(x)$. The parameters a and b in the error function change the width and shift the position of the mean of the corresponding Gaussian function, respectively. These values increase with the size of the graph, for several different rules. Finally, we use this method to study the nature of the reliability polynomials of the synthetic social networks for NRV, Mexico City, Sierra Leone and Liberia. We conclude that these two parameters and the values of k_{min} and k_{max}, or equivalently or x_{min} and x_{max}, for a particular network, summarize the reliability of the network.

We suggest several ways to use the analyses presented here. First, we can use the values x_{min}, x_{max}, a, and b as descriptive statistics for a network that are more informative for many purposes than the usual statistics such as degree distribution, assortativity, etc. Second, they form a set of sufficient statistics for a given feature of diffusive dynamics on a network. Understanding the relationship between network structure and these statistics provides insight into the structure-to-function problem for networks. Finally, this parameterized form for the reliability polynomial can be useful for studying critical point phenomenology in finite-size systems. Here we have described its use for phenomena related to the epidemic transition, but this is just one instance of a percolation transition.

Acknowledgment. This work has been partially supported by the Defense Threat Reduction Agency (DTRA) [award number HDTRA1-11-1-0016]; the National Institute of General Medical Sciences of the National Institutes of Health (NIH) [Models of Infectious Disease Agent Study (MIDAS) award number 2U01GM070694-09] and by the National Science Foundation (NSF) [Network Science and Engineering Grant CNS-1011769]. The content is solely the responsibility of the authors and does not necessarily represent the official views of DTRA, the NSF, or the NIH.

References

1. Moore, E., Shannon, C.: Reliable circuits using less reliable relays. Journal of the Franklin Institute 262, 191–208 (1956)
2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows: theory, algorithms, and applications. Prentice Hall (1993)
3. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press (1962)
4. Colbourn, C.J.: The Combinatorics of Network Reliability. Oxford University Press (1987)
5. Eubank, S., Youssef, M., Khorramzadeh, Y.: Determining and understanding dynamically important differences between complex networks using reliability-induced structural motifs. In: 2013 International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), Complex Networks Workshop, Tokyo, Japan, December 2-5 (2013)
6. Eubank, S., Youssef, M., Khorramzadeh, Y.: Using the network reliability polynomial to characterize and design networks. Journal of Complex Networks, 1–17 (2014)
7. Halloran, M., Vespignani, A., Bharti, N., Feldstein, L., Alexander, K., Ferrari, M., Shaman, J., Drake, J., Porco, T., Eisenberg, J., Valle, S., Lofgren, E., Scarpino, S., Eisenberg, M., Gao, D., Hyman, J., Eubank, S.: Ebola: Mobility data. Science 346 (2014)
8. Youssef, M., Khorramzadeh, Y., Eubank, S.: Network reliability: the effect of local network structure on diffusive processes. Physical Review E 66 (2013)

Analysis of the Effects of Communication Delay in the Distributed Global Connectivity Maintenance of a Multi-robot System

Vinícius A. Battagello and Carlos H.C. Ribeiro

Aeronautics Institute of Technology (ITA), Marechal Eduardo Gomes Square, 50,
São José dos Campos (SP) - Brazil
{batta,carlos}@ita.br

Abstract. To perform cooperative tasks in a decentralized manner, multi-robot systems are often required to communicate with each other. For that reason, maintaining the communication graph connectivity is a fundamental issue. In this paper, we analyse the effects of communication time delay upon a connectivity maintenance control strategy for robotic agents. The results show that the connectivity strategy is resilient to the negative effects of such disturbance only at small values in the communication delay. However, the inherent inertial characteristics of most terrestrial and aquatic robots opens the perspective of applying the connectivity maintenance strategy to adaptive schemes that consider, for instance, autonomous adaptation to constraints other than the connectivity itself, e.g. communication efficiency and energy harvesting.

1 Introduction

We analyse here the effects of communication delays upon the control strategy for connectivity maintenance first introduced in [1], evaluating its impact in multi-robot systems. As this is a decentralized strategy that requires local communication between agents, the main requirement is to keep the agents always connected during communication, not only for connectivity maintenance *per se*, but also for other tasks that require connectivity among agents, *e.g.* information exhange.

The control strategy considered here is a representative of the so-called *global* connectivity techniques, allowing possible elimination of redundant links and creation of new ones as needed, a typical feature of a mobile network. This is in opposition with *local* connectivity techniques as proposed by [2], [3] and [4], which ensure that once two agents exchange information through a communication link at time $t = 0$, this link will be active $\forall t > 0$, hence creating an initial path from the outset. Imposing the maintenance of each single communication link is costly from an Engineering point of view, and a more convenient scheme can be achieved guaranteeing that redundant links are removed and additional links are generated as needed. Indeed, information exchange among all the agents is guaranteed under *global* connectivity of the communication graph.

The control strategy introduced in [1] ensures that *global* connectivity is maintained in a system of agents, even in the presence of an obstacle set. This approach is inspired

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_15

in the strategy previously defined in [6] and introduces a way of obtaining, in a distributed way, the value of the second smallest eigenvalue of the Laplacian matrix, that, as shown in [7], measures the connectivity of a graph. Regarding other estimate procedures that can be found in [8] and [9], one of the principal advantages of the method described in [1] is that it provides, besides the Fiedler eigenvalue, estimations of its own gradients that are useful in real-time computations, as we will see.

2 Background on Graph Theory

The connectivity scheme is based on graph-theoretical considerations that we outline in this section. Further background details can be found in [10].

Given N mobile robots, we model the instantaneous communication links among them as edges in an undirected graph where each robot corresponds to a node. Communication is assumed to be local, in the sense that each robot i communicates only with a topological neighbourhood, defined as $\mathcal{N}_i$, i.e. the set of robots that can exchange information with it. The complete communication graph is represented by the adjacency matrix $A \in \mathbb{R}^{N \times N}$, where each a_{ij} is defined as the weight of the edge between robots i and j, and is positive if $j \in \mathcal{N}_i$, zero otherwise. This value needs to be computed in order to allow the local Laplacian evaluation, as we'll see next. As we are considering undirected graphs, $a_{ij} = a_{ji}$.

Finally, consider the Laplacian matrix of the graph, defined as $L = D - A$ where $D = diag(\{d_i\})$, and $d_i = \sum_{j=1}^{N} a_{ij}$ is the degree of the i-th node of the graph. L holds some important properties, among which a remarkable relationship between its eigenvalues and the graph connectivity. Namely, let λ_i, $i = 1, \ldots, N$ be the eigenvalues of L. Then

- The eigenvalues can be ordered such that

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N \tag{1}$$

- $\lambda_2 > 0$ if and only if the graph is connected: λ_2 is then defined as the algebraic connectivity of the graph.

This means that any procedure that keeps λ_2 at positive values guarantees graph connectivity, *i.e.*, guarantees a communication path between any pair of nodes.

3 Connectivity Maintenance

In this section we initially summarize a control strategy that assumes that each agent can compute the actual value of λ_2. In the sequence, this hypothesis will be removed with the description of the distributed procedure.

3.1 Centralized Connectivity Maintenance

For the sake of clarity, we present here a summary of the estimation and connectivity maintenance procedures introduced in [1]. Basically, each agent applies a control law

that guarantees connectivity with a value of λ_2 that is kept larger than a predefined lower bound ε.

Let $p_i \in \mathbb{R}^m$ be the state vector describing the position of the i-th agent and u_i be its control input. Considering a group of N single integrator agents (that is, whose motion model is described by $\dot{p} = u$), we have:

$$u_i^c = \dot{p}_i \tag{2}$$

where $p = \left[p_1^T \dots p_N^T\right]$ and u_i^c, defined as:

$$u_i^c = csch^2(\lambda_2 - \varepsilon).\frac{\partial \lambda_2}{\partial p_i} \tag{3}$$

is a control law that guarantees connectivity in a centralized framework (that is, assuming that the real value of the partial derivative of the algebraic connectivity w.r.t. the agent's state is somehow calculated and informed to each agent). This assumption, however, will be removed in Section 3.2.

Now consider that the neighborhood of agent i is denoted by $\mathcal{N}_i$ and the maximum communication range for each agent is denoted by R. Each j-th agent is inside $\mathcal{N}_i$ if $||p_i - p_j|| \leq R$. Also, let υ_2 be the eigenvector corresponding to the eigenvalue λ_2, and let υ_2^k be the k-th component of υ_2. According to [6], $\frac{\partial \lambda_2}{\partial p_i}$ can then be computed as:

$$\frac{\partial \lambda_2}{\partial p_i} = \sum_{j \varepsilon N_i} -a_{ij}\left(\upsilon_2^i - \upsilon_2^j\right)^2.\frac{p_i - p_j}{\sigma^2}$$

where the edge-weights a_{ij} are defined as in Eq. 4:

$$a_{ij} = \begin{cases} e^{-(||p_i - p_j||)^2/(2.\sigma^2)} & , if\ ||p_i - p_j|| \leq R \\ 0 & , otherwise \end{cases}$$

and the scalar parameter σ is chosen to satisfy the boundary condition $e^{\frac{-(R^2)}{(2.\sigma^2)}} = \Delta$, in which Δ is a small defined threshold.

3.2 Decentralized Connectivity Maintenance

The control law we use as a basis in this work was presented in [11] and is an extension of the one presented in Eq. 2, by adding a bounded control term u^d in order to obtain some desired formation behaviour:

$$\dot{p}_i = u_i^c + u_i^d \tag{4}$$

where u_i^c being now given by a variation of Eq. 3 that considers local estimates of eigenvalues and their variations:

$$u_i^c = csch^2(\lambda_2 - \tilde{\varepsilon}).\frac{\partial \tilde{\lambda}_2}{\partial p_i} \tag{5}$$

with $\tilde{\lambda}_2$ being now computed by each agent from an estimate of υ_2^i, given by $\tilde{\upsilon}_2^i$. Let $\tilde{\upsilon}_2 = \left[\tilde{\upsilon}_2^1 \dots \tilde{\upsilon}_2^N\right]^T$, thus $\tilde{\lambda}_2$ is the second smallest eigenvalue that the Laplacian matrix would take if $\tilde{\upsilon}_2$ were the corresponding eigenvector.

From [11], $\tilde{\lambda}_2$ can be expressed according to:

$$\tilde{\lambda}_2 = \frac{k_3}{k_2}.\left[1 - Ave(\{(\tilde{\upsilon}_2^i)^2\})\right] \tag{6}$$

where k_3 and k_2 are control gains and $Ave(.)$ is the averaging operation.

Notice that the actual value of $\tilde{\lambda}_2$ cannot be computed by each agent (because, actually, the real value of $Ave(\{(\tilde{\upsilon}_2^i)^2\})$ cannot be calculated in a distributed way), however an estimate of this average is available to each agent through a consensus procedure (see [6] for further details). Let us call this estimate z_2^i, thus we have for each agent the (calculated) estimate of λ_2^i:

$$\tilde{\lambda}_2^i = \frac{k_3}{k_2}.\left[1 - z_2^i\right] \tag{7}$$

Following the procedure proposed by [11], each agent also computes:

$$\frac{\partial \tilde{\lambda}_2}{\partial p_i} = \sum_{j \varepsilon N_i} -a_{ij}\left(\tilde{\upsilon}_2^i - \tilde{\upsilon}_2^j\right)^2.\frac{p_i - p_j}{\sigma^2} \tag{8}$$

As shown in [11], λ_2^i is a good estimate of both λ_2 and $\tilde{\lambda}_2$. Specifically, given a positive Ξ value, it is possible to ensure that, for every agent i, the absolute difference between the *real* value assumed by λ_2 and its estimate (given by λ_2^i) is bounded from 0 by Ξ. Furthermore, given another positive Ξ' value, the absolute difference between the second smallest eigenvalue of L (that can be computed distributedly and denoted by $\tilde{\lambda}_2$) and λ_2^i is bounded from 0 by Ξ'.

That is, given a positive Ξ'' (given by $\Xi + \Xi'$), the absolute difference between the value assumed by λ_2 and $\tilde{\lambda}_2$ is bounded from 0 by Ξ''. Put differently, each agent is able to locally compute using $\tilde{\lambda}_2$ instead of λ_2 and still be able to obtain a valid measure of the global system connectivity.

Then, although the actual value of $\tilde{\lambda}_2$ is not available to each agent, its value and partial derivatives can be obtained in a decentralized manner, using Eqs. 7 and 8, and then using 5 to keep $\lambda_2 > 0$.

3.2.1 Formation Control Strategy

We will now summarize the main notions of the control strategy used in this work to deal with formation control problems among a set of agents, for more details please refer to [11]. Basically, each agent now implements its own version of Eq. 4 to simultaneously deal with connectivity maintenance and formation control.

As each agent can have a different formation control strategy, a vector form for the connectivity control must be defined. From [11], Eq. 4 is rewritten as:

$$\dot{p} = -\bar{L}.p + u^d \tag{9}$$

where u^d is a vector containing the formation control laws for *each one* of the N agents, and $\bar{L}$ is the modified Laplacian, defined as $\bar{L} = \bar{D} - \bar{A}$. Here, $\bar{D} = diag(\{\bar{d}_i\})$, where $\bar{d}_i = \sum_{j=1}^{N} \bar{a}_{ij}$ and each $\bar{a}_{ij}$ is a modified edge-weight for the vector form of the connectivity maintenance, as defined in [11].

Consensus Based Formation Control

As stated in [13], the following control law can be used:

$$u^d = -L_*.p + b_i(p) \tag{10}$$

where

$$b_i(p) = \begin{cases} \sum_{j \varepsilon N_i} \left(1 + \bar{a}_{ij}(\lambda_2^i)\right).(\bar{p}_i - \bar{p}_j) & , \; if \; \lambda_2^i > k.\tilde{\varepsilon} \\ \sum_{j \varepsilon N_i} (1 + \bar{a}_{ij}(k.\tilde{\varepsilon})).(\bar{p}_i - \bar{p}_j) & , \; otherwise \end{cases}$$

for some $k > 1$, where $\bar{p}_i$ represents the desired relative position for robot i in the formation. This way, when the estimate of the algebraic connectivity is sufficiently greater than $\tilde{\varepsilon}$ (i.e., $\lambda_2^i > k.\tilde{\varepsilon}$), the bias term is computed with the Laplacian matrix $\tilde{L} = \bar{L} + L_*$. Otherwise, when the value of the estimate of the algebraic connectivity falls and approaches $\tilde{\varepsilon}$, the bias term of the second case in Eq. 11 ensures u^d is bounded and guarantees the connectivity maintenance among the agents.

In this arrangement, the multi-robot system was involved in a formation control task, analogous to the described in [11], in which agents were supposed to converge to a regular formation, and move along the x-axis while collisions with randomly placed point obstacles were avoided.

4 The Disturbance Model

In this work we propose a model that adds *communication delay* as a source of corruption into the arrangement proposed by [11]. The model studied here can be represented then by the block diagram in Fig. 1, in which $f(\cdot)$ is given by the expression in Eq. 8. Basically, each agent i computes its own υ_2 estimate, given by υ_2^i, and from it obtains the spatial variation of the corresponding eigenvalue, given by $\frac{\partial \tilde{\lambda}_2}{\partial p_i}$ as explained in Section 3.2.

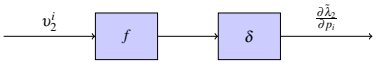


Fig. 1. Block Diagram of the Model with Communication Delay

4.1 Modelling Communication Delay

A key process to keep the system connected is the local estimation of υ_2. Errors in this estimate automatically imply incorrect values of $\tilde{\lambda}_2$, which may result in $\lambda_2 = 0$. Given N agents interacting in an unknown environment, we consider communication delay in data reception. As a result, each agent computes with information that reflects the past of the system. Let $\upsilon_2^{i'}(t)$ be the estimate of υ_2 made by agent i in a given instant t. Then:

$$\upsilon_2^{i'}(t) = \upsilon_2^{i}(t-\delta)\ ,\ for\ t > 0 \tag{11}$$

where δ is the communication delay in *seconds*.

4.2 Control Strategy in the Presence of Communication Delay

Consider λ_2' as the algebraic connectivity of a graph and $\lambda_2^{i'}$ as the estimate of λ_2' made by agent i in the presence of communication delay. If we assume that the disturbance effect of communication delay is limited (that is, information is delivered sometime), then we have, similarly to what was seen in the absence of disturbances, that λ_2^i is a good estimate of both λ_2' and $\tilde{\lambda}_2'$.

Under this condition, we notice that the estimation error of λ_2' (the connectivity measure in the presence of communication delay) is limited (that is, $\exists \Xi > 0$ such that $|\lambda_2^{i'} - \lambda_2| \leq \Xi, \forall i = 1,\ldots,N$). Likewise, we find that the difference between the connectivity estimates and the second smallest eigenvalue of L is also limited (in other words, $\exists \Xi' > 0$ such that $|\lambda_2^{i'} - \tilde{\lambda}_2| \leq \Xi', \forall i = 1,\ldots,N$). Hence, we find that we can use λ_2' to locally estimate the real value of λ_2, inaccessible to the agents.

5 Computer Simulations

The results of the main simulations and experiments presented in this work are available *online*[1]. In Section 5.1 we make an analysis of the performance of *ideal agents* facing the impact of communication delay in their interactions, evaluating the evolution of λ_2 against the value assumed by u^c in order to maintain $\lambda_2 > 0$. The agents are modeled as ideal in the sense that the response to high and low frequencies of the control effort is expected to be not attenuated.

A formation control problem with a varying number of agents ranging from $N = 3$ to $N = 10$ in an environment with $N_{obst} = 0, 1, \ldots, 150$ *obstacles* was considered for our experiments. Simulations have been carried out by considering the following parameter set: $\delta = \{0\ s, 10^{-5}\ s, 5.10^{-5}\ s, \ldots, 0.5\ s\}$.

For the sake of simplicity, from now on we refer to the connectivity measure, its estimates and the control effort in the presence of communication delay (represented until now as λ_2', $\lambda_2^{i'}$ e $u^{c'}$) simply as λ_2, λ_2^i e u^c, and for reference purposes, the connectivity measure in the absence of communication delay will be represented as $\bar{\lambda}_2$.

The results consider a typical execution of the connectivity maintenance algorithm described in [11]. Typical runs of five agents performing formation control correspond to the robots starting at random initial positions and supposed to converge to a pentagonal configuration, while deviating from randomly placed point obstacles along the path.

[1] At `http://goo.gl/7HbaAu`

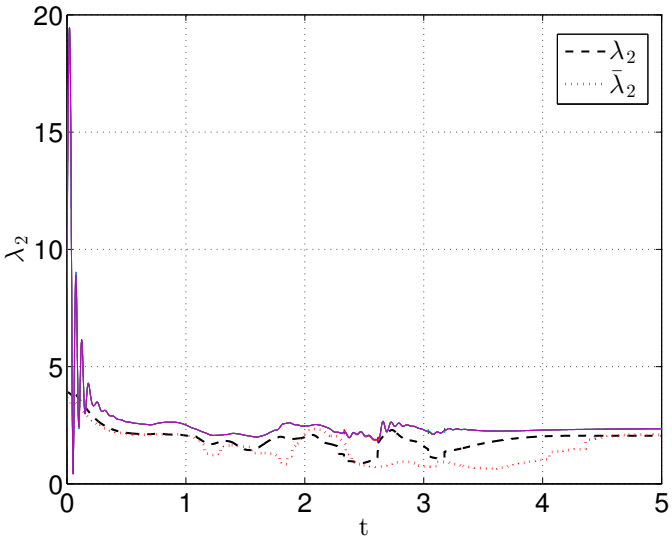


Fig. 2. λ_2 *and* λ_2^i *for* $\delta = 0.001$ *s with* $N_{obst} = 150$

5.1 Communication Delay in Ideal Agents

In Fig. 2 we can find the connectivity and its estimates evolution for $N = 5$ ideal agents interacting with $\delta = 5.10^{-3}$ *s* on an environment with $N_{obst} = 150$ obstacles.

Even despite the existence of communication delay, the results in Fig. 2 show that the system kept itself connected during the interaction. The connectivity ($\lambda_2 \approx 3.92$ in $t = 0$) declines initially and is followed by the estimates after they achieve its highest value ($\lambda_2^{i^{max}} = 15.89$ in $t_{max} \approx 0.03$ *s*). Agents cross the obstacles set between $t_{obst_0} \approx 0.84$ *s* and $t_{obst_1} \approx 3.30$ *s*, which separates temporarily the group of robots (what can be noticed by the two valleys in the value of λ_2 in Fig. 2).

As we can notice, adding communication delay in the agents interaction did not modify substantially the results observed in [11] regarding the connectivity dynamics, once both λ_2 and λ_2^i remained positive in the interaction. In this case, the estimates reasonably follow λ_2 in Fig. 2 and interaction ends with $\lambda_2^i \approx 2.35$ e $\lambda_2 \approx 2.05$ in $t = 5$ *s*.

In Fig. 3 we can find the value of u^c relative to this dynamics. Initially, u^c reaches its highest value ($u^{c^{max}} \approx 5.09.10^5$ in $t_{max} \approx 0.05$ *s*) referring in Fig. 2 to the moment in which the estimates reach their minimum value ($\lambda_2^i \approx 0,69$) after its initial overshoot. After that, estimates rise and become greater than the connectivity ($\lambda_2^i > \lambda_2$ for $t > 0.16$ *s*), before the agents cross the obstacles set (between t_{obst_0} and t_{obst_1}). As can be seen, connectivity is maintained throughout the dynamics.

The analysis of the amplitude spectrum of u^c (not shown here) indicates that the existence of communication delays enhances the contribution of frequency components up to 1.4 *KHz*. This increased participation of high-frequencies in the control effort can be explained by the obstacle avoidance role against the communication delay, as the

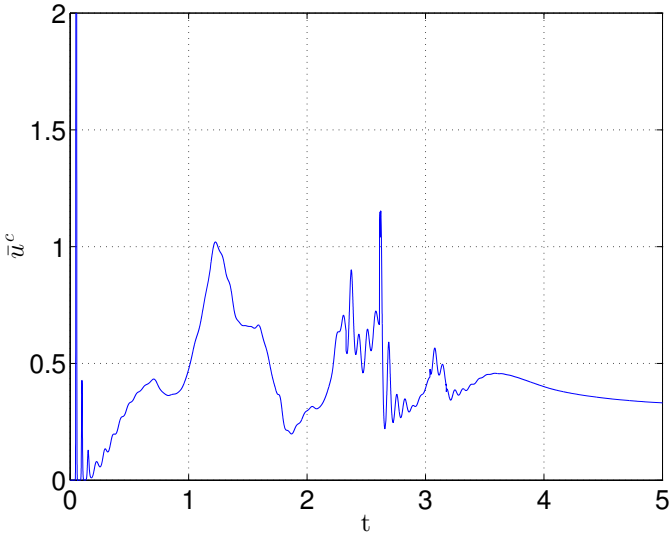


Fig. 3. u^c *for* $\delta = 0.005$ *s with* $N_{obst} = 150$

algorithm must act in an increasingly shorter time as robots deviate from the obstacles with the delayed information.

The analysis here made was based in the parameter set tipically used in other references (as in [11] and [1]). For ideal agents with $\delta > 5.10^{-3}$ *s*, the global connectivity is lost as a consequence of the obstacle avoidance process, what happened slightly earlier for greater values of δ (in not shown results). For $\delta \leq 5.10^{-3}$ *s*, connectivity maintenance was always observed (just like in Section 5.1, everything happened as if there was not any lag in data).

6 Conclusion

In this work, we made an analysis of the effect of communication delay over a control algorithm that, through a decentralized estimation of the algebraic connectivity in a communication graph, guarantees the connectivity maintenance in a multi-robot system. In this analysis, the group connectivity was always maintained *for small values of* δ in agents with ideal controllers (with an equitable response both for high and low frequencies of the control effort). However, for a sensible delay in information exchange, connectivity was not maintained in any studied case.

As the connectivity maintenance is a necessary condition to the estimation procedure described in [11], once it is lost, it is not possible to rely on the local estimates (λ_2^i) made by agents in the presence of any sensible value of $\delta > 0$ in the communication.

As future work, we can relate the analysis of the algorithms described in [17] and [18], that propose a solution to deal with time delay in the communication between agents. Other approaches involve a) evaluate the model dependence upon different sets

of parameters and b) validating the results here proposed in real robots, investigating the pertinence of the assumptions to verifiable realistic situations.

Acknowledgement. The authors thank CAPES. Carlos H. C. Ribeiro thanks FAPESP (proc. nr. 2013/13447-3).

References

1. Sabattini, L., Chopra, N., Secchi, C.: On decentralized connectivity maintenance for mobile robotic systems. In: CDC-ECC, Bologna, Italy (2011)
2. Ji, M., Egerstedt, M.: Coordination Control of Multiagent Systems While Preserving Connectedness. IEEE Transactions on Robotics 23(4), 693–703 (2007)
3. Hsieh, M.A., Cowley, A., Kumar, R.V., Taylor, C.J.: Maintaining network connectivity and performance in robot teams. Journal of Field Robotics 25(1-2), 111–131 (2008)
4. Cao, Y., Ren, W.: Distributed coordinated tracking via a variable structure approach? part I: consensus tracking. part II: swarm tracking. In: Proceedings of the American Control Conference, pp. 4744–4755 (2010)
5. Hollinger, G., Singh, S.: Multi-Robot Coordination with periodic connectivity. In: IEEE International Conference on Robotics and Automation (May 2010)
6. Yang, P., Freeman, R., Gordon, G., Lynch, K., Srinivasa, S., Sukthankar, R.: Decentralized estimation and control of graph connectivity for mobile sensor networks (2010)
7. Fiedler, M.: Algebraic connectivity of graphs. Czechoslovak Mathematical Journal (1973)
8. De Genaro, M.C., Jadbabaie, A.: Decentralized Control of Connectivity for Multi-Agent Systems. In: Proceedings of the IEEE International Conference on Decision and Control, p. 3628 (2006)
9. Zavlanos, M.M., Tanner, H.G., Jadbabaie, A., Pappas, G.J.: Hybrid control for connectivity preserving flocking. IEEE Transactions on Automatic Control 54, 2869–2875 (2009)
10. Godsil, C., Royle, G.: Algebraic Graph Theory. Graduate Texts in Mathematics. Springer (2001)
11. Sabattini, L., Secchi, C., Chopra, N., Gasparri, A.: Distributed Global Connectivity Maintenance for Multi-Robot Systems. In: AUTOMATICA IT Congress, Benevento, Italy (2012)
12. Saber, R., Murray, R.: Consensus Problems in Networks of Agents with Switching Topology and Time-Delays (2003)
13. Fax, J., Murray, R.: Information flow and cooperative control of vehicle formations. IEEE Trans. Automat. Contr. (2004)
14. Yan, S., Zhang, F., Qin, Z., Wen, S.: A 3-DOFs mobile robot driven by a piezoelectric actuator. Smart Materials and Structures (2006)
15. Tan, X., Kim, D., Usher, N., Laboy, D., Jackson, J., Kapetanovic, A., Rapai, J., Sabadus, B., Xin, Z.: An Autonomous Robotic Fish for Mobile Sensing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (2006)
16. Trefethen, L., Bau, D.: Numerical Linear Algebra (1997)
17. Secchi, C., Sabattini, L.: Decentralized global connectivity maintenance for interconnected lagrangian systems with communication delays. In: Proceedings of the IFAC Workshop on Lagrangian and Hamiltonian Methods for Non Linear Control (LHMNLC), Bertinoro, Italy (2012)
18. Secchi, C., Sabattini, L., Fantuzzi, C.: Decentralized global connectivity maintenance for interconnected Lagrangian systems in the presence of data corruption. European Journal of Control (2013)

Inter-layer Degree Correlations in Heterogeneously Growing Multiplex Networks

Babak Fotouhi[1,2] and Naghmeh Momeni[1]

[1] Department of Electrical and Computer Engineering
McGill University, Montréal, Canada
[2] Department of Sociology, McGill University, Montréal, Canada
{babak.fotouhi,naghmeh.momenitaramsari}@mail.mcgill.ca

Abstract. The multiplex network growth literature has been confined to homogeneous growth hitherto, where the number of links that each new incoming node establishes is the same across layers. This paper focuses on heterogeneous growth in a simple two-layer setting. We first analyze the case of two preferentially growing layers and find a closed-form expression for the inter-layer degree distribution, and demonstrate that non-trivial inter-layer degree correlations emerge in the steady state. Then we focus on the case of uniform growth. We observe that inter-layer correlations arise in the random case, too. Also, we observe that the expression for the average layer-2 degree of nodes whose layer-1 degree is k, is identical for the uniform and preferential schemes. Throughout, theoretical predictions are corroborated using Monte Carlo simulations.

1 Introduction

Multiplex networks are tools for modeling networked systems in which units have heterogeneous types of interaction, making them members of distinct networks simultaneously. The multiplex framework envisages different layers to model different types of relationships between the same set of nodes. For example, we can take a sample of individuals and constitute a social media layer, in which links represent interaction on social media, a kinship layer, a geographical proximity layer, and so on. Examples of real systems that have been conceptualized so far using the multiplex framework include citation networks, online social media, airline networks, scientific collaboration networks, and online games [9].

Theoretical analysis of multiplex networks was initiated by the seminal papers [1,2] that invented and introduced theoretical measures for quantifying multiplex networks. Consequently, multiplex networks were utilized for the theoretical study of phenomena such as epidemics [3], pathogen-awareness interplay [4], percolation processes [5], evolution of cooperation [6], diffusion processes [7] and social contagion [8]. For a thorough review, see [9].

In the present paper we focus on the problem of growing multiplex networks. In [13], the case where two layers are homogeneously growing (that is, the number of links that each newly-born node establishes is the same for both layers)

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_16

according to preferential attachment is considered, and it is shown that $\overline{\ell}(k)$ (which is the average layer-2 degree of nodes whose layer-1 degree is k) is a function of k.

Previous results on growing multiplex networks are confined to homogeneously-growing layers [9,11,13]. In the present paper, we consider heterogeneously-growing layers: each incoming node establishes β_1 links in layer 1 and β_2 links in layer 2. We also solve the problem for the case where growth is uniform, rather than preferential. We demonstrate that, surprisingly, the expression for $\overline{\ell}(k)$ is identical to that of the preferential case. We verify the theoretical findings with Monte Carlo simulations.

2 Setup and Notation

The two-layer multiplex network we consider in the present paper possesses one set of nodes and two distinct sets of links. The network comprises two layers, corresponding to the two sets of links. Each node resides in both layers. The degree of node x in layer 1 is denoted by k_x, and its degree in layer 2 is denoted by ℓ_x. The number of nodes at time t is denoted by $N(t)$ and the number of links at layer i is denoted by $L_i(t)$, and $N_{k\ell}(t)$ is the number of nodes that have degrees k and ℓ at time t. We denote the fraction of these nodes by $n_{k,\ell}(t)$. Each incoming node establishes β_1 links in layer 1 and β_2 links in layer 2.

At the inception, there are $L_1(0)$ links in the first layer and $L_2(0)$ links in the second layer. The network grows by the successive addition of new nodes. Each node establishes m links in each layer. So the number of links in layer i at time t is $L_i(0) + \beta_i t$.

3 Model 1: Preferential Attachment

In the first model, incoming nodes choose their destinations according to the preferential attachment mechanism posited in [10]. The probability that an existing node (call it x) receives a layer-1 link from the newly-born node is proportional to k_x, and similarly, the probability for it to receive a layer-2 link is proportional to ℓ_x. Note that to obtain the normalized link-reception probabilities at time t , the former should be divided by $L_1(0) + 2\beta_1 t$ and the latter should be divided by $L_2(0) + 2\beta_2 t$—the number of links in the first and second layers, respectively.

The addition of a new node at time t can alter the values of $N_{k,\ell}$. If a node with layer-1 degree $k-1$ and layer-2 degree ℓ receives a layer-1 link, its layer-1 degree increments to k, and $N_{k\ell}$ increments as a consequence. If a node with layer-1 degree k and layer-2 degree $\ell - 1$ receives a link, its layer-2 degree increments and consequently, $N_{k,\ell}$ increments. There are two events which would result in a decrease in $N_{k,\ell}$: if a node with layer-1 degree k and layer-2 degree ℓ receives a link in either layer. Finally, each incoming node has an initial layer-1 degree and layer-2 degree of β, and increments $N_{\beta 1,\beta 2}$ when it is introduced. The following rate equation quantifies the evolution of the expected value of $N_{k,\ell}$ upon the

introduction of a single node by addressing the aforementioned events with their corresponding probabilities of occurrence:

$$N_{k,\ell}(t+1) = N_{k,\ell}(t) + \beta_1 \frac{(k-1)N_{k-1,\ell}(t) - kN_{k\ell}(t)}{L_1(0) + 2\beta_1 t} + \beta_2 \frac{(\ell-1)N_t(k,\ell-1) - \ell N_t(k,\ell)}{L_2(0) + 2\beta_2 t} + \delta_{k\beta_1}\delta_{\ell\beta_2}. \tag{1}$$

Alternatively, we can write the rate equation for $n_{k\ell}$. Using the substitution $N_{k\ell} = (N(0)+t)n_{k\ell}$, we obtain

$$\begin{aligned}\big[N(0)+t\big]\big[n_{k,\ell}(t+1) - n_{k,\ell}(t)\big] + n_{t+1}(k,\ell) = \\ \beta_1 \frac{(k-1)N_{k-1,\ell}(t) - kN_{k\ell}(t)}{L_1(0) + 2\beta_1 t} \\ + \beta_2 \frac{(\ell-1)N_t(k,\ell-1) - \ell N_t(k,\ell)}{L_2(0) + 2\beta_2 t} + \delta_{k\beta_1}\delta_{\ell\beta_2}.\end{aligned} \tag{2}$$

Now we focus on the limit as $t \to \infty$, when the values of $n_{k\ell}$ reach steady states, and we have

$$\begin{cases} \lim\limits_{t\to\infty} \beta_1 \dfrac{N(0)+t}{L_1(0)+2\beta_1 t} = \dfrac{1}{2} \\ \lim\limits_{t\to\infty} \beta_2 \dfrac{N(0)+t}{L_2(0)+2\beta_2 t} = \dfrac{1}{2} \end{cases} . \tag{3}$$

In this limit (2) transforms into

$$n_{k\ell} = \frac{(k-1)n_{k-1,\ell} - kn_{k\ell}}{2} + \frac{(\ell-1)n_{k,\ell-1} - \ell n_{k\ell}}{2} + \delta_{k\beta_1}\delta_{\ell\beta_2}, \tag{4}$$

Rearranging the terms, this can be equivalently expressed as follows

$$n_{k\ell} = \frac{k-1}{k+\ell+2} n_{k-1,\ell} \frac{\ell-1}{k+\ell+2} n_{k,\ell-1} + \frac{2\delta_{k\beta_1}\delta_{\ell\beta_2}}{2+\beta_1+\beta_2}. \tag{5}$$

This difference equation is solved in Appendix A. The solution is

$$n_{k,\ell} = \frac{2\beta_1(\beta_1+1)\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)k(k+1)\ell(\ell+1)} \frac{\binom{\beta_1+\beta_2+2}{\beta_1+1}}{\binom{k+\ell+2}{k+1}} \binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}. \tag{6}$$

This is depicted in Figure 1a. As a measure of correlation between the two layers, we find the average layer-2 degree of the nodes whose layer-1 degree is k. Let us denote this quantity by $\bar{\ell}(k)$. To calculate $\bar{\ell}(k)$, we need to perform the following summation:

$$\begin{aligned}
\bar{\ell}(k) &= \sum_{\ell} \ell n_{\ell|k} = \sum_{\ell} \ell \frac{n_{k,\ell}}{n_k} \\
&= \sum_{\ell} \ell \frac{\frac{2\beta_1(\beta_1+1)\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)k(k+1)\ell(\ell+1)} \frac{\binom{\beta_1+\beta_2+2}{\beta_1+1}}{\binom{k+\ell+2}{k+1}} \binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{\frac{2\beta_1(\beta_1+1)}{k(k+1)(k+2)}} \\
&= \sum_{\ell} \frac{\beta_2(\beta_2+1)(k+2)}{(2+\beta_1+\beta_2)(\ell+1)} \frac{\binom{\beta_1+\beta_2+2}{\beta_1+1} \binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{\binom{k+\ell+2}{k+1}} \\
&= \sum_{\ell} \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)} \frac{\binom{\beta_1+\beta_2+2}{\beta_1+1} \binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{\binom{k+\ell+2}{\ell}} \qquad (7)
\end{aligned}$$

In Appendix B, we perform this summation. The answer is

$$\bar{\ell}(k) = \frac{\beta_2}{\beta_1+1}(k+2). \qquad (8)$$

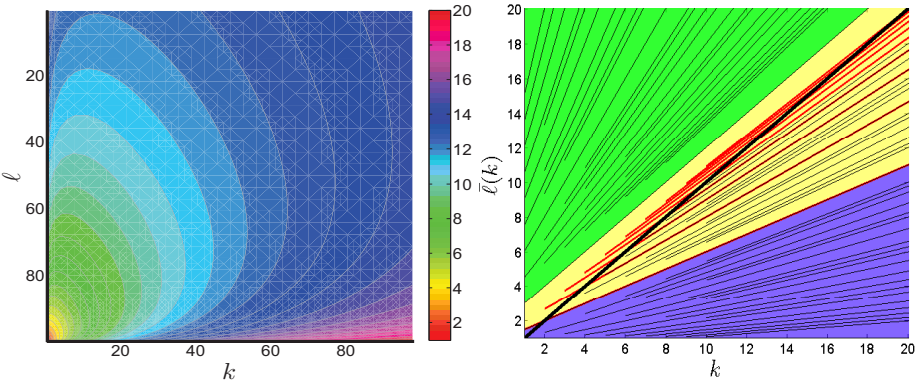
In the special case of $\beta_1 = \beta_2 = m$, this reduces to $\frac{m(k+2)}{1+m}$, which is consistent with the previous result in the literature [13].

Note that (8) if we take the expected value of (8), we obtain

$$\sum_{k} \overline{\ell}(k)p(k) = \frac{\beta_2}{\beta_1+1}(\overline{k}+2) = \frac{\beta_2}{\beta_1+1}(2\beta_1+2) = 2\beta_2, \qquad (9)$$

which coincides with the mean degree in layer 2.

Now let us analyze how adding a layer affects inequality in degrees. We ask, what is the probability that a node has higher degree in layer 2 than in layer 1 (on average)? That is, we seek $P(k < \overline{\ell}(k))$. Analyzing the inequality $k < \frac{\beta_2}{\beta_1+1}(k+2)$, we observe that if $\beta_2 < \beta_1$, then for every k the inequality holds, if $\beta_2 > \beta_1$, then k must be less than $k_c = \frac{2\beta_2}{\beta_1+1-\beta_2}$. So a node with degree below k_c is on average more connected in layer 2 than in layer 1. Note that since the minimum degree in layer 1 is β_1, we should impose an additional constraint on k_c, namely, $k_c \geq \beta_1$. This leads to $\beta_2 \leq \beta_1 - \frac{\beta_1}{\beta_1+2}$. Since β_1 and β_2 can only take integer values, since yields $\beta_2 < \beta_1$. So in order for a node with degree k to have greater expected degree in layer 2 than its given degree in layer 1, first we should have $\beta_2 < \beta_1$, and second, $k \leq k_c$. In short, there are three distinct cases to discern: **(a)** If $\beta_2 > \beta_1$, the inequality holds for all k, that is, on average, every node is more connected in layer 2 than in layer 1. **(b)** If $\beta_2 < \beta_1$, then the inequality never holds. That is, everyone is on average more connected in layer 1. **(c)** If $\beta_1 = \beta_2 = m$, then for nodes whose degree in layer 1 is smaller than $2m$ (which coincides with $\overline{k}$), the inequality holds, and for others it does not. So in the case of homogeneous growth, nodes whose degree in one layer is below the mean degree are on average more connected in the other layer, and nodes with degree higher $2m$ are on average less connected in the other layer. These three cases are depicted in Figure 1b. The purple area pertains to case (a), where curves are $\overline{\ell}(k)$ are always below k, regardless of β_1 and β_2. The green area corresponds to



(a) The inter-layer joint degree distribution for preferential growth with $\beta_1 = 2$ and $\beta_2 = 4$, as given by Equation (6). The function decays fast in k and ℓ, so we have depicted the logarithm of the inverse of this function, for better visibility. Note the skew in the contours. Had β_1 and β_2 been equal, the distribution would be symmetric. The function attains its maximum at $k = \beta, 1$ and $\ell = \beta_2$.

(b) $\overline{\ell}(k)$ for all combinations of $1 \leq \beta_1, \beta_2 \leq 10$. There are three distinct regions. In the green region, $\overline{\ell}(k) > k$ regardless of k, β_1, β_2. In the purple region, the converse is true. In the yellow region, $\overline{\ell}(k) > k$ up to some critical degree $k_c(\beta_1, \beta_2)$, and above the critical degree, $\overline{\ell}(k) < k$. The top boundary corresponds to the case of $\beta_2 = 2, \beta_1 = 1$ and the bottom one pertains to $\beta_1 = \beta_2 = 1$.

Fig. 1. Inter-layer joint degree distribution for preferential growth. The left figure also applies to the case of uniform growth. symmetric.

case (c), where k is always above $\overline{\ell}(k)$. The middle region is the one that $\overline{\ell}(k)$ curves for the cases of $\beta_1 = \beta_2 = m$ reside in. Those curves are depicted in red. It is visible that for each red curve, there is a cutoff degree above which $\overline{\ell}(k) < k$.

4 Model 2: Uniform Attachment in both Layers

In this model, we assume that each incoming node establishes links in both layers by selecting destinations from existing nodes uniformly at random. The rate equation (2) should be modified to the following:

$$\big[N(0)+t\big]\big[n_{k,\ell}(t+1) - n_{k,\ell}(t)\big] + n_{t+1}(k,\theta,\ell) = \\ \beta_1 \frac{N_{k-1,\ell}(t) - N_{k\ell}(t)}{N(0)+t} + \beta_2 \frac{N_t(k,\theta,\ell-1) - N_t(k,\theta,\ell)}{N(0)+t} + \delta_{k\beta_1}\delta_{\ell\beta_2}. \tag{10}$$

Using the substitution $n_{k,\ell}(t) = \frac{N_{k\ell}(t)}{N(0)+t}$, this becomes

$$\big[N(0)+t\big]\big[n_{k,\ell}(t+1) - n_{k,\ell}(t)\big] + n_{t+1}(k,\theta,\ell) = \\ \beta_1 \frac{N_{k-1,\ell}(t) - N_{k\ell}(t)}{N(0)+t} + \beta_2 \frac{N_t(k,\theta,\ell-1) - N_t(k,\theta,\ell)}{N(0)+t} + \delta_{k\beta_1}\delta_{\ell\beta_2}. \tag{11}$$

In the steady state, that is, in the limit as $t \to \infty$, this becomes

$$n_{k\ell} = \beta_1 \frac{n_{k-1,\ell} - n_{k,\ell}}{1} + \beta_2 \frac{n_{k,\ell-1} - n_{k,\ell}}{1} + \delta_{k,\beta_1}\delta_{\ell,\beta_2}. \tag{12}$$

This can be simplified and equivalently expressed as follows

$$n_{k,\ell} = \frac{\beta_1}{1+\beta_1+\beta_2} n_{k-1,\ell} + \frac{\beta_2}{1+\beta_1+\beta_2} n_{k,\ell-1} + \frac{\delta_{k,\beta_1}\delta_{\ell,\beta_2}}{1+\beta_1+\beta_2}. \tag{13}$$

This difference equation is solved in Appendix C. The solution is

$$n_{k,\ell} = \frac{\beta_1^{k-\beta_1}\beta_2^{\ell-\beta_2}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{(1+\beta_1+\beta_2)^{k-\beta_1+\ell-\beta_2+1}} \tag{14}$$

To find the conditional average degree, that is, $\bar{\ell}(k)$, we first need the degree distribution of single layers in order to constitute the conditional degree distribution. This is found previously for example in [13,14]. The degree distribution in the first layer is $n_k = \frac{1}{\beta_1}\left(\frac{\beta_1}{\beta_1+1}\right)^{k-\beta_1+1}$. We need to compute

$$\begin{aligned}\bar{\ell}(k) &= \sum_\ell \ell n_{\ell|k} = \sum_\ell \ell \frac{n_{k,\ell}}{n_k} = \sum_\ell \ell \frac{\dfrac{\beta_1^{k-\beta_1}\beta_2^{\ell-\beta_2}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{(1+\beta_1+\beta_2)^{k-\beta_1+\ell-\beta_2+1}}}{\frac{1}{\beta_1}\left(\frac{\beta_1}{\beta_1+1}\right)^{k-\beta_1+1}} \\ &= \frac{(\beta_1+1)^{k-\beta_1+1}}{(\beta_1+\beta_2+1)^{k-\beta_1+1}} \sum_\ell \ell \frac{\beta_2^{\ell-\beta_2}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{(1+\beta_1+\beta_2)^{\ell-\beta_2}}\end{aligned} \tag{15}$$

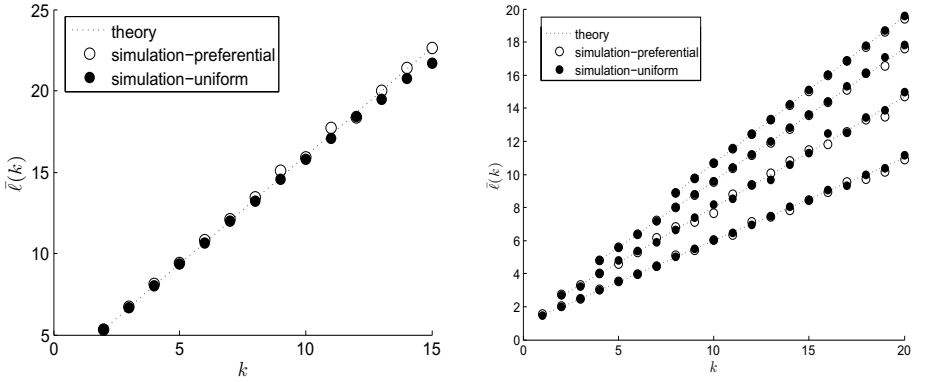
We have performed this summation in Appendix D. The result is

$$\bar{\ell}(k) = \frac{\beta_2}{\beta_1+1}(k+2). \tag{16}$$

This is identical to (8).

5 Simulations

We performed Monte Carlo simulations to verify the results. Figure 2a depicts $\bar{\ell}(k)$ as a function of k for both uniform and preferential attachment for $\beta_1 = 2, \beta_2 = 4$. The two curves are visibly linear and overlapping. Figure 2b depicts $\bar{\ell}(k)$ for both uniform and preferential attachment for $\beta_1 = \beta_2 = m$ for the cases $m = 1, 2, 4, 8$. It can be observed from Figure 2b that in all cases the curves for preferential and uniform growth overlap, and that the slope increases as m increases. This is consistent with the predictions of (16) and (8), where the slope is given by $\frac{m}{m+1}$. This attains its minimum at $m = 1$, and reaches unity for $m \to \infty$.



(a) $\beta_1 = 2, \beta_2 = 4$. (b) $\beta_1 = \beta_2 = m$, for $m = 1, 2, 4, 8$.

Fig. 2. $\overline{\ell}(k)$ for preferential and uniform growth. The left figure depicts $\overline{\ell}(k)$ for an example configuration of heterogeneous growth (i.e., $\beta_1 \neq \beta_2$). The right figure represents results for homogeneous growth. It depicts different $\overline{\ell}(k)$ curves obtained for different values of m, where $\beta_1 = \beta_2 = m$ (the top line is for $m = 8$, and the bottom-most line is for $m = 1$). It can be seen that the slope of $\overline{\ell}(k)$ increases as m increases. The results are averaged over 500 Monte Carlo Trials.

6 Summary and Future Work

We studied the problem of duplex network growth, where the two layers were heterogeneously growing. We considered the cases of preferential and uniform growth separately. We obtained the inter-layer joint degree distribution for both settings. We calculated $\overline{\ell}(k)$, and observed that it is identical in both scenarios. We corroborated the theoretical findings with Monte Carlo simulations.

While the average degree $\overline{\ell}(k)$ are calculated to be the same in Eqs. (8) and (16), it does not mean the two cases have entirely the same correlation properties. Note, for example, that it was obtained in [12] that the two cases have different inter-degree correlation coefficients.

Plausible extensions of the present analysis are as follows. First, there is no closed-form solution in the literature for the inter-layer joint degree distribution of growing multiplex networks with nonzero coupling, where the link reception probabilities in one layer depends on the degrees in both layers. Second, it would be informative to analyze the growth problem in arbitrary times, to grasp the finite size effects and to understand how $\overline{\ell}(k)$ evolves over time, and how the time evolution differs in the preferential and uniform settings. Third, it would plausible to endow the nodes with initial attractiveness, that is, to consider a shifted-linear kernel for the preferential growth mechanism. Fourth, a more realistic and practical model would require intrinsic fitness values for nodes, so it would be plausible to analyze the multiplex growth problem with intrinsic fitness. Finally, since most real systems are multi-layer, it would be plausible to extend the bi-layer results to arbitrary $M > 2$ layers.

References

1. De Domenico, M., Sole-Ribalta, A., Cozzo, E., Kivela, M., Moreno, Y., Porter, M.A., Gomez, S., Arenas, A.: Mathematical formulation of multilayer networks. Phys. Rev. X 3, 041022 (2013)
2. Kivela, A., Arenas, A., Barthelemy, M., Gleeson, J., Moreno, Y., Porter, M.: Multilayer Networks. J. Complex Netw. 2, 203–271 (2014)
3. Son, S.W., Bizhani, G., Christensen, C., Grassberger, P., Paczuski, M.: Percolation theory on interdependent networks based on epidemic spreading. Europhysics Lett. 97, 16006 (2012)
4. Granell, C., Gomez, S., Arenas, A.: Dynamical interplay between awareness and epidemic spreading in multiplex networks. Phy. Rev. Lett. 111, 128701 (2013)
5. Cellai, D., Lopez, E., Zhou, J., Gleeson, J.P., Bianconi, G.: Percolation in multiplex networks with overlap. Phys. Rev. E 88, 052811 (2013)
6. Gomez-Gardenes, J., Reinares, I., Arenas, A., Floria, L.M.: Evolution of cooperation in multiplex networks. Sci. Rep. 2, 620 (2012)
7. Gomez, S., Diaz-Guilera, A., Gomez-Gardenes, J., Perez-Vicente, C.J., Moreno, Y., Arenas, A.: Diffusion dynamics on multiplex networks. Phys. Rev. Lett. 110, 028701 (2013)
8. Cozzo, E., Banos, R.A., Meloni, S., Moreno, Y.: Contact-based social contagion in multiplex networks. Phys. Rev. E 8, 050801 (2013)
9. Boccaletti, S., Bianconi, G., Criado, R., Del Genio, C.I., Gómez-Gardenes, J., Romance, M., Sendina-Nadal, I., Zanin, M.: The structure and dynamics of multilayer networks. Phys. Rep. 544, 1–122 (2014)
10. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
11. Nicosia, V., Bianconi, G., Latora, V., Barthelemy, V.: Non-linear growth and condensation in multiplex networks. Phys. Rev. E 90, 042807 (2014)
12. Kim, J.Y., Goh, K.-I.: Coevolution and correlated multiplexity in multiplex networks. Phys. Rev. Lett. 111(5), 058702 (2013)
13. Nicosia, V., Bianconi, G., Latora, V., Barthelemy, M.: Growing multiplex networks. Phys. Rev. Lett. 111, 058701 (2013)
14. Fotouhi, B., Rabbat, M.: Network growth with arbitrary initial conditions: Degree dynamics for uniform and preferential attachment. Phys. Rev. E 88, 062801 (2013)

A Solving Difference Equation (5)

We need to solve

$$n_{k\ell} = \frac{k-1}{k+\ell+2} n_{k-1,\ell} \frac{\ell-1}{k+\ell+2} n_{k,\ell-1} + \frac{2\delta_{k\beta_1}\delta_{\ell\beta_2}}{2+\beta_1+\beta_2}. \tag{17}$$

We define the new sequence

$$m_{k\ell} \stackrel{\text{def}}{=} \frac{(k+\ell+2)!}{(k-1)!(\ell-1)!} n_{k\ell}. \tag{18}$$

The following holds

$$\begin{cases} \dfrac{k-1}{k+\ell+2} n_{k-1,\ell} = \dfrac{(k-1)!(\ell-1)!}{n}_{k\ell} (k+\ell+2)! m_{k-1,\ell} \\ \dfrac{\ell-1}{k+\ell+2} n_{k,\ell-1} = \dfrac{(k-1)!(\ell-1)!}{n}_{k\ell} (k+\ell+2)! m_{k,\ell-1}. \end{cases} \tag{19}$$

Plugging these into (17), we can recast it as

$$m_{k\ell} = m_{k-1,\ell} + m_{k,\ell-1} + 2\,\frac{(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!}\delta_{k\beta_1}\delta_{\ell\beta_2}. \tag{20}$$

Now define the Z-transform of sequence $m_{k,\ell}$ as follows:

$$\begin{cases} \psi(z,y) \stackrel{\text{def}}{=} \displaystyle\sum_k \sum_\ell m_{k,\ell} z^{-k} y^{-\ell} \\ m_{k,\ell} = \dfrac{1}{(2\pi i)^2} \displaystyle\oint\oint \psi(z,y) z^{k-1} y^{\ell-1} dz dy. \end{cases} \tag{21}$$

Taking the Z transform of every term in (20), we arrive at

$$\psi(z,y) = z^{-1}\psi(z,y) + y^{-1}\psi(z,y) + 2\,\frac{(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} z^{-\beta_1} y^{-\beta_2}. \tag{22}$$

This can be rearranged and rewritten as follows

$$\psi(z,y) = \frac{2}{1-z^{-1}-y^{-1}}\,\frac{(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} z^{-\beta_1} y^{-\beta_2} \tag{23}$$

The inverse transform is given by

$$\begin{aligned} m_{k,\ell} &= \frac{2(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} \oint\oint \frac{z^{k-\beta_1-1} y^{\ell-\beta_2-1} dz dy}{(-4\pi^2)(1-z^{-1}-y^{-1})} \\ &= \frac{2(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} \oint\oint \frac{z^{k-\beta_1} y^{\ell-\beta_2} dz dy}{(-4\pi^2)(zy-z-y)} \\ &= \frac{2(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} \oint\oint \frac{z^{k-\beta_1} y^{\ell-\beta_2} dz dy}{(-4\pi^2)(y-1)\left[z-\frac{y}{y-1}\right]}. \end{aligned} \tag{24}$$

First we integrate over z. We get

$$\begin{aligned} m_{k,\ell} &= \frac{2(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} \oint \frac{\left(\frac{y}{y-1}\right)^{k-\beta_1} y^{\ell-\beta_2} dy}{(2\pi i)(y-1)} \\ &= \frac{2(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} \oint \frac{y^{k-\beta_1+\ell-\beta_2} dy}{(2\pi i)(y-1)^{k-\beta_1+1}}. \end{aligned} \tag{25}$$

Now note that the residue of $\dfrac{f(y)}{(y-1)^n}$ for positive integer equals $\dfrac{f^{(n-1)}(1)}{(n-1)!}$, where the numerator denotes the $(n-1)$th derivative of the function $f(y)$, evaluated at $y=1$. Also, note that the m-th derivative of the function y^n, for integer n and m, equals $\dfrac{m!}{(n-m)!} y^{n-m}$. Combining these two facts, we obtain

$$m_{k,\ell} = \frac{2(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!} \binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}. \tag{26}$$

Using (18), we arrive at

$$n_{k,\ell} = \frac{2(\beta_1+\beta_2+1)!}{(\beta_1-1)!(\beta_2-1)!}\,\frac{1}{k(k+1)\ell(\ell+1)}\,\frac{\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{\binom{k+\ell+2}{k+1}}. \tag{27}$$

This can be equivalently expressed as follows:

$$n_{k,\ell} = \frac{2\beta_1(\beta_1+1)\beta_2(\beta_2+1)}{(\beta_1+\beta_2+2)k(k+1)\ell(\ell+1)}\,\frac{\binom{\beta_1+\beta_2+2}{\beta+1}}{\binom{k+\ell+2}{k+1}}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}. \tag{28}$$

B Performing the Summation in (7)

We need to calculate

$$\bar{\ell}(k) = \sum_{\ell}\frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\,\frac{\binom{\beta_1+\beta_2+2}{\beta+1}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{\binom{k+\ell+2}{\ell}}. \tag{29}$$

We use the following identity: $\frac{1}{\binom{n}{m}} = (n+1)\int_0^1 t^n(1-t)^{n-m}dt$, to rewrite the binomial reciprocal of the coefficient as follows

$$\frac{1}{\binom{k+\ell+2}{\ell}} = (k+\ell+3)\int_0^1 t^{\ell}(1-t)^{k+2}dt. \tag{30}$$

Also, from Taylor expansion, it is elementary to show that

$$S_1(x,n) \overset{\text{def}}{=} \sum_m x^m\binom{m}{n} = \frac{x^n}{(1-x)^{n+1}}. \tag{31}$$

This identity will be used in the steps below. Plugging (30) into (33), we have

$$\begin{aligned}
\bar{\ell}(k) &= \sum_{\ell}\frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\,\frac{\binom{\beta_1+\beta_2+2}{\beta+1}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{\binom{k+\ell+2}{\ell}}\\
&= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\sum_{\ell}(k+\ell+3)\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}\int_0^1 t^{\ell}(1-t)^{k+2}dt\\
&= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\int_0^1(1-t)^{k+2}t^{-k-2}\sum_{\ell}(k+\ell+3)t^{k+\ell+2}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}dt\\
&= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\int_0^1(1-t)^{k+2}t^{-k-2}\frac{d}{dt}\left[\sum_{\ell}t^{k+\ell+3}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}\right]dt\\
&= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\int_0^1(1-t)^{k+2}t^{-k-2}\frac{d}{dt}\left[t^{3+\beta_1+\beta_2}\sum_{\ell}t^{k-\beta_1+\ell-\beta_2}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}\right]dt.
\end{aligned} \tag{32}$$

Using (31), this becomes:

$$\begin{aligned}
\overline{\ell}(k) &= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\int_0^1 (1-t)^{k+2}t^{-k-2}\,\frac{d}{dt}\left[t^{3+\beta_1+\beta_2}\,\frac{t^{k-\beta_1}}{(1-t)^{k-\beta_1+1}}\right]dt \\
&= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\int_0^1 (1-t)^{k+2}t^{-k-2}\,\frac{d}{dt}\left[\frac{t^{k+\beta_2+3}}{(1-t)^{k-\beta_1+1}}\right]dt \\
&= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\int_0^1 (1-t)^{\beta_1}t^{\beta_2}\left[k+\beta_2+3-(1+\beta_1+\beta_2)t\right]dt \\
&= \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\left[(k+\beta_2+3)\int_0^1 (1-t)^{\beta_1}t^{\beta_2}dt-(1+\beta_1+\beta_2)\int_0^1 (1-t)^{\beta_1}t^{\beta_2+1}dt\right] \\
&\overset{(30)}{=} \frac{\beta_2(\beta_2+1)}{(2+\beta_1+\beta_2)}\binom{\beta_1+\beta_2+2}{\beta_1+1}\left[(k+\beta_2+3)\,\frac{\beta_1!\beta_2!}{(\beta_1+\beta_2+1)!}-(1+\beta_1+\beta_2)\,\frac{\beta_1!(\beta_1+1)!}{(\beta_1+\beta_2+2)!}\right] \\
&= \frac{\beta_2(\beta_2+1)\beta_1!\beta_2!}{(2+\beta_1+\beta_2)(1+\beta_1+\beta_2)!}\binom{\beta_1+\beta_2+2}{\beta_1+1}\left[(k+\beta_2+3)-(\beta_2+1)\right] \\
&= \frac{\beta_2}{\beta_1+1}(k+2) \qquad (33)
\end{aligned}$$

C Solving Difference Equation (13)

Let us repeat the equation we need to solve for easy reference

$$n_{k,\ell} = \frac{\beta_1}{1+\beta_1+\beta_2}n_{k-1,\ell} + \frac{\beta_2}{1+\beta_1+\beta_2}n_{k,\ell-1} + \frac{\delta_{k,\beta_1}\delta_{\ell,\beta_2}}{1+\beta_1+\beta_2}. \qquad (34)$$

Let us define the following quantities from brevity:

$$q_1 \overset{\text{def}}{=} \frac{\beta_1}{1+\beta_1+\beta_2} \quad , \quad q_2 \overset{\text{def}}{=} \frac{\beta_2}{1+\beta_1+\beta_2} \qquad (35)$$

Taking the Z transform from both sides of (34), we get

$$\psi(z,y) = q_1 z^{-1}\psi(z,y) + q_2 y^{-1}\psi(z,y) + \frac{z^{-\beta_1}y^{-\beta_2}}{1+\beta_1+\beta_2}. \qquad (36)$$

This can be rearranged and recast as

$$\psi(z,y) = \frac{1}{1-q_1z^{-1}-q_2y^{-1}}\,\frac{z^{-\beta_1}y^{-\beta_2}}{1+\beta_1+\beta_2}. \qquad (37)$$

This can be inverted through the following steps

$$\begin{aligned}
n_{k\ell} &= \frac{1}{(1+\beta_1+\beta_2)(2\pi i)^2}\oint\psi(z,y)z^{k-1}y^{\ell-1}dzdy \\
&= \frac{1}{(1+\beta_1+\beta_2)(2\pi i)^2}\oint\oint\frac{z^{k-\beta-1}y^{\ell-\beta-1}}{1-q_1z^{-1}-q_2y^{-1}}dzdy \\
&= \frac{1}{(1+\beta_1+\beta_2)(2\pi i)^2}\oint\oint\frac{z^{k-\beta_1}y^{\ell-\beta_2}}{zy-yq_1-zq_2}dzdy \\
&= \frac{1}{(1+\beta_1+\beta_2)(2\pi i)^2}\oint\oint\frac{z^{k-\beta_1}y^{\ell-\beta_2}}{z-\frac{yq_1}{y-q_2}}\,\frac{1}{y-q_2}dzdy. \qquad (38)
\end{aligned}$$

There is a single simple pole at $z = \frac{yq_1}{y-q_2}$, which renders the integral trivial:

$$n_{k\ell} = \frac{\oint \frac{y^{\ell-\beta_2}}{y-q_2}\left(\frac{yq_1}{y-q_2}\right)^{k-\beta_1} dzdy}{(1+\beta_1+\beta_2)(2\pi i)} = \frac{q_1^{k-\beta_1}\oint \frac{y^{k-\beta_1+\ell-\beta_2}}{(y-q_2)^{k-\beta_1+1}} dzdy}{(1+\beta_1+\beta_2)(2\pi i)}$$
$$= \frac{q_1^{k-\beta_1}(k-\beta_1+\ell-\beta_2)!}{(1+\beta_1+\beta_2)(k-\beta_1)!(\ell-\beta_2)!} q_2^{\ell-\beta_2} = \frac{q_1^{k-\beta_1}q_2^{\ell-\beta_2}}{(1+\beta_1+\beta_2)} \binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}. \tag{39}$$

After inserting the expressions for q_1, q_2 from (35), this becomes

$$n_{k,\ell} = \frac{\beta^{k-\beta_1}\beta_2^{\ell-\beta_2}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{(1+\beta_1+\beta_2)^{k-\beta_1+\ell-\beta_2+1}}. \tag{40}$$

D Performing the Summation in (15)

We need to perform the following summation:

$$\bar{\ell}(k) = \frac{(\beta_1+1)^{k-\beta_1+1}}{(\beta_1+\beta_2+1)^{k-\beta_1+1}} \sum_{\ell} \ell \frac{\beta_2^{\ell-\beta_2}\binom{k-\beta_1+\ell-\beta_2}{k-\beta_1}}{(1+\beta_1+\beta_2)^{\ell-\beta_2}} \tag{41}$$

Let us denote $k-\beta_1$ by k' and $\ell-\beta_2$ by ℓ'. Also let us denote $\frac{\beta_2}{1+\beta_1+\beta_2}$ by x. We need to evaluate the following sum: $\sum_{\ell'}(\ell'+\beta_2)x^{\ell'}\binom{k'+\ell'}{k'}$. Let us use (31) and define $S_1(x,n) \stackrel{\text{def}}{=} \sum_m x^m \binom{m}{n} = \frac{x^n}{(1-x)^{n+1}}$. We have:

$$\sum_{\ell'}(\beta_2+\ell')x^{\ell'}\binom{k'+\ell'}{k'} = \beta_2 x^{-k'} S_1(x,k') + x\sum_{\ell'}\ell' x^{\ell'-1}\binom{k'+\ell'}{k'}$$
$$= \beta_2 x^{-k'} S_1(x,k') + x\frac{d}{dx}\Big(x^{-k'}S_1(x,k')\Big) = \beta_2 x^{-k'}\frac{x^{k'}}{(1-x)^{k'+1}} + x\frac{d}{dx}\Big(\frac{x^{k'}}{(1-x)^{k'+1}}\Big)$$
$$= \frac{1}{(1-x)^{k'+2}}\big[\beta_2 + x(k'+1-\beta_2)\big]. \tag{42}$$

Replacing x with $\frac{\beta_2}{1+\beta_1+\beta_2}$ and inserting this result into (41), we get

$$\frac{1}{[1-(\frac{\beta_2}{1+\beta_1+\beta_2})]^{k-\beta_1+2}}\big[\beta_2 + \frac{\beta_2}{1+\beta_1+\beta_2}(k-\beta_1+1-\beta_2)\big]$$
$$= \frac{(1+\beta_1+\beta_2)^{k-\beta_1+2}}{(1+\beta_1)^{k-\beta_1+2}}\big[\beta_2+2+\frac{\beta_2}{1+\beta_1+\beta_2}(k-\beta_1+1-\beta_2)\big]$$
$$= \frac{(1+\beta_1+\beta_2)^{k-\beta_1+2}}{(1+\beta_1)^{k-\beta_1+2}}\big[\frac{\beta_2(k+2)}{1+\beta_1+\beta_2}\big] \tag{43}$$

Plugging this into (41), we get

$$\bar{\ell}(k) = \frac{\beta_2(k+2)}{1+\beta_1} \tag{44}$$

Dynamics of Conflicting Beliefs in Social Networks

Shuwei Chen, David H. Glass, and Mark McCartney

School of Computing and Mathematics, University of Ulster,
Newtownabbey, Co. Antrim, BT37 0QB, UK
{s.chen,dh.glass,m.mccartney}@ulster.ac.uk

Abstract. This paper analyzes two proposed models for simulating opinion dynamics in social networks where beliefs might be considered to be competing. In both models agents have a degree of tolerance, which represents the extent to which the agent takes into account the differing beliefs of other agents, and a degree of conflict, which represents the extent to which two beliefs are considered to be competing. In this paper, we apply different tolerance and conflict degrees to different groups in a network, and see how these groups affect each other. Simulations show that the groups having different tolerance degrees do not have significant effect upon each other in both Models I and II. On the other hand, the group perceiving a conflict causes more diversity in the agents based on Model I, but introduces a higher consensus level among agents when the fraction becomes larger in Model II.

Keywords: Opinion dynamics, Social network, Conflicting beliefs, Bounded confidence.

1 Introduction

Computer simulations have been employed successfully in the study of agent-based opinion dynamics in social networks for a long time from a wide range of perspectives, e.g., sociology, physics and philosophy [1,2,3,4,5,6]. In these models of opinion dynamics, a group of agents who hold beliefs about a given topic interact with each other to seek truth or reach consensus. Multidimensional opinion dynamics have recently become an active research area [7,8,9,10,11], where agents interact with each other based on their opinions on several topics, e.g., sports and politics. Following the ideas on multidimensional opinion dynamics, we have proposed two models for simulating the scenarios where the beliefs of agents about two (or more) topics may be perceived to be competing, e.g. two explanations of a given phenomenon [12].

The proposed models consider two competing beliefs, i.e., two dimensions, and they both consist of two updating steps. In the first step, the agents update their beliefs via network interaction by talking to their neighbors whose opinions are similar to theirs, and the similarity is decided by bounded confidence (tolerance degree) of each agent. The second step involves an internal update process allowing agents to update their beliefs based on the perceived conflict between beliefs. In the previous simulations, all of the agents in the network were assumed to have the same tolerance

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_17

degrees with respect to two beliefs and the same conflict degree between the beliefs. In reality, however, the conflict and tolerance degrees of the agents may well differ from each other in most cases. It is therefore interesting and worthwhile to investigate how different groups of agents with different conflict and tolerance degrees affect each other during the belief update process.

The rest of this paper is structured as follows. We give an overview of the two belief update models in Section 2. Computer simulations and analysis are provided in Section 3 to investigate the impact of a fraction of the group having a particular conflict or tolerance degree on the belief update in the proposed models. Conclusions and discussions are presented in Section 4.

2 The Models

Assume that we have a network of n vertices, representing agents. Each agent holds two, possibly conflicting, beliefs about two topics, denoted as A and B, and the degrees of both beliefs may change along a set of discrete time points according to a given update mechanism. Both of the proposed models consist of two steps where the first step is to update the belief degrees of agents via network interaction and the second step involves an internal agent update process by taking the perceived conflict into consideration [12].

2.1 Network Update

For the first step (network update), we extend the well-known Hegselmann-Krause (HK) model [4,5,8,13] to include two-dimensional beliefs. The HK model involves a complete graph but the agents are only influenced by the neighbors who have similar opinions to theirs, where the similarity is decided by so-called bounded confidence. Suppose that $A_i(t)$ and $B_i(t)$ are the degrees of beliefs on two topics A and B of the ith agent at time t, where $A_i(t)$, $B_i(t) \in [0, 1]$, with 0, 1, 0.5 corresponding to total disbelief, total belief, and indifference respectively, for all i and t, then the new belief degrees for agent i at time t+1 based on the HK model are

$$A_i(t+1) = \left|I_A(i,t)\right|^{-1} \sum_{j \in I_A(i,t)} A_j(t),$$

$$B_i(t+1) = \left|I_B(i,t)\right|^{-1} \sum_{j \in I_B(i,t)} B_j(t). \quad (1)$$

Here $I_A(i,t) = \{j : \left|A_i(t) - A_j(t)\right| \le \varepsilon_A\}$ and $I_B(i,t) = \{j : \left|B_i(t) - B_j(t)\right| \le \varepsilon_B\}$ are epistemic neighborhoods of agent i at time t with respect to A and B correspondingly, that is, the sets of agents whose belief degree in A or B at t is close to that of the corresponding belief of agent i at that time [8]. The parameters ε_A and ε_B, called tolerances [14], decide the bounded confidence intervals for the two beliefs, and $\left|I_A(i,t)\right|$ and $\left|I_B(i,t)\right|$ represent the cardinalities of the corresponding sets.

2.2 Internal Update

To consider conflict between the two beliefs, two models have been proposed at the internal update step, which represent different attitudes of people towards conflict between beliefs [12]. The degree of conflict is denoted as $c_i \in [0, 1]$, where 0 and 1 correspond to no perceived conflict and total conflict respectively.

The first model (Model I) suggests that if there is no perceived conflict or if $A_i(t) \leq 0.5$ and $B_i(t) \leq 0.5$, then the internal agent update will result in no change in both beliefs. Further, if one, or both of the belief degrees are greater than 0.5 and $c_i > 0$, then the perceived conflict will decrease the degree of the lesser held belief, but not increase the degree in the other. Specifically, if $c_i = 1$ then the lesser held belief should be rejected, i.e., its degree should be set to zero. It means that Model I represents the attitude of a group of people who incline to accept only one of the beliefs with larger degree but reject the other one if there is conflict between them. A rule for achieving this is

$$A_i^*(t) = \begin{cases} A_i(t), & \text{if } A_i(t), B_i(t) \leq 1/2 \text{ or } A_i(t) > B_i(t), \\ \max(\min(A_i(t), B_i(t) - c_i), 0), & \text{if } A_i(t) < B_i(t), B_i(t) > 1/2, \\ \max(\min(A_i(t), B_i(t) - c_i), 0), & \text{if } A_i(t) = B_i(t), B_i(t) > 1/2 \end{cases} \tag{2}$$

with a corresponding rule for belief B, where the * superscript signifies an internal agent update. It is noted that the last rule contains the assignment at probability of p to prevent a 'stalemate' at equality, i.e., we randomly pick one of the beliefs to decrease. We usually set $p = 0.5$ based on the assumption that there is no bias between the two beliefs.

Different from Model I, which decreases the degree of the lesser held belief if there is a perceived conflict, the second model (Model II) tries to make the sum of the two belief degrees closer to 1, reaching unity when there is maximum conflict ($c_i = 1$). It also assumes that the beliefs will not change if there is no perceived conflict, i.e. $c_i = 0$. A rule that achieves this can be given as

$$A_i^*(t) = (1 - c_i)A_i(t) + c_i \frac{A_i(t)}{A_i(t) + B_i(t)}, \tag{3}$$

with a corresponding rule for belief B. This model is more appropriate for cases where the agent is unlikely to reject or accept both beliefs and might apply, for example, in contexts where an explanation is needed and there are only two plausible competing explanations.

The two proposed models represent two possible strategies for agents to update their beliefs when there is perceived conflict between them. The previous simulations have shown that, when there is a conflict between the two beliefs, Model I is more likely to partition the agents into several distinct groups with one of the beliefs being rejected, while Model II is highly likely to make the agents reach consensus in both beliefs.

3 Simulations and Results

The simulations are implemented in a complete network with a fixed number of 100 agents. The initial degrees of the two beliefs are both generated randomly (uniformly distributed) for each agent, as in most of the existing multidimensional models based on the assumption that there is no pre-defined bias between the two beliefs. Given randomly generated initial belief degrees, simulations might show variant results even with the same settings. We therefore implement 100 runs with all the other conditions being the same and study the average performance.

As a measure of consensus we use the average standard deviation. The standard deviation is calculated after each run with respect to the obtained belief degrees of agents, and the average standard deviation is then obtained across 100 runs. It is then not difficult to see that the larger the average standard deviation is, the more diverse the agents are, i.e., the lower consensus level the agents can achieve. The average standard deviation being zero means that there is a total consensus among the agents in the corresponding belief. We explore these two quantities in the simulations: the average degree of beliefs and the average standard deviation of beliefs.

3.1 Fraction of Tolerance

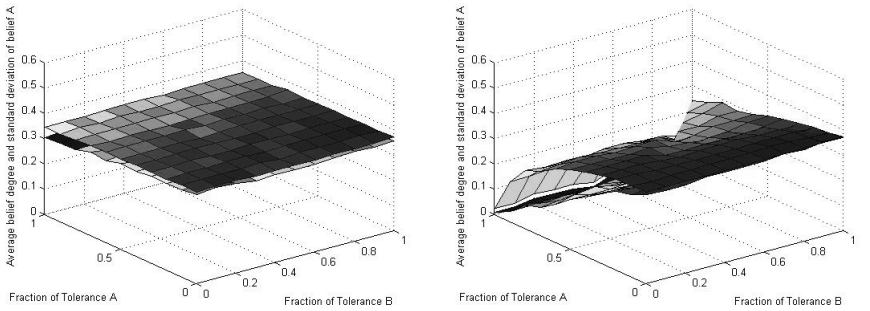
In previous work fixed tolerance degrees were used for two beliefs [12], i.e., a larger tolerance degree ($\varepsilon_A = 0.25$) for belief *A* and a smaller degree ($\varepsilon_B = 0.05$) for belief *B*. It was also assumed that all the agents hold the same tolerance degrees for the corresponding beliefs. Here, we divide the agents into two groups holding different tolerance degrees to see how the different groups affect each other during the belief update process.

The division of agents is realized by a fraction of tolerance either in belief *A* or *B*. Suppose that the fraction is 0.6, this means that 60% of agents take the predefined tolerance degree in belief *A* or *B*, while the remaining 40% take a small tolerance degree 0.05, which means they are highly intolerant. We fix both the degree of conflict and the fraction of conflict to be 1 in this subsection to avoid confusion. The tolerance degrees of the two beliefs are assumed to be equal to each other and we consider two possible degrees of tolerance, 0.2 and 0.4.

Model I

Fig. 1 shows the simulation results using Model I for average degree and average standard deviation of belief *A* across different fractions of tolerances about two beliefs. It can be seen that the agents maintain a high level of diversity (large average standard deviation value) in belief *A* across the fraction of tolerances when the tolerance degrees are small (=0.2). This is caused by the nature of Model I accepting only one of the beliefs with larger degree but rejecting the other one when there is perceived conflict between them, and this makes the agents highly likely to partition into distinct groups. When the tolerance degrees are high enough (=0.4), the agents can reach consensus in belief *A* at the borderline where the fraction of tolerance of belief

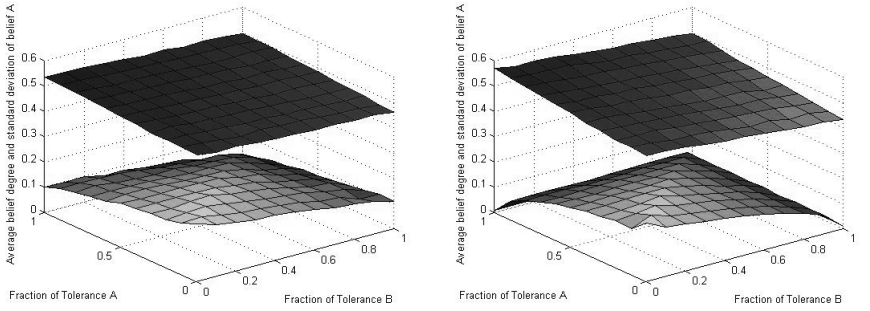
A is 1, but the diversity still remains high when the fraction of tolerance of belief *A* is low no matter what the fraction of tolerance of belief *B* is. The simulation results on belief *B* are symmetric to that of belief *A*, and are not included here.



(a) (b)

Fig. 1. Simulation results for Model I. Average belief degrees (upper surface) and average standard deviations (lower surface) of belief *A* with respect to fraction of tolerance with tolerance degrees being (a) 0.2, (b) 0.4

Model II



(a) (b)

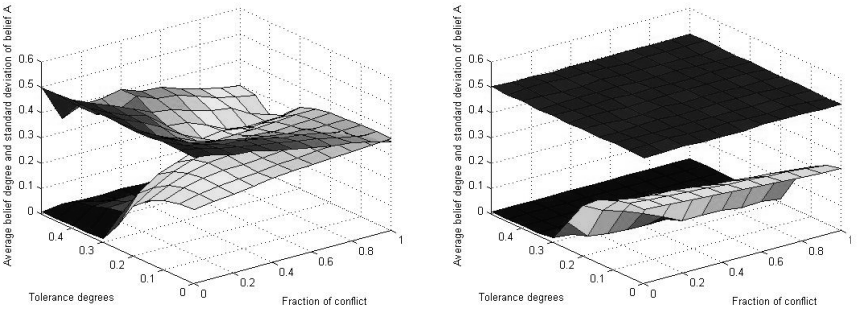
Fig. 2. Simulation results for Model II. Average belief degrees (upper surface) and standard deviations (lower surface) of belief *A* with respect to fraction of tolerance with tolerance degrees being (a) 0.2, (b) 0.4

Similarly, Fig. 2 shows that the fraction of tolerance has little impact on consensus of agents in belief *A* based on Model II when the tolerance degrees are small (=0.2). This is because that the change of fraction of tolerance does not essentially change the fact that all the agents are intolerant given small tolerance degrees. When the tolerance degree becomes larger (=0.4), the agents are able to reach consensus at both the borderlines where either fraction of tolerance of belief *A* or *B* is 1 unlike Model I where this does not occur when the fraction of tolerance of belief *B* is 1. This is

caused by the nature of Model II interpreting conflict in terms of the belief degrees summing to one and so when the agents can reach consensus in one of the beliefs the other will also achieve consensus. Hence, this is primarily the result of the impact of the conflict on consensus rather than the effect of the two groups upon each other, given the fact that the agents in the 'intolerant' group reject interaction with the other agents and maintain their beliefs. The simulation results on belief *B* are not included due to the fact that they are symmetric to that of belief *A*.

3.2 Fraction of Conflict

It is also worthwhile to investigate how the groups holding different conflict degrees affect each other during the belief update process. To make the situation simpler, we fix the conflict degree to be 1, and apply this to a fraction of the agents, which divides the agents into two groups where one group holds total conflict and another holds no conflict. The fractions of tolerances with respect to both beliefs are also fixed to be 1 so that all agents have the same tolerance degrees. We consider two situations where both the tolerance degrees change equally from 0 to 0.5 in the first situation, and in another situation where there is a larger tolerance degree for one belief than the other.



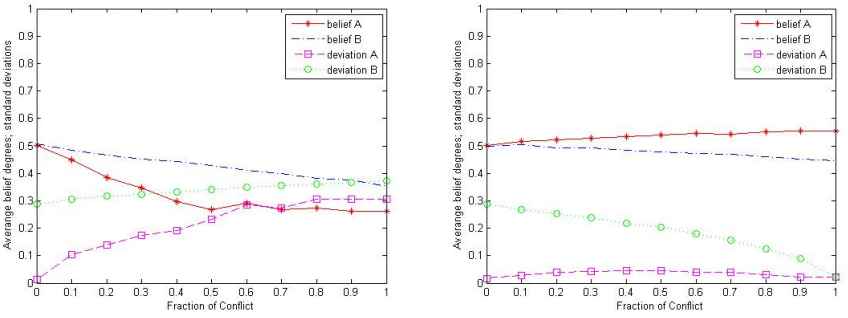
(a) (b)

Fig. 3. Simulation results for (a) Model I and (b) Model II. Average belief degrees (upper surface) and standard deviations (lower surface) of belief *A* with respect to fraction of conflict and tolerance degrees

It can be seen from Fig. 3 (a) that, given larger tolerance degrees, the agents become more likely to reach consensus when the fraction of conflict decreases. This can be explained as due to the way conflict is represented in Model I since it usually makes the agents accept only the belief with the larger degree of belief but reject the other one. Thus introducing more agents holding perceived conflict between beliefs makes the agents more likely to form multiple groups with belief *A* or *B* being rejected. On the other hand, Fig. 3 (b) shows that the fraction of conflict has little impact on Model II when the tolerance degrees are larger than 0.3 or smaller than 0.2, while the increase of fraction of conflict makes the agents become more likely to reach consensus when the tolerance degrees are between 0.2 and 0.3. In other words,

introducing more agents holding conflicting beliefs in Model II lowers the consensus threshold from around 0.3 to 0.2.

For another scenario where the tolerance degree of belief *A* is fixed at 0.3, and the tolerance degree of belief *B* is 0.05, Fig. 4 show the results of the impact of fraction of conflict in the two models. It can be seen from Fig. 4 (a) that the increase of the fraction of conflict from 0 to 1 in Model I decreases the consensus in the belief with larger tolerance degree (belief *A*), and this is also the case for the belief with smaller tolerance degree (belief *B*) although the decrease is less dramatic. On the other hand, Fig. 4 (b) shows that introducing more conflicting agents in Model II has little impact on the belief with larger tolerance degree (belief *A*), but makes the agents increase consensus in the belief with the smaller tolerance degree (belief *B*). These results further verify the natures of the two models on conflict between beliefs. When there is perceived conflict between the two beliefs, the agents in Model I are more likely to accept only one of the beliefs but reject another, while Model II makes the agents to reach consensus in both beliefs if they can reach consensus in one of the beliefs.



(a) (b)

Fig. 4. Simulation results for (a) Model I and (b) Model II. Average belief degrees and average standard deviations of both beliefs with respect to fraction of conflict with tolerance degree of belief *A* being 0.3 and that of belief *B* being 0.05

4 Conclusions

Based on the two proposed models on two-dimensional opinion dynamics when there is perceived conflict between the two beliefs, this paper has examined the effect of varying the fraction of the population having given tolerance and conflict degrees to investigate group behavior. Simulation results show that the groups having different tolerance degrees do not have significant effect upon each other in both Models I and II, because one of the groups is 'intolerant' whose agents reject interaction with the other agents. On the other hand, the fraction of the group holding perceived conflict causes more diversity in the agents based on Model I, but introduces a higher consensus level among agents when the fraction becomes larger in Model II.

This paper considers two competing beliefs, but the ideas contained herein are generalizable to cases where there are a larger set of beliefs. The current paper considered the case that the agents only update their beliefs according to the beliefs of their neighbors. In future work this will be extended so that the agents can take reported information, external to the network, into consideration when updating their beliefs. Different network structures will also be explored to see the impact of network topology on the conflicting opinion dynamics.

Acknowledgments. This publication was made possible by a grant from the John Templeton Foundation (Grant no. 40676). The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the John Templeton Foundation.

References

1. French, J.R.P.: A Formal Theory of Social Power. Psychological Review 63, 181–194 (1956)
2. Harary, F.: A Criterion for Unanimity in French's Theory of Social Power. In: Cartwright, D. (ed.) Studies in Social Power. Institute for Social Research, Ann Arbor (1959)
3. Deffuant, G., Neau, D., Amblard, F., Weisbuch, G.: Mixing Beliefs among Interacting Agents. Advances in Complex Systems 3, 87–98 (2000)
4. Krause, U.: A discrete nonlinear and non-autonomous model of consensus formation. In: Elaydi, S., Ladas, G., Popenda, J., Rakowski, J. (eds.) Communications in Difference Equations, pp. 227–236. Gordon and Breach Publ., Amsterdam (2000)
5. Hegselmann, R., Krause, U.: Opinion Dynamics and Bounded Confidence: Models, Analysis, and Simulations. Journal of Artificial Societies and Social Simulation 5(3), 1–33 (2002)
6. Weisbuch, G., Deffuant, G., Amblard, F., Nadal, J.P.: Meet, Discuss and Segregate! Complexity 7, 55–63 (2002)
7. Pluchino, A., Latora, V., Rapisarda, A.: Compromise and Synchronization in Opinion Dynamics. European Physical Journal B 50, 169–176 (2006)
8. Riegler, A., Douven, I.: Extending the Hegselmann–Krause Model III: From Single Beliefs to Complex Belief States. Episteme 6, 145–163 (2009)
9. Jacobmeier, D.: Multidimensional Consensus Model on a Barabasi-Albert Network. International Journal of Modern Physics C 16, 633–646 (2005)
10. Fortunato, S., Latora, V., Pluchino, A., Rapisarda, A.: Vector Opinion Dynamics in a Bounded Confidence Consensus Model. International Journal of Modern Physics C 16(10), 1535–1551 (2005)
11. Lorenz, J.: Fostering Consensus in Multidimensional Continuous Opinion Dynamics under Bounded Confidence. In: Helbing, D. (ed.) Managing Complexity, pp. 321–334. Springer, Berlin (2008)
12. Chen, S., Glass, D.H., McCartney, M.: Dynamics of Multidimensional Conflicting Opinions in Social Networks. In: European Conference on Social Intelligence (ECSI 2014), pp. 76–86. CEUR Proceedings, Barcelona (2014)
13. Douven, I., Riegler, A.: Extending the Hegselmann–Krause Model I. Logic Journal of the IGPL 18, 323–335 (2010)
14. Zollman, K.J.: Social Network Structure and the Achievement of Consensus. Politics, Philosophy and Economics 11(1), 26–44 (2012)

Building Mini-Categories in Product Networks

Dmitry Zinoviev[1], Zhen Zhu[2], and Kate Li[3]

[1] Department of Mathematics and Computer Science
[2] Department of Marketing
[3] Department of Information Systems and Operations Management
Suffolk University,
73 Tremont St., Boston, MA 02108, USA
{dzinoviev,zzhu,kjli}@suffolk.edu

Abstract. We constructed a product network based on the sales data collected and provided by a major nationwide retailer. The structure of the network is dominated by small isolated components, dense clique-based communities, and sparse stars and linear chains and pendants. We used the identified structural elements (tiles) to organize products into mini-categories—compact collections of potentially complementary and substitute items. The mini-categories extend the traditional hierarchy of retail products (group–class–subcategory) and may serve as building blocks towards exploration of consumer projects and long-term customer behavior.

Keywords: retailing, product network, mini-category, category management.

1 Introduction

Consumer projects are large and major customer undertakings, often involving a considerable amount of money, effort, and emotions. Examples of consumer projects include porch renovation, Christmas decoration, wedding planning, and moving into a college dorm. For each project, customers often make multiple cross-category purchases through multiple shopping trips. Such projects, in light of their significant relevance to retailers' financial outcomes and customer relationship [1], are subject to thorough academic and managerial investigations.

Theoretically, customer project management represents the frontier of the category management domain, which is considered crucial by 72% of retailers surveyed by Kantar Retail in 2011. For years, most retailers have been using only standard market research tools, mostly for within-transaction product associations (e.g., market basket analysis [2]) and only from the functional or manufacturers' perspectives for understanding product categories [3]. Few studies have explored product association at the consumer project level.

The criticality of category management and the dearth of understanding of consumers' project purchase behaviors serve as the impetus of this research. This study aims to answer a key question: how to categorize purchased products properly to prepare for project detection? Equipped with the new advancements

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_18

in complex network analysis techniques [4,5], we expect our study to discover product associations from the customers' view point, identify mini-categories that serve as building blocks of project material list, and provide guidance on managing project-level shopping behaviors. In particular, we use Product Network Analysis (PNA) as the primary analytical tool for this study. PNA applies Social Network Analysis (SNA) algorithms to category management and is the automated discovery of relations and key products within a product portfolio.

Methodologically, our research applies network analysis methods to categorize products based on community discovery, a novel and potentially insightful approach to the retailing field. Managerially, findings of this study will facilitate improving consumer-centric category management beyond the traditional market basket analysis [6]. Our results will also provide guidance on designing customized recommendation and promotion systems based on identified project shopping behaviors [7].

The rest of the paper is organized as follows. We overview prior work in Section 2. In Section 3, we describe the data set. In Section 4, we explain the product network construction algorithm. We explore the structure of the constructed network and introduce mini-categories in Section 5. We conclude and outline future work in Section 6.

2 Prior Work

Raeder and Chawla [8] are among the pioneers of product network-leveled analysis. The authors follow an intuitive approach to constructing a network of products from a list of sales transactions: each node in the network represents a product, and two nodes are connected by an edge if they have been bought together in a transaction. Many real-world interaction networks contain communities, which are groups of nodes that are heavily connected to each other, but not much to the rest of the network. It is logical to expect that product networks contain communities as well. Detecting communities in complex networks is known as "community discovery"[9]. In recent years, it has been one of the most prolific sub-branches of complex network analysis, with dozens of algorithms proposed and the agreement within the scientific community that there is no unique solution to this problem given the many different possible definitions of "community" for different applications [10]. Raeder and Chawla [8] focus on community discovery in product networks and show how communities of products can be used to gain insight into customer behavior.

Pennacchioli et al. [10] compare two community discovery approaches: a partitioning approach, where each product belongs to a single community, and an overlapping approach, where each product may belong to multiple communities. The authors apply the approaches to a data set of an Italian retailer and find that the former is useful to improve product classification while the latter can create a collection of different customer profiles. Xie et al. [7] provide a review and comparative study of overlapping community discovery techniques. Videla-Cavieres and Ríos [11] propose a community discovery approach based on graph

mining techniques that distinguishes two forms of overlapping: crisp overlapping, where each product belongs to one or more communities with equal strength; and fuzzy overlapping, where each product may belong to more than one community but the strength of its membership in each community may vary. Kim et al. [6] extend the idea of using only sales transaction data to build product networks by utilizing customer information as well. The authors construct two types of product networks: a market basket network (MBN), which spatially expands the relationship between products purchased together into relationship among all products using network analysis; and a co-purchased product network (CPN), which is extracted from customer-product bipartite network obtained using transaction data. The topological characteristics and performances of the two types of networks are compared.

3 The Data Set

The data set provided to us through the Wharton Customer Analytics Initiative (WCAI) [12], consists of product descriptions and purchase descriptions.

The product part includes descriptions of ca. 111,000 material items, 351 non-material items (such as gift cards, warranties, deposits, rental fees, and taxes), and 71 Sell, Furnish, and Install (SF&I) items that combine materials and services. Since the descriptions of the non-material items are generic and not easy to associate with particular customer projects, we excluded them from our analysis.

The products are organized into a three-level hierarchy of 1,778 subcategories (e.g., *shrub/landscape*), 235 classes (e.g., *live goods*), and 15 groups loosely corresponding to departments (e.g., *grd/outdoor*). The members at each level in the hierarchy are non-overlapping.

The purchase part contains the information of about 11,631,000 sales[1] and 545,000 returns. For each sale and return, we know the product ID, the buyer ID, and the location (store ID and register ID), date, time, quantity, and price of the sale, and discounts, if applicable. The sales recorded in the data set took place over two years between 05/03/2012 and 02/03/2014. 99.6% of the sales were initiated and completed in stores; the remaining sales were made online.

The members at each level in the product hierarchy significantly vary in size. The variance can be estimated in terms of the observed entropy H_1 versus the entropy H_0 of a uniform, homogeneous distribution of member sizes (higher entropy means higher homogeneity). The data set group sizes range from 6 to 25,888 (H_1=3.57 vs. H_0=3.91); class sizes—from 1 to 21,167 (H_1=4.37 vs. H_0=7.88); subcategory sizes—from 1 to 12,355 (H_1=4.55 vs. H_0=10.79). The striking heterogeneity of the hierarchy members makes it hard to treat them as first-order building blocks for further research.

The data set product hierarchy reflects the store organization by departments, sections, and subsections/shelves. While this grouping makes perfect sense from

[1] For the purpose of this study, all items with the same product ID, purchased by the same customer at the same register at the same time, are considered one sale.

the functional perspective (items performing similar functions or intended for similar purposes, such as nails and screws, are shelved together), it does not reveal latent task-oriented connections between products. For example, 91% of *screws* are in the *hardware* and *electrical* groups, but 82% of *screwdrivers* are in the *tools* group, another 18% are in the *electrical* group, and none are in the *hardware*. The assignment of *screws* and *screwdrivers* to different groups (and, therefore, different departments) ignores the fact that both are required for *screwdriving*. As a consequence, by observing the purchase of *screws* as an item from the *hardware* group and a *screwdriver* as an item from the *tools* group, a researcher may not be able to detect that the customer is about to start a *screwdriving* "project."

To circumvent the problems of heterogeneity and lack of support for task- or project-orientated classification, we introduce another level in the data set hierarchy—mini-categories. We later define the mini-categories as structural sub-networks within the overall product network. The product network construction algorithm is described in the next Section.

4 Product Network Construction

A product network [3,6,10] is a graph G reflecting the product co-occurrences in a customer's "basket" [6,8,11]. The graph nodes represent individual material items purchased by customers. Two nodes A and B are connected with an edge if the products A and B are frequently purchased together (not necessarily by the same customer). The existence of an edge between two products suggests a purposeful connection between the products, such as co-suitability for a certain task, as in the *screws* and *screwdriver* example above.

A product network graph G is undirected (if A is connected to B then B is connected to A). It does not contain loops (a node cannot be connected to itself) or parallel edges (A can be connected to B at most once). The graph in general is disconnected—it consist of multiple components, one of which, the *giant connected component* (GCC), may have a substantially bigger size than the others.

Depending on the construction procedure, the graph G can be unweighted or weighted. In the former case, the existence of an edge indicates that the strength of the connection between the two incident nodes (e.g., the likelihood of the two items to be in the same "basket") is simply at or above certain threshold T. In the latter case, the strength of the connection is treated as an attribute of the edge; this way, some edges are stronger than others. A weighted graph can be converted to an unweighted graph by eliminating weak edges and treating strong edges as unweighted. An unweighted, undirected graph with no loops and parallel edges is called a simple graph.

While weighted graphs are more detailed, simple graphs are easier to visualize and comprehend. Many graph processing algorithms (and applications) are optimized for simple graphs. In our quest for mini-categories, which are ambiguously defined, we believe that the benefits of having a more detailed representation of

product interconnections are offset by the fuzzy mini-category detection techniques, and do not outweigh the added complexity of handling weighted graphs. That is why we chose simple graphs as the representation of the product network.

At the first stage of the network construction, we create a graph node for each material item that has been purchased by a customer at least once over the observation period, to the total of 85,865 nodes.

At the second stage, two nodes are connected if the corresponding items have been purchased *together* at least N times. To quantify the concept of *togetherness*, we first observed that the customers are more likely to visit the store every $k = 1, 2, 3 \ldots$ weeks (Fig. 1), which must be caused by the weekly work cycle. We use one week as a natural window span and consider two purchases by the same customer to be in the same "basket" if they were made within seven days (not necessarily within one calendar week).

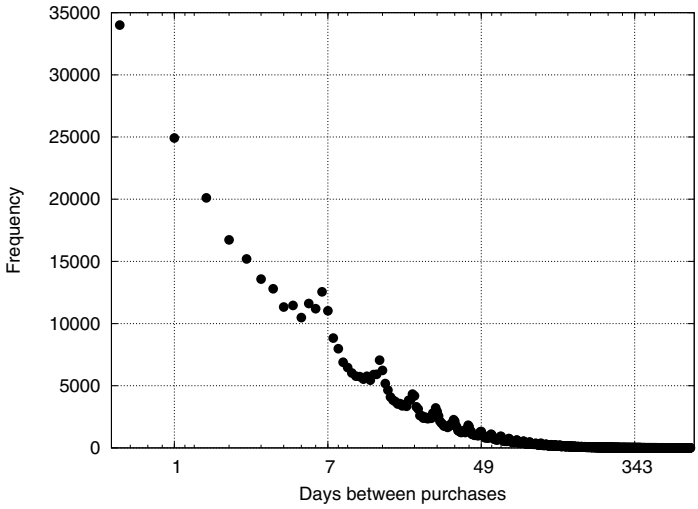


Fig. 1. Days between consecutive purchases by the same customer

The choice of N controls the density and the connectedness of the product network. A bigger N results in a sparse network with many tiny isolated components that cannot be efficiently grouped into mini-categories. A smaller N yields a very dense network, dominated by the GCC and unsuitable for community detection algorithms.

Table 1 presents product network statistics for N=1, 5, 10, and 20: numbers of edges, nodes, isolated single nodes, isolated pairs of nodes, and components; the size of the giant connected component, and the relative volume of sales of the GCC items. The two least dense networks (N=10 and 20) have a subtle GCC and many isolates. The densest network (N=1) essentially consists only of a very dense, nonclusterable GCC. The transition from N=1 to N=5 substantially

Table 1. Product network statistics for N=1, 5, 10, 20. See Section 4.1 for the explanation of 5*.

N	1	5	5*	10	20
Edges	8,066,192	104,643	28,760	26,187	7,126
Nodes	85,865	85,865	85,053	85,865	85,865
Isolated nodes	1,026	67,007	69,619	78,283	82,982
Isolated pairs	71	682	953	494	244
Components	1,107	67,989	71,069	79,051	83,352
Absolute GCC size	84,669	16,215	11,164	5,296	1,677
Relative GCC size	98.6%	18.9%	13.1%	6.2%	2.0%
Sales in the GCC	99.9%	70.0%	51.3%	45.0%	26.0%

reduces the GCC size while preserving its relative sales volume, thus making it possible, without the loss of generality, to disregard the sales of the isolated items. For this reason, we adopted N=5.

4.1 Staples

The resulting product network is a power-law graph with a long-tail degree distribution with $\alpha \approx -1.25$ (Fig. 2). The distribution of sales volumes for individual items also follows the power law[2] with $\alpha \approx -1.06$. Most items are isolated nodes or have fewer than 10 connections. However, there is a number of staples [14] in the tails of the distributions that are (a) frequently purchased from the store and (b) frequently purchased together with other items.

The top 20 staples in the product network are shown in Table 2.

The staples are either not related to any specific projects or are generic and can be related to a multitude of projects. Since staples belong to many "baskets," they lay on many network shortest paths and connect nodes that otherwise would probably be disconnected. The shortest paths induced by the staples, increase graph coupling and lower its modularity, thus eroding potential mini-communities. To minimize the influence of the staples, we eliminate, in the spirit of market basket analysis, 5% of the GCC nodes with the highest degrees—that is, 812 nodes with the degree $d > 45$. The product network G^* with the truncated tail is referenced in Table 1 as 5*.

5 Network Structure and Mini-Categories

A visual inspection of G^* reveals rich internal structure of the product network. In particular, we noticed three major types of structural tiles: dense clique-based communities, sparse stars, and linear chains and pendants—and randomly structured connecting matter. Often, the tiles overlap (e.g., a node can be a leaf

[2] In fact, node degrees and the corresponding sales volumes are correlated with $\rho \approx 0.867$.

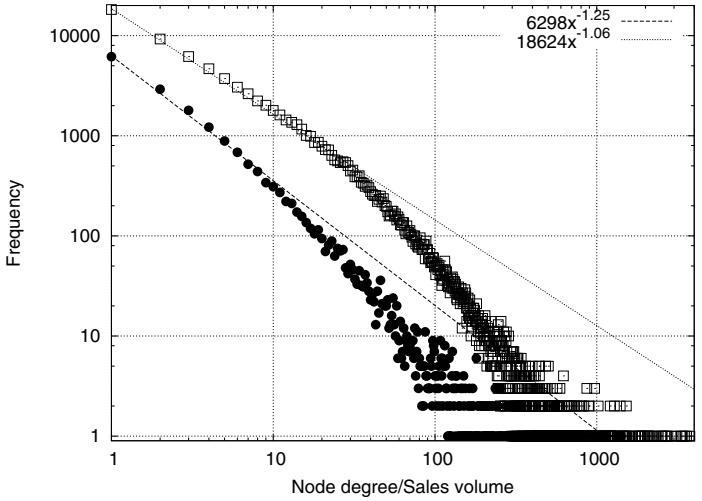


Fig. 2. Node degrees (*circles*) and item sales volumes (*boxes*) in the product network for N=5

Table 2. Top 20 most connected products (staples)

Product	Degree	Sales
2×4-96" premium kd whitewood stud	1,410	3,305
5gal homer bucket	1,333	3,344
9 in plastic tray liner—white	1,049	3,078
e/o bonnie eco prem peatveg herb 5in	1,031	3,756
1/2"×260" ptfe thrd seal tape	986	2,491
20 oz classic coca-cola	943	3,407
scotch blue 1.88" painters 2090	897	2,258
20 oz diet coke	810	2,897
chip 2.0 flat brush	715	2,241
1/2"×4'×8' usg ultralight drywall	681	1,395
better 9 in tray set—8 piece	677	2,051
40 lb topsoil	674	3,498
e/o vegetable peat pot red 5in	634	2,706
16oz gaps & cracks great stuff	613	2,103
plastic bag goods	593	2,529
alex plus white	587	1,453
1 cu ft mg flower & veg garden soil	586	2,753
20oz dasani water	556	2,523
better 9 × 3/8 in knit poly roll 3pk	549	1,524
42gal 3mil contractor trashbag 32pk	549	2,057

of a star and a member of a dense community). We propose an automated procedure for the structural tile extraction.

5.1 Tile Extraction

At the pre-processing stage, all small unconnected components (having fewer than five nodes) are removed from the network. The new network has 12,416 nodes and 26,943 edges.

We define an imperfect star as a connected subgraph of G^* that consists of at least four nodes of degree ≤ 2, connected to a common central node. We allow for a modest number $(n/2)$ of chords in an n-node star, because the graph G^* was constructed through a binarization procedure with an arbitrary chosen threshold and an absence of a connection between two nodes does not imply a zero co-occurrence.

A chain/pendant is a linear sequence of nodes that is connected to anchor nodes at one (pendant) or both (chain) ends. We define an imperfect chain/pendant (a linear tile) as a connected subgraph of G^* that consists of nodes of degree 1 through 3. The nodes of degree 3 introduce defects (chords and mini-stars) but do not significantly distort the linear structure of the subgraphs.

An anchor node is a node that is shared by a linear tile and the rest of G^*. We attached anchor nodes to the incident linear tiles. As a result, we get 5,197 small linear tiles with < 5 nodes and 375 large linear tiles with ≥ 5 nodes. In the spirit of restricting the size of individual tiles to ≥ 5 nodes, we combined the small linear tiles with their larger immediate neighbors.

We used CFinder [5] for the extraction of dense communities. CFinder is based on the Clique Percolation Method: it builds k-cliques—fully connected subgraphs of G^* of size k—and then computes the union of all k-cliques that share $k-1$ nodes pairwise. Clique-based communities have an important advantage over k-cliques: they are less rigidly defined and can absorb more potentially related nodes than a clique, thus improving the tile coverage of G^* and reducing the number of required tiles.

We eliminated communities with < 5 nodes to be consistent with the previously adopted approach to small tiles.

5.2 Coverage Optimization

As a result of the network decomposition, we constructed 5,035 possibly overlapping tiles of three different types: stars (3,553), dense clique-based communities (1,107), and chains/pendants (375). The union of all tiles contains 12,370 product network nodes, with the average coverage of 2.45 nodes per tile. Table 3 shows the summary of the tile coverage (before and after optimization).

The amount of overcoverage (average number of tiles that a node belongs to) can be reduced by optimizing the coverage, identifying essential tiles, and discarding redundant tiles. For optimization, we chose a variant of a greedy maximum coverage algorithm [13]. We start with an empty set of covering tiles.

Table 3. Structural tiles of the product network before and after coverage optimization

Tile type	Count		Node Coverage		Mean Size
	Original	Optimized	Original	Optimized	Optimized
Stars	3,553	289	10,486	5,589	30
Dense communities	1,107	216	5,457	8,123	47
Pendants/chains	375	313	2,065	4,278	17
Total:	5,035	818	12,370	12,274	

At each iteration, we select an unused tile that, if added to the coverage set, minimizes the number of uncovered nodes and increases the number of covered nodes. The process stops when no such tile exists.

The optimization reduced the number of essential tiles to 818—16% of the original tile set (see Table 3). Only 142 nodes remained uncovered by any essential tile. As a result, the average number of nodes per tile increased to 15, and the amount of overcoverage was dramatically reduced (Fig. 3).

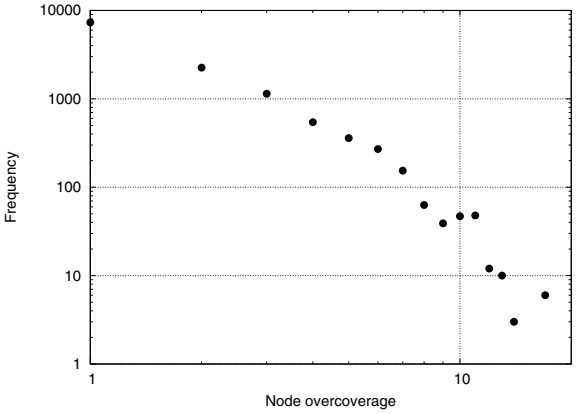


Fig. 3. Overcoverage (number of structural tiles per node) of product network nodes

Figure 4 shows the outlines of three randomly selected average-sized sample tiles of each type.

5.3 Mini-Categories

The optimized tile set contains a reasonable number of members and has a good uniformity. The entropy of the tile size distribution for the set is $H_1 \approx 8.92$ versus $H_0 \approx 9.68$ for the uniform, homogeneous distribution. The collection of essential tiles forms a good structural basis for further research of customer behavior and customer-driven projects.

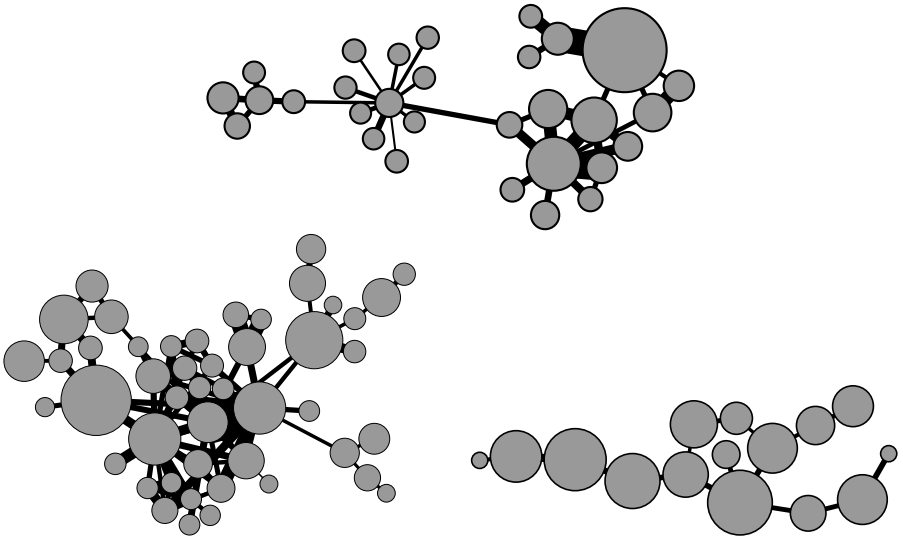


Fig. 4. Outlines of sample tiles: (a) star "*Ice melt and shovels*" (*top*), (b) community "*Alarms and smoke detectors*" (*bottom left*), and (c) chain "*Zinc screws*" (*bottom right*). Node size represents item sale volume, edge thickness—the number of co-occurrences.

From a retailing perspective, different types of structural tiles correspond to different relationships between the products associated with the tile nodes. We call these building blocks mini-communities and suggest that they reflect consumers' view on the product hierarchy.

A cliques-based community (and especially a generating k-clique) is characterized by homogeneity and complete or almost complete connectivity between the nodes. In other words, any product in a community is commonly purchased together with all other products in the community. The products in a community form a topical complementary group [14,15,16], e.g., *alarms* and *smoke detectors*—elements of home security.

On the contrary, a star is heterogeneous. The nodes in a star form two different groups: the high-degree hub (the lead product) and small-degree spokes (the peripheral products). The lead product is frequently purchased together with one or few peripheral products. However, the peripheral products are never or almost never purchased together. The hub with the peripherals form a group of substitutes [14,15,16], e.g., snow removal tools and materials: *ice melt bag* as the lead and *shovels*, *rock salt*, and *sand* as the peripherals (Figure 4a).

Chains/pendants (linear tiles) are perhaps the hardest mini-category to interpret. They describe a set of products that are almost never purchased together, but often purchased pairwise. An almost perfect example of a chain is shown in (Figure 4c): all products in the tile are *zinc flat head philips wood screws* and

differ only in length and number (diameter). Most of the screws are #8 and #10. Any two neighbors differ either in diameter (#8 vs. #10) or length, and the difference between the neighbors is always smaller than between any non-neighbors. We hypothesize that a customer buys a pair of items if she is not sure about the precise values of certain attributes (such as screw dimensions). In other words, a linear tile represents substitutes by ignorance, as opposed to substitutes by choice.

6 Conclusion and Future Work

The goal of this research is to pave the road to the automated identification of consumer projects, based on the available retail data. One possible direction that we explored is to deconstruct the product network into structural tiles that correspond to groups of products—mini-categories.

We built a product network from the purchase data provided by a major nationwide retailer through the Wharton Customer Analytics Initiative (WCAI). A visual inspection of the network revealed three major types of structural blocks: dense clique-based communities, stars, and linear structures (chains and pendants).

We developed a procedure for the automated tile extraction and coverage optimization. As a result, we produced a reasonably uniform in size collection of ca. 800 tiles of all three types that cover the majority of the giant connected component of the product network. We associate each tile type with the nature of the products in the tile: either complements or substitutes.

We believe that the extracted mini-categories represent consumer view on the retail product hierarchy and can be used as an efficient managerial and research tool.

In the future, we plan to study mini-categories as first-class objects, rather than building blocks for possible consumer projects. That way, there will be no need to minimize their count and lump mini-chains into adjacent stars and cliques, thus preserving relative cleanness of the stars and cliques and making them easier to analyze.

We hope that the planned use of structural role extraction algorithms [17] will uncover more tile categories, that, in turn, would yield more retailing-related mini-categories.

Finally, we will look into validating our complement/substitute tile theory using Amazon Mechanical Turk [18] crowdsourcing platform.

Acknowledgments. The authors would like to thank Wharton Customer Analytics Initiative (WCAI) for the provided data set that made this research possible and an anonymous reviewer for the suggestion to use role extraction algorithms.

References

1. Tuli, K.R., Kohli, A.K., Bharadwaj, S.G.: Rethinking Customer Solutions: from Product Bundles to Relational Processes. J. of Marketing 71(3), 1–17 (2007)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proc. of the ACM SIGMOD International Conference on Management of Data, Washington, D.C., pp. 207–216 (1993)
3. Forte Consultancy. Product Network Analysis—the Next Big Thing in Retail Data Mining, `http://forteconsultancy.wordpress.com/2013/02/19/product-network-analysis-the-next-big-thing-in-retail-data-mining/`
4. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast Unfolding of Communities in Large Networks. J. of Statistical Mechanics: Theory and Experiment 10, 10008 (2008)
5. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society. Nature 435(7043), 814–818 (2005)
6. Kim, H.K., Kim, J.K., Chen, Q.Y.: A Product Network Analysis for Extending the Market Basket Analysis. Expert Systems with Applications 39, 7403–7410 (2012)
7. Xie, J., Kelley, S., Szymanski, B.K.: Overlapping Community Detection in Networks: The State-of-the-Art and Comparative Study. ACM Computing Surveys 45(4), 1–37 (2013)
8. Raeder, T., Chawla, N.V.: Market Basket Analysis with Networks. Social Network Analysis and Mining 1(2), 97–113 (2011)
9. Coscia, M., Giannotti, F., Pedreschi, D.: A Classification for Community Discovery Methods in Complex Networks. Statistical Analysis and Data Mining 4(5), 512–546 (2011)
10. Pennacchioli, D., Coscia, M., Pedreschi, D.: Overlap versus Partition: Marketing Classification and Customer Profiling in Complex Networks of Products. In: 2014 IEEE 30th International Conference on Data Engineering Workshops (ICDEW), pp. 103–110. IEEE (2014)
11. Videla-Cavieres, I.F., Ríos, S.A.: Extending Market Basket Analysis with Graph Mining Techniques: a Real Case. Expert Systems with Applications 41(4), 1928–1936 (2014)
12. Wharton Customer Analytics Initiative: Using Purchase History to Identify Customer "Projects." Data Key 3.0, available through WCAI (2014)
13. Johnson, D.S.: Approximation Algorithms for Combinatorial Problems. J. Comput. Syst. Sci. 9(3), 256–278 (1974)
14. Brijs, T., et al.: Building an Association Rules Framework to Improve Product Assortment Decision. Data Mining and Knowledge Discovery 8, 7–23 (2004)
15. Lattin, J.M., McAlister, L.: Using a Variety-Seeking Model to Identify Substitute and Complementary Relationships among Competing Products. J. of Marketing Research 22(3), 330–339 (1985)
16. Elrod, T., et al.: Inferring Market Structure from Customer Response to Competing and Complementary Products. Marketing Letters 13(3), 221–232 (2002)
17. Henderson, K., et al.: RolX: Structural Role Extraction and Mining in Large Graphs. In: Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1231–1239. ACM (2012)
18. Buhrmester, M., Kwang, T., Gosling, S.D.: Amazon's Mechanical Turk: A New Source of Inexpensive, yet High-Quality, Data? Perspectives on Psychological Science 6, 3–5 (2011)

Categorical Framework for Complex Organizational Networks: Understanding the Effects of Types, Size, Layers, Dynamics and Dimensions

Chris Arney and Kate Coronges

United States Military Academy, West Point, NY
{kate.coronges,david.arney}@usma.edu

Abstract. Organizational network modeling can exhibit complexity in many forms to embrace the reality of an organization's processes and capabilities. Networks enable modelers to account for many structural and attributional elements of organizations in ways that can be more powerful than statistical data mining methods or stochastic models. However, the price paid for this increased modeling strength can come in the form of increased complexity, sensitivity and fragility. Traditional network methods and measures can be sensitive to changing, unknown, or inaccurate topology; fragile to dynamic and algorithmic processing; and computationally stressed when incorporating high-dimensional data. Sensitivity and fragility of network models can be managed by setting boundaries around network states, within which specific geometries and topologies can be robustly measured. We propose a categorical framework that identifies such boundaries and develops appropriate modeling methodology and measuring tools for various categories of organizational networks. Categorization of networks along important dimensions such as type, size, layers, dynamics and dimensions provide boundaries of paradigm shifts (from a social scientific perspective) or phase transitions (from physical sciences) -- points at which the fundamental properties or dynamics of the networks change. Not adjusting for these categorical issues can lead to poor methodology, flawed analysis, and deficient results. The purpose of our work is to: 1) develop a framework to enable the construction of a network organizational modeling theory, and 2) identify measures, methods and tools that are appropriate for specific categories (and inappropriate for others) within this field of study. We believe that such a framework can help guide underlying theory and serve as a basis for further formalization of network studies.

1 Background

Network science research has matured rapidly and has moved beyond simple graph models to investigate more realistic and complex models (Brandes, Robins, McCranie, Wasserman, 2013). Through this process, the science of networks wrestles between its development of universal principles and its rapidly increasing array of specialized methodological tools and measures. Many previous sciences have had to resolve these kinds of issues as they matured and grew in utility. Often categorical frameworks

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_19

helped these scientific developments, either temporarily until the science unified its theory and application or as more permanent elements.

In the 1830's, Schleiden and Schwann's discovery that plant and animals were all made of simple units that functioned both as distinct entities and as building blocks capable of self-assembly led to major advances in biological insights (Schwann and Schleiden, 1847). Classification schema went from one where living forms are made up of unique forms of tissues to one that can be reduced to common 'basic units of life'. Not unlike networks, simple recombination of cellular material and connections with other cells result in vastly different appearances and functions. Modern cell theory was born, enabling the science to shift focus from the specific forms of living material, to exploration of common cellular processes such as homeostasis, energy production, and reproduction. Similarly, the 'discovery' that all networks share basic building blocks and are governed by common forces in their construction produced powerful geometric and topological insights that enable us to characterize general processes of networks.

Like the biological sciences though, deeper insights can arise after more sophisticated classification systems are developed to account for the parameters within which these generalized phenomena are meaningful such as the distinctions between plants, animal, fungus, etc. In the next decade, we expect to see network classification schemas that resemble biological taxonomic ranks, with network measures to determine the equivalents of domains, kingdoms, and phylum, to distinguish fundamentally different types of network systems. The work presented here is an incremental step towards that goal.

Like other fields, models for social or organizational networks struggle with the tension between holistic approaches that can capture the appropriate network dynamics and the fidelity achieved in modeling contextually rich information associated with social processes. This bifurcation has lead to a disconnection between theoretical and methodological approaches used to study networks. On the one hand, efforts largely led by physicists and computer scientists focus on the phenomenology and topology of networks – where the goal is to develop a unifying framework to model structural linkages among entities (Barabasi & Albert, 1999; Barabasi, 2009; West, 2011). This search for universality leads to the development and use of general models and methods. Social scientists on the other hand focus on understanding drivers of social processes under specific conditions of influence, learning, and supportive roles of social ties (Borgatti, Mehra, Brass, & Labianca, 2009). This produces a specialization of methodology and can result in complex models. One approach leads to generalizable network properties that result from inherent mathematical relationships, while the other generates a formalization of mechanisms meaningful within specific contexts and exogenous elements.

The two perspectives have contributed to important advances in the new science of networks over the last two decades. We argue here that the field is ready to evolve from these two approaches to one that lies somewhere in between. One of the important consequences of this unresolved tension is that it is not clear in what contexts it is appropriate to apply which network measures and dynamical

assumptions. For instance, the role of high degree nodes or the capabilities of a highly centralized network will be very different in dense verses sparse, layered verses flat, networks. What do measures like centrality or density, or topologies like small world or scale-free mean differentially for networks of ten, thousand, million, or a billion nodes? This line of inquiry is not new, but systematic solutions have not yet been developed. We present a framework that can begin to resolve the dilemma. The critical aspect is to identify points at which models shift from one set of properties and dynamics to another set (analogous to taxonomic classifications in life sciences). These are often called phase transitions (by physicists) and paradigm shifts (by social scientists). At these boundary points, the rules governing the formation and growth of the network are transitioned to a different set of network forces. Our goal is to organize these transitions into a classification system that will guide the use of appropriate network methodologies.

Using organizational networks as a prototype, we present a framework to reflect transitional phases that distinguish the role of structural elements, processing functions, and nodal attributes. We present an outline for a framework that seeks to bridge context-rich processes and generalized phenomena of networks to help build a more trans-disciplinary science. To date, there have been only a few efforts to build a generalizable framework that explicitly attempts to integrate both the social and physical approaches (see Carley, 2003). In our schema, we show categorizations across properties of the network model which may require differing methodologies, measures or tools. This approach recognizes the need for categorizations, but identifies less discrete boundaries than the phenomenologist approach. We seek to identify such categorizations appropriate for organizational networks to include type, layers, dimensionality, size and dynamics. The next step in this research will be to use the various features in this classification system to identify network measures and network dynamical relationships that are appropriate or feasible for those categories.

2 Organizational Networks

The power of network models for managing organizations is affected by a lack of a historical record of successful test cases and the lack of valid (comparable and benchmarked) data. The few data sets and formal assessments that do exist are not satisfying. On the micro scale of organizational analysis, centrality measures can offer meaningful insights into the structure of subgroups and the identity of power-players within the formal organization but often lack details about what these metrics mean for communication, diffusion, or innovation (Borgatti, Carley, and Krackhardt, 2006; Borgatti, 2006). On the macro scale, the dynamics of the topology and the statistics of the nodal or link attributes can help to characterize the overall capabilities of the organization, but the lack of granularity reduces the rigor of the analysis and questions the validity of results, particularly for a

decision-maker in an organizational setting. Depending on the categorization of the model, traditional centrality measures of the known elements of an organization may provide little meaning to understanding the real issues involved with managing an organization. As an example, dark (hidden) organizations and agents are characterized by their ability to coalesce and metastasize subgroups to produce unpredictable events, appear in unanticipated places, and create considerable confusion for outside analysis. Bell (2014) developed a subgroup centrality measure that deals with the specific considerations of dark networks. Nonlinearity is present in most of the properties of dark organizations: high volatility, appearance of randomness, and considerable asymmetry (Taleb, 2012; Xu and Chen, 2008). In many cases, the dark network is embedded within a light network such that differences between legitimate connections and those designed for deceit are often treated together. Ignoring ties between the dark and light components or treating them uniformly are poor strategies for handling these data in a meaningful way. Bell's subgroup centrality measure takes both micro and macro level settings into account by allowing for the division of the network into local (targeted) and global (untargeted) influence. This technique generates centrality rankings that differ substantially from the traditional approaches and therefore are relevant to this type of network. This is an example of developing network measures that are appropriate for specific type of network because it accounts for its unique features. If the network were to move from dark to light, this measure would no longer be appropriate.

3 Organizational Network Classification Schema

Some of the most compelling properties of organizational networks stem from their type, layers, dimensionality, size and dynamics. Examples of these elements are:

- Types: Hierarchical, flat, bright, dark, cooperative team, competitive team
- Size: Number of nodes or links
- Layers: Single or multiple
- Dynamics: Discrete events, continuous evolution, adaptations, resources flows, attribute spreading, controlled
- Dimensionality: Kinds of nodes and links (scalar or vectors), number of attributes of each node and link

Other potential elements to include in such a framework are: Functions of the organization, eg, business, government, service (non-profit), sports, recreational, entertainment, and topology, eg, templated, random, small-world, scale-free, scale-rich, core- periphery, cellular, modular. Sometimes these categories are generalized by modelers into schema such as meta-networks (multi-modal and multi-layered), which can lead to confusion.

3.1 Network Type

Contrasts among types of networks can be characterized in terms of their structure, process, and attributes across organizational function. For instance, dark verses light organizations, cooperative verses competitive teams, and hierarchical verses flat communication will have very different structural properties and will evolve based on distinct dynamical processes. Information about various cognitive and psychological elements of the network, such as types of leadership and decision-making, roles of central actors and individuals on the edge, information flow and influence are also incorporated in the characterization of each category. Proper categorization of these structures and processes can help determine which network measures and concepts can be used effectively (Salas, Cooke, and Rosen 2008; and Stokols et al. 2008). As an example, Table 1 shows a framework for understanding structural and processing elements and potential attributes for organizational networks of several types.

Table 1. Network implications for structural, processing and attributional elements across network types

Network Elements	**Network Type**					
	Hierarchical Organizations	Flat Organizations	Bright Networks	Dark Networks	Cooperative Teams	Competitive Teams
Structural Properties						
Command/ leadership	Role by position, not by talent or need	Roles can be earned but changed as needed	Through competence and need	Ephemeral, hidden and limited, sometimes fragile	Shared responsibility (coach coordinated)	Centralized controller (coach or manager)
Individuals on the Edge	Constrained, limited	Emerge as needed	Empowered, not constrained, produces anti-fragility	Valued and empowered, only when needed	Team relies on all members	Roles are designated but valued
Information	Redundant, stored in many layers	Unstructured and possibly hidden and unavailable	Shared and highly valued	Hidden and protected	Valued and shared	Valued and disseminated as needed

Table 1. (*Continued*)

Processing Functions						
Control/ influence	By directive	Emerges as needed	Through emergence of need and skill	Only as necessary	Emergent, with emphasis on influence by the coach	As determined by the coach/ manager
Decision-making	Only by leaders	By many, but possibly conflicting	By everyone, as appropriate	Uneven, by leader only if necessary	As determined by the coach/ manager with input from everyone, as appropriate	As determined by the coach/ manager
Operational processing	Prescribed, sequential, doctrinal	Emer - gent	Dynamic, concurrent	Scattered, incomplete	Dynamic, concurrent	Efficiency is paramount
Communication	Inefficient, thru Chain of Cmd, as necessary	Varied in form and in availability	Shared and Efficient	Uneven, limited and coded	Shared and Efficient	Shared and Efficient
Attributes						
Intelligence/ wisdom	As valued by leaders	Shared, but risky	Adaptable and eclectic	Hoarded and secret	Coach led with some sharing	Coach is the primary source
Security	As required	Very difficult, often	As needed	Highly valued	Mixed, fragile	Mixed, fragile

3.2 Network Size and Layers

Organizations come in many sizes, from small partnerships to giant conglomerates. So it is not surprising that one methodology does not fit all sizes. Similarly, number of layers in an organizational network model can dramatically change the methodology and measures as well. This next example shown in Figure 2 provides tools and measures to calculate nodal influence in a network with varying size and layers. For this framework, we choose a flat, bright, organization with a scale-free topology, and single mode.

Table 2. Network measures of influence by number of layers and size of network

	Size			
Layers	10	10^3	10^6	10^9
1	Traditional centrality/ centralization measures	Traditional centrality/ centralization measures	Limited centrali- ty measures (degree) & cen- tralization (densi- ty)	Specialized criteria counts and streaming data accumula- tions
2	Average of traditional centrality/ centralization measures	Average of traditional centrality/ centralization measures	Average of limited centrality measures (de- gree) & centrali- zation (density)	Specialized criteria counts and streaming data accumula- tions
3-10	Vector of traditional centrality/ centralization measures	Vector of traditional centrality/ centralization measures	Vector of limited centrality measures (de- gree) & centrali- zation (density)	Probably not feasible
11-∞	Weighted measure of the vector of traditional centrality/ centralization measures	Weighted measure of the vector of traditional centrality/ centralization measures	Weighted measure of the vector of limited cen- trality measures (degree) & cen- tralization (densi- ty)	Not yet feasible

3.3 Multi-Layered Networks

Complexity must be included in the basic theories of network science, especially in the multilayered modeling. Network complexity often rises from the numerous interdependent components interacting in nonlinear, random or disordered ways. Complexity is reflected by 1) many components and layers, 2) interdependency of these components and layers, and 3) interactions that are nonlinear causing the nature of the relationships to be noncumulative or unpredictable. Like many systems, organizational "systems include multiple subsystems and layers of connectivity, and developing a deep understanding of multilayer systems necessitates generalizing traditional network theory" (Kivela et al, 2013). For example, human-capital layers can connect with other dimensions including information flow and informal influence layers of the organization. This layered complexity makes network analysis for organizations a challenging venture that is much like Silver's (2012) description of finding the signal through the noise. In this case, the signal consists of the functionally performing elements of the organization while the noise is the disruption. Our framework accepts the idea that

"the study of multilayer networks has become one of the most important directions in network science" (Kivela et al, 2013).

3.4 Network Dynamics and Dimensions

Capturing the performance effects of network dynamics in an organization is an important element of organizational management. Decision makers are often basing their decisions on predictions and movement toward performance optimality. The next example framework seeks to describe organizational performance measures as categorized by the type of dynamics and the dimension of the nodes and links. Often modelers include multiple modes in organizational models (e.g., employees, units, systems, facilities). Table 3 shows tools to obtain good performance measures for these categories of network models.

Table 3. Organizational performance by dynamics and dimension

Dynamics	**Dimension**		
	Single Mode	Multi-modal	Hypergraph (vector of links)
Discrete event	Time series of a performance measure at event intervals	Time series of a performance measures of various modal accumulations at event intervals	Time series of a specialized performance measure at event intervals
Continuous evolution	Sequence of approximations of a performance measure	Sequence of approximations of a performance measure for various modes	Sequence of approximations of a specialized performance measure
Adaptation (learning network)	Measure of comparison of performance measures at learn- ing stages	Measure of comparison of performance measures at learning stages	Measure of comparison of performance measures at learning stages
Flow of physical resource (conserved)	Rate of flow output	Rate of flow output and other network attributes	Heuristic measures for the rate of flow output and other network attributes
Spreading of an attrib- ute	Rate of attribute adoption	Rate of attribute adoption	Rate of attribute adoption
Controlled dynamics	Measure of comparison of performance measures	Measure of comparison of performance measures	Measure of comparison of performance measures

3.5 Topology and Function

Including structural topology is an evolving element in our framework. Many network structures are either not part of an existing topology or fall into more than one category. Therefore, rigorous use of this element in our framework is problematic. Similarly, network function is both overlapping and incomplete and therefore not yet useful. While we would like to include these elements in our framework, we will hold back their inclusion until more appropriate categorizations are available.

4 Conclusion

There are at least two levels of modeling density to resolve -- the underlying purpose of the model and the methodology in building the model. While good modeling often seeks simplification to enable clear explanation and clarification, the actual goal of the modeler is to find the appropriate level of resolution for the utility and purpose of the model. Models that are too simple miss the necessary detail and nuance of the organization and unnecessarily restrict the functionality and reality of the model. Models that are too detailed, lead to poor solutions or produce entangled results which do not resolve the phenomena being modeled. The best models make difficult ideas easier to understand without missing important ideas caused by over simplifying.

While this paper only introduces some of the ideas and elements of an organizational network modeling framework, it also raises the conversation of the utility of classification as a step in the development of network science as an interdisciplinary science. Using such a framework, we seek to both understand and embrace the resolution in structure, process, data attributes of organizations; to examine the sensitivity, fragility and complexity of network models through adjustments in categorical methodology based on type, topology, function, layers, dimension, size, and dynamics. Ultimately, our goal is to develop a construct to better understand and discover the science of networks.

References

1. Barabasi, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
2. Barabasi, A.L.: Scale-free networks: a decade and beyond. Science 325, 412–413 (2009)
3. Bell, J.: Subgroup Centrality Measures. Network Science 2(2), 277–297 (2014)
4. Borgatti, S.P., Everett, M.G.: Models of core / periphery structures. Social Networks 21, 375–395 (1999)
5. Borgatti, S.P., Mehra, A., Brass, D., Labianca, G.: Network analysis in the social sciences. Science 323, 892–895 (2009)
6. Borgatti, S., Carley, K., Krackhardt, D.: On the Robustness of Centrality Measures under Conditions of Imperfect Data. Social Networks 28, 124–136 (2006)

7. Brandes, U., Robins, G., McCranie, A., Wasserman, S.: What is network Science? Network Science 1, 1–15 (2013)
8. Carley, K.M.: Dynamics Network Analysis. In: Breiger, R., Carley, K., Pattison, P. (eds.) Dynamic Social Network Modelling and Analysis: Workshop Summary and Papers, pp. 133–145. National Research Council (2003)
9. Kivela, M., Arenas, A., Barthelemy, M., Gleeson, J., Moreno, Y., Porter, M.: Multilayer Networks. J. Complex Netw. 2(3), 203–271 (2014); arXiv preprint arXiv:1309.7233 (2013)
10. Salas, E., Cooke, N.J., Rosen, M.A.: On Teams, Teamwork, and Team Performance: Discoveries and Developments. Human Factors: The Journal of the Human Factors and Ergonomics Society 50(3), 540–547 (2008)
11. Silver, N.: The Signal and the Noise: The art and science of prediction. Penguin Books (2012)
12. Schwann, T., Schleyden, M.J.: Microscopical researches into the accordance in the structure and growth of animals and plants. Printed for the Sydenham Society, London (1847)
13. Stokols, D., Hall, K.L., Taylor, B.K., Moser, R.P.: The Science of Team Science: Overview of the Field and Introduction to the Supplement. Am. J. Prev. Med. 35(2S), S77–S89 (2008)
14. Taleb, N.: Antifragile: Things that gain from disorder. Random House (2012)
15. West, B.: Complex Webs (2011)
16. Xu, J., Chen, H.: The topology of Dark Networks. Communications of the ACM 51(10), 58–65 (2008)

Studying Reciprocity and Communication Probability Ratio in Weighted Phone Call Ego Networks

Carolina Ribeiro Xavier[1,2], Vinícius da Fonseca Vieira[1,2], Nelson Francisco Favilla Ebecken[1,2], and Alexandre Gonçalves Evsukoff[2]

[1] UFRJ/COPPE - Federal University of Rio de Janeiro, RJ, Brazil
[2] UFSJ/DCOMP - Federal University of São João del Rei, MG, Brazil

Abstract. Currently, mobile phone calling is one of most widely used communication mode. The records of calls among users can show much about human communication pattern and this pattern can help us to infer about interpersonal relationships. In this work we use CDR (call details record) data for modelling the whole network and choose random nodes for a deep study of their ego networks. In each ego networks we study and discuss the reciprocity of the weight of connections and the correlation between time spend per relationship, number of calls per relationship and their respective reciprocity index.

1 Introduction

Phone calls are one of the main ways in which our contemporary society communicate. Information about phone calls traffic in a phone carrier may hide much information about communication pattern and human relationship. It is known that the behaviour expressed by an individual in specific medium is useful to understand the behaviour in the other media [4].

This work aims at the study of reciprocity in the communication of users of mobile phone based on two measures of interaction: intensity of communication (sum of all phone calls in a month, peer to peer) and frequency of communication (number of phone calls performed in a month, peer to peer). Thus, networks considered in this work are weighted. More specifically, we are interested in the reciprocity index of specific individuals. With this purpose, we will consider the so called ego networks.

We define ego networks as networks consisting of a single actor (ego) connected to alters and the links among those alters. These networks are also known as the neighbourhood networks or first order neighbourhoods of ego. The attraction of ego networks is the ease of collection of data compared with collecting data on whole networks. Information on the alters, including how they are connected, is usually obtained entirely from ego. Such structures can be sampled from large populations and can be used to make statistically significant conclusions about the whole population. There are many areas in which such networks have been studied empirically, for example social support or reciprocity.

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_20

According to Tilburg et al [9], reciprocity in relationships can be interpreted as an indicator of the level of intimacy of such relationship. Tilburg et al [9] also affirm that the unbalancing in the support provided by the individuals involved in a relationship is related to low levels of welfare.

The main goal of this work is to investigate different ego networks, calculating the reciprocity contained in the dyads in order to obtain an index of reciprocity of the relationships ego node in each network. This index enables a comparative study between the level of support that an ego provides and receives. Some insights about the network can be performed based on this index. For instance, one can predict the extinction of a relationship by observing a lack of reciprocity in it.

2 Basic Concepts

Ego networks consist of a focal node ("ego") and the nodes to whom ego is directly connected to (these are called "alters") plus the ties, if any, among the alters. In this paper ego networks will be extracted from the complete network of CDR data.

Assortativity, or assortative mixing is a preference for a network's nodes to attach to others that are similar in some way. Though the specific measure of similarity may vary, network theorists often examine assortativity in terms of a node's degree [6],[7]. In this work we will use the assortativity degree and the assortativity strength for two metrics: #calls strength and duration strength of a node.

The weighted reciprocity for networks is studied in [10] and [2]. However, the reciprocity as originally stated, is only about the dyadic. In order to represent the support received by "Ego", we will take into consideration the mean of the reciprocity metric observed for several nodes.

In the formulation proposed by Wanget al. [10], reciprocity of a dyad is given by Equation 1:

$$R_{ij} = |ln(p_{ij}) - ln(p_{ji})| \tag{1}$$

where $p_{ij} = \frac{w_{ij}}{w_{i+}}$ and w_{ij} is the weight corresponding to the directed edge $i \rightarrow j$, and w_{i+} is the output strength of the i^{th} node, as stated by Barrat et al. [1]. The formulation is symmetric, since $R_{ij} = R_{ji}$.

The correlation between the reciprocities was calculated with Spearman's rank correlation coefficient, because we can not guarantee this correlation is linear. The Sperman's coefficient is a nonparametric measure of statistical dependence between two variables. It assesses how well the relationship between two variables can be described using a monotonic function. Using distinct data values, a perfect Spearman correlation of $+1$ or -1 occurs when each of the variables is a perfect monotone function of the other.

3 Experiments and Discussion

3.1 Dataset

The dataset used in this work consists of phone call records obtained by a phone carrier. The data is composed by a list of the observed interactions between the users of the carrier at each month. Each record shows the anonymized user that performed the phone call(s), the anonymized user that received the phone call(s), the aggregated number of calls performed in a month and the aggregated duration of calls performed in a month. The dataset also contains information about the interaction between the users of the studied carrier and the users of other carriers. However, the interactions involving only the users of the other carriers can not be investigated, limitating a broader view of the network, which can be considered as another motivation in the use of ego networks.

In order to perform the experiments, a graph $\mathcal{G} = (V, E)$ was constructed, in which $v \in V$ is a node that represents a user of mobile phone and $e \in E$ is a directed edge, which represents a connection between two nodes, such that the source node is the caller user and the target node is the called user. The edges have two important attributes: total number and total duration of the phone calls performed from each node $v_i \in V$ to each node $v_j \in V$. These attributes are used for the analysis of reciprocity and distribution of time in the communication of the nodes, in order to understand the behaviour of the relationship involving the individuals.

3.2 Information about the Complete Network

The complete network has 408309 nodes connected by 62214482 edges and, among all the nodes, 113611 correspond to users from the carrier that provided the data. The remainder nodes correspond to users from other carriers that interact to users of the carrier that provided the data. Table 1 shows some global metrics of the network.

We can observe from Table 1 that the measures of maximum indegree and maximum outdegree are very high, and it is unlikely for a person to keep 60000 contacts.

The maximum duration and number of calls associated to an edge is also way beyond what can be expected. These values allowus to conclude that these relationships are not associated to people, but to organizations that use a single phone number shared with several users, which is not explicit in the dataset. The maximum duration is 183598.9 minutes, which is impossible to be associated to a relationship between two common users, since it represents 127 days of phone calls. The same observation can be made when the maximum number of calls (9976) is analysed, since it leads to 332 average daily calls from one individual to one other. These values suggest, in both cases, that these relationships can be associated to telephone stations of two organizations.

The average degree (in/out) of the network is 152, very close to the Dunbar number [3], based on the theory of social brain which states that a person is able

Table 1. Properties of the complete directed network

Max degree in	69466
Max degree out	74252
Mean degree	304
Average degree in/out	152
Max duration	183598.9
Min duration	0.01
Average duration	6.33
Max # calls	9976.0
Min # calls	1.0
Average # calls	4.76
Binary reciprocity	0.349
$\hat{CC}$	0.008
Average shortest paths	2.96
Assortativity degree	-0.008
Assortativity strenght duration	0.052
Assortativity strenght # calls	-0.011

to manage, at most, about 150 "friends". The average aggregated call duration between pairs of vertices in the network is 6.33 minutes in 4.76 calls per month, resulting in 45 seconds per average call. The clustering coefficient ($\hat{CC}$) presented by the network can be considered high and the shortest path distance is low.

Unlike the assortativity values found in social networks studied in other works [7], the correlation found is null for both degree assortativity and strength assortativity (considering number and duration of calls).

The ratio of reciprocal edges in the directed network is nearly 34%, thus, to better understand the reciprocity we also analysed the measures presented in Table 1 for the mutual undirected version of the network. The mutual network was constructed from the directed version previously presented, removing the direction of the relationship. Thus, an edge between nodes A and B only exists if the both edge, from A to B and from B to A exists. The properties of this new edge are calculated by the sum of the properties of the original edges. I.e., the weight of the number(duration) of calls of the undirected edge is the sum of the weight of the number(duration) of calls represented by the edge $\bar{AB}$ to the number(duration) of calls represented by the edge $\bar{BA}$.

Some metrics, extracted from the giant component of the mutual undirected network are presented in Table 2.

In the mutual network, still, the maximum degree (1873) indicates that some nodes do not represent a single person, but organizations, since it is avery high value for any single individual. The average degree of the network shows that the relations are closer in the mutual network and the observed value (58) is consistent to the theory of groups sizes of Dunbar [5], that suggests that an average individual have about 50 distant friends.

The maximum value of call durations is still high (17996), even it is unlikely, the communication of two mobile phones for 10 hours a day. The maximum number of calls does not differ from the values previously observed, since the mutual version represents the sums of the edges.

The assortativity coefficients show that the mutual network is disassortative regarding to the degree, which is not observed in social networks. The network

Table 2. Properties of the complete undirected network

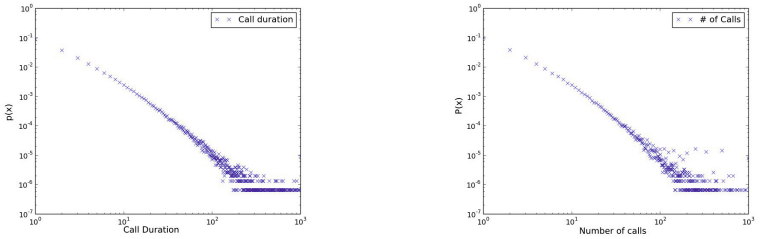
Max degree	1873
Average degree	58
Max duration (min)	17996.36
Min duration (min)	0.01
Average duration (min)	18.50
Max # calls	17796.0
Min # calls	2.0
Average # calls	18.28
Assortativity degree	-0.195
Assortativity strenght duration	-0.088
Assortativity strenght # calls	-0.138
$\hat{CC}$	0.005
Average shortest paths	3.316

is also disassortative regarding strength considering thenumber of calls. On the other hand, the assortativity is almost null considering the strength of duration of calls. The clustering coefficient $\hat{CC}$ is still low and the shortest distance is slightly higher when the non-mutual edges are removed.

After observing Tables 1 and 2, we chose to study the dyads in the ego networks, due to the lack of some properties of networks in the complete directed network and also in the mutual undirected network.

3.3 Probability Ratio Distribution

The probability ratio distribution is the probability of communication from A to B and it is given by the ratio of the total duration of calls in months (or # of calls in month) in calls from A to B, over the total duration of calls (or # of calls) from B. This distribution, like in [8], refers to how ego distribute their attention among several alters. In Fig. 1 the distribution of this ratio is illustrated.



(a) Duration (b) # Calls

Fig. 1. Probability ratio distribution

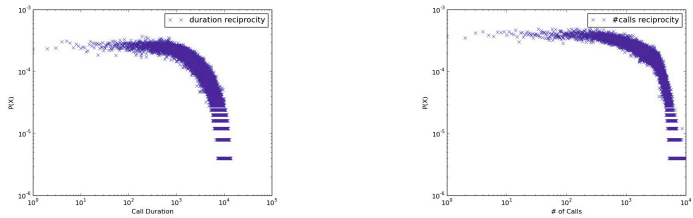
In Fig. 1, we can see that both distributions are very similar, one can think in a strong correlation between the two metrics. In other words, "if people talks with other for long time, these people call for this others several times", but Spearman correlation show us that it is not true.

For each list of probability ratio, we rank the alters by the weighed # of calls probability ratio and duration calls probability ratio, calculate the coefficient between two ranked list. We can observe that the set of people that a certain node talk for a large duration is not the same set of people that this node talk with more frequency. For a sample of 2000 nodes in network the average correlation coefficient is very low, only 0.09, which shows almost a null correlation between both probability ratios.

3.4 Reciprocity

The binary reciprocity is not enough to state, considering a specific individual, if his contacts are reciprocal regarding the dedication to the relation. For instance, if an individual A calls an individual B for 300 minutes per month in average, and the individual B calls the individual A for 10 minutes, it can lead us to a misinterpretation that the individual A invests much more to the relation than individual B. However, this simplification can be unfair if the individual A spends 1000 minuter per month in phone calls, while the individual B spends only 60 minutes.If we take this into account, the individual A dedicates 1% of his time to his relation with the individual B, while B dedicates more than 16% of his call time to the individual A.

For a sample of 2000 nodes we plot the distribution of the reciprocity index of them in a graph, represented by Figure 2. Again, we can see that both distributions are very similar, but Spearman correlation coefficient shows us that the correlation between them is not strong.



(a) Duration Reciprocity (b) # Calls Reciprocity

Fig. 2. Reciprocity Distribution

For each list of reciprocity, we rank the alters by the weighed # calls reciprocity and duration calls reciprocity and calculated the coefficient between two ranked list. From these results, we observe that the person that a certain node is reciprocal in aggregated duration in month is not usually the some person that this node is reciprocal in frequency. For the whole sample, the average correlation of reciprocity is 0.145, which can be considered as a weak correlation. In other hand, the average coefficient with other metrics present lower values, which will be discussed in the following sections.

3.5 Reciprocity vs. Probability Ratio

We analyse the correlation coefficient between the reciprocity using two metrics and the probability ratio in the same metric and we observe that there is no significant correlation. For reciprocity and probability ratio using the duration of calls the average of calls was only 0.03 and for reciprocity and probability ratio using number of calls the correlation coefficient was considerably low 0.002. This coefficients show that the reciprocity of the chosen metrics is not correlated to the respective probability ratio. In others words, egos are not reciprocal with alters who he talks for a long time nor with alters to which he talks with more frequency.

4 Conclusions and Future works

In this work we study the reciprocity in the communication of users of mobile phone based on two measures of interaction: intensity of communication (sum of all phone calls in a month, peer to peer) and frequency of communication (number of phone calls performed in a month, peer to peer). The networks considered in this work are weighted and we use a sample of 2000 ego networks to calculate the reciprocity.

We calculate and discuss the correlation between reciprocity and two measures and conclude that the reciprocity in call duration is weakly correlated to reciprocity in number of calls peer to peer.

The probability ratio of communication between two nodes is not correlated to the reciprocity neither in number of call nor call duration. And unlike the reciprocity using two measures, the probability ratio distributions using the two measures are not correlated.

We find a lack of a metric of the reciprocity index of a node, which could help us to understand people behaviour but we can confront the results extracted from the networks to social and psychological theories. Because that, for future work we intend to propose a new metric in order to measure the reciprocity received by a node, observing the probability ratio for distinguish closer contacts from distant contacts.

Acknowledgements. The authors thank the financial support of the agencies: Capes, FAPERJ and CNPq.

References

1. Barrat, A., Barthelemy, M., Pastor-Satorras, R., Vespignani, A.: The architecture of complex weighted networks. PNAS USA 101(11), 3747–3752 (2004) 202
2. Chawla, N.V., Hachen, D., Lizardo, O., Toroczkai, Z., Strathman, A., Wang, C.: Weighted reciprocity in human communication networks. CoRR (2011) 202
3. Dunbar, R.I.: The social brain hypothesis. Brain 9(10), 178–190 (1998) 203

4. Haythornthwaite, C.: Social networks and internet connectivity effects. Information, Communication and Society 8(2), 125–147 (2005) 201
5. Hill, R.A., Dunbar, R.I.M.: Social network size in humans. Human Nature 14(1), 53–72 (2003), http://www.liv.ac.uk/evolpsyc/Hill_Dunbar_networks.pdf 204
6. Newman, M.E.J.: Mixing patterns in networks. Phys. Rev. E 67(2), 026126 (2003) 202
7. Newman, M.E.: Assortative mixing in networks. Phys. Rev. Lett. 89(20), 208701 (2002) 202, 204
8. Saramaki, J., Leicht, E.A., Lopez, E., Roberts, S.G.B., Reed-Tsochas, F., Dunbar, R.I.M.: Persistence of social signatures in human communication. PNAS 111(3), 942–947 (2014) 205
9. Tilburg, T.V., Sonderen, E.V., Ormel, J.: The measurement of reciprocity in egocentered networks of personal relationships: A comparison of various indices. Social Psychology Quarterly 54(1), 54–66 (1991) 202
10. Wang, C., Lizardo, O., Hachen, D., Strathman, A., Toroczkai, Z., Chawla, N.V.: A dyadic reciprocity index for repeated interaction networks. Network Science 1, 31–48 (2013) 202

NetSci High: Bringing Network Science Research to High Schools

Catherine Cramer[1], Lori Sheetz[2], Hiroki Sayama[3,4], Paul Trunfio[5], H. Eugene Stanley[5], and Stephen Uzzo[1]

[1] New York Hall of Science
[2] Network Science Center, US Military Academy at West Point
[3] Collective Dynamics of Complex Systems Research Group, Binghamton University
[4] Center for Complex Network Research, Northeastern University
[5] Center for Polymer Studies, Boston University
ccramer@nysci.org, Lori.Sheetz@usma.edu

Abstract. We present NetSci High, our NSF-funded educational outreach program that connects high school students who are underrepresented in STEM (Science Technology Engineering and Mathematics), and their teachers, with regional university research labs and provides them with the opportunity to work with researchers and graduate students on team-based, year-long network science research projects, culminating in a formal presentation at a network science conference. This short paper reports the content and materials that we have developed to date, including lesson plans and tools for introducing high school students and teachers to network science; empirical evaluation data on the effect of participation on students' motivation and interest in pursuing STEM careers; the application of professional development materials for teachers that are intended to encourage them to use network science concepts in their lesson plans and curriculum; promoting district-level interest and engagement; best practices gained from our experiences; and the future goals for this project and its subsequent outgrowth.

Keywords: Network science and education, educational outreach, teaching and learning network science, high school student research, NetSci High.

1 Introduction

Educational systems worldwide are not keeping up with the explosion in the big data and data-driven sciences that inform us about vital trends, have the potential to empower us to solve our greatest social and environmental challenges, and increasingly affect our lives. This gap is coinciding with an escalation in the complexity of the kinds of biomedical, socio-economic, environmental, and technological problems science is addressing, along with the ability to gather and store the subsequent vast amounts of data (American Association for the Advancement of Science (AAAS) 1990, Watts 2007). The skills needed by the 21st century STEM workforce include:

G. Mangioni et al. (eds.), *Complex Networks VI*,
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_21

- The ability to interact with large amounts of data. Facility with visual metaphors and granularity for both static and dynamic data streams is needed in order to see patterns in complex data.
- The ability to understand the changing role of models. The higher-order thinking associated with model development allows both exploratory and inductive skills to be used to identify general patterns and characterize their behavior across a wide range of differing environments and processes.

Students in the STEM "pipeline" need to be prepared for this new reality as they enter the modern day workforce and higher education. However, exposure to these data-driven science skills is unavailable to most primary and secondary school students. Furthermore, summer or academic year research experiences for high school students under researchers' guidance are often inaccessible to disadvantaged young learners. Such lack of access sends students down a path that is devoid of opportunities to fully participate in advances in our modern society.

Network science has emerged as a possible solution. It is a promising way to address data-intensive real-world problems, employing graph theory, statistical analysis and dynamical systems theory to large, complex data sets, seeking patterns and leveraging them against large-scale knowledge management and discovery in business, medicine, policymaking, and virtually all complex science disciplines today. Network science is being used to understand everything from the human brain, to the origins of cancer, to the growth of cities, to our impact on the environment (Barabasi 2002, Pastor-Satorras and Vespignani 2001, Lazer et al 2013). Network science demands that we revise our thinking about what kinds of technical and process skills are needed to design, create and explore these emerging and accumulating data and technological structures.

We believe that network science can provide a novel pathway for high school students to learn about traditional topics across many disciplines, including social studies, science, computer science and technology. Many of the problems explored through a network science approach are in the everyday experience of students, such as the network flow of air traffic, interconnectivity of coupled networks in political and social systems, and human networks as seen through technology activities such as Facebook and Twitter.

To test this solution – using network science to close the skills gap – we have developed and are running "NetSci High", a regional educational outreach program designed to empower high school students and teachers to harness the power of network modeling and analysis, resulting in a more holistic, dynamic understanding of the "interdependence" among components and the evolution of relationships among various things around us. NetSci High provides interventions in STEM teaching and learning that directly address the need for twenty-first century skills while targeting female, minority and economically disadvantaged students. It provides an advanced, alternative pathway to developing rigorous skills-based curricula, resources and programs that utilize the rapidly growing science of complex networks as a vehicle through which students can learn computational and analytical skills for network-oriented data analysis, as well as how these skills can lead to breakthroughs in solving large-scale, real-world problems. NetSci High explores innovative approaches that, as

our work demonstrates, can capture the interest and imagination of underrepresented populations to explore science research problems using computational tools and methods (Buldyrev et al. 1994, Cohen et al. 2000, Trunfio et al. 2003).

2 What Is NetSci High?

NetSci High began as a small pilot project in 2010 with financial support from the Office of International Science and Engineering at the US National Science Foundation (NSF) as well as a corporate donation from BAE Systems. The first year of NetSci High (2010-2011) was run as a competition for high school research posters. Seven student projects were conducted through collaboration between participating high schools and their local research labs in New York City, Boston, and Binghamton, NY. Their posters were reviewed by the Scientific Committee, and the students and teachers of the two winning posters were supported to attend the NetSci 2011 conference in Budapest, Hungary, in June 2011. All of the seven posters were presented at the poster sessions of the conference. The posters were also displayed at the Eighth International Conference on Complex Systems in Boston on June 26-July 1, 2011. The second year of NetSci High (2011-2012) was run as scholarships offered to high school student teams. Two student teams participated from the Binghamton area. Those teams were offered a scholarship to attend the NetSci 2012 conference in Evanston, IL, in June 2012, and to present their posters.

This pilot program paved the way to a much larger NSF-funded ITEST Strategies project, "Network Science for the Next Generation," or NetSci High. Since 2012, Boston University, the New York Hall of Science, SUNY Binghamton and United States Military Academy at West Point have been collaborating on this ITEST Strategies project, which provides opportunities for disadvantaged high school students to participate in cutting-edge network science research. This project bridges information technology practice and advances in network science research to provide career and technical education opportunities for young people underrepresented in data-driven STEM.

The goals of NetSci High are the following:

1. Improve computational and statistical thinking and stimulate interest in computer programming and computational scientific methods by providing students and teachers with opportunities to create and analyze network models for real-world problems through a mentoring and training program.
2. Increase students' potential for success in STEM in a technical career or college through applied problem solving across the curriculum using tested units of instruction that clarify complex STEM topics and provide new applied approaches for critical thinking in STEM.
3. Prepare learners for 21st century science and engineering careers through the use of data-driven science literacy skills, and motivate them to elucidate social and scientific problems relevant to the disciplines and to their lives.
4. Develop curricular resources that help learners achieve the following set of basic skills that are crucially needed to succeed in the data-driven work environment in the 21st century:

- *Ability to synthesize*, seek and analyze patterns in large-scale data systems;
- *Gain facility with data visualization*, filtering, federating, and seeking patterns in complex data;
- *Understand the changing role of models,* higher-order thinking, emphasizing exploratory skills to identify and characterize behavior of patterns in differing environments;
- *Use network science and statistical approaches* to break down traditional silos in order to compare and contrast processes across domains;
- *Build data fluency* to be able to identify, clean, parse, process and apply appropriate analysis skills to large quantities of data;
- *Gain facility with data mining and manipulation* with increasingly semantic and statistical approaches, superseding logic models for searching and comparing data; and
- *Understand the role of data sharing*, collaboration, interoperability of tools and data types, along with skills in using collaborative tools and methods to maximize data discovery.

3 Contents and Materials

NetSci High has developed and implemented a rich, experiential, research-based program for disadvantaged high school students, science research graduate student mentors, and high school STEM teacher mentors throughout New York State and Boston, Massachusetts. This project works to close the gap between the teaching and learning of STEM disciplines and STEM practice, and to prepare the next generation of the STEM workforce to conduct a mode of research that differs markedly from that currently mandated by K-12 curricula and educational practice.

The program includes a 2-week intensive summer workshop and an academic year research program utilizing collaborative IT tools, plus periodic special workshops, industry lab tours, and participation in the International School and Conference on Network Science (NetSci).

Organizers have assembled teams of high school students from New York State and Boston area Title 1 schools, plus their science teachers and graduate students from network science research labs, to spend a year collaborating on cutting-edge research on a network science topic of their choosing. The research component of this project focuses on the efficacy of intensive training and support of high school student teams and an academic year of research with cooperating university-based network science research labs; the labs' participation is facilitated by a graduate student who learns valuable mentorship skills as part of the experience. Because the network science field is relatively new, much of this research is novel, with practical implications.

Each yearly cohort of students begins their experience by participating in a summer residency-workshop led by network science faculty and researchers (Fig. 1). This summer workshop is an intensive two-week experience, at which student teams, their teachers and graduate student mentors are immersed in learning network science concepts and programming languages such as Python, NetLogo and JavaScript; applying

network analysis tools such as NetworkX and Gephi; attending hands-on workshops and talks given by top network science researchers; and collaboratively brainstorming about research questions that will form the basis of the year's research projects.

During the academic year students refine their coding and programming skills, conduct their research, visit their host research labs, and have regular weekly meetings with graduate student mentors. Layers of support and mentoring throughout the academic year come from graduate students in partnering labs, high school teachers and project staff, as well as online and face-to-face field trips, meetings, seminars and work sessions. After their research is complete the teams prepare their findings for publication and presentation.

Fig. 1. NetSci High New York teams gather for a conference at the close of the July 2014 workshop at Boston University

Evaluation of this model to date has indicated that it is extremely effective for students in incentivizing and achieving success in mastering and applying data-driven STEM skills to real problem solving. It is an empowering and engaging pathway into data and computational literacy and computer programming skills. Further planned evaluation will aim to assess participants' higher education and career choices and their relevance to STEM fields. The subsequent evaluative results will provide

substantive evidence of the return on investment (ROI) for funding priorities and future growth of the project.

More information can be found at the NetSci High websites (https://sites.google.com/a/nyscience.org/netscihigh/ and http://www.bu.edu/networks/). Professional development and workshop resources including curricular modules can be found at the workshop website (http://www.bu.edu/networks/workshop/). Additional resources for network science education can be found on the NetSciEd website (https://sites.google. com/a/nyscience.org/netscied/).

4 Accomplishments

Over the past four years, high school students have worked on a wide variety of research projects through the NetSci High program. Table 1 shows the list of project titles, listed in chronological order.

Table 1. List of titles of NetSci High student research projects

2010-11
- A Comparative Study on the Social Networks of Fictional Characters
- Academic Achievement and Personal Satisfaction in High School Social Networks
- Does Facebook Friendship Reflect Real Friendship?
- Inter-Species Protein-Protein Interaction Network Reveals Protein Interfaces for Conserved Function
- The Hierarchy of Endothelial Cell Phenotypes
- Preaching To The Choir? Using Social Networks to Measure the Success of a Message
- Identification of mRNA Target Sites for siRNA Mediated VAMP Protein Knockdown in *Rattus Norvegicus*

2011-12
- A Possible Spread of Academic Success in a High School Social Network: A Two-Year Study
- Research on Social Network Analysis from a Younger Generation

2012-13
- Interactive Simulations and Games for Teaching about Networks
- Mapping Protein Networks in Three Dimensions
- Main and North Campus: Are We Really Connected?
- High School Communication: Electronic or Face-to-Face?
- An Analysis of the Networks of Product Creation and Trading in the Virtual Economy of Team Fortress 2

2013-14
- A Network Analysis of Foreign Aid Based on Bias of Political Ideologies
- Comparing Two Human Disease Networks: Gene-Based and System-Based Perspectives

- How Does One Become Successful on Reddit.com?
- Influence at the 1787 Constitutional Convention
- Quantifying Similarity of Benign and Oncogenic Viral Proteins Using Amino Acid Sequence
- Quantification of Character and Plot in Contemporary Fiction
- RedNet: A Different Perspective of Reddit
- Tracking Tweets for the Superbowl

NetSci High has facilitated sending a group of high school students and teachers from New York City to NetSci 2011 in Budapest, Hungary; a group from Endwell and Vestal, NY to NetSci 2012 in Evanston, IL; and a group from Vestal, NY to NetSci 2014 in Berkeley, CA. The high school student teams presented their work at poster sessions at all of these conferences. High school student research has also been published in peer-reviewed journals such as PLOS ONE (Blansky et al. 2013).

During the Spring 2014 semester, professional development training in network science concepts and tools was provided to the entire 9th grade faculty at Chelsea CTE High School in New York City and faculty from Newburgh Free Academy. The faculty were then encouraged to apply these concepts to meet objectives in their current lesson plans.

Historically each of the program elements have been initiated by network science researchers approaching school district administrators and teachers with ideas for research programs, professional development, workshops, and events all to bring network science concepts, tools and resources into the high school as an apparatus for learning. While the districts appreciated the successes realized by the participating students and teachers, the districts had not developed a deep enough understanding of network science to grasp the full potential of using network science as a curriculum tool. In Fall 2014 district administrators reached out to mentor teachers requesting development of a high school level network science elective course. This request represents a fundamental shift at the district level. The districts are now reaching out to the network science community seeking more resources to bring to their students.

To more widely disseminate the success of this program and to meet the demand for expanding the role of network science in education (a demand coming both from the high school educational community as well as the community of network scientists), the partners initiated a symposium at NetSci 2012 called Network Science in Education (NetSciEd). Since then these symposia have been held annually in the U.S. and Europe and have led to a significant rise in interest in educational and learning applications of network science and the subsequent formation of the NetSciEd community. The NetSciEd community undertook the development and articulation of a set of Network Literacy Essential Concepts that all citizens should know by the time they graduate from high school. These can be found on the NetSciEd website (https://sites.google.com/a/nyscience.org/netscied/). Moreover, for the first time, Networks and Education will be an official strand at the 2015 International NetSci Conference to be held in Zaragoza, Spain, in June 2015 (http://netsci2015.net/).

5 Conclusions and Future Work

As described above, NetSci High has made significant educational impacts on regional high school students and teachers, and is also prompting strong social commitments from the Network Science community as a whole (Harrington et al 2013, Sanchez and Brandle 2014). It aims to address the challenge of transforming the way we educate our citizens in order to keep pace with not only the amount of data we collect, but to appreciate how network science identifies, clarifies, and solves complex 21st century challenges in the environment, medicine, agriculture-urbanization, social justice and human wellbeing. This project provides a pathway to integrate science research and programming skills for high school students who would not otherwise have these opportunities. Additionally, this project encourages high school teacher mentors to broaden their STEM understanding and informs their current teaching in terms of content and practice.

Through evaluation and remediation in our current NetSci High project, we have identified the following successful strategies for bringing network science into high school teaching and learning:

- Original student and teacher research projects are not only possible, but form an essential incentive and commitment for participants to remain engaged in and to bring projects to completion.
- While there is significant interest in broad collaboration among teachers in different domains, they prefer to start with small, easily definable curriculum units or lessons that can be implemented within a class.
- It is possible to train a broad spectrum of students and teachers in enough computer programming (e.g., Python or R) to use sophisticated network analysis tools within programming environments.
- A supportive community and consistent mentorship are essential to success.
- Teachers can and have assumed an active leadership role in mentoring students who are engaged in network science research, provided the right supports are in place.
- Students and teachers are remarkably innovative in terms of how they develop and pursue project-based learning approaches in network science.
- It became immediately apparent in the first year of the project that participating teachers were most effective if they had the same level of training as their students: they want to be active mentors, rather than co-learners side by side. As the project progressed and interactions with teachers became deeper and more meaningful and teachers took on new roles and pursued their own interests and took ownership of network science approaches, a path to scalability began to emerge.

NetSci High is still developing, and there are a number of aspects on which further development and expansions are needed. To accomplish this, we are looking for support and collaboration from the entire Network Science community. Over the next few years, we plan to achieve the following in order to make this successful program more organized, more scalable and more accessible to everyone on the globe:

- Refine our learning materials, publish a Network Science Workshop Training Manual, and develop network science mobile teaching kits. Such field resources will be a blueprint for future training of participants as well as disseminating and replicating our work.
- Expand on successful live network science professional development workshops for high school teachers and develop interdisciplinary network science curriculum.
- Promote new projects on data mining of educational data and using network science to understand the performance of educational institutions. There is an increasing amount of work looking at the organizational structure of education through a network science lens, particularly at how we might mine student data and use network analysis to determine the impact of churn on organizational structure and efficaciousness in schools and districts.
- Finalize the Network Literacy Essential Concepts that all citizens should know. This will be used to create a framework for developing curriculum that can better support data-driven STEM than is currently possible, and will support Next Generation Science Standards (NGSS Lead States 2013) and Common Core standards (National Governors Association Center for Best Practices 2010).
- Expand the professional development, student research and curriculum development projects that benefit from a global community of scientists and policymakers who see network science as an accessible entry point for vital computational, data literacy and algorithmic skills.
- Maintain and increase dialog with the private sector to expand support for initiatives in network science in teaching and learning and engage STEM professionals in awareness and participation in this work.
- Author an accessible network science e-book for general readership.
- Establish a network science e-badging system for the entire network science teaching, learning and research community.
- Host a network and data science festival for the public.
- Develop international partnerships with network science researchers and educators outside the US and promote similar educational activities at international scales.

Acknowledgments. The NetSci High program is supported by the US National Science Foundation through the Cyber-enabled Discovery and Innovation program (CDI) and the Office of International Science and Engineering (OISE) (Award # 1027752) and the Innovative Technology Experiences for Students and Teachers program (ITEST) (Award # 1139478/1139482), as well as a corporate donation from BAE Systems.

References

American Association for the Advancement of Science (AAAS), Project 2061: Science for all Americans. Oxford University Press, New York (1990)

Barabási, A.-L.: Linked: How everything is connected to everything else and what it means. Plume (2002)

Blansky, D., Kavanaugh, C., Boothroyd, C., Benson, B., Gallagher, J., Endress, J., Sayama, H.: Spread of academic success in a high school social network. PLOS ONE 8(2), e55944 (2013)

Buldyrev, S., Parshani, R., Paul, G., Stanley, H.E., Havlin, S.: Catastrophic cascade of failures in coupled networks. Nature 464, 7291 (2010)

Cohen, R., Erez, K., Ben Avraham, D., Havlin, S.: Resilience of the inter-net to random breakdowns. Phys. Rev. Lett. 85, 4646 (2000)

Harrington, H.A., Beguerisse-Díaz, M., Rombach, M.P., Keating, L.M., Porter, M.A.: Teach network science to teenagers. Network Science 1(2), 226–247 (2013)

Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabàsi, A.-L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., Jebara, T., King, G., Macy, M., Roy, D., Van Alstyne, M.: Computational social science. Science 323, 721 (2009)

National Governors Association Center for Best Practices, Council of Chief State School Officers, Common Core State Standards. National Governors Association Center for Best Practices, Council of Chief State School Officers, Washington D.C. (2010)

NGSS Lead States, Next Generation Science Standards: For States, By States. The National Academies Press, Washington, DC (2013)

Pastor-Satorras, R., Vespignani, A.: Epidemic spreading in scale free networks. Phys. Rev. Lett. 86, 3200 (2001)

Sanchez, A., Brandle, C.: More network science for teenagers. arXiv:1403.3618 (2014)

Trunfio, P., Hoffman, M., Shann, M.: Partnerships between graduate fellows and Boston area high school teachers. Presented at Annual meeting of American Chemical Society, New York, September 7 (2003)

Watts, D.J.: A twenty-first century science. Nature 445(7127), 489–489 (2007)

From Criminal Spheres of Familiarity to Crime Networks

M. Oliveira[1], H. Barbosa-Filho[1], T. Yehle[2], S. White[3], and R. Menezes[3]

[1] BioComplex Laboratory, Department of Computer Sciences,
Florida Institute of Technology, Melbourne, Florida, USA
{moliveira,hbarbosa,rmenezes}@biocomplexlab.org
[2] School of Computing, University of Utah, Salt Lake City, Utah, USA
tobin.yehle@utah.edu
[3] College of Arts and Sciences, University of North Carolina at Chapel Hill,
Chapel Hill, North Carolina, USA
sarahw14@live.unc.edu

Abstract. We have never lived in a safer world. After peaking around 1985, both violent crime (homicide, robbery, assaut and rape) and property crimes (burglary, larceny and vehicle theft) are on a downward trend; from 1993 and 2012 crime activity has dropped by more than 40% (total number of crimes). Despite the good news, crime is still prevalent in most large cities. FBI reports that in 2013 there were about 3,098 crimes per 100,000 habitants in the USA, with 2,730 of them being property crimes and 367 violent. What most people can agree is that one preventable crime is one crime that should not have taken place. The unveiling of the structure of criminal activity can lead to a better understanding of crime as a whole which in turn can help us provide better protection to our citizens. We demonstrate in this paper that crime follows a very intersting spatial community pattern regardless of the type of crime, criminal activity aggregates in communities of well defined sizes. We believe the results of this paper is a first step towards a theory of crime modeling using network science.

Keywords: Crime Networks, Crime Structure, Crime Analysis, Community Structure.

1 Introduction

The understanding of crime activity has for a long time puzzled government officials, law-enforcement officers, and researchers. A well-performed study on crime structure may have direct benefits to people's lives as it can lead to safer cities. According to the FBI Annual Crime Report [28], the USA is today much safer than it used to be in the 80s and 90s with about half of the number of crimes per 100,000 inhabitants, but still higher than the levels we enjoyed in the 60s and also higher than many countries in Europe. Indeed crime rate is dropping but the understanding of crime as a complex system can lead to further gains in public safety.

Law enforcement tends to be reactive and many times a step behind criminal activity. What if we could change this "game"? What if we could give the police an edge by making them understand criminal structure and perhaps prevent some activity before

G. Mangioni et al. (eds.), *Complex Networks VI,*
Studies in Computational Intelligence 597, DOI: 10.1007/978-3-319-16112-9_22

it takes place? This is becoming reality in this big-data world we live in. The change in crime rate from the 60s to today can probably be explained by a technology lag. In the 60s, we had a smaller population and hence crime was easier to understand and prevent with "manual" approaches. As the population grew, our ability to effectively keep track of what was going on diminished and consequently crime rate ramped up. More recently, we have seen technology catching up via the use of data analysis and mining. What if we could do more? Like many complex systems we believe there is a structure that governs the interactions of criminals. This paper is an initial step towards the understanding of this structure.

Most of the works in crime structure start from the premise that crime is a consequence of factors such as wealth (or lack of) [14], education levels [17], age [19], and many others. However, more recently we have seen scientist starting to look at structure in particular social networks, as a way to explain the existence of crime in certain neighborhoods [7,8] but to our knowledge scientist are yet to look at the structure of spatial distributions of crime. Few have attempted to look at spatial data and analysis in the context of crime control [1] with most of the studies being related to understanding the emergence of hotspots of crime [11,25]. In this paper we show that the use of hotspots to understand crime spatial structure misses important features that can be better represented and analyzed using networks. In fact, we show that crime networks built from spatial data about crime location appears to reveal social structures when the spatial resolution is high. Our results show that hidden in the distribution of crime (hotspots) is a social structure that may be related to the social network of criminals or the social network of people affected by crimes. In this paper we show how we can uncover this structure.

2 Related Work

Crime is a complex issue and many factors affects its occurrence including: sociological, economic, psychological, biological, philosophical and even religious factors [12].

With regards to crime structure two theories in criminology can be highlighted: the *routine activity* and *social disorganization* theories. The former argues that criminal activity occurs at the convergence of three things: a potential offender, a lack of guardianship or supervision, and a target [5]. The latter contends that criminal activity is the result of the social and physical environments of the neighborhood at hand [32]. Both theories seek to model crime phenomena using spatial and geographical context.

The aforementioned opportunistic nature of the routine activity theory supports that criminal activity typically occurs in the sphere of familiarity of the criminal. Despite this sphere of familiarity being peculiar to the individual, areas of high traffic, such as downtown areas, lie within the sphere of familiarity of many individuals; it is feasible that these criminals with the same sphere of familiarity are aware of each other. This aspect is also related to the fact that criminals typically commit crimes within a short distance from their home [15].

Metropolitan areas are typically organized by regions of different land uses such as: residential, commercial, and industrial use. The presence of types of crimes differs between these land uses; neighborhoods with residential housing and no commercial

businesses are perceived as *safe* and non-residential land uses are correlated with an increase in criminal activity [10]. Non-residential areas are typically found to have higher traffic in comparison to residential areas, consequently they witness to more crime [31]. Non-residential land use, such as shopping centers or public parks, coincides with an increase in *foreign* or non-residential presence. This presence of such *strangers* negatively impacts a neighborhood's social structure [24].

Street network (from layout) are not only correlated with an increase in crime incidence but additionally have a relationship with the typical *journey-to-crime* length of an offender [15]. Roadways and public transportation link together different areas of a criminal's sphere of familiarity and facilitate travel outside of a criminal's immediate neighborhood. The type of crime can affect the *journey-to-crime* length. For example, violent crime trips are shorter in length than property crime trips [15].

Despite the understanding we have of crime activity, its causes and consequences, recent studies continue to look at spatial crime analysis using approaches related to the formation of hotspots [20]. Additionally, there has been many efforts that tries to analyze crime activity in light of the existence of social networks. Many studies have looked into characteristics of ties such as their strength [23], the frequency of ties [6], and the race and gender of those with more ties [30,22]. Yet, these studies rarely consider the structure of the overall network and they assume the existence of some information regarding the social structure of criminals. However, this is not always possible and, in fact, such structure may not be available. Law enforcement datasets rarely include information about criminals acquaintances and when they do, the reliability of such information is doubtful.

The approach we propose then is to focus on the *journey-to-crime* [15] and build networks out of the distance between crimes. Rather than social networks we have crime networks where nodes represents actual crimes and links between crimes related to a distance (or sphere) between the crimes. Our results are important because we demonstrate that a spatial networks of crime appear to contain information about the social structure of the people involved in the criminal activity.

3 Constructing Network of Crimes

The network creation mechanism is based on the geographical proximity between crimes. Two events are *connected* if they occurred within a certain distance. This network creation model is as simple as possible. In fact, the mechanism is the same as used to generate random geometric graphs [9].

Notice that the connection definition we are using here is basically the same as in the context of geometric graphs and does no presume any actual relationship between the events other than their proximity. Therefore, the network structures are going to be fully determined by the spatial distribution of the crimes. Each point in our dataset can be seen as the location of a person—in this particular case, an offender— at a given moment, in a similar manner as checkins in geolocated social networks [21] or mobile phone activities in Call Detail Records datasets [29,26]. The main difference however is that in our data there is no individual-level identification.

Although such data could evidently yield a higher-resolution analysis, we decided to focus on the coarse-grained spatial distribution of crimes. The rationale for this is

twofold: (1) in this paper we aim to uncover network structures (possibly) embedded within the spatial distribution of crimes; (2) for practical reasons, we based our analysis exclusively on publicly available data and hence we do not use any individual-level information.

That said, the theoretical basis supporting our approach are grounded mainly on two principles, both very well documented in the criminology literature:

1. For most crime trips, the distance from the offender's home to the crime location is relatively short and the probability of an offender committing a crime decays with the distance from their home [13,15];
2. Offenders tend to live near to their associates and long-distance ties are rare [18].

Not surprisingly, these characteristics conform to two behaviors largely observed in general human dynamics: (1) most of our trips are for short distances and very long jumps are less likely to occur [27,29] and (2) the probability of finding a social tie between two individuals decays as a power function of the distance [16,2,29]. Therefore, it is plausible to assume that patterns on spatial distribution of crimes should emerge from the convolution of both the individual and social level dynamics.

4 Experimental Results

4.1 Spatial Distribution of Crimes

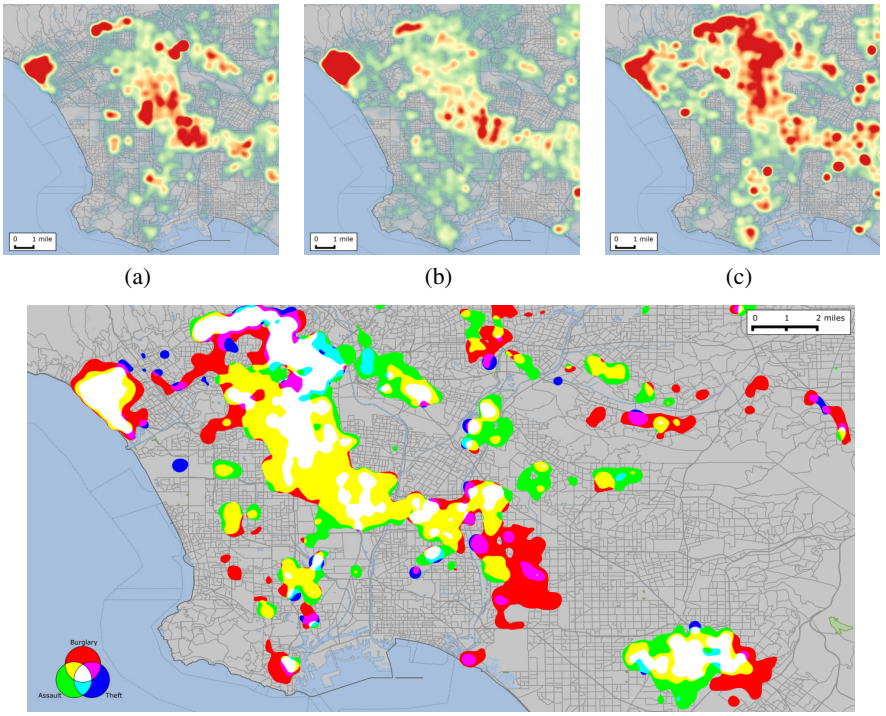
Hot spots of crimes do not occur uniformly in a region. This aspect of the criminal activity can be visualized in the heatmaps depicted in Figures 1(a-c) from the Los Angeles area. This type of map, which shows the places where most crimes were committed, is widely used as a tool to understand the emergence of hot spots as well as to elaborate law enforcement strategies. These heatmaps are geolocalized histograms that allow a prompt analysis of the crime frequency in a specific region. For example, as stated, the aforementioned maps show that there are certain sub-regions with high criminal activities placed across the Los Angeles area. These maps in Figure 1(a), 1(b) and 1(c) depict the placement of the hot spots regarding assaults, burglaries and thefts, respectively. Their analyses suggest that these types of crime have particular arrangements in the region and that they may occur due to different kinds of crime activity dynamics.

However, such maps do not allow analyses beyond the criminal activity frequency of a region. An example of this insufficient data description is that although these visualizations make possible to see many different hot spots together, there is no information about their relationships nor the overall structure that may enable the emergence of the hot spots. Actually, this structural analysis is carried out more by the viewer of the map than brought by the heatmap as a tool. Nevertheless, this structural information can be useful to understand underlying mechanisms in criminal phenomena. For instance, although the examination of Figure 1(a), 1(b) and 1(c) suggests that these particular kind of criminal activities have different dynamics across the region, this comparison may neglect similar underlying mechanisms related to the emergence of hot spots.

In order to capture the similarity of different types of crimes to subsequently analyses, Figure 1(d) is elaborated in such way that only the hottest spots of each kind of

Table 1. The criminal hot-spot mixing in the Los Angeles metropolitan area presents the coexistence of different criminal activities regions. This mixing reveals that the hot spots of thefts usually happen in companion to burglaries and assaults. On the other hand, the other two seem to be less linked to other crimes, resulting in more independent hot spots. In the table below the letters indicate the types of crime, hence A & B represents the region of the overlap of assauts and burglaries.

Assault (A) (green)	Burglary (B) (red)	Theft (T) (blue)	A & B (yellow)	A & T (cyan)	B & T (magenta)	A & B & T (white)
20%	24%	3%	26%	3%	5%	19%



(a) (b) (c)

(d)

Fig. 1. The places where crimes occur are not uniformly distributed in a region. The heatmaps of these events, in the Los Angeles metropolitan area, for different types of crime, assault (a), burglary (b) and theft (c), help the realization that hot spots of crime exist, but the approach is not adequate to carry out structural analysis of the crimes. These heatmaps together can help us visualize the different placements of the hot spots when different crimes are taken into account. The coincidence map (d), an overlap of the hottest spots from these heatmap, shows that thefts tend to happen in places where other crimes are also intensively happening, while burglaries and assault may occur more independently.

crime are considered. The hottest spots are the ones that the crime frequency is two standard deviations higher than the average frequency of this type of criminal activity, thus the map does not present any intensity interval. The intersections between these spots are shown in the map by different colors, as described in the map legend. The rationale of this visualization is to understand the hot-spot mixing in the region, *i.e.* the places where different crimes are concentrated. In the Los Angeles metropolitan area the mixing of the criminal activities coexistence are related to the colors in the map and their percentage is shown in Table 1.

The hot-spot mixing indicates that assaults, burglaries and thefts tend to coexist as hot spots in Los Angeles area. This finding may hint to the existence of some similarities in possible underlying mechanisms that lead to the emergence of these hot spots. Conversely, assault and burglary do present some particularities that allow them to occur more independently across the area considered. In other words, these results suggest that these different types of crime seem to have a core behavior as well as particular behaviors. Regardless of this analysis, heatmaps look at crime frequency and are not enough to assess such underlying mechanisms.

4.2 Microinteractions and the Spatial Distribution of Crimes

Complex networks of a particular class often share several common topological features. For example, a social network is expected to have a high coefficient of clustering while having a short average path length. On the other hand, technological networks such as the Internet tend to have a hierarchical topology.

The existence of such structural and topological patterns plays a central role in order to have a better understanding of the various phenomena and real dynamics driven by one or more network structures. This is especially important when the complex network underlying a particular phenomenon can not be observed directly.

In this section, we seek to extract and identify possible network structures beyond the spatial geometric network itself that we built. When we build a network by simply connecting points geographically close to a distance d, this network will have features of a spatial or geometric network.

4.2.1 Clustering Coefficient

Many complex networks are characterized by a high clustering coefficient. That is specially true for geographically constrained networks where the characteristic link length is bounded up to a distance d. In a spatial network, the global clustering coefficient is expected to increase as a function of the distance d from fully disconnected nodes (when $d = 0$) to a single clique of size N (when $d \to \infty$) where N is the population size. However, neither of these two extremes are of much help in understanding a complex phenomenon such as the dynamics behind criminal activities. Hence, there must be a characteristic radius d (or a function $f(d)$) where the underlying networks unvail themselves.

In such spatial geometric networks, the clustering coefficient is a function of the connection threshold d and should increase monotonically with it. To test this hypothesis we analyzed the changes in the structure of the network for small increments in d starting from $d = 0.02$ miles to $d = 3.2$.

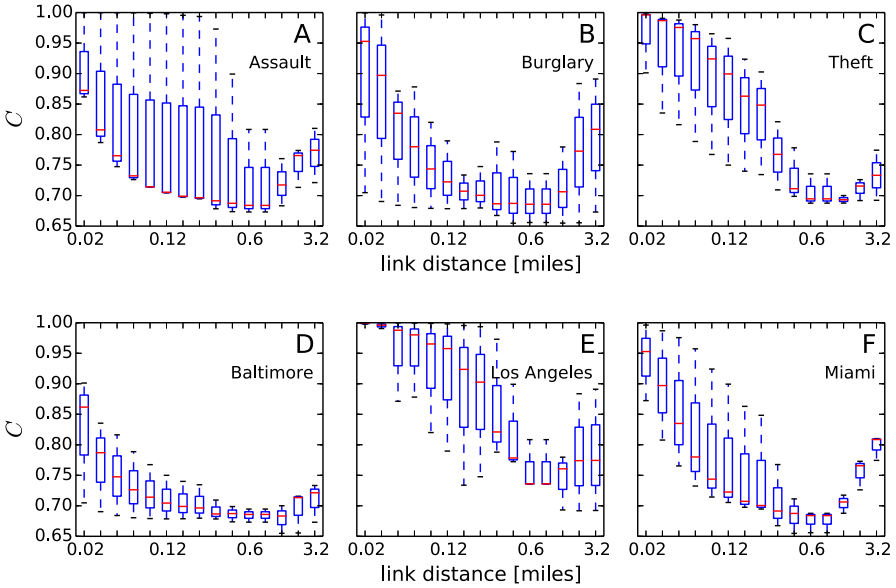


Fig. 2. The plots depict the evolution of the global clustering coefficients by the different linking distance threshold d. The top row shows the clustering coefficient for each type of crime, *assault*, *burglary* and *theft*. The bottom row shows the clustering coefficient for each of the three metropolitan areas. The correlation between d and the clustering coefficient suggest a marked structural change in the network with a critical point $0.4 \leq d \leq 0.8$ miles. Even though the actual shape of the curves varies over different networks, in all of them, the minimum clustering degree was reached in the region close to $d \approx 0.6$.

What was unexpected however is a gradual decrease observed in the clustering coefficient for a particular range of d (as in Figure 2), deviating from the characteristics of a spatial network [4,33] whose clustering coefficient should increase monotonically with d once the spatial boundaries are growing and the longer links are becoming more frequent.

It is also noteworthy the fact that the clustering coefficient reached its minimum for $0.4 \leq d \leq 0.8$, for different cities and crime types suggesting that the networks are undergoing a phase transition for some critical value of $d \approx 0.6$. This behavior could be related to the case in which for very small values of d, the spatial constraints does not play a role anymore and therefore the remaining network structure could result from some other dynamic factor. To test such hypothesis, we investigate what other structural characteristics are also changing with d by comparing their properties for $d < 0.4$ and $d > 0.4$.

4.2.2 Degree Distribution

One next natural step would be an analysis to the degree distribution of the networks, assessing how good they fit to a heavy-tailed distribution. The rationale here is that

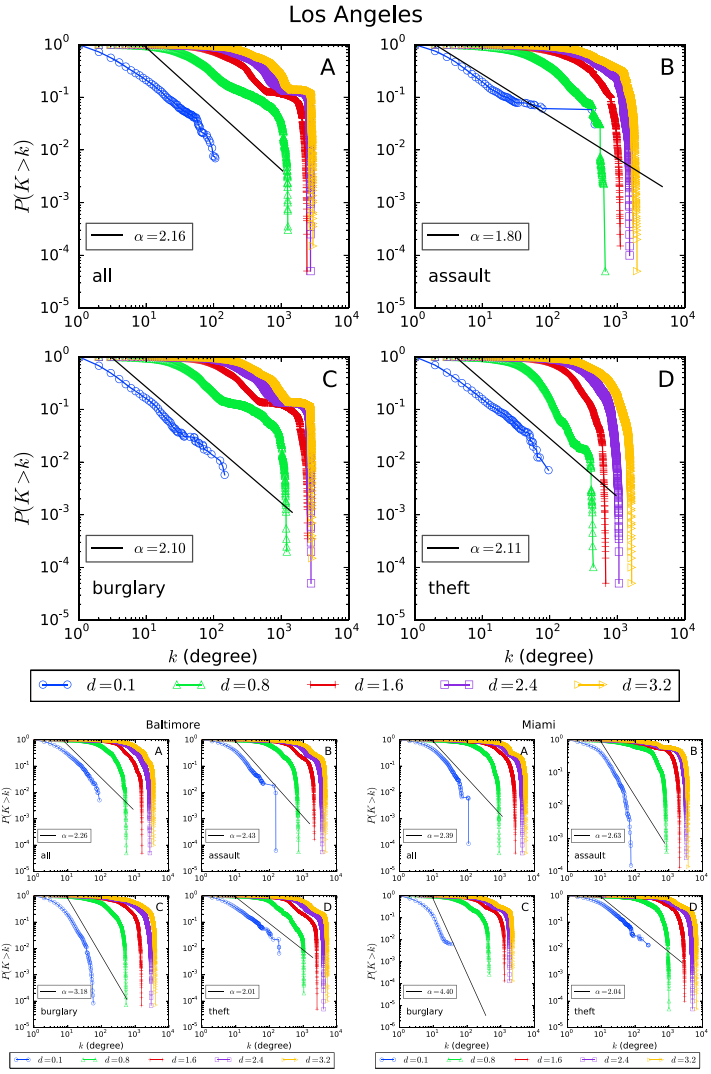


Fig. 3. The cumulative degree distribution of the networks exhibit a strong pattern accross different cities and crime types. In all the networks we investigated, the degree distribution exhibited a heavy tail but only up to a critical value of $d = \delta$. Beyond this point the heavy tail vanishes. On the other hand, for $d < \delta$ almost all the networks had degree distribution in agreement with a power law with exponent $\alpha \approx 2.1$. Straight lines are shown as a guide.

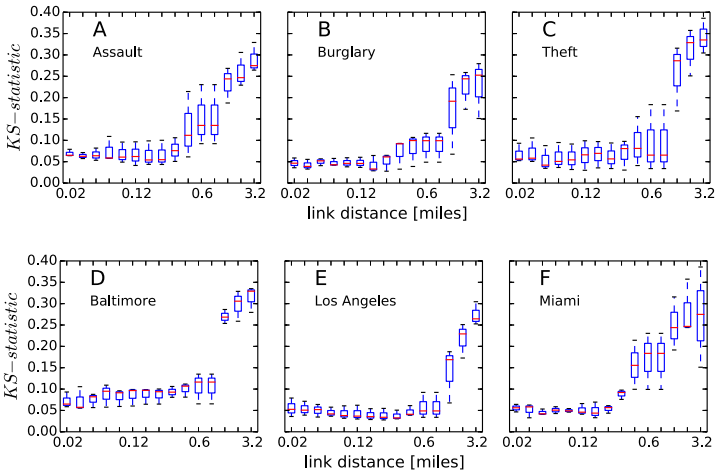


Fig. 4. Kolmogorov-Smirnov distance from the empirical degree distribution to a theoretical power law for different metropolitan areas and crime types. The KS statistic supports our claim that the degree distributions of the crimes networks follow a power-law distribution up to a certain value $d = \delta$.

a heavy tailed degree distribution is a key signature of some interesting complex networks such as social networks [3]. On the other hand, this property does not hold for other classes of networks, including spatial networks [4] which could indicate that the networks are not just undergoing structural transformations but also their signatures are transitioning from of one class of network to another.

From Figure 3, the linking threshold capable of producing networks with heavy-tailed degree distribution happens when $d = 0.1$. Another interesting result is that the power-law exponents of most of the networks have an exponent $\alpha \approx 2.1$ in agreement to the characteristic exponent of scale-free networks.

Although the cumulative degree distributions were consistent with the findings about the clustering coefficients in Section 4.2.1, this analysis is not sufficient to assess the correlation between the value of d and the goodness of fit of a power law to the degree distribution. For this task we used the Kolmogorov-Smirnov test to check for which ranges of d the power-law distribution presents a good fit to the degree distribution. Figure 4 depicts the KS distance from nodes degree cumulative distribution function to a theoretical power-law distribution. However, it is important to emphasize that our focus is not to determine whether the degree distribution is indeed a power law but rather to assess the intervals for the parameter d for which the degree distribution agrees to or deviates from a heavy tailed.

The KS test confirmed our hypothesis that the degree distribution for values of d beyond a certain point have no interesting feature. Based on the test results with KS, for $d > 0.8$ we witness an abrupt increase in the distance from the degree distribution to the power law, in agreement with the results previously found.

It is clear that these tests are not sufficient to prove that the networks emerging for small d are actually the social networks of criminals. In fact, what we are arguing instead is that the dynamics that produced the spatial distribution of crimes result from a

combination of influences of two complex systems: the social dynamics and spatial constraints. However, when analyzing the network of crimes in a high resolution where the characteristic edge length is very short, our results suggest that the observed network no longer behaves as a spatial network and starts to display characteristics observed also in social networks.

5 Conclusion and Future Work

In this paper we looked at the structure of crime in urban environments and demonstrated that one may be able to use spatial networks [4] to extract social information. This seems to be quite clear to case of crime. Our results show that in higher spatial resolutions (less than a mile), network of crimes appear to contain information of the social structure of the individuals involved in the criminal activity. One questions that arrises here is do other spatial networks could also contain social information. We are currently working on other datasets.

In addition to the contribution of showing that social information may be extracted from spatial networks, Our work may be used in the decision-making process of law enforcement officials. We have mentioned earlier that in many instances, the law enforcement agencies may not have in their datasets social information about the criminals and that sometimes the information is incomplete. We believe further work on our approach may lead to the ability of reconstructing these structures. As is, the work can already help decision making because theories from network science can tell us which nodes to focus if we want to disrupt the network; the social structure of crime can be used as a way to understand where the police should focus.

There are several points that need to be studied further. One of the main points is the possibility of defining a scaling law for different types of crimes. Our results appear to show that the social structure emerges at slight different scales depending on the type of crime. However one needs to understand the other variables that may play a role in this such as city demographics and city layout, to name a few.

The test on other cities may also be useful. We tested with 3 cities in the USA. We have not used any variable that is particular to the USA and we have no reason to believe the approach would not be applicable to other places. However we intend to apply the same approach to cities in South America and Europe.

References

1. Anselin, L., Cohen, J., Cook, D., Gorr, W., Tita, G.: Spatial analyses of crime. Criminal Justice 4(2), 213–262 (2000)
2. Backstrom, L., Sun, E., Marlow, C.: Find me if you can: improving geographical prediction with social and spatial proximity. In: Proceedings of the 19th International Conference on World Wide Web (2010)
3. Barabási, A.-L., Albert, R.: Emergence of scaling in random networks. Science, 11 (October 1999)
4. Barthélemy, M.: Spatial networks. Physics Reports 499(1), 1–101 (2011)
5. Beavon, D., Brantingham, P., Brantingham, P.: The Influence of Street Networks on the Patterning of Property Offenses. Crime Prevention Studies (1994)

6. Bellair, P.E.: Social interaction and community crime: Examining the importance of neighbor networks. Criminology 35(4), 677–704 (1997)
7. Browning, C.R., Dietz, R.D., Feinberg, S.L.: The paradox of social organization: Networks, collective efficacy, and violent crime in urban neighborhoods. Social Forces 83(2), 503–534 (2004)
8. Calvó-Armengol, A., Zenou, Y.: Social networks and crime decisions: The role of social structure in facilitating delinquent behavior. International Economic Review 45(3), 939–958 (2004)
9. Dall, J., Christensen, M.: Random geometric graphs. Physical Review E 66(1), 016121 (2002)
10. Foster, S., Wood, L., Christian, H., Knuiman, M., Giles-Corti, B.: Planning safer suburbs: Do changes in the built environment influence residents' perceptions of crime risk? Social Science & Medicine 97, 87–94 (2013)
11. Furtado, V., Melo, A., Coelho, A., Menezes, R.: A crime simulation model based on social networks and swarm intelligence. In: Proceedings of the 2007 ACM Symposium on Applied Computing, pp. 56–57. ACM (2007)
12. Guarino-Ghezzi, S., Treviño, A.J.: Understanding Crime: A Multidisciplinary Approach. Elsevier (2010)
13. Harries, K.: Mapping crime: Principle and practice. Technical report, U.S. Department of Justice - Office of Justice Program, Washington, DC, US (1999)
14. Kennedy, B.P., Kawachi, I., Prothrow-Stith, D., Lochner, K., Gupta, V.: Social capital, income inequality, and firearm violent crime. Social Science & Medicine 47(1), 7–17 (1998)
15. Levine, N., Lee, P.: Journey-to-crime by gender and age group in manchester, england. In: Crime Modeling and Mapping Using Geospatial Technologies, pp. 145–178. Springer (2013)
16. Liben-Nowell, D., Novak, J.: Geographic routing in social networks. Proceedings of the National Academy of Sciences of the United States of America, 1–6 (2005)
17. Lochner, L., Moretti, E.: The effect of education on crime: Evidence from prison inmates, arrests, and self-reports. Technical report, National Bureau of Economic Research (2001)
18. Malm, A.E., Kinney, J.B., Pollard, N.R.: Social Network and Distance Correlates of Criminal Associates Involved in Illicit Drug Production. Security Journal 21(1-2), 77–94 (2008)
19. Marvell, T.B., Moody Jr., C.E.: Age structure and crime rates: The conflicting evidence. Journal of Quantitative Criminology 7(3), 237–273 (1991)
20. Murray, A.T., Grubesic, T.H.: Exploring spatial patterns of crime using non-hierarchical cluster analysis. In: Crime Modeling and Mapping Using Geospatial Technologies, pp. 105–124. Springer (2013)
21. Noulas, A., Scellato, S., Lambiotte, R., Pontil, M., Mascolo, C.: A tale of many cities: universal patterns in human urban mobility. PloS One 7(5), e37027 (2012)
22. Rountree, P.W.: A reexamination of the crime-fear linkage. Journal of Research in Crime and Delinquency 35(3), 341–372 (1998)
23. Sampson, R.J.: Networks and neighbourhoods: The implications of connectivity for thinking about crime in the modern city. In: McCarthy, H., Miller, P., Skidmore, P. (eds.) Network Logic: Who Governs in an Interconnected World?, pp. 105–124. Demos, London (2004)
24. Sampson, R.J., Groves, W.B.: Community structure and crime: Testing social-disorganization theory. American Journal of Sociology, 774–802 (1989)
25. Short, M.B., Brantingham, P.J., Bertozzi, A.L., Tita, G.E.: Dissipation and displacement of hotspots in reaction-diffusion models of crime. Proceedings of the National Academy of Sciences 107(9), 3961–3965 (2010)
26. Simini, F., Maritan, A., Néda, Z.: Human mobility in a continuum approach. PloS One 8(3), e60069 (2013)
27. Song, C., Koren, T., Wang, P., Barabási, A.-L.: Modelling the scaling properties of human mobility. Nature Physics 6(10), 818–823 (2010)

28. The Federal Bureau of Investigation. Crime in the united states (cius). Technical report, Department of Justice (2013), http://www.fbi.gov/about-us/cjis/ucr/ucr
29. Wang, D., Pedreschi, D., Song, C., Giannotti, F., Barabási, A.-L.: Human mobility, social ties, and link prediction. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2011, p. 1100. ACM Press, New York (2011)
30. Warner, B.D., Rountree, P.W.: Local social ties in a community and crime model: Questioning the systemic nature of informal social control. Soc. Probs. 44, 520 (1997)
31. White, G.F.: Neighborhood permeability and burglary rates. Justice Quarterly 7(1), 57–67 (1990)
32. Willits, D., Broidy, L., Gonzales, A., Denman, K.: Place and Neighborhood Crime: Examining the Relationship between Schools, Churches, and Alcohol Related Establishments and Crime. Technical report, Institute for Social Research (March 2011)
33. Wong, L., Pattison, P., Robins, G.: A spatial model for social networks. Physica A: Statistical Mechanics and its Applications, 1–23 (2006)

Author Index