Thesis for the Degree of Doctor of Philosophy

Improving Community Detection Methods for Network Data Analysis

Farnaz Moradi

Division of Networks and Systems
Department of Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden 2014

Improving Community Detection Methods for Network Data Analysis
Farnaz Moradi
ISBN: 978-91-7597-041-7

Doktorsavhandlingar vid Chalmers tekniska högskola
Ny serie nr 3722
ISSN: 0346-718X

Technical report 112D
Department of Computer Science and Engineering

Division of Networks and Systems
Chalmers University of Technology
SE-412 96 GÖTEBORG, Sweden
Phone: +46 (0)31-772 10 00

Author e-mail: `moradi@chalmers.se`

Printed by Chalmers Reproservice
Göteborg, Sweden 2014

ABSTRACT

Empirical analysis of network data has been widely conducted for understanding and predicting the structure and function of real systems and identifying interesting patterns and anomalies. One of the most widely studied structural properties of networks is their community structure. In this thesis we investigate some of the challenges and applications of community detection for analysis of network data and propose different approaches for improving community detection methods.

One of the challenges in using community detection for network data analysis is that there is no consensus on a definition for a community despite excessive studies which have been performed on the community structure of real networks. Therefore, evaluating the quality of the communities identified by different community detection algorithms is problematic. In this thesis, we perform an empirical comparison and evaluation of the quality of the communities identified by a variety of community detection algorithms which use different definitions for communities for different applications of network data analysis. Another challenge in using community detection for analysis of network data is the scalability of the existing algorithms. Parallelizing community detection algorithms is one way to improve the scalability of community detection. Local community detection algorithms are by nature suitable for parallelization. One of the most successful approaches to local community detection is local expansion of seed nodes into overlapping communities. However, the communities identified by a local algorithm might cover only a subset of the nodes in a network if the seeds are not selected carefully. The selection of good seeds that are well distributed over a network using only the local structure of a network is therefore crucial. In this thesis, we propose a novel local seeding algorithm, which is based on link prediction and graph coloring, for selecting good seeds for local community detection in large-scale networks.

Overall, mining network data has many applications. The focus of this thesis is on analyzing network data obtained from backbone Internet traffic, social networks, and search query log files. We show that mining the structural and temporal properties of email networks generated from Internet backbone traffic can be used to identify unsolicited email from the mixture of email traffic. We also show that a link based community detection algorithm can separate legitimate and unsolicited email into distinct communities. Moreover, we show that, in contrast to previous studies, community detection algorithms can be used for network anomaly detection. We also propose a method for enhancing community detection algorithms and present a framework for using community detection as a basis for network misbehavior detection. Finally, we show that network analysis of query log files obtained from a health care portal can complement the existing methods for semantic analysis of health related queries.

Keywords: Networks, Community Detection Algorithms, Overlapping Communities, Seed Selection, Misbehavior Detection, Spam, Medical Query Logs

Preface

This thesis is based on the work contained in the following publications:

- Farnaz Moradi, Tomas Olovsson, Philippas Tsigas, "*Towards Modeling Legitimate and Unsolicited Email Traffic Using Social Network Properties*," in *Proceedings of the 5th Workshop on Social Network Systems (SNS'12)*, pp. 9:1 - 9:6, ACM, Bern, Switzerland, April, 2012.
- Farnaz Moradi, Tomas Olovsson, Philippas Tsigas, "*An Evaluation of Community Detection Algorithms on Large-Scale Email Traffic,*" in *Proceedings of the 11th International Conference on Experimental Algorithms (SEA'12)*, Lecture Notes in Computer Science Vol.: 7276, pp. 283 - 294, Springer-Verlag, Bordeaux, France, June, 2012.
- Farnaz Moradi, Tomas Olovsson, Philippas Tsigas, "*Overlapping Communities for Identifying Misbehavior in Network Communications,*" in *Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'14),* Lecture Notes in Computer Science Vol.: 8443, pp. 398-409, Springer-Verlag, Tainan, Taiwan, May, 2014.
- Farnaz Moradi, Tomas Olovsson, Philippas Tsigas, "*A Local Seed Selection Algorithm for Overlapping Community Detection,*" in *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'14),* Beijing, China, August, 2014.
- Farnaz Moradi, Ann-Marie Eklund, Dimitrios Kokkinakis, Philippas Tsigas, Tomas Olovsson, "*A Graph-Based Analysis of Medical Queries of a Swedish Health Care Portal,*" in *Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi'14),* pp. 2–10, Gothenburg, Sweden, April, 2014.

Acknowledgments

First and foremost, I would like to express my profoundest gratitude to my supervisors, Prof. Philippas Tsigas and Associate Prof. Tomas Olovsson, for their constant guidance and support. They have always inspired me by showing excitement for any result I have presented during our meetings and cheering me up anytime I was disappointed. I am also very much in their intellectual debt.

I extend my sincere gratitude to Associate Prof. Dimitrios Kokkinakis for the excellent collaboration we had. I also thank Prof. Per-Larsson Endefors for his invaluable suggestions during my PhD follow up meetings.

I am also grateful to my colleagues in the Networks and Systems division who have contributed immensely to a friendly and productive working environment. I thank Magnus for being supportive, friendly, and fun and for all the advice he has given me. I also thank Marina and Ali for always being helpful and supportive. I would also like to give my appreciation to all the current and former members of the division. Many thanks to Andreas, Bapi, Daniel, Elad, Erland, Georgios, Iosif, Laleh, Nhan, Olaf, Oscar, Pierre, Thomas, Valentin, Vilhelm, Vincenzo, Wolfgang, Yiannis, Zhang, and all the other new members of the division. I am also thankful to all my colleagues in the department for an excellent working environment. I would especially like to express my gratitude to Peter, Eva, Tiina, and Marianne. I also thank my friends Negin, Fatemeh, and Behrooz for the good times we spent in the department.

Finally, my deepest appreciation goes to my family and friends. I am especially grateful to my parents for their unwavering love, selfless support, and encouragement over the years. I would also like to thank my wonderful husband, Mohammad Reza, who has supported me at each step of the way with his love and patience. You are the best and I am really grateful to everything you have done for me and I am proud of everything we have achieved together.

Farnaz Moradi
Göteborg, 2014

Contents

List of Figures

Part I

INTRODUCTION

1 INTRODUCTION

Advances in technology and computation have provided the possibility of collecting and mining a massive amount of real-world data. Mining such "big data" allows us to understand the structure and the function of real systems and to find unknown and interesting patterns.

Many types of real-world datasets can be modeled with *networks*. A network provides a powerful mathematical tool to represent the relations in the data. Networks generated from real-world data are often divided into four categories, social, information, technological, and biological networks [1]. A *social network* is a network connecting the people who contact or interact with each other. Social networks are not limited to "online social networks" such as Facebook, Twitter, or LinkedIn. Other examples of social networks are the network of people collaboration, co-authorships, and co-appearance, as well as networks of communication between people such as telephone calls and emails. An *information network* is a network of entities containing information such as World Wide Web, network of citations, and word co-occurrence networks. A *technical network* refers to a man-made network such as the Internet, the electric power grid, networks of roads, railways, and airline routes. A *biological network* represents a biological system such as a network of metabolic pathways, protein-protein interactions, the food web, and the network of blood vessels.

In this thesis we consider networks from two categories, i.e., social networks and information networks. The focus of the thesis is on the structural properties of these networks and the algorithms which exist for study of these properties, particularly their community structure.

This thesis is organized into two parts. The first part is an introduction to the thesis and the second part consists of a collection of papers. The remainder of the introduction is organized as follows. In Section 1.1 we briefly summarize the structural properties of social and information networks. In Section 1.2 we focus on the community structure of networks and existing algorithms for identifying network communities and investigate a number of challenges in community detection, namely quality evaluation, scalability, and seed selection. In Section 1.3 we look into a number of applications of mining real network data for identifying

interesting patterns and anomalies. In particular we look into identifying sources of unsolicited email traffic based on the communication patterns observed on an Internet backbone link. We also study the application of intrusion detection using network flow data, scalable identification of communities in social networks, and analysis of large query log files by identifying communities of related words from a word co-occurrence network. In Section 1.4 we present the real datasets which we have used in this thesis for generating different networks and analyzing their structural properties. More specifically we describe the collection process of email and flow data from an Internet backbone link, as well as the data which was obtained from different social networks and the query logs of a health care portal. In Section 1.5 our approaches towards analysis of network data and a brief description of the appended papers are presented. Section 1.6 summarizes our contributions in the thesis and, finally, Section 1.7 concludes the thesis and present possible future research directions.

1.1 Structural Properties of Networks

A great deal of work has been devoted to study the structure and dynamics of networks generated from real-world data. These networks are not random networks and the nodes in these networks are organized into specific structures. A wide variety of network mining methods and algorithms exists which can be used to uncover the structure of such networks.

Traditionally, network data was modeled as random graphs [2]. However, empirical studies on different types of real network data have revealed interesting properties such as the "small-world effect" [3], also known as "six degrees of separation" [4], and the scale-free behavior of networks [5, 6]. These properties show that social and information networks are fundamentally different from other types of networks such as random networks [1]. A review of the structural properties of these networks can be found in [7].

Many real networks have been modeled as *small-world* networks. A *small-world* network has a small *effective diameter* and the distance between any pair of nodes in the network is relatively short. The distance between two nodes is measured as the number of edges in the shortest path connecting them. In addition to small effective diameters or short average path lengths, small-world networks tend to be highly clustered which can be quantified using the average *clustering coefficient* of the networks [3].

Another robust measure of the structure of networks is their *degree distribution* which characterizes the spread in the node degrees. It has been shown that for social and information networks the degree distribution has a power law tail. This means that in these networks most of the nodes have a very low degree while a few of the nodes have very high degrees. Such networks are also known as *scale-free* networks [5, 6].

Numerous attempts to model the structure of social networks have also taken other structural properties into account: the distribution of the size of the connected components of the network, the presence of a giant connected component (GCC), and the community structure of the networks. The studies of the changes of structural properties of networks over time have also revealed interesting properties of network evolution. As the networks grow over time, they become more dense (*densification power law*) and the average distance between their nodes shrinks (*shrinking diameter*) [9]. There are many other patterns which have been observed in real world networks. A summary of different patterns, particularly the patterns observed in weighted networks can be found in [8].

1.2 Community Detection

An excessively studied structural property of real-world networks is their community structure. The community structure captures the tendency of nodes in the network to group together with other similar nodes into communities. This property has been observed in many real-world networks. Despite excessive studies of the community structure of networks, there is no consensus on a single quantitative definition for the concept of *community* and different studies have used different definitions. A community, also known as a *cluster*, is usually thought of as a group of nodes that have many connections to each other and few connections to the rest of the network. Identifying communities in a network can provide valuable information about the structural properties of the network, the interactions among nodes in the communities, and the role of the nodes in each community.

1.2.1 Algorithms

A wide variety of *community detection* algorithms, also known as *clustering* algorithms, have been proposed to identify the communities in a network. Since different community detection algorithms use different definitions of a community, they yield different communities. Figure 1.1 shows an example of the communities identified by two fundamentally different community detection algorithms on a real network (Zachary's network of karate club members [10]).

Many traditional community detection methods are borrowed or inspired from graph clustering algorithms. *Partitioning* the nodes in a network into a predetermined number of disjoint communities is one of the traditional methods for identifying communities. However, since the community structure of real-world networks are not usually known, making assumptions about the number of communities or the size of the communities are not realistic. Moreover, many real-world networks have a hierarchical structure where meaningful communities at different scales can exist and such community structures cannot be captured by partitioning algorithms. Therefore, another group of community detection algorithms have been introduced which can identify hierarchical communities. *Hierarchical* clus-

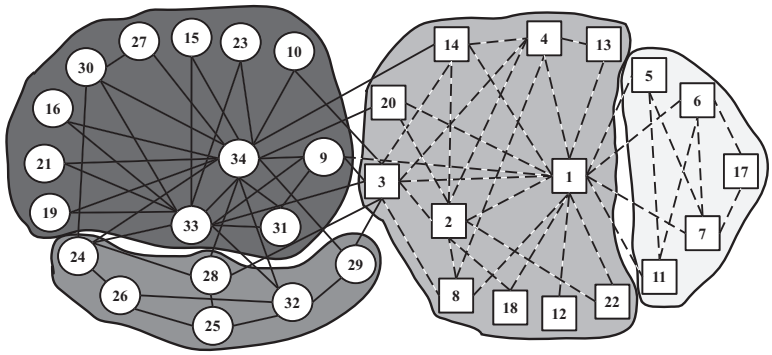


Figure 1.1: *The square and round nodes show the two groups of the members in the Zachary karate club network. The four grey communities are found by applying a node-based modularity optimization algorithm [11]. The solid and dashed edges show the two communities identified by a link-based community detection algorithm [12].*

tering techniques can be divided into *agglomerative* and *divisive* methods [13]. Agglomerative algorithms use a bottom-up approach where clusters are iteratively merged. Divisive algorithms use a top-down approach where the clusters are iteratively split. Overall, using hierarchical algorithms allow us to choose the suitable level of hierarchy and study the communities at that level of hierarchy.

In many real-world networks, nodes can naturally belong to multiple communities, therefore the communities can overlap. In social networks, an individual can belong to a community of family members, to a community of friends, and to a community of colleagues. In an information network, a web page can cover topics that are associated with different communities. Traditional community detection algorithms fail to uncover the community overlaps. Not being able to identify community overlaps in networks with naturally overlapping communities means missing valuable information about the structure of the network [14]. Therefore, *overlapping community detection algorithms* have gained a lot of attention. Overlapping communities can be identified using different approaches. One of these approaches is based on partitioning the edges of a network into communities rather than partitioning the nodes [12, 15]. A thorough review and comparison of different types of overlapping community detection algorithms can be found in [16].

The majority of existing community detection algorithms implicitly assume that the entire structure of the network is known and is available. We refer to these types of algorithms as *global algorithms*, since they require a global knowledge of the whole network in order to uncover all the communities in that network. Since such knowledge might not be available for large networks, *local algorithms* are gaining more popularity [23, 27–29]. Local algorithms typically start from a number of given *seed* nodes and expand them into possibly overlapping communities by examining only a small part of the network. Since it is possible to find local com-

Table 1.1: *Community Detection algorithms.*

	Algorithm	Type	Description	Complexity
Non-Overlapping	Blondel [11]	G,H	*Fast modularity maximization (Louvain)* is a greedy approach to modularity maximization and unfolds a hierarchical community structure.	$O(m)$
	Infomap [17], InfoH [18]	G,H	*Maps of random walks* finds communities based on the compression of the description length of the average path of a random walker over the network. *Multilevel compression of random walks* is the hierarchical version of infomap which minimizes a hierarchical map equation to find the shortest multilevel description length.	$O(m)$
	RN [19]	G,H	*Potts model community detection* minimizes the Hamiltonian of a local objective function (the absolute Potts model).	$O(m^{1.3})$
	MCL [20]	G,NH	*Markov Clustering* is based on the probability of random walks remaining for a long time in a dense community before moving to another community.	$O(nK^2)$
Overlapping	LC [15]	G,H	*Link Community detection* uses the similarity of the edges to identify hierarchical communities of edges rather than communities of nodes.	$O(nK^2)$
	LG [12]	G,H	*Line Graph and graph partitioning* runs a non-overlapping node-based algorithm on a line graph induced from the original graph to identify overlapping link-based communities.	$O(nm^2)$
	SLPA [21]	G,H	*Speaker listener Label Propagation* is an extension to the label propagation algorithm where nodes adopt multiple labels based on the majority labels in their neighborhood.	$O(tm)$
	OSLOM [22]	L,H	*Order Statistics Local Optimization Method* identifies significant communities with respect to a Null model similar to modularity.	$O(n^2)$
	DEMON [23]	L,NH	*Democratic Estimate of the Modular Organization of a Network* is a local algorithm which uses the label propagation algorithm to find communities in the egonet of each node and then merges them into larger communities.	$O(nK^{3-\alpha})$
	PPR [24]	L,NH	*Personalized PageRank-based*, is a local algorithm which uses the *PageRank-Nibble* algorithm [25] to approximate a personalized PageRank vector from a given *seed* node and then uses the method in [26] to create the communities based on a scoring function.	$O(\sum_{C \in \mathcal{C}} vol(C))$

In the "Type" column, L and G denote local and global, and H and NH denote hierarchical and non-hierarchical, respectively. The LG algorithm can find hierarchical communities if the node-based algorithm is hierarchical.
In the "Complexity" column, n denotes the number of nodes, m denotes the number of edges, K is the maximum node degree, t is the number of algorithm iterations selected, α is the power-law exponent, $vol(C)$ is the sum of the degree of all the nodes in a community C, and $\mathcal{C}$ is the set of all the identified communities.

munities from each seed independently, they are very suitable for being parallelized and therefore can scale well. The local communities identified from each seed can be aggregated in order to uncover the global community structure of the network. However, if the local community detection algorithm is naively started from each node in a network, it can lead to many redundant communities and therefore is computationally expensive. Therefore, it is important to identify a number of good seeds which are well distributed over the network by using a *seeding algorithm* before running the local community detection. On the other hand, if the seeding algorithm does not select enough seeds, the communities might only cover a subset of the nodes in a network and therefore, the problem of selecting a reasonable number of seeds which are well-distributed over the network is challenging. These challenges are further investigated in Section 1.2.4.

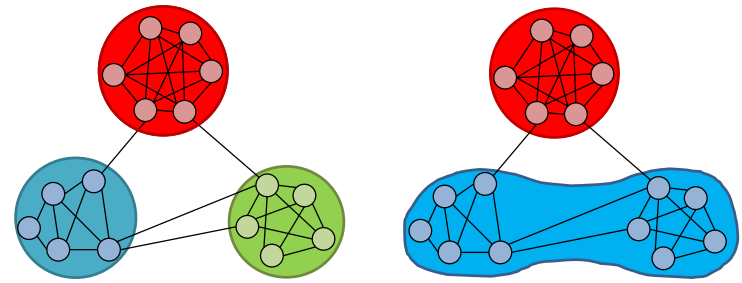
In addition to different types of community detection algorithms, recently, a number of studies have focused on proposing methods for improving the quality of the existing community detection algorithms. Ciglan et al. [30] introduced a method for adding edge weights to unweighted networks as a pre-processing step to improve the quality of the identified communities with respect to ground truth data. Soundarajan et al. [28] introduced a template for using existing community detection algorithms for identifying more realistic communities. Another approach for improving community detection is to use ensemble clustering, which is inspired by ensemble learning, where multiple community detection algorithms run as an ensemble and the identified communities are combined to improve the community qualities. Staudt et al. [31] showed that ensemble clustering can be used to achieve the best trade-off between quality of the communities and the speed of community detection.

Thorough reviews of different types of community detection algorithms can be found in [13, 16, 32]. Table 1.1 summarizes the algorithms which are used throughout this thesis.

1.2.2 Quality Evaluation

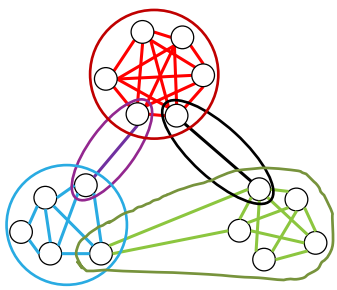
Given the diverse nature of real-world networks and the high diversity of community detection algorithms, it is necessary to perform experimental evaluation of the algorithms to find the most suitable method for each type of network. However, due to the ambiguity in the definition of a community, extracting communities and evaluating their quality is proven to be very difficult.

Figure 1.2 shows the communities identified by different community detection algorithms (see Table 1.1) in a toy network. It can be seen that different types of algorithms identify different communities in the network since they use different definitions for communities and take different approaches for identifying these communities. In order to find out which algorithm yields the best set of communities, it is necessary to use a quantitative measure to evaluate the quality of the communities identified by each algorithm.

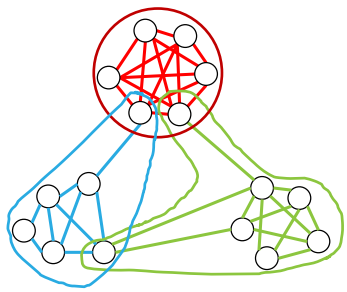


(a) Blondel, Infomap, RN, MCL, PPR

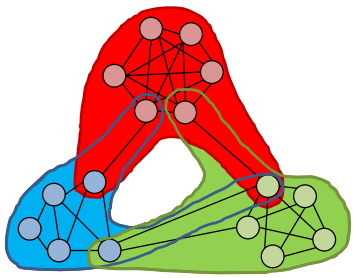
(b) OSLOM



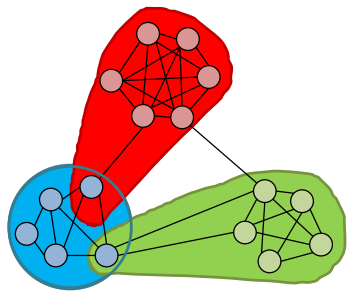
(c) LC



(d) LG



(e) DEMON



(f) SLPA

Figure 1.2: *A comparison of the communities yield by different community detection algorithms on a toy example network.*

The most widely used structural *quality function* is *modularity* [33] which is also widely used as an *objective function* or *scoring function* to be optimized by community detection algorithms. In addition to modularity, many other quality functions have been used and proposed in the literature. However, it has been shown that there is no single perfect quality function for comparison of the quality of the communities identified by different algorithms [34]. Moreover, many of the existing quality functions are designed for evaluating disjoint communities and extending them for evaluation of overlapping communities is not straightforward [16].

One of the methods which is widely used for evaluating and comparing the identified communities by different algorithms is to use synthetic networks from different *benchmarks*. In the GN benchmark [35], communities of the same size are embedded into a network for a given expected degree and a given ratio of internal to external connections between the communities. Other benchmarks have been proposed to improve and complement GN for example for overlapping communities. One such widely used benchmark is the LFR benchmark [36] which introduces heterogeneity into degree and community size distributions of a network.

The main reason for using benchmark graphs for evaluating community detection algorithms, is the lack of *ground truth* information about the communities in real-world networks. Recently, more studies have used ground truth data. Ground truth data is usually obtained from meta data or explicit group memberships of the nodes. Ahn et al. [15] used meta data, e.g., tags assigned by users to annotate the items in a co-purchase network, to define a number of quality functions based on the purity of the attributes of nodes in communities and to assess how well the identified communities reflect the meta data. Abrahao et al. [37] identified ground truth communities from annotations, e.g., product categories and groups of protein functions, and compared the structural properties of the communities detected by different algorithms with ground truth communities. Yang and Leskovec [24] have studied a large number of social, collaboration, and information networks to define ground truth communities based on the explicit declaration of group membership by the nodes. Their comparison of the ground truth communities with different definitions of communities have shown that *conductance* is the best scoring function for networks with well-separated and non-overlapping communities, while the *triad-participation ratio* is the best scoring function for networks with densely overlapping communities.

In this thesis, in addition to the above methods for evaluating community quality, we also propose to evaluate the *logical quality* of the communities identified by different algorithms. The logical quality is defined based on the type of the edges inside communities and how homogeneous these edges are. In other words, the communities in which all of the edges are homogeneous, i.e., are of the same type, are considered to have perfect logical quality (see Section 1.5.2).

1.2.3 Scalability

Identifying high quality communities from large-scale real-world networks is typically computationally expensive and does not scale well. One approach for improving the scalability of community detection is to use parallelism. Parallelism can significantly speed up the community detection and is also necessary for coping with the massive volume of real-world datasets.

Recently, a number of studies have proposed parallel community detection algorithms. Yang and Leskovec [42] proposed BigClam which is a model-based parallel algorithm for community detection. Prat-perez [43] proposed SCD which is a parallel scalable algorithm which identifies disjoint communities.

In addition to designing new parallel algorithms, there has been a number of attempts to parallelize conventional community detection algorithms in order to improve their scalability. Staudt et al. [31] provided the parallel implementation of the Louvain algorithm by Blondel et al. [11] and the label propagation algorithm [38]. Cheong et al. [39] proposed a hierarchical parallel algorithm based on the Louvain algorithm implemented on single- and multi-GPU (Graphics Processing Unit). Soman et al. [40] proposed a community detection algorithm based on label propagation optimized for GPU architectures. Kuzmin et al. [41] proposed a parallel version of the SLPA [21] algorithm for shared and distributed memory machines.

Another fast and scalable approach to community detection is to use local community detection algorithms. In local algorithms, the computations can be done in parallel starting from seed nodes and expanding them into communities by only investigating the neighborhood of the seed nodes in the network. A naive approach to local community detection is to expand every node in the network into a community. However, this approach is computationally expensive and will generate many duplicate communities. Therefore, the challenge is to select an optimal number of seeds to be expanded into communities which can cover the majority of the nodes in a network.

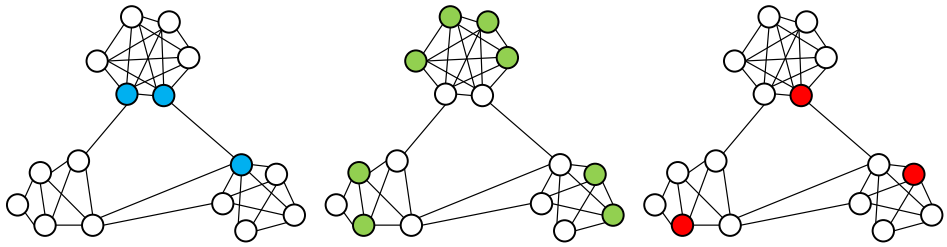
1.2.4 Seed Selection

One of the most successful community detection methods is local seed expansion which is, as mentioned earlier, also very scalable since it is parallelizable by nature. However, the problem of selecting *good seeds* to be expanded into high quality overlapping communities is far from trivial and is not widely studied.

A good seed is usually assumed to have many neighbors inside the target community. Andersen et al. [25] theoretically showed that a seed set that is "nearly contained" in a target community is a good seed set for that community. They also showed that a randomly selected seed set from a target community can also be a good choice for identifying that community. However, Whang et al. [29] showed that careful selection of seeds leads to better results compared to a simple random selection.

One approach for selecting good seeds in a network is to use non-structural knowledge of the network if such information exists. As an example, Gargi et al. [14] have considered non-structural properties of the Youtube video network and have selected the nodes which correspond to videos with the highest view count as the seeds. Unfortunately, such non-structural information might not be available for many types of networks particularly when no global knowledge about the network exists.

In other studies, the structural properties of the networks have been used for seed selection. Shen et al. [44] proposed to use maximal cliques as seeds since they form the core of the communities. However, this approach is computationally expensive. It was shown by Gleich et al. [45] that the *egonets* with low conductance



(a) SH (k=3), MD (b) EC (c) CN+coloring (our algorithm)

Figure 1.3: *A comparison of the seeds yield by different seed selection algorithms on a toy example network.*

(EC) are good seeds for finding the best communities of a network with respect to conductance. However, Whang et al. [29] showed that the communities expanded from these egonets do not achieve a good coverage of the network. Chen et al. [46] proposed an algorithm for selecting the nodes with local maximal degree (MD) as seeds and suggested to repeatedly remove the identified communities expanded from the selected seeds from the network and find new seeds in the remaining parts of the network to improve the coverage.

Whang et al. [29] have proposed two seeding algorithms which can achieve good coverage: Graclus centers and Spread hub. In the *Graclus centers*, first a partitioning algorithm is used in order to find k partitions, where k is pre-determined, and then the nodes in the center of these partitions are selected as seeds. In the *spread hub* algorithm (SH), first the nodes in the network are sorted based on their degree, then the nodes with the highest degree are selected as seeds until at least k nodes are selected. These seeding methods are both shown to perform well in large real-world networks. However, these methods require that the number of seeds to be selected is known in advance. Unfortunately, making assumptions about the number of communities in a network is not realistic since the community structure of real-world networks is normally unknown to us.

Figure 1.3 shows the seed nodes which are selected by different seeding methods. It can be seen that different algorithms pick different nodes as seeds since they take different structural properties of the nodes into account. In this thesis, we propose a new seed selection algorithm which does not require global information about the network nor the number of seeds to be picked, and still is able to select a reasonably small number of good seeds which are well distributed over the network (see Section 1.5.4).

1.2.5 Other Challenges

Despite the excessive number of community detection algorithms proposed in the literature, identifying communities in real-world networks is still a challenge. The

challenges are not limited to quality evaluation of the identified communities and the scalability of the algorithms. Some other challenges, which are not covered in the thesis, but are very important to be studied are as follows.

- Identifying communities in dynamic networks, where new nodes can join, existing nodes can leave the network and new edges can be formed and existing edges can break.
- Studying the stability of communities identified by different algorithms, particularly in evolving networks.
- Combining structural and non-structural information, where such knowledge exists, for identifying more realistic communities.
- Interpreting what the identified communities show about the function of the system and how the output of a community detection algorithm can be used for different applications.

1.3 Applications

Mining large-scale real-world network data has many different applications such as understanding the function of a system, modeling and predicting its behavior, and identifying outliers and anomalies. In this section we present three network data analysis applications which are the focus of this thesis.

1.3.1 Unsolicited Email Detection

Email is one of the most common services on the Internet with everyday business and personal communications depending on it. Unfortunately, the vast amount of unsolicited email (*spam*) consumes network and mail server resources, imposes security threats, and costs businesses significant amounts of money. Spam can also be exploited for phishing and scam and it can carry Trojans, worms, or viruses, making email unreliable.

It is known that a large fraction of spam originates from *botnets* [47, 48]. A botnet is a collection of compromised hosts (*bots*) where each bot contributes to conducting malicious activities or attacks such as distributed denial of service (DDoS), scanning, click frauds, and sending spam. Therefore, identifying the source of spam can lead to the detection of the source of other malicious activities on the Internet.

Numerous attempts to fight spam have led to implementation of anti-spam tools that are quite successful in hiding the spam from users' mailboxes. Most of the conventional approaches inspect email contents at the receiving mail servers, and are very resource-intensive. Although such *content-based filters* are effective in learning what the content of spam looks like, the spammers are very agile in obfuscating email contents and encapsulating their messages in other formats such as images to bypass these filters.

As a complement to content-based filters, *pre-filtering* strategies are widely used to stop spam before the email content is received and examined by the mail servers. A commonly used pre-filtering method is *IP blacklisting*. The receiving mail servers can consult IP blacklists to decide whether to accept or reject an incoming email. However, IP addresses are not persistent, they can be obtained from dynamic pools of addresses and they can be stolen [47, 49]. In addition, bots usually send spam at a low rate to each individual domain and do not reuse IP addresses that have become blacklisted.

In addition to the above mentioned anti-spam strategies, numerous other spam detection and prevention techniques have been introduced. Approaches such as enforcing laws and regulations, requesting proof-of-work (e.g., processing time) [50], mail quota enforcement [51], port blocking, and user monitoring are proposed to stop spam at the sender side. Greylisting [52], reputation-based approaches, sender authentication, and domain verification are approaches that can be used on the receiver side before accepting email contents. Replacing SMTP with a new protocol or deploying overlay authentication protocols, are some other ideas proposed to stop spam during transit.

Recently, approaches that focus on the network-level behavior of spam have gained attention. These approaches are concerned about email sending behavior of the spammers, which is expected to be more difficult for them to change than the content of the email [53–55]. In order to improve and come up with more such methods, there is a need to understand the network-level characteristics of spam and how it differs from legitimate email (*ham*) traffic.

It is known that spam is sent automatically, therefore it is expected that it does not exhibit the *social* properties of human-generated communications [56–59]. The social properties of email communications can be studied by analyzing the structure of *email networks* generated from email traffic. An email network is an implicit social network in which each node represents an email address and each edge represents an email. It has been shown that email networks have the same structural properties that other social and interaction networks have [60–62]. Our intuition is that the structural properties of email networks containing unsolicited email are not similar to the structure of email networks containing only legitimate email. Therefore, analysis of email networks generated from a mixture of email communications can be used for identifying the distinguishing properties of ham and spam which can potentially be used for detecting the botnets based on their anti-social behavior rather than on the content of what they send.

1.3.2 Network Intrusion Detection

Networked systems are continuously under attack causing considerable damages, therefore, network intrusion detection systems are widely deployed. Network intrusions can be identified using two different approaches, i.e., misuse detection and anomaly detection. Techniques for misuse detection rely on the signatures of attacks, and search for patterns of well-known attacks to identify intrusions, there-

fore, they lack the ability to detect new intrusions or zero-day attacks. Anomaly detection techniques, on the other hand, do not require prior knowledge of an attack signature. However, they might have a high false positive rate.

In this thesis, we focus on anomaly detection-based intrusion detection systems. Anomaly detection has been extensively studied in the context of different application domains and a variety of techniques have been proposed. An overview of anomaly detection methods can be found in [63].

Anomalies are patterns in network traffic that do not conform to normal behavior. Any change in the network usage behavior or malicious activities such as DoS attacks, port scanning, unsolicited traffic, and worm outbreaks, can be seen as anomalies in the traffic.

The main challenge in using anomaly detection for identifying misbehaving hosts is to define normal behavior and draw boundaries between normal and abnormal communication patterns. One approach to defining normality is to look into the social behavior of normal nodes. Since many types of intrusions are automatically generated, it is expected that they do not conform to the expected normal social behavior. Therefore, a number of features that are representative of (anti)social communication patterns can be extracted for identification of misbehaving nodes.

Recently, it was shown that network intrusions can successfully be detected by examining the network communications that do not respect the community boundaries [64]. In such an approach, normality is defined with respect to social behavior of nodes concerning the communities to which they belong and intrusion is defined as "*entering* communities to which one does not belong". In this thesis we propose an alternative definition for anomaly/intrusion and study how the network structure and the community structure of graphs generated from network traffic can be used for network misbehavior detection (see Section 1.5.3).

1.3.3 Query Analysis

Logs of search engines contain a wealth of information from the queries submitted by users. Query logs have been widely studied and analyzed in order to improve the service provided to the users and to better understand their behavior and needs. Analysis of web query logs can provide useful information regarding the use of a site considering when and how users seek information for topics covered by the site [65]. Extracting information from query logs can also be useful for different types of users such as terminologists, infodemiologists, and web analysts, as well as specialists in Natural Language Processing (NLP) technologies such as information retrieval and text mining.

Medical and health information seeking on the Internet is quite common. Mining query logs of medical search can be beneficial to public officials in health and safety organizations, epidemiologists, and medical data analysts. Information extracted from large-scale logs can be used both for a general understanding of public health awareness and the information seeking patterns of users, and for optimizing

search indexing, recommendations, query completion and presentation of results for improved public health information.

In order to study query logs, several graph-based relations among queries can be used [66]. A co-occurrence network for the words which co-occurred in different queries is an information network which we use to capture the relations between the words. We further study the structural and temporal properties of the co-occurrence network and show that it is similar to other information and social networks. We also look into the community structure of the network and how the identified communities can potentially be used for improving our understanding of the language used by users of the health care portal and improving their search experience (Section 1.5.5).

1.4 Data Collection

Getting access to and performing analysis of large-scale real-world datasets is crucial for many different applications. Collection and processing of real data is far from trivial. The challenges involved are both of general and technical nature. Getting access to the data, privacy and ethical concerns, pre-processing and analysis of the dataset are just a number of challenges that need to be addressed before the data can be used for an application. The main challenge, however, is handling the massive amount of data. The data collection process has to keep up with the speed in which the data is being produced or received. It is usually inevitable to sample the data, to process summaries of the data or to only focus on analyzing snapshots of data obtained during limited time windows. In some cases such as Internet traffic collection, special measurement equipments which can cope with full link-speed or allow high sampling frequencies are required. After the collection, the data also needs to be parsed or pre-processed before it is possible to extract relevant information for example to create a network from the relations observed in the datasets. In many cases, obtaining ground-truth data for evaluating the results of the data analysis can also be impossible or non-trivial. In this thesis we have collected and obtained different types of real data including data captured from a high speed Internet backbone link, data from social and information networks, and query log files from a health care portal.

1.4.1 Email Dataset

One of the datasets which is collected by us is an email dataset which is used for understanding the characteristics of legitimate and unsolicited email. The study of the characteristics of email and spam can be conducted using different types of email data. A number of studies have used SMTP log files from mail servers [49, 57, 59, 67–69]. Although such datasets are limited to communications to/from a single domain, they contain detailed information about each email and the statistical summaries of accepted and rejected email communications, which

allows comparison of the behavior of spam, ham, and the rejected traffic. The spam captured in honeypots or relay sinkholes have also been used to study the characteristics of spam [53, 70]. The honeypots only attract spammers, therefore they do not allow the comparison of different characteristics and communication patterns of spam and ham. Flow-level data collected on access routers have also been used to study the properties of spam and rejected traffic [71]. These flows only contain packet headers, and although they are not limited to a single domain, they do not carry enough information to allow distinguishing spam from ham to study their distinct characteristics. Another type of data that has been used to understand the sending behavior of spam was collected from inside spam campaigns [48, 72, 73]. The data collected at these campaigns has the view point of spammers and makes it possible to closely investigate how spam is sent.

In our studies, we have used yet another type of email data. Our dataset enables us to study the behavior of legitimate and unsolicited traffic from the perspective of a network device which monitors the traffic traversing a backbone link. The collected email traffic is not limited to a single organization or domain and allows us to classify the observed email into ham, spam, and rejected communications to compare their characteristics.

Collection of large datasets from backbone Internet traffic can face several challenges [74]. Not only is mere physical access to optical Internet backbone links needed, but also rather expensive equipment in order to deal with the large data volumes arriving at high speeds. Adding to the complexity, the collected data traces must be desensitized since they may contain privacy-sensitive data. Packets also need to be reassembled into application level "conversations" so that, finally and maybe the most challenging part, methods and algorithms suitable for analysis of massive data volumes can be run [75].

Our datasets were generated passively capturing traffic on a 10 Gbps backbone link of SUNET (the Swedish University Network) [76]. The collection location is shown in Figure 1.4. Each dataset was collected over 14 consecutive days with roughly a year time span between them.

The process of collecting data and generating the first dataset is described in more detail in the following. Table 1.2 summarizes the collected data during 14 consecutive days in March 2010. The second dataset was also collected similarly during 14 consecutive days in spring 2011.

We used a hardware filter to only capture traffic to and from *port 25* which resulted in more than 183 GB of SMTP data. The captured packets belonging to a single flow were then aggregated to allow the analysis of complete SMTP sessions.

The collected data contained both *SMTP requests* and *SMTP replies*. As each SMTP request flow corresponds to an SMTP session, it can carry one or more emails, thus we had to extract each email from the flows by examining the SMTP commands. The resulting extracted email transaction contained the SMTP commands including the email addresses of the sender and the receiver(s), email headers, and the email content.

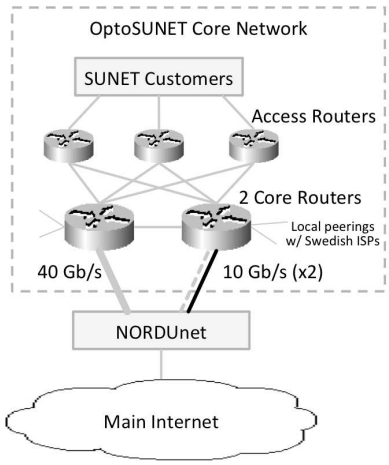


Figure 1.4: *OptoSUNET core topology. All SUNET customers are via access routers connected to two core routers. The SUNET core routers have local peering with Swedish ISPs, and are connected to the international commodity Internet via NORDUnet. SUNET is connected to NORDUnet via three links: a 40 Gbps link and two 10 Gbps links. Our measurement equipment collects data on the first of the two 10 Gbps links (black) between SUNET and NORDUnet.*

After the collection phase, first the dataset was pruned of all unusable email traces. For example, flows with no payload are mainly scanning attempts and should not be considered in the classification. Also, SMTP flows missing the proper commands were excluded from the dataset as they most likely belong to other applications using port 25. Encrypted email communications cannot be analyzed, and were also eliminated.[1] Any email with an empty sender address is a notification message, such as a non-delivery message [77]; it does not represent a real email transmission and was also excluded. Finally, any email transaction that was missing either the proper starting/ending or any intermediate packet was considered as incomplete. Possible reasons for having incomplete flows include transmission errors and measurement hardware limitations caused by a framing synchronization problem.

The remaining email transactions were then classified as *accepted*, i.e. those emails that are delivered by the mail servers, or *rejected*. An email transaction can fail at any time before the transmission of the email data (header and content) due to rejection by the receiving mail server. Therefore, *rejected* emails are those that do not finish the SMTP command exchange phase and consequently do not send any email data. The rejections are mostly because of spam pre-filtering strategies

[1] Around 3.8% of the flows carried encrypted SMTP sessions.

Table 1.2: *Email dataset statistics (2010).*

	Incoming ($/10^6$)	Outgoing ($/10^6$)
Packets	626.9	170.1
Flows	34.9	11.9
Distinct source IPs	2.30	0.01
Distinct destination IPs	0.57	1.94
SMTP Replies	2.84	9.14
Email:	19.3	0.73
Ham email	1.32	0.21
Spam email	1.66	0.20
Rejected email	16.3	0.31

deployed by mail servers including blacklisting, greylisting, DNS lookups, and user database checks.

Finally, we discriminated between *spam* and *ham* in our dataset. As we have captured the complete SMTP flows, including IP addresses, SMTP commands, and email contents, we can establish a ground truth for further analysis of *only* the spam traffic properties and a comparison with the corresponding legitimate email traffic. We deployed the widely-used spam detection tool called SpamAssassin[2] to mark emails as spam and ham. SpamAssassin uses a variety of techniques for its classification, such as header and content analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases.[3]

The final pre-processing step of the dataset was to desensitize any user data. Immediately after the classification of emails into ham and spam, we discard the content of the emails and anonymized the email and IP addresses in the headers [75]. Once the sensitive data was discarded, the resulting anonymized dataset had a size of 37 GB.

The second dataset from 2011 was collected and pre-processed similarly to the first dataset. The infrastructure and the data collection equipment was updated during the one year time span between the collections. Although, the changes have caused differences in the collected data, these differences are in our favor since they allow us to compare our observations over time and verify that our findings are not limited to a single vantage point.

[2] `http://spamassassin.apache.org`

[3] The well-trained SpamAssassin applied to our dataset was in use for a long time at our university, incurring an approximate false positive rate of less than 0.1%, and an detection rate of 91.3% after around 94% of the spam being rejected by blacklists.

Table 1.3: *Unique hosts during the data collection 2010-04-01.*

	Inside SUNET		Outside SUNET	
Incoming Link	Destination IPs	970,149	Source IPs	24,587,096
Outgoing Link	Source IPs	23,600	Destination IPs	18,780,894

1.4.2 Flow Dataset

In order to study other types of misbehavior in network traffic such as network intrusions, we have used network flow data collected from the backbone link of SUNET. The network flow data was collected from the same location as the email dataset (see Figure 1.4).

For a period of more than six months, a 24 hour snapshot of all flows was regularly collected once a week. The dataset contains a total of 12 billion flows in both directions. Table 1.3, summarizes all unique IP addresses found during a single collection day to give an idea of the scale of the traffic passing by the measuring point.

This dataset also contains metadata, including, for example, hosts known to aggressively spread malware at the time of the collected snapshots. The source addresses of these malicious sources in the dataset were defined by using the lists reported by DShield and SRI Malware Threat Center during the data collection period [78, 79]. By using the flow data together with this information, we can then make more targeted types of analysis of hosts, despite their addresses being anonymized.

We have used flow data from seven days in the dataset in order to study a community-based network intrusion detection method (Section 1.5.3). More details about the collection of the dataset and other analysis performed on the data can be found in [80].

1.4.3 Social and Information Network Datasets

In addition to data from real network traffic, we have used data from other types of social and information networks. We have used publicly available datasets provided by the Stanford Large Network Collection [81] including a product network from Amazon, a collaboration network from DBLP computer science bibliography, and the social networks of users in Youtube and Livejournal. These datasets also include the information about the ground truth communities.

In the *Amazon* network, nodes are products in the Amazon website and two products have an edge if they were co-purchased frequently. The ground truth is based on the product categories defined by Amazon. In the *DBLP* network, nodes are authors and two authors are connected with an edge if they have co-authored at least one paper, and the ground truth is obtained based on the publication venues. In the *Youtube* and *LiveJournal* networks the nodes are the users of the

video sharing and online blogging websites, respectively, and the edges correspond to friendships and the ground truth is based on user-defined groups.

In addition to above datasets, we have collected a dataset from the *SoundCloud* sound sharing site (`http://soundcloud.com/`). In SoundCloud, similar to Twitter, users can follow each other, and popular artists tend to attract a large number of followers. For the collection of Soundcloud data, we alternated between random sampling and breadth-first-search, so that we could capture local neighborhood information while covering different parts of the network [82]. After data collection, we generated a network of "follow" relations, where the nodes are the users, and an edge (u, v) exists if the user u follows the user v.

The data collection from SoundCloud is an ongoing process and by the time this thesis is being written, we have collected data from more than 39 million users with more than 642 million follows and around 76 thousand groups. We are going to publish a more complete version of the datasets after finishing the collection process. By the time we started to use the SoundCloud dataset, we had around 5 million users in the dataset. Even though our work is focused on a small subset of the whole user base, this network has been the largest social network which we used in our studies. In this thesis, we have used the datasets presented in this section for evaluating our proposed local seed selection algorithm. Our algorithm selects seeds by merely investigating the direct neighborhood of each node in the network and therefore does not require the global structure of the network to be accessible, so our analysis is not affected from the lack of global data.

1.4.4 Medical Query Logs

The last dataset which we used was obtained from the query logs of a Swedish health care portal. We obtained 67 million queries for the period October 2010 to the end of September 2013. The data was provided by vardguiden.se through an agreement with the company Euroling AB which provides indexing and searching functionality to vardguiden.se. 27 million of the queries are unique before any kind of normalization, and 2.2 million after case folding.

The obtained queries are then automatically annotated with semantic labels using two medically-oriented semantic resources, i.e., the Systematized Nomenclature of Medicine - Clinical Terms (SNOMED CT) and the National Repository for Medicinal Products (NPL), as well as a named entity (including the ontological categories location, organization, person, time, and measure entities) recognizer. We used these labels to identify semantic communities based on the co-occurrence of words in the queries.

Moreover, from each query which contained more than one word/term, we extracted the words and created a network of word co-occurrences. We are interested in analyzing the relations between the words and the language being used in the queries, so the single-word queries were not of interest to us. This network was used for structural analysis and identification of graph communities.

Overall, the semantic and graph analysis of query logs can be of great interest for different types of studies and can reveal important information about the usage patterns, information needs, and the language of the users of the website (Section 1.5.5).

1.5 Our Approach

As presented in the previous section we have collected and obtained large volumes of real-world data and constructed different networks from the datasets and studied their structural properties. In this section we summarize our approaches towards the different applications which we had at hand. The details of our approaches are covered in the appended papers.

1.5.1 Structural and Temporal Analysis of Email Networks

In order to understand the characteristics of unsolicited email traffic and how they differ from legitimate traffic, we have performed a *social network analysis* of real email traffic (Section 1.4.1). Our hypothesis is that social network analysis of email traffic can reveal the differences in the communication patterns of legitimate and unsolicited email traffic and can be used for identifying the sources of spam.

In order to verify our hypothesis, we have generated *email networks* from the observed email communications in which each node represents an email address and each edge represents an observed email communication between a pair of nodes. The generated email network from the larger dataset contains 10,544,647 nodes and 21,562,306 edges, and the email network from the smaller dataset contains 4,525,687 nodes and 8,709,216 edges. Based on our ground truth, we have generated a number of ham, spam, rejected, and complete email networks, and have studied and compared their structural and temporal properties. We have looked into the (in-/out-)degree distribution, average shortest path length, average clustering coefficient, distribution of the size of the connected components, the percentage of total nodes in the giant connected component, as well as how these properties change over time as the networks grow.

Our study reveals that the legitimate email traffic exhibit similar structural properties as other social and interaction networks, and therefore a ham network can be modeled as a scale-free small-world network. We also show the similarities and differences in the structural and temporal properties of email networks of ham and spam, and show that the anti-social behavior of spam and rejected traffic is not hidden in a mixture of email traffic and causes anomalies (outliers) in the structural properties of email networks. We also propose a method for identifying spamming nodes by finding the outliers in the structural properties of email networks which mainly are caused by the spammers.

1.5.2 Evaluation of Community Detection Algorithms

Despite the excessive number of studies on community detection there is no consensus on a definition for a community and different community detection algorithms have been proposed in the literature based on the different definitions. Therefore, it is not clear how to evaluate which algorithm is most suitable to be used for different types of networks. Moreover, due to the ambiguity in the definitions for community, assessing the quality of the communities identified by different algorithms can be challenging.

In this thesis, we have conducted an empirical study to compare and evaluate a variety of community detection algorithms based on a set of structural and logical quality functions on our email networks. We have evaluated the structural quality of the communities using different well-known and widely-used quality functions, namely modularity, coverage, and conductance. We have also proposed to use the *logical quality* of the communities based on how homogeneous the edges inside the communities are. A community which only contains the same type of edges is considered to have a perfect logical quality. Our aim is to find the most suitable approach that can separate ham and spam emails from the mixture of traffic into distinct communities.

Our study shows that both ham and spam networks, as well as networks containing a mixture of both, exhibit a community structure, and that different community detection algorithms can be used to unfold the communities of these networks. However, we also show that there is a trade-off in creating high structural quality and high logical quality communities. We reveal that although different community detection algorithms use different approaches to define and extract the communities of a network, algorithms that create communities with similar granularity and size distribution also achieve similar structural and logical qualities. We confirm that community detection algorithms which find coarse-grained communities achieve high structural quality. However, we reveal that they fail to find communities with high logical quality since they tend to combine smaller homogeneous communities into mixed communities in favor of better structural quality. We also show that an edge-based community detection algorithm can achieve a high logical quality since it can separate ham and spam emails into distinct communities.

1.5.3 Identifying Misbehavior Using Community Detection Algorithms

Recently, it was shown that the community structure of a flow network can be used for successful intrusion detection [64]. In a community-based anomaly detection method, normality is defined with respect to the social behavior of nodes concerning the communities to which they belong. Nodes that participate in anti-social communications and disrespect community boundaries by "*entering* communities to which they do not belong" can be identified as anomalous by a community-based anomaly detection method. Despite the fact that these methods use a notion of

community, Ding et al. [64] showed that a traditional modularity maximizing community detection algorithm is not suitable for intrusion detection in network flow data since the majority of intruders end up inside a large community and do not enter other communities.

Our intuition is that, in contrast to Ding et al. [64], community detection algorithm can be used for successful network anomaly/intrusion detection. In order to verify this, we look into communities identified by different types of community detection algorithms to extend and complement the work in [64]. Our hypothesis is that misbehaving nodes tend to *belong to multiple communities*. However, a vast variety of community detection algorithms partition network nodes into disjoint communities where each node only belongs to a single community, therefore they cannot be directly used for verifying our hypothesis. Therefore, we introduce *auxiliary communities* to enhance non-overlapping community detection algorithms. This enhancement is achieved by adding a layer of auxiliary communities over the boundary nodes of neighboring communities, allowing nodes to be members of several communities. Therefore, this enhancement enables us to show that, in contrary to [64], it is possible to use community detection algorithms for identifying anomalies in network traffic.

In addition to traditional community detection algorithms, numerous overlapping algorithms exist which allow a node to belong to several overlapping communities [16]. We also compare our proposed enhancement method for non-overlapping community detection algorithms with a number of overlapping algorithms for network anomaly detection, and show that they have comparable performance.

Finally, we propose a framework for network misbehavior detection. The framework allows us to incorporate a community detection algorithm for identifying anomalous nodes that belong to multiple communities. However, since legitimate nodes can also belong to several communities [24], we also introduce a number of application-specific filters based on different graph properties to be used for discriminating the legitimate nodes from the anti-social nodes in the community overlaps, thus reducing the induced false positives. Our experiments show that our framework is suitable for identifying intruders and the sources of scanning attacks from flow networks, and the sources of spam from email networks.

1.5.4 Local Seed Selection for Overlapping Community Detection Algorithms

Local community detection algorithms are gaining more attention than global algorithms which require the structure of the whole network to be known. In local algorithms, first local communities are identified independently of each other only based on local knowledge of the network, then they are combined to provide the global community structure of the network. Local algorithms are easy to parallelize and therefore can scale well. However, the selection of good seeds to be expanded into communities that achieve good coverage of the network is challenging. Our

aim is to design a local *seeding algorithms* which can select a reasonable number of seeds which are well-distributed over the network and therefore can lead to communities covering the majority of the nodes.

Existing seeding algorithms either require a global knowledge of the entire network to be available or they will fail to pick an adequate number of seeds which can lead to incomplete coverage of the network. Therefore, in this thesis we further study the problem of local seed selection for finding a reasonably small number of seeds. The seeds identified by such a seeding algorithm can then be expanded into high quality overlapping communities using high quality local community detection algorithms such as the Personalized PageRank-based algorithm (PPR) [24, 83].

We propose a novel seed selection algorithms for local overlapping community detection. First, we define a *similarity score* which is calculated as the sum of the similarity of a node with all of its connected neighbors by adopting the *similarity indices* from *link prediction* techniques. In link prediction, the aim is to estimate connections that are very likely to be formed between nodes in a network, therefore link prediction methods typically use a similarity index to calculate the similarity of the nodes which are not directly connected. If two nodes have a high similarity, it is predicted that an edge will be formed between them. However, in our algorithm, we use similarity indices to calculate the similarity of the nodes which already share an edge. Our intuition is that a node that has a high aggregated similarity with its neighbors is expected to belong to the same community as its neighbors. Therefore, we propose to select the node with the highest score in its neighborhood as a seed and expand it into a community. We have compared a number of different widely used similarity indices for our seeding algorithm and have also compared our seeding algorithm with a number of existing local seeding algorithms.

Although we show that by using similarity scores we can identify a small number of very good seeds, we can also show that similar to other local seeding algorithms, the expanded communities from these seeds do not achieve a high coverage of the network. Therefore, we propose to use distributed random graph coloring for enhancing our local seed selection algorithm. In order to combine similarity scores with graph coloring for seed selection, we propose a *biased graph coloring algorithm* in which the nodes with high similarity score are assigned a specific color and color conflicts between neighbors are resolved at random. This enhancement of our seeding algorithm makes sure that good seeds which have received the specific color are well distributed over the network. Our biased coloring algorithm can also be used for enhancing and improving other existing local seeding methods.

Our novel local seeding algorithms is parameter free, finds seeds that are well distributed over the network, and does not pick neighboring nodes as seeds and therefore does not lead to many duplicate communities. We empirically evaluate the execution time of local community detection when seeding is used as the first step of community detection and compare the quality and the coverage of the communities expanded from the selected seeds using large-scale real-world networks. Our experiments show that by using seeding, the execution time of community

detection is dramatically reduced and the average quality of the communities is preserved and a high coverage is achieved.

1.5.5 Graph-based Analysis of Medical Queries

Large search query logs carry a wealth of information about the behavior of the users in information seeking and the language they use. Similar to many other types of data, query log files can also be modeled as networks.

Our hypothesis is that graph-based analysis of words which have co-occurred in different queries can provide a better understanding of the relations of words and terms in different domains and in different languages. In order to verify our hypothesis, we have generated a *word co-occurrence network* from the query logs of a Swedish health care website. We study the structural and temporal properties of the generated network and show that it is similar to other existing information and social networks. We also look into the community structure of the word co-occurrence network in order to understand the relation between the words in a medical domain.

Moreover, we have introduced *semantic communities* which are communities of words which have co-occurred with a semantic label. These labels are added to the queries using medically-oriented semantic resources. We also apply a personalized PageRank-based community detection algorithm to the generated word co-occurrence network and compare the identified *graph communities* with the semantic communities. Our experiments show that while semantic communities can cover only a small percentage of all the words in the logs, the graph communities can cover the vast majority of the words. Therefore, the graph-based analysis can capture more relations among the words which have been used in the queries. Moreover, the graph and semantic analysis capture different relations between the words and identify communities which are only partially similar and therefore can be used to complement each other. Overall, our graph-based approach can be used as the first step towards a better understanding of the language usage in medical domain as well as for providing better services and recommendations to the users of the health care portal.

1.6 Summary of Contributions

This section summarizes the contributions of the papers included in this thesis.

1.6.1 PAPER I

In this paper, we show that an email network generated from legitimate email traffic collected on an Internet backbone link (a ham network) can be modeled as a scale-free small-world network similar to other social and interaction networks. We also show the similarities and the differences in the structure of ham and spam

networks and how they change over time. We reveal that the anti-social behavior of spam is not hidden in a mixture of email traffic and causes anomalies (outliers) in the structural properties of email networks. Moreover, we propose a simple method for identifying the nodes that correspond to outliers in the degree distribution of email networks and show that they are mainly sending spam.

1.6.2 PAPER II

In this paper, we study the community structure of ham, spam, and email networks generated from real email traffic and compare a number of well-known community detection algorithms for identifying the communities of these networks. Our experiments reveal that there is a trade-off in creating high structural quality and high logical quality communities. We propose to evaluate the logical quality of the communities based on the homogeneity of the edges inside each community, and show that regardless of the approaches used to define and extract communities, the algorithms that create communities with similar granularity and size distribution also achieve similar structural and logical qualities. We also show that the most successful community detection algorithm for achieving high logical quality (i.e., clustering ham and spam emails into distinct communities), finds overlapping communities by partitioning the edges of the network instead of the nodes.

1.6.3 PAPER III

In this paper, we extend and complement the previous work on community-based intrusion detection. We hypothesize that misbehaving nodes tend to *belong to multiple communities*. To investigate our hypothesis, we consider different definitions for communities, and propose a framework in which different types of community detection algorithms can be used as the basis for network anomaly and intrusion detection. We propose two enhancement methods for adding auxiliary communities over the disjoint communities identified by non-overlapping community detection algorithms. We show that by using our enhancement methods, it is possible to use traditional community detection algorithms for identifying anomalies in network traffic which is in contrast to the observations in [64].

Moreover, we propose a framework that allows us to incorporate communities identified by overlapping algorithms for identifying anomalous nodes that belong to multiple communities. We show that the algorithms which tend to identify coarse-grained communities are not suitable for network misbehavior detection. We also propose to use application-specific filters to filter out legitimate nodes which can naturally belong to several communities. Our experiments reveal that our framework is suitable for identifying scanning nodes from network flow traffic as well as spammers from email traffic.

1.6.4 PAPER IV

In this paper, we propose a novel distributed seed selection algorithm for local overlapping community detection. We define a similarity score using the similarity indices from link prediction techniques and propose an algorithm in which each node compares its similarity score with all its neighbors, and the nodes which have the highest score in their neighborhood are selected as seeds. We show that this algorithm succeeds in selecting a small number of very good seeds which are expanded into high quality communities but cannot cover the whole network. We also propose to use graph coloring for enhancing our local seed selection algorithm in order to improve the coverage. We propose a *biased graph coloring* algorithm in which the nodes with high similarity score are assigned a specific color and color conflicts between neighbors are resolved at random. Our experiments using large-scale real-world social networks show that our seeding algorithm is fast, and leads to high quality communities with a good coverage of the networks.

1.6.5 PAPER V

In this paper, we create a word co-occurrence network from query log files obtained from a medical and health care portal. We show that this network has the same structural and temporal properties that other information networks exhibit. We use a local overlapping community detection algorithm to identify the communities in the co-occurrence network. We also use the semantic labels assigned to the queries in the log files and define semantic communities which are communities of words which have co-occurred with a semantic label. We compare the graph communities with the semantic communities and show that our graph-based analysis of queries can improve and complement the semantic analysis. We also study how the length of the time window in which queries are observed can affect our graph-based analysis.

1.7 Conclusions and Future Work

In this thesis, we have looked into algorithms and methods for analyzing networks generated from large-scale real-world datasets. Particularly, we have focused on the community structure of networks and have looked into the challenges and the applications of community detection algorithms.

One of the challenges in identifying communities in a network is the selection of the most suitable algorithm for the network, since different algorithms use different definitions for communities and use different methods for identifying the communities. In this thesis we have performed an empirical comparison and evaluation of a number of different community detection algorithms and show that there is a trade-off between the structural and the logical quality of the communities identified by different algorithms. Therefore, an algorithm which can create communities

with very high structural quality might not be the most suitable algorithm for the application at hand, for example, separating different types of edges into distinct communities.

Another challenge in using community detection algorithms for analysis of large datasets is scalability. It has been shown that local seed expansion algorithms are very successful in fast and scalable detection of high quality communities. In this thesis, we have proposed a fast local seed selection algorithm which can be used as a pre-processing step for local community detection using seed expansion. Our algorithm can dramatically reduce the execution time of community detection while preserving the quality of the identified communities and achieving a good coverage of the network. Moreover, there are many interesting trade-offs between the number of selected seeds, the quality, and the coverage of communities which can be further studied. Another property which can further be taken into account for seed selection is to reduce the number of duplicate communities.

In addition to investigating and addressing some of the challenges of community detection, we have also looked into some of the applications of network analysis and community detection. One of the applications which has been considered in this thesis is identifying the source of unsolicited email. Our goal has been to reveal the differences and similarities in the communication patterns of legitimate and unsolicited email by mining email networks generated from traffic seen on an Internet backbone link. To pursue this goal, we have taken a social network analysis approach and show that the behavior of spam senders causes anomalies in the structural properties of email networks, and these anomalies can be detected using an outlier detection approach. We can also show that spam and ham, which are mixed in the observed traffic, can be separated into distinct communities by deploying a link community detection algorithm. Moreover, we have proposed a framework for network misbehavior detection which takes advantage of overlapping communities for identifying sources of spam as well as sources of other types of malicious traffic such as scanning. We are able to show that misbehaving nodes belong to multiple communities and they can be identified by either using overlapping community detection algorithms or by enhancing non-overlapping algorithms with auxiliary communities.

The proposed approaches in this thesis for identifying sources of misbehavior are promising and can potentially be used to complement existing anti-spam and intrusion detection methods. The advantage of deploying our approaches is that they provide us with the possibility of stopping unwanted traffic closer to its source by merely observing the communication patterns of network traffic, for example email communications. However, there is more work to be done before our findings can be deployed practically as part of a working anti-spam or intrusion detection tool. One desirable future direction is to investigate how our methods can be combined with each other to be used as a stand-alone detection system or in cooperation with existing tools. One possibility is to deploy a network device that monitors the traffic on a link and that is able to tag suspicious traffic or populate

a blacklist. Moreover, a study of the robustness of our findings in order to see how easy it is for the spammers or intruders to change their sending behavior and how easy it is to evade detection is another future research area.

Another goal of this thesis has been to improve community detection algorithms so that they could be used for different applications. We have introduced auxiliary communities to enhance existing non-overlapping community detection algorithms in order to identify sources of misbehavior from real network traffic. However, our approach can potentially be extended for converting disjoint communities into overlapping communities which will allow the use of existing non-overlapping community detection algorithms for identifying overlapping communities.

In this thesis, we have also shown how to use network mining and community detection methods to analyze other types of large datasets such as the query logs obtained from a Swedish health care portal. A future direction is to improve our graph-based query analysis by improving the pre-processing of the data, for example by representing different variations of words with a single node in the word co-occurrence network, filtering out non-medical related words, and introducing edge weights based on the frequency of word co-occurrences. Moreover, other information from the logs can be deployed to better understand the language used by users and to be able to improve the search experience of the users by providing better suggestions and recommendations to them.

Overall, with advances in technology and computation and proliferation of smart and mobile devices, new opportunities for collecting and analyzing big data emerge and more and more applications can benefit from the extracted knowledge from the data. Therefore, there is an increasing need for fast, dynamic, and scalable solutions which also open more research questions. One of the challenges is to design new network mining algorithms and to improve existing ones to run in parallel and in distributed settings. The designed parallel and distributed algorithms also need to cope with the lack of global knowledge of the networks, as well as the dynamically changing structure of networks. Moreover, there is also a need for improving the quality of the network mining algorithms, particularly community detection algorithms. Recent studies using ground truth data have revealed that existing community detection algorithms are not very successful in identifying the real communities in large networks, therefore new approaches to community detection which for example take non-structural properties of the networks into account, are desirable. Another interesting future research direction is to develop efficient methods, such as visualization, for interpreting the output of different graph algorithms, to allow better understanding of the structure of networks and identifying interesting patterns and anomalies. Finally, extending the applicability of network mining algorithms to more real-time domains and applications is another challenging future direction.

Bibliography

[1] MEJ Newman and Juyong Park, "Why social networks are different from other types of networks," *Physical Review E*, vol. 68, no. 3, Sept. 2003.

[2] Paul Erdos and Alfred Renyi, "On The Evolution of Random Graphs," *Publication of the Mathematical Institute of the Hungarian Academy of Science*, vol. 5, pp. 17–61, 1960.

[3] D J Watts and S H Strogatz, "Collective Dynamics of 'Small-World' Networks.," *Nature*, vol. 393, no. 6684, pp. 440–2, June 1998.

[4] S Milgram, "The Small World Problem," *Psychology today*, vol. 2, pp. 60–67, 1967.

[5] A.L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509, 1999.

[6] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 251–262, 1999.

[7] Reka Zsuzsanna Albert, *Statistical Mechanics of Complex Networks*, Ph.D. thesis, University of Notre Dame, 2001.

[8] Leman Akoglu, *Mining and Modeling Real-world Networks : Patterns , Anomalies , and Tools*, Ph.D. thesis, Carnegie Mellon University, 2012.

[9] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos, "Graphs Over Time: Densification Laws, Shrinking Diameters and Possible Explanations," in *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining - KDD '05*. 2005, p. 177, ACM Press.

[10] W.W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.

[11] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, "Fast Unfolding of Communities in Large Networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. P10008, Oct. 2008.

[12] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, pp. 1–8, July 2009.

[13] Santo Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, Feb. 2010.

[14] Ullas Gargi and Wenjun Lu, "Large-Scale Community Detection on YouTube for Topic Discovery and Exploration," in *Proceedings of the Fifth International Conference on Weblogs and Social Media*. 2011, The AAAI Press.

[15] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann, "Link communities reveal multiscale complexity in networks.," *Nature*, vol. 466, no. 7307, pp. 761–4, Aug. 2010.

[16] Jierui Xie, S Kelley, and BK Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Computing Surveys*, vol. 45, no. 4, 2013.

[17] Martin Rosvall and Carl T Bergstrom, "Maps of random walks on complex networks reveal community structure.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–23, Jan. 2008.

[18] Martin Rosvall and Carl T Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems.," *PloS one*, vol. 6, no. 4, pp. e18209, Jan. 2011.

[19] Peter Ronhovde and Zohar Nussinov, "Multiresolution community detection for megascale networks by information-based replica correlations," *Physical Review E*, vol. 80, no. 1, pp. 1–18, July 2009.

[20] Stijn VAN Dongen, *Graph Clustering by Flow Simulation*, Ph.D. thesis, University of Utrecht, The Netherlands, 2000.

[21] Jierui Xie and BK Szymanski, "Towards Linear Time Overlapping Community Detection in Social Networks," in *the 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*. 2012, pp. 25–36, Springer-Verlag.

[22] Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, and Santo Fortunato, "Finding statistically significant communities in networks.," *PloS one*, vol. 6, no. 4, pp. e18961, Jan. 2011.

[23] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi, "DEMON: a local-first discovery method for overlapping communities," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. 2012, p. 615, ACM Press.

[24] Jaewon Yang and Jure Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2012, pp. 1–8.

[25] Reid Andersen and Kevin Lang, "Communities from seed sets," in *Proceedings of the 15th international conference on World Wide Web - WWW '06*. 2006, p. 223, ACM Press.

[26] Daniel A Spielman and Shang-Hua Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *Proceedings of the 36th annual ACM symposium on Theory of computing - STOC '04*. 2004, p. 81, ACM Press.

[27] Aaron Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, no. 2, pp. 026132, Aug. 2005.

[28] Sucheta Soundarajan and John E Hopcroft, "Use of Local Group Information to Identify Communities in Networks," *ACM Transactions on Knowledge Discovery from Data (to appear)*, 2014.

[29] Joyce Jiyoung Whang, David F Gleich, and Inderjit S Dhillon, "Overlapping community detection using seed set expansion," in *Proceedings of the 22nd ACM international Conference on information & knowledge management - CIKM '13*. 2013, pp. 2099–2108, ACM Press.

[30] Marek Ciglan, Michal Laclavík, and Kjetil Nø rvåg, "On community detection in real-world networks and the importance of degree assortativity," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*, New York, New York, USA, 2013, p. 1007, ACM Press.

[31] Christian L Staudt and Henning Meyerhenke, "Engineering High-Performance Community Detection Heuristics for Massive Graphs," in *International Conference on Parallel Processing*, 2013.

[32] Satu Elisa Schaeffer, "Graph Clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, Aug. 2007.

[33] M E J Newman, "Modularity and community structure in networks.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, no. 23, pp. 8577–82, June 2006.

[34] Helio Almeida, Dorgival Guedes, Wagner Meira Jr., and Mohammad J. Zaki, "Is There a Best Quality Metric for Graph Clusters?," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD*, Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, Eds. 2011, pp. 44–59, Springer-Verlag.

[35] M Girvan and M E J Newman, "Community structure in social and biological networks.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–6, June 2002.

[36] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi, "Benchmark graphs for testing community detection algorithms," *Physical Review E*, vol. 78, no. 4, pp. 1–5, Oct. 2008.

[37] Bruno Abrahao, Sucheta Soundarajan, John Hopcroft, and Robert Kleinberg, "On the separability of structural classes of communities," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. 2012, p. 624, ACM Press.

[38] Usha Raghavan, Réka Albert, and Soundar Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, pp. 036106, Sept. 2007.

[39] Chun Yew Cheong, Huynh Phung Huynh, David Lo, Rick Siow, and Mong Goh, "Hierarchical Parallel Algorithm for Modularity-Based Community Detection Using GPUs," in *Proceedings of the 19th international conference on Parallel Processing*, 2013, pp. 775–787.

[40] Jyothish Soman and Ankur Narang, "Fast Community Detection Algorithm with GPUs and Multicore Architectures," *2011 IEEE International Parallel & Distributed Processing Symposium*, pp. 568–579, May 2011.

[41] Konstantin Kuzmin, S. Yousaf Shah, and Boleslaw K. Szymanski, "Parallel Overlapping Community Detection with SLPA," *2013 International Conference on Social Computing*, pp. 204–212, Sept. 2013.

[42] Jaewon Yang and Jure Leskovec, "Overlapping community detection at scale," in *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13*, New York, New York, USA, 2013, p. 587, ACM Press.

[43] Arnau Prat-pérez and David Dominguez-sal, "High Quality , Scalable and Parallel Community Detection for Large Real Graphs Categories and Subject Descriptors," in *WWW*, 2014.

[44] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.

[45] David F. Gleich and C Seshadhri, "Vertex neighborhoods, low conductance cuts, and good seeds for local community methods," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. 2012, pp. 597–605, ACM Press.

[46] Qiong Chen and Ming Fang, "An Efficient Algorithm for Community Detection in Complex Networks," in *the 6th Workshop on Social Network Mining and Analysis*, 2012.

[47] Anirudh Ramachandran and Nick Feamster, "Understanding the Network-Level Behavior of Spammers," in *Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications - SIGCOMM '06*, New York, New York, USA, 2006, p. 291, ACM Press.

[48] Christian Kreibich, Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage, "On the Spam Campaign Trail," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, June 2008, vol. 453, pp. 697–8.

[49] Zhenhai Duan, K. Gopalan, and X. Yuan, "Behavioral Characteristics of Spammers and Their Network Reachability Properties," in *2007 IEEE International Conference on Communications*. June 2007, pp. 164–171, IEEE.

[50] Martin Abadi, Mike Burrows, Mark Manasse, and Ted Wobber, "Moderately hard, memory-bound functions," *ACM Transactions on Internet Technology*, vol. 5, no. 2, pp. 299–327, May 2005.

[51] Michael Walfish, J D Zamfirescu, Hari Balakrishnan, David Karger, and Scott Shenker, "Distributed Quota Enforcement for Spam Control," in *Proceedings of the 3rd conference on Networked Systems Design & Implementation*. 2006, USENIX Association.

[52] Evan Harris, "The Next Step in the Spam Control War: Greylisting," *http://projects.puremagic.com/greylisting/whitepaper.html*, 2003.

[53] Anirudh Ramachandran, Nick Feamster, and Santosh Vempala, "Filtering Spam with Behavioral Blacklisting," in *Proceedings of the 14th ACM conference on Computer and communications security - CCS '07*, New York, New York, USA, 2007, p. 342, ACM Press.

[54] Guofei Gu, Roberto Perdisci, Junjie Zhang, and Wenke Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection," in *the 17th conference on Security symposium*. 2008, pp. 139–154, USENIX Association.

[55] Robert Beverly, "Exploiting Transport-Level Characteristics of Spam," *5th Conference on Email and Anti-Spam (CEAS)*, 2008.

[56] P.O. Boykin and V.P. Roychowdhury, "Leveraging social networks to fight spam," *Computer*, vol. 38, no. 4, pp. 61–68, Apr. 2005.

[57] Luiz H Gomes, Rodrigo B Almeida, and Luis M A Bettencourt, "Comparative Graph Theoretical Characterization of Networks of Spam and Legitimate Email," in *Conference on Email and Anti-Spam (CEAS)*, 2005.

[58] Ho-yu Lam and Dit-yan Yeung, "A Learning Approach to Spam Detection based on Social Networks," in *Conference on Email and Anti-Spam (CEAS)*, 2007.

[59] Chi-Yao Tseng and Ming-Syan Chen, "Incremental SVM Model for Spam Detection on Dynamic Email Social Networks," *2009 International Conference on Computational Science and Engineering*, pp. 128–135, 2009.

[60] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt, "Scale-free topology of e-mail networks," *Physical Review E*, vol. 66, no. 3, pp. 1–4, Sept. 2002.

[61] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos, "Graph Evolution: Densification and Shrinking Diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 2–es, Mar. 2007.

[62] Gueorgi Kossinets and Duncan J Watts, "Empirical Analysis of an Evolving Social Network," *Science (New York, N.Y.)*, vol. 311, no. 5757, pp. 88–90, Jan. 2006.

[63] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. Sep, pp. 1–72, 2009.

[64] Qi Ding, Natallia Katenka, Paul Barford, Eric Kolaczyk, and Mark Crovella, "Intrusion as (anti)social communication," in *Proceedings of the 18th ACM SIGKDD conference on Knowledge discovery and data mining - KDD '12*, 2012, p. 886.

[65] Judit Bar-Ilan, Zheng Zhu, and Mark Levene, "Topic-specific analysis of search queries," in *Proceedings of the 2009 workshop on Web Search Click Data - WSCD '09*. 2009, pp. 35–42, ACM Press.

[66] Ricardo Baeza-Yates, "Graphs from Search Engine Queries," in *Theory and Practice of Computer Science*. 2007, vol. 4362, pp. 1–8, Springer.

[67] Luiz Henrique Gomes, Cristiano Cazita, Jussara M. Almeida, Virgílio Almeida, and Wagner Meira, "Workload models of spam and legitimate e-mails," *Performance Evaluation*, vol. 64, no. 7-8, pp. 690–714, Aug. 2007.

[68] Yinglian Xie, Fang Yu, Kannan Achan, Eliot Gillum, Moises Goldszmidt, Ted Wobber, C Computer Communication, and Networks Network, "How Dynamic are IP Addresses?," in *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'07)*. 2007, pp. 301–312, ACM.

[69] Yehonatan Cohen, Daniel Gordon, and Danny Hendler, "Early Detection of Outgoing Spammers in Large-Scale Service Provider Networks," in *Detection of Intrusions and Malware, and Vulnerability Assessment*. 2013, pp. 83–101, Springer Berlin Heidelberg.

[70] Abhinav Pathak, Y Charlie Hu, and Z Morley Mao, "Peeking into Spammer Behavior from a Unique Vantage Point," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*. 2008, pp. 3:1—-3:9, USENIX Association.

[71] Dominik Schatzmann, Martin Burkhart, and Thrasyvoulos Spyropoulos, "Inferring Spammers in the Network Core," in *Proceedings of the 10th International Conference on Passive and Active Network Measurement*. 2009, pp. 229–238, Springer-Verlag.

[72] Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M Voelker, Vern Paxson, and Stefan Savage, "Spamalytics: An Empirical Analysis of Spam Marketing Conversion," in *Proceedings of the 15th ACM conference on Computer and communications security - CCS '08*, New York, New York, USA, 2008, p. 3, ACM Press.

[73] Christian Kreibich, C Kanich, and K Levchenko, "Spamcraft: An inside look at spam campaign orchestration," in *the 2nd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*. 2009, USENIX Association.

[74] Wolfgang John, Sven Tafvelin, and Tomas Olovsson, "Passive internet measurement: Overview and guidelines based on experiences," *Computer Communications*, vol. 33, no. 5, pp. 533–550, Mar. 2010.

[75] Farnaz Moradi, Magnus Almgren, Wolfgang John, Tomas Olovsson, and Philippas Tsigas, "On Collection of Large-Scale Multi-Purpose Datasets on Internet Backbone Links," in *Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, 2011.

[76] "SUNET (Swedish University Network), http://www.sunet.se/," .

[77] J. Klensin, "Simple Mail Transfer Protocol, Request for Comments, RFC 5321 (Draft Standard)," Oct. 2008.

[78] DShield, "Recommended block list," 2010.

[79] SRI International Malware Threat Center, "Most aggressive malware attack source and filters," 2010.

[80] Magnus Almgren and Wolfgang John, "Tracking Malicious Hosts on a 10Gbps Backbone Link," in *15th Nordic Conference in Secure IT Systems*, 2010.

[81] "Stanford Large Network Dataset Collection, http://snap.stanford.edu/data/index.html," .

[82] Ömer Yüksel, "Local Community Detection in Complex Networks," *Master thesis, Chalmers University of Technology*, 2013.

[83] Reid Andersen, Fan Chung, and Kevin Lang, "Local Graph Partitioning using PageRank Vectors," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 2006, pp. 475–486, IEEE.

Part II

PAPERS

PAPER I

Farnaz Moradi, Tomas Olovsson, Philippas Tsigas

Towards Modeling Legitimate and Unsolicited Email Traffic Using Social Network Properties

Proceedings of the 5th Workshop on Social Network Systems (SNS'12),
pp. 9:1 - 9:6, ACM, Bern, Switzerland, April, 2012.

In order to comply with the thesis layout, this paper has been reformatted.

2 Towards Modeling Legitimate and Unsolicited Email Traffic Using Social Network Properties

Identifying unsolicited email based on their network-level behavior rather than their content have received huge interest. In this study, we investigate the social network properties of large-scale *email networks* generated from real email traffic to reveal the properties that are indicative of spam as opposed to the expected legitimate behavior.

By analyzing the structural and temporal properties of the email networks we confirm that legitimate email traffic generates a small-world, scale-free network similar to other social networks. However, email traffic as a whole contains unsolicited email, thus the structure of email networks deviates from that of social networks. Our study points out the distinctive characteristics of spam traffic and reveals that the anomalies in the structural properties of email networks are due to the unsocial behavior of spam.

2.1 Introduction

Eliminating the excessive amount of unsolicited *spam* which is consuming network and mail server resources is quite challenging. These email communications are mostly originated from botnets of compromised machines [1, 2] that are also likely the source of other malicious activities on the Internet. Although current anti-spam tools are efficient in hiding spam from users' mailboxes, there is a clear need for moving the defense against spam as close to its source as possible. Therefore, it is necessary to understand the network-level behavior of spam and how it differs from legitimate traffic in order to design anti-spam mechanisms that can identify spamming bots on the network. In this paper, we study the network-level behavior of email by examining real email traffic captured on an Internet backbone link. From the collected traffic, we have generated *email networks* in which the nodes represent email addresses and the edges represent email communications. To the best of our knowledge, this is the largest email traffic dataset used to study the

structure of email networks which contain both legitimate (*ham*) and unsolicited email traffic.

In this study, we show that the legitimate email traffic exhibit the same structural properties that other social and interaction networks (e.g., on-line social networks, the Internet topology, the Web, and phone call graphs) typically exhibit, therefore, it can be modeled as a *scale-free*, *small-world* network. We also show that the email traffic containing spam cannot be modeled similarly, and because the unsocial behavior of spam is not hidden behind the social behavior of legitimate traffic, the structure of email networks containing both ham and spam differ from other social networks. Moreover, we show that the temporal variations in the social network properties of email traffic can reveal more distinct properties of the behavior of spam.

In this study our goal is to identify the differences in the social network properties of spam and ham traffic, and leverage these differences to spot the abusive nodes in the network.

The remainder of this paper is organized as follows. Section 2.2 presents the related works. The collected email datasets and their properties are discussed in Section 2.3. Section 2.4 presents and discusses the observed structural and temporal properties of our email networks. Section 2.5 presents a method to spot spam senders in the structure of email networks. Finally, Section 2.6 concludes the paper.

Table 2.1: *Summary of the datasets of related works in comparison to our datasets.*

Reference	Nodes $\|V\|$	Edges $\|E\|$	Email types	Dataset
Ebel et al. [3] (2002)	59,812	86,130	ham	log files of the mail server at Kiel University
Gomes et al. [4] (2005)	265,144	615,102	ham & spam	log files of a university mail server in Brazil
Boykin et al. [5] (2005)	-	-	ham & spam	headers of emails in one user's inbox
Lam et al. [6] (2007)	9,150	-	ham & simulated spam	Enron dataset and simulated spam
Tseng et al. [7] (2009)	637,064	2,865,633	ham & spam	a mail server in National Taiwan University
Leskovec et al. [8] (2007)	35,756	123,254	ham	emails of a EU research institution
Kossinets et al. [9] (2006)	43,553	*14,584,423	ham	emails at a large university
This paper, *dataset A*	10,544,647	21,562,306	ham & spam	Internet backbone SMTP traffic
This paper, *dataset B*	4,525,687	8,709,216	ham & spam	Internet backbone SMTP traffic

* Total number of emails exchanged during 355 days (separate graphs within time windows of 60 days)

Table 2.2: *Statistics of the collected data for* dataset A.

	Packets	Flows	Email	Ham	Spam	Rejected	Src[1]	Dst[2]	Domains[3]
Incoming	$627M$	$35M$	19302206	1319273	1663698	16319235	7780897	3169712	446694
Outgoing	$170M$	$12M$	729553	213306	202879	313368	324657	408429	167907

[1] Distinct sender email addresses. [2] Distinct receiver email addresses. [3] Distinct domain names in email addresses.

2.2 Related Work

Social network analysis has been widely used in order to study the structural properties of real-world networks such as the Web graph [10], the Internet topology [11], phone call and SMS networks [12], and online social networks [13]. The structure of email networks was first studied by Ebel et al. [3] showing that an email network generated from mail server log files of a university is a scale-free, small-world network. Leskovec et al. [8] studied the evolution of a variety of real networks, including an email network of a large institution, and observed that these social networks densify over time and their diameter shrinks, while their power law degree distribution exponent remains constant.

Deployment of social network analysis for discriminating spammers and legitimate users was first proposed in Boykin et al. [5]. They generated an email network from email headers in one user's mailbox and found distinguishing structural properties of spam and ham messages. Gomes et al. [4] generated distinct graphs from ham and spam email collected from mail server log files of their university department, and found graph theoretical metrics that structurally and dynamically differ for spam and ham. Lam et al. [6] and Tseng et al. [7] extracted different structural features from email social networks and deployed them in building learning-based spam detection systems.

Table 2.1 summarizes the properties of the email networks studied in the related works. All of the above studies have taken place on relatively limited email datasets. In addition to previous studies, we perform an analysis of the structural and temporal characteristics of email networks, reveal properties that distinguish ham from spam, compare our observations with previous studies, and show how our findings could reveal the spam sending nodes in the email networks.

2.3 Data Collection and Pre-processing

In this study we have used two distinct email datasets to generate email networks. The datasets were created from passively captured SMTP packets on a 10 Gbps link of the core-backbone of the SUNET[1]. Each dataset was collected during 14 consecutive days with a year time span between the collections. Throughout the

[1]Swedish University Network (`http://www.sunet.se/`) serves as a backbone for university traffic, student dormitories, research institutes, etc. exchanging large amount of traffic with commercial companies.

paper, we refer to the larger dataset as *dataset A*, and the smaller dataset as *dataset B*.

The unusable email flows, including those with no payload or missing packets and encrypted communications were pruned from the datasets. The remaining emails were first classified as being either *accepted* (delivered by the receiving mail server) or *rejected* (unfinished SMTP command exchange phase and consequently not containing any email headers and body). Rejection is generally the result of spam pre-filtering strategies deployed by mail servers (e.g., blacklisting, greylisting, DNS lookups). Then, all accepted email communications were classified to be either *spam* or *ham* to establish a ground truth for our study. Similar to [4, 7], the classification was done by a well-trained filtering tool[2]. Finally, all email addresses were anonymized and email contents were discarded in order to preserve privacy.

After data collection and pre-processing, a number of email networks have been generated from the datasets. In an email network the email addresses, which are extracted from the SMTP commands ("MAIL FROM" and "RCPT TO"), represent the nodes, and the exchanged emails represent the edges. In order to study and compare the characteristics of different categories of email, from each dataset we have generated a *ham network*, a *spam network*, and a *rejected network*, in addition to the complete *email network*.

Table 2.2 summarizes the properties of the dataset A as an example. More details on the measurement location, data collection, and pre-processing can be found in [14].

2.4 Structural and Temporal Properties of Email Networks

In this section we briefly introduce the most significant structural and temporal properties of social networks.

Degree distribution. The degree distribution of a network is the probability that a randomly selected node has k edges. In a *power law distribution*, the fraction of nodes with degree k is $n(k) \propto k^{-\gamma}$, where γ is a constant exponent. Networks characterized by such degree distribution are called *scale-free* networks. Many real networks such as the Internet topology [11], the Web [10], phone call graphs [12], and on-line social networks [13] are scale free.

Average path length. In *small-world* networks any two nodes in the network are likely to be connected through a short sequence of intermediate nodes, and the network diameter shrinks as the network grows [8].

Clustering coefficient. In addition to a short average path length, *small-world* networks have high clustering coefficient values [15]. The clustering coef-

[2]The SpamAssassin (`http://spamassassin.apache.org`) was in use for a long time in our University mail server and it incurs a false positive rate of less than 0.1%, and the detection rate of 91.4% after 94% of the spam being rejected by blacklists.

ficient of a node v is defined as $C_v = 2E_v/(k_v(k_v - 1))$, where, k_v denotes the number of neighbors of v, $k_v(k_v - 1)/2$ the maximum number of edges that can exist between the neighbors, and E_v the number of the edges that actually exist. The average C_v of a social network shows to what extent friends of a person are also friends with each other and its value is independent of the network size [16].

Connected components. A connected component (CC) is a subset of nodes of the network where a path exists between any pair of them. As social networks grow a giant CC (GCC), which contains the vast majority of the nodes in the network, emerges in the graph and its size increases over time [16]. Moreover, the distribution of CC size for some social networks follows a power law pattern [10, 12].

2.4.1 Measurement Results

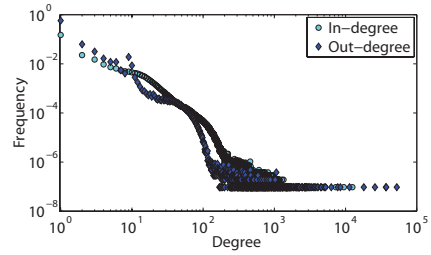
In the following the observed structural and temporal properties of our email networks are presented. These properties can be used in order to model the behavior of legitimate traffic and to find the distinguishing properties of the unsocial behavior of spam. Although the duration of our data collections is not long enough to study the evolution of email networks, it is still possible to track the changes in the structure of email networks in order to better understand the distinct characteristics of ham and spam traffic.

Degree distribution. Figures 2.1(a) and 2.1(e) show that none of the email networks generated from datasets A and B exhibit a power law degree distribution in all points. However, the ham networks generated from each of the datasets are scale free as their degree distribution closely follow the distribution $n(k) \propto k^{-\gamma}$ with $\gamma_A = 2.7$ and $\gamma_B = 2.3$, respectively [3]. The in-degree (out-degree) distribution for ham networks, which are shown in Figures 2.1(b) and 2.1(f), also follows a power-low distribution with $\gamma_{A_{in}} = 3.2$ ($\gamma_{A_{out}} = 2.3$) and $\gamma_{B_{in}} = 3.2$ ($\gamma_{B_{out}} = 2.1$), respectively. Moreover, in contrast to previous studies [4, 5], neither the spam, nor the rejected networks are completely scale free (Figures 2.1(c), 2.1(g), 2.1(d), and 2.1(h)).

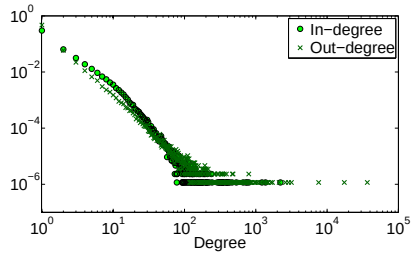
Figure 2.2(a) and 2.2(e) show that the shape of the degree distributions of the complete email networks may change over time as the networks grow. The shape of the degree distribution of spam and rejected networks can also change over time (Figures 2.2(c), 2.2(g), 2.2(d), and 2.2(h)). However, the ham networks always follow a power law distribution with an almost constant exponent (Figures 2.2(b) and 2.2(f)).

Clustering coefficient. The observed average clustering coefficients for our ham (spam) networks generated from both dataset are quite similar: $C_{A_{ham}} = 9.92 \times 10^{-3}$ ($C_{A_{spam}} = 1.59 \times 10^{-3}$) and $C_{B_{ham}} = 9.80 \times 10^{-3}$ ($C_{B_{spam}} = 1.54 \times 10^{-3}$). These values, similar to small-world networks, are significantly greater than that of random networks with the same number of nodes and average number

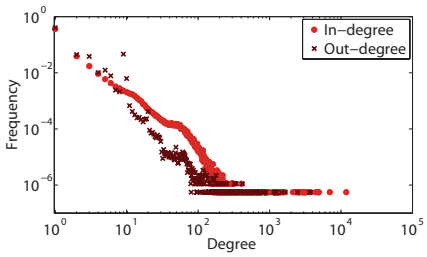
[3]The power law fits were calculated using the Maximum Likelihood estimator for power law and Kolmogorov-Smirnov (KS) goodness-of-fit as presented in [17].



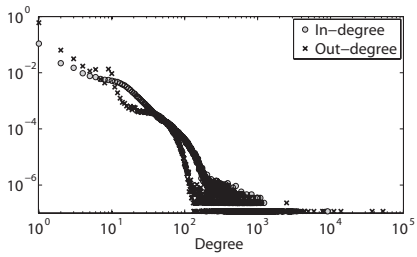
(a) Email network (A)



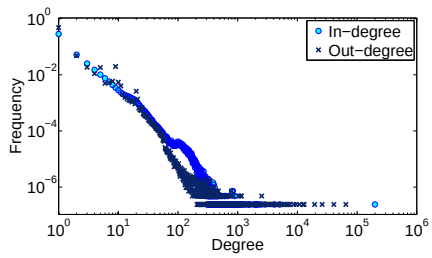
(b) Ham network (A)



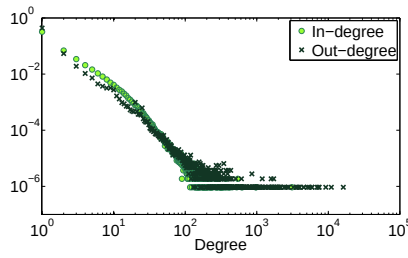
(c) Spam network (A)



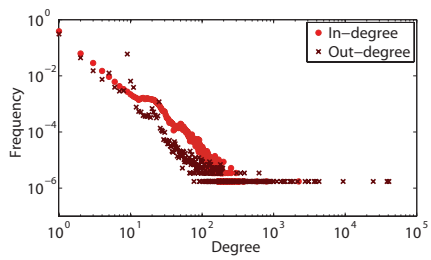
(d) Rejected network (A)



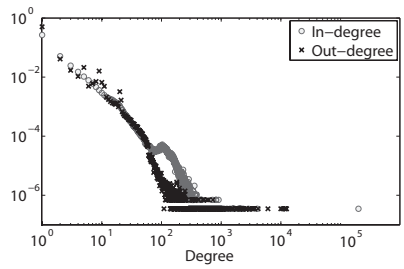
(e) Email network (B)



(f) Ham network (B)

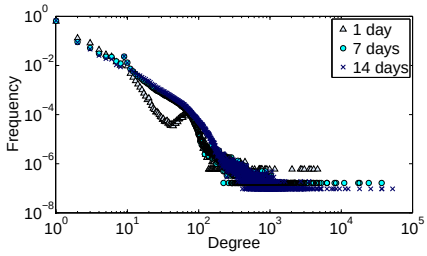


(g) Spam network (B)

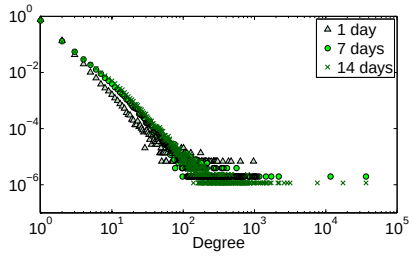


(h) Rejected network (B)

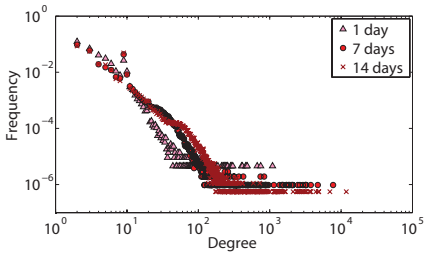
Figure 2.1: *Only the ham network is scale free as the other networks have outliers in their degree distribution.*



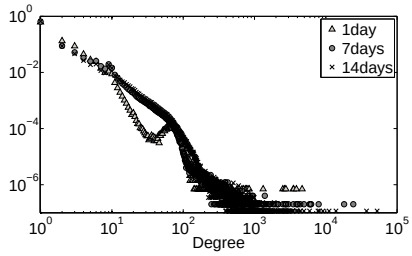
(a) Email network (A)



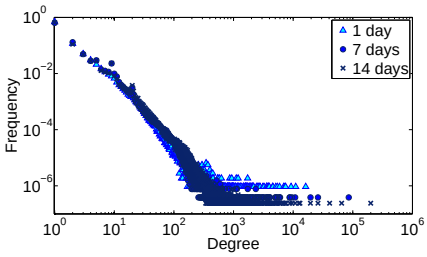
(b) Ham network (A)



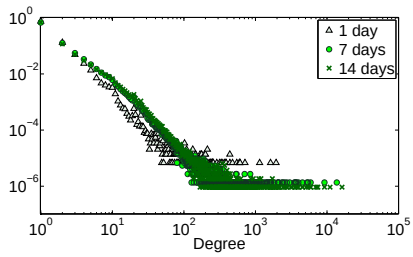
(c) Spam network (A)



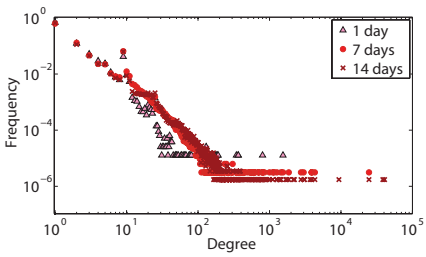
(d) Rejected network (A)



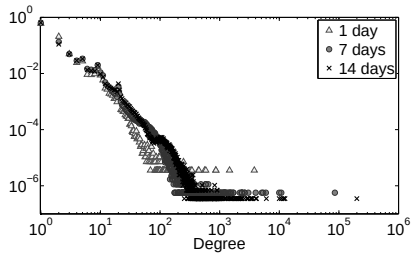
(e) Email network (B)



(f) Ham network (B)



(g) Spam network (B)



(h) Rejected network (B)

Figure 2.2: *Temporal variation of in the degree distribution of the email networks.*

of edges per node, and as Figures 2.3(b) and 2.3(f) show they remain relatively constant as the networks grow.

Average path length. The ham and spam networks generated from both datasets have short average path lengths, $\langle l \rangle$, as expected in small-world networks: $\langle l_{ham_A} \rangle = 7.0$, $\langle l_{spam_A} \rangle = 8.5$, $\langle l_{ham_B} \rangle = 6.7$, and $\langle l_{spam_B} \rangle = 7.8$. Figures 2.3(a) and 2.3(e) show that $\langle l \rangle$ decreases for all networks as they grow, confirming the shrinking diameter phenomenon observed in [8] for other social networks.

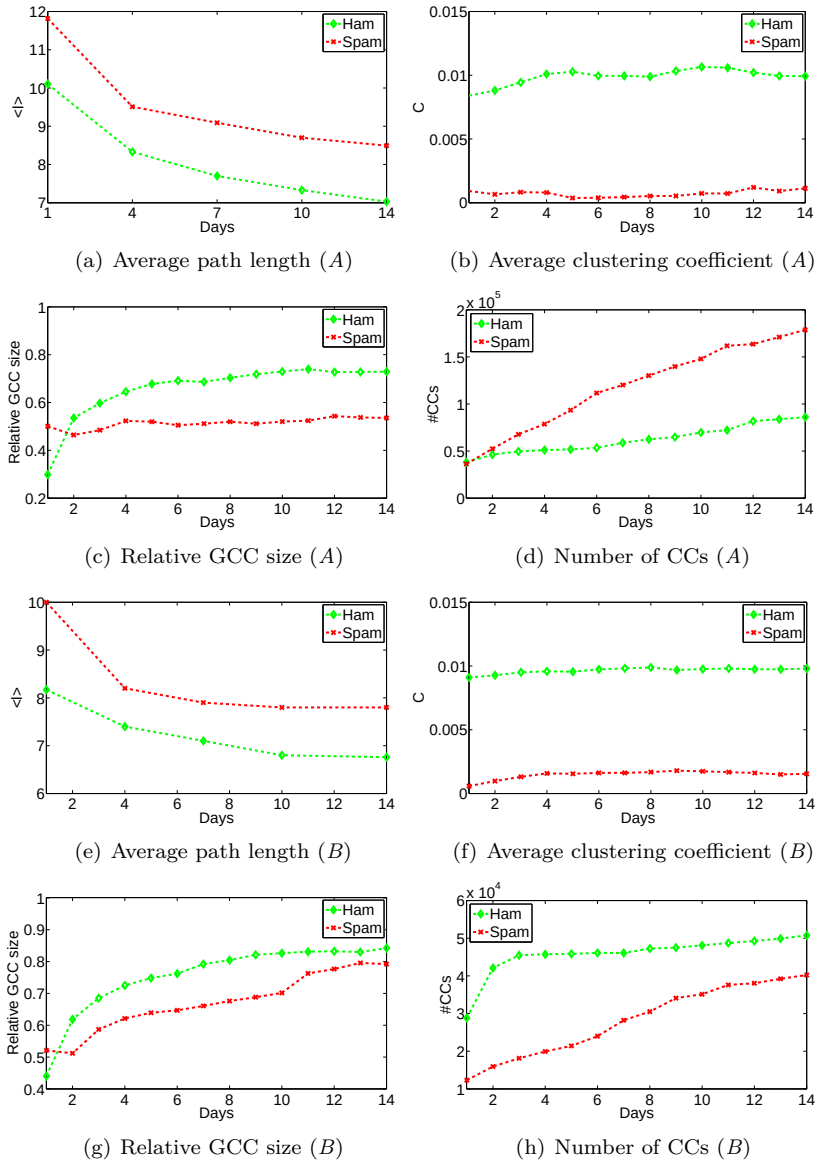
Connected components. Figure 2.4 shows the distribution of the size of the CCs for ham and spam networks. It can be seen that the GCCs of the networks are orders of magnitude larger than the other CCs. The distribution of the CC size for the ham network, similar to Web [10] and phone call graphs [12], follows a power law pattern, but the spam network have outliers in their distribution. Figures 2.3(d) and 2.3(h) show that the number of CCs in all of the ham and the spam networks increases over time, but this increase is much faster for spam. Moreover, as shown in Figure 2.3(c), the respective size of the GCC of the networks generated from dataset A increases for the ham but does not change much for the spam network. However, although the ham network generated from dataset B shows exactly the same behavior (Figure 2.3(g)), the spam network shows an increase in the percentage of nodes in its GCC over time.

2.4.2 Discussion

In the following paragraphs we briefly discuss our observations regarding the structure of email networks and discuss to what extent our dataset is representative for the structural and temporal analysis of email networks.

Table 2.3 summarizes the observed similarities and differences in the structure of the ham and spam networks. Although the studied datasets differ in size and collection time, our observations reveal that legitimate email always exhibit the structural properties that are similar to other social and interaction networks. Previous studies on the structure of legitimate email networks have also shown that these networks can be modeled as scale free, small-world networks [3–5, 8, 9]. In contrast, a vast majority of spam are automatically sent, typically from botnets, and it is expected that they show unsocial behavior. We have shown that the structural and temporal properties of spam networks can reveal their anomalous nature. Although spam networks show some properties that are similar to ham (i.e., small-world network properties), they can still be distinguished from ham networks as they have significantly smaller average clustering coefficient and larger average path length, regardless of the size of the networks. Overall, we have shown that although the behavior of spam might change over time, its unsocial behavior is not hidden in the mixture of email traffic, even when the amount of spam is less than ham (dataset B).

The datasets used in this study to analyze the characteristics of spam do not contain the email communications that do not pass the measurement location. Due to asymmetric routing and load-balancing policies deployed by the network



(a) Average path length (A)

(b) Average clustering coefficient (A)

(c) Relative GCC size (A)

(d) Number of CCs (A)

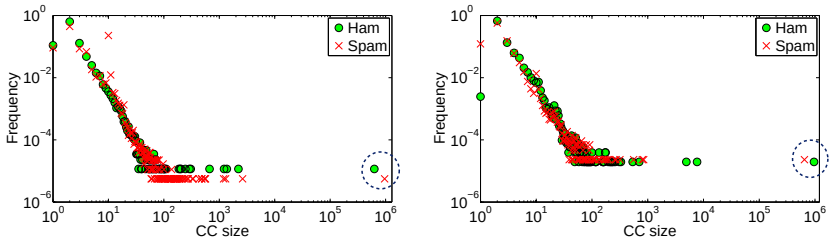
(e) Average path length (B)

(f) Average clustering coefficient (B)

(g) Relative GCC size (B)

(h) Number of CCs (B)

Figure 2.3: *Both networks are small-world networks (a,b,e,f), however, ham has a higher average clustering coefficient. The ham networks become more connected over time (c,g), and the number of CCs increases faster for the spam networks (d,h).*



(a) CC distribution (A) (b) CC distribution (B)

Figure 2.4: *The distribution of size of CCs. The GCCs of the networks are orders of magnitude larger than other CCs.*

Table 2.3: *Structural properties of the ham and the spam networks.*

Dataset	Network	Nodes	Edges	C ($\times 10^{-3}$)	$\langle l \rangle$	relative GCC size	No. CCs	γ deg. dist.
A	Ham	859,623	1,060,380	9.92	7.0	72.90%	85,992	2.7
	Spam	1,795,197	2,506,298	1.59	8.5	53.53%	178,754	-
B	Ham	1,077,042	1,593,042	9.80	6.7	84.24%	50,742	2.3
	Spam	578,158	1,044,714	1.54	7.8	79.21%	40,236	-

routers, not all traffic travels the link, and less traffic is seen in the outgoing than the incoming direction of the link (Table 2.2). However, our goal is to perform a comparative analysis of the distinguishing behavior of spam and ham traffic that are observed over the link. Therefore, it is not required to generate a complete email network of all exchanged emails to be able to study the differences in the social network properties of legitimate and spam traffic.

In addition, the "missing past" problem, which is not limited to our dataset, exists since it is not possible to gather data reaching all the way back to a network's birth. Leskovec et al. [8] showed that the effect of missing past is minor as we move away from the beginning of the data observation. We investigated the effect of missing past by constructing an email network which lacked the first week of data from dataset A and comparing it with the network containing both weeks. We have observed that the structural properties of the email networks were relatively similar for both of the networks particularly for the legitimate email.

Earlier studies [3–7, 9] have also used incomplete email networks to study the structure of email networks or to deploy a social network-based approach to mitigate spam. Even though our measurement duration was shorter than previous studies [3, 4, 8, 9], we have generated the largest and most general datasets used for this type of analysis. The 14 days of data collection might not be large enough to study the evolution of email networks, but our analysis of the temporal variation in the structure of email networks provides us with evidence on how their structure might change with longer periods of measurements.

Overall, this work has provided us with very large datasets of real traffic traversing a high speed Internet backbone link. These datasets allow us to model the behavior of email traffic as observed from the vantage point of a network device on the link and reveal the differences in the network-level behavior of ham and spam traffic.

2.5 Anomalies in Email Network Structure

The structural properties of real networks that deviate from the expected properties for social networks, suggest anomalous behavior in the network [18]. In this section, we show that the anomalies caused by the unsocial behavior of spam can be detected in the email networks by using an outlier detection mechanism.

We have shown in Section 2.4 that the ham networks exhibit power law out-degree distributions with $\gamma_{A_{out}=2.3}$ and $\gamma_{B_{out}=2.1}$ for dataset A and B respectively. The outliers in the out-degree distribution of the email networks are of particular importance, as we are interested in finding the nodes that send spam.

Procedure 2.1 presents the process of detecting outliers from the out-degree distribution. First the ratio of the out-degree distribution of the email network, containing both ham and spam, and our model is calculated. Then the Median Absolute Deviation (MAD) method is deployed to calculate the median of the absolute differences of the obtained ratios from their median. The nodes in the network that have an out-degree that deviates a lot (based on a threshold value) from the median are marked as outliers.

Table 2.5 shows the percentage of spam that were sent in different networks and the percentage of spam sent by the identified outlier nodes. The nodes in the email networks generated from dataset A (B) have sent in average around 70% (40%) spam and the identified outlier nodes have sent just slightly more spam than the average node. The reason is that the outlier detection method tends to mark both nodes that have sent only one email and those that have sent a large number of email as outliers. However, we have observed that the nodes which have sent only one email had sent ham and spam with the same probability, and the nodes with high out-degree have mostly sent legitimate email (e.g., mailing lists). By excluding the nodes that have a high out-degree (100 in our experiments) from the outliers as well as the nodes that have only sent one email during the collection period, we can see that more than 95% (81%) of the email sent by the identified outliers in dataset A (B) have actually been spam. Moreover, these nodes have actually sent around 25% (35%) of the total spam in the network.

The outliers in the out-degree distribution of the complete email network which in addition to ham and spam contains rejected email can be identified similarly. As an example, the nodes in the complete email network generated from one day of email traffic in dataset A have sent in average 94.8% spam and rejected email. The emails sent by the outlier nodes detected by our method have been 99.3% spam or rejected.

Procedure 2.1: Finding out-degree distribution outliers

OUTLIERS_DETECTION
$G\_odd \leftarrow$ out-degree distribution for graph G
$M\_odd \leftarrow Ck^{-\gamma}$ (the power law distribution model)
$r \leftarrow$ the ratio between G_odd and M_odd
$m \leftarrow MAD(r)$
for all nodes $v \in G$ **do**
  if $r(k_v) > m \times threshold$ **then**
    add v to the list of *outliers*
  end if
end for

Table 2.4: *Percentage of total spam, spam sent by all the identified outlier nodes, and those with degree between one and 100, in email networks containing both ham and spam.*

Dataset	Network	Total spam	Spam sent by outliers	Spam sent by outliers with $1 < k < 100$
A	1 day	68%	69.9%	95.5%
	7 days	70%	74.0%	96.8%
	14 days	70%	74.8%	96.9%
B	1 day	40%	43.6%	82.7%
	7 days	35%	42.8 %	81.3%
	14 days	39 %	46.7%	87.3%

2.6 Conclusions

In this study we have investigated the social network properties of email networks to study the characteristics of legitimate and unsolicited emails. The email networks were generated from real email traffic which was captured on an Internet backbone link. We have analyzed the structural and temporal properties of the email networks and have shown that legitimate email traffic generates a small-world, scale-free network that can be modeled similar to many other social networks. Moreover, the unsocial behavior of spam, which might change over time, is not hidden in the mixture of email traffic. Therefore, email networks that contain spam do not exhibit all properties commonly present in social networks.

Moreover, we have shown that by identifying the anomalies in the structural properties of email networks, it is possible to reveal a number of abusive nodes in the network. More specifically, we have shown that the outliers in the out-degree distribution of email networks to a large extent represent the spamming nodes in the network. Therefore, the social network properties of email networks can potentially be used to detect malicious hosts on the network.

Acknowledgments

This work was supported by .SE – The Internet Infrastructure Foundation and SUNET. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/ 2007-2013) under grant agreement no. 257007.

Bibliography

[1] Anirudh Ramachandran and Nick Feamster, "Understanding the network-level behavior of spammers," in *ACM SIGCOMM Computer Communication Review*. Aug. 2006, vol. 36, pp. 291–302, ACM.

[2] Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M Voelker, Vern Paxson, and Stefan Savage, "Spamalytics: An Empirical Analysis of Spam Marketing Conversion," in *Proceedings of the 15th ACM conference on Computer and communications security - CCS '08*, New York, New York, USA, 2008, p. 3, ACM Press.

[3] Holger Ebel, Lutz-Ingo Mielsch, and Stefan Bornholdt, "Scale-free topology of e-mail networks," *Physical Review E*, vol. 66, no. 3, pp. 1–4, Sept. 2002.

[4] Luiz H Gomes, Rodrigo B Almeida, and Luis M A Bettencourt, "Comparative Graph Theoretical Characterization of Networks of Spam and Legitimate Email," in *Conference on Email and Anti-Spam (CEAS)*, 2005.

[5] P.O. Boykin and V.P. Roychowdhury, "Leveraging social networks to fight spam," *Computer*, vol. 38, no. 4, pp. 61–68, Apr. 2005.

[6] Ho-yu Lam and Dit-yan Yeung, "A Learning Approach to Spam Detection based on Social Networks," in *Conference on Email and Anti-Spam (CEAS)*, 2007.

[7] C. Tseng and M. Chen, "Incremental SVM model for spam detection on dynamic email social networks," in *Conf. on Computational Science and Engineering*, 2009.

[8] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos, "Graph Evolution: Densification and Shrinking Diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 2–es, Mar. 2007.

[9] Gueorgi Kossinets and Duncan J Watts, "Empirical Analysis of an Evolving Social Network," *Science (New York, N.Y.)*, vol. 311, no. 5757, pp. 88–90, Jan. 2006.

[10] Andrei Broder, Ravi Kumar, Farzin Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener, "Graph Structure in the Web," *Computer networks*, vol. 33, no. 1, pp. 309–320, 2000.

[11] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On Power-Law Relationships of the Internet Topology," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 251–262, 1999.

[12] A.A. Nanavati, Rahul Singh, Dipanjan Chakraborty, K. Dasgupta, Sougata Mukherjea, Gautam Das, Siva Gurumurthy, and Anupam Joshi, "Analyzing the Structure and Evolution of Massive Telecom Graphs," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 5, pp. 703–718, 2008.

[13] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 2007, p. 29, ACM Press.

[14] Farnaz Moradi, Magnus Almgren, Wolfgang John, Tomas Olovsson, and Philippas Tsigas, "On collection of large-scale multi-purpose datasets on internet backbone links," in *Proc. of Building Analysis Datasets and Gathering Experience Returns for Security Workshop*, 2011.

[15] D J Watts and S H Strogatz, "Collective Dynamics of 'Small-World' Networks.," *Nature*, vol. 393, no. 6684, pp. 440–2, June 1998.

[16] Reka Zsuzsanna Albert, *Statistical Mechanics of Complex Networks*, Ph.D. thesis, University of Notre Dame, 2001.

[17] A Clauset, CR Shalizi, R. Cosma, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Reviews*, vol. 51, no. 4, pp. 661–703, 2009.

[18] Leman Akoglu and M McGlohon, "Oddball: Spotting Anomalies in Weighted Graphs," in *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, 2010, pp. 410–421.

PAPER II

Farnaz Moradi, Tomas Olovsson, Philippas Tsigas

An Evaluation of Community Detection Algorithms on Large-Scale Email Traffic

Proceedings of the 11th International Conference on Experimental Algorithms (SEA'12), Lecture Notes in Computer Science Vol.: 7276, pp. 283 - 294, Springer-Verlag, Bordeaux, France, June, 2012.

In order to comply with the thesis layout, some small non-technical changes has been made and the appendix of the paper has been added to the main content.

3 An Evaluation of Community Detection Algorithms on Large-Scale Email Traffic

Community detection algorithms are widely used to study the structural properties of real-world networks. In this paper, we experimentally evaluate the qualitative performance of several community detection algorithms using large-scale email networks. The email networks were generated from real email traffic and contain both legitimate email (ham) and unsolicited email (spam). We compare the quality of the algorithms with respect to a number of structural quality functions and a logical quality measure which assesses the ability of the algorithms to separate ham and spam emails by clustering them into distinct communities. Our study reveals that the algorithms that perform well with respect to structural quality, don't achieve high logical quality. We also show that the algorithms with similar structural quality also have similar logical quality regardless of their approach to clustering. Finally, we reveal that the algorithm that performs link community detection is more suitable for clustering email networks than the node-based approaches, and it creates more distinct communities of ham and spam edges.

3.1 Introduction

Unfolding the communities in real networks is widely used to determine the structural properties of these networks. Community detection or clustering algorithms aim at finding groups of related nodes that are densely interconnected and have fewer connections with the rest of the network. These groups of nodes are called communities or clusters and they exist in a variety of different networks [1].

The problem of how to find communities in networks has been extensively studied and a substantial amount of work has been done on developing clustering algorithms (an overview can be found in [2, 3]). However, there is no consensus on which algorithm is more suitable for which type of network. Therefore, a number of studies have experimentally compared the qualitative performance of different community detection algorithms on synthetic and benchmark graphs with built-in community structure [4, 5]. However, these graphs are different from real-world

networks as the assumptions they make are not completely realistic [2]. Delling et al. [6] have shown that the implicit dependencies between community detection algorithms, synthetic graph generators, and quality functions used for assessing the qualitative performance of the algorithms make meaningful benchmarking very difficult. Therefore, empirical studies of the existing algorithms on real-world networks are crucial in order to evaluate different algorithms and to find the most suitable methods for different types of networks.

Moreover, community detection in real-networks has many different applications. Community detection algorithms can be used to find users with similar interests in a social network in order to provide recommendations to them, to group the peers that are geographically close in a peer-to-peer system to improve the performance of the system, or to detect the communities generated by malicious users in order to mitigate Sybil attacks [7]. In this paper, we study the community structure of a number of large *email networks* containing both legitimate *ham* and unsolicited *spam* emails. In an email network, the nodes represent email addresses and the edges represent email communications. In addition to a qualitative comparison of the algorithms, our goal is to find the best community detection algorithm that can separate spam and ham emails by clustering them into distinct communities. Such an algorithm can potentially be deployed in spam detection mechanisms that aim at mitigating the spam problem by looking at email traffic rather than email contents.

In order to achieve our goals, we have selected a number of broadly used community detection algorithms that are known to perform well on synthetic, benchmark, and a limited number of real graphs. In this study we evaluate and compare the qualitative performance of these algorithms when they are applied to large-scale email networks. Since the true community structure of our networks is unknown, it is important to use a quality measure to compare the algorithms. It is known that there is no single perfect quality metric for the comparison of the communities detected by different algorithms [8], therefore we use a number of *structural quality* functions such as modularity [9], coverage, and conductance [10], as well as a *logical quality* measure which is determined based on how homogeneous the edges inside the communities are. We use this measure to investigate and compare the ability of the selected algorithms in separating ham and spam emails into distinct communities.

The contributions of the paper are as follows. We show that there is a trade-off between creating high structural and high logical quality communities. Therefore, hierarchical and multiresolution algorithms which allow us to select the granularity of the clustering are more suitable to create the communities with the desired quality. We reveal that different algorithms that create communities with similar size distribution achieve similar structural and logical qualities, even though they use different approaches for clustering. Finally, we show that an algorithm that clusters networks based on the similarity of edges is superior to the algorithms that perform node-based clustering.

The rest of this paper is organized as follows. Section 3.2 presents the quality functions which are used for evaluating and comparing the algorithms. The community detection algorithms being compared are presented in Section 3.3. Section 3.4 reviews related previous research. In Section 3.5, the dataset used for empirical comparison is presented and the experimental results are discussed. Finally Section 3.6 concludes the work.

3.2 Quality of Community Detection Algorithms

In this section, we present the notations and the quality functions that are used in the rest of the paper.

Preliminaries

Let $G = (V, E)$ represent a connected, undirected, and unweighted graph where V is the set of n nodes and E is the set of m edges of G. A *clustering* $\mathcal{C} = \{C_1, \ldots, C_k\}$ is a partitioning of V into k clusters C_i, by a node-based community detection algorithm. A cluster containing only a single node is called a *singleton*, and a cluster with only one internal edge is called *trivial*. If nodes can be shared between clusters, the clustering is called *overlapping*. The number of intra- and inter-cluster edges of a cluster C are denoted by $m(C)$ and $\overline{m}(C)$, respectively and $m(\mathcal{C}) := \sum_{C \in \mathcal{C}} m(C)$ is the total number of intra-cluster edges in $\mathcal{C}$.

Quality Functions

A quality function is used either as an objective function to be optimized in order to find the communities of a network, or as a measure for assessing the quality of a clustering [6]. When the true community structure of a network is not known, quality functions are necessary for evaluating the qualitative performance of clustering algorithms. Since no single best quality function exists [8], we investigate three widely used structural quality functions: coverage, modularity [9], and conductance [10].

Coverage. Coverage of a clustering,

$$cov(\mathcal{C}) := \frac{m(\mathcal{C})}{m},$$

is the most simple quality function, however, it is biased towards coarse-grained clusterings.

Modularity. Modularity of a clustering is defined as

$$Q(\mathcal{C}) := \frac{m(\mathcal{C})}{m} - \frac{1}{4m^2} \sum_{C \in \mathcal{C}} \left(\sum_{v \in C} deg(v) \right)^2,$$

and is based on the idea that a good cluster should have higher internal and lower external density of edges compared to a *null model* with similar structural properties but without a community structure [9].

Conductance. Conductance of a cut $(C, V \setminus C)$ in a graph is defined as

$$\phi(C) := \frac{\overline{m}(C)}{\min(\sum_{v \in C} deg(v), \sum_{v \in V \setminus C} deg(v))},$$

and tends to favor clusterings with fewer number of clusters [8]. Inter-cluster conductance, $\delta(\mathcal{C}) := 1 - \max_i \phi(C_i)$, $i \in \{1, \ldots, k\}$, is usually used as a worst-case measure to assess the quality of a clustering. The average conductance $(\frac{1}{|C|} \sum_{C \in \mathcal{C}} \phi(C))$ is also a useful metric, since if an algorithm creates singletons, the inter-cluster conductance value will be dominated by the zero value for these clusters, while the average would not [11].

The above widely used structural quality functions cannot be directly calculated for assessing the quality of link community detection methods because of the community overlaps. For instance, modularity of a link community can be calculated by applying a modified modularity function on a projected and weighted transformation of the network [12]. In this paper we investigate the structural quality of link communities by using two of the quality measures introduced in [13]. *Community coverage* measures the fraction of the nodes that belong to at least one non-trivial community, and *Overlap coverage* measures the average number of times a node is clustered inside non-trivial communities. Higher values for overlap coverage mean that the algorithm has extracted more information from the network. The algorithms that don't find overlapping communities yield the same value for both overlap and community coverage.

In addition to the structural quality, we determine the *logical quality* of a clustering based on the type of the edges inside its communities. A clustering which yields only homogeneous communities, in which all of the edges are of the same type, has a perfect logical quality. For instance, a clustering with communities that contain only spam emails or only ham emails has higher logical quality compared to a clustering which yields communities containing a mixture of both ham and spam. In addition, the amount of spam and ham emails that can be separated into distinct homogeneous communities by an algorithm is used to determine its logical quality.

3.3 Studied Community Detection Algorithms

In this section we briefly describe the community detection algorithms we have selected and compared using our email networks.

Fast modularity optimization (Blondel) by Blondel et al. [14]. This algorithm, also known as *Louvain* method, is a greedy approach to modularity maximization. The algorithm starts with assigning each node to a singleton and progresses

by moving nodes to neighboring clusters in order to improve modularity. This method has complexity $O(m)$ and unfolds a hierarchical community structure with increasing coarseness and meaningful intermediate communities.

Maps of random walks (Infomap) by Rosvall and Bergstrom [15]. This algorithm is a flow-based and information theoretic clustering approach with complexity $O(m)$. It uses a random walk as a proxy for information flow on a network and minimizes a *map equation*, which measures the description length of a random walker, over all the network clusters to reveal its community structure. Infomap aims at finding a clustering which generates the most compressed description length of the random walks on the network.

Multilevel compression of random walks (InfoH) by Rosvall and Bergstrom [16]. This method generalizes the Infomap method to reveal multiple levels of a network. InfoH minimizes a *hierarchical map equation* to find the shortest multilevel description length of a random walker.

Potts model community detection (RN) by Ronhovde and Nussinov [17]. This algorithm is based on minimization of the Hamiltonian of a local objective function, the absolute Potts model. The multiresolution variant of the algorithm deploys information theory-based measures to find the best communities on all scales. The complexity of this method is superlinear $O(m^{1.3})$ for the community detection algorithm and $O(m^{1.3} \log n)$ for the multiresolution algorithm.

Markov clustering (MCL) by Dongen [18]. MCL is based on the idea that a random walk entering a dense cluster likely remains for a long time inside the cluster before switching between sparsely connected communities. The random walks are calculated deterministically and simultaneously using a matrix of transition probabilities. The MCL algorithm has a complexity of $O(nk^2)$, where k refers to the average or maximum number of allowed neighbors for the nodes.

Link community detection (LC) by Ahn et al. [13]. All of the above algorithms aim at clustering nodes into densely connected communities. However, Ahn et al. [13] have defined communities as a group of topologically similar edges and have introduced a link community detection algorithm for revealing them. Their algorithm has complexity $O(nk_{max}^2)$, where k_{max} is the maximum degree, and unfolds the hierarchical structure and overlapping communities of a network. Although the clustering is meaningful at all scales, an objective function, the *partition density*, is used to select the optimum level of hierarchy.

All of the above algorithms are known to perform well on large networks. Infomap, InfoH, and MCL are suitable for clustering networks where edges represent flows. Emails can be seen as flows of data between people, so flow-based approaches are good candidates for clustering email networks. Email communications can also be seen as pairwise relationships between people, so the other topological methods could also be suitable. LC which is based on calculating the similarity of the edges in a network can also be suitable since we are interested in grouping the same type of edges into the same clusters.

In this study, we have used the implementations of the algorithms, which were publicly available, in order to conduct the experiments. Blondel creates a hierarchy of clusterings where the best modularity is achieved at its last level. We have also looked at the clustering yield at Blondel's first level of hierarchy, which has smaller meaningful communities, and refer to it as *Blondel L1*. We have also used the basic RN algorithm instead of its multiresolution variant to be able to choose the desired clustering granularity. The granularity of the clusterings should be considered when comparing the quality of the algorithms since structural quality functions are usually in favor of coarse-grained clusterings [8].

3.4 Related Work

Experimental comparisons of different community detection algorithms have been conducted on both real and benchmark graphs. Lancichinetti and Fortunato [4] compared different algorithms including Blondel, Infomap, RN, and MCL, on GN and LFR benchmark graphs. They showed that Infomap, Blondel, and RN perform well, but MCL performs worse especially for large communities. They also showed that the performance of Blondel decreases for large graphs, whereas Infomap remains stable. Brandes et al. [11] conducted an experimental evaluation of three clustering methods including MCL using random clustered graphs and showed that MCL performs well with respect to some quality functions but produces more clusters than contained in the network.

Community detection algorithms have also been evaluated and compared using different real networks. Tibély et al. [19] have analyzed the community structure of a large mobile phone call graph using Blondel L1, Infomap, and an overlapping method. Leskovec et al. [20] studied a number of real networks, including the Enron email network and an email network of a large organization, to empirically compare two different clustering methods. The latter dataset was also used by Lancichinetti et al. [21], in addition to other real networks, to study the characteristics of communities in different types of complex networks. They used Infomap together with another algorithm to show that although different methods output different clusterings, the statistical properties of their communities are quite similar for similar classes of networks. Studies of the community structure of email networks have also been conducted by Guimerà et al. [22] using emails in a university.

In contrast to previous studies, the dataset used in this study is based on email traffic captured on a high speed Internet backbone link, and is not limited to a single organization. To the best of our knowledge, this is the first study of the community structure of large-scale email networks containing spam. This dataset enables us to evaluate the ability of the community detection algorithms in separating spam from ham by clustering them into distinct clusters.

3.5 Experimental Evaluation

In this section, the email dataset and the experimental results are presented.

3.5.1 Dataset

The dataset used for creating the email networks was generated by collecting SMTP packets on a 10 Gbps link of the core-backbone of SUNET[1] during a period of 14 consecutive days in March 2010. During the collection period more than 797 million SMTP packets were collected, which were sent and received by 614,601 distinct domains. Around 3.4 million emails were extracted from the collected packets after removing unusable and rejected email transmissions. These emails were then classified to be either *spam* or *ham* using a well-trained filtering tool [2]. Following that, email contents were discarded and email addresses were anonymized in order to preserve privacy in a way that no information about the senders, receivers, and content of the emails are retrievable.

In addition to a complete email network, we generated daily and weekly email networks. An email network consists of email addresses as nodes, and the email communications between them as edges. More details on the measurement location, data collection and pre-processing, and the structural and temporal properties of the email networks can be found in [23] and [24], respectively.

3.5.2 Comparison of the Algorithms

In this section, the experimental results regarding the qualitative performance of the clustering algorithms with respect to their structural and logical quality is presented. A summary of the results can be found at the end of the section.

Table 3.1 shows the total number of nodes and edges, and the number of spam edges in each studied email network, as well as the number of communities created by each clustering algorithm. The algorithms were applied to the giant connected component (GCC) of each email network, which is a subset of the nodes in the network where a path exists between any pair of them. The networks are also considered as unweighted and undirected. The distribution of the community sizes for one daily, one weekly, and the complete email network created by the different community detection algorithms is shown in Figure 3.1. Since LC creates communities with overlapping nodes, we have also plotted the distribution of the number of communities per node in Figure 3.1(h). It can be seen that the shapes of the distributions do not change much as the size of the networks grow.

Blondel creates a coarse-grained clustering and in average achieves 46% modularity gain over Blondel L1. InfoH also creates coarse clusters and in average gains

[1]The Swedish University Network (`http://www.sunet.se/`) serves as a backbone for university traffic, student dormitories, research institutes, etc.

[2]The SpamAssassin (`http://spamassassin.apache.org`) was in use for a long time in our University mail server and it incurs high detection and low false positive rates.

Table 3.1: *The properties of the GCC of the generated email networks (larger networks become more connected) and the number of communities created by each algorithm.*

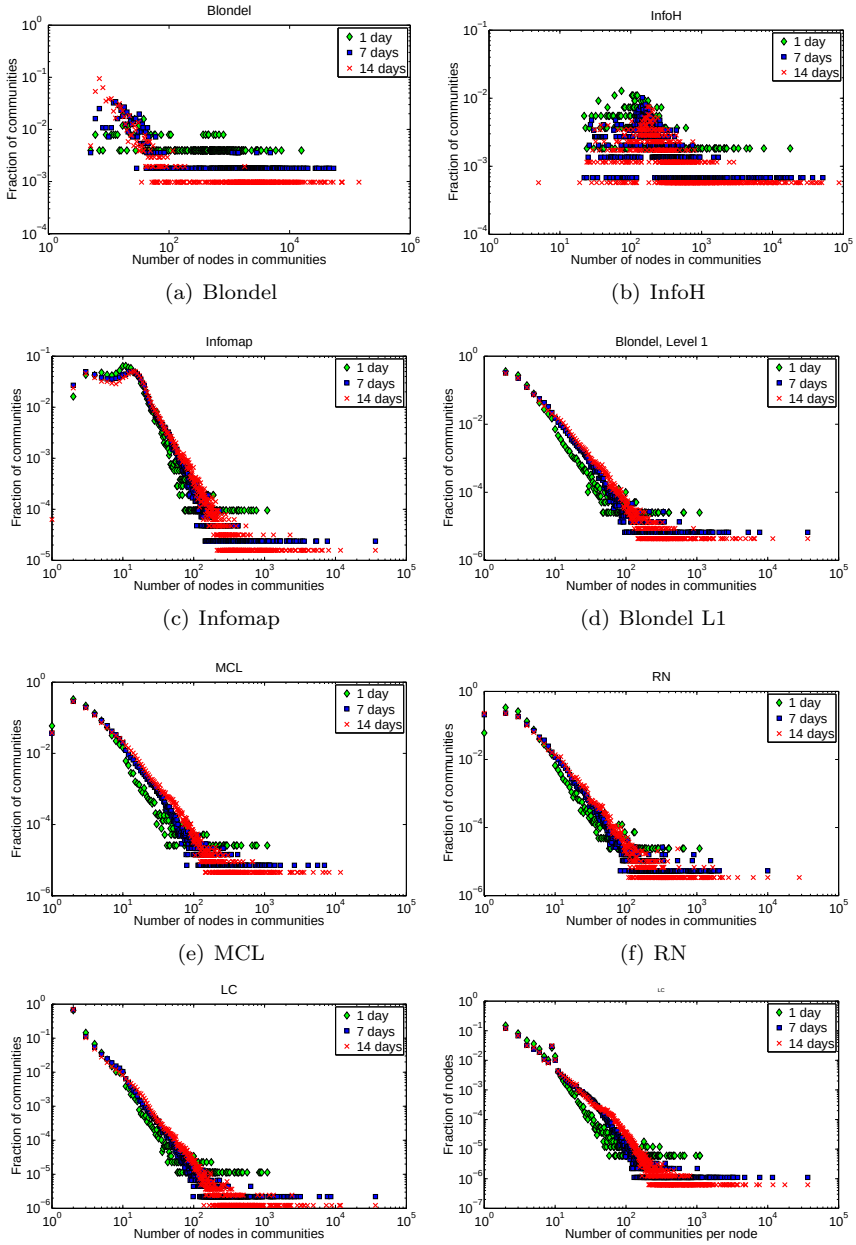
	Nodes	Edges	Spam	Bl.	InfoH	Infom.	Bl. L1	MCL	RN	LC
Day 1	167,329	236,673	173,640	253	546	10,505	39,477	38,775	41,215	88,028
Day 2	153,734	194,797	97,260	194	397	8,025	28,077	27,011	28,499	61,027
Day 3	123,878	168,896	108,996	218	412	8,151	29,150	28,031	30,022	64,310
Day 4	128,200	172,836	113,299	218	398	8,484	29,123	28,043	30,167	63,165
Day 5	101,643	135,195	89,119	195	311	6,664	22,212	21,593	23,935	46,928
Day 6	72,068	99,361	75,713	236	183	4,714	13,904	13,716	17,697	30,236
Day 7	73,131	103,293	85,879	199	271	4,842	17,305	16,808	18,631	37,581
Week 1	901,699	1,441,731	961,809	558	1,470	41,916	149,131	144,054	187,960	451,275
Day 8	115,232	155,919	90,299	234	379	7,745	27,661	26,514	28,409	57,931
Day 9	112,713	152,569	88,273	188	383	7,521	26,395	25,549	26,942	56,443
Day 10	140,843	195,999	121,158	255	441	8,664	31,033	30,231	39,020	67,741
Day 11	125,029	179,410	116,056	192	398	8,171	28,501	27,897	30,484	65,285
Day 12	106,816	149,407	100,595	211	380	7,319	25,314	24,328	28,040	54,317
Day 13	73,325	98,713	71,954	339	296	5,275	16,736	16,074	22,476	32,403
Day 14	68,315	100,089	76,408	179	210	4,741	14,567	14,254	17,822	31,463
Week 2	810,543	1,348,373	859,324	436	380	40,553	143,569	139,366	156,822	430,232
All	1,599,732	2,790,322	1,858,686	1,028	1,740	63,471	230,013	220,346	294,581	817,074

more than 15% in the compression of the description length of the random walks on the networks over the non-hierarchical version (Infomap). MCL allows us to select the granularity of the clustering by choosing an inflation parameter. It is also possible to choose the resolution parameter for RN to achieve a clustering with the desired granularity. We have selected the inflation parameter in MCL and the resolution parameter in RN so that for most of the networks they yield clusterings with a close granularity to that of Blondel L1. This allows us to further investigate and compare the effect of the granularity of the clusterings on their quality. LC is different in nature from the other algorithms as it is based on link community detection rather than a node-based approach. LC yields the finest-grained clustering for all of the networks at its best level of hierarchy.

Figure 3.2 summarizes the distribution of the size of the communities created by the different algorithms for the "week 2" email network. The distributions for other daily and weekly networks are quite similar. It can be seen that Blondel and InfoH, which create very coarse-grained clusters, have very different community size distributions compared to each other and the rest of the algorithms. It can also be seen in Figure 3.2(b) that, surprisingly, Blondel L1, MCL, and RN follow similar distributions. The main difference is that MCL and RN create a number of singletons, but Blondel L1 does not. The community size distribution of LC is also close to the other three methods, but it creates more clusters.

Structural Quality

Figure 3.3 shows a comparison of the structural quality of the different clusterings. Each bar corresponds to a daily network (day 1 to day 14), except the last three bars from the left for each of the algorithms, which correspond to week 1, week 2, and complete email networks, respectively. It can be seen that Blondel, which aims at maximizing modularity, have the highest structural quality with respect to all of the quality functions. Although InfoH uses a fundamentally different approach



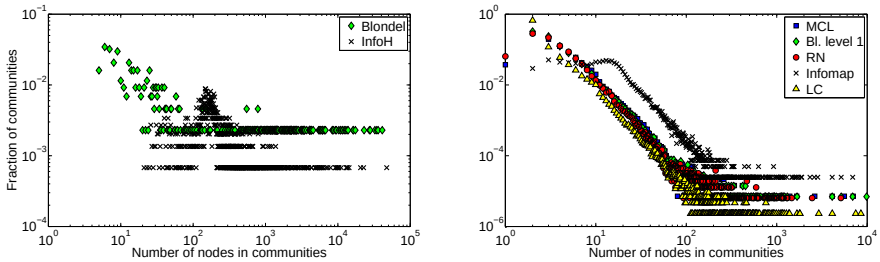
(a) Blondel (b) InfoH

(c) Infomap (d) Blondel L1

(e) MCL (f) RN

(g) LC (nodes per community) (h) LC (communities per node)

Figure 3.1: *Comparison of community size distribution for email networks.*



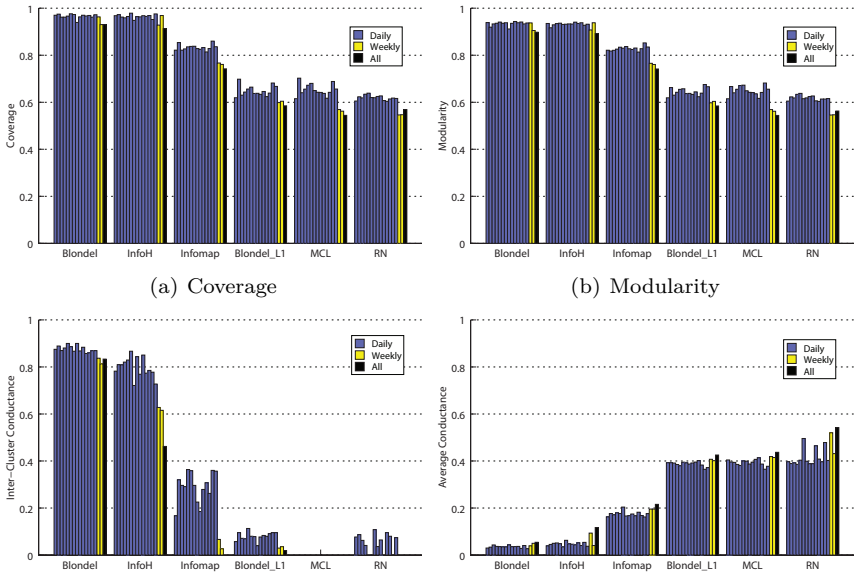
(a) (b)

Figure 3.2: *A comparison of community size distribution using "Week 2" email network. Blondel L1, MCL, and RN follow very similar distributions.*

it achieves equally good structural quality, however its quality degrades for larger networks. Blondel L1, MCL, and RN, which have closer granularities, also show similar quality with respect to coverage, modularity, and average conductance. However, based on the inter-cluster conductance, MCL and RN do not perform well since they might create a number of singletons which results in an inter-cluster conductance of zero.

Our experimental results reveal that the structural quality of clusterings are roughly consistent for different daily networks. The clusterings with similar granularity and community size distribution also show similar structural quality, therefore, it is important to take the granularity of the clusterings into account when comparing the algorithms. LC creates a clustering with the finest granularity, however the studied structural quality functions cannot be directly used for assessing the quality of this algorithm due to its different nature. In this paper, we look at community coverage and overlap coverage which were introduced for assessing the quality of link-based clustering by Ahn et al. [13].

LC, Blondel, and InfoH yield full community coverage for all of the email networks. Infomap, Blondel Ll, MCL, and RN achieve community coverage of around 0.99, 0.84, 0.83, and 0.8, respectively. However, this function on its own is not enough for assessing the quality of a clustering method, it is also important to consider the overlap coverage of the clusterings. None of the algorithms, except MCL and LC, find overlapping clusters so their overlap coverage is equal to their community coverage. MCL is not an overlapping clustering method, but for some specific graphs it might find overlaps [18]. In our email networks, MCL yields very few overlapping communities so its overlap coverage is just slightly higher than its community coverage. LC yields overlap coverage of 2.6, 3.1, and 3.4 in average for the daily, weekly, and complete email networks, meaning that it unfolds more overlaps in larger networks.



(a) Coverage

(b) Modularity

(c) Inter-cluster conductance

(d) Average conductance

Figure 3.3: *Comparison of structural quality of the algorithms on daily, weekly, and complete email networks. Blondel and InfoH yield the best structural quality.*

Logical Quality

Our experiments show that all algorithms create a number of *spam communities* that only contain spam, *ham communities* that only contain ham, and *mix communities* with a mixture of both ham and spam edges. Figure 3.4 shows a comparison between the percentage of spam, ham, and mix communities created by the different algorithms. The last three bars from the left for each of the algorithms correspond to week 1, week 2, and the complete email networks, respectively. It can be seen that InfoH and Blondel perform worse, since these algorithms tend to merge smaller homogeneous communities into mix communities to achieve higher structural quality. The best results for all networks are achieved by LC.

Moreover, it is important to assess the amount of spam and ham emails that can be separated by community detection algorithms, in order to investigate the possibility of deploying clustering approaches to perform spam detection. Figure 3.5 shows the ratio of total spam and ham edges that are inside homogeneous spam and ham communities. In all of the networks, communities created by LC contain the highest number of spam and ham edges. Blondel and InfoH have the worst logical quality and Blondel L1, MCL, and RN have almost similar quality. For all algorithms, except LC, some of the spam and ham emails end up as inter-cluster edges and can therefore not be separated by the clustering algorithms. It can also

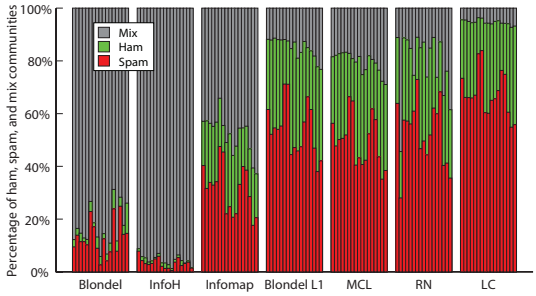
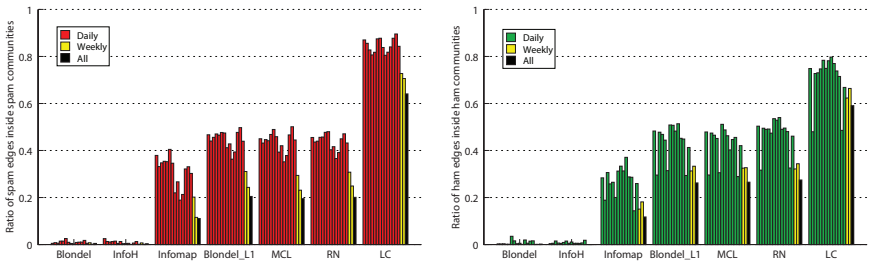


Figure 3.4: *Comparison of percentage of spam, ham, and mix communities created by different algorithms. LC creates the highest number of homogeneous communities.*



(a) Spam communities (b) Ham communities

Figure 3.5: *Ratio of spam (ham) in homogeneous spam (ham) communities. LC clusters have a higher ratio of total spam (ham) edges inside the spam (ham) communities.*

be seen that the percentage of spam (ham) edges which are clustered inside spam (ham) communities decreases for larger networks.

Our experiments suggest that the logical quality tends to be higher for fine-grained clusterings. The granularity of the best clustering created by LC is finer than the other clusterings in our experiments. LC cuts its hierarchy of clustering at an optimum threshold which results in maximal partition density. By choosing a threshold below the optimum value, we can have a clustering with coarser granularity. Since the algorithm reveals meaningful communities at all scales, we changed the threshold so that the granularity of the clustering became more similar to that of Blondel L1, MCL, and RN. Our experiments with the new clusterings show that, the percentage of spam (ham) edges inside the spam (ham) communities was reduced. For instance, for the first daily network the percentage of spam (ham) edges decreased from 87% to 66% (from 76% to 56%). Although the logical

quality degrades by changing the coarseness of the clustering, LC still shows higher logical quality than all of the other algorithms.

Furthermore, we have looked at the communities found in ham and spam networks and compared them with the communities identified in the mix email networks. Figure 3.6 shows the community size distribution for communities of ham and spam networks generated from "week 1" email data. The *ham network* only contains ham edges and the *spam network* only spam emails. It can be seen that these networks, similar to the complete email networks, exhibit community structure. The spam network has $n = 531,856$ nodes and $m = 928,329$ edges which is larger than the ham network with $n = 349,814$ nodes and $m = 457,912$ edges, therefore all of the algorithms find more communities in the spam network. Although the type and the nature of the edges in these networks are different, each community detection algorithm creates clusterings with roughly similar community size distribution shapes for both the ham and the spam network.

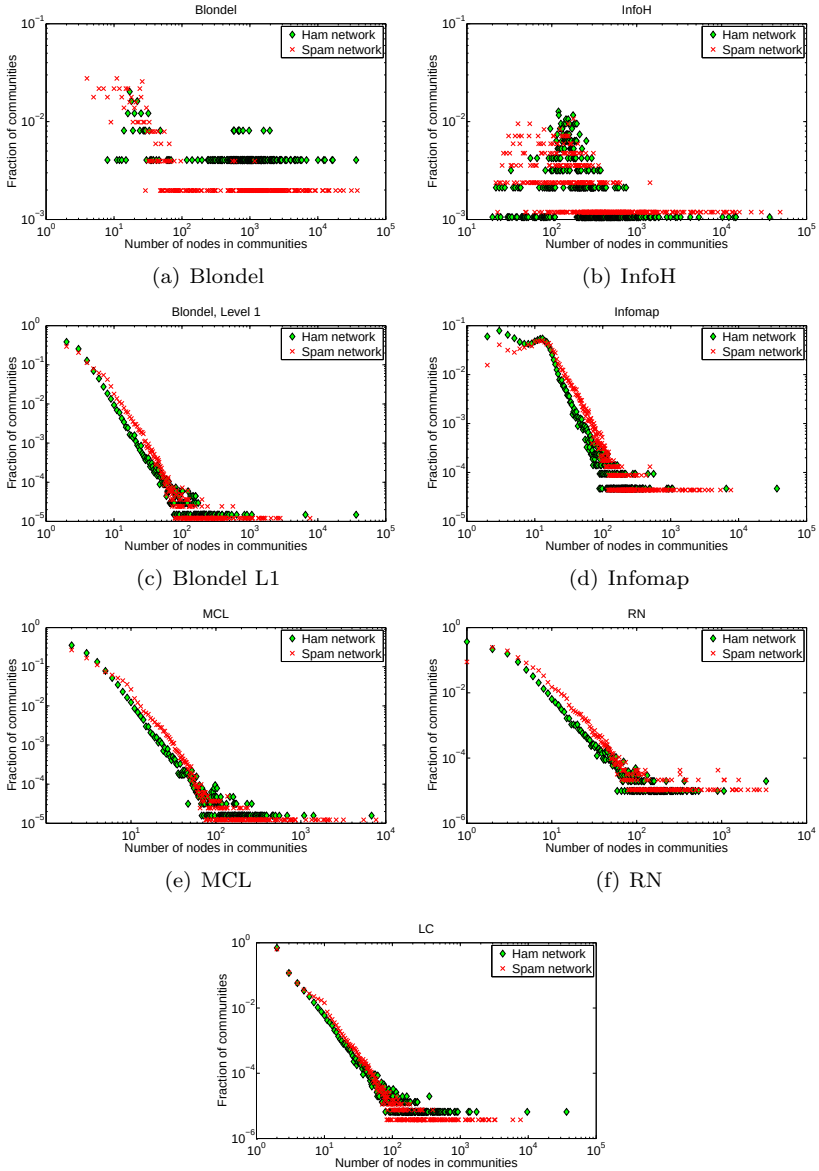
Moreover, Figure 3.7 shows the community size distribution of the distinct homogeneous ham and spam communities which were created from the "week 1" mix email network. Blondel L1, MCL, RN, and LC create spam and ham communities that follow a size distribution which is more or less similar to the distribution of the communities created in the distinct ham and spam networks. This suggests that even though the mix email network contains a mixture of both type of edges, these algorithms are still able to extract most of the communities that they also found in the corresponding ham and spam networks. However, it seems that Blondel, InfoH, and Infomap perform worse in this respect as they also find more mix communities than the other algorithms (see Figure 3.4).

Summary of the Experimental Results

- Blondel and InfoH create coarse-grained clusters and achieve the best quality with respect to all of the structural quality functions. However, they have the worst logical quality with respect to both number of homogeneous communities and amount of spam and ham emails that are clustered inside these homogeneous communities.

- Infomap, which is the non-hierarchical version of InfoH, achieves quite good structural quality and decent logical quality. However, Blondel L1, which is based on the first level of Blondel's hierarchy of clusterings, yields much better logical quality than Infomap, but worse structural quality with respect to all of the structural quality functions.

- MCL and RN allow us to change the resolution of the clustering by modifying different parameters. When the granularity of their clusterings is set to be close to that of Blondel L1, they show almost similar community size distribution as well as similar structural and logical quality. However, Blondel L1 is superior to the other two methods due to its lower complexity.



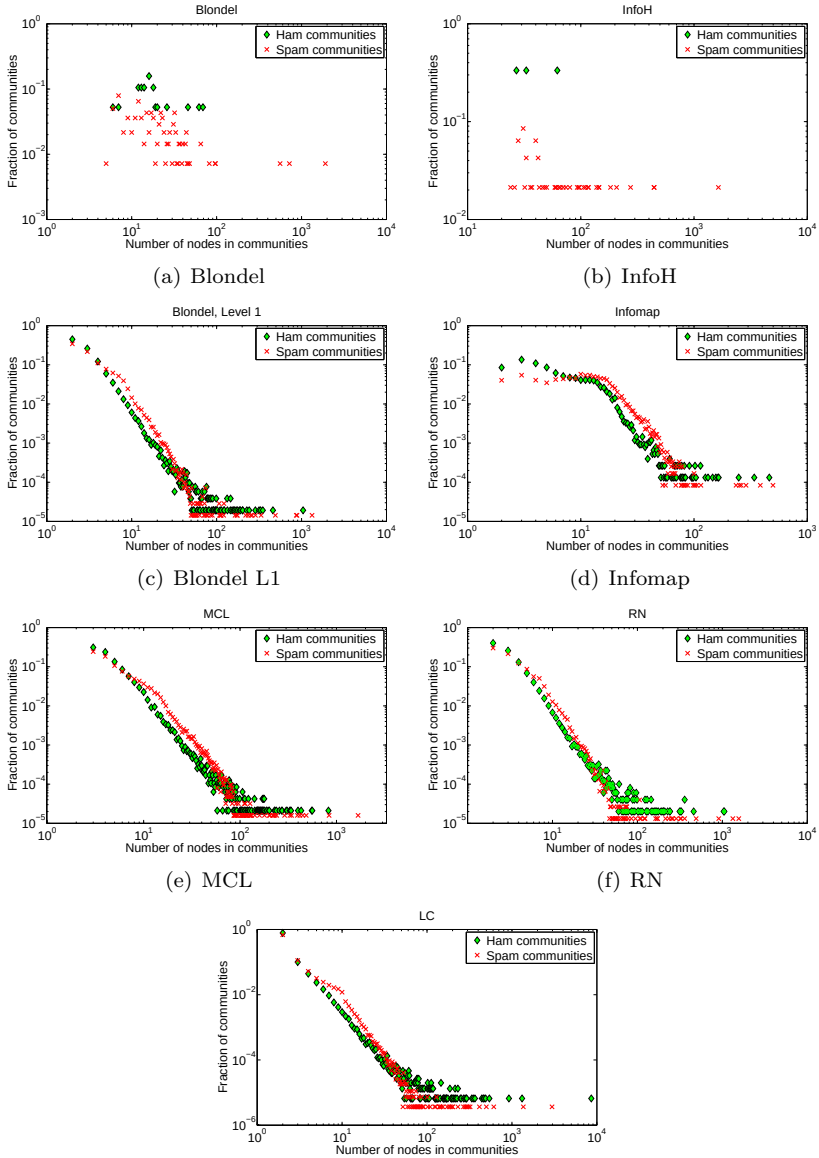
(a) Blondel (b) InfoH

(c) Blondel L1 (d) Infomap

(e) MCL (f) RN

(g) LC

Figure 3.6: *Comparison of community size distribution for the communities created by different algorithms from distinct ham and spam networks which were generated from "week 1" email data.*



(a) Blondel (b) InfoH

(c) Blondel L1 (d) Infomap

(e) MCL (f) RN

(g) LC

Figure 3.7: *Comparison of community size distribution for ham and spam communities (i.e., the mix communities are excluded) created by different algorithms from the mix "week 1" email network.*

- LC, which performs link community detection, has the best logical quality and separates the highest amount of spam and ham emails into distinct homogeneous communities.

3.6 Conclusions

In this study, we have performed an empirical comparison and evaluation of a number of high quality community detection algorithms using large-scale email networks. The studied email networks contain both legitimate and spam emails and are created from real email traffic. Our study reveals that yielding high structural quality by community detection algorithms is not enough to unfold the true logical communities of the email networks. Therefore, it is necessary to deploy more realistic measures for clustering real-world networks.

More specifically, our study suggests that the community detection algorithms that achieve maximum modularity, coverage, inter-cluster conductance, or minimum average conductance do not reveal the communities that coincide with the true clustering of the email networks. For instance the algorithms which yield worse, but acceptable, average conductance values actually could separate a large number of spam (ham) emails into distinct spam (ham) communities. Therefore, the value of this function can be indicative of good logical quality. However, this observation is based on our email networks, and might not be conclusive as it was shown that different classes of networks show different community structures [4, 8].

Overall, our experiments reveal that link community detection is the most suitable approach for separating spam and ham emails into distinct communities compared to the other node-based algorithms.

Acknowledgments

This work was supported by .SE – The Internet Infrastructure Foundation and SUNET. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/ 2007-2013) under grant agreement no. 257007.

Bibliography

[1] M Girvan and M E J Newman, "Community structure in social and biological networks.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, no. 12, pp. 7821–6, June 2002.

[2] Santo Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, Feb. 2010.

[3] Satu Elisa Schaeffer, "Graph clustering," *Computer Science Review*, vol. 1, no. 1, pp. 27–64, Aug. 2007.

[4] Andrea Lancichinetti and Santo Fortunato, "Community detection algorithms: A comparative analysis," *Physical Review E*, vol. 80, no. 5, pp. 1–11, Nov. 2009.

[5] Leon Danon, Albert Díaz-Guilera, Jordi Duch, and Alex Arenas, "Comparing community structure identification," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2005, no. 09, pp. P09008–P09008, Sept. 2005.

[6] Daniel Delling, Marco Gaertler, G Robert, Zoran Nikoloski, and Dorothea Wagner, "How to Evaluate Clustering Techniques," Tech. Rep., no. 2006-4, Universität Karlsruhe, 2006.

[7] Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove, "An analysis of social network-based Sybil defenses," in *Proceedings of the ACM SIGCOMM 2010 conference*, New York, New York, USA, 2010, p. 363, ACM Press.

[8] Helio Almeida, Dorgival Guedes, Wagner Meira Jr., and Mohammad J. Zaki, "Is There a Best Quality Metric for Graph Clusters?," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD*, Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, Eds. 2011, pp. 44–59, Springer-Verlag.

[9] M. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, pp. 1–15, Feb. 2004.

[10] R. Kannan, S. Vempala, and A. Veta, "On clusterings-good, bad and spectral," in *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 2000, pp. 367–377, IEEE Comput. Soc.

[11] Ulrik Brandes, Marco Gaertler, and Dorothea Wagner, "Experiments on Graph Clustering Algorithms," in *Proceedings of the 11th European Symposium on Algorithms*. 2003, pp. 568–579, Springer-Verlag.

[12] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, pp. 1–8, July 2009.

[13] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann, "Link communities reveal multiscale complexity in networks.," *Nature*, vol. 466, no. 7307, pp. 761–4, Aug. 2010.

[14] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. P10008, Oct. 2008.

[15] Martin Rosvall and Carl T Bergstrom, "Maps of random walks on complex networks reveal community structure.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–23, Jan. 2008.

[16] Martin Rosvall and Carl T Bergstrom, "Multilevel compression of random walks on networks reveals hierarchical organization in large integrated systems.," *PloS one*, vol. 6, no. 4, pp. e18209, Jan. 2011.

[17] Peter Ronhovde and Zohar Nussinov, "Multiresolution community detection for megascale networks by information-based replica correlations," *Physical Review E*, vol. 80, no. 1, pp. 1–18, July 2009.

[18] Stijn VAN Dongen, *Graph clustering by flow simulation*, Ph.D. thesis, University of Utrecht, The Netherlands, 2000.

[19] Gergely Tibély, Lauri Kovanen, Márton Karsai, Kimmo Kaski, János Kertész, and Jari Saramäki, "Communities and beyond: Mesoscopic analysis of a large social network with complementary methods," *Physical Review E*, vol. 83, no. 5, pp. 1–10, May 2011.

[20] Jure Leskovec, Kevin J. Lang, and Michael Mahoney, "Empirical comparison of algorithms for network community detection," in *Proceedings of the 19th international conference on World wide web*, New York, New York, USA, 2010, p. 631, ACM Press.

[21] Andrea Lancichinetti, Mikko Kivelä, Jari Saramäki, and Santo Fortunato, "Characterizing the community structure of complex networks.," *PloS one*, vol. 5, no. 8, pp. e11976, Jan. 2010.

[22] R. Guimerà, L. Danon, A. Díaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions.," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 68, no. 6 Pt 2, pp. 065103, Dec. 2003.

[23] Farnaz Moradi, Magnus Almgren, Wolfgang John, Tomas Olovsson, and Philippas Tsigas, "On Collection of Large-Scale Multi-Purpose Datasets on Internet Backbone Links," in *Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, 2011.

[24] Farnaz Moradi, Tomas Olovsson, and Philippas Tsigas, "Structural and Temporal Properties of E-mail and Spam Networks," Tech. Rep., no. 2011-18, Chalmers University of Technology, 2011.

PAPER III

Farnaz Moradi, Tomas Olovsson, Philippas Tsigas

Overlapping Communities for Identifying Misbehavior in Network Communications

Proceedings of the 18th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'14), Lecture Notes in Computer Science Vol.: 8443, pp. 398-409, Springer-Verlag, Tainan, Taiwan, May, 2014.

In order to comply with the thesis layout, this paper has been reformatted.

4 Overlapping Communities for Identifying Misbehavior in Network Communications

In this paper, we study the problem of identifying misbehaving network communications using community detection algorithms. Recently, it was shown that identifying the communications that do not respect community boundaries is a promising approach for network intrusion detection. However, it was also shown that traditional community detection algorithms are not suitable for this purpose.

In this paper, we propose a novel method for enhancing community detection algorithms, and show that contrary to previous work, they provide a good basis for network misbehavior detection. This enhancement extends disjoint communities identified by these algorithms with a layer of auxiliary communities, so that the boundary nodes can belong to several communities. Although non-misbehaving nodes can naturally be in more than one community, we show that the majority of misbehaving nodes belong to multiple overlapping communities, therefore overlapping community detection algorithms can also be deployed for intrusion detection.

Finally, we present a framework for anomaly detection which uses community detection as its basis. The framework allows incorporation of application-specific filters to reduce the false positives induced by community detection algorithms. Our framework is validated using large *email networks* and *flow graphs* created from real network traffic.

4.1 Introduction

Network intrusion detection systems are widely used for identifying anomalies in network traffic. Anomalies are patterns in network traffic that do not conform to normal behavior. Any change in the network usage behavior, for example caused by malicious activities such as DoS attacks, port scanning, unsolicited traffic, and worm outbreaks, can be seen as anomalies in the traffic.

Recently, it was shown that network intrusions can successfully be detected by examining the network communications that do not respect the community boundaries [1]. In such an approach, normality is defined with respect to social

behavior of nodes concerning the communities to which they belong and intrusion is defined as "*entering* communities to which one does not belong".

A community is typically referred to as a group of nodes that are densely interconnected and have fewer connections with the rest of the network. However, there is no consensus on a single definition for a community and a variety of definitions have been used in the literature [2–4]. For network intrusion detection, Ding et al. [1] defined a community as a group of source nodes that communicate with at least one common destination. They also showed that a traditional community detection algorithm which is based on a widely used definition, i.e., modularity, is not useful for identifying intruding nodes.

In this paper, we extend and complement the work of Ding et al. [1] by looking into other definitions for communities, and investigate whether the communities identified by different types of algorithms can be used as the basis for anomaly detection. Our hypothesis is that misbehaving nodes tend to *belong to multiple communities*. However, a vast variety of community detection algorithms partition network nodes into disjoint communities where each node only belongs to a single community, therefore they cannot be directly used for verifying our hypothesis. Therefore, we propose a simple novel method which enhances these disjoint communities with a layer of *auxiliary communities*. An auxiliary community is formed over the boundary nodes of neighboring communities, allowing nodes to be members of several communities. This enhancement enables us to show that, in contrary to [1], it is possible to use traditional community detection algorithms for identifying anomalies in network traffic.

In addition to traditional community detection algorithms, another class of algorithms exist which allow a node to belong to several overlapping communities [5]. In this study, we compare a number of such *overlapping algorithms* with our proposed enhancement method for non-overlapping community detection algorithms for network anomaly detection.

Finally, we propose a framework for network misbehavior detection. The framework allows us to incorporate different community detection algorithms for identifying anomalous nodes that belong to multiple communities. However, since legitimate nodes can also belong to several communities [4], application-specific filters can be used for discriminating the legitimate nodes from the anti-social nodes in the community overlaps, thus reducing the induced false positives.

We have evaluated the framework by using it for network intrusion detection and unsolicited email detection in large-scale datasets collected from a high-speed Internet backbone link. These types of misbehavior have traditionally been very hard to detect without inspecting the content of the traffic. To conclude, we show that by using our methodology, it is possible to effectively detect misbehaving traffic by only looking at the network communication patterns.

The remainder of the paper is organized as follows. Section 4.2 presents related work. Section 4.3 presents our proposed method for uncovering community

overlaps. The framework is presented in Section 4.4. Section 4.5 summarizes our findings and experimental results. Finally, Section 4.6 concludes our work.

4.2 Related Work

Anomaly detection has been extensively studied in the context of different application domains [6]. In this study, we propose a new graph-based anomaly detection method for identifying network intrusion and unsolicited email in real network traffic. Although there has been considerable amount of research on detecting these types of misbehavior, it is still a challenge to identify anomalies by merely investigating communication patterns without inspecting their content.

A taxonomy of graph-based anomaly detection methods can be found in [7]. A number of previous studies have proposed methods for finding unusual subgraphs, anomalous substructure patterns, and outlier nodes inside communities in labeled graphs [8–10]. In this study, we merely use the graph structure and therefore we consider only plain graphs without any labels.

Akoglu et al. [11] proposed a method to assign anomaly scores to nodes based on *egonet* properties in weighted networks. Our framework allows us to incorporate such properties as application-specific filters. Sun et al. [12] proposed a method for identifying anomalous nodes that are connected to irrelevant neighborhoods in bipartite graphs. Ding et al. [1] showed that although finding the cut-vertices can be used for intrusion detection, more robust results can be achieved by using clustering coefficient in a one-mode projection of a bipartite network. Moreover, they showed that using a modularity maximization community detection algorithm [13] is not suitable for spotting network intruders.

In this paper, we revisit the problem of finding anomalous nodes in bipartite/unipartite plain graphs by using community detection algorithms. We deploy an alternative definition for an anomaly as suggested in [1] and confirm their finding that maximizing modularity is not suitable for identifying intruders on its own. However, we show that there are several types of algorithms which are useful for misbehavior detection if enhanced with auxiliary communities.

4.3 Community Detection

In this section, we introduce a novel approach which enables us to deploy existing community detection algorithms for identifying anomalies in network traffic.

4.3.1 Auxiliary Communities

In this paper, we introduce the concept of auxiliary communities. An auxiliary community is added over the boundary nodes of disjoint communities, forcing nodes to become members of more than one community.

Algorithm 4.1: Neighboring Auxiliary Communities (NA)

Input: a graph $G(V, E)$; a non-overlapping community set $\mathcal{C}$;
Output: auxiliary community set $\mathcal{A}$;

```
1: for all v ∈ V do
2:    Com(v) = {C ∈ C : v ∈ V(C)};
3:    for all u ∈ Neighbors(v) do
4:       if Com(v) ≠ Com(u) then
5:          A ← A ∪ {u, v};
6:       end if
7:    end for
8:    A ← A ∪ A;
9: end for
10: return A
```

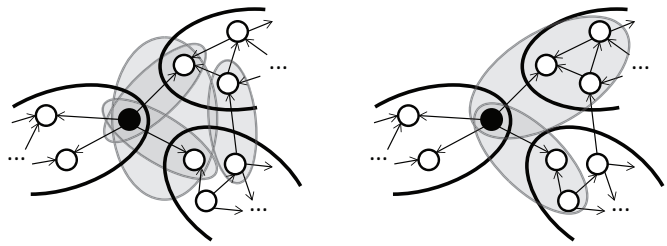
Algorithm 4.2: Egonet Auxiliary Communities of Sinks (EA)

Input: a graph $G(V, E)$; a non-overlapping community set $\mathcal{C}$;
Output: auxiliary community set $\mathcal{A}$;

```
1: for all v ∈ V do
2:    Com(v) = {C ∈ C : v ∈ V(C)};
3:    for all u ∈ Neighbors(v) do
4:       if Com(v) ≠ Com(u) and Sink(u) then
5:          A ← Egonet(u);
6:          A ← A ∪ A;
7:       end if
8:    end for
9: end for
10: return A
```

The most basic approach is to introduce one auxiliary community for each *boundary edge* between two different communities. However, a *boundary node* can have multiple boundary edges. Therefore, an improvement over the above approach is to add only one auxiliary community over a boundary node and all its boundary edges, covering all its neighbors that are members of other external communities (Algorithm 4.1). Our approach can be further refined to consider the whole one-step neighborhood, i.e., *egonet*, of a boundary node as an auxiliary community instead of just its boundary neighbors.

Ding et al. [1] defined a community in a directed bipartite network as a group of source nodes that have communicated with at least one common destination. In a bipartite network, there are two distinct sets of source nodes and destination nodes. Based on this definition, the source nodes that belong to the egonet of a destination node form a community. In a unipartite network, a distinct set



(a) Neighboring (b) Egonet of sinks

Figure 4.1: *Auxiliary communities.*

of source and destination nodes does not exist. Therefore, we apply the above definition of communities only to the sink nodes which have only incoming edges (Algorithm 4.2).

Figure 4.1 shows a comparison of the proposed methods for adding auxiliary communities. It can be seen that each approach places the intruding node (black node) in different auxiliary communities (grey communities). The main difference of our methods is that Algorithm 4.1 only adds neighboring auxiliary (NA) communities over the boundary nodes, whereas Algorithm 4.2 also allows the neighbors of the boundary sink nodes to be covered by egonet auxiliary (EA) communities. Therefore, a misbehaving node which is not in the boundary of its community can still belong to multiple communities by using Algorithm 4.2.

The complexity of adding auxiliary communities for a network with a degree distribution $p_k = k^{-\alpha}$, is $O(nk_{max}^{3-\alpha})$, where n is the number of nodes, k_{max} is the highest degree, and α is the exponent of the degree distribution.

4.3.2 Community Detection Algorithms

In this paper, we use a number of well-known and computationally efficient (overlapping) community detection algorithms, which are listed in Table 4.1. Our goal is to investigate which definition of a community and which types of algorithms are more suitable for network misbehavior detection.

LC and LG find overlapping communities in a graph based on the edges. LG, induces a *line graph* from the original network to which any non-overlapping algorithm can be applied. In this paper, we uses a weighted line graph with self-loops, E, and refer to LG using this graph as *LG(E)*. SLPA and OSLOM are both node-based methods and have very good performance [5]. Finally, DEMON is an state-of-the-art node-based, local, overlapping community detection algorithm.

The non-overlapping algorithms used in this study also have very good performance [14]. Blondel greedily maximizes modularity and unfolds a hierarchical community structure with increasing coarseness. In this study, we consider the communities identified at both the last and the first level of the hierarchy and refer

Table 4.1: *Community detection algorithms. n and m denote the number of nodes and edges, respectively, k_{max} is the maximum degree, t is the number of iterations, and α is the exponent of the degree distribution.*

Algorithm	Complexity	
Overlapping	*LC* [15]	$O(nk_{max}^2)$
	LG [16]	$O(nm^2)$
	SLPA [17]	$O(tm)$
	OSLOM [18]	$O(n^2)$
	DEMON [19]	$O(nk_{max}^{3-\alpha})$
Non-Overlapping	*Blondel* (also known as Louvain method) [20]	$O(m)$
	Infomap [21]	$O(m)$
Auxiliary	*NA* (Neighboring Auxiliary Communities)	$O(nk_{max}^{3-\alpha})$
	EA (Egonet Auxiliary Communities)	$O(nk_{max}^{3-\alpha})$

to them as *Blondel* and *Blondel L1*, respectively. We also use the communities formed by Blondel as input to OSLOM, which modifies these communities in order to improve their statistical significance. Finally, Blondel L1 is also used to partition the nodes in the induced line graphs by LG(E).

4.4 Framework

This section presents our framework for community-based anomaly detection. Algorithm 4.3 shows the first component of our framework, where overlapping algorithms can be directly used, but non-overlapping algorithms only after being enhanced with auxiliary communities.

The second component of our framework consists of a set of graph properties which are used as *filters*. Our hypothesis is that intruding nodes are likely to be placed in community overlaps. However, non-misbehaving nodes can also belong to more than one community, and basing detection merely on community overlaps, can lead to false positives. Therefore, these filters are used to reduce the induced false positives by the community detection algorithms.

The framework uses a simple method for combining the extracted properties. For each node v in the graph, the anomaly score is calculated as $score(v) = \sum_i w_{i\mathcal{I}}(\phi_i(v), t_i)$, where i is the index of the property which is being aggregated, w_i is a weight for property ϕ_i where $\sum w_i = 1$, and $_{\mathcal{I}}(\phi_i(v), t_i)$ is an indicator function which compares the value of a graph property $\phi_i(v)$ to a corresponding threshold value t_i such that $_{\mathcal{I}}(\phi_i(v), t_i) = \begin{cases} 1, & \phi_i(v) > t_i \\ 0, & \text{otherwise.} \end{cases}$

The threshold values and weights are dependent on the type of data and prior knowledge of normal behavior, which is necessary for anomaly detection and can be achieved from studies of anomaly-free data. Finally, the anomaly score $score(v)$ can be used to quantify to what extent a node v is anomalous.

Algorithm 4.3: Community-based anomaly detection

Input: a graph $G(V, E)$; a community detection algorithm CD;
Output: a set AS of $\langle v, score(v)\rangle$;

```
1: Set AS = ∅; Set C = ∅; Set A = ∅;
2: C = CD(G);
3: if CD is non-overlapping then
4:    A ← Auxiliary(G, C);
5:    C ← C ∪ A;
6: end if
7: for all v ∈ V do
8:    score(v) ← Filters(v, G, C);
9:    AS ← ⟨v, score(v)⟩;
10: end for
11: return AS
```

Algorithm 4.4: Application-specific filters

Input: a node v; a graph $G(V, E)$; a set of communities $\mathcal{C}$; weights $w_i \in [0, 1]$ s.t. $\sum w_i = 1$; user-defined threshold values t_i, where i is the index of the property;
Output: an anomaly score $score(v)$;

```
1: Coms(v) = {C ∈ C : v ∈ V(C)};
2: φ1(v) = |Coms(v)|;
3: φ2(v) = |Coms(v)|/|Neighbors(v)|;
4: φ3(v) = 1 − ClusteringCoeff(v);
5: φ4(v) = OutDeg(v)/Deg(v);
6: φ5(v) = Deg(v)/EdgeWeights(v);
7: score(v) = Σ wiI(φi(v), ti);
8: return score(v)
```

The properties presented in Algorithm 4.4 are examples of community and neighborhood properties that we have used as filters in our experiments for intrusion and unsolicited email detection. The selection of appropriate filters depends on the application of anomaly detection.

Network intruders are normally not aware of the community structure of the network, and therefore communicate to random nodes in the network [22]. It is expected to be very expensive for attackers to identify the network communities, and even if they do, limiting their communication with the members in the same community can inversely affect their gain. Therefore, the number of communities per node, as well as the ratio of the number of communities per node over the number of its neighbors, which correspond to ϕ_1 and ϕ_2 in Algorithm 4.4, respectively, are expected to be promising properties for finding intruders.

The rest of the properties, are graph metrics that correspond to the social behavior of nodes and can be extracted from the direct neighborhood of the nodes. We have used these properties for detecting unsolicited email (Section 4.5.3) and therefore in the following we explain them in the context of spam detection.

The *clustering coefficient* of a node is known to have a lower value for spammers than legitimate nodes [23, 24]. Property ϕ_3 calculates one minus the clustering coefficient so that spammers are assigned higher values. It has also been shown that spammers are mostly using randomized fake source addresses and therefore it is not expected that they receive many emails [25]. Property ϕ_4 calculates the ratio of the out-degree over the degree of the nodes, which is expected to be high for the spammers. Finally, it has been shown that spammers tend to use the fake source email address to send only a few spam, and target each receiving email address only once [25]. Therefore, the degree of a node over its edge weights, property ϕ_5, is expected to be higher for spammers than legitimate nodes, where the edge weights correspond to the number of exchanged emails.

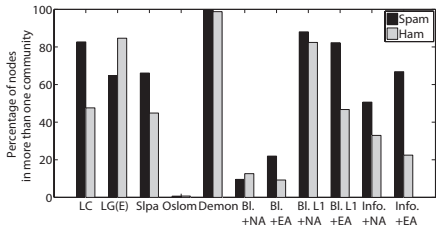
4.5 Experimental Results

We have evaluated the usability of different algorithms in our framework using two different datasets which were generated from network traffic collected on a 10 Gbps Internet backbone link of a large national university network.

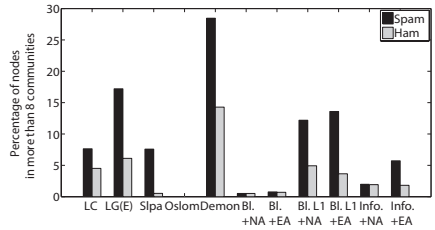
Flow Dataset. The flow level data was collected from the incoming network traffic once a week during 24 hours for seven weeks in 2010 [26]. The flows were used to generate bipartite networks where source and destination IP addresses form the two node sets. The malicious source addresses in the dataset were taken from the lists reported by DShield and SRI during the data collection period [27, 28]. This dataset is used to compare our approach with the method proposed by Ding et al. [1] for network intrusion detection. The datasets are similar with respect to the ground truth and only differ with respect to the collection location and the sampling method used.

Email Dataset. This dataset is generated from captured SMTP packets in both directions of the backbone link. The collection was performed twice (2010 and 2011), where the duration of each collection was 14 consecutive days. This dataset was used for generating *email networks*, in which email addresses represent the nodes, and the exchanged emails represent the edges. The ground truth was obtained from a well-trained content-based filtering tool[1] which classified each email as legitimate (*ham*) or unsolicited (*spam*).

[1]SpamAssassin (`http://spamassassin.apache.org`) which provided us with an estimated false positive rate of less than 0.1% and a detection rate of 91.4%



(a) percentage of nodes in more than one community



(b) Percentage of nodes in more than 8 communities

Figure 4.2: *Percentage of nodes in multiple communities in email dataset (2010).*

4.5.1 Comparison of Algorithms

In this section, we present a comparison of the algorithms using the email dataset. Figure 4.2(a) shows the percentage of ham and spam nodes (averaged over the 14 days in 2010), which are placed in multiple communities by different algorithms. It can be seen that many ham nodes belong to more than one community, which is an expected social behavior. It can also be seen that, most algorithms place the majority of spammers into more than one community, except OSLOM and Blondel which tend to form very coarse-grained communities.

The figure also shows that, regardless of which non-overlapping algorithm being used, adding egonet auxiliary communities (Algorithm 4.2) places more spam than ham nodes into several communities compared to adding neighboring auxiliary communities (Algorithm 4.1). The reason is that NA communities are only added over the boundary nodes, however, EA communities also allow the neighbors of the boundary sink nodes to be covered by auxiliary communities.

Finally, Figure 4.2(b) shows that a higher percentage of spammers belong to more than eight communities compared to legitimate nodes. The same observation holds for the data collected in 2011. Therefore, we can confirm that both fine-grained algorithms enhanced with EA communities, and overlapping algorithms can be used to spot misbehaving nodes based on the number communities to which they belong.

4.5.2 Network Intrusion Detection

It has been shown that a non-overlapping community detection algorithm (which maximizes modularity) is not suitable for identifying intruders in network flow data [1]. In this study, we have further investigated the possibility of using different community detection algorithms, including a modularity-based one, by using auxiliary communities for network intrusion detection.

One example of network intrusion is port scanning, where a scanner searches for open/vulnerable services on selected hosts. Current intrusion detection systems are

quite successful in identifying scanners. In this paper, we just verify the possibility of detecting scanners using a community-based technique.

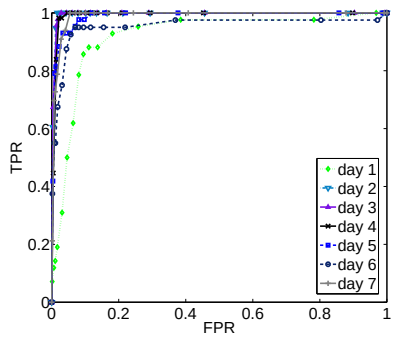
We generated one bipartite graph from the flows collected for each day. As an example, the flow graph generated from the first day of data contained 51,720 source nodes sending 93,113 flows to 32,855 destination nodes. This includes 607 malicious nodes (based on DShield/SRI reports) that have sent 7,861 flows. We made the assumption that the malicious source nodes that have tried to communicate with more than 50 distinct destinations are suspected of scanning. Figure 4.3(a) shows the ROC curves for seven different days. These curves show the trade-off between the true positive rate (TPR) and the false positive rate (FPR). We have used Blondel L1 enhanced with egonet auxiliary communities (EA), and have only used property ϕ_1, i.e., the number of communities to which a node belongs, as the filter. It can be seen that this approach yields high performance with mean area under curve (AUC) of 0.98, where around 90% to 100% of malicious scanners are detected with a FPR of less than 0.05. This observation confirms that our framework is successful in identifying scanners.

Network intrusion attacks are not limited to scanning attacks, therefore we have also tried to identify other malicious (DShield/SRI) sources and have compared our approach with the method proposed by Ding et al. [1]. Our experiments show that the performance of both methods are quite consistent with mean AUC 0.60 (standard error 0.009) for the method by Ding et al. and 0.62 (standard error 0.015) for our approach using LG(E) as the overlapping community detection and properties ϕ_1 and ϕ_2 as filters. Overall, these results confirm that the community structure of a network provides a good basis for network intrusion detection and both non-overlapping communities enhanced with EA communities and overlapping communities can indeed be used for this purpose.

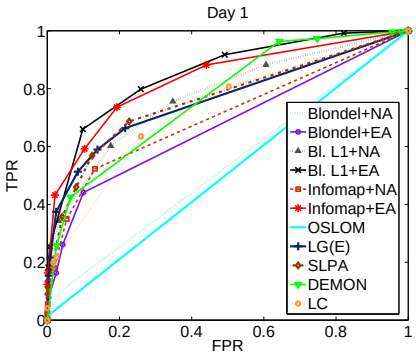
4.5.3 Unsolicited Email Detection

Our experimental comparison of community detection algorithms in Section 4.5.1 showed that most of the studied algorithms place spammers into multiple communities. In this section, we investigate how these algorithms can be used in our framework to detect these spammers only by observing communication patterns.

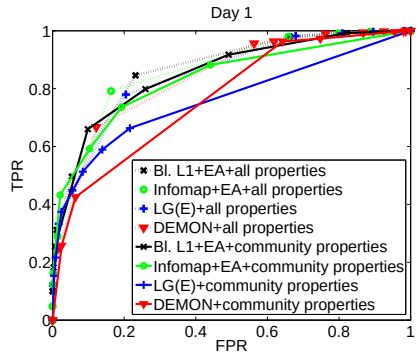
For this study, we have generated one email network from the emails collected for each day. The community detection algorithms were applied to the undirected and unweighted giant connected component of each email network. The edge directions and weights were later taken into account for adding auxiliary communities and calculating different graph properties. We consider an email address to be a spammer if it has sent more than one spam to more than one recipient. As an example, the email network generated from the first day of data in 2010, contains 167,329 nodes and 236,673 edges, where 23,628 nodes were spammers sending 126,145 spam emails. It is important to note that the vast majority of the spammers have not sent large volumes of email and therefore a simple volume-based detection method would not be suitable for spammer detection.



(a) Scanner detection using community properties (2010)



(b) Spam detection using community properties (2010)

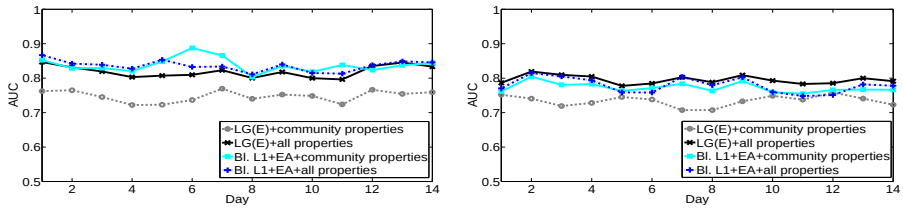


(c) Spam detection using neighbor and community properties (2010)

Figure 4.3: *Performance of different algorithms for network misbehavior detection.*

Figure 4.3(b) shows the ROC curves for our spam detection method using different algorithms and the community-based properties ϕ_1 and ϕ_2. It can be seen that OSLOM, which aims at forming statistically significant communities fails to identify spamming nodes. It can also be seen that a node-based overlapping algorithm, SLPA, and an edge-based algorithm, LG(E), perform similarly, and the AUC (not shown in the figure) is identical for both algorithms (0.76).

Figure 4.3(b) also shows the ROC curves for non-overlapping algorithms which are enhanced with our auxiliary communities. It can be seen that Blondel, which aims at optimizing modularity, performs very poor. This observation is in accord with the observation in [1] that a modularity maximization algorithm is not suitable for anomaly detection due to its resolution limit. However, Blondel L1 (first level in the community hierarchy of Blondel), which forms finer granularity communities, performs dramatically better than its last level using either type of the auxiliary



(a) 2010 (b) 2011

Figure 4.4: *Area under the ROC curve for spam detection over time.*

communities. Moreover, it can be seen that adding EA communities leads to better results compared to NA communities.

Overall, our experiments for different days in both email datasets showed that Blondel L1 and Infomap enhanced with EA, SLPA, LG(E), and DEMON all perform well with respect to placing spamming nodes into multiple communities. In practice, low false positive rates are essential for spam detection, therefore both Blondel L1 with EA communities and LG(E) that allow us to, on average, detect more than 25% and 20% of spamming nodes, respectively, for different days with very low FPR (less than 0.01) are the most suitable algorithms.

These results confirm that our method for adding EA communities to enhance non-overlapping algorithms yields not only comparable, but even better, results than an overlapping algorithm. Although both Blondel L1 with EA communities and LG(E) use the same modularity-based algorithm as their basis (we have applied Blondel L1 on the induced line graph of LG(E)), adding EA communities has also a lower complexity than inducing weighted line graphs (Table 4.1).

As mentioned earlier, our framework allows us to incorporate a number of application-specific filters to reduce the induced false positives (Algorithm 4.4). Figure 4.3(c) shows a comparison of the spam detection using filters based on community properties (ϕ_1 and ϕ_2 only) and the combination of community and neighborhood properties (ϕ_1 - ϕ_5) for the first day of data in 2010. It can be seen that use of additional filters improves the detection (the same observation also holds for the algorithms not shown).

Finally, Figure 4.4 shows the AUC for spam detection using our framework with LG(E) and Blondel L1 enhanced with EA communities over 14 days during 2010 and 2011. It can be seen that the results are quite stable over time and the AUC of our method for adding EA communities compared to a more complex overlapping algorithm is much better when only community properties are used.

4.6 Conclusions

In this paper, we have evaluated the performance of community detection algorithms for identifying misbehavior in network communications. This paper extends and complements the previous work on community-based intrusion detection, by investigating a variety of definitions for a community, introducing auxiliary communities for enhancing traditional community detection algorithms, and showing that, in contrary to previous work, these algorithms can indeed be deployed as the basis for network anomaly detection.

We have also provided a framework for community-based anomaly detection which allows us to find the nodes that belong to multiple communities by either using auxiliary communities or overlapping algorithms. It also enables us to deploy neighborhood properties, which are indicative of social behavior, for discriminating the nodes that naturally belong to more than one community from the anti-social ones. The applicability of our framework for identifying network intrusions and unsolicited emails was evaluated using two different datasets coming from traffic captured on an Internet backbone link. Our experiments show that our framework is quite effective and provides a consistent performance over time. These results suggest that detecting community overlaps is a promising approach for identifying misbehaving network communications.

Acknowledgments

This work was supported by .SE – The Internet Infrastructure Foundation and SUNET. The research leading to these results has also received funding from the European Union Seventh Framework Programme (FP7/ 2007-2013) under grant agreement no. 257007.

Bibliography

[1] Qi Ding, Natallia Katenka, Paul Barford, Eric Kolaczyk, and Mark Crovella, "Intrusion as (anti)social communication," in *Proceedings of the 18th ACM SIGKDD conference on Knowledge discovery and data mining - KDD '12*, 2012, p. 886.

[2] Santo Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, Feb. 2010.

[3] Jure Leskovec, Kevin J. Lang, and Michael Mahoney, "Empirical Comparison of Algorithms for Network Community Detection," in *Proceedings of the 19th international conference on World wide web*, 2010, p. 631.

[4] Jaewon Yang and Jure Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2012, pp. 1–8.

[5] Jierui Xie, S Kelley, and BK Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Computing Surveys*, vol. 45, no. 4, 2013.

[6] Varun Chandola, Arindam Banerjee, and Vipin Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys*, vol. 41, no. Sep, pp. 1–72, 2009.

[7] Leman Akoglu and Christos Faloutsos, "Anomaly, event, and fraud detection in large network datasets," in *Proceedings of the sixth ACM international conference on Web search and data mining - WSDM '13*. 2013, p. 773, ACM Press.

[8] William Eberle and Lawrence Holder, "Anomaly detection in data represented as graphs," *Intelligent Data Analysis*, vol. 11, no. 6, pp. 663–689, 2007.

[9] Caleb C Noble and Diane J Cook, "Graph-based anomaly detection," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '03*, 2003, vol. 1, p. 631.

[10] Jing Gao, Feng Liang, Wei Fan, Chi Wang, Yizhou Sun, and Jiawei Han, "On community outliers and their efficient detection in information networks," in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '10*. 2010, p. 813, ACM Press.

[11] Leman Akoglu and M McGlohon, "Oddball: Spotting Anomalies in Weighted Graphs," in *Proceedings of the 14th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*, 2010, pp. 410–421.

[12] Jimeng Sun, Deepayan Qu, Huiming Chakrabarti, and Christos Faloutsos, "Neighborhood Formation and Anomaly Detection in Bipartite Graphs," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*. 2005, pp. 418–425, IEEE.

[13] Aaron Clauset, M E J Newman, and Cristopher Moore, "Finding community structure in very large networks.," *Physical review. E, Statistical, nonlinear, and soft matter physics*, vol. 70, no. 6 Pt 2, pp. 066111, Dec. 2004.

[14] Andrea Lancichinetti and Santo Fortunato, "Community Detection Algorithms: A Comparative Analysis," *Physical Review E*, vol. 80, no. 5, pp. 1–11, Nov. 2009.

[15] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann, "Link communities reveal multiscale complexity in networks.," *Nature*, vol. 466, no. 7307, pp. 761–4, Aug. 2010.

[16] T. Evans and R. Lambiotte, "Line graphs, link partitions, and overlapping communities," *Physical Review E*, vol. 80, no. 1, pp. 1–8, July 2009.

[17] Jierui Xie and BK Szymanski, "Towards Linear Time Overlapping Community Detection in Social Networks," in *the 16th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining*. 2012, pp. 25–36, Springer-Verlag.

[18] Andrea Lancichinetti, Filippo Radicchi, José J Ramasco, and Santo Fortunato, "Finding statistically significant communities in networks.," *PloS one*, vol. 6, no. 4, pp. e18961, Jan. 2011.

[19] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi, "DEMON: a local-first discovery method for overlapping communities," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. 2012, p. 615, ACM Press.

[20] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre, "Fast Unfolding of Communities in Large Networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, pp. P10008, Oct. 2008.

[21] Martin Rosvall and Carl T Bergstrom, "Maps of random walks on complex networks reveal community structure.," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 105, no. 4, pp. 1118–23, Jan. 2008.

[22] Nisheeth Shrivastava, Anirban Majumder, and Rajeev Rastogi, "Mining (Social) Network Graphs to Detect Random Link Attacks," in *24th International Conference on Data Engineering*. Apr. 2008, vol. 00, pp. 486–495, IEEE.

[23] Luiz H Gomes, Rodrigo B Almeida, and Luis M A Bettencourt, "Comparative Graph Theoretical Characterization of Networks of Spam and Legitimate Email," in *Conference on Email and Anti-Spam (CEAS)*, 2005.

[24] Farnaz Moradi, Tomas Olovsson, and Philippas Tsigas, "Towards modeling legitimate and unsolicited email traffic using social network properties," in *Proceedings of the Fifth Workshop on Social Network Systems - SNS '12*, 2012.

[25] Christian Kreibich, Chris Kanich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage, "On the Spam Campaign Trail," in *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, June 2008, vol. 453, pp. 697–8.

[26] Magnus Almgren and Wolfgang John, "Tracking Malicious Hosts on a 10Gbps Backbone Link," in *15th Nordic Conference in Secure IT Systems*, 2010.

[27] DShield, "Recommended block list," 2010.

[28] SRI International Malware Threat Center, "Most aggressive malware attack source and filters," 2010.

PAPER IV

Farnaz Moradi, Tomas Olovsson, Philippas Tsigas

A Local Seed Selection Algorithm for Overlapping Community Detection

Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'14), Beijing, China, August, 2014.

In order to comply with the thesis layout, this paper has been reformatted.

5 A Local Seed Selection Algorithm for Overlapping Community Detection

One of the widely studied structural properties of social and information networks is their community structure, and a vast variety of community detection algorithms have been proposed in the literature. Expansion of a seed node into a community is one of the most successful methods for local community detection, especially when the global structure of the network is not accessible. An algorithm for local community detection only requires a partial knowledge of the network and the computations can be done in parallel starting from seed nodes. The parallel nature of local algorithms allow for fast and scalable solutions, however, the coverage of the communities heavily depends on the seed selection. The communities identified by a local algorithm might cover only a subset of the nodes in a network if the seeds are not selected carefully.

In this paper, we propose a novel *seeding algorithm* which is parameter free, utilizes merely the local structure of the network, and identifies good seeds which span over the whole network. In order to find such seeds, our algorithm first computes similarity indices from local link prediction techniques to assign a *similarity score* to each node, and then a *biased graph coloring* algorithm is used to enhance the seed selection. Our experiments using large-scale real-world networks show that our algorithm is able to select good seeds which are then expanded into high quality overlapping communities covering the vast majority of the nodes in the network using a personalized PageRank-based community detection algorithm. We also show that using our local seeding algorithm can dramatically reduce the execution time of community detection.

5.1 Introduction

The emergence of large-scale social and information networks have motivated numerous studies of the structural properties of these networks such as their community structure. A community typically refers to a group of densely connected

nodes which have sparse connections with the rest of the nodes in the network, and a wide variety of algorithms have been proposed for identifying communities [1, 2].

Community detection algorithms can be divided into global and local algorithms. *Global algorithms* require a global knowledge of the entire structure of the network in order to uncover all the communities in that network. Since such knowledge might not be available for large-scale networks, local algorithms are gaining more popularity [3–6]. *Local algorithms* typically start from a number of *seed* nodes (sets) and expand them into possibly overlapping communities by examining only the neighborhood of the seeds. Due to their nature, local algorithms can be parallelized and are scalable. However, they might only cover a subset of the nodes in a network if the seeds are not chosen carefully. A naive approach for achieving high coverage is therefore to consider all the nodes in a network as seeds. However, this approach is computationally expensive and leads to many redundant communities. Although the goal of local algorithms is not to achieve a complete coverage of a network, finding a small number of seeds which are well distributed over the network and can lead to a high coverage is very desirable.

Since our knowledge of the community structure of large-scale real-world networks is usually limited, finding good seeds that span over the network using only the knowledge of the local structure of a network is a challenging problem. In this paper, we present a novel local seed selection algorithm which achieves a high coverage and a community quality similar to the naive approach (where all nodes are used as seeds) but with a significantly lower execution time.

Our algorithm uses similarity indices from *link prediction* techniques. In link prediction, similarity indices are used to estimate the similarity of nodes which are expected to get connected, however, we use them to asses the similarity of nodes which are already connected. We assign a local *similarity score* to each node based on a similarity index and identify nodes that are similar to their neighbors and therefore are expected to be in the same community. Andersen et al. [7] theoretically showed that a seed set that is "nearly contained" in a target community is a good seed set for that community. We select a node as a seed if it has the highest score among its neighbors, and we show that this method is very effective in finding a small number of very good seeds in a network which can be expanded into high quality communities. However, similar to other existing local seeding algorithms, the communities expanded from these seeds do not achieve a high coverage of the network.

In order to improve the coverage, we propose to use distributed *graph coloring*. Although we show that we can select good seeds using graph coloring, we also introduce a new distributed *biased graph coloring* algorithm to further enhance our seeding algorithm, where the nodes with the highest local similarity score, which are expected to be good seeds, are assigned a specific color. Then the ties are broken at random so that no two adjacent nodes pick the same color. In the end, the nodes which received the specific color are selected as seeds. Our proposed algorithm is parameter free, is computed locally, selects seeds from parts of the

network where the other local similarity methods fail to pick any seeds, and does not lead to many duplicate communities since it does not pick any neighboring nodes as seeds.

The selected seeds are then expanded into overlapping communities using a personalized PageRank-based local community detection algorithm, which can be computed locally and is known to result in high quality communities [8]. We have empirically compared our proposed seeding algorithm with a number of existing seeding methods, as well as a state-of-the-art local community detection algorithm with respect to quality and coverage of the identified communities. The quality is assessed using ground truth data where such data exists, and *conductance* which is a widely used quality function.

Overall, our contributions in this paper are as follows.

- We define a similarity score which is calculated as the sum of the similarity of a node with all of its connected neighbor by adopting the similarity indices from link prediction techniques.
- We propose a new local seeding algorithm which uses these similarity scores (link prediction-based seeding).
- We propose to use graph coloring for picking random seeds in a network and introduce biased graph coloring for enhancing our seeding algorithm (biased coloring-based seeding).
- We empirically compare the different similarity indices which we have used in our seeding algorithm. We also experimentally evaluate our seeding algorithm and show that it can find a reasonably small number of seeds which are expanded into communities with high coverage and a similar quality compared to when all the nodes are used as seeds but with significantly reduced execution time.
- We show that our biased coloring algorithm is also successful in improving the coverage of other existing local seeding algorithms.

The remainder of the paper is organized as follows. Sections 5.2 and 5.3 present the related work and the background, respectively. Our seeding algorithm is presented in Section 5.4. Section 5.5 presents the experimental results. Finally, Section 5.6 concludes our work.

5.2 Related Work

There have been numerous studies proposing different types of community detection algorithms [1, 2]. In this paper, we only consider local algorithms.

Coscia et al. [6] have proposed the *Demon* algorithm, which starts from all the nodes in a network to identify the local communities in each neighborhood and then

uses merging to form the optimal global communities. A closely related approach is the *Node Perception* by Soundarajan et al. [4] which is a template for first finding local sub-communities and then identifying all the communities.

There are a variety of local community detection algorithms which assume that the seeds are given, e.g., [3] or can be picked at random, e.g., [9]. However, there are not many studies which have looked into the problem of selecting *good seeds*. Shen et al. [10] proposed to use maximal cliques, which form the core of the communities, as seeds which is computationally expensive. Gargi et al. [11] used the number of times a video has been viewed in the Youtube network to select the top videos as seeds, however, this type of non-structural information is not available for many networks.

Gleich et al. [12] showed that the *egonets* with low conductance are good seeds for finding the best communities of a network with respect to conductance. However, Whang et al. [5] showed that these communities do not achieve high coverage. Chen et al. [13] proposed an algorithm for selecting the nodes with local maximal degree as seeds. The authors suggested to remove the identified communities expanded from these seeds from the network and find new seeds in the remaining parts of the network repeatedly to improve the coverage. These methods are explained in more detail in the next section and are compared against our proposed seeding algorithm.

Whang et al. [5] have proposed two seeding algorithms which achieve high coverage. In the *Graclus centers* they run a partitioning algorithm to create k network partitions and then the nodes in the center of these partitions are selected as seeds. In the *spread hub* algorithm, at least k nodes with the highest degree in the network are selected as seeds. Both seeding algorithms require some global knowledge as well as the number of seeds to be known which is not a realistic assumption since we typically do not know the community structure of the real-world networks in advance.

Our seeding algorithm is parameter free and uses similarity indices from local link prediction and local graph coloring. Yan and Gregory [14] have used a similarity index to add edge weights to unweighted networks in order to improve the quality of existing global community detection algorithms. Psicologia et al. [15] have used simple graph coloring as the first step for a label propagation community detection algorithm. These works do not introduce local seeding algorithms and therefore are fundamentally different from our work.

Our algorithm can be used for seeding any local community detection algorithm. In this paper, we have used a variant of a personalized PageRank algorithm by Yang et al. [8]. Although Yang et al. have shown that this algorithm is very successful in identifying the communities to which a given seed belongs, they did not investigate the effect of using a seeding algorithm.

5.3 Background

5.3.1 Notations

Let $G = (V, E)$ be a connected, undirected, and unweighted graph, where V is the set of n nodes and E is the set of m edges or links of G. Let $v \in V$ be a node in G. The set of the neighbors of v is denoted by $\Gamma(v) = \{u : u \in V, (u, v) \in E\}$. The degree of v is shown as $k_v = |\Gamma(v)|$, and Δ refers to the maximum degree in the graph. The egonet of v is the subgraph induced by the node and its neighbors and is defined as $egonet(v) = \{v\} \cup \{u : u \in \Gamma(v), (u, v) \in E\}$.

A local community detection algorithm expands a seed node s into a community C which is a set of nodes including s. We denote by $\mathcal{C} = \{C_1, \ldots, C_k\}$ the collection of overlapping communities expanded from k distinct seed nodes which are selected by a seeding algorithm. The *coverage* of the collection of communities $\mathcal{C}$ is defined as $cov(\mathcal{C}) = \frac{|\bigcup_{i=1}^{k} C_i|}{|V|}$. The *conductance* of a community, which is used both as a scoring function and as a quality function, is defined as $\phi(C) = \frac{\overline{m}(C)}{\min(vol(C), vol(V \setminus C))}$, where $\overline{m}(C) = |\{(u, v) \in E : u \in C, v \notin C\}|$ is the number of inter-cluster edges and $vol(C) = \sum_{v \in C} k_v$ is the volume of a community C and corresponds to the sum of the degree of all the nodes in the community.

5.3.2 Existing Seeding Methods

In this study, we have selected a number of state-of-the-art algorithms to be compared against our proposed algorithm.

Spread hub (SH) [5] In this method, first the nodes are sorted in order of decreasing degree. Then, as long as the number of selected seeds is less than k, the nodes with the maximum degree are greedily chosen as seeds. This algorithm can pick more than k seeds, where k is given as input, and only picks neighboring nodes as seeds when their degree is equal. The complexity of SH is $O(n \log n + k)$.

Low conductance cuts (EC) [12] Gleich et al. have shown that the low conductance egonets are good seed sets. This algorithm selects around 3% of the network nodes as seeds. A node v can be a seed if for all $u \in \Gamma(v)$, $\phi(egonet(v)) \leq \phi(egonet(u))$. EC can find these seeds with time complexity $O(m\Delta)$. Whang et al. [5] showed that this method performs poorly with respect to coverage.

Local maximal degree (MD) [13] This algorithm uses a list of nodes in the graph. If a node has the highest local degree, it is added to a seed set and is removed from the list together with all its neighbors with lower degrees. If a node is not a *local-maximal-degree* node, it is also removed from the list. This process is repeated until all the nodes are removed from the list. The complexity of MD is $O(n\Delta)$.

5.3.3 Link Prediction and Similarity Indices

Link prediction is the problem of predicting the relations that should exist in a network or are very likely to be formed in the future. These methods typically estimate the similarity of nodes which are not connected to each other using similarity indices. We have selected a number of basic and widely used similarity indices for local link prediction [16].

Neighbors index (CN) is a very basic metric which calculates the size of the neighborhood overlap of two nodes and is formally defined as

$$CN(u, v) = |\Gamma(u) \cap \Gamma(v)|.$$

Hub promoted index (HP) assigns higher scores to the edges adjacent to high degree nodes (*hubs*) and is defined as

$$HP(u, v) = CN(u, v)/min(k_u, k_v).$$

Leicht-Holme-Newman index (LHN) assigns high values to the nodes that have many common neighbors compared to the expected number of neighbors and is defined as

$$LHN(u, v) = CN(u, v)/(k_u \times k_v).$$

Resource Allocation index (RA) is motivated by the resource allocation process where the common neighbors of two nodes act like transmitters which distribute their recourses to all their neighbors. Therefore, the amount of recourses a node u receives from a node v can be used for calculating their similarity as

$$RA(u, v) = \sum_{w \in \Gamma(u) \cap \Gamma(v)} \frac{1}{k_w}.$$

Preferential Attachment (PA) is motivated by the preferential attachment mechanism, where the probability that a new link is connected to a node v is proportional to the degree of the node k_v and is defined as

$$PA(u, v) = k_u \times k_v.$$

5.3.4 Graph Coloring

The problem of coloring the nodes of a graph with a small number of colors is a fundamental graph problem and has been widely studied. The goal of a graph coloring algorithm is to color the nodes in a graph with at most $\Delta + 1$ colors, where Δ is the maximum degree in the graph, so that no two neighboring nodes share the same color. Coloring has many applications such as assigning time or frequency slots for communications of wireless devices.

Algorithm 5.1: Link prediction-based seed selection

Input: A graph $G(V, E)$.
Output: The seed set S.
Let $S = \emptyset$;
for all $v \in V$ **do**
  $score(v) = \sum_{u \in \Gamma(v)} sim(u, v)$;
end for
for all $v \in V$ **do**
  if $score(v) > 0$ **and** $(\forall u \in \Gamma(v))\ score(v) \geq score(u)$ **then**
    $S = S \cup \{v\}$;
  end if
end for
return S

The most well-known distributed algorithm for $\Delta+1$ graph coloring is a randomized algorithm based on the maximum independent set algorithm of Luby [17, 18] which needs $O(\log n)$ time. Barenboim et al. [19] have shown that deterministic distributed coloring can be implemented in linear $O(\Delta)$ time.

In distributed graph coloring, each node picks a color uniformly at random from the set of colors which are available to it, and solves the conflicts with its neighbors by picking new colors and exchanging confirmations. Eventually, the algorithm converges when each node has a color different from the colors of all its neighbors.

5.4 Our Method

In this section we present our approach to overlapping community detection in large-scale networks using our novel seeding algorithm and a personalized PageRank-based seed expansion algorithm.

5.4.1 Link Prediction-based Seed Selection

In our seeding algorithm, we propose to use similarity indices from link prediction methods to calculate the similarity of the nodes which are directly connected. Our intuition is that if a node has high similarity with its neighbors, it is expected that they belong to the same community. Moreover, a node is a good seed if it has many neighbors in the target community [7]. Therefore, a node which is very similar to its neighbors can be a good representative for its neighborhood, thus can be selected as a seed for local community detection.

Our seed selection algorithm is presented in Algorithm 5.1. Each node v calculates its similarity with its direct neighbors and assigns a $score(v)$ to itself based on the sum of the similarities. The $sim(u, v)$ function refers to any of the similarity

Algorithm 5.2: Biased coloring-based seed selection

Input: A graph $G(V, E)$.
Output: The seed set S.

```
    Let S = ∅;
 2: for all v ∈ V do
      score(v) = Σ_{u∈Γ(v)} sim(u, v);
 4: end for
    for all v ∈ V do
 6:   Let SC = ∅;
      ∀u ∈ Γ(v), confirm(u, v) = 0; converge(v) = false; color(v) = 0;
 8:   available_colors(v)={c_1, ..., c_{k_v+1}} where k_v = |Γ(v)|;
      SC = {score(u) : ∀u ∈ egonet(v)};
10:   for all u ∈ egonet(v) do
        if score(u) = max(SC) then
12:       color(u) = c_1;
        end if
14:   end for
      if color(v) = 0 then
16:     color(v) = pick_color(available_colors(v));
      end if
18:   while converge(v) = false do
        for all u ∈ Γ(v) do
20:       if color(v) = color(u) and score(v) ≤ score(u) then
            color(v) = pick_color(available_colors(v));
22:       else if color(u) > 0 then
            confirm(u, v) = 1;
24:       end if
        end for
26:     if ∀u ∈ Γ(v), confirm(u, v) = 1 and color(v) > 0 then
          converge(v) = true;
28:     end if
      end while
30:   if color(v) = c_1 and k_v > 1 then S = S ∪ {v}; end if
    end for
32: return S
```

indices introduced in the previous section. Then, each node compares its score with its neighbors and decides if it is a seed or not.

Table 5.1 shows a summary of the names we use in the rest of the paper for the instances of our seeding algorithm when different similarity indices are used for calculating the score of the nodes.

Table 5.1: *Summary of the names used for the instances of our seed selection algorithm based on the similarity indices being used.*

	Similarity index $sim(u, v)$	Instance name
Link prediction-based Seeding (Algorithm 5.1)	$CN(u, v)$	CN
	$HP(u, v)$	HP
	$LHN(u, v)$	LHN
	$RA(u, v)$	RA
	$PA(u, v)$	PA
Biased coloring-based Seeding (Algorithm 5.2)	$CN(u, v)$	CN + coloring
	$HP(u, v)$	HP + coloring
	$LHN(u, v)$	LHN + coloring
	$RA(u, v)$	RA + coloring
	$PA(u, v)$	PA + coloring
Random coloring	-	RN (coloring)

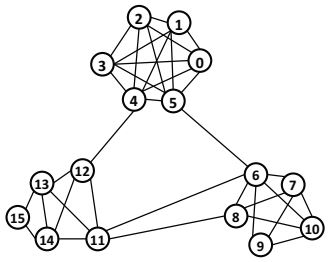
5.4.2 Biased Coloring-based Seed Selection

Although our proposed seeding algorithm using similarity scores can be used on its own for seed selection, we propose to enhance it by adopting a graph coloring algorithm. Coloring helps us to pick seeds that are better distributed over the network and therefore can lead to improved coverage. First, we propose a basic random coloring method for seed selection based on the randomized distributed coloring algorithm of Luby [18].

Random Coloring (RN) can be directly used for selecting seeds, by picking the nodes which have the same color, for example color c_1. The RN seed selection has some advantages over simply picking seeds at random. It does not require the number of seeds to be picked to be known and it does not pick two neighbors as seeds resulting in fewer redundant communities.

Although basic random coloring can be used for seed selection, we also propose a biased graph coloring algorithm which favors the nodes with high similarity scores to improve the seed selection. The main difference between the biased and the basic coloring is that, in biased coloring, the nodes which are expected to be better seeds with respect to link prediction-based similarity scores pick a specific color, but in basic coloring, random nodes get the specific color.

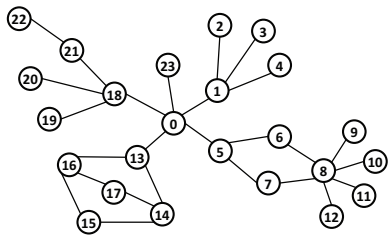
Algorithm 5.2 shows our enhanced seeding algorithm with our biased graph coloring. First each node v calculates its score using a local similarity function, and then assigns the color c_1 to the nodes with the highest score in its egonet, $egonet(v)$. If a node has not received the color c_1 from itself or any of its neighbors, it picks a color for itself at random from the set of available colors. In other words, if a node has the highest score in at least one neighborhood it gets the color c_1, otherwise, it picks a random color. After initialization, each node checks the color of its neighbors, if there is no conflict, the color is confirmed. Otherwise, if the score of the node is less than or equal to the score of its conflicting neighbor, the node picks a new color uniformly at random using *pick_color*. This makes sure that the nodes with high scores preserve their original color c_1.



Method	Seeds
CN	0, 1, 2, 3, 4, 5, 13, 14
HP	0, 1, 2, 3, 13, 14
LHN	0, 1, 2, 3, 7, 13, 14
RA	4, 5, 13, 14
PA	5
MD [13]	4, 5, 6
EC [12]	0, 1, 2, 3, 7, 10, 13, 14
SH(k=3)[5]	4, 5, 6

Method	Seeds
CN + coloring	5, 7, 14
HP + coloring	3, 6, 14
LHN + coloring	3, 10, 14
RA + coloring	5, 8, 14
PA + coloring	5, 8, 14
MD + coloring	4, 6, 14
EC + coloring	0, 9, 13
SH + coloring	0, 6, 14
RN (coloring)	3, 7, 13

(a)



Method	Seeds
CN	-
HP	-
LHN	-
RA	-
PA	0, 8
MD [13]	0, 8, 14, 16
EC [12]	1, 8, 18
SH(k=4)[5]	0, 8, 14, 16

Method	Seeds
CN + coloring	5, 15, 21
HP + coloring	6, 13, 15, 21
LHN + coloring	6, 15, 17, 18
RA + coloring	1, 8, 16, 21
PA + coloring	7, 14, 16, 21
MD + coloring	0, 8, 15, 21
EC + coloring	5, 14, 16, 21
SH + coloring	0, 8, 14, 16, 21
RN (coloring)	6, 15, 21

(b)

Figure 5.1: *Example graphs and the selected seeds using different methods. Biased coloring improves the seed selection.*

The algorithm converges when all the nodes in the network have a confirmed color. After convergence, the nodes which have the color c_1 are selected as the seeds, since these nodes have the highest similarity score in their neighborhood and are expected to be good seeds.

Figure 5.1 shows two scenarios where coloring dramatically improves seed selection[1]. Figure 5.1(a) shows an example where three densely connected communities exist and therefore it is expected that a good seed selection algorithm can pick at least one seed in each community. However, it can be seen that while PA only picks one seed, the others pick many seeds including neighboring nodes. For instance, SH (see Section 5.3.2) which requires the number of seeds k to be known in advance, picks node 4, 5, and 6 which have the highest degree in the network but are directly connected. We can also see that by adding biased coloring, the seed selection improves. For instance, PA combined with coloring selects one seed from each community and the methods which earlier picked many neighbors, now pick fewer seeds which are better distributed across the network.

[1]In practice, due to the randomness in the coloring, the selected seeds are not deterministic. In our experiments section we discuss this topic further.

Figure 5.1(b) shows another example where the neighboring nodes do not have any common neighbors. Therefore, by using the common neighbor-based similarity indices, i.e., CN, HP, LNH, and RA, all the nodes get a similarity score of zero, so our algorithm fails to pick any seeds at all. However, the figure also shows that when adding biased coloring to the local seeding methods, a number of seeds are selected which are well distributed over the graph. In these scenarios, the biased coloring actually works similar to the random coloring, since a node will only receive color c_1 if it has picked it at random.

Time complexity: The time complexity of link prediction-based score calculation is $O(n\Delta)$. Our distributed biased coloring algorithm which is used for enhancing seeding is based on the algorithm by Luby which can run in $O(\log n)$.

5.4.3 Local Community Detection

After selecting the seeds, any type of seed expansion algorithm can be used to identify local communities. In this paper, we use a local algorithm by Yang et al. [8] which uses truncated random walks to approximate personalized PageRank. The main advantages of random walk-based techniques are that they can be computed locally and in parallel, the time and space requirements of such algorithms do not depend on the size of the network [7], and the communities identified by these types of algorithms are structurally close to real-world communities [20].

The algorithm by Yang et al. works as follows. First, the *PageRank-Nibble* algorithm of Andersen et al. [7] is used to compute an approximate personalized PageRank vector starting from the seed node.[2] Then, the algorithm by Spielman and Teng [21] is used to create a collection of sets of nodes. The set which has the first local optima of a scoring function is selected as the final community. The details of the algorithm can be found in [7, 8, 21]. In this study, we have used conductance as the scoring function which has been shown to be good for identifying ground truth communities [8].

Time complexity: The overall complexity of the local community detection algorithm can be approximated with $O(\sum_{i=1}^{k}(vol(C_i)))$, where k is the number of the seeds obtained from the seeding algorithm[3].

5.5 Experimental Results

In this section, we evaluate and compare our local seeding algorithm with other existing algorithms using large scale real-world networks.

[2]The community detection algorithm approximates PageRank with an accuracy value ϵ. In our experiments, we use a constant $\epsilon = 10^{-4}$ for comparing different seeding algorithms, instead of trying to find the accuracy value which leads to the best conductance.

[3]The complexity of PageRank-Nibble, which is the main components of the community detection algorithm, is $O(|S|\frac{\log^3 m}{\phi^2})$, where it can return a community S with conductance $< \phi$.

Table 5.2: *Summary of the networks.*

Dataset	$\|V\|$	$\|E\|$	$\|\mathcal{C}_T\|^*$
Amazon [8]	334,863	925,872	151,037
DBLP [8]	317,080	1,049,866	13,477
Youtube [22]	1,134,890	2,987,624	8,385
LiveJournal [8]	3,997,962	34,681,189	287,512
SoundCloud	5,187,722	36,989,364	N/A

*the number of ground truth communities

5.5.1 Datasets

The networks we have used for this study are listed in Table 5.2. We have selected different types of publicly available real-world datasets. Additionally, we have collected a subset of users from an online social network of a sound sharing website (SoundCloud) and have generated a new network for this study.

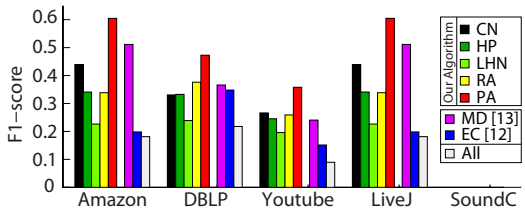
Amazon is a product network in which nodes are products and two products have an edge if they were co-purchased frequently. *DBLP* is a collaboration network where nodes are authors and two authors are connected with an edge if they have co-authored at least one paper. In the *Youtube* and *LiveJournal* networks, the nodes are the users of the video sharing and online blogging websites, respectively, and the edges correspond to friendships. In the *SoundCloud* network, nodes are users and edges correspond to *following* relations.

5.5.2 Comparison

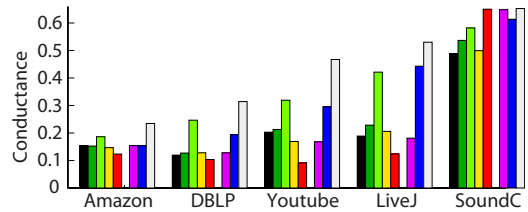
In order to compare the seeding algorithms, we have considered the number of nodes which are selected as seeds by each algorithm, the quality of the identified communities from these seeds, and the number of nodes being covered by these communities.

In order to compare the quality of the identified communities, we use both the conductance of the communities and the similarity with the ground truth communities. The similarity is calculated using the *F1-score* which is defined as $F1\text{-}score = 2\ \frac{precision\ .\ recall}{precision+recall}$, where $recall = \frac{|S \cap C|}{|C|}$, $precision = \frac{|S \cap C|}{|S|}$, and S and C denote the detected and the ground truth community, respectively. The average f1-score over all the communities is used to compare the communities expanded from the seeds by different seeding algorithms.

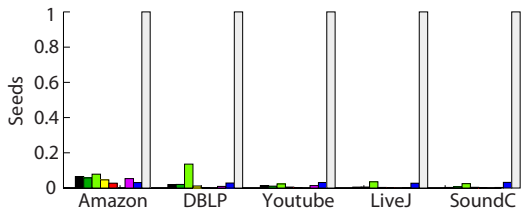
If there is more than one community that overlaps with a ground truth community, we select the one with the highest f1-score, and the duplicate communities are ignored. Moreover, communities which do not have any common nodes with the ground truth communities are not considered in the calculation of the average f1-score. Such communities exist, since there are nodes in the networks which belong to a community but are not annotated to be in the ground truth community, i.e., the networks are "partially annotated" [5].



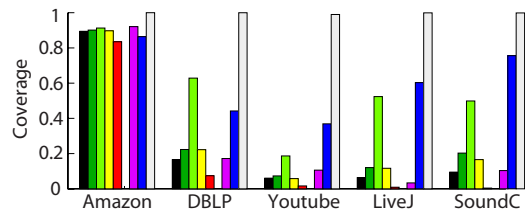
(a) F1-score



(b) Conductance

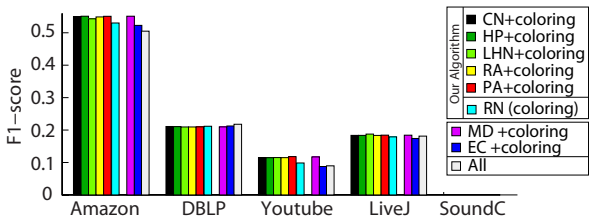


(c) Number of seeds

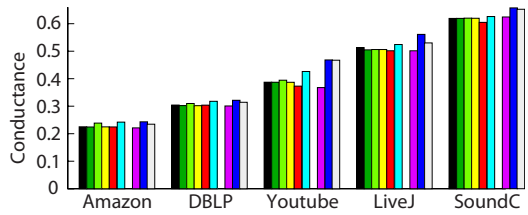


(d) Coverage

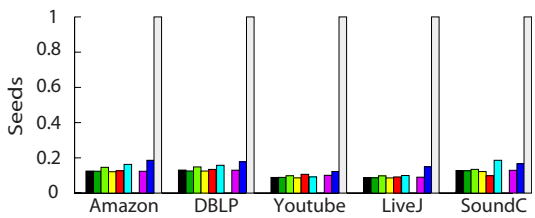
Figure 5.2: *A comparison of different local seeding algorithms and the expanded communities from the selected seeds. CN, HP, LHN, RA, and PA refer to our local seeding algorithm (Algorithm 5.1) using the respective similarity indices (see Table 5.1). EC [12] and MD [13] refer to the local seeding algorithms being compared with our algorithm, and All refers to when all the nodes in the network are used as seeds.*



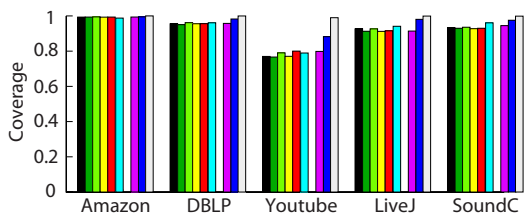
(a) F1-score



(b) Conductance



(c) Number of seeds



(d) Coverage

Figure 5.3: *A comparison of different local seeding algorithms and the expanded communities from the selected seeds. CN, HP, LHN, RA, and PA refer to our local seeding algorithm enhanced with biased coloring (Algorithm 5.2) using the respective similarity indices, and RN refers to our basic random coloring algorithm (see Table 5.1). EC + coloring and MD + coloring refer to existing local seeding algorithms which are also enhanced with our biased coloring algorithm, and All refers to when all the nodes in the network are used as seeds.*

Link Prediction-based Seed Selection

Figure 5.2 shows a comparison of our link prediction-based seeding algorithm (Algorithm 5.1) using similarity indices CN, HP, LHN, PA, and RA (see Table 5.1) with two other local seeding algorithms EC [12] and MD [13] (see Section 5.3.2), as well as when all the nodes in the network are used as seeds (All). It can be seen that PA results in the highest average f1-score and the lowest average conductance for most of the networks being studied. The other four similarity indices used in our algorithm also succeed in selecting a small number of good seeds, which are expanded into high quality communities. However, none of the local seeding methods can achieve a high coverage in all the networks.

Biased Coloring-based Seed Selection

Figure 5.3 shows a comparison of seed selection enhanced with biased coloring, as well as the basic random coloring (RN). It can be seen, that by adding biased coloring, the coverage of the communities is dramatically improved regardless of the similarity index being used. Without biased coloring, our seeding algorithm (Algorithm 5.1) was able to identify a few very high quality communities, but after being enhanced with coloring (Algorithm 5.2), it selects a small number of seeds but now leads to communities with a similar average quality compared to when all the nodes are used as seeds (All). The figure also shows that using biased coloring has improved the coverage of existing local seeding methods, i.e., EC and MD (see Section 5.3.2).

Note that the biased coloring is not deterministic since the color conflicts are resolved at random. Although it is possible to use a deterministic distributed coloring algorithms, e.g., [19], our experiments have shown that the induced randomness does not affect the community detection much and the results are quite stable.[4]

Local versus Global Seeding

The seeding algorithms compared up to this point are all local methods. There are also seeding algorithms which assume that a global knowledge of a network exists, and therefore this knowledge can be used for selecting good seeds. In this study, we include the Spread hub (SH) algorithm [5] which requires the degree of all the nodes in the network to be known and which is shown to select good seeds (see Section 5.3.2). Table 5.3 shows the results using SH for three of the networks.

In addition to the global knowledge, SH requires the minimum number of seeds, k, to be known in advance. Unfortunately, our knowledge of the real community structure of many real networks is very limited, therefore it is not easy to estimate a correct value for k. It can be seen in the table that the selection of k dramatically

[4] In the figures, all the results for the coloring enhanced seeding methods are computed at least 5 times and the figures show the mean values with 95% confidence interval (the error bars were too small to be shown).

affects the quality and the coverage of the communities. The table also shows the community quality and coverage when SH is enhanced with our biased coloring, and it can be seen that coloring can compensate for a bad selection of k. Although the global knowledge is available in this scenario, our experiments show that using local coloring for seed selection is a good and safe choice, since even with a global knowledge of the network, selecting the right number of seeds is not easy.

Table 5.3: *Comparison of SH with different percentage of graph nodes as k*

Dataset	k (% of n)	Seeds	F1-score	Conductance	Coverage
Amazon	3%	0.03	0.50	0.16	0.89
	10%	0.11	0.53	0.20	0.98
	15%	0.18	0.52	0.23	0.99
	3%+coloring	0.11	0.56	0.22	0.99
DBLP	3%	0.03	0.28	0.25	0.83
	10%	0.12	0.23	0.28	0.96
	15%	0.17	0.21	0.30	0.98
	3%+coloring	0.16	0.21	0.30	0.99
Youtube	3%	0.03	0.10	0.40	0.61
	10%	0.10	0.11	0.40	0.87
	15%	0.18	0.10	0.41	0.94
	3%+coloring	0.15	0.10	0.41	0.92

Execution Time

Finally, we have compared the execution time of personalized PageRank-based community detection using our seeding algorithm (PA + coloring) versus running the community detection for all the nodes in the network (All). We have also compared the execution times with an state-of-the-art local overlapping community detection algorithm, DEMON [6], which is based on the idea that different nodes have different views of the communities in their neighborhood and these communities can be merged into the global communities of the network. All the implementations we have used are in Python.[5]

Table 5.4 summarizes the execution times. It can be seen that our seeding algorithm (PA + coloring) is very fast and that the use of seeding dramatically reduces the execution time of the community detection. It can also be seen that our algorithm leads to a better combination of high coverage with good quality communities compared to DEMON.

5.6 Conclusions

In this paper, a novel distributed parameter-free seed selection algorithm is presented which only requires local computations. In our algorithm, we have taken advantage of the similarity indices widely used for link prediction to select a small number of good seeds. We have also enhanced our seeding algorithm with a novel

[5]We have used the implementation of Demon provided by its authors, and have used $\epsilon = 0.3$ and the default minimum community size for the experiments.

Table 5.4: *Execution time*

		Seeding	Community Detection	F1-Score	Conductance	Coverage
Amazon	PA+coloring	52 s	2 h 38 m	0.55	0.22	0.99
	All	-	17 h 15 m	0.51	0.23	1.00
	Demon	-	37 h 40 m	0.51	0.50	0.79
DBLP	PA+coloring	2 m 16 s	1 h 12 m	0.19	0.30	0.96
	All	-	8 h 42 m	0.21	0.31	1.00
	Demon	-	32 h 54 m	0.25	0.63	0.85
Youtube	PA+coloring	7 m 54 s	1 h 38 m	0.12	0.37	0.80
	All	-	14 h 47 m	0.09	0.47	0.99
	Demon	-	52 h 48 m	0.23	0.73	0.23

biased coloring algorithm to further improve the seed selection. The seeds identified by our algorithm have then been expanded into high quality overlapping communities using a personalized PageRank-based community detection algorithm which can also be computed locally.

Experiments using different types of large-scale real-world networks have shown that our seeding algorithm is able to pick nodes that are well-distributed over the networks and are expanded into communities with both high coverage and good quality. Our results also show that using seed selection can dramatically reduce the execution time of community detection while preserving the quality of the identified communities.

Bibliography

[1] Santo Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, Feb. 2010.

[2] Jierui Xie, S Kelley, and BK Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Computing Surveys*, vol. 45, no. 4, 2013.

[3] Aaron Clauset, "Finding local community structure in networks," *Physical Review E*, vol. 72, no. 2, pp. 026132, Aug. 2005.

[4] Sucheta Soundarajan and John E Hopcroft, "Use of Local Group Information to Identify Communities in Networks," *ACM Transactions on Knowledge Discovery from Data (to appear)*, 2014.

[5] Joyce Jiyoung Whang, David F Gleich, and Inderjit S Dhillon, "Overlapping community detection using seed set expansion," in *Proceedings of the 22nd ACM international Conference on information & knowledge management - CIKM '13*. 2013, pp. 2099–2108, ACM Press.

[6] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi, "DEMON: a local-first discovery method for overlapping communities," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. 2012, p. 615, ACM Press.

[7] Reid Andersen and Kevin Lang, "Communities from seed sets," in *Proceedings of the 15th international conference on World Wide Web - WWW '06*. 2006, p. 223, ACM Press.

[8] Jaewon Yang and Jure Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2012, pp. 1–8.

[9] Andrea Lancichinetti and Santo Fortunato, "Community Detection Algorithms: A Comparative Analysis," *Physical Review E*, vol. 80, no. 5, pp. 1–11, Nov. 2009.

[10] Huawei Shen, Xueqi Cheng, Kai Cai, and Mao-Bin Hu, "Detect overlapping and hierarchical community structure in networks," *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1706–1712, Apr. 2009.

[11] Ullas Gargi and Wenjun Lu, "Large-Scale Community Detection on YouTube for Topic Discovery and Exploration," in *Proceedings of the Fifth International Conference on Weblogs and Social Media*. 2011, The AAAI Press.

[12] David F. Gleich and C Seshadhri, "Vertex neighborhoods, low conductance cuts, and good seeds for local community methods," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. 2012, pp. 597–605, ACM Press.

[13] Qiong Chen and Ming Fang, "An Efficient Algorithm for Community Detection in Complex Networks," in *the 6th Workshop on Social Network Mining and Analysis*, 2012.

[14] Bowen Yan and Steve Gregory, "Detecting community structure in networks using edge prediction methods," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2012, no. 09, pp. P09008, Sept. 2012.

[15] Gennaro Cordasco and Luisa Gargano, "Label propagation algorithm : a semi-synchronous approach," *International Journal of Social Network Mining*, vol. 1, no. 1, pp. 3–26, 2012.

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

[17] Michael Luby, "A simple parallel algorithm for the maximal independent set problem," in *Proceedings of the seventeenth annual ACM symposium on Theory of computing - STOC '85*. 1985, pp. 1–10, ACM Press.

[18] Michael Luby, "Removing Randomness in Parallel Computation Without a Processor Penalty," *Journal of Computer and System Sciences*, pp. 162–173, 1988.

[19] Leonid Barenboim and Michael Elkin, "Distributed (Delta+1)-Coloring in Linear (in Delta) Time," in *Procideeings of Symposium on Theory of Computing, STOC'09*, 2009, pp. 111–120.

[20] Bruno Abrahao, Sucheta Soundarajan, John Hopcroft, and Robert Kleinberg, "On the separability of structural classes of communities," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*. 2012, p. 624, ACM Press.

[21] Daniel A Spielman and Shang-Hua Teng, "Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems," in *Proceedings of the 36th annual ACM symposium on Theory of computing - STOC '04*. 2004, p. 81, ACM Press.

[22] Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee, "Measurement and analysis of online social networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 2007, p. 29, ACM Press.

PAPER V

Farnaz Moradi, Ann-Marie Eklund, Dimitrios Kokkinakis, Tomas Olovsson, Philippas Tsigas

A Graph-Based Analysis of Medical Queries of a Swedish Health Care Portal

Proceedings of the 5th International Workshop on Health Text Mining and Information Analysis (Louhi), pp. 2–10, Gothenburg, Sweden, April, 2014.

In order to comply with the thesis layout, this paper has been reformatted.

6 A Graph-Based Analysis of Medical Queries of a Swedish Health Care Portal

Today web portals play an increasingly important role in health care allowing information seekers to learn about diseases and treatments, and to administrate their care. Therefore, it is important that the portals are able to support this process as well as possible. In this paper, we study the search logs of a public Swedish health portal to address the questions if health information seeking differs from other types of Internet search and if there is a potential for utilizing network analysis methods in combination with semantic annotation to gain insights into search behaviors. Using a semantic-based method and a graph-based analysis of word co-occurrences in queries, we show there is an overlap among the results indicating a potential role of these types of methods to gain insights and facilitate improved information search. In addition we show that samples, windows of a month, of search logs may be sufficient to obtain similar results as using larger windows. We also show that medical queries share the same structural properties found for other types of information searches, thereby indicating an ability to re-use existing analysis methods for this type of search data.

6.1 Introduction

Query logs which are obtained from search engines contain a wealth of information about the language used in the logs and the behavior of users. Searching for health and medical related information is quite common, and therefore analysis of query logs of medical websites can give us insight into the language being used and the information needs of the users in the medical domain.

In this study, we analyze 36 months of query logs from a Swedish health care portal, which provides health, disease, and medical information. On one hand, we perform a semantic enhancement on the queries to allow analysis of the language and the vocabulary which has been used in the queries. On the other hand, we perform a graph-based analysis of the queries, where a word co-occurrence graph is generated from the queries. In a word co-occurrence graph each node corresponds

to a word and an edge exists between two words if they have co-occurred in the same query.

Our study reveals that a word co-occurrence graph generated from medical query logs has the same structural and temporal properties, i.e., small world properties and power law degree distribution, which has been observed for other types of networks generated from query logs and different types of real-world networks such as word association graphs. Therefore, the existing algorithms and data mining techniques can be applied directly for analysis of word co-occurrence graphs obtained from health search.

One of the widely studied structural properties of real-world networks is the communities in these networks. In this study, we apply a state-of-the-art local community detection algorithm on the word co-occurrence graph. A community detection algorithm can uncover a *graph community* which is a group of words that have co-occurred mostly with each other but not with the rest of the words in the network. The community detection algorithm used in this study is based on random walks on the graph and can find overlapping communities.

The communities of words, identified from the graph, are then compared with the communities of words obtained from a semantic analysis of the queries. In semantic enhancement, if a word or term in a query exists in medical oriented semantic resources, it is assigned a label. The words and terms which have co-occurred with these labels are used to create a *semantic community*. We have compared the obtained semantic communities with the graph communities using a well-known similarity measure and observed that the communities identified from these two different approaches overlap. Moreover, we observed that the graph communities can cover the vast majority of the words in the queries while the semantic communities do not cover many words. Therefore, the graph-based analysis can be used to improve and complement the semantic analysis.

Furthermore, we study the effect of the time window lengths for analysis of log queries. Our goal is to investigate whether short snapshots of log queries also can be useful for this type of analysis, and how the increase in the size of the log files over time can affect the results.

The reminder of this paper is organized as follows. In Section 6.2 we review the related work. Section 6.3 presents the Swedish log corpus used for this study. Section 6.4 describes the semantic enhancement on the query logs. In Section 6.5 we describe the graph analysis methods. Section 6.6 summarizes our experimental results. Finally, Section 6.7 concludes our work.

6.2 Related Work

In this paper, we study the co-occurrence of words in medical queries and perform both a semantic and graph analysis to identify and compare the communities of related words. In this section, we briefly present a number of related works which deal with analysis of query logs.

```
Q 929C0C14C209C3399CAE7AEC6DB92251 1377986505 symptom brist folsyra hidden:meta:region:00 = 13 1 -N - sv =
Q 2E6CD9E0071057E4BEDC0E52B0B0BDAC 1377986578 folsyra hidden:meta:region:00 = 36 1 -N - sv =
Q 527049C35E3810C45B22461C4CCB2C23 1377986649 kroppens anatomi hidden:meta:region:01 = 25 1 -N - sv =
Q F86B6B133154FD247C1525BAF169B387 1377986685 stroke hidden:meta:region:00 = 320 1 -N - sv =
Q 17CCB738766C545BFE3899C71A22DE3B 1377986807 diabetes typ 2 vad beror på hidden:meta:region:12 = 61 1 -N - sv =
```

Figure 6.1: *Example queries. A query consist of (Q)uery, session ID, time stamp, search query, metadata, number of links returned, the batch ID of the visited link, (N)o spelling suggestions, Swedish search.*

Query logs have been previously studied for identifying clusters of similar queries. In [2] a method was described for clustering similar queries using different notions of query distance, such as string matching of keywords. In [1] clicked Web page information (terms in URLs) was used in order to create term-weight vector models for queries, and cosine similarity was used to calculate the similarity of two queries based on their vector representations.

Several previous works have also dealt with graph analysis of query logs. In [3] several graph-based relations were described among queries based on different sources of information, such as words in the text of the query, clicked URL terms, clicks and session information. In [4] vector space models were compared, by embedding them in graphs, and graph random walk models in order to determine similarity between concepts, and showed that some random walk models can achieve results as good as or even better than the vector models. In [5], it was shown that drawing clusters of synonyms in which pairs of nodes have a strong confluence is a strong indication of aiding two synonymy graphs accommodate each others' conflicting edges. Their work was a step for defining a similarity measure between graphs that is not based on edge-to-edge disagreement but rather on structural agreement.

6.3 Material - a Swedish Log Corpus

The Stockholm Health Care Guide, `http://www.vardguiden.se/`, is the official health information web site of the County of Stockholm, sponsored by the Stockholm County Council and used mostly by people living in the Stockholm area and provides information on diseases, health and health care. In January 2013 the Stockholm County Council reported that vardguiden.se had two million visitors per month. As of November 2013, vardguiden.se and another similar portal, 1177.se (which was a common web site for Swedish regions and counties, and the official national telephone number for health information and advice), are merged into one called 1177 Vårdguiden, sharing the same interface and search engine. The corpus data used in this study consists of the search queries for the period October 2010 to the end of September 2013. The data is provided by vardguiden.se, through an agreement with the company Euroling AB which provides indexing and searching functionality to vardguiden.se. We obtained 67 million queries in total, where

27 million are unique before any kind of normalization, and 2.2 million after case folding. Figure6.1 shows an example of a query log.

Information acquisition from query logs can be useful for several purposes and potential types of users, such as terminologists, infodemiologists, epidemiologists, medical data and web analysts, specialists in NLP technologies such as information retrieval and text mining, as well as, public officials in health and safety organizations. Analysis of web query logs can provide useful information regarding when and how users seek information for topics covered by the site [6]. Such information can be used both for a general understanding of public health awareness and the information seeking patterns of users, and for optimizing search indexing, query completion and presentation of results for improved public health information. For an overview of some common applications and methods for log analysis see [7].

Deeper mining into queries can reveal more important information about search engine users and their language use and also new information from the search requests; cf. [8]. The basis for Search Analytics is made of different kinds of logs of search terms and presented and chosen results by web site users [9]. At a syntactic level queries may contain e.g., synonyms and hyponyms, and to be able to study patterns of search behavior at a more abstract level, we map the syntactic terms to semantic concepts. To our knowledge this is the first of its kind resource for Swedish and as such it can be used as a test bed for experimental work in understanding the breadth and depth of usage patterns, the properties of the resource and the challenges involved in working with such type of data. The only study we are aware of using Swedish log data, in the context of health-related information, is described by [10]. In their study, three million search logs from vardguiden.se (June '05 to June '07) were used for the purpose of influenza surveillance in Sweden, and seven symptoms, roughly corresponding to cough, sore throat, shortness of breath, coryza (head cold), fever, headache, myalgia (muscle pain) were studied.

6.4 Semantic Enhancement

Description of various corpus analytics that enables us to gain insights into the language used in the logs; e.g., terminology and general vocabulary provide, to a certain degree, an indication of the search strategies applied by the users of the web site service from where the logs are obtained. Findings can serve as background work that, e.g., can be incorporated in search engines or other web-based applications to personalize search results, provide specific site recommendations and suggest more precise search terms, e.g., by the automatic identification of laymen/novices or domain experts. The logs have been automatically annotated with two medically-oriented semantic resources [11] and a named entity recognizer [12]. The semantic resources are the Systematized Nomenclature of Medicine — Clinical Terms (SNOMED CT) and the National Repository for Medicinal Products (NPL,

http://www.lakemedelsverket.se/)[1]. We perceive all these resources as highly complementary for our task since the Swedish SNOMED CT does not contain drug names and of course none of the two contain information about named entities.

6.4.1 SNOMED CT and NPL

SNOMED CT provides a common language that enables consistency in capturing, storing, retrieving, sharing and aggregating health data across specialties and sites of care. SNOMED CT provides codes and concept definitions for most clinical areas. SNOMED CT concepts are organized into 18 top-level hierarchies, such as Body Structure and Clinical Finding, each subdivided into several sub-hierarchies and contains around 280,000 terms. More detailed information about SNOMED CT can be found at the International Health Terminology Standards Development Organisation's web site, IHTSDO, at: http://www.ihtsdo.org/snomed-ct/.

The NPL is the official Swedish product registry for drugs and contains 11,250 entries. Every product in the registry contains metadata about its substance(s), names, dosages, producers and classifications, like prescription and Anatomical Therapeutic Chemical codes (ATC). For instance, for the question "missbruk st göranssjukhus" ("abuse st göran hospital") from the query "Q \t C7ED234574EE24 \t 1326104437 \t missbruk st göranssjukhus meta:category:PageType;Article \t = \t 0 \t ..." (here "\t" signals a tab separation), we add three new tab-delimited columns (named entity label, SNOMED-CT, NPL or N/A if no match can be made) to each query. In this case, the three added columns for this particular query will get the labels "FUNCT-ENT", "finding–32709003–missbruk" and "N/A" (no annotation), where the first stands for a FUNCTional-ENTity, the second for a finding category with concept-id "32709003" and "missbruk" as the recommended term.

6.4.2 Semantic Communities

We use the semantic labels obtained from the semantic enhancement to group words into communities. Communities can be used for getting insight into the language and the related words being used for medical search. The words which are matched with the same semantic label are clearly relevant to each other as they belong to the same semantic hierarchy. For each semantic label, we create a set of all the words in the queries which received this label. In other words, the words in queries that co-occurred with the same label are assumed to belong to the same community.

[1]Named entities have not been used for this study. However, we intend to use them in future studies. Nevertheless, the named entity annotation includes the ontological categories location, organization, person, time, and measure entities. Such entities can capture a wide range of entities searched by in such logs such as addresses to health care centers and various health care organizations.

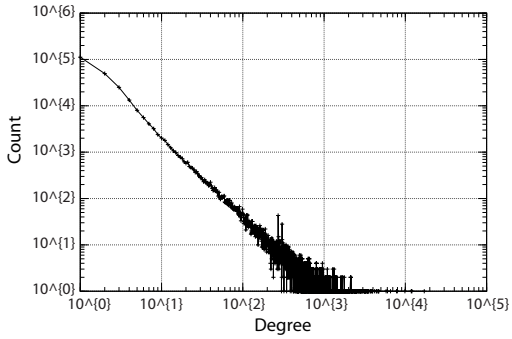


Figure 6.2: *The degree distribution of the co-occurrence graph.*

We have generated such communities only from SNOMED CT and NPL labels and refer to them as *semantic communities* in the rest of the paper. As an example, the community {borrelia, serologiska, blodprover, test, serologisk, testning} was obtained from the queries which received the label "qualifier value–27377004–serologisk".

6.5 Graph Analysis

Query log data can be modeled using different types of graphs [3]. In this study, we have generated a word co-occurrence graph, in which each node corresponds to a word and two nodes are connected with an edge if they have appeared in the same query. The generated graph is undirected and unweighted and has no multiedges. To generate the graph we have used the words as they appeared in the logs, i.e., we did not replace words with their synonyms, correct misspellings, or translate non-Swedish words to Swedish. For example, "eye", "öga", "ögat", "ögon", and "ögonen" appear as five different nodes in the graph but mean the same thing.

The graph $G(V, E)$ generated from the queries which contained two or more words has $|V| = 265{,}785$ nodes and $|E| = 1{,}555{,}149$ edges. The words in one-word queries which did not co-occur with any other words could not be considered for the graph analysis. The generated graph consists of 6,688 connected components. A connected component is a group of nodes where a path exists between any pair of them. The largest connected component of the graph, also known as giant connected component (GCC), contains around 95% of the nodes in the graph.

It was shown in [13], that a graph generated from the co-occurrence of words in sentences in human languages, exhibit two structural properties that other types of complex networks have, i.e, the graph is a *small world* network and it has a *power-law degree distribution* [14]. Later studies on different types of word graphs have also been shown to follow the above properties. In this paper, we also show

Table 6.1: *Structural properties of the word co-occurrence graph over time.*

Time window	$\|V\|$	$\|E\|$	$\|V_{GCC}\|$	clustering coeff.	effective diameter
1 month	16,045	52,403	14,877	0.29	5.47
3 months	30,681	168,045	29,220	0.30	5.42
6 months	48,229	298,331	46,435	0.31	5.38
12 months	69,380	414,643	67,245	0.32	4.97
36 months	265,785	1,555,149	251,597	0.34	4.88

that a word co-occurrence graph generated from medical queries exhibits the same structural properties.

In small world networks, there is a short path connecting any pair of nodes in the GCC of the network. This property can be examined by calculating the *effective diameter* of the network [15]. Small word networks also are highly clustered and therefore have a high *clustering coefficient* value. The effective diameter of our co-occurrence graph is 4.88, and it has an average clustering coefficient of 0.34. These values confirm that our word co-occurrence graph is a small world network.

The degree distribution of the co-occurrence graph is shown in Figure 6.2. It can be seen that the degree distribution follows a power law distribution. This observation is similar to the observations presented by [16] that almost all the measures of a graph generated from query log files follow power laws. Therefore, the user behavior in medical search does not seem different from general search behavior. In addition to networks of word relations, power law degree distributions have also been observed in social, information, and interaction networks where there are many nodes with low degrees and a few nodes with very high degrees [17]. The word with the highest degree in our graph is “barn” (child/children) which has 17,086 edges. Some other high-degree nodes are “sjukdom” (disease), “behandling” (treatment), “ont” (pain), “gravid” (pregnant), and “feber” (fever).

We have also looked into how the structural properties of the word co-occurrence graph change over time as the graph increases in size with an increasing number of queries. Table 6.1 summarizes the results. It can be seen that similar to many other networks, the diameter of the graph shrinks when more nodes become connected and its average clustering coefficient does not change much as the graph becomes larger.

Overall, the structural properties of the word co-occurrence graph are similar to many other real-world networks. Although it was shown in [18] that the queries and information needs of medical practitioners in accessing electronic health records are different from users of general search engines, our analysis reveals that there are similarities between information seeking of general users on health data and on general data. Therefore, the algorithms introduced for analysis of such networks can be directly deployed for analysis of word co-occurrence graphs.

6.5.1 Graph Community Detection

One of the widely studied structural properties of real-world networks is their community structure. A community, also known as a cluster, is defined as a group of nodes in a graph which have dense connections to each other, but have few connections to the rest of the nodes in the network. There have been numerous studies on the community structure of social and information networks and a variety of algorithms have been proposed for identifying the communities in these networks. A thorough overview of different types of community detection algorithms can be found in [19, 20].

Community detection algorithms can be divided into global and local algorithms. The global algorithms require a global knowledge of the entire structure of the network to be able to find its communities. Therefore, these types of algorithms do not scale well for log analysis since query logs are usually very large and are continuously growing. The local algorithms, on the other hand, only require a partial knowledge of the network and therefore can identify network communities in parallel. However, the identified communities might not cover all the nodes in a network.

Moreover, community detection algorithms can be divided into overlapping and non-overlapping algorithms. Traditional partitioning and clustering algorithms typically divide the nodes in a network into disjoint communities. But in many real networks, a node can actually belong to more than one community. For example, in a social network, a user can belong to a community of family members, a community of friends, and a community of colleagues. In a co-occurrence graph, a symptom can co-occur with different types of diseases. Therefore, a community detection algorithm which can identify overlapping communities is more suitable for analysis of the graphs generated from search queries.

For the analysis of log queries, we have used a local overlapping community detection algorithm. This algorithm is a random walk-based algorithm which uses an approximation of a personalized PageRank [21, 22] and is shown to perform well in detecting real communities in social and interaction networks [23]. The algorithm starts from a seed node and expands the seed into a community until a scoring function is optimized. One of the widely used functions for community detection is *conductance*. The conductance of a community C in a graph $G(V, E)$ is defined as $\phi(C) = \frac{\overline{m}(C)}{\min(vol(C), vol(V \setminus C))}$, where $\overline{m}(C)$ is the number of inter-cluster edges and $vol(C) = \sum_{v \in C} deg(v)$ is the volume of a community and corresponds to the sum of the degree of all the nodes in the community. The lower the conductance of a community, the better quality the community has. The complexity of this algorithm is independent of the size of the network and only depends on the size of the target communities.

6.6 Experimental Results

In this section we present our experimental results and discuss the possible applications for graph-based analysis of medical data.

6.6.1 Semantic and Graph Analysis

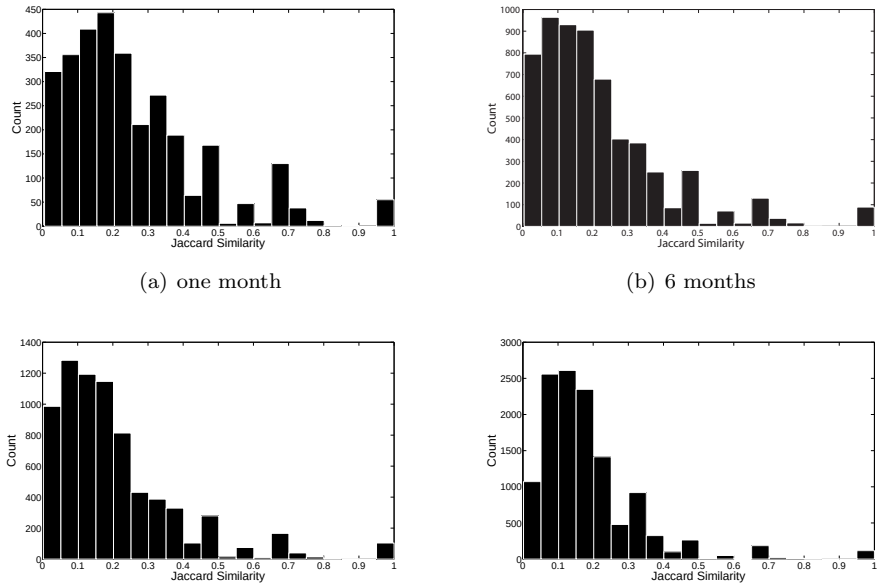
From the semantic enhancement, we have generated 16,427 unique semantic communities which cover less than 11% of the nodes in the network. This means that, the majority of the queries in the network did not contain words that match the medical concepts provided by of SNOMED CT and NPL. This observation suggests that a semantic enhancement of queries on its own is not adequate for understanding the relations between all the words used in medical search.

For the graph analysis, we have used the local overlapping community detection algorithm of [23] to identify the communities from the co-occurrence graph generated from the complete query logs. The algorithm identified 107,765 unique communities in the GCC of the graph with average conductance 0.74. This shows that the communities are not well separated from each other and that there are many edges between distinct communities. Moreover, the identified communities cover 93% of the nodes in the network which means that the graph analysis is more suitable for the study of the relations between the words than the semantic analysis.

The semantic communities and the graph communities are both dependent on the co-occurrence of words in queries, but identify communities differently. The semantic method places the nodes which belong to the same semantic hierarchy together with the words that co-occurred with them in the same community. However, the graph-based method places the words based on the structure of the generated network in the communities.

We have compared and calculated the similarity between the graph communities and the semantic communities using the *jaccard index* which is defined as $JI(C,S) = \frac{|C \cap S|}{|C \cup S|}$. The jaccard index shows the normalized size of the overlap between a graph community C and a semantic community S. Similarity functions, including Jaccard, have been used before for measuring the distance of two different queries. In this study we use similarity to assess the similarity of communities of words obtained from the two distinct methods.

We have compared each semantic community with all the graph communities and show the similarity distribution in Figure 6.3. It can be seen that the majority of the communities partially overlap. As an example, from the word “tandsjukdom” (dental disease) as the seed, we identified the graph community {tandsjukdom, licken, munhåleproblem, rubev, emalj, tändernaamelin, hypopla, permanentatänder, lixhen, hypoplazy, hipoplasy, hypoplazi, bortnött, hipoplazy}. From the semantic enhancement, “tandsjukdom” and “tandsjukdomar” both have received semantic label “disorder–234947003–tandsjukdom”. From the queries which re-



(a) one month

(b) 6 months

(c) 12 months

(d) 36 months

Figure 6.3: *The distributions of jaccard similarity of semantic-based and graph-based communities.*

ceived this label we have generated the semantic community {tandsjukdom, emalj, olika, vanligaste, tandsjukdomar, licken, plack, ovanliga}. The similarity of these communities is low, i.e., 0.16, however, they both contain the words which are clearly relevant to teeth and dental diseases.

As another example, "osteoklast" and "osteoklaster" both receive the semantic label "cell–27770000–osteoklast". From the graph analysis, we have found {osteoklaster, osteoblster, osteocyter, osteoblaster} as a community with "osteoklaster" as the seed. We have also obtained the semantic community {osteoblaster, osteoklast, osteoporos, osteocyter, benskörhet, osteoklaster, osteoblster}. In this example, the graph community is a subset of the semantic community, and their similarity is 0.57. The above examples suggest that a graph-based analysis of medical queries can be used to complement the semantic analysis.

6.6.2 Frequent Co-Occurrence Analysis

In the query logs, we observed that there are many misspellings, meaningless words, etc. In order to clear the dataset, it is common in different studies of log files, to filter out queries which appeared less frequently. By removing such queries, we can dramatically reduce the number of such words.

In this study, we have generated another graph from the words which co-occurred frequently in different queries. We have only considered words that co-occurred five times or more, and the graph contains 32,449 nodes and 217,320 edges, with average clustering coefficient of 0.29 and effective diameter of 5.66.

In the GCC of this graph we found 22,890 graph communities with average conductance of 0.65 and coverage of 95%. Moreover, we have also used the words which co-occurred at least five times to generate the semantic communities. The similarity of these communities with graph communities using jaccard similarity was 0.16 in average which is slightly lower than when no filtering was used. Overall, our observations suggest that filtering can be used to reduce the noise in the datasets and allow us to perform a faster analysis on a smaller graph.

6.6.3 Time Window Analysis

Another property which we have empirically studied in this paper is the effect of time window length during which the queries are analyzed. We have observed that, in average, more than 31% of the nodes and 12% of the edges have re-appeared in each month compared to their previous month. This suggests that the search content changes over time perhaps depending on the changes in the monthly or seasonal information requirements of the users. It also means that over time the size of the word co-occurrence graph increases (see Table 6.1), and since in each month new co-occurrences shape, the graph becomes more and more connected. Therefore, when the time window is long, the analysis requires more time and the identified communities do not have good conductance. When the time window is short, the small size of the graph speeds up the analysis but might affect the analysis result. In this section we investigate the effect of time window length on our analysis.

We started by setting the time window length to one month. From the queries which were observed during each month, we generated a co-occurrence graph and identified the graph communities and the semantic communities. As presented in Section 6.5, the structural properties of a graph generated from one month are quite similar to that of the complete graph. We have also observed that the average conductance of the communities identified by the community detection algorithm is around 0.5 which is lower than when the complete graph was used. This means that the communities in the graphs generated from one month of queries have better quality since they have fewer connections to the rest of the graph.

We observed that the similarities between graph communities and semantic communities are higher when a one-month window is used (in average 0.26). By increasing the length of the time window from one to three, six, twelve, and thirty-six months, we observed a reduction in the similarities (in average 0.23, 0.22, 0.21, and 0.19, respectively). The similarity distributions are shown in Figure 6.3. It seems that with more queries over time, more words get connected and it becomes more difficult to identify good communities. Therefore, using short time windows can improve the quality of the analysis. Moreover, analysis of different time win-

dows can also shed light on how the word relations and user requirements are affected by the months or seasons of the year.

6.6.4 Discussion

Our empirical analysis of a large-scale query log of medical related search presented in this paper can be used to improve our knowledge of the terminology and general vocabulary, as well as the search strategies of the users. In addition to providing a background for language analysis, a potential application for community detection could be to provide better spelling suggestions to users. We have observed that there are communities with very low conductance which contain a number of words which seem to correspond to guessing attempts to find a correct spelling, e.g., {shoulder, froozen, frosen, cholder, sholder, fingers, frozen, scholder, shulder, schoulder, shoulders}. The low conductance of the community means that the community is very isolated and has very few edges outside it and therefore it can easily be cut from the graph. Therefore, the community detection can be used for identifying such cases.

Another potential application of our graph analysis method is to provide recommendations and suggest more precise search terms based on the words that appear in the same community as the keywords entered by the users. For example, since the communities can overlap, each word can belong to more than one graph community or semantic community. We observed that in average, in the complete graph (generated from 36 months of logs), each word belongs to 3.8 unique graph communities and 3.6 semantic communities. It means that a word which can be related to multiple groups of words or have different meanings, can belong to several communities. This knowledge can potentially be used to provide suggestions to the users and help them to select the intended meaning and therefore reducing the ambiguity in the searched queries.

Overall, in this paper, we have presented a promising approach for analysis of medical queries using co-occurrence graphs. As a future work, the following improvements could be of interest for complementing our empirical study:

- Representing different variations of the words with only a single node in the graph, e.g., “öga” for “ögat”, and “ögon”.

- Filtering out the non-medical related words such as person and location entities from the queries based on the semantic enhancement with name entities from NER. Overall, more than 136,000 queries contained a person name entity, and around 127,000 contained a place entity.

- Filtering out high frequency words/terms which do not have medical significance, e.g., “olika” (different).

6.7 Conclusions

Our analysis of a large-scale medical query log corpus is the first step towards understanding the language and the word relations in health/medical related queries. We have performed a semantic enhancement of queries based on medically related semantic resources to find the communities of words which have co-occurred with a semantic label. We have also performed a graph-based analysis of the word co-occurrences and have shown that since a word co-occurrence graph has similar structural properties to many types of real-world networks, existing algorithms for network analysis can be deployed for our study. We then have used a random walk-based community detection algorithm in order to identify communities of words in our graph. Our empirical results show that the communities identified from the semantic analysis and the graph analysis overlap, however the graph-based analysis can identify many more communities and achieves much higher coverage of the words in the queries. Therefore, the graph-based analysis can be used in order to improve and complement the semantic analysis. Our experiments also show that short time window lengths for analysis of query logs, such as a month, would suffice for graph-based analysis of medical queries.

Acknowledgments

We are thankful to Adam Blomberg, CTO, Euroling AB for providing the log data. We are also thankful for the support by the Centre for Language Technology (`http://clt.gu.se`).

Bibliography

[1] Ricardo Baeza-Yates, Carlos Hurtado, and Marcelo Mendoza, "Query Clustering for Boosting Web Page Ranking," in *Advances in Web Intelligence*. 2004, vol. 3034, pp. 164–175, Springer.

[2] Ji-rong Wen, Jian-yun Nie, and Hong-Jiang Zhang, "Clustering user queries of a search engine," in *Proceedings of the tenth international conference on World Wide Web - WWW '01*. 2001, pp. 162–168, ACM Press.

[3] Ricardo Baeza-Yates, "Graphs from Search Engine Queries," in *Theory and Practice of Computer Science*. 2007, vol. 4362, pp. 1–8, Springer.

[4] Amaç Herdagdelen, Katrin Erk, and Marco Baroni, "Measuring semantic relatedness with vector space models and random walks," in *In Proceedings of the TextGraphs-4 (Graph-based Methods for Natural Language Processing)*, 2009, pp. 50–53.

[5] Benoît Gaillard and Bruno Gaume, "Invariants and Variability of Synonymy Networks : Self Mediated Agreement by Confluence," in *Proceedings of the TextGraphs-6 Workshop (Graph-based Algorithms for Natural Language Processing)*, 2011, pp. 15–23.

[6] Judit Bar-Ilan, Zheng Zhu, and Mark Levene, "Topic-specific analysis of search queries," in *Proceedings of the 2009 workshop on Web Search Click Data - WSCD '09*. 2009, pp. 35–42, ACM Press.

[7] Adam Oliner, U C Berkeley, and Archana Ganapathi, "Advances and Challenges in Log Analysis Logs contain a wealth of information for help in managing systems .," *Queue - Log Analysis*, pp. 1–11, 2011.

[8] Olena Medelyan, "Why Not Use Query Logs As Corpora?," in *Proceedings of the Ninth ESSLLI Student Session*, 2004, pp. 1–10.

[9] Mazlita Mat-Hassan and Mark Levene, "Associating search and navigation behavior through log analysis," *Journal of the American Society for Information Science and Technology*, vol. 56, no. 9, pp. 913–934, July 2005.

[10] Anette Hulth, Gustaf Rydevik, and Annika Linde, "Web queries as a source for syndromic surveillance.," *PloS one*, vol. 4, no. 2, pp. e4378, Jan. 2009.

[11] Dimitrios Kokkinakis, "What is the Coverage of SNOMED CT®on Scientific Medical Corpora?," *MIE: XXIII International Conference of the European Federation for Medical Informatics. Studies in Health Technology and Informatics*, vol. 169, pp. 814 – 818, 2011.

[12] Dimitrios Kokkinakis, "Reducing the Effect of Name Explosion," in *In Proceedings of the LREC Workshop: Beyond Named Entity Recognition, Semantic Labelling for NLP tasks. Fourth Language Resources and Evaluation Conference (LREC)*, 2004, pp. 1–6.

[13] R Ferrer i Cancho and R V Solé, "The small world of human language.," *Proceedings. Biological sciences / The Royal Society*, vol. 268, no. 1482, pp. 2261–5, Nov. 2001.

[14] A.L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, no. 5439, pp. 509, 1999.

[15] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos, "Graph Evolution: Densification and Shrinking Diameters," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 2–es, Mar. 2007.

[16] Ricardo Baeza-Yates and Alessandro Tiberi, "Extracting semantic relations from query logs," *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*, p. 76, 2007.

[17] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman, "Power-Law Distributions in Empirical Data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, Nov. 2009.

[18] Lei Yang, Qiaozhu Mei, Kai Zheng, and David a Hanauer, "Query log analysis of an electronic health record search engine.," *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, vol. 2011, pp. 915–24, Jan. 2011.

[19] Santo Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, Feb. 2010.

[20] Jierui Xie, S Kelley, and BK Szymanski, "Overlapping community detection in networks: the state of the art and comparative study," *ACM Computing Surveys*, vol. 45, no. 4, 2013.

[21] Reid Andersen and Kevin Lang, "Communities from seed sets," in *Proceedings of the 15th international conference on World Wide Web - WWW '06*. 2006, p. 223, ACM Press.

[22] Reid Andersen, Fan Chung, and Kevin Lang, "Local Graph Partitioning using PageRank Vectors," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. 2006, pp. 475–486, IEEE.

[23] Jaewon Yang and Jure Leskovec, "Defining and evaluating network communities based on ground-truth," in *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2012, pp. 1–8.